Generalisert algebraisk spesifikasjon med en anvendelse på indirekte semantikk

Jo Erskine Hannay
joh@ifi.uio.no

29. september 1995

Sammendrag

Abstrakte datatyper har lenge representert et paradigme innen program-design og program-verifikasjon. Abstrakt representasjon og spesifikasjon av programmer tillater matematisk resonnering om vesentlige egenskaper ved programmer uten brysomme implementatoriske detaljer. Det har blitt sett at algebraiske metoder kan tas ibruk for abstrakt representasjon og spesifikasjon av programmer, og *algebraisk spesifikasjon* ved *ligninger* tilbyr en abstrakt men konstruktiv måte å spesifisere funksjoner på: Ligninger kan sees på som abstrakte programmer.

Denne hovedoppgaven er bygget rundt følgende to punkter: 1) Vi skal *generalisere* algebraisk spesifikasjon på en måte som åpner for en spesiell type *modulær* spesifikasjon. 2) Vi skal innføre en spesifikasjonsmåte som vi skal kalle *indirekte spesifikasjon*, som en variant av algebraisk spesifikasjon. Vi skal se at indirekte spesifikasjon innfører en *inhomogen* spesifikasjonssituasjon i forhold til vanlig algebraisk spesifikasjon. Den modulære egenskapen ved vår generaliserte spesifikasjonsmåte kan så brukes for å fange opp denne inhomogene spesifikasjonssituasjon. *Konsistens* er et gjennomgangstema i diskusjonen.

Innhold

1 Innledning **1**

2 Abstrakte og formelle datatyper **9**
2.1 Grunnleggende begreper 9
2.2 Algebra 12
2.2.1 Formelt språk 12
2.2.2 Tolking av formelt språk 13
2.2.3 Homomorfier og term-algebraer 14
2.2.4 Full uttrykkbarhet 15
2.2.5 Formulering av matematiske påstander 16
2.2.6 Omskrivningssystemer og ligningslogikk 17
2.2.7 Termunivers 19
2.2.8 Algebraisk spesifikasjon 19
2.2.9 Formell resonnering 21
2.3 Semantikk 22
2.3.1 Implementasjon 22
2.3.2 Initialsemantikk 24
2.3.3 Finalsemantikk 26
2.3.4 Alternativ definisjon av finalsemantikk 30
2.3.5 Initialsemantikk generalisert 31
2.3.6 Konsistens 32
2.3.7 Sammenheng mellom initial- og finalsemantikk 33
2.3.8 Inkonsistens og tolkninger 44
2.3.9 Inkonsistens sett som inkongruens 44
2.3.10 Modulær oppbygging av semantikk 46
2.4 Mot mekanisk resonnering 47
2.4.1 Induktive vs. logiske konsekvenser 47
2.4.2 Konvergente omskrivningssystemer 49
2.4.3 Resolusjonsmetoder for basis-semantikker 50
2.4.4 Knuth&Bendix-komplettering 51
2.4.5 Induktiv komplettering 56
2.4.6 Metoder som søker generell resolusjon 57
2.4.7 Algoritmisk oppdagbarhet av inkonsistens 58
2.5 Oppsummering 61

3 Semantikkgivende syntaktiske funksjoner **63**
3.1 Syntaktiske funksjoner 63
3.2 Funksjonsspesifikasjon over kanoniske representanter 68
3.2.1 Mekanisk generering av kanoniske representanter 69
3.2.2 Eksempler på funksjonsspesifikasjon 71
3.2.3 Utledning og resolusjon 73
3.3 Algebraiske spesifikasjoner av syntaktiske funksjoner 74

3.4 Reduksjon til basis-semantikker 82
3.4.1 Kongruens og indirekte spesifikasjon 83
3.4.2 Formell omgivelse... 84
3.4.3 ...og formelle datatyper 87
3.4.4 Reduksjon til basis-finalsemantikk 88
3.4.5 Reduksjon til basis-initialsemantikk 90
3.5 Inkonsistens i tilknytning til indirekte algebraisk spesifikasjon . . 107
3.6 Komplettering av *id*-utvidelser. 111
3.6.1 *id*-utvidelser og mangel på kjernebevaring 115
3.6.2 Synliggjøring av kjernebevaring 120
3.6.3 Kjernebevaring av *id*-utvidelser og inkonsistens 121
3.6.4 Synliggjøring av kjernesemantikk 122
3.6.5 *id*-utvidelser og inkongruens 123
3.6.6 Nytten av *id*-utvidelser som basis-initialsemantikk spesifikatorer 124
3.7 Skjuling av hjelpefunksjoner 125
3.7.1 Operasjonell skjuling 126
3.7.2 Operasjonell skjuling i resolusjonsmetoder 131
3.7.3 Skjuling på spesifikasjonsnivå 133
3.7.4 Skjuling på spesifikasjonsnivå og reduksjon til basis-initialsemantikk 135
3.7.5 Skjuling ved innføring av typer 136
3.8 Alternativ til *id*-utvidelser under konsistens 138
3.9 Verifikasjon av indirekte algebraiske spesifikasjoner 139
3.9.1 Program-analyse 139
3.9.2 Verifikasjon relativt til en direkte algebraisk spesifikasjon 140
3.10 Andre temaer 141
3.10.1 Generaliserte *id*-utvidelser 141
3.10.2 *id*-utvidelser og finalsemantikk 142
3.10.3 Fikspunkt-semantikk 143
3.11 Oppsummering 144

4 Sekvens-utvidet Knuth&Bendix-komplettering 147
4.1 Definisjon av sekvens-utvidet Knuth&Bendix-komplettering . . . 147
4.1.1 Utledningssekvenser 148
4.1.2 Utvidelsen 151
4.1.3 Godtgjørelse for sekvens-utvidet Knuth&Bendix-komplettering 151
4.2 Anvendelser av sekvens-utvidet Knuth&Bendix-komplettering . . 154
4.2.1 *id*-utvidelser og kjernebevaring igjen 154
4.2.2 Skjuling av hjelpefunksjoner 158
4.2.3 Basis-initialsemantikk og induktive konsekvenser 161
4.2.4 Basis-initialsemantikk og konsistens 162

5 Konklusjon og videre arbeid 163

A Surjektivitet og grunnterm-algebraer 167
A.1 Mer om formelle datatyper og semantikk 167
A.1.1 Bevis av (2.2) side 17 167
A.1.2 Semantikk, initialalgebra og finalalgebra 167
A.1.3 Elementær ekvivalens og isomorfi 169
A.1.4 Semantikk på termer med variable 172
A.2 Induktive konsekvenser og surjektivitet 173
A.3 Omskrivningssystemer og konvergens 175
A.3.1 Bevis for (2.16) side 49 175
A.3.2 Delvis konvergens vs. full konvergens 175

Figurer

1.1 Implementasjon via formelle datatyper 2
1.2 Svak simulering og implementasjon av abstrakt datatype ved moduldeklarasjon 6

2.1 Sammenhenger mellom initial- og final-semantikk 34
2.2 Del-bevisskisse for sats 2.7 36
2.3 Inkonsistens sett på to måter. Inversjon 37
2.4 Inferensregler i Knuth&Bendix-komplettering 54

3.1 Syntaktiske funksjoner 67
3.2 «Uprogrammert» og «pre-programmert» maskin 70
3.3 Illustrasjon til eksempel 42 75
3.4 Möbius-bånd. Sammenbrudd av skillet mellom syntaks og semantikk 77
3.5 Semantikkspesifikasjon på to nivåer 82
3.6 Semantikkspesifikasjon på ett nivå og sammenslåtte funksjonsspesifikasjoner/beskrivelser 83
3.7 Formell omgivelse for reduksjon til basis-semantikker 85

4.1 Inferensregler i sekvens-utvidet Knuth&Bendix-komplettering . . 152
4.2 Figur til eksempel 87 159

Forord

Dette er en hovedfagsrapport for graden Cand. Scient. i informatikk, studieretning databehandling ved Universitet i Oslo. Her er det på sin plass å rette en stor takk til min veileder Olav Lysne for tålmodig, samvittighetsfull og god veiledning. Takk også til Henrik Linnestad og Peter Ølveczky som har funnet tid til å kommentere denne rapporten.

Takk til Marit med sitt nesten uknekkelige humør og morsomme lyder som har holdt ut en frustrert og irritabel Jo, og som har vært 3550 m.o.h. og på mange fjelltopper. Takk til Kjetil og Rune; poeten Kjetil med hans underfundige mentale bilder og Rune med sin morsomme absurde fantasi og sitt forsøk på teselskap på Dyrehaugsryggen. Takk til Peter og hans dyktighet som guide i Egypt. Hadde det ikke vært for at Peter fikk meg til å ta 200-kurs, hadde jeg sikkert aldri tatt sikte på informatikk hovedfag (jeg hadde bare pause fra Musikkhøgskolen). Men takk til de ovenstående først og fremst som mine venner og takk til alle de venner jeg ikke har nevnt. En takk til Skiforeningen og til skogene rundt Oslo som gir mulighet for rask oppladning og gjenoppbygging av fragmenterte hovedfagsstudentsjeler. Jeg vil også få lov til å uttrykke stor takknemlighet for Jotunheimen.

Hva er min presentasjon av kjent stoff og hva er nye tanker? Jeg har ikke skrevet med tanke på å få dette skillet klart frem, fordi jeg finner det distraherende (og faglig uinteressant) å måtte tenke på å fremheve eget åndsverk. Men grovt sett kan man regne at alt fra og med avsnitt 2.3.3 side 26 er (til mitt kjennskap) nytt stoff. Ting som etter dette ikke er nytt stoff er da eksplisitt fremstilt som presentasjon av kjent stoff. I det stoffet som er *før* avsnitt 2.3.3 presenteres stort sett kjente begreper, men jeg nevner at presentasjonen til tider, på både godt og vondt, er *sterkt* preget av *min* følelse for stoffet og den vinkling jeg synes er hensiktsmessig for diskusjonen. Det finnes her også mindre resonnementer og observasjoner som er egne, men som muligens ikke står frem som sådane.

Tilleggskapitlene er egenhendig utført. Noen av resultatene er her kjent stoff, selv om måten jeg har vist disse på muligens er ny. Andre resultater er *sannsynligvis* kjent stoff, men for å spare tid har jeg vist slike resultater selv.

Jeg har i referanselisten tatt med referanser til to arbeidsnotater som jeg har skrevet til veileder underveis. De er tatt med fordi de inneholder ting jeg har tenkt på i forbindelse med stoffet i denne rapporten, og fordi en av dem omhandler noen tanker som tar for seg et morsomt sidetema til diskusjonen.

På side 177 finnes en symboltabell, i tilfelle symbolbruken blir for voldsom. Det er ingen dårlig idé å rive den ut. Bakerst finnes også et stikkordsregister. God lesing!

Oslo, mai 1995 Jo Erskine Hannay

Kapittel 1

Innledning

Vi er, selvsagt, interessert i at de datamaskinprogrammer vi skriver er «riktige». Desverre kan det være svært vanskelig å gå god for at et program av vesentlig størrelse eller kompleksitet er korrekt; i den forstand at det eksekverte programmet alltid gir det output man tilsiktet for ethvert input.

Her er det to viktige sider. For det første kan det meget godt tenkes at man ikke har en presis forestilling om hva man ønsker at et program skal gjøre. For det andre overlates ofte verifikasjon av programmer mht. eventuelle slike presise forestillinger til synsing; eller utelates helt og overlates til testing og debugging (som gjerne varer ut et programs levetid).

Slik «intuitiv» programmering er ofte tilstrekkelig. I kritiske situasjoner kan derimot et programs uforutsette oppførsel være katastrofal.[1] Metoder for presis spesifisering av hva et program gjør eller skal gjøre, samt metoder for på en strukturert måte å resonnere om hvorvidt programmer er korrekte mht. slike spesifikasjoner, er derfor svært viktige. Dette sier ikke at metoder som garanterer «riktigheten» av programmer finnes eller engang er mulige. Ved forskjellige metoder kan imidlertid *visse* egenskaper ved programmer spesifiseres og vises å holde. Presis spesifikasjon er også essensiell for modularisering av programmer og for gjenbruk av moduler; trekk som begge kan lette oversikten i programmeringen og således også bidra til «riktige» programmer.

Matematisk terminologi kan være svært presis, og flere spesifikasjons- og verifikasjonsstrategier bruker matematiske vendinger for å beskrive presist hva et program gjør eller skal gjøre. Felles for mange slike strategier, er å avbilde programtekst (*syntaks*) til en form for meningsfull (*semantisk*) matematisk størrelse som på en eller annen måte sier noe om hva programmet gjør. Å beskrive et program ved *denotasjonell semantikk* (se f.eks. [Mos90]) går eksempelvis ut på å avbilde programmer til matematiske funksjoner. Beskrivelse ved *aksiomatisk semantikk* (ved f.eks. såkalt «Hoare-analyse» [Hoa69]) avbilder predikatlogiske utsagn og programtekst til matematiske påstander om programmet. Vår diskusjon utover er først og fremst tilknyttet aksiomatisk semantikk.

Sentralt for oss er begrepet *datatype*. Programmering innbefatter generelt datastrukturer som variable, lister, trær, grafer, databaser osv. og operasjoner på disse som sortering av lister eller trær, pruning av grafer m.m. For resonneringsformål er det hensiktsmessig å betrakte datastrukturer og deres tilhørende operasjoner på et abstrakt nivå, uten tanke på implementasjonsdetaljer. Dette kan gjøres ved å innse at en programmerer sannsynligvis forestiller seg at datastrukturer lagrer matematiske verdier; i en vid forstand. Vi kan da betrakte en

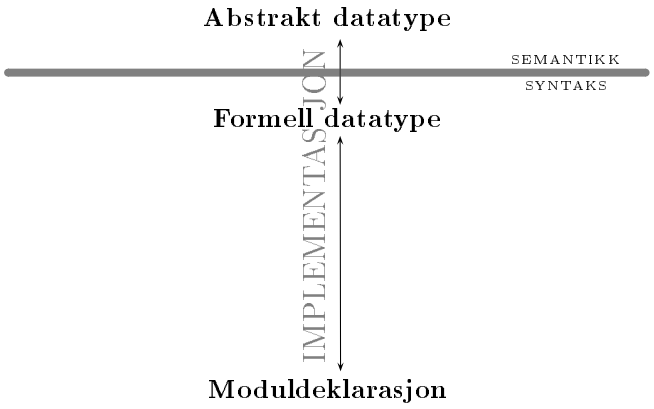
[1] 'Uforutsett oppførsel' har her ingenting med maskiner som «tenker på egenhånd», ei heller med ikke-determinisme å gjøre. 'Uforutsett oppførsel' er snarere en detalj i programmet programmereren ikke visste han programmerte.

mengde matematiske verdier tenkt lagret i datastrukturen og en mengde matematiske funksjoner svarende til operasjonene på strukturen. En samling av en slik matematisk verdimengde, tilhørende funksjonsmengde samt definisjoner av funksjonene, utgjør grovt sett hva vi kaller en *abstrakt datatype*.

En abstrakt datatype er da en «utkrystallisering» av et lettfattelig matematisk objekt som formodentlig søkes implementert i et program. Den virkelige fortreffelighet av abstrakte datatyper vises likevel først når et programmeringsspråk har mekanismer som tilbyr en måte å «implementere» abstrakte datatyper. Eksempelvis tillater *modul*-konstruksjonene i f.eks. SIMULA [DMN70] og Smalltalk [GR83] (modulkonstruksjonene kalles i disse språkene 'klasser'), blant mye annet, at datastrukturer og tilhørende operasjoner kan deklareres og brukes som en enhet i programteksten. Abstrakte datatyper sammen med slike mekanismer oppfordrer da til en *modulær* tankegang ved utvikling av programmer.

Et ideelt scenario for datatype-tankegangen, ville være om enhver programmeringsoppgave kunne resonneres om og uttrykkes utelukkende ved hjelp av abstrakte datatyper. Dette ville gi presis matematisk spesifikasjon på et nivå hvor forstyrrende implementasjonsdetaljer er abstrahert vekk. Siden implementeres de abstrakte datatyper i et (imperativt) programmeringsspråk ved f.eks. modulkonstruksjoner. Programmeringsoppgaven er da korrekt løst i den grad implementasjonene av de abstrakte datatyper er korrekt. I hvilken grad og hvordan det er mulig å «høy»-programmere utelukkende ved abstrakte datatyper skal ikke være vårt tema. Vår diskusjon hører til under problemstillingen korrekt implementasjon av abstrakte datatyper.

Anta vi vil vise at en moduldeklarasjon i et (imperativt) programmeringsspråk er en 'korrekt implementasjon' av en gitt abstrakt datatype. Dette burde kunne gjøres ved strategien å avbilde programtekst på semantiske matematiske objekter; dvs. her fortrinnsvis å avbilde moduldeklarasjoner på abstrakte datatyper. Vi skal gjøre dette, men avbildningen skal gå via noe vi skal kalle *formelle datatyper*. Formelle datatyper fungerer i to roller: Som syntaktiske formelle uttrykk for abstrakte datatyper men også som abstrakte uttrykk for (imperative) moduldeklararsjoner. Formelle datatyper utgjør således et (toveis) *grensesnitt* mellom moduldeklarasjoner og abstrakte datatyper. Formelle datatyper er syntaktiske objekter, så det *semantiske spranget* fra syntaks til semantikk befinner seg her mellom formelle og abstrakte datatyper. Dette spranget er trivielt sammenlignet med et direkte semantisk sprang fra moduldeklarasjon til abstrakt datatype. Se ellers figur 1.1.



Figur 1.1: Implementasjon av abstrakt datatype ved moduldeklarasjon. Implementasjonen går via en formell datatype.

En formell datatype består essensielt av en mengde *termer* og en mengde *ligninger*. Det er dessuten forbundet et *formelt system* kalt *ligningslogikk* til formelle datatyper. Termene er bl.a. ment å være symbolske representasjoner av elementer i verdimengden til en abstrakt datatype. Ligningene sammen med ligningslogikk gir en formell beskrivelse av funksjonene i en abstrakt datatype. Implementasjonsdetaljer er, som i abstrakte datatyper, abstrahert vekk i formelle datatyper.

Det er stor interesse for å mekanisere deler av programverifikasjon. Ved siden av å fungere som grensesnitt mellom moduldeklarasjoner og abstrakte datatyper, befinner i den forbindelse formelle datatyper seg på et abstraksjonsnivå som egner seg for *formell resonnering*.

Ved matematisk resonnering kan man tidvis ta seg i å bedrive «bevisstløs» symbolmanipulasjon etter mer eller mindre implisitte regler. «Bevisstløs» i den forstand at både betydningen til de matematiske symboler som manipuleres og opphavet til reglene de manipuleres etter, er glemt. Formell resonnering er rendyrket symbolmanipulasjon utfra *eksplisitt* gitte regler for hvordan symbolene kan manipuleres. Formelle systemer som predikatkalkyler, ligningslogikk, λ-kalkyle m.m. er beskrivelser av forskjellige typer formell resonnering.

Symboler er i formell resonnering ment å representere matematiske objekter, og for å få noe nytte ut av formell resonnering må resultater av symbolmanipulasjonen (f.eks. et bevistre med aksiomer som blader eller en omskrivning i λ-kalkyle) også tolkes. Men gitt regler og symboler kan selve symbolmanipulasjonen eller «resonneringen» skje uten annen viten om de matematiske objekter symbolene representerer, enn den viten som er nedfelt i reglene. Det er ikke vanskelig å forestille seg mekaniserbare aspekter ved formell resonnering.

Abstrakte maskiner som endelige automater, Pushdown-automater og i ytterste instans Turing maskiner, er matematiske beskrivelser av hva man forestiller seg at 'mekaniske prosesser' er. Abstrakte maskiner egner seg for å bli resonnert matematisk om, og gir da en mulighet for å resonnere matematisk om mekaniske prosesser. (Abstrakte maskiner er på den annen side ofte lite egnet for programmering i praksis.) Også abstrakte maskiner er symbolmanipulatorer, og symboler sett som «maskindeler» er helt strippet for mening: Maskiner kan kun skjelne syntaks eller «form» og ikke semantikk eller «mening». Mange formelle systemer kan naturlig betraktes og beskrives som abstrakte maskiner, noe vi kommer til å gjøre i diskusjonen fremover, siden vi skal være interessert i å *resonnere matematisk om mekaniserbare aspekter ved formell resonnering.* [2]

I forbindelse med formelle datatyper, kreves symbolsk representasjon av elementer i muligens uendelige verdimengder. Vi kunne ha et symbol for hvert element i aktuell verdimengde, men vi skal insistere på å ha abstrakte maskiner med endelig mange «deler». Verdimengder skal derfor betraktes som generert av endelig mange *generatorfunksjoner*. F.eks. kan de naturlige tall genereres av konstanten 0 og etterfølgerfunksjonen *succ*. Dersom symbolene $\mathsf{0}$ og succ representerer 0 og *succ* hhv., kan ethvert naturlig tall representeres av en term på formen $\mathsf{succ}(\mathsf{succ}(\cdots(\mathsf{0})\cdots))$. Vi betrakter således kun abstrakte datatyper med

[2] At en oppgave kan mekaniseres betyr i praktisk forstand at et datamaskinprogram kan skrives for å utføre oppgaven. Tilsynelatende er vi tilbake i programmeringsverdenen igjen, snarere enn i den matematiske. I siste instans er det jo ønskelig å implementere et formelt system på en datamaskin. Men vårt fokus her på formelle systemer og abstrakte maskiner er primært å demonstrere *at* en gitt resonneringsprosess er mekaniserbar. Derfor holder vi oss (eller befinner oss fortsatt) på det matematiske nivå med abstrakte matematiske beskrivelser i form av formelle systemer og abstrakte maskiner.

rekursivt tellbare (mekanisk beregnbare) verdimengder. Vi skal også kun betrakte abstrakte datatyper med endelige funksjonsmengder. *Termuniverset* — mengden av alle representasjoner av funksjonsapplikasjoner — sikres da også å være rekursivt tellbart. Mengden av *generatortermer* i en formell datatype utgjør *generatoruniverset* som representerer verdimengden i en gitt abstrakt datatype. I eksemplet med de naturlige tall er generatoruniverset i et en-til-en forhold med den korresponderende verdimengden. Dette er ikke alltid tilfellet. En naturlig måte å generere f.eks. de hele tall på, er med generatorfunksjonene 0, *succ* og forgjengerfunksjonen *pred*. Da vil f.eks. termene succ(pred(0)) og 0 representere samme element i verdimengden.

Den type formelle resonnering vi skal befatte oss med er avleding av ligninger fra gitte initialligninger eller aksiomer. Det grunnleggende formelle systemet skal være det som er forbundet med formelle datatyper; nemlig ligningslogikk. Ligningslogikk kan gis en spesielt enkel og intuitiv beskrivelse som direkte formaliserer den velkjente matematiske resonneringen vi gjør når vi resonnerer utfra ligninger. Eksempelvis kan vi fra Peano-aksiomene

$$\mathsf{x{+}0 = x} \quad \text{og} \quad \mathsf{x{+}succ(y) = succ(x{+}y)}$$

for addisjon i de naturlige tall, avlede ligningen

$$\mathsf{succ(0){+}succ(succ(0)) = succ(succ(0)){+}succ(0)}$$

ved å starte på hhv. venstre og høyre side i ligningen og erstatte «like for like» til vi får syntaktisk like termer på følgende måte:

$$\begin{array}{lcl} \mathsf{succ(0){+}succ(succ(0))} & & \mathsf{succ(succ(0)){+}succ(0)} \\ \quad \rightsquigarrow \mathsf{succ(succ(0){+}succ(0))} & & \\ \quad \rightsquigarrow \mathsf{succ(succ(succ(0){+}0))} & \equiv & \mathsf{succ(succ(succ(0){+}0))} \rightsquigarrow \end{array}$$

En formell spesifikasjon av en funksjon kan utrykke en egenskap ved funksjonen; f.eks. kommutativitet $x+y = y+x$. Vårt mål er imidlertid implementasjon av abstrakte datatyper, så vi skal være interessert i å finne *konstruktive* spesifikasjoner av funksjoner som forteller hvordan verdien av en funksjonsapplikasjon evalueres. Aksiomene over for addisjon i naturlige tall er konstruktive i så måte. F.eks. kan man «regne ut» $2+1$ ved å avlede:

$$\mathsf{succ(succ(0)){+}succ(0)} \rightsquigarrow \mathsf{succ(succ(succ(0){+}0))} \rightsquigarrow \mathsf{succ(succ(succ(0)))}$$

I tillegg til å implementere funksjoner i en abstrakt datatype, er det svært relevant å implementere likhets-/identitetsrelasjonen for datatypen. (Dette er relevant for betingede kontrollstrukturer i programmeringsspråket, men også for formell resonnering.) I den forbindelse kan det gis ligninger som gir likhet mellom generatortermer i tilfellet mange-til-en generatorunivers. For generatoruniverset for de hele tall som over kan vi eksempelvis gi ligningene

$$\mathsf{succ(pred(x)) = x} \quad \text{og} \quad \mathsf{pred(succ(x)) = x}$$

for å utrykke hvilke generatortermer som skal forstås som like.

Formelle datatyper har altså et formelt språk og et formelt system knyttet til seg. Dette språket kan sies å være et abstrakt programmeringsspråk i hvilket abstrakte programmer kan skrives. Slik kan formelle datatyper abstrahere moduldeklarasjoner skrevet i et mer konkret programmeringsspråk.

Alle som har drevet med formelle systemer eller matematisk bevisføring overhodet vet at det ofte er forbundet strategiske vanskeligheter med hvordan man skal gå frem underveis. Ligningslogikk har dette iboende i og med at det ofte

er flere ligninger som er anvendbare ved et gitt bevissteg. Et uheldig valg fører kanskje ikke fram, eller kanskje bevisføringen går i det uendelige. Noen ligninger er imidlertid slik at de kan orienteres og alltid brukes én vei og likevel beholde den samme bevisstyrken. Videre kan de ha egenskapen at enhver utledning er endelig og forskjellige valg av anvendte ligninger alltid fører til samme resultat. Slike ligninger utgjør det vi kaller et *konvergent omskrivningssystem* og er essensielle for mekaniserbarheten av ligningslogikk; og derved for mekaniserbarheten av vår formelle resonnering i formelle datatyper. Ligningene succ(pred(x)) = x og pred(succ(x)) = x orientert mot høyre utgjør et konvergent omskrivningssystem. I lys av at *ikke-determinisme* er latent i ligningslogikk, utgjør konvergente omskrivningssystemer en *deterministisk* del av ligningslogikk; i den forstand at konvergente omskrivningssystemer sett på som abstrakte deterministiske programmer gir samme output for et gitt input.

Formelle datatyper representerer et «minste steg» fra abstrakte datatyper inn i den syntaktiske formelle verden. Ved programutvikling tilbyr således formelle datatyper en meget presis uttrykksform for matematiske idéer som søkes implementert, uten forstyrrende og på dette nivå uvesentlige detaljer. Det finnes programmeringsspråk som søker å forene slik høynivå spesifikasjon ved formelle datatyper og mer implementatoriske mekanismer. Eksempler på slike formelle spesifikasjons- og programmeringsspråk er ABEL [DO91] og OBJ [GW88]. Merk også at *applikative* programmeringsspråk kan sies å ligge nær opp til abstraksjonsnivået til formelle datatyper.

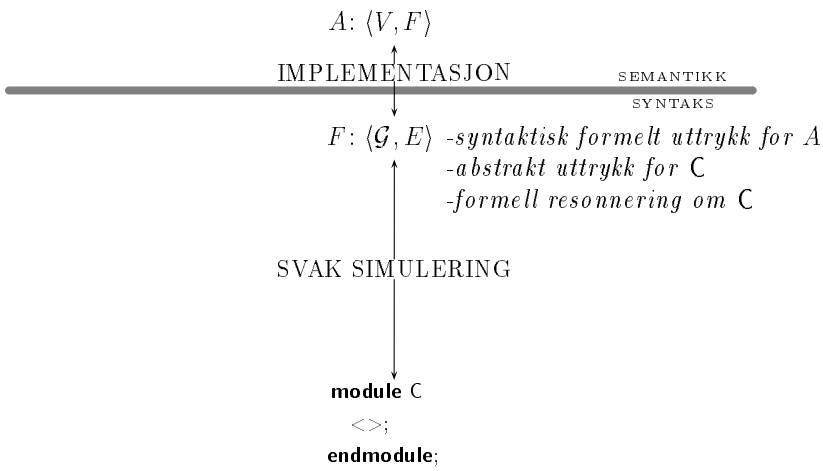
Verifikasjonen av at en moduldeklarasjon implementerer en abstrakt datatype består nå av to hoveddeler. For det første må et samsvar mellom den (imperative) moduldeklarasjonen og den korresponderende formelle datatypen vises. Dette er beskrevet utfyllende i [Dah92]. (En modul kan vises å *imperativt simulere* en formell datatype. Dette gjøres ved å identifisere en såkalt *typeekvivalent* til modulen, som så vises å *simulere* den formelle datatypen. Sentralt er å finne *effektfunksjoner* som beskriver (funksjons-)prosedyrers effekt på datastrukturen i modulen. 'Hoare-logikk' [Hoa69] kan brukes til å vise korrespondanse mellom effektfunksjon og (funksjons-)prosedyre.) Dette samsvaret kalles *svak simulering* og tar bl.a. i betraktning motsetningen mellom fysiske begrensninger på en fysisk datamaskin og den abstraherte og idealiserte situasjon forbundet med teoretiske abstrakte maskiner.

Gitt at en moduldeklarasjon således simulerer en formell datatype, tilbyr den formelle datatypen muligheten å resonnere formelt om modulen: En modul som tenkes å implementere naturlige tall og addisjon, har kanskje en prosedyre **pluss(x nat, y nat)** som vises å ha de abstrakte egenskaper beskrevet ved Peanoaksiomene over. Ved formell resonnering (ved f.eks. såkalt *generatorinduksjon*; ren ligningslogikk er her for «svak») kan vi avlede x+y = y+x, som da gir i en passende forstand at **pluss(m,n)** = **pluss(n,m)**, for alle aktualparametre **m,n** av type **nat**; under forutsetning av at evaluering av applikajonene av **pluss** ikke fører til run-time feil.

Slike resultater om en modul kan brukes i Hoare-analyse av progamtekst der instanser av modulen blir brukt; det være i et evt. hovedprogram eller f.eks. i deklarasjonen av en annen modul.[3]

Den andre delen av verifikasjonsoppgaven består i å vise at den formelle

[3]De fleste programmeringsspråk tilbyr jo datatypen 'naturlige tall med addisjon' ferdig implementert med korrekt semantikk. Med mekanismer for implementasjon av abstrakte datatyper, kan imidlertid programmereren implementere egne datatyper, og det er da nødvendig å etablere «oppførselen», eller semantikken, til disse.



Figur 1.2: Svak simulering og implementasjon av abstrakt datatype A ved moduldeklarasjon C og formell datatype F.

datatypen i en presis forstand er en *implementasjon* av den abstrakte datatypen. Kriteriet vi skal bruke utrykkes i begrepet *induktiv ekvivalens*, og ligningslogikk er sentral for dette. I tilnærmelser til induktiv ekvivalens, vil formell resonnering være sentral. Figur 1.2 oppsummerer litt av fremstillingen så langt.

Vi skal ikke snakke mer om samsvar mellom moduldeklarasjoner og formelle datatyper (nedre del av figur 1.2). Vår diskusjon skal dreie seg om formelle datatyper og formell resonnering og om samsvar mellom formell- og abstrakt datatype (øvre del av figur 1.2).

Ligninger deler termuniverset opp i klasser; hver klasse bestående av termer som kan vises «like» ved ligningslogisk utledning. En slik klasseinndeling kan vi kalle å *gi semantikk* til termer. Det at en formell datatype implementerer en abstrakt datatype, hviler da i essens på en riktig klasseinndeling av termuniverset. For formell resonnering er det sentralt at semantikk spesifiseres på en måte som er tilnærmbar for formelle systemer. Ligninger og ligningslogikk er som sagt våre midler her.

Ligningslogisk omskrivning til syntaktisk like termer er én måte å gi semantikk på. Det finnes andre måter å gi semantikk på enn ved slik direkte omskrivning. Især for semantikkgiving av generatortermer fra mange-til-en generatorunivers, finnes det en rekke teknikker foruten den direkte metoden. Eksempelvis kan man definere semantikk på basis av *observatorfunksjoner* ved å spesifisere at to generatortermer g, g' er semantisk like hvis og bare hvis $\mathsf{h}(g) = \mathsf{h}(g')$ for gitte observatorsymboler h, er utledbart fra gitte ligninger.

En av de større diskusjonene i denne hovedoppgaven tar for seg nok en metode: En utledning fra en term g i et konvergent omskrivningssystem gir alltid en unik normalform. Et konvergent omskrivningssystem kan således sies å *beregne* en syntaktisk funksjon *synt* som gitt en term g, gir dennes normalform. En slik funksjon gir for hver term g en unik representant for klassen til g ifølge semantikken gitt av ligningene i omskrivningssystemet. Vi kan derfor gi semantikk til termer ved å bestemme at to termer g, g' skal forstås som like hvis og bare hvis

$$synt(g) = synt(g') \qquad (*)$$

dvs. dersom deres normalformer er like. Sålangt er jo dette ikke annet enn semantikken gitt direkte av ligningene i omskrivningssystemet som implementerer *synt*. Imidlertid kan semantikkgiving etter prinsipp $(*)$ generaliseres til semantikkgivende syntaktiske funksjoner som ikke er beregnbare ved konvergente omskrivningssystemer i den forstand som over.

En syntaktisk funksjon som implementeres av et konvergent omskrivningssystem, gir altså for hver term g en unik representant for klassen til g. Denne representanten er selv et medlem av klassen; en egenskap vi essensielt skal la kvalifisere til navnet *kanonisk representant* for klassen. Alle syntaktiske funksjoner implementert av konvergente omskrivningssystemer i ovenstående forstand, er slike kanonisk-representant funksjoner. Men ikke alle kanonisk-representant funksjoner kan implementeres på den måten: En vanlig symbolsk representasjon for mengder av naturlige tall, er sekvenser induktivt bygget opp av symbolene 0, succ, Ø og add. Mengden $\{0,1\}$ kan f.eks. representeres av termen add(add(add(Ø,succ(0)),0),succ(0)). Det er klart at vi her har et mange-til-en generatorunivers, og den ønskete semantikk (klasse-inndeling) til disse termer gis av ligningene

$$\mathsf{add(add(s,x),x)} = \mathsf{add(s,x)} \quad \text{og} \quad \mathsf{add(add(s,x),y)} = \mathsf{add(add(s,y),x)}$$

For en klasse, kunne en naturlig kanonisk representant være den sorterte termen uten repetisjon; for klassen til termen over: add(add(Ø,0),succ(0)). Det finnes imidlertid intet konvergent omskrivningssystem som implementerer en syntaktisk funksjon som gitt en «mengde»-term, gir den kanoniske representanten til termens klasse. Grunnen til dette ligger i at ligningen

$$\mathsf{add(add(s,x),y)} = \mathsf{add(add(s,y),x)}$$

ikke kan orienteres uten at beviskraft mistes. Utbygginger av ligningslogikk som f.eks. betingete og ordnede omskrivningssystemer kan dog bøte på orienterbarhetsproblemet i akkurat dette tilfellet.

Syntaktiske funksjoner som ikke er kanonisk-representant funksjoner, kan også gi semantikk til termer ved prinsipp $(*)$. Også slike syntaktiske funksjoner går ut over det som er implementerbart av konvergente omskrivningssystemer i ovennevnte forstand.

Generelle semantikkgivende syntaktiske funksjoner løfter idéen om semantikkgiving ved omskriving til syntaktisk like termer *ut av* det begrensende domenet til ren ligningslogikk. I lys av orienterbarhetsproblemet over, kan styrken til betingete og ordnede omskrivningssystemer også sies å være innlemmet i generelle semantikkgivende syntaktiske funksjoner.

Som andre funksjoner, kan semantikkgivende syntaktiske funksjoner spesifiseres ved ligninger og ligningslogikk. Ligninger som spesifiserer semantikkgivende syntaktiske funksjoner gir opphav til det vi skal kalle *indirekte spesifikasjon* av semantikk. Indirekte spesifikasjon spesifiserer semantikk på en måte avledet fra vanlige måter ligninger gir semantikk på. Via indirekte spesifikasjon innplantes dermed ligningslogikk med kraften til *generelle* semantikkgivende syntaktiske funksjoner.

Klassen av semantikkgivende syntaktiske funksjoner er altså større enn klassen av funksjoner implementerbare av konvergente omskrivningssystemer. Indirekte spesifikasjon har derfor et potensiale for å utvide området av semantikkspesifisering som det er mulig å få mekanisk grep på ved konvergente omskrivningssystemer (i sin enkleste versjon).

Vi skal se at spesifikasjon av semantikkgivende syntaktiske funksjoner begrepsmessig kan forestilles å være på et «meta»-nivå forskjellig fra den abstrakte datatypen om hvilken det i utgangspunktet skal resonneres. Vi må i prinsipp da

behandle indirekte spesifikasjon *separat* fra ligninger som definerer funksjoner (i den opprinnelige abstrakte datatypen). Vanlige måter å spesifisere semantikk på er ikke tilstrekkelige for å uttrykke slik *sammensatt* semantikk. Vi utvikler derfor nye *generaliserte* semantikker. Dette er en annen hoveddiskusjon i denne oppgaven. Vår generaliserte semantikk har en *modulær* oppbygning, og åpner for en modulær oppbygging av formelle datatyper. Denne modularitet er for oss essensiell når indirekte spesifikasjon skal sees som del av en større spesifikasjonsomgivelse.

En stor del av diskusjonen skal også gå til å vise at indirekte spesifikasjon, under visse interessante forutsetninger faktisk *kan* integreres med ligninger for vanlige funksjoner.

Indirekte spesifikasjon skal vise seg å være et interessant supplement til algebraisk spesifikasjon. Ett særtrekk ved indirekte spesifikasjon er en *operasjonell* spesifikasjonsstil som faktisk gjør spesifikasjoner tilgjengelige for programverifikasjons-metoder.

Konsistens er et gjennomgangstema. Konsistens er relatert til grunnleggende oppfatninger om den semantiske verden som nedfelles som *forutfatninger* i semantikkspesifikasjon og som ikke ønskes rørt ved. Eksempelvis nedfelles vår oppfatning om at en matematisk påstand ikke både kan være sann og gal i predikatlogikk ved predefinerte (forutfattede) tolkninger av symbolene true og false, samt ved å kalle en aksiommengde for *inkonsistent* dersom true = false er utledbart fra mengden. (In)konsistens må således alltid sees *relativt* til slike forutfatninger.

Vi skal sammen med utviklingen av generaliserte semantikker, utvikle tilhørende konsistensbegreper. Vi skal se hvordan inkonsistens kan oppdages og hvordan konsistens kan etableres.

Vi skal dessuten innføre begrepet *kunstig* inkonsistens. Dette er inkonsistens med opphav i *hjelpefunksjoner*; funksjoner som er nyttige hjelpemidler under implementasjon men som begrepsmessig kan sees ikke å høre hjemme på det semantiske plan. Vi skal søke å *skjule* hjelpefunksjoner og dermed kunstig inkonsistens fra den formelle resonnering, slik at de implementatoriske fordeler ved bruken av hjelpefunksjoner likevel kan utnyttes.

*

Universell algebra fanger inn mange begreper relatert til formelle og abstrakte datatyper. I kapittel 2 presenteres noen for oss vesentligere deler av universell algebra, innledet av en kort oppsummering av grunnleggende matematiske begreper. Presentasjonen av algebra går så over i et avsnitt om semantikk. Vi utvikler generaliserte måter å spesifisere semantikk på vha. ligningslogikk — vårt grunnleggende formelle system. Vi utvikler også konsistens-begreper knyttet til disse spesifikasjonsmåtene. Kapitlet avsluttes med mekanisering av ligningslogikk og relevante formelle systemer bygget på ligningslogikk for formell resonnering i formelle datatyper.

I kapittel 3 tar vi så for oss semantikkgivende syntaktiske funksjoner. Vi lar slike funksjoner motiveres som en eksplisittgjøring av konvergente omskrivningssystemer. Vi generaliserer den semantikkgivende egenskapen og anerkjenner kanonisk-representant funksjoner som et viktig spesialtilfelle. Vi studererer så spesifikasjoner av semantikkgivende syntaktiske funksjoner og bruker disse i indirekte spesifikasjon. Disse benyttes i anvendelser av semantikkbegrepene utviklet i kapittel 2. I dette kapitlet utvikles også begrepet kunstig inkonsistens, og vi utvikler teori og antyder metoder for skjuling av kunstig inkonsistens.

I kapittel 4 definerer vi en utvidelse av *Knuth&Bendix-komplettering*. Denne utvidelsen er interessant i sammenheng med *konstruktive* bevis. Vi relaterer dessuten denne utvidelsen til temaer i kapittel 2 og kapittel 3.

Kapittel 2

Abstrakte og formelle datatyper

I dette kapitlet defineres begrepene *abstrakt datatype* og *formell datatype*. Vi etablerer hva vi mener med at en formell datatype er en *korrekt implementasjon* av en abstrakt datatype.

Vi presenterer forskjellige måter å definere formelle datatyper på ved hjelp av ligningslogikk. Slike ulike måter fremkommer ved ulike metoder for å dele opp et termunivers i klasser, slik at hver klasse består av termer vi ønsker skal representere samme matematiske objekt i en abstrakt datatype.

Kjente måter som initialsemantikk og finalsemantikk presenteres. Vi innfører *generaliseringer* av initialsemantikk og finalsemantikk. Disse generaliseringer gir oss et begrepsapparat som gjør oss istand til å uttrykke vesentlige egenskaper ved spesifikasjonen av formelle datatyper.

Konsistens er her en sentral egenskap. Ved hjelp av generalisert initial- og finalsemantikk, knytter vi konsistens til en *intensjon* bak måtene å spesifisere forskjellige formelle datatyper på.

Våre generaliseringer av initial- og finalsemantikk er også en forberedelse til diskusjonen i kapittel 3.

Vi er interessert i formell men også *mekanisk* resonnering. Kapitlet avsluttes med en presentasjon av sentrale begreper for mekanisk resonnering.

2.1 Grunnleggende begreper

Vi oppfrisker her noen grunnleggende matematiske begreper som brukes utover i den videre diskusjon. Vi bygger det hele opp ved begrepet *mengde*.

Mengder

Vi antar en intuitiv forståelse av begrepet mengde. Vi antar også en grunnleggende kjennskap til notasjoner og begreper knyttet til mengder. Vi skal her likevel gi noen avledninger og konvensjoner som vi kommer til å bruke i det følgende.

Vi skal alltid (implisitt eller eksplisitt) se matematiske objekter som elementer i mengder. Dersom a, a' betegner samme element i en mengde, skal vi si de er ***identiske***. Vi skriver da $a = a'$. Vi bruker denne notasjon kun for slik elementidentitet. Vi skriver $a \neq a'$ når a og a' ikke er identiske.

n-tupler

Et n-***tuppel*** $\langle a_1, \ldots, a_n \rangle$ for et naturlig tall n, er en ***ordnet*** mengde[1]. Et m-tuppel $\langle b_1, \ldots, b_m \rangle$ og et n-tuppel $\langle a_1, \ldots, a_n \rangle$ er identiske hvis og bare hvis $m = n$ og $a_i = b_i$ for $1 \leq i \leq n$. Vi kaller a_i for i-***te komponent*** i $\langle a_1, \ldots, a_n \rangle$. En ***sekvens*** q er et n-tuppel for en uspesifisert n og der n er ***lengden*** av q. En sekvens kan dessuten ha uendelig lengde. Vi betegner sekvenser av lengde 0 med ε. Vi betegner lengden av en sekvens s med len_s.

For vilkårlige mengder $A_1, \ldots, A_n$, er mengden betegnet $A_1 \times \cdots \times A_n$ — ***det n-foldige kartesiske produkt*** av $A_1, \ldots, A_n$ —mengden av n-tupler $\langle a_1, \ldots, a_n \rangle$ slik at $a_1 \in A_1, \ldots, a_n \in A_n$. Dersom alle $A_1, \ldots, A_n$ er identiske med en mengde A, betegner vi ofte $A_1 \times \cdots \times A_n$ med A^n, der $A^0 = \emptyset$.

Relasjoner

En n-***ær relasjon*** $\Re$ på $A_1, \ldots, A_n$ for et naturlig tall n, er en delmengde av $A_1 \times \cdots \times A_n$ for gitte mengder $A_1, \ldots, A_n$. Dersom $A_1, \ldots, A_n$ alle er identiske med en mengde A, sier vi bare at $\Re$ er en relasjon på A og kaller da A ***relasjonsdomenet*** (eller bare ***domenet***) til $\Re$. ***Restriksjonen*** av en relasjon $\Re \subseteq A_1 \times \cdots \times A_n$ til $A'_1 \times \cdots \times A'_n$ (eller til A' for $A'_1 = \cdots = A'_n = A'$) for $A'_1 \subseteq A_1, \ldots, A'_n \subseteq A_n$, er relasjonen $\Re \cap A'_1 \times \cdots \times A'_n$ og betegnes $\Re_{A'_1 \times \cdots \times A'_n}$ (eller $\Re_{A'}$ for $A'_1 = \cdots = A'_n = A'$).

Vi skal i vår diskusjon fremover snakke mye om binære (2-ære) relasjoner. La $\Re \subseteq A \times B$ og $\Re' \subseteq B \times C$ være vilkårlige binære relasjoner for vilkårlige mengder A, B, C. $\Re$ sin ***invers*** $\Re^{-1}$, er relasjonen slik at $\langle b, a \rangle \in \Re^{-1}$ hvis og bare hvis $\langle a, b \rangle \in \Re$. Enhver binær relasjon har en entydig invers. ***Sammensetningen*** $\Re \circ \Re'$ av $\Re$ og $\Re'$ er relasjonen $\{\langle a, c \rangle \mid \exists b \in B \mid \langle a, b \rangle \in \Re \text{ og } \langle b, c \rangle \in \Re'\}$. For $\Re \subseteq A \times A$ kan vi definere $\Re^0 = \{\langle a, a \rangle \mid a \in A\}$, $\Re^1 = \Re$ og $\Re^n = \Re^{n-1} \circ \Re; n > 1$. (Denne notasjonen skal ikke forveksles med den for kartesiske produkt.)

La $\Re$ være en vilkårlig binær relasjon på A for en eller annen mengde A (altså $\Re \subseteq A \times A$). $\Re$ kalles

refleksiv dersom $\langle a, a \rangle \in \Re$, for alle $a \in A$.

symmetrisk dersom $\langle a, b \rangle \in \Re \quad \Rightarrow \quad \langle b, a \rangle \in \Re$. (Dvs. $\Re = \Re^{-1}$.)

transitiv dersom $\langle a, b \rangle \in \Re$ og $\langle b, c \rangle \in \Re \quad \Rightarrow \quad \langle a, c \rangle \in \Re$.

En ***ekvivalens***-relasjon er en (binær) relasjon som er refleksiv, symmetrisk og transitiv.

For en vilkårlig binær relasjon $\Re$ på en eller annen mengde A, definerer vi forskjellige ***tillukninger*** av $\Re$ slik:

$\Re^\times = \Re^0 \cup \Re$, den ***refleksive tillukning*** av $\Re$.

$\Re^+ = \bigcup_{n>0} \Re^n$, den ***transitive tillukning*** av $\Re$.
M.a.o. $\langle a, c \rangle \in \Re^+ \Leftrightarrow \langle a, c \rangle \in \Re$ eller det finnes en b slik at $\langle a, b \rangle \in \Re$ og $\langle b, c \rangle \in \Re^+$.

$\Re^* = \Re^0 \cup \Re^+$, den ***refleksiv-transitive tillukning*** av $\Re$.

$\Re^\bullet = \Re \cup \Re^{-1}$, den ***symmetriske tillukning*** av $\Re$.

$\Re$ er ***velfundert*** dersom det ikke finnes noen uendelig sekvens $\langle a, b, c, \ldots \rangle$ for $a, b, c, \ldots \in A$ slik at $\langle a, b \rangle \in \Re, \langle b, c \rangle \in \Re, \ldots$.

Den ***universelle*** relasjon på en $A_1 \times \cdots \times A_n$ er mengden $\{\langle a_1, \ldots, a_n \rangle \mid a_i \in A_i, 1 \leq i \leq n\}$. Vi skal ofte skrive $a \; \Re \; b$ og $a \; \not\Re \; b$ i stedet for $\langle a, b \rangle \in \Re$ og $\langle a, b \rangle \notin \Re$ hhv.

[1] En ordnet mengde er mengde-teoretisk ikke en ny type objekt, men defineres utfra (vanlige) mengder. Paret $\langle a, b \rangle$ defineres i mengde-teori som mengden $\{\{a\}, \{a, b\}\}$. Et vilkårlig n-tuppel $\langle a_1, \ldots, a_n \rangle; n > 2$ defineres som paret $\langle \langle a_1, \ldots, a_{n-1} \rangle, a_n \rangle$.

Funksjoner

En ***funksjon*** f fra A til B for vilkårlige mengder A og B, er en binær relasjon $\Re_f \subseteq A \times B$ slik at det finnes nøyaktig ett tuppel (2-tuppel) i $\Re_f$ slik at $a \in A$ er første komponent, for alle $a \in A$. For hver $a \in A$ kan vi da identifisere en unik $b \in B$ — ***verdien*** av a under f — slik at $a\ \Re_f\ b$ og som vi skal betegne $f(a)$. Vi kaller A og B hhv. ***funksjonsdomenet*** (eller bare ***domenet***) og ***kodomenet*** til f. Vi skal tidvis forestille oss funksjoner som ***operatorer*** på sitt domene. For en vilkårlig funksjon f fra A til B for mengder A og B, sier vi da at f *tar* elementer fra A og *gir* elementer fra B. Vi skal da kalle $f(a)$ en ***applikasjon*** av f på a for en $a \in A$. Vi betegner mengden av alle funksjoner med domene A og kodomene B med $(A \rightarrow B)$.

La $f \in (A \rightarrow B)$ være en vilkårlig funksjon for mengder A og B. For en $A' \subseteq A$ sier vi at $f(A') = \{f(a) \mid a \in A'\}$ er ***bildet*** av A' under f. En funksjon $f \in (A \rightarrow B)$ er ***injektiv*** dersom for alle $a, a' \in A$: $a \neq a' \ \Rightarrow\ f(a) \neq f(a')$. En funksjon $f \in (A \rightarrow B)$ er ***surjektiv*** dersom det for alle $b \in B$ finnes en $a \in A$ slik at $f(a) = b$; dvs. at B er identisk med bildet av A under f. En ***bijektiv*** funksjon er en funksjon som er både injektiv og surjektiv. Alle binære relasjoner har som sagt en invers. Bijektive funksjoner (spesielle binære relasjoner) har dessuten alltid en invers som selv er en bijektiv funksjon. Sammensetningen av funksjoner er en funksjon. Sammensetninger av bijektive funksjoner er en bijektiv funksjon.

Et ***fikspunkt*** til en funksjon f er et element a i domenet til f slik at $f(a) = a$. En ***konstantfunksjon*** f_a er en funksjon slik at for en a så er $f_a(x) = a$ for alle x i domenet til f_a. En mengde A er ***lukket*** under en funksjon f, dersom $f(a) \in A$ for alle $a \in A$.

La $f \in (A_1 \times \cdots \times A_n \rightarrow B), n > 0$. Vi sier at f er n-***ær***. La $a_1 \in A_1, \ldots, a_n \in A_n$. Vi skal lette skrivemåten for $f(\langle a_1, \ldots, a_n \rangle)$ ved å skrive $f(a_1, \ldots, a_n)$.

Funksjoner er relasjoner, så relasjon-språkbruken 'en n-ær funksjon f *på* A' for en mengde A, henspeiler på en delmengde av A^{n+1}. Funksjonsdomenet til f er da A^n. Videre er begrepene 'restriksjon', 'sammensetning' samt notasjonen f^n for funksjoner definert ved relasjoner.

Vi kommer til å bruke ordene *funksjon* og *avbildning* om hverandre.

*

Vi har definert relasjoner som en spesiell type mengde og funksjoner som en spesiell type relasjon. Dette definisjonshierarkiet skal vi ha nytte av å snu på hodet som følger: Vi kan betrakte en n-ær relasjon $\Re$ på mengder $A_1, \ldots, A_n$ som en n-ær funksjon $f_\Re$ — ***funksjonsvarianten*** til $\Re$ — fra domenet $A_1 \times \cdots \times A_n$ til kodomenet $Bool = \{\bot, \top\}$ hvor $f_\Re(a_1, \ldots a_n) = \top$ hvis og bare hvis $\langle a_1, \ldots, a_n \rangle \in \Re$, for $a_1 \in A_1, \ldots, a_n \in A_n$. (Men merk at $f_\Re$ jo selvfølgelig igjen er en (spesiell) $n+1$-ær relasjon $\Re_{f_\Re}$ på $A_1, \ldots, A_n, Bool$.)

Vi skal også tidvis se på elementer som konstantfunksjoner; dvs. et vilkårlig element $a \in A$ sees på som en konstantfunksjon med kodomene $\{a\}$.

Store mengder

Vi betegner mengden av alle mengder med $\mathbb{S}$. Mengden av alle delmengder av en mengde A betegnes med $\mathcal{P}(A)$. Mengden av alle funksjoner betegnes med $\mathbb{F}$.[2]

[2] 'Mengden av alle mengder' er ikke et uproblematisk begrep (jf. 'Russels paradoks'), men for vår diskusjon lar vi dette ligge.

Avgjørbarhet

En egenskap $\mathcal{P}$ sies å være ***(effektivt) avgjørbar*** , dersom det finnes en deterministisk mekanisk prosess (deterministisk abstrakt maskin) som i endelig tid (i et endelig antall diskrete steg) gir en bestemt symbolkonfigurasjon $\mathcal{S}(\mathcal{P})$ hvis og bare hvis $\mathcal{P}$ holder.

En mengde sies å være avgjørbar, dersom medlemskap i mengden er avgjørbar.

En oppgave $\mathcal{O}$ er ***algoritmisk***, dersom det finnes en deterministisk mekanisk prosess som i endelig tid i en passende forstand utfører $\mathcal{O}$.

2.2 Algebra

Fra det grunnleggende begrep 'mengde' har vi definert «høyere-nivå» begreper som n-tupler, relasjoner og funksjoner som spesielle typer mengder. Vi definerer nå et begrep på et enda høyere nivå.

En *algebra* er essensielt et tuppel $\langle A, F\rangle$, der A er en mengde og F er en mengde funksjoner på A. En algebra kan således avgrense et interesse-område. Ønsker man f.eks. å studere trigonometriske funksjoner på de reelle tall, vil man kanskje betrakte algebraen $\langle \mathbb{R}, \{+, *, \sin, \cos\}\rangle$. Vi skal med begrepet algebra presisere begrepet abstrakt datatype og også begrepet formell datatype.

Vi søker å *implementere* abstrakte datatyper ved formelle datatyper. Ved å implementere en abstrakt datatype menes her 1) å representere verdimengden i datatypen symbolsk, og 2) å modellere funksjonene i datatypen formelt. Implementasjon kan altså sees som bestående av en *representasjonell* del og en *operasjonell* del. Vi ønsker også å resonnere formelt og i siste instans også mekanisk ved hjelp av formelle datatyper. Til dette trengs også symbolske representasjoner og/eller et formelt språk.

Vi gir her først det formelle språket. Deretter defineres avbildninger som «tolker» språket til semantiske størrelser.

2.2.1 Formelt språk

Det formelle språket skal være typet. Typeuniverset er imidlertid svært enkelt og en delmengde av f.eks. typeuniverset i typet λ-kalkyle.

La $\mathcal{GT}ype$ være en gitt symbolmengde av ***grunntyper***. Vi definerer en mengde $\mathcal{T}ype$ av ***typer*** fra $\mathcal{GT}ype$ og symbolmengden $\{\times, \rightarrow\}$ som den minste mengden av symbolsekvenser (skrives tettløpende) som tilfredstiller:

- $\mathcal{GT}ype \subseteq \mathcal{T}ype$
- $T_1, \ldots, T_n, T \in \mathcal{GT}ype; n \geq 1 \quad \Rightarrow \quad T_1 \times \cdots \times T_n \rightarrow T \in \mathcal{T}ype$

(Merk at denne definisjonen av typer *ikke* er induktiv. De typer vi her befatter oss med, kan derfor ikke være av vilkårlig kompleksitet.)

En ***funksjonsprofil*** er en symbolsekvens på formen

$$f : T$$

der f er et ***funksjonsymbol*** og T er en type. Vi sier at f her er *av type* T. Dersom $T = T_1 \times \cdots \times T_n \rightarrow T_C; n \geq 1$, sier vi at f har ***aritet*** n. Hvis $T \in \mathcal{GT}ype$, kalles f en ***konstant***.

En ***(funksjons)signatur*** Σ er en endelig mengde funksjonsprofiler[3]. Et funksjonsymbol kan høyst forekomme én gang i en signatur. Dersom det kun

[3] Vi lar informasjon om ariteter og om hvilke grunntyper som er involvert fremgå implisitt gjennom profilene i signaturen.

forekommer én grunntype i en signatur, skal vi tidvis forkorte skrivemåten av signaturen ved å bare angi funksjonsymbolene (og deres ariteter) uten fullstendig profil. Denne skrivemåten inspirerer til å kalle en slik en-typet situasjon for ***utypet***. Vi skal også bruke denne forkortede skrivemåten ellers, hvis typeinformasjon er overflødig for diskusjonen.

En ***variabelprofil*** er en symbolsekvens på formen

$$x : T$$

der x er et ***variabel(symbol)*** og T er en grunntype. Vi sier at x er av type T. En ***variabelsignatur*** $\mathcal{V}$ er en endelig mengde variabelprofiler. Et variabelsymbol kan kun forekomme én gang i en variabelsignatur. Dersom det kun forekommer én grunntype i en variabelsignatur, eller hvis det ellers er hensiktsmessig, skal vi tidvis angi bare variabelsymbolene, uten fullstendig profil.

La Σ og $\mathcal{V}$ være en vilkårlig funksjonssignatur og variabelsignatur hhv. Vi definerer mengden $\mathcal{T}_\Sigma(\mathcal{V})$ av ***termer*** over Σ og $\mathcal{V}$ som den minste mengden av symbolsekvenser induktivt bygget over Σ og $\mathcal{V}$ (samt noen skillesymboler) som tilfredstiller følgende:

- $\mathcal{V} \subseteq \mathcal{T}_\Sigma(\mathcal{V})$
- Hvis $k : T \in \Sigma$ og $T \in \mathcal{GT}ype$, så er $k \in \mathcal{T}_\Sigma(\mathcal{V})$.
- Hvis $f : T_1 \times \cdots \times T_n \rightarrow T \in \Sigma$ og $t_i \in \mathcal{T}_\Sigma(\mathcal{V})$ og t_i er av type T_i for $0 < i \leq n$, så er $f(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma(\mathcal{V})$. Vi sier at $f(t_1, \ldots, t_n)$ er av type T.

Vi betegner mengden av ***grunntermer*** $\mathcal{T}_\Sigma(\emptyset)$ med $\mathcal{G}_\Sigma$.

2.2.2 Tolking av formelt språk

En ***typetolk*** er en avbildning $\Phi_\mathcal{T} \in (\mathcal{T}ype \rightarrow \mathbb{S} \setminus \emptyset)$, slik at for vilkårlige $T_0, T_1, \ldots, T_n, T \in \mathcal{GT}ype; n \geq 1$:

- $\Phi_\mathcal{T}(T_0) \in \mathbb{S} \setminus \emptyset$.
- $\Phi_\mathcal{T}(T_1 \times \cdots \times T_n \rightarrow T) = (\Phi_\mathcal{T}(T_1) \times \cdots \times \Phi_\mathcal{T}(T_n) \rightarrow \Phi_\mathcal{T}(T))$.

For en vilkårlig $T \in \mathcal{T}ype$, sier vi at $\Phi_\mathcal{T}(T)$ er en ***tolkning*** av T.

En ***funksjonsprofiltolk*** er en avbildning $\Phi_\mathcal{F}$ slik at for en vilkårlig profil $f : T$:

$$\Phi_\mathcal{F}(f : T) \in \Phi_\mathcal{T}(T)$$

for en (forutbestemt) tolk av typer $\Phi_\mathcal{T}$. Vi sier at $\Phi_\mathcal{F}(f : T)$ er en ***tolkning*** av f, for et vilkårlig funksjonssymbol f; underforstått en tilhørende funksjonsprofil. Vi sier at $\Phi_\mathcal{T}$ her ***inngår*** i (definisjonen av) $\Phi_\mathcal{F}$.

*

En Σ-***algebra*** A for en signatur Σ er et $r+1$-tuppel $\langle D_1, \ldots, D_r, F \rangle \in \mathbb{S}^r \times \mathcal{P}(\mathbb{F})$, slik at $D_i = \Phi^A_\mathcal{T}(T_i), 1 \leq i \leq r$ for en typetolk $\Phi^A_\mathcal{T}$ og der $\{T_1, \ldots, T_r\}$ er mengden av alle grunntyper som forekommer i Σ. For $1 \leq i \leq r$ kalles D_i ***tolkningen*** av T_i i A. Videre er F en mengde funksjoner (også konstanter) slik: For hver funksjonsprofil $f : T$ i Σ skal det være en tolkning $f_A = \Phi^A_\mathcal{F}(f : T)$ i F for en funksjonsprofiltolk $\Phi^A_\mathcal{F}$ hvori $\Phi^A_\mathcal{T}$ inngår. Vi sier at f_A er ***tolkningen*** av f i A. Algebraen A er en ***tolkning*** av Σ og er entydig bestemt av $\Phi^A_\mathcal{T}$ og $\Phi^A_\mathcal{F}$. Som regel skal vi ikke eksplisitt nevne tolkene som inngår i definisjonen av en Σ-algebra.

Dersom vi snakker om en Σ-algebra for en eller annen ikke nærmere spesifisert signatur Σ, skal vi bare bruke benevnelsen ***algebra***. Vi kaller $D_i, 1 \leq i \leq r$, ***bæremengder*** til A, og F kalles ***funksjonsmengden*** til A. Vi skal ofte betrakte bæremengdene til en algebra under ett og da som unionen av alle bæremengdene. Denne unionen refereres da til som algebraens ***domene***. Vi betegner ofte en algebra og dens domene med det samme; altså $A = \langle A, F \rangle$. Vi betegner da bæremengden $D_i \subseteq A$ for en $1 \leq i \leq r$ tidvis med A_{T_i}.

Eksempel 1 Betrakt den (utypete) signaturen $\mathsf{Int} = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}\}$. Ariteter er hhv. 0, 1 og 1. En mulig tolkning av Int er algebraen $\mathcal{I}nt = \langle \mathbb{Z}, \{0, succ, pred\} \rangle$ der $succ$ og $pred$ er hhv. etterfølger- og forgjengerfunksjonen for hele tall. Disse er tolkningene i $\mathcal{I}nt$ av hhv. succ og pred. Konstantfunksjonen 0 er tolkningen i $\mathcal{I}nt$ av $\mathsf{0}$.
○

Et ***redukt*** av en Σ-algebra $A = \langle D_A, F_A \rangle$ er en Σ'-algebra $B = \langle D_B, F_B \rangle$ for en $\Sigma' \subseteq \Sigma$ slik at $D_B \subseteq D_A, F_B \subseteq F'_A$, der F'_A er restriksjonene av funksjonene i F_A til D_B. Spesialtilfellet for $F_B = F'_A$ kalles en ***subalgebra*** av A. Et redukt av A som ikke er en subalgebra av A er et ***ekte redukt*** av A.

Eksempel 2 For signaturen $\mathsf{Int}^+ = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}, \mathsf{+}\}$ med ariteter $0, 1, 1, 2$ hhv., betrakt Int^+-algebraen $\mathcal{I}nt^+ = \langle \mathbb{Z}, \{0, succ, pred, +_{\mathbb{Z}}\} \rangle$ for $+_{\mathbb{Z}}$ addisjon på hele tall. For $\mathsf{Nat}^+ = \{\mathsf{0}, \mathsf{succ}, \mathsf{+}\} \subset \mathsf{Int}^+$, er Nat^+-algebraen $\mathcal{N}at^+ = \langle \mathbb{N}, \{0, succ, +_{\mathbb{N}}\} \rangle$ for $+_{\mathbb{N}}$ addisjon på hele tall, et ekte redukt av $\mathcal{I}nt^+$.
○

Med en 'abstrakt datatype' skal vi nå mene en 'algebra' — tolkningen av en funksjonssignatur; altså en tolkning av et syntaktisk objekt. Vi har nå etablert et grunnlag for vårt semantiske sprang mellom syntaks og semantikk.

Grunntermer er delen av det formelle språk som skal representere elementer og funksjonsapplikasjoner i en algebra. Gitt en Σ-algebra A, følger en *grunntermtolk* naturlig utfra typetolken og funksjonsprofiltolken i definisjonen av A. Vi skal imidlertid generelt ha bruk for mer «abstrakte» termtolker. Til det trengs et par begreper til.

2.2.3 Homomorfier og term-algebraer

La A, B være to vilkårlige Σ-algebraer for en signatur Σ. En ***homomorfi*** ϕ fra A til B er en avbildning fra A sitt domene til B sitt domene slik at for alle $f : T_1 \times \cdots \times T_n \rightarrow T \in \Sigma; n \geq 1$

$$\phi(f_A(a_1, \ldots, a_n)) = f_B(\phi(a_1), \ldots, \phi(a_n))$$

for alle $\langle a_1, \ldots, a_n \rangle \in A_1 \times \cdots \times A_n$, der A_i er tolkningen av $T_i, 1 \leq i \leq n$, i A, og f_A og f_B er tolkningene av f i A og B hhv. Vi krever at $\langle \phi(a_1), \ldots, \phi(a_n) \rangle \in B_1 \times \cdots \times B_n$ der B_i er tolkningen av $T_i, 1 \leq i \leq n$, i B.

Vi skriver av og til ϕ_A^B for en homomorfi fra en algebra A til en algebra B. Mengden av homomorfier fra A til B betegnes med $\mathcal{H}om_A^B$. Dersom det finnes en surjektiv homomorfi fra A til B, kalles B et ***homomorft bilde*** av A. Sammensetningen av homomorfier er en homomorfi.

La $\mathcal{K}$ være en *klasse* (mengde) algebraer. En algebra $A \in \mathcal{K}$ kalles ***initiell*** i $\mathcal{K}$ hvis det for alle algebraer $B \in \mathcal{K}$ fins nøyaktig én homomorfi fra A til B. På den annen side kalles A ***final*** i $\mathcal{K}$ hvis det for alle algebraer $B \in \mathcal{K}$ fins nøyaktig én homomorfi fra B til A.

For en gitt Σ og variabelsignatur $\mathcal{V}$ definerer vi en ***term-algebra*** fra Σ og $\mathcal{V}$ som Σ-algebraen med bæremengde $\mathcal{T}_\Sigma(\mathcal{V})$ og med den minste funksjonsmengde slik at det for hver funksjonsprofil $f : T_1 \times \cdots \times T_n \rightarrow T \in \Sigma; n \geq 1$, finnes

en tolkning f_T av f slik at $f_T(t_1,\ldots,t_n) = f(t_1,\ldots,t_n)$ for alle t_i av type $T_i, 0 \leq i \leq n$. Funksjonene i en term-algebra kan, om man vil, betraktes som bestanddelene til den algoritmiske oppbyggingen fra funksjonsymboler og variable, av et rekursivt tellbart univers av termer. Term-algebraer betegnes, som andre algebraer, ofte med deres bæremengde.

En homomorfi ϕ fra en term-algebra $\mathcal{T}_\Sigma(\mathcal{V})$ til en eller annen Σ-algebra A er entydig bestemt av sin restriksjon ϕ' til $\mathcal{V} \times A$. Det følger da at for en gitt Σ, så er $\mathcal{G}_\Sigma$ initiell i klassen av Σ-algebraer. Betrakt derfor nå den unike homomorfi $\phi^A_{\mathcal{G}_\Sigma}$ for en vilkårlig Σ-algebra A. For en vilkårlig term $f(t_1,\ldots,t_n) \in \mathcal{G}_\Sigma, n \geq 0$, har vi

$$\phi^A_{\mathcal{G}_\Sigma}(f(t_1,\ldots,t_n)) = \phi^A_{\mathcal{G}_\Sigma}(f_T(t_1,\ldots,t_n)) = f_A(\phi^A_{\mathcal{G}_\Sigma}(t_1),\ldots,\phi^A_{\mathcal{G}_\Sigma}(t_n))$$

Denne unike homomorfi utgjør en ***grunntermtolk*** for $\mathcal{G}_\Sigma$.[4]

Eksempel 3 For signaturen Int og algebraen $\mathcal{I}nt$ som i eksempel 1, betrakt grunnterm-algebraen $\mathcal{G}_{\mathsf{Int}}$. Den unike homomofien $\phi^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ tolker hver grunnterm $g \in \mathcal{G}_{\mathsf{Int}}$ til et element i $\mathcal{I}nt$. F.eks. er $\phi^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}(\mathsf{succ}(\mathsf{pred}(\mathsf{0}))) = succ(pred(0)) = 0$.
○

Grunnterm-algebraer utgjør grunnlaget for definisjonen av formelle datatyper. Før vi definerer formelle datatyper, skal vi fullføre presentasjonen av symbolsk representasjon.

2.2.4 Full uttrykkbarhet

Ifølge definisjonen, kan en Σ-algebra A ha andre «ekstra» funksjoner i sin funksjonsmengde enn dem som er tolkninger av en profil i Σ. Domenet kan dessuten ha elementer som ikke er *nåbare* fra $\mathcal{G}_\Sigma$, i den forstand at det ikke finnes noen $g \in \mathcal{G}_\Sigma$ slik at $\phi^A_{\mathcal{G}_\Sigma}(g) = a$ for et ikke-nåbart element $a \in A$. Vi kaller her a ***skrot*** i forhold til Σ.

Eksempel 4 Betrakt igjen signaturen $\mathsf{Nat}^+ = \{\mathsf{0}, \mathsf{succ}, \mathsf{+}\}$. La $a \notin \mathbb{N}$. Algebraen $\mathcal{N}at^{+a} = \langle \mathbb{N} \cup \{a\}, \{0, succ^a, +^a\}\rangle$, der 0, $succ^a$ og $+^a$ er tolkningene av hhv. $\mathsf{0}$, succ og $\mathsf{+}$, er en Nat^+-algebra der a er skrot i forhold til Nat^+. Funksjonene $succ^a$ og $+^a$ er utvidelser av $succ$ og $+$ på $\mathbb{N}$ til $\mathbb{N} \cup \{a\}$; f.eks. slik at $succ^a(a) = a$, $a + x = a$ og $x + a = x$.
○

Vi vil implementere abstrakte datatyper, og er da interessert i at det formelle språket kan uttrykke mest mulig av den abstrakte datatypen. Hvis grunntermtolken $\phi^A_{\mathcal{G}_\Sigma}$ fra en $\mathcal{G}_\Sigma$ til en Σ-algebra A er surjektiv, er alle elementer nåbare. Siden signaturer er endelige, er ethvert termunivers rekursivt tellbart. Dette er viktig, fordi formell resonnering i formelle datatyper ved de metoder vi skal snakke om, avhenger av rekursivt tellbare termunivers. Dersom grunntermtolken er surjektiv, må domenet til A være rekursivt tellbart. Ønsker vi å kunne uttrykke alle elementer i bæremengder til algebraer, avgrenses da de algebraer som vi

[4] De svært enkle sammensatte typer i vårt utsnitt av universell algebra, representerer en begrenset uttrykkskraft i forhold til det fulle språket til typet λ-kalkyle. Det er ikke hvilke algebraer som kan representeres som begrenses, men heller graden av detalj med hvilken elementer og funksjonsapplikasjoner i enkelte algebraer kan representeres. Betrakter vi eksempelvis signaturen $\Sigma = \{k : T, f : T \rightarrow T\}$, så kan typen T tolkes til for eksempel en mengde $A \times B$. En gitt term i $\mathcal{G}_\Sigma$ vil da representere et element $\langle a, b\rangle \in A \times B$, men uttrykker ikke at $\langle a, b\rangle$ er et tuppel med to komponenter a og b. Denne begrensningen i detaljnivå gjelder også når T tolkes til en mengde på formen $(A \rightarrow B)$. Imidlertid er vår kontekst implementasjon av, og resonnering om programmer. I den forbindelse skal vi ikke behøve mer enn dette begrensete detaljnivå.

kan implementere til dem som er homomorfe bilder av grunnterm-algebraer, og altså har rekursivt tellbare domener.

Merk at selv om $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv, kan A fortsatt ha «ekstra» funksjoner. Men har man først surjektiv grunntermtolk, kan slike funksjoner symboliseres ved å bruke en fornuftigere signatur.

La A være en Σ-algebra for en Σ. Dersom grunntermtolken $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv og det ikke finnes funksjoner i A som ikke er en tolkning av et symbol i Σ, skal vi si at vi har ***full uttrykkbarhet (for A)***.

2.2.5 Formulering av matematiske påstander

Termer med variable kan benyttes for å uttrykke matematiske påstander. For en signatur Σ betegner vi mengden $\{\langle s,t\rangle \mid s,t \in \mathcal{T}_\Sigma(\mathcal{V})$ *og* s,t *er av samme type*$\}$ med $\mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$. Vi kaller elementer i $\mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ ***ligninger***; skrevet $s = t$.[5]

For vilkårlige $s = t \in \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ og en Σ-algebra A, skriver vi $A \models s = t$ for å antyde at $\phi(s) = \phi(t)$ for alle $\phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})}$ og sier da at A ***tilfredstiller*** ligningen $s = t$ eller at $s = t$ er ***sann*** i A. Hver $\phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})}$ er som sagt entydig bestemt av sin restriksjon til $\mathcal{V} \times A$. Følgelig sier påstanden $A \models s = t$ at s og t representerer samme element i A for hver tolkning til elementer i A av variablene i s og t.[6]

Eksempel 5 For Int-algebraen $\mathcal{I}nt$ fra eksempel 1, sier påstanden $\mathcal{I}nt \models$ succ(pred(x)) = x at $succ(pred(x)) = x$ for alle $x \in \mathbb{Z}$.
○

Merk at for alle $\phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})}$ og $g \in \mathcal{G}_\Sigma \subseteq \mathcal{T}_\Sigma(\mathcal{V})$ har vi $\phi(g) = \phi^A_{\mathcal{G}_\Sigma}(g)$. For $g, g' \in \mathcal{G}_\Sigma$ har vi da

$$A \models g = g' \;\Leftrightarrow\; \phi^A_{\mathcal{G}_\Sigma}(g) = \phi^A_{\mathcal{G}_\Sigma}(g') \tag{2.1}$$

En ***substitusjon*** er en funksjon som gitt en term *instansierer* variablene i termen med andre termer. Mer presist er en substitusjon en homomorfi $\sigma \in \mathcal{H}om^{\mathcal{T}_\Sigma(\mathcal{V})}_{\mathcal{T}_\Sigma(\mathcal{V})}$. Her er σ entydig bestemt av sin restriksjon til $\mathcal{V} \times \mathcal{T}_\Sigma(\mathcal{V})$, m.a.o. entydig bestemt av verdiene den tilordner variablene i sitt domene. Vi skriver $t\sigma$ for applikasjonen av en substitusjon σ på en term t. En substitusjon i $\mathcal{H}om^{\mathcal{G}_\Sigma}_{\mathcal{T}_\Sigma(\mathcal{V})}$ kalles en ***grunnsubstitusjon (for Σ)***.

Eksempel 6 (Her og andre steder der det er visuelt hensiktsmessig, benyttes mixfix-notasjon for enkelte termer.) Betrakt signaturen $\mathsf{Nat}^+ = \{\mathsf{0}, \mathsf{succ}, +\}$. La $\sigma \in \mathcal{H}om^{\mathcal{T}_{\mathsf{Nat}^+}(\mathcal{V})}_{\mathcal{T}_{\mathsf{Nat}^+}(\mathcal{V})}$ for $\mathcal{V} = \{\mathsf{x}, \mathsf{y}\}$, entydig bestemt ved $\sigma(\mathsf{x}) = \mathsf{succ(x)}$ og $\sigma(\mathsf{y}) = \mathsf{succ(y)}$. Da er $(\mathsf{x+y})\sigma = \mathsf{x}\sigma\mathsf{+y}\sigma = \mathsf{succ(x)+succ(y)}$.

La så τ være en grunnsubstitusjon for Nat^+ entydig bestemt ved $\tau(\mathsf{x}) = \mathsf{0}$ og $\tau(\mathsf{y}) = \mathsf{succ(0)}$. Da er $(\mathsf{x+y})\tau = \mathsf{x}\tau\mathsf{+y}\tau = \mathsf{0+succ(0)}$.
○

En substitusjon er altså entydig bestemt av verdiene den tilordner variablene i sitt domene. Den til enhver tid aktuelle variabelsignatur er ofte implisitt eller gitt utfra kontekst. Derfor skal vi istedenfor $\sigma \in \mathcal{H}om^{\mathcal{T}_\Sigma(\mathcal{V})}_{\mathcal{T}_\Sigma(\mathcal{V})}$ heller skrive

[5] Vi definerer ligninger som *tupler* dvs. *ordnete* mengder. Dette fordi ligninger for oss er syntaktiske konstruksjoner som skal behandles av abstrakte maskiner. Abstrakte maskiner behandler ikke syntaks som uordnet. De må isåfall programmeres til det. (F.eks. må kalkyler for logikk eksplisitt ha inferensregler som uttrykker eventuelle symmetri-egenskaper.) Merk forresten at vi bruker et fett likhetstegn '=' i ligninger i motsetning til tegnet '=' for elementidentitet.

[6] Dette tilsvarer tolkningen av det analoge predikatlogiske utsagnet $\forall \bar{x} : s = t$, der $\bar{x}$ er en variabelliste som inkluderer variable forekommende i s og t. Merk at *vi* ikke har allkvantorsymbol og at likhetssymbolet i våre ligninger ikke tolkes eksplisitt.

$\sigma \in \mathcal{S}bst^{\mathcal{T}_\Sigma(\mathcal{V})}$, og si at σ er en $\mathcal{T}_\Sigma(\mathcal{V})$-***substitusjon***. Vi skal dessuten skrive $\mathcal{S}bst^{\mathcal{T}_{\Sigma'}(\mathcal{V}')}$ for $\Sigma' \subseteq \Sigma$, med den forståelse at en $\sigma \in \mathcal{S}bst^{\mathcal{T}_{\Sigma'}(\mathcal{V}')}$ sin restriksjon til $\mathcal{V} \times \mathcal{T}_\Sigma(\mathcal{V})$ er inneholdt i $\mathcal{V} \times \mathcal{T}_{\Sigma'}(\mathcal{V}')$. Det skal følgelig forståes at $\mathcal{S}bst^{\mathcal{T}_{\Sigma'}(\mathcal{V}')} \subseteq \mathcal{H}om^{\mathcal{T}_\Sigma(\mathcal{V})}_{\mathcal{T}_\Sigma(\mathcal{V})}$.

En påstand som $A \models s = t$ er også en påstand om *funksjonsapplikasjoner* i A i kraft av tolken av funksjonssymboler som inngår i tolken av Σ til A. Dersom A har skrot i forhold til Σ, og s,t har variable, er $A \models s = t$ en påstand om funksjonsapplikasjoner også på skrotet. Men en påstand om identitet med skrot kan ikke uttrykkes ved ligninger i grunntermer:

Eksempel 7 For Nat^+-algebraen $\mathcal{N}at^{+a}$ fra eksempel 4, sier påstanden $\mathcal{N}at^{+a} \models \mathsf{x+y = y+x}$ at $x +^a y = y +^a x$ ikke bare for alle $x,y \in \mathbb{N}$, men endog for alle $x,y \in \mathbb{N} \cup \{a\}$. Påstanden sier f.eks. at $a +^a 0 = 0 +^a a$. Denne delpåstanden kan ikke uttrykkes ved en ligning i grunntermer fra $\mathcal{G}_{\mathsf{Nat}^+}$.
○

Hvis derimot $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv, forstår vi intuitivt, siden alle elementer i bæremengden til A er nåbare fra $\mathcal{G}_\Sigma$ — altså uttrykkbare ved grunntermer — at

$$A \models s = t \;\Leftrightarrow\; \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid A \models s\sigma = t\sigma \tag{2.2}$$

for alle $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$. Dvs. at alle identiteter i A kan uttrykkes ved ligninger i grunntermer. (For et bevis av (2.2), se avsnitt A.1.1 i tilleggskapittel A.)

Nå kan det virke som om definisjonen av algebra er uhensiktsmessig, siden vi stadig må forfekte surjektivitet av grunntermtolken. I forbindelse med ligningslogikk og logisk gyldighet skal vi imidlertid siden se det meningsfylte i at en tolkning av en signatur ikke er et homomorft bilde av den korresponderende grunnterm-algebraen.

Vi avslutter avsnittet med noen begreper som for oss skal bli sentrale:

Definisjon 2.1 *La A og B være to Σ-algebraer for en vilkårlig signatur Σ. B er* ***induktivt implisert*** *av A dersom for alle $g,g' \in \mathcal{G}_\Sigma$*

$$A \models g = g' \;\Rightarrow\; B \models g = g'$$

B er ***elementært implisert*** *av A dersom for alle $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$*

$$A \models s = t \;\Rightarrow\; B \models s = t$$

Dersom A og B er gjensidig induktivt/elementært implisert av hverandre sier vi at A og B er ***induktivt/elementært ekvivalente.***

2.2.6 Omskrivningssystemer og ligningslogikk

Vi har definert et formelt språk. Representasjons-delen av implementasjon er således beskrevet. Nå skal den operasjonelle delen av formelle datatyper beskrives. Vi trenger nå litt mer detaljerte begreper om termer.

Vi definerer en ***posisjon*** i t for en vilkårlig term t som en sekvens av positive heltall slik:

- ε er en posisjon i t.
- For $t = f(t_1,\ldots,t_n)$ er $p = i.q$ en posisjon i t, hvis $1 \leq i \leq n$ og q er en posisjon i t_i.

(Merk her forkortet skrivemåte for sekvenser av tall.) For en vilkårlig term t og en posisjon p i t definerer vi en funksjon $|p$ slik:

- $t|\varepsilon = t$
- $t|i.q = t_i|q$ for $t = f(t_1,\ldots,t_n); 1 \leq i \leq n$.

For en term t og en posisjon p i t kalles $t|p$ en ***subterm*** av t. Dersom $p \neq \varepsilon$, kalles $t|p$ en ***ekte subterm*** av t.

Det er tidvis hensiktsmessig å visualisere termer som ***trær***[7], med funksjonssymboler med aritet større enn 1 som intern-noder, og med variable og konstanter som blader. En (ekte) subterm svarer da til et (ekte) ***subtre***. For en term $t = f(t_1,\ldots,t_n)$ sies den viste forekomsten av f å være ***rot*** i termen (treet) t. En konstant k er rot i k. Vi sier at en subterm s er på ***dybde*** d i en term t, dersom $t|p = s$ for en posisjon p og $len_p = d$.

For vilkårlige termer t, s og posisjon p i t definerer vi en *erstatter*-funksjon $[]_p$ slik:

- $t[s]_\varepsilon = s$
- $f(t_1,\ldots,t_i,\ldots,t_n)[s]_{i.q} = f(t_1,\ldots,t_i[s]_q,\ldots,t_n)$ for $t = f(t_1,\ldots,t_n); 1 \leq i \leq n$.

Dvs. $t[s]_p$ er termen t, men med subtermen i posisjon p erstattet med s. En ***kontekst*** er en term bygget over funksjonssymboler og den spesielle konstanten $\Box$; kalt et ***hull***. La c være en kontekst med n forekomster av hull. Vi betegner c med hullene substituert (fylt) med termene $s_1,\ldots,s_n$, ved $c[s_1,\ldots,s_n]$. (Vi skal her aldri fylle kun noen hull i en kontekst.) Dersom $c[s] = s$ sier vi at konteksten c er ***tom***. Vi skriver $c \in \mathcal{T}_\Sigma(\mathcal{V})$ istedenfor $c \in \mathcal{T}_{\Sigma\cup\{\Box\}}(\mathcal{V})$.

Et ***termomskrivningssystem*** eller her bare ***omskrivningssystem***, er et formelt system og en abstrakt maskin. Et omskrivningssystem $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ har et *univers* $\mathcal{T}_\Sigma(\mathcal{V})$ for en signatur Σ og en endelig mengde $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ av tupler, nå kalt ***regler***, skrevet $v \rightarrow h$. (Merk at tupler er ordnede.) For vilkårlige $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$ sier vi at $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ ***omskriver*** s til t i ett steg dersom det finnes en $v \rightarrow h \in R$ slik at $s|p = v\sigma$ og $t = s[h\sigma]_p$, for en posisjon p i s og substitusjon σ på $\mathcal{T}_\Sigma(\mathcal{V})$. Deloppgaven å finne en σ slik at $s|p = v\sigma$ kalles ***matching*** og er algoritmisk.

Eksempel 8 For signaturen $\mathsf{Nat}^+ = \{\mathsf{0}, \mathsf{succ}, \mathsf{+}\}$, betrakt følgende regelmengde fra $\mathcal{E}(\mathcal{T}_{\mathsf{Nat}^+}(\mathcal{V}))$:

$$R = \left\{ \begin{array}{l} \mathsf{+(x,0)} \rightarrow \mathsf{x}, \\ \mathsf{+(x,succ(y))} \rightarrow \mathsf{succ(+(x,y))} \end{array} \right\}$$

Da kan termen $\mathsf{+(succ(succ(0)),succ(0))}$ omskrives til $\mathsf{succ(+(succ(succ(0)),0))}$, ved regelen $\mathsf{+(x,succ(y))} \rightarrow \mathsf{succ(+(x,y))}$ i posisjon ε og substitusjonen σ slik at $\sigma(\mathsf{x}) = \mathsf{succ(succ(0))}$ og $\sigma(\mathsf{y}) = \mathsf{0}$. Videre kan $\mathsf{succ(+(succ(succ(0)),0))}$ omskrives til $\mathsf{succ(succ(succ(0)))}$.
○

Når universet fremgår av diskusjonen, betegner vi ofte et omskrivningssystem kun ved dets regelmengde.

En binær relasjon $\Re$ på $\mathcal{T}_\Sigma(\mathcal{V})$ for en signatur Σ er ***monoton mhp. substitusjon*** dersom $s \,\Re\, t \;\Rightarrow\; s\sigma \,\Re\, t\sigma$ for alle $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$ og alle substitusjoner σ på $\mathcal{T}_\Sigma(\mathcal{V})$. $\Re$ er ***monoton mhp. kontekstapplikasjon*** dersom $s \,\Re\, t \;\Rightarrow\; c[s] \,\Re\, c[t]$ for alle $c, s, t \in \mathcal{T}_\Sigma(\mathcal{V})$. Dersom $\Re$ er monoton på begge disse måter, kalles $\Re$ en ***omskrivningsrelasjon***. En $\Re$***-utledning i*** $\mathcal{T}$, for $\Re$ en omskrivningsrelasjon

[7] I stil med vår oppsummering av grunnleggende begreper i avsnitt 2.1, kan trær defineres som spesielle typer sekvenser. Her er det imidlertid mer hensiktsmessig å hvile på en intuitiv visuell forestilling av trær.

og $\mathcal{T}$ et vilkårlig redukt av $\mathcal{T}_\Sigma(\mathcal{V})$, er en sekvens $\langle t_0, t_1, t_2, \ldots\rangle$ av termer i $\mathcal{T}$ slik at $t_i \Re t_{i+1}$ for alle $i = 0, 1, 2, \ldots$. En omskrivningsrelasjon som er velfundert kalles en ***reduksjonsordning***.

Betrakt nå en vilkårlig $M \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$. Vi betrakter M som en binær relasjon på $\mathcal{T}_\Sigma(\mathcal{V})$. Vi betegner den ***monotone tillukning*** $\{\langle c[s\sigma], c[t\sigma]\rangle \mid \langle s,t\rangle \in M\}$ for alle (etthulls-)kontekster $c \in \mathcal{T}_\Sigma(\mathcal{V})$ og substitusjoner σ på $\mathcal{T}_\Sigma(\mathcal{V})$, av M med $\underset{M}{\rightarrow}$. Den inverse til $\underset{M}{\rightarrow}$ skriver vi som $\underset{M}{\leftarrow}$ og den symmetrisk-monotone tillukning av M som $\underset{M}{\leftrightarrow}$. En $\underset{M}{\leftrightarrow}$-utledning kalles bare en ***M-utledning***. En $\underset{M}{\rightarrow}$-utledning kalles en ***ensrettet M-utledning***. Dersom det for en ligning $s = t$ finnes en M-utledning $\langle s, \ldots, t\rangle$ sier vi at $s = t$ er ***utledbar*** i M. Vi sier også at s er ***omskrivbar*** til t i M.

Det er lett å se for alle $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$ at $s \underset{R}{\rightarrow} t$ hvis og bare hvis omskrivningssystemet $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ omskriver s til t i ett steg. Videre har vi $s \underset{R}{\leftrightarrow} t$ hvis og bare hvis $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ omskriver s til t *eller* t til s i ett steg. At $s \overset{*}{\underset{R}{\leftrightarrow}} t$ betyr da at s og t kan omskrives til hverandre (i ingen eller fler steg) i omskrivningssystemet $R \cup R^{-1}$, som for hver regel $v \rightarrow h$ i R, også har en invers $h \rightarrow v$. Et slikt symmetrisk omskrivningssystem utgjør en for våre formål hensiktsmessig beskrivelse av ***ligningslogikk***. Ligningslogikk er det grunnleggende formelle system for oss.

2.2.7 Termunivers

Vi skyter her inn en generell kommentar. I definisjonen av monoton tillukning av en $M \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$, betraktet vi M som en relasjon på $\mathcal{T}_\Sigma(\mathcal{V})$. Men M kan også betraktes som en relasjon på en vilkårlig $\mathcal{T}_{\Sigma'}(\mathcal{V}') \supseteq \mathcal{T}_\Sigma(\mathcal{V})$. Utfra dette kan vi få forskjellige tillukninger $\overset{*}{\underset{M}{\leftrightarrow}}$. I tillegg fås forskjellige tillukninger $\{\langle c[s\sigma], c[t\sigma]\rangle \mid \langle s,t\rangle \in M\}$ på vilkårlige $\mathcal{T}_{\Sigma'}(\mathcal{V}') \subseteq \mathcal{T}_\Sigma(\mathcal{V})$ ved å betrakte $\sigma \in \mathcal{S}bst^{\mathcal{T}_{\Sigma'}(\mathcal{V}')}$ og kontekster $c \in \mathcal{T}_{\Sigma'}(\mathcal{V}')$.

Vi antar imidlertid alltid gitt, implisitt eller eksplisitt, en signatur som bestemmer det aktuelle ***termunivers*** for diskusjonen. Eksempler på dette har vært i definisjonene av substitusjoner, omskrivningssystemer, monoton tillukning samt i definisjonen av induktiv/elementær implikasjon/ekvivalens. Vi skal etterhvert se flere eksempler hvor det er nødvendig å anta et slikt avgrensende termunivers.

2.2.8 Algebraisk spesifikasjon

Vi skal nå bruke ligninger og ligningslogikk til å gi ***konstruktive*** spesifikasjoner av funksjoner i abstrakte datatyper. Med 'konstruktiv' spesifikasjon/definisjon av en funksjon mener vi her en spesifikasjon som gir nok informasjon om funksjonen til at verdien under funksjonen av et vilkårlig element i domenet til funksjonen, kan fremskaffes (fra spesifikasjonen). Vi presiserer:

Definisjon 2.2 *La A være en Σ-algebra. La E være en endelig ligningsmengde. For en n-ær funksjon f_A i A er E en **(algebraisk funksjons-)spesifikasjon** av f_A, dersom følgende to krav er oppfylt:*

1. ***Kompletthet:***
   $f_A(a_1, \ldots, a_n) = b \quad$ *for* $a_1, \ldots, a_n, b \in A$
   $\Downarrow$
   det fins en $f \in \Sigma$ *og* $g_1, \ldots, g_n, g \in \mathcal{G}_\Sigma$, *slik at* f_A *er tolkningen av* f *i* A, *og* $a_1, \ldots, a_n, b$ *er tolkningene av hhv.* $g_1, \ldots, g_n, g$, *og slik at*

   $f(g_1, \ldots, g_n) \overset{*}{\underset{E}{\leftrightarrow}} g$

2. ***Sunnhet:***
   For vilkårlige $f \in \Sigma$ *og* $g_1, \ldots, g_n, g \in \mathcal{G}_\Sigma$

$$f(g_1, \ldots, g_n) \overset{*}{\underset{E}{\leftrightarrow}} g \;\Rightarrow\; A \models f(g_1, \ldots, g_n) = g$$

Hvis punkt 2 er oppfylt, kalles E *en* ***(algebraisk funksjons-)beskrivelse*** *av* f_A*.*

Ved modellering av funksjoner i en abstrakt datatype, er altså målet å finne algebraiske spesifikasjoner av funksjonene. Merk at punkt 1 i definisjon 2.2 fordrer full uttrykkbarhet.

Eksempel 9 Følgende ligningsmengde

$$E_P = \left\{ \begin{array}{l} \mathsf{x{+}0 = x}, \\ \mathsf{x{+}succ(y) = succ(x{+}y)} \end{array} \right\}$$

er en algebraisk spesifikasjon av funksjonen $+$ i Nat^+-algebraen $\mathcal{N}at^+$ fra eksempel 2. Dette er lett å vise ved induksjon over termoppbygningen til grunntermer i $\mathcal{G}_{\mathsf{Nat}^+}$: Vi viser at alle ligninger

$$\mathsf{succ}^n(0){+}\mathsf{succ}^m(0) = \mathsf{succ}^p(0)$$

er utledbare i E_P for $p = n + m$, og *ikke* utledbare i E_P for $p \neq n + m$. Spesielt sistnevnte oppgave forenkles kraftig ved å innse at E_P er *konvergent* (se avsnitt 2.4.2 på side 49). Grunntermtolken $\phi^{\mathcal{N}at^+}_{\mathcal{G}_{\mathsf{Nat}^+}}$ er selvfølgelig surjektiv.
○

Det er mange relasjoner som kan defineres på verdimengden til en abstrakt datatype. (Ved definisjonen har algebraer kun funksjoner assosiert med seg. Men enhver relasjon har en funksjonsvariant i en algebra utvidet med en bæremengde $Bool = \{\bot, \top\}$.) Alle datatyper har imidlertid naturlig en identitetsrelasjon knyttet til seg. Det er vesentlig i programmering at identitetsrelasjonen på en datatype er implementert. Implementasjonen av identitetsrelasjonen er også vesentlig for den formelle resonnering vi er interessert i. For en Σ-algebra A, er identitetsrelasjonen på A *implementert* av en ligningsmengde E hvis vi har full uttrykkbarhet og

$$g \overset{*}{\underset{E}{\leftrightarrow}} g' \;\Leftrightarrow\; A \models g = g' \tag{2.3}$$

for alle $g, g' \in \mathcal{G}_\Sigma$. I tilfellet en-til-en generatorunivers, har vi at identitetsrelasjonen er implementert av E, hvis og bare hvis E er en algebraisk spesifikasjon av alle funksjonene i A. Dette sees lett ved å observere at punkt 1 i definisjon 2.2 gitt full uttrykkbarhet og en-til-en generatorunivers, kan uttrykkes ved

1'. For vilkårlige $f \in \Sigma$ *og* $g_1, \ldots, g_n, g \in \mathcal{G}_\Sigma$

$$f(g_1, \ldots, g_n) \overset{*}{\underset{E}{\leftrightarrow}} g \;\Leftarrow\; A \models f(g_1, \ldots, g_n) = g$$

Eksempel 10 Umiddelbart kan vi altså si at Peano-aksiomene E_P og $\mathcal{N}at^+$ fra eksempel 9 tilfredstiller (2.3), dvs.

$$g \overset{*}{\underset{E_P}{\leftrightarrow}} g' \;\Leftrightarrow\; \mathcal{N}at^+ \models g = g'$$

for alle $g, g' \in \mathcal{G}_{\mathsf{Nat}^+}$.
○

I tilfellet mange-til-en generatorunivers, vil en ligningsmengde E implementere identitetsrelasjonen på A hvis og bare hvis E er en algebraisk spesifikasjon av alle funksjonene i A, og E i tillegg spesifiserer hvilke generatortermer som skal forstås som like, dvs. for alle generatortermer $g_c, g'_c \in \mathcal{G}_\Sigma$:

$$g_c \overset{*}{\underset{E}{\leftrightarrow}} g'_c \;\Leftrightarrow\; A \models g_c = g'_c \tag{2.4}$$

Isåfall kan punkt 1 i definisjon 2.2 gitt full uttrykkbarhet også nå uttrykkes ved punkt 1'.

Eksempel 11 Betrakt Int^+ og $\mathcal{I}nt^+$ fra eksempel 2. Betrakt følgende ligningsmengder

$$E_{+_{\mathbb{Z}}} = \left\{ \begin{array}{l} \mathsf{x{+}0 = x}, \\ \mathsf{x{+}succ(y) = succ(x{+}y)}, \\ \mathsf{x{+}pred(y) = pred(x{+}y)} \end{array} \right\}$$

$$E_{\mathcal{I}nt_c} = \left\{ \begin{array}{l} \mathsf{succ(pred(x)) = x}, \\ \mathsf{pred(succ(x)) = x} \end{array} \right\}$$

$E_{+_{\mathbb{Z}}}$ er en algebraisk spesifikasjon av addisjon på hele tall, og $E_{\mathcal{I}nt_c}$ tilfredstiller (2.4), dvs.

$$g_c \overset{*}{\underset{E_{\mathcal{I}nt_c}}{\leftrightarrow}} g'_c \;\Leftrightarrow\; \mathcal{I}nt^+ \models g_c = g'_c$$

for alle $g_c, g'_c \in \mathcal{G}_{\mathsf{Int}}$. Grunntermtolken er her surjektiv, så da er det lett å se at identitetsrelasjonen på $\mathcal{I}nt^+$ er implementert ved $E_{\mathcal{I}nt^+} = E_{+_{\mathbb{Z}}} \cup E_{\mathcal{I}nt_c}$. D.v.s. vi har (2.3):

$$g \overset{*}{\underset{E_{\mathcal{I}nt^+}}{\leftrightarrow}} g' \;\Leftrightarrow\; \mathcal{I}nt^+ \models g = g'$$

for alle $g, g \in \mathcal{G}_{\mathsf{Int}^+}$.
○

Ialt er det da for en Σ-algebra A rimelig å si at $\mathcal{G}_\Sigma$ og en ligningsmengde E som tilfredstiller (2.3) er, gitt full uttrykkbarhet, en *korrekt implementasjon* av A.

Ligninger sammen med ligningslogikk utgjør på denne måten den operasjonelle delen av implementasjon: Ligninger sammen med ligningslogikk implementerer funksjoner og identitetsrelasjoner for abstrakte datatyper. I tråd med dette er det naturlig å se på ligninger som ***(ikke-deterministiske) abstrakte programmer***.

2.2.9 Formell resonnering

Vårt mål for en ligningsmengde er i vår sammenheng, korrekt implementasjon uttrykt ved (2.3) på side 20. Vi vet f.eks. at Peano-aksiomene E_P fra eksempel 9 er en algebraisk spesifikasjon av addisjon på de naturlige tall, og er således en korrekt implementasjon av algebraen $\mathcal{N}at^+$ (eksempel 10).

Men det er ikke alltid like lett å avgjøre om en ligningsmengde er en korrekt implementasjon som det er for Peano-aksiomene og algebraen $\mathcal{N}at^+$ i eksempel 10. I tilnærminger til korrekt implementasjon av en abstrakt datatype, vil man derfor generelt ta utgangspunkt i viten om visse (semantiske) egenskaper i den abstrakte datatypen, og så søke å verifisere at en ligningsmengde implementerer disse egenskaper (i det minste).

Sett f.eks. at vi ikke allerede visste at E_P tilfredstiller

$$g \overset{*}{\underset{E_P}{\leftrightarrow}} g' \;\Leftrightarrow\; \mathcal{N}at^+ \models g = g'$$

for alle $g, g' \in \mathcal{G}_{\mathsf{Nat}^+}$. Siden vi vet at addisjon på naturlige tall er f.eks. kommutativ; altså at $n+m = m+n$ for alle naturlige tall m, n, dvs. $\mathcal{N}at^+ \models \mathsf{x+y = y+x}$, må E_P for å være en korrekt implementasjon av $\mathcal{N}at^+$ i det minste tilfredstille

$$\mathsf{x}\sigma\mathsf{+y}\sigma \overset{*}{\underset{E_P}{\leftrightarrow}} \mathsf{y}\sigma\mathsf{+x}\sigma$$

for hver instansiering $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\mathsf{Nat}^+}}$. Fra ligningene E_P er det derfor interessant å kunne utlede kommutativitets-ligningen $\mathsf{x+y = y+x}$ formelt og aller helst mekanisk.

Så generelt; dersom vi vet for en abstrakt datatype A at

$$A \models s = t$$

så er det for en ligningsmengde E interessant å etablere påstander som

$$\forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \overset{*}{\underset{E}{\leftrightarrow}} t\sigma \tag{2.5}$$

i tilknytning til verifiseringen av om E er del av en korrekt implementasjon av A.

Således resonnerer vi formelt om abstrakte programmer i håp om å kunne etablere korrekt implementasjon av abstrakte datatyper. For å videreføre dette (ned) til konkrete programmer, anta eksempelvis en funksjonsprosedyre **pluss(x: nat, y: nat)** ment å implementere addisjon på naturlige tall. Anta vi har vist at **pluss(x: nat, y: nat)** har de abstrakte egenskaper uttrykt i E_P (ved svak simulering (se kapittel 1). Det å etablere $\mathsf{x}\sigma\mathsf{+y}\sigma \overset{*}{\underset{E_P}{\leftrightarrow}} \mathsf{y}\sigma\mathsf{+x}\sigma$ for alle grunnsubstitusjoner (instansieringer) $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\mathsf{Nat}^+}}$, tilsvarerer da å vise at **pluss(m,n)** og **pluss(n,m)** evaluerer til det samme for alle aktualparametre **m,n** av type **nat** (under forutsetning av ingen runtime-feil).

2.3 Semantikk

I dette avsnitt skal vi presisere og utvikle begrepet 'formell datatype'.

Mens vi holder fast ved den konstruktive måten å spesifisere funksjoner på gitt ved algebraisk funksjons-spesifikasjon (definisjon 2.2), skal vi betrakte andre måter å spesifisere identitet mellom generatortermer på, enn den ved direkte omskrivning til syntaktisk like termer. Vi finner derfor en generalisering av kravet for 'korrekt implementasjon' (2.3) på side 20. Vi definerer så utfra denne en entitet som skal fungere som en mal for de formelle datatyper som defineres.

2.3.1 Implementasjon

Grunnterm-algebraer er utgangspunktet for formelle datatyper. En grunnterm-algebra med sine noe uinteressante funksjoner kan betraktes, om man vil, som en «rå-maskin» uten annen evne enn å generere (ved sine funksjoner) det rekursivt tellbare antall termer fra en endelig signatur. Siden vi vil implementere abstrakte datatyper ved formelle datatyper, skal vi prøve å «programmere» slike «rå-maskiner». Først ser vi hvordan grunnterm-algebraer kan tilføres mer struktur.

La A være en vilkårlig Σ-algebra for en signatur Σ. En ***kongruensrelasjon*** $\simeq$ på A er en ekvivalensrelasjon på A som er ***kongruent***, dvs. tilfredstiller for hvert funksjonssymbol f med aritet n i Σ:

$$a_1 \simeq b_1, \ldots, a_n \simeq b_n \;\Rightarrow\; f_A(a_1, \ldots, a_n) \simeq f_A(b_1, \ldots, b_n)$$

for vilkårlige $a_i, b_i \in A$, $1 \leq i \leq n$, der f_A er tolkningen av f i A. Vi skal her definere kongruensrelasjoner kun i tilknytning til algebraer.

Kvotient-algebraen av A ***over*** $\simeq$, skrevet $A/\simeq$ for en vilkårlig kongruensrelasjon $\simeq$, er Σ-algebraen med bæremengde $\{[a]_\simeq \mid a \in A\}$ der $[a]_\simeq = \{b \in A \mid b \simeq a\}$ kalt ***kongruensklassen til*** a, og med funksjonsmengde bestående av en funksjon $f_{A/\simeq}$ for hver f i Σ slik at $f_{A/\simeq}([a_1]_\simeq, \ldots, [a_n]_\simeq) = [f(a_1, \ldots, a_n)]_\simeq$ for vilkårlige $a_1, \ldots, a_n \in A$. Merk at $f_{A/\simeq}$ er veldefinert i kraft av at $\simeq$ er en kongruensrelasjon. Vi har

$$A/\simeq \models a = a' \;\Leftrightarrow\; a, a' \text{ er i samme } q \in A/\simeq \;\Leftrightarrow\; a \simeq a' \tag{2.6}$$

for alle $a, a' \in A$.

Vi skal nå betrakte en spesiell type kongruensrelasjon:

Sats 2.1 *La A og B være vilkårlige Σ-algebraer for en signatur Σ. For en homomorfi ϕ_A^B fra A til B, er relasjonen $\simeq$ i A slik at*

$$a \simeq a' \;\Leftrightarrow\; \phi_A^B(a) = \phi_A^B(a')$$

en kongruensrelasjon.

Godtgjørelse: Ekvivalensegenskapene følger fra at identitetsrelasjonen i B er en ekvivalensrelasjon. Kongruensegenskapen følger fra definisjonen av homomorfi. $\diamond$

Relasjonen $\simeq$ i sats 2.1 kalles ***kongruensrelasjonen indusert (i*** A***) av*** ϕ_A^B.

Spesielt, la A være en vilkårlig Σ-algebra for en signatur Σ, og betrakt kongruensrelasjonen $\simeq^A_{\mathcal{G}_\Sigma}$ indusert i $\mathcal{G}_\Sigma$ av den unike grunntermtolk $\phi^A_{\mathcal{G}_\Sigma}$. Denne relasjon er ved (2.1) på side 16 slik at for alle $g, g' \in \mathcal{G}_\Sigma$

$$g \simeq^A_{\mathcal{G}_\Sigma} g' \;\Leftrightarrow\; \phi^A_{\mathcal{G}_\Sigma}(g) = \phi^A_{\mathcal{G}_\Sigma}(g') \;\Leftrightarrow\; A \models g = g' \tag{2.7}$$

Betrakt så kvotientalgebraen $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$. Bæremengden i algebraen $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ består av klasser av termer. Hver klasse består av termer som tolkes likt i A av grunntermtolken $\phi^A_{\mathcal{G}_\Sigma}$. For $\phi^A_{\mathcal{G}_\Sigma}$ surjektiv, kaller vi $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ ***implementasjonen*** av A. Algebraen $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ skal fungere som en mal og et mål ved definisjonen av formelle datatyper.

Vi skal etablere noen egenskaper ved implementasjoner. Generelt har vi for en kongruensrelasjon $\simeq$ på en $\mathcal{G}_\Sigma$ følgende:

1. $\mathcal{G}_\Sigma/\simeq \models g = g' \;\Leftrightarrow\; g \simeq g'$, for alle $g, g' \in \mathcal{G}_\Sigma$ (Følger ved (2.6).)
2. $\mathcal{G}_\Sigma/\simeq \models s = t \;\Leftrightarrow\; \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \simeq t\sigma$ (Følger ved (2.2) på side 17, siden homomorfien fra $\mathcal{G}_\Sigma$ til $\mathcal{G}_\Sigma/\simeq$ er surjektiv.)

For en $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ og en Σ-algebra A, har vi derfor ved punkt 1 og (2.7) induktiv ekvivalens med A:

$$\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma} \models g = g' \;\Leftrightarrow\; A \models g = g' \tag{2.8}$$

for alle $g, g' \in \mathcal{G}_\Sigma$. Dersom $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ er implementasjonen av A (surjektiv $\phi^A_{\mathcal{G}_\Sigma}$), har vi dessuten ved punkt 2 og (2.2) side 17 elementær ekvivalens:

$$\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma} \models s = t \;\Leftrightarrow\; A \models s = t \tag{2.9}$$

for alle $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$. Dette innebærer en strukturlikhet mellom en algebra A og implementasjonen av A til den grad at enhver påstand uttrykt ved ligninger er sann i implementasjonen hvis og bare hvis påstanden er sann i A. Strukturen i algebraene $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ og A er dessuten like i den forstand at det kun er *navnene* på komponentene som er forskjellige (*isomorfi*).

Kongruensrelasjoner er i kraft av kongruensklassene de induserer i term-algebraer, istand til å beskrive hvilke termer i et termunivers som skal forstås som like. Vi skal derfor si at en kongruensrelasjon på en $\mathcal{G}_\Sigma$ utgjør en ***semantikk*** for $\mathcal{G}_\Sigma$. (Ekvivalens- og kongruensegenskapene ved en kongruensrelasjon er essensielle for slik semantikkgiving. Det er innlysende at ekvivalensegenskapene i denne sammenheng er essensielle. Kongruensegenskapen sikrer at funksjonssymboler (alltid) kan tolkes til funksjoner. Anta nemlig for en relasjon $\Re$ ment å utgjøre en semantikk, at $g \Re g'$ men $f(g) \not\Re f(g')$. M.a.o. g og g' skal tolkes til samme objekt, men $f(g)$ og $f(g')$ skal tolkes til forskjellige objekter. Da kan ikke funksjonssymbolet f tolkes til en funksjon.) Semantikken der alle termer skal tolkes forskjellig ($\emptyset^0_{\mathcal{G}_\Sigma}$ —den refleksive tillukning av $\emptyset$ på $\mathcal{G}_\Sigma$), skal vi kalle ***fri***. Vi skal ofte også si at funksjonssymboler gis semantikk av en kongruensrelasjon.

En detalj: I diskusjonen utover skal ligninger og ligningslogikk være sentral i forskjellige måter å definere semantikk på. I den forbindelse fremmer vi et krav om at termer av ulik type ikke skal stå i relasjon til hverandre i en semantikk. Dette kravet oppfylles sålenge domenet til semantikken ikke kun består av konstanter. I motsatt fall krever vi:

$$k \simeq k' \text{ og } k, k' \text{ er konstanter} \quad \Rightarrow \quad k, k' \text{ er av samme type} \tag{2.10}$$

Ofte vil de semantikker vi spesifiserer oppfylle (2.10) automatisk, men noen ganger skal vi bli nødt til presisere (2.10) eksplisitt.

I avsnitt A.1 i tilleggskapittel A utvikles noen flere begreper og utdypende sammenhenger omkring semantikk, induktiv og elementær ekvivalens og isomorfi, men som ikke har *essensiell* betydning for vår diskusjon videre her.

Konguensrelasjoner tilfører struktur på term-algebraer. Den ønskede struktur i en grunnterm-algebra $\mathcal{G}_\Sigma$ i forhold til en algebra A, gis altså ved kongruensrelasjonen $\simeq^A_{\mathcal{G}_\Sigma}$ indusert av den unike grunntermtolken $\phi^A_{\mathcal{G}_\Sigma}$. Vi skal nå se hvordan slik struktur kan gjøres tilgjengelig for mekanisk håndtering.

2.3.2 Initialsemantikk

Den første «instansensieringen» av vårt nå presise begrep 'implementasjon' følger lett fra diskusjonen om 'korrekt implementasjon' i avsnitt 2.2.8 og uttrykt i (2.3) på side 20. Følgende resultat er essensielt:

Sats 2.2 *La Σ være en vilkårlig signatur. For en vilkårlig $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ er $\overset{*}{\underset{E}{\leftrightarrow}}$ en kongruensrelasjon på $\mathcal{T}_\Sigma(\mathcal{V})$.*

Bevis: Siden $\overset{*}{\underset{E}{\leftrightarrow}}$ er en refleksiv-symmetrisk-transitiv tillukning, er det innlysende at $\overset{*}{\underset{E}{\leftrightarrow}}$ er refleksiv, symmetrisk og transitiv på $\mathcal{T}_\Sigma(\mathcal{V})$. La så f_T være en vilkårlig funksjon i $\mathcal{T}_\Sigma(\mathcal{V})$. Anta f_T er n-ær. Anta $s_i \overset{*}{\underset{E}{\leftrightarrow}} t_i$ for vilkårlige $s_i, t_i \in \mathcal{T}_\Sigma(\mathcal{V}), 1 \leq i \leq n$. Ved monotonitet mhp. kontekstapplikasjon får vi $f_T(s_1, \ldots, s_n) \overset{*}{\underset{E}{\leftrightarrow}} f_T(t_1, \ldots, t_n)$, og $\overset{*}{\underset{E}{\leftrightarrow}}$ er en kongruensrelasjon på $\mathcal{T}_\Sigma(\mathcal{V})$.
□

For et vilkårlig redukt $\mathcal{T}$ av $\mathcal{T}_\Sigma(\mathcal{V})$ er så restriksjonen $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{T}}$ av $\overset{*}{\underset{E}{\leftrightarrow}}$, ved sats A.12 side 173, en kongruensrelasjon på $\mathcal{T}$. Vi skriver $\mathcal{T}/E$ for kvotient-algebraen av $\mathcal{T}$ over $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{T}}$.

Definisjon 2.3 *For en signatur Σ og ligningsmengde E, kaller vi kvotientalgebraen $\mathcal{G}_\Sigma/E$ den **formelle basis-initielle datatypen** spesifisert av Σ og E. Relasjonen $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma}$ kalles **basis-initialsemantikken** spesifisert av Σ og E.*

Det viktige er nå at *ligningsmengder sammen med ligningslogikk gir en formell-syntaktisk måte å spesifisere kongruensrelasjoner i grunnterm-algebraer på.*

Definisjon 2.4 *La $\simeq$ være en vilkårlig kongruensrelasjon på (domenet til) en term-algebra $\mathcal{T}$. En endelig ligningsmengde E er en* ***(direkte) algebraisk spesifikasjon av*** $\simeq$ *dersom*

$$\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{T}} = \simeq$$

Gitt en Σ-algebra A, er vårt mål å finne en ligningsmengde som er en algebraisk spesifikasjon av $\simeq^A_{\mathcal{G}_\Sigma}$. Dersom $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv, er nemlig den formelle datatypen $\mathcal{G}_\Sigma/E$ i så fall implementasjonen av A.

Eksempel 12 Ligningsmengden $E_{\mathcal{I}nt^+} = E_{+_{\mathbb{Z}}} \cup E_{\mathcal{I}nt_c}$ fra eksempel 11 side 21 er en algebraisk spesifikasjon av $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}^+}}$. Dette siden

$$g \; {}_E\overset{*}{\underset{\mathcal{I}nt^+}{\leftrightarrow}} \; g' \;\Leftrightarrow\; \mathcal{I}nt^+ \models g = g'$$

for alle $g, g \in \mathcal{G}_{\mathsf{Int}^+}$. Altså er restriksjonen ${}_E\overset{*}{\underset{\mathcal{I}nt^+}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}^+}$ av ${}_E\overset{*}{\underset{\mathcal{I}nt^+}{\leftrightarrow}}$ identisk med $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}^+}}$. Den formelle datatypen $\mathcal{G}_{\mathsf{Int}^+}/E_{\mathcal{I}nt^+}$ er da implementasjonen av $\mathcal{I}nt^+$.
○

*

Vi skal nå definere andre typer semantikk. Diskusjonen kommer nå i hovedsak til å relatere seg til det syntaktiske plan. Vi skal da finne det hensiktsmessig å identifisere de funksjonssymboler i en signatur med tilsiktet tolkning generatorfunksjoner og de symboler med tilsiktet tolkning funksjoner på mengden generert av generatorfunksjonene. En signatur Σ skal da sees som delt disjunkt i Σ^c bestående ***generator(profil)er***, og Σ^d bestående av ***definerte funksjonssymbol(profil)er***.

Følgende definisjon ekstraherer syntaktiske egenskaper fra begrepet 'algebraisk funksjonsspesifikasjon' definert i definisjon 2.2 på side 19.

Definisjon 2.5 **(**en utvidelse av et begrep i [Gut77]**)** *La Σ være en signatur som her bestemmer termuniverset. La $\Sigma^c \subseteq \Sigma$ være mengden av alle generatorprofiler i Σ. La videre $\Sigma^x \subseteq \Sigma^c$, og la $\mathcal{T}ype^x \subseteq \mathcal{T}ype$ være mengden av typer som termene i $\mathcal{G}_{\Sigma^x}$ er av.*

For en ligningsmengde E og en $\Sigma^f \subseteq \Sigma$, er E ***tilstrekkelig Σ^f-komplett*** *mhp. $\mathcal{G}_{\Sigma^x}$, dersom det for enhver $g \in \mathcal{G}_{\Sigma^c \cup \Sigma^f}$, g av en type i $\mathcal{T}ype^x$, finnes en $g_x \in \mathcal{G}_{\Sigma^x}$ slik at $g \overset{*}{\underset{E}{\leftrightarrow}} g_x$. Dersom E er tilstrekkelig Σ-komplett mhp. $\mathcal{G}_{\Sigma^x}$, sier vi bare at E er* ***tilstrekkelig komplett*** *mhp. $\mathcal{G}_{\Sigma^x}$.*

Eksempel 13 Vi deler signaturen $\mathsf{Nat}^+ = \{\mathsf{0}, \mathsf{succ}, +\}$ inn i generatorer $\mathsf{Nat} = \{\mathsf{0}, \mathsf{succ}\}$ og definerte funksjonssymboler $\{+\}$. Ligningsmengden E_P fra eksempel 9 på side 20 er da tilstrekkelig $\{+\}$-komplett mhp. $\mathcal{G}_{\mathsf{Nat}}$, siden enhver term i $\mathcal{G}_{\mathsf{Nat}^+}$ kan omskrives i E_P til en generatorterm i $\mathcal{G}_{\mathsf{Nat}}$.
○

Tilstrekkelig kompletthet uttrykker den syntaktiske del av konstruktiv funksjonsdefinisjon (se begynnelsen av avsnitt 2.2.8 side 19). Tilstrekkelig kompletthet mhp. generatorer innebærer en ***fullstendig definerthet*** av definerte funksjonssymboler; i den forstand at et funksjonssymbol kan tolkes til høyst én funksjon i en gitt algebra. M.a.o. semantikk til funksjonssymboler er spesifisert til en slik grad, at det ikke er rom for fler enn én tolkning av et gitt funksjonssymbol. Vi presiserer: Anta gitt et funksjonssymbol $f \in \Sigma$, si av en type $T_D \rightarrow T_C$ der T_C er en generatortype, og anta gitt en ligningsmengde E tilstrekkelig f-komplett mhp. $\mathcal{G}_{\Sigma T_C}$ (husk fra definisjonen av algebra at $\mathcal{G}_{\Sigma T_C}$ betegner tolkningen av T_C i $\mathcal{G}_\Sigma$; altså mengden av alle generatorer av type T_C). Anta så

at E inneholder en kopi av spesifikasjonen for f, men der alle forekomster av f er byttet ut med et annet symbol f' av samme type som f. Anta nå at det finnes en algebra A slik at $g \overset{*}{\underset{E}{\leftrightarrow}} g' \Rightarrow A \models g = g'$ for alle $g, g' \in \mathcal{G}_\Sigma$ (A sies å *realisere* semantikken på $\mathcal{G}_\Sigma$ gitt av E. Se ellers avsnitt A.1 i tillegg A). Vi skal vise at at tolkningene f_A og f'_A i A av f og f' ikke er identiske.

For enhver g slik at $f(g)$ og $f'(g)$ er av type T_C, er det lett å se at

$$f(g) \overset{*}{\underset{E}{\leftrightarrow}} g_c$$

og

$$f'(g) \overset{*}{\underset{E}{\leftrightarrow}} g_c$$

for en $g_c \in \mathcal{G}_{\Sigma T_C}$. Betrakt så en vilkårlig a i domenet til f_A og f'_A. Antar vi surjektiv grunnterm-tolk $\phi^A_{\mathcal{G}_\Sigma}$, har vi fra dette spesielt $f(g_a) \overset{*}{\underset{E}{\leftrightarrow}} g_{a_c}$ og $f'(g_a) \overset{*}{\underset{E}{\leftrightarrow}} g_{a_c}$ for en g_a slik at $\phi^A_{\mathcal{G}_\Sigma}(g_a) = a$. Vi har altså

$$f(g_a) \overset{*}{\underset{E}{\leftrightarrow}} g_{a_c} \overset{*}{\underset{E}{\leftrightarrow}} f'(g_a)$$

Siden A realiserer semantikken gitt av E, må derfor $f_A(a) = f'_A(a)$. Følgelig må tolkningene av f og f' i A være identiske.

Dette begrepet om fullstendig definerthet er beslektet med begrepet *implisitt definerthet* fra predikatlogikk. Se f.eks [GJ67].

Det rake motsatte av tilstrekkelig kompletthet skal senere være av interesse for oss, og defineres som følger:

Definisjon 2.6 *La som i definisjon 2.5 Σ være en signatur som bestemmer vårt termunivers. La $\Sigma^c \subseteq \Sigma$ være mengden av alle generatorprofiler i Σ. La videre $\Sigma^x \subseteq \Sigma^c$, og la $Type^x \subseteq Type$ være mengden av typer som termene i $\mathcal{G}_{\Sigma^x}$ er av.*

*For en ligningsmengde E og en $\Sigma^f \subseteq \Sigma$, er E **fullstendig Σ^f-ukomplett** mhp. $\mathcal{G}_{\Sigma^x}$, dersom det ikke for noen $g \in \mathcal{G}_{\Sigma^c \cup \Sigma^f} \setminus \mathcal{G}_{\Sigma^c}$, g av en type i $Type^x$, finnes noen $g_x \in \mathcal{G}_{\Sigma^x}$ slik at $g \overset{*}{\underset{E}{\leftrightarrow}} g_x$. Dersom E er fullstendig Σ-ukomplett mhp. $\mathcal{G}_{\Sigma^x}$ sier vi bare at E er **fullstendig ukomplett** mhp. $\mathcal{G}_{\Sigma^x}$.*

Eksempel 13 (forts.) Ligningsmengden

$$E_{ass} = \{\mathsf{x+y = y+x}\}$$

er fullstendig $\{+\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Nat}}$, siden ingen term i $\mathcal{G}_{\mathsf{Nat}^+}$ med forekomster av symbolet $+$ kan omskrives ved E_{ass} til en generatorterm i $\mathcal{G}_{\mathsf{Nat}}$.
○

2.3.3 Finalsemantikk

Ligninger sammen med ligningslogikk gir initialsemantikk ved direkte omskrivning til syntaktisk like termer. Semantikk kan også bestemmes ved følgende prinsipp: La $\simeq$ være en kongruensrelasjon på en $\mathcal{G}_\Sigma$. Vi har da for alle $c, g, g' \in \mathcal{G}_\Sigma$:

$$c[g] \not\simeq c[g'] \Rightarrow g \not\simeq g' \tag{2.11}$$

Det faktum at $c[g]$ ikke er kongruent med $c[g']$ bestemmer at $g \not\simeq g'$. En del av semantikken blir her *bestemt* av en annen *bestemmende* del av semantikken.

Abstrakte maskiner skjelner kun syntaks. Elementene i bæremengden til en term-algebra er symbolene i en «rå-maskin», og hver term er forskjellig fra enhver annen term. Hvilke termer som skal forstås som like må eksplisitt «programmeres» inn i «rå-maskinen». Algebraisk spesifikasjon gjør nettopp dette

(i form av abstrakte programmer). I tråd med dette kan man si at initialsemantikk spesifiserer to termer som semantisk ulike med mindre den algebraiske spesifikasjonen spesifiserer dem som semantisk like.

En annen tilnærmingsmåte er å først gi den universelle semantikk for hver type til «rå-maskinen», for deretter å spesifisere *ulikheter*. M.a.o. for hver type, spesifiseres to termer som semantisk like med mindre de blir *tvunget semantisk ulike* ved prinsippet i (2.11). Følgende definisjon beskriver en *pseudo*-semantikk gitt på denne måten. Denne pseudo-semantikken er ikke nødvendigvis en kongruensrelasjon. Den formelle maskin er også her ligningslogikk.

Definisjon 2.7 *Betrakt en vilkårlig $\mathcal{G}_\Sigma$. La $\mathcal{G}_{\Sigma^x} \subseteq \mathcal{G}_\Sigma$. La $\simeq^x$ være en gitt semantikk på $\mathcal{G}_{\Sigma^x}$. For en ligningsmengde E defineres en relasjon $\simeq^\omega$ på $\mathcal{G}_\Sigma$ slik: For alle $g, g' \in \mathcal{G}_\Sigma$*

$$g \simeq^\omega g' \quad \Leftrightarrow \quad \left\{ \begin{array}{l} g, g' \text{ er av samme type} \quad og \\ \neg\exists c \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^x} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'] \end{array} \right.$$

*Vi kaller $\simeq^\omega$ den **finale pseudo-semantikk relativ til** $\simeq^x$ spesifisert av Σ og E. Vi kaller $\simeq^x$ **kjernen** i $\simeq^\omega$.*

*Final (pseudo-)semantikk med fri eller basis-initiell kjerne spesifisert av E, skal vi kalle **basis-final (pseudo-)semantikk.***

Semantikken $\simeq^x$ er den bestemmende semantikk. Semantikken $\simeq^x$ gir «ankerfester» i hvilke semantisk ulikhet «trekkes ut av» den universelle kongruensrelasjon på $\mathcal{G}_{\Sigma T}$, for hver type T som forekommer i Σ. Vi har at $\simeq^\omega$ er *maksimal* i den forstand at $\simeq^\omega$ ikke spesifiserer andre semantiske ulikheter enn dem som er tvingende nødvendige utfra typing, $\simeq^x$ og (2.11). Vi legger merke til følgende:

Observasjon 2.3 *For en Σ, $\simeq^x$ på $\mathcal{G}_{\Sigma^x} \subseteq \mathcal{G}_\Sigma$ og $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$, la $\simeq^\omega$ være den finale pseudo-semantikk relativ til $\simeq^x$ spesifisert av Σ og E. Alle ulikheter i $\simeq^x$ er bevart i $\simeq^\omega$, dvs.*

$$g \not\simeq^x g' \quad \Rightarrow \quad g \not\simeq^\omega g'$$

Vi nevnte at $\simeq^\omega$ i definisjon 2.7 ikke nødvendigvis er en kongruensrelasjon. Nærmere bestemt er ikke refleksivitet og transitivitet nødvendigvis tilfredsstilt:

Eksempel 14 La $\simeq^x$ være den frie semantikk på $\mathcal{G}_{\{a,b\}}$ for konstanter a, b. La $E = \{a = b\}$. Da har vi $a \overset{*}{\underset{E}{\leftrightarrow}} a \not\simeq^x b \overset{*}{\underset{E}{\leftrightarrow}} a$, så $a \not\simeq^\omega a$.
○

Eksempel 15 La så $\simeq^x$ på $\mathcal{G}_{\{a,b,c,d\}}$ være slik at $a \not\simeq^x b$, $c \simeq^x d$ og $d \simeq^x e$. La $E = \{f(c) = a, f(e) = b\}$. Da har vi $f(c) \underset{E}{\leftrightarrow} a \not\simeq^x b \overset{*}{\underset{E}{\leftrightarrow}} f(e)$, så $c \not\simeq^\omega e$. Men $c \simeq^\omega d$ og $d \simeq^x e$. (Dette sees for $c \simeq^\omega d$ ved 1) $c \simeq^x d$ og 2) det ikke finnes noen $\alpha \in \mathcal{G}_{\{a,b,c,d\}}$ slik at $f(d) \overset{*}{\underset{E}{\leftrightarrow}} \alpha$.)
○

Vi har dog:

Sats 2.4 *Relasjonen $\simeq^\omega$ i definisjon 2.7 er symmetrisk og kongruent.*

Bevis: La signaturer, relasjoner og E være som i definisjon 2.7. Symmetri for $\simeq^\omega$ følger trivielt ved at både $\overset{*}{\underset{E}{\leftrightarrow}}$ og $\simeq^x$ er symmetriske. Anta så at $g \simeq^\omega g'$ for vilkårlige $g, g' \in \mathcal{G}_\Sigma$. Anta derimot at $c[g] \not\simeq^\omega c[g']$ for en kontekst $c \in \mathcal{G}_\Sigma$. M.a.o. det finnes $c' \in \mathcal{G}_\Sigma, g_1, g_2 \in \mathcal{G}_{\Sigma^c}$ slik at

$$c'[c[g]] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c'[c[g']]$$

Altså gir konteksten $c'' = c[c']$ at $g \not\simeq^{\omega} g'$; som er en motsigelse.
□

Final pseudo-semantikk er altså ikke nødvendigvis en kongruensrelasjon, og er derfor generelt ubrukelig om ønsket er å oppnå en implementasjon av en abstrakt datatype. Imidlertid er det i vår spesifikasjonssammenheng naturlig at følgende kriterier er oppfylt for signaturer, relasjoner og ligningsmengde E fra definisjon 2.7:

`KONSERV:` $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_{\Sigma^x}} \subseteq \simeq^x$

`TK:` E er tilstrekkelig komplett mhp. $\mathcal{G}_{\Sigma^x}$.

Eksempelvis er `KONSERV` oppfylt hvis E spesifiserer $\simeq^x$ basis-initialsemantisk. Vårt ønske om konstruktiv algebraisk funksjonsspesifikasjon fordrer at `TK` er oppfylt.

Sats 2.5 *Under antagelsene* `KONSERV` *og* `TK`*, er relasjonen* $\simeq^{\omega}$ *i definisjon 2.7 en kongruensrelasjon.*

Bevis: Vi trenger kun å vise refleksivitet og transitivitet. La signaturer, relasjoner og E være som i definisjon 2.7. Anta $g \not\simeq^{\omega} g$ for en $g \in \mathcal{G}_{\Sigma}$. Altså

$$c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g]$$

Men da har vi $g_1 \overset{*}{\underset{E}{\leftrightarrow}} c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_2$ som ved `KONSERV` gir $g_1 \simeq^x g_2$. Dette er en motsigelse, så $g \simeq^{\omega} g$.

Anta så at $g \simeq^{\omega} g'$ og $g' \simeq^{\omega} g''$, men $g \not\simeq^{\omega} g''$ for vilkårlige $g, g', g'' \in \mathcal{G}_{\Sigma}$. Observer at g, g', g'' må være av samme type, så $g \not\simeq^{\omega} g''$ er ikke resultatet av at g og g' er av forskjellige typer. Vi har derfor

$$c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'']$$

for noen $c \in \mathcal{G}_{\Sigma}, g_1, g_2 \in \mathcal{G}_{\Sigma^x}$. (Vi observerer da at dersom E er fullstendig ukomplett mhp. $\mathcal{G}_{\Sigma^x}$, så er dette ikke mulig, så transitivitet følger i så fall umiddelbart.)

Ved `TK` finnes, siden $c[g']$ er av en type av hvilken termer i $\mathcal{G}_{\Sigma^x}$ er, en $g_3 \in \mathcal{G}_{\Sigma^x}$ slik at $c[g'] \overset{*}{\underset{E}{\leftrightarrow}} g_3$. Nå må $g_1 \simeq^x g_3$, ellers var ikke $g \simeq^{\omega} g'$. På samme måte må $g_3 \simeq^x g_2$. Men da har vi $g_1 \simeq^x g_2$ som gir en motsigelse. Følgelig må $g \simeq^{\omega} g''$.
□

Kriteriet `TK` er en sterkere enn nødvendig betingelse for sats 2.5, men vi skal ikke gå i mer detalj her enn å påpeke det som parentetisk ble antydet i beviset for sats 2.5, om at fullstendig ukompletthet mhp. $\mathcal{G}_{\Sigma^x}$ kan avløse `TK` som et tilstrekkelig kriterium for transitivitet. Vi kan nå definere:

Definisjon 2.8 *La* $\simeq^{\omega}$ *være definert som i definisjon 2.7. Anta* $\simeq^{\omega}$ *er en kongruensrelasjon. Vi kaller* $\simeq^{\omega}$ *den* ***finale semantikken relativ til*** $\simeq^x$ *spesifisert av* Σ *og* E*, og vi kaller kvotientalgebraen* $\mathcal{G}_{\Sigma}/\simeq^{\omega}$ *den* ***formelle finale datatypen relativ til*** $\simeq^x$ *spesifisert av* Σ *og* E*.*

En spesifiseringsstrategi som naturlig faller inn under finalsemantikk, er semantikkgiving ved *observatorer*. En ***observator*** er et definert funksjonssymbol $f : T_1 \times \cdots \times T_n \rightarrow T$ slik at $T \neq T_i, 1 \leq i \leq n$. La nå $\mathcal{G}_{\Sigma^{xc}}$ være generatortermer av en type T^x med semantikk $\simeq^{xc}$. La $\mathcal{G}_{\Sigma^{yc}}$ være generatortermer av en type T^y som ønskes gitt en semantikk. La Σ^o være en signatur med minst en observator $h : T_1 \times \cdots \times T^y \times \cdots \times T_{n_i} \rightarrow T^x$. Med grunnlag i semantikken $\simeq^{xc}$ skal nå semantikken $\simeq^{yc}$ på $\mathcal{G}_{\Sigma^{yc}}$ bestemmes. Både $\simeq^{xc}$ og $\simeq^{yc}$ skal betraktes som en del av en semantikk $\simeq^{\omega}$ på $\mathcal{G}_{\Sigma}$ der $\Sigma = \Sigma^{xc} \cup \Sigma^{yc} \cup \Sigma^o$, slik at

$$\simeq^{xc} = \simeq^{\omega}_{\mathcal{G}_{\Sigma^{xc}}}$$

og

$$\simeq^{yc} = \simeq^{\omega}_{\mathcal{G}_{\Sigma^{yc}}}.$$

Semantikken $\simeq^{yc}$ bestemmes nå bl.a. ved at vi for alle generatorer $g_{yc}, g'_{yc} \in \mathcal{G}_{\Sigma^{yc}}$ nødvendigvis har:

$$\exists h : \cdots \times T^y \times \cdots \rightarrow T^x \in \Sigma^o \mid h(\cdots, g_{yc}, \cdots) \not\simeq^{xc} h(\cdots, g'_{yc}, \cdots)$$

$$\Downarrow$$

$$g_{yc} \not\simeq^{yc} g'_{yc}$$

Anta $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ er tilstrekkelig komplett mhp. $\mathcal{G}_{\Sigma^{xc}}$. Anta videre at $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_{\Sigma^{xc}}} \subseteq \simeq^{xc}$. Ved sats 2.5 kan vi da betrakte den finale semantikken $\simeq^{\omega}$ relativ til $\simeq^{xc}$:

$$g \simeq^{\omega} g' \;\Leftrightarrow\; \neg\exists c \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^{xc}} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^{xc} g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

Eksempel 16 Vi betrakter binære trær. Signaturer er som følger:

$$\Sigma^{yc} = \left\{ \begin{array}{l} \varepsilon : \mathsf{Tre}, \\ \mathsf{node} : \mathsf{Tre} \times \mathsf{Tre} \rightarrow \mathsf{Tre} \end{array} \right\}$$

$$\Sigma^{xc} = \left\{ \begin{array}{l} \mathsf{0} : \mathsf{Int}, \\ \mathsf{succ} : \mathsf{Int} \rightarrow \mathsf{Int}, \\ \mathsf{pred} : \mathsf{Int} \rightarrow \mathsf{Int} \end{array} \right\}$$

$$\Sigma^{o} = \left\{ \begin{array}{l} \mathsf{bIns} : \mathsf{Tre} \rightarrow \mathsf{Int}, \\ + : \mathsf{Int} \times \mathsf{Int} \rightarrow \mathsf{Int} \end{array} \right\}$$

Følgende ligningsmengde E er algebraiske spesifikasjoner av

- en funksjon som finner differansen mellom antall venstre-noder og høyre-noder i et tre
- addisjon på hele tall, samt
- en spesifikasjon av initialsemantikk i $\mathcal{G}_{\Sigma^{xc}}$:

$$E = \left\{ \begin{array}{l} \mathsf{bIns}(\varepsilon) = \mathsf{0}, \\ \mathsf{bIns}(\mathsf{node}(\mathsf{t},\mathsf{t}')) = \mathsf{succ}(\mathsf{bIns}(\mathsf{t}))+\mathsf{pred}(\mathsf{bIns}(\mathsf{t}')), \\ \\ \mathsf{x}+\mathsf{0} = \mathsf{x}, \\ \mathsf{x}+\mathsf{succ}(\mathsf{y}) = \mathsf{succ}(\mathsf{x}+\mathsf{y}), \\ \mathsf{x}+\mathsf{pred}(\mathsf{y}) = \mathsf{pred}(\mathsf{x}+\mathsf{y}), \\ \\ \mathsf{succ}(\mathsf{pred}(\mathsf{x})) = \mathsf{x}, \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) = \mathsf{x} \end{array} \right\}$$

Basis-finalsemantikken $\simeq^{\omega}$ på $\mathcal{G}_\Sigma$ for $\Sigma = \Sigma^{xc} \cup \Sigma^{yc} \cup \Sigma^o$ slik at

$$g \simeq^{\omega} g' \;\Leftrightarrow\; \neg\exists c \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^{xc}} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \overset{\not}{\underset{E}{\leftrightarrow}} g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

gir da ved observatoren bIns og initialsemantikken i $\mathcal{G}_{\Sigma^{xc}}$, at generatortermer over $\mathcal{G}_{\Sigma^{yc}}$ er semantisk like hvis og bare hvis de representerer trær med samme forhold i antallet venstre- og høyrenoder.
○

I implementasjonssammenheng, kan en naturlig bakenforliggende tanke eller intensjon ved generell finalsemantisk spesifikasjon være følgende:

Intensjon A

1. Kjernesemantikken er semantikk på generatortermer av typer $T_1^x, \ldots, T_n^x$.
2. Ligningsmengden E gir semantikk til generatortermer av typer $T_1^y, \ldots, T_m^y$ ved prinsipp (2.11) på side 26.
3. Ligningsmengden E gir semantikk til definerte funksjonssymboler av både typer $T_1^x, \ldots, T_n^x$ og $T_1^y, \ldots, T_m^y$.

Således kan en spesifikasjonsoppgave sees på som bestående av avgrensede deler. Kjernesemantikken kan her være en initialsemantikk, eller igjen en finalsemantikk — eller andre former for semantikk. Vi skal i kapittel 3 spesifisere kjernesemantikk såkalt *indirekte*.

2.3.4 Alternativ definisjon av finalsemantikk

I [Lys92] defineres følgende semantikk, her restriktert til grunntermer: For signaturer, relasjoner og E som i definisjon 2.7

$$g \simeq^{\zeta} g' \;\Leftrightarrow\; \neg\exists c \in \mathcal{G}_\Sigma; g_x \in \mathcal{G}_{\Sigma^x} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_x \not\overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

Denne semantikken ligner vår basis-final (pseudo)semantikk, men har den vesentlige egenskap å være en kongruensrelasjon umiddelbart. Vi kan, inspirert av dette, definere en generalisert finalsemantikk $\simeq^{\omega\prime}$: For signaturer, relasjoner og E som i definisjon 2.7

$$g \simeq^{\omega\prime} g' \;\Leftrightarrow\; \neg\exists c \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^x} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \simeq^x g_2 \not\overset{*}{\underset{E}{\leftrightarrow}} c[g'] \tag{2.12}$$

Under `TK` eller fullstendig ukompletthet mhp. $\mathcal{G}_{\Sigma^x}$, er $\simeq^{\omega\prime}$ og $\simeq^{\omega}$ identiske. Forskjellen på $\simeq^{\omega\prime}$ og $\simeq^{\omega}$ oppstår altså i «gråsonen» mellom fullstendig ukompletthet og tilstrekkelig kompletthet.

Eksempel 17 Betrakt $\mathcal{G}_{\{a,b,c,d\}}$, $\simeq^x$ og E som i eksempel 15. For den finale pseudo-semantikk $\simeq^{\omega}$ har vi $c \simeq^{\omega} d$ siden det ikke finnes noen $\alpha \in \mathcal{G}_{\{a,b,c,d\}}$ slik at $f(d) \overset{*}{\underset{E}{\leftrightarrow}} \alpha$. For $\simeq^{\omega\prime}$ som definert i (2.12), har vi derimot $f(c) \not\simeq^{\omega\prime} f(d)$, siden det finnes en $\beta \in \mathcal{G}_{\{a,b,c,d\}}$ slik at $f(c) \overset{*}{\underset{E}{\leftrightarrow}} \beta$. Dermed har vi $c \not\simeq^{\omega\prime} d$.
○

Når vi fra nå av velger å bruke pseudo-semantikken $\simeq^{\omega}$ istedenfor den mer elegante $\simeq^{\omega\prime}$ definert i (2.12) og inspirert utifra semantikken definert i [Lys92], er det i hovedsak på grunnlag av følgende: 1) Det er for våre formål lettere å resonnere om og med $\simeq^{\omega}$. 2) Fullstendig ukompletthet samt «gråsonen» mellom fullstendig ukompletthet og tilstrekkelig kompletthet, er interessant i forbindelse med resonnering utfra *egenskaper* til en funksjon, i motsetning til resonnering utfra en konstruktiv spesifikasjon av funksjonen. Eksempelvis tilfredstiller algebraen $\mathcal{I}nt^+ = \langle \mathbb{Z}, \{0, succ, pred, +_{\mathbb{Z}}\}\rangle$ fra eksempel 2 utvidet med inversfunksjonen $\Leftrightarrow$, de ligningslogiske aksiomer for Abelske grupper:

$$\{\mathsf{x{+}(y{+}z) = (x{+}y){+}z},\ \mathsf{x{+}0 = 0},\ \mathsf{x{+}(-x) = 0},\ \mathsf{x{+}y = y{+}x}\}$$

For slik resonnering er trolig en semantikk basert på uttrekking av ulikheter fra en universell semantikk ikke det som ønskes for definerte funksjonssymboler. Dette fordi slik semantikk og også initialsemantikk gir ***fullstendig*** semantikk, i den forstand at likheter *og* ulikheter spesifiseres (initialsemantikk: alt som ikke spesifiseres som likt er ulikt, finalsemantikk: alt som ikke spesifiseres som ulikt er likt). Det som er ønskelig i tilfellet resonnering utfra ikke-konstruktive

egenskaper er heller såkalt ***løs*** semantikk. For aksiomene for Abelske grupper, ville vi eksempelvis trolig ønske å spesifisere to termer som semantisk like hvis de kan vises like ligningslogisk, men unnlate å spesifisere termer som semantisk ulike hvis de ikke kan vises like ligningslogisk. Semantikk som definert ved (2.12) synes i noen tilfeller (f.eks. for aksiomene over) å takle «gråsoner» bra, men for tilfellet fullstendig ukompletthet fungerer begge varianter av finalsemantikk like dårlig. I anerkjennelsen av at det først og fremst er under `TK` at finalsemantikk er ment å fungere, kan vi argumentere for å velge den semantikken som er mest «kompromissløs»: $\simeq^\omega$ som definert i definisjon 2.7 er som sagt maksimal i den forstand at $\simeq^\omega$ ikke spesifiserer andre semantiske ulikheter enn dem som er tvingende nødvendige utfra typing, kjernesemantikken $\simeq^x$ og (2.11) på side 26.

2.3.5 Initialsemantikk generalisert

En finalsemantikk defineres ved å spesifisere ulikhet med grunnlag i en (annen) gitt semantikk. Analogt skal vi nå definere generalisert initialsemantikk ved å spesifisere likhet med utgangspunkt i en (annen) gitt semantikk.

Definisjon 2.9 *Betrakt en vilkårlig $\mathcal{G}_\Sigma$. La $\mathcal{G}_{\Sigma^x} \subseteq \mathcal{G}_\Sigma$. La $\simeq^x$ være en gitt semantikk på $\mathcal{G}_{\Sigma^x}$. La E være en vilkårlig ligningsmengde. Betrakt omskrivningsrelasjonen $\Re$ på $\mathcal{G}_\Sigma$ slik at for alle $g, g' \in \mathcal{G}_\Sigma$*

$$g \;\Re\; g' \;\Leftrightarrow\; \begin{cases} g \underset{E}{\leftrightarrows} g' \quad \textit{eller} \\ \exists g_1, g_2 \in \mathcal{G}_{\Sigma^x}; c \in \mathcal{G}_\Sigma \mid g = c[g_1],\ g' = c[g_2],\ g_1 \simeq^x g_2 \end{cases}$$

Vi definerer så $\simeq^\alpha$ på $\mathcal{G}_\Sigma$ som

$$\simeq^\alpha = \Re^*$$

*Vi kaller $\simeq^\alpha$ den **initielle semantikken relativ til** $\simeq^x$ spesifisert av Σ og E, og vi kaller kvotientalgebraen $\mathcal{G}_\Sigma/\simeq^\alpha$ den **formelle initielle datatypen relativ til** $\simeq^x$ spesifisert av Σ og E. Vi kaller $\simeq^x$ **kjernen** i $\simeq^\alpha$.*

(Lesere med sans for subtiliteter vil nå ha merket at vi bruker α (alpha) og ω (omega) som superskript på initial- hhv. finalsemantikk. Videre vil (ikke nærmere spesifisert) kjernesemantikk generelt ha superskriptet x.)

Relasjonen $\simeq^\alpha$ i definisjon 2.9 er opplagt en kongruensrelasjon. Spesialtilfellet $\simeq^x = \overset{*}{\underset{E}{\leftrightarrows}}_{\mathcal{G}_{\Sigma^x}}$ i definisjon 2.9 gir basis-initialsemantikk som definert i definisjon 2.3 på side 24.

I implementasjonssammenheng, er en naturlig bakenforliggende tanke eller intensjon ved generell initialsemantisk spesifikasjon følgende:

Intensjon B

1. Kjernesemantikken er semantikk på generatortermer.
2. Ligningsmengden E gir semantikk til definerte funksjonssymboler.

En spesifikasjonsoppgave deles således opp i avgrensede deler. Som for vår generelle finalsemantikk, kan kjernesemantikken her være en finalsemantikk, eller igjen en initialsemantikk, eller en helt annen type semantikk. Dette siste er et hovedpoeng med både generell initialsemantikk og generell finalsemantikk. Vi holder fast ved konstruktiv funksjonsspesifikasjon for definerte funksjonssymboler, men har flere måter å spesifisere generatorsemantikk på (vi skal som sagt utvikle nok en måte i kapittel 3). Ofte kan spesifikasjon utføres ved basissemantikker. Men våre generelle semantikker åpner for spesifikasjon der dette ikke er mulig — et konkret eksempel får vi i kapittel 3. Vi kan og skal i de neste avsnittene utvikle viktige begreper for generell semantikk *uten å behøve å ta*

stilling til hvordan kjernesemantikken er spesifisert. Et annet poeng ved generell semantikk som er verdt å fremheve er den modulære oppbygging. Dette skal vi komme kort tilbake til om en stund. Merk at selv om basis-semantikker kan sees på som degenererte generelle semantikker, så kan basis-semantikker også «heves opp» og uttrykkes i form av generell semantikk. Slik gir generell semantikk en mulighet til å betrakte vanlige basis-semantikker på en modulær måte.

Før vi går igang med resten av diskusjonen, gir vi den initialsemantiske analoge observasjon til observasjon 2.3 på side 27:

Observasjon 2.6 *For en Σ, $\simeq^x$ på $\mathcal{G}_{\Sigma^x} \subseteq \mathcal{G}_\Sigma$ og $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$, la $\simeq^\alpha$ være initialsemantikken relativ til $\simeq^x$ spesifisert av Σ og E. Alle likheter i $\simeq^x$ er bevart i $\simeq^\alpha$, dvs.*

$$g \simeq^x g' \;\Rightarrow\; g \simeq^\alpha g'$$

2.3.6 Konsistens

Ved spesifisering ved basis-initialsemantikk, er det naturlig å spesifisere semantikk til generatorer ved ligninger mellom generatortermer. Men ved (2.11) på side 26 kan også semantikk av definerte funksjonssymboler gi semantikk til generatorer. I en viss forstand, kan vi da få uønsket semantikkspesifikasjon:

Eksempel 18 Betrakt generatorsignaturen

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0}, \\ \mathsf{succ}, \\ \mathsf{pred} \end{array} \right\}$$

La f være et definert funksjonssymbol. Betrakt ligningsmengden

$$E_\mathsf{f} = \left\{ \begin{array}{l} \mathsf{f(0) = 0}, \\ \mathsf{f(succ(x)) = succ(f(x))}, \\ \mathsf{f(pred(x)) = f(x)} \end{array} \right\}$$

Relativt til basis-initialsemantikken spesifisert av

$$E_{\mathcal{I}nt_c} = \left\{ \begin{array}{l} \mathsf{succ(pred(x)) = x}, \\ \mathsf{pred(succ(x)) = x} \end{array} \right\}$$

gir E_f initialsemantisk

$$\mathsf{f(succ(pred(0)))} \simeq^\alpha \mathsf{succ(0)} \quad \text{og} \quad \mathsf{f(0)} \simeq^\alpha \mathsf{0}$$

og dermed

$$\mathsf{succ(0)} \simeq^\alpha \mathsf{0}.$$

Vi har ikke basis-initialsemantisk $\mathsf{succ(0)} \overset{*}{\underset{E_{\mathcal{I}nt_c}}{\longleftrightarrow}} \mathsf{0}$. Ligningsmengden E_f gir således ikke bare semantikk til det definerte funksjonssymbol f, men også semantikk til generatorer.

Gitt at vi ønsker at generatorsemantikk skal være den basis-initielle gitt av $E_{\mathcal{I}nt_c}$, har vi her at E_f innfører uønsket semantikk på generatorene.
○

Definisjon 2.10 (se også [Gut77]) *La Σ^c være en signatur av generatorer. La E være en vilkårlig ligningsmengde og la $E^c \subseteq E$ være slik at $E^c = E \cap \mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$. E er **initielt konsistent** for Σ^c dersom $\overset{*}{\underset{E}{\longleftrightarrow}}\mathcal{G}_{\Sigma^c} = \overset{*}{\underset{E^c}{\longleftrightarrow}}\mathcal{G}_{\Sigma^c}$.*

Vi utvikler et generalisert konsistensbegrep for initialsemantikk:

Definisjon 2.11 *For en Σ, $\simeq^x$ på $\mathcal{G}_{\Sigma^x} \subseteq \mathcal{G}_\Sigma$ og $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$, la $\simeq^\alpha$ være initialsemantikken relativ til $\simeq^x$ spesifisert av Σ og E. E er* ***initielt konsistent relativt til*** *$\simeq^x$ dersom*

$$\simeq^x = \simeq^\alpha_{\mathcal{G}_{\Sigma^x}}$$

Definisjon 2.10 er spesialtilfellet av definisjon 2.11 for $\simeq^x$ en semantikk på generatortermer spesifisert algebraisk ved en ligningsmengde $E^c \subseteq E$.

Vi utvikler nå konsistensbegrep for final (pseudo)-semantikk. Det er naturlig å identifisere en mengde definerte funksjonssymboler som er tilsiktet alene å gi semantikk ved prinsipp (2.11) på side 26.

Definisjon 2.12 *For en Σ, la $\Sigma^c \subseteq \Sigma$ være generatorene i Σ. La $\Sigma^o \subseteq \Sigma$ være en signatur av definerte funksjonssymboler, og la $\simeq^x$ være en gitt semantikk på $\mathcal{G}_{\Sigma^x}$, for $\Sigma^x \subseteq \Sigma$. Betrakt pseudo-semantikken $\simeq^o$ på $\mathcal{G}_\Sigma$ slik at*

$$g \simeq^o g' \;\Leftrightarrow\; \begin{cases} g, g' \text{ er av samme type} \quad \text{og} \\ \neg\exists c \in \mathcal{G}_{\Sigma^o \cup \Sigma^c} \setminus \mathcal{G}_{\Sigma^c}; g_1, g_2 \in \mathcal{G}_{\Sigma^x} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'] \end{cases}$$

For den finale pseudo-semantikk $\simeq^\omega$ relativ til $\simeq^x$ bestemt av Σ og E, er E

- ***finalt konsistent mhp.*** *Σ^o* ***relativt til*** *$\simeq^x$, dersom $\simeq^o = \simeq^\omega$.*
- ***indre finalt konsistent (mhp.*** *Σ^o**) relativt til*** *$\simeq^x$, dersom $\simeq^o_{\mathcal{G}_{\Sigma^x}} = \simeq^\omega_{\mathcal{G}_{\Sigma^x}}$.*
- ***finalt kjernebevarende relativt til*** *$\simeq^x$, dersom $\simeq^\omega_{\mathcal{G}_{\Sigma^x}} = \simeq^x$.*

Signaturen Σ^o i definisjon 2.12 består av profiler til de definerte funksjonssymboler som er ment å gi semantikk ved prinsipp (2.11) på side 26. I den forstand er $\simeq^o$ den tilsiktede finale (pseudo-)semantikk ifølge Σ^o. Dersom $\simeq^o \neq \simeq^\omega$ finnes derfor funksjonssymboler i $\Sigma \setminus \Sigma^o$ hvis semantikk gitt av E bidrar til å spesifisere flere ulikheter ved prinsipp (2.11) enn det som er tilsiktet.

2.3.7 Sammenheng mellom initial- og finalsemantikk

Vi skal nå vise noen sammenhenger mellom begreper knyttet til initial- og finalsemantikk. Foruten å etablere resultater for diskusjonen videre, er hensikten å styrke intuisjonen på disse semantikkene, samt å vise at disse semantikkene, under konsistens, ivaretar den for implementasjon tilsiktede hensikt som uttrykt i intensjonene A side 30 og B side 31. Sistnevnte viser vi via noe vi skal kalle en *separabel* semantikk.

Vår reiserute skal være som følger: Vi skal vise ekvivalens under visse forutsetninger mellom

1. initial konsistens og final kjernebevaring
2. final kjernebevaring og indre final konsistens, for en degenerert form for finalsemantikk
3. indre final konsistens og final konsistens, for nok en degenerert form for finalsemantikk.

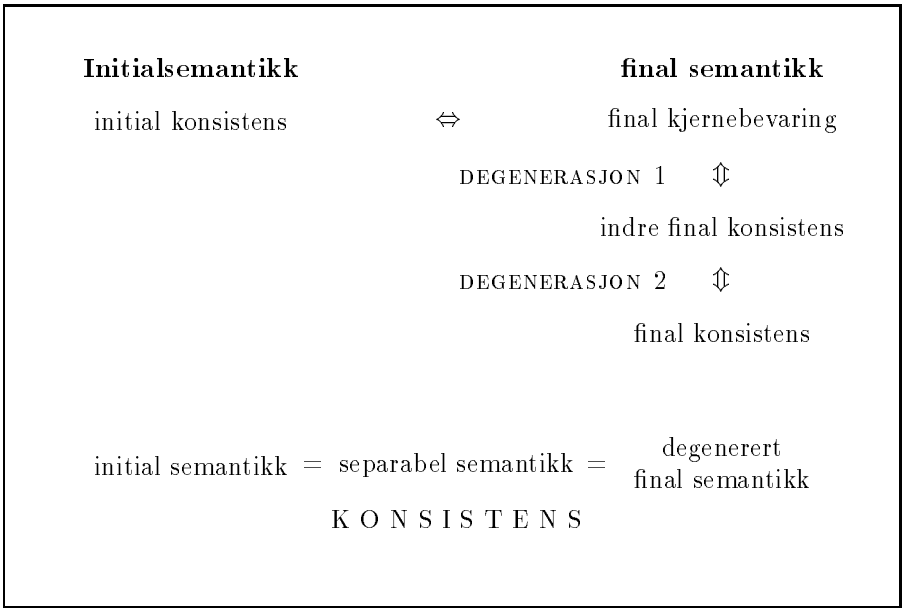
Deretter viser vi identitet under visse forutsetninger mellom

4. initial semantikk og den siste formen for degenerert finalsemantikk.

Sistnevnte skal vi gjøre indirekte ved å vise identitet under konsistens mellom

5. initial semantikk og separabel semantikk og mellom degenerert finalsemantikk og separabel semantikk.

Sammenhengene er også vist i figur 2.1.



Figur 2.1: Sammenhenger mellom begreper knyttet til initial- og final-semantikk.

1. Kjernebevaring

Final kjernebevaring betyr konservering av kjerne-semantikken $\simeq^x$ i den finale pseudo-semantikken $\simeq^\omega$. Initialkonsistens betyr på sin side konservering av kjerne-semantikken $\simeq^x$ i initialsemantikken $\simeq^\alpha$. Intuisjonen på definisjonene av initiell- og finalsemantikk og deres konsistensbegreper, vil at final kjernebevaring og initialkonsistens skal være ekvivalente begreper. Denne intuisjon viser seg å stemme; ihvertfall dersom den aktuelle ligningsmengde er tilstrekkelig komplett (`TK`) eller fullstendig ukomplett mhp. $\mathcal{G}_{\Sigma^x}$.

Merk at antagelsen `KONSERV` (side 28) allerede utelukker en direkte kilde til ikke kjernebevaring. Denne kilde er forskjellig fra den representert av prinsipp (2.11) på side 26:

Eksempel 19 Betrakt generatorsignaturen

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0}, \\ \mathsf{succ}, \\ \mathsf{pred} \end{array} \right\}$$

La f være et definert funksjonssymbol. Betrakt ligningsmengden

$$E'_{\mathsf{f}} = \left\{ \begin{array}{l} \mathsf{f(0) = 0}, \\ \mathsf{f(0) = succ(0)} \end{array} \right\}$$

eller for den saks skyld

$$E_x = \{\ \mathsf{0 = succ(0)}\ \}$$

Relativt til basis-initialsemantikken gitt av

$$E_{\mathcal{I}nt_c} = \left\{ \begin{array}{l} \mathsf{succ(pred(x)) = x}, \\ \mathsf{pred(succ(x)) = x} \end{array} \right\}$$

gir både E'_{f} og E_x initialsemantisk

$\mathsf{succ}(0) \simeq^{\alpha} 0$.

Slike direkte opphav til initial inkonsistens som E'_{f} og E_x representerer, utelukkes av KONSERV.
○

Før vi viser ekvivalens av final kjernebevaring og initialkonsistens, bemerker vi for ordens skyld: Vi kan ikke vise ekvivalens av final kjernebevaring og initialkonsistens ved å vise at $\simeq^{\omega}_{\mathcal{G}_{\Sigma^x}} = \simeq^{\alpha}_{\mathcal{G}_{\Sigma^x}}$. Dette fordi $\simeq^{\omega}_{\mathcal{G}_{\Sigma^x}} \neq \simeq^{\alpha}_{\mathcal{G}_{\Sigma^x}}$ med mindre vi har både final kjernebevaring og initialkonsistens. Sistnevnte faktum er motstykket til observasjonene 2.6 og 2.3: Ikke final kjernebevaring vil si

$$\simeq^{\omega}_{\mathcal{G}_{\Sigma^x}} \subset \simeq^{x}$$

mens initiell inkonsistens betyr

$$\simeq^{x} \subset \simeq^{\alpha}_{\mathcal{G}_{\Sigma^x}}$$

Således er initiell inkonsistens og ikke final kjernebevaring *inversjoner* av hverandre: Initiell inkonsistens «kollapser» kjernen; dvs. tilfører kjernen flere elementer (elementene i kjernen er tupler som representerer identiteter), mens ikke final kjernebevaring «oppløser» kjernen; dvs. fjerner elementer fra kjernen.

Eksempel 20 Betrakt E_{f} og $E_{\mathcal{I}nt_c}$ fra eksempel 18. Relativt til basis-initialsemantikken $\simeq$ gitt av $E_{\mathcal{I}nt_c}$, er E_{f} både initielt inkonsistent og ikke finalt kjernebevarende. For $\simeq^{\alpha}$ og $\simeq^{\omega}$ — initial- og finalsemantikkene relativ til $\simeq$ (bestemt av $\Sigma \cup \{\mathsf{f}\}$ og E_{f}), gir dette seg bl.a. tilkjenne på den ene side i at

$$\mathsf{succ}(0) \simeq^{\alpha} 0$$

og på den annen side i at

$$\mathsf{succ}(\mathsf{pred}(0)) \not\simeq^{\omega} 0$$

Relasjonene $\simeq^{\alpha}$ og $\simeq^{\omega}$ er ikke identiske: Vi har nemlig $\mathsf{succ}(0) \not\simeq^{\omega} 0$; og dessuten også $\mathsf{succ}(\mathsf{pred}(0)) \simeq^{\alpha} 0$.
○

Vi viser nå ekvivalens av final kjernebevaring og initialkonsistens under TK på side 28. Vi trenger her ikke antagelsen KONSERV.

Sats 2.7 *Anta* TK. *For Σ^x, $\simeq^x$ og E i definisjonene 2.11 og 2.12 har vi at E er finalt kjernebevarende relativt til $\simeq^x$ hvis og bare hvis E er initielt konsistent relativt til $\simeq^x$.*

Bevis: La $\simeq^{\alpha}$ være den initielle semantikken relativ til $\simeq^x$.

Anta at E er ikke finalt kjernebevarende relativt til $\simeq^x$. Ved (2.10) side 24 og ved observasjon 2.3 side 27 må det da finnes $c \in \mathcal{G}_{\Sigma}$ og $g_x, g'_x, g_1, g_2 \in \mathcal{G}_{\Sigma^x}$ slik at

$$c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x] \quad \text{men} \quad g_x \simeq^x g'_x$$

Siden $g_x \simeq^x g'_x$ har vi ved observasjon 2.6 $g_x \simeq^{\alpha} g'_x$, og da har vi $c[g_x] \simeq^{\alpha} c[g'_x]$. Siden $c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} g_1$ og $g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x]$, får vi da $g_1 \simeq^{\alpha} g_2$. Men $g_1 \not\simeq^x g_2$, så E kan ikke være initielt konsistent relativt til $\simeq^x$.

For den motsatte implikasjon, anta først at KONSERV ikke holder. Da finnes $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$ slik at

$$g_x \overset{*}{\underset{E}{\leftrightarrow}} g'_x \quad \text{men} \quad g_x \not\simeq^x g'_x$$

Men da har vi $g_x \not\simeq^x g'_x \overset{*}{\underset{E}{\leftrightarrow}} g_x$, og E er trivielt ikke finalt kjernebevarende.

Anta derfor KONSERV. Anta E ikke er initielt konsistent relativt til $\simeq^x$. Ved observasjon 2.6 finnes da $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$ slik at

$$g_x \simeq^\alpha g'_x \quad \text{men} \quad g_x \not\simeq^x g'_x$$

Vi skriver $g \underset{x}{\leftrightarrow} g'$ dersom $g \simeq^\alpha g'$, og ikke $g \overset{*}{\underset{E}{\leftrightarrow}} g'$. M.a.o.

$$g \underset{x}{\leftrightarrow} g' \;\Leftrightarrow\; \begin{cases} g \not\overset{*}{\underset{E}{\leftrightarrow}} g' \quad \text{og} \\ \exists c \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^x} \mid g = c[g_1],\ g' = c[g_2],\ g_1 \simeq^x g_2 \end{cases}$$

Betrakt en vilkårlig $\simeq^\alpha$-utledning $\langle g_x, \ldots, g'_x \rangle$ i $\mathcal{G}_\Sigma$. Ved KONSERV må $g_x \not\overset{*}{\underset{E}{\leftrightarrow}} g'_x$, så det må finnes minst ett $\underset{x}{\leftrightarrow}$-steg i utledningen. Og siden $g_x \not\simeq^x g'_x$, må det også finnes minst ett $\underset{E}{\leftrightarrow}$-steg i utledningen. Vi kan anta at dette steget er slik at $g \underset{E}{\leftrightarrow} g'$ der minst en av g, g' ikke er i $\mathcal{G}_{\Sigma^x}$; ellers var dette steget ved KONSERV et $\underset{x}{\leftrightarrow}$-steg. Det finnes altså en term $g \notin \mathcal{G}_{\Sigma^x}$ i utledningen. Men da må det finnes *to* $\underset{E}{\leftrightarrow}$-steg; siden et $\underset{x}{\leftrightarrow}$-steg ikke kan være mellom en term som *ikke* er i $\mathcal{G}_{\Sigma^x}$ og en term som *er* i $\mathcal{G}_{\Sigma^x}$ (og $g_x \neq g'_x$, siden $\simeq^\alpha$ er refleksiv). Begge disse to $\underset{E}{\leftrightarrow}$-steg må være mellom en term i $\mathcal{G}_{\Sigma^x}$ og en term *ikke* i $\mathcal{G}_{\Sigma^x}$. Utledningen må altså ha formen:

$$\begin{aligned} g_x \overset{*}{\underset{x}{\leftrightarrow}} g_{x0} \overset{+}{\underset{E}{\leftrightarrow}} c_1[g_1] \overset{*}{\underset{x}{\leftrightarrow}} c_1[g'_1] \overset{*}{\underset{E}{\leftrightarrow}} \cdots & \\ \vdots & \\ \cdots \overset{*}{\underset{E}{\leftrightarrow}} c_k[g_k] \overset{+}{\underset{x}{\leftrightarrow}} c_k[g'_k] \overset{*}{\underset{E}{\leftrightarrow}} c_{k+1}[g_{k+1}] \overset{+}{\underset{x}{\leftrightarrow}} c_{k+1}[g'_{k+1}] \overset{*}{\underset{E}{\leftrightarrow}} \cdots & \\ \vdots & \\ \cdots \overset{*}{\underset{E}{\leftrightarrow}} c_n[g_n] \overset{*}{\underset{x}{\leftrightarrow}} c_n[g'_n] \overset{+}{\underset{E}{\leftrightarrow}} g_{xn} \overset{*}{\underset{x}{\leftrightarrow}} g'_x & \end{aligned}$$

for $g_{x0}, g_{xn}, g_i, g'_i \in \mathcal{G}_{\Sigma^x}$ og $c_i \in \mathcal{G}_\Sigma$; $1 \leq i \leq n$. Ved TK har vi imidlertid

$$c_i[g'_i] \overset{*}{\underset{E}{\leftrightarrow}} g_{xi} \overset{*}{\underset{E}{\leftrightarrow}} c_{i+1}[g_{i+1}]$$

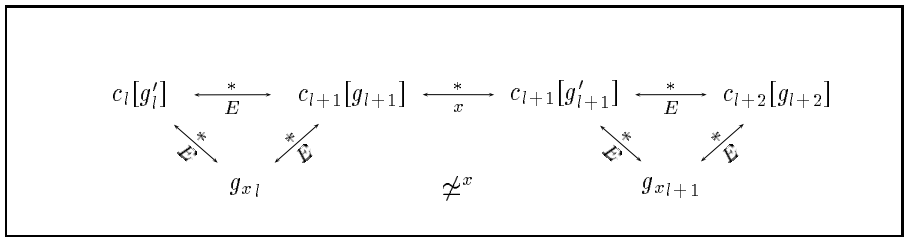
for noen $g_{xi} \in \mathcal{G}_{\Sigma^x}$; $1 \leq i \leq n$. (Dette har vi ved at at hver komponent i $\simeq^\alpha$-utledningen må være av samme type som g_x og g'_x.) Nå kan ikke $g_{x0} \simeq^x \cdots \simeq^x g_{xk} \simeq^x \cdots \simeq^x g_{xn}$, da det ville innebære $g_x \simeq^x g'_x$. Altså må det finnes en $0 \leq l < n$ slik at

$$g_{xl} \not\simeq^x g_{xl+1}$$

(Se figur 2.2.) Men da har vi

$$c_{l+1}[g_{l+1}] \overset{*}{\underset{E}{\leftrightarrow}} g_{xl} \not\simeq^x g_{xl+1} \overset{*}{\underset{E}{\leftrightarrow}} c_{l+1}[g'_{l+1}]$$

Siden $g_{l+1} \simeq^x g'_{l+1}$, er derfor E ikke finalt kjernebevarende relativt til $\simeq^x$.
□

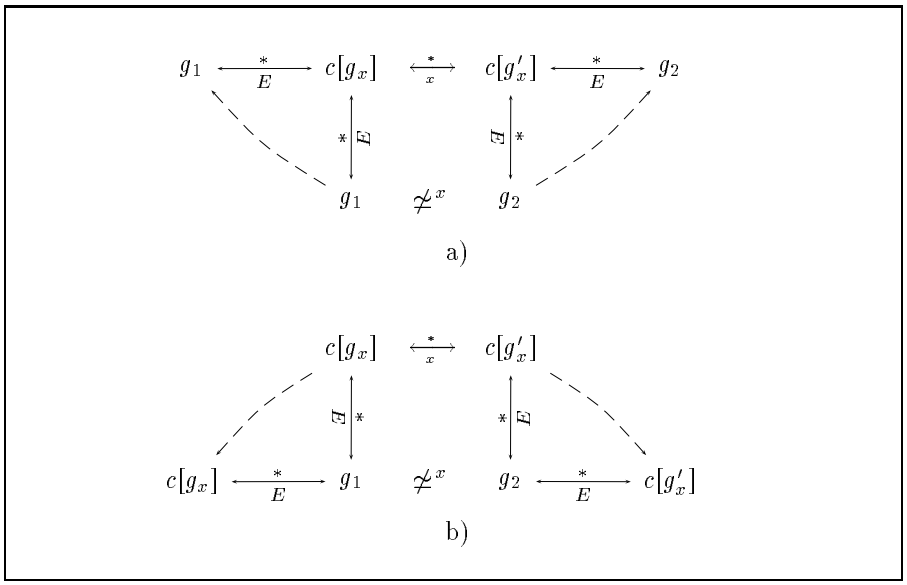


Figur 2.2: Del-bevisskisse for sats 2.7. Ved TK har vi $c_i[g'_i] \overset{*}{\underset{E}{\leftrightarrow}} g_{x_i} \overset{*}{\underset{E}{\leftrightarrow}} c_{i+1}[g_{i+1}]$ for noen $g_{x_i} \in \mathcal{G}_{\Sigma^x}$; $1 \leq i \leq n$. Det må finnes en $0 \leq l < n$ slik at $g_{x_l} \not\simeq^x g_{x_{l+1}}$.

Kommentar:

> Det er i flere konkrete tilfeller mulig å vise at dersom E ikke er initielt konsistent relativt til $\simeq^x$, så vil det for to $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$ slik at $g_x \simeq^\alpha g'_x$ men $g_x \not\simeq^x g'_x$, alltid finnes en $\simeq^\alpha$-utledning $\langle g_x, \ldots, g'_x \rangle$ på formen beskrevet i beviset for sats 2.7, men slik at $n = 1$. Dette henger sammen med den generatorinduktive måten å spesifisere konstruktive funksjons-spesifikasjoner på. Ligninger i slike spesifikasjoner vil ofte ha formen $c[f_c(x)] = c'[x]$, der f_c er en generator. Dette kan brukes til å redusere en evt. lang utledningskjede (stor n) ned til $n = 1$.

Med bakgrunn i sats 2.7, er sammenhengen mellom final kjernebevaring og initiell konsistens under `KONSERV` og for tilfellet '$n = 1$', illustrert i figur 2.3.



Figur 2.3: Inversjon. Inkonsistens sett på to måter. Final kjernebevaring og initiell konsistens er under `KONSERV` to sider av samme sak. I a) ser vi inkonsistens tilkjennegitt som initial inkonsistens, i og med at $g_1 \not\simeq^x g_2$, men $g_1 \simeq^\alpha g_2$. I b) sees inkonsistens manifestert i ikke final kjernebevaring, idet $g_x \not\simeq^\omega g_x$, men $g_x \simeq^x g_x$.

Sats 2.7 hevder ekvivalens mellom final kjernebevaring og initialkonsistens, forutsatt `TK`. Hvis vi har det rake motsatte av `TK`; nemlig fullstendig Σ-ukompletthet mhp. $\mathcal{G}_{\Sigma^x}$, kan vi også slå fast denne ekvivalens. Men dette er fordi fullstendig Σ-ukompletthet garanterer kjernebevaring:

Sats 2.8 *Anta* `KONSERV`. *For Σ^x, $\simeq^x$ og E i definisjonene 2.11 og 2.12, anta E fullstendig Σ-ukomplett mhp. $\mathcal{G}_{\Sigma^x}$. Da har vi at E er finalt kjernebevarende relativt til $\simeq^x$, og at E er initielt konsistent relativt til $\simeq^x$.*

Bevis: Anta $g_x \simeq^\alpha g'_x$, for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$. Siden E er fullstendig Σ-ukomplett mhp. $\mathcal{G}_{\Sigma^x}$, må enhver $\simeq^\alpha$-utledning $\langle g_x, \ldots, g'_x \rangle$ være en utledning i $\mathcal{G}_{\Sigma^x}$. Ved `KONSERV` har vi da for enhver g_i, g_{i+1} i en slik utledning slik at $g_i \rightleftarrows_E g_{i+1}$, at $g_i \simeq^x g_{i+1}$. Følgelig har vi $g_x \simeq^x g'_x$, så E er initielt konsistent relativt til $\simeq^x$.

Anta at E er ikke finalt kjernebevarende relativt til $\simeq^x$. Da fins $c \in \mathcal{G}_\Sigma$ og $g_1, g_2 \in \mathcal{G}_{\Sigma^x}$ slik at

$$c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x] \quad \text{til tross for at} \quad g_x \simeq^x g'_x$$

Men siden E er fullstendig Σ-ukomplett mhp. $\mathcal{G}_{\Sigma^x}$, må $c \in \mathcal{G}_{\Sigma^x}$. Ved `KONSERV` får vi da

$$c[g_x] \simeq^x g_1 \not\simeq^x g_2 \simeq^x c[g'_x]$$

Ved kongruens har vi $c[g_x] \simeq^x c[g'_x]$, og dermed $g_1 \simeq^x g_2$, som gir en motsigelse. Altså er E finalt kjernebevarende relativt til $\simeq^x$.
□

Final kjernebevaring er triviell i en semantikk der kjernen er fri:

Observasjon 2.9 *Dersom en final pseudo-semantikk er en kongruensrelasjon og dens kjerne $\simeq^x$ er fri, har vi, ved observasjon 2.3 side 27, final kjernebevaring relativt til $\simeq^x$.*

Vi kan da merke oss følgende interessante korollar til satsene 2.7 og 2.8:

Korollar 2.10 *Anta* `TK` *eller fullstendig Σ-ukompletthet mhp. $\mathcal{G}_{\Sigma^x}$. Dersom en final pseudo-semantikk er en kongruensrelasjon og dens kjerne $\simeq^x$ er fri, har vi initialkonsistens relativt til $\simeq^x$ (for den korresponderende initialsemantikk).*

Kommentar:

> Mange av de resultater vi etablerer som antar (full) tilstrekkelig kompletthet og evt. fullstendig ukompletthet kan også formuleres som partielle resultater i korrelasjon til antagelser om partiell tilstrekkelig kompletthet. Å gjennomføre diskusjonen med slike partielle resultater, ville forkludre fremstillingen veldig. Motivasjon for å se på tilfeller mellom ytterpunktene (full) tilstrekkelig kompletthet og fullstendig ukompletthet er på dette tidspunkt heller ikke til stede. Derfor velger vi en linje som forenkler bildet, ved å anta (full) tilstrekkelig kompletthet eller evt. fullstendig ukompletthet. Det kan dog tenkes at disse antagelser ofte er mer enn nødvendig sterke. Med dette sagt, skal vi i neste kapittel finne motivasjon for partielle resultater om konsistens, og i avsnitt 3.7 skal vi vise noen slike.

2. Første degenerasjon. Final kjernebevaring og indre final konsistens

Betrakt nå spesialtilfellet $\Sigma^o = \emptyset$ for Σ^o i definisjon 2.12 (side 33). Dette betyr at vi ikke ønsker at semantikk gis ved prinsipp (2.11) på side 26. Dette kjennetegner jo også initialkonsistens. Men nå er ikke final- og initialkonsistens ekvivalente generelt; selv ikke under antagelsen $\Sigma^o = \emptyset$. Initialkonsistens sier kun noe om kjernesemantikken $\simeq^x$. Dette er jo ikke tilfellet for finalkonsistens:

Eksempel 21 Betrakt eksempel 16 på side 29. Her er ligningsmengden E ikke finalt konsistent mhp. $\Sigma^o = \emptyset$ relativt til initialsemantikken i $\mathcal{G}_{\Sigma^{xc}}$, pga. finalsemantikken gitt til generatortermer i $\mathcal{G}_{\Sigma^{yc}}$. Men E er jo initielt konsistent relativt til initialsemantikken i $\mathcal{G}_{\Sigma^{xc}}$.
○

Og på den annen side sier finalkonsistens kun noe om semantikk gitt ved prinsipp (2.11). Det er lett å finne ligningsmengder som er finalt konsistente men initielt inkonsistente.

Under antagelsen $\Sigma^o = \emptyset$ bør imidlertid *indre* final konsistens og initiell konsistens være nært knyttet. Vi viser dette ved å stadfeste en knytning mellom indre final konsistens og final kjernebevaring. Knytningen videre mellom final kjernebevaring og initiell konsistens viste vi i forrige delavsnitt. Betrakt først følgende ikke overraskende lemma:

Lemma 2.11 *Anta* `KONSERV`*. For signaturer, relasjoner og E som i definisjon 2.12, anta* $\Sigma^o = \emptyset$*. Da er* $\simeq^o_{\mathcal{G}_{\Sigma^x}} = \simeq^x$*.*

Bevis: Vi har trivielt $\simeq^o_{\mathcal{G}_{\Sigma^x}} \subseteq \simeq^x$. Anta derfor at det finnes $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$ slik at $g_x \simeq^x g'_x$, men $g_x \not\simeq^o g'_x$. Siden g_x, g'_x må være av samme type, finnes altså $c \in \mathcal{G}_\Sigma$ og $g_1, g_2 \in \mathcal{G}_{\Sigma^x}$ slik at

$$c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x]$$

Siden $\Sigma^o = \emptyset$, reduserer dette seg til

$$g_x \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} g'_x$$

Videre gir `KONSERV` $g_x \simeq^x g_1$ og $g_2 \simeq^x g'_x$. Dette gir da ialt $g_1 \simeq^x g_2$ som er absurd.
□

(Antagelsen `KONSERV` var her nødvendig for å sikre refleksivitet av $\simeq^o_{\mathcal{G}_{\Sigma^x}}$.) Forbindelsen mellom indre final konsistens og final kjernebevaring følger umiddelbart:

Sats 2.12 *Anta* `KONSERV`*. For signaturer, relasjoner og E som i definisjon 2.12, anta* $\Sigma^o = \emptyset$*. Da er E indre finalt konsistent hvis og bare hvis E er finalt kjernebevarende.*

Bevis: Anta $\simeq^o_{\mathcal{G}_{\Sigma^x}} = \simeq^\omega_{\mathcal{G}_{\Sigma^x}}$. Ved lemma 2.11 har vi $\simeq^o_{\mathcal{G}_{\Sigma^x}} = \simeq^x$, så $\simeq^\omega_{\mathcal{G}_{\Sigma^x}} = \simeq^x$ følger umiddelbart.

Anta $\simeq^\omega_{\mathcal{G}_{\Sigma^x}} = \simeq^x$. Vi får umiddelbart ved lemma 2.11 $\simeq^o_{\mathcal{G}_{\Sigma^x}} = \simeq^\omega_{\mathcal{G}_{\Sigma^x}}$.
□

Under antagelsene `KONSERV`, `TK` og $\Sigma^o = \emptyset$, følger det således gjennom sats 2.12 og sats 2.7, at indre final konsistens og initiell konsistens er ekvivalente begreper. (Dette følger altså gjennom forbindelser mellom initiell konsistens og final kjernebevaring og mellom final kjernebevaring og indre final konsistens. Se figur 2.1 side 34. Vi kunne vise forbindelsen mellom indre final konsistens og initiell konsistens direkte. Men merk at en antagelse (lignende `KONSERV`) er nødvendig for å sikre refleksivitet av den finale pseudo-semantikken.)

3. Annen degenerasjon. Indre final konsistens og final konsistens

Betrakt nå følgende kriterium:

`DEGEN:` Alle generatorene i Σ fins i Σ^x, for Σ^x og Σ i definisjonene 2.9 side 31 og 2.7 side 27.

Observasjon 2.13 *For* Σ^x *og* Σ *i definisjonene 2.9 og 2.7, innebærer kriteriet* `DEGEN` *at alle termer i* $\mathcal{G}_\Sigma$ *er av typer som termene i* $\mathcal{G}_{\Sigma^x}$ *er av.*

I tilknytning til finalsemantikk utelukker `DEGEN` semantikkgiving til generatorer ad prinsipp (2.11) side 26 uten inkonsistens. Dette fjerner oss fra en av hovedintensjonene med finalsemantikk, nemlig å beskrive situasjoner hvor generatorer

(av en type) gis semantikk utifra ankerfester i andre generatortermer (av en annen type). Se punkt 2 på side 30.

På den annen side skal vi se at under **DEGEN**, kan finalsemantikk gi et annet uttrykk for initialsemantikk. Vi går veien om en forbindelse mellom indre final konsistens og (full) final konsistens:

Under antagelsen $\Sigma^o = \emptyset$ og **DEGEN**, er det ikke unaturlig å forestille seg at indre final konsistens er ekvivalent med (full) final konsistens. Følgende eksempel illustrerer imidlertid at dette ikke er tilfellet:

Eksempel 22 La $\simeq^x$ være den frie semantikk på $\mathcal{G}_{\{a,b\}}$. Betrakt

$$E = \left\{ \begin{array}{l} h(f(a)) = a, \\ h(f(b)) = b \end{array} \right\}$$

For den finale pseudo-semantikk $\simeq^\omega$ relativ til $\simeq^x$ bestemt av $\{a, b, f, h\}$ og E, er **KONSERV** oppfylt, så $\simeq^\omega$ er refleksiv. Videre er det lett å sjekke at $\simeq^\omega$ er transitiv. Ved sats 2.4 er da $\simeq^\omega$ en kongruensrelasjon. Ved observasjon 2.9 side 38 er da E finalt kjernebevarende relativt til $\simeq^x$. Ved sats 2.12 er så E indre finalt konsistent relativt til $\simeq^x$. Derimot er E ikke finalt konsistent mhp. $\Sigma^o = \emptyset$ relativt til $\simeq^x$. Vi har nemlig $f(a) \simeq^o f(b)$, men ikke $f(a) \simeq^\omega f(b)$, siden

$$h(f(a)) \overset{*}{\underset{E}{\leftrightarrow}} a \not\simeq^x b \overset{*}{\underset{E}{\leftrightarrow}} h(f(b))$$

○

Ligningsmengden E i eksempel 22 tilfredstiller ikke **TK**. Imidlertid har vi:

Sats 2.14 *Under antagelsene $\Sigma^o = \emptyset$,* **DEGEN**, *samt* **TK**, *er final konsistens ekvivalent med indre final konsistens.*

Bevis: La signaturer, semantikker og E være som i definisjon 2.12.

For den ikke-trivielle implikasjonen, anta ikke final konsistens for en ligningsmengde E mhp. Σ^o relativt til en $\simeq^x$. Vi har da opplagt $\simeq^\omega \subset \simeq^o$ (pr. def.). Så anta $g \not\simeq^\omega g'$, men $g \simeq^o g'$ for noen $g, g' \in \mathcal{G}_\Sigma$. Da finnes altså, siden g, g' må være av samme type, $c \in \mathcal{G}_\Sigma$ og $g_1, g_2 \in \mathcal{G}_{\Sigma^x}$ slik at

$$c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

Ved **DEGEN** med observasjon 2.13, og **TK** har vi $g \overset{*}{\underset{E}{\leftrightarrow}} g_x$ og $g' \overset{*}{\underset{E}{\leftrightarrow}} g'_x$ for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$. Vi får da

$$c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'] \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x]$$

Dette gir $g_x \not\simeq^\omega g'_x$. Imidlertid må vi ha $g_x \simeq^x g'_x$, ellers ville $g \not\simeq^o g'$ ved

$$g \overset{*}{\underset{E}{\leftrightarrow}} g_x \not\simeq^x g'_x \overset{*}{\underset{E}{\leftrightarrow}} g'$$

Altså er ikke E finalt kjernebevarende relativt til $\simeq^x$, og ved sats 2.12 har vi da at E ikke er indre finalt konsistent.

□

4./5. Møte mellom initial- final- og separabel semantikk

For degenerert finalsemantikk ifølge $\Sigma^o = \emptyset$ og **DEGEN**, følger det under **KONSERV** og **TK**, nå at initial- og finalkonsistens er ekvivalente begreper. Ikke overraskende kan vi gjøre følgende oppsummering:

Proposisjon 2.15 *For signaturer, semantikker og ligningsmengde som i definisjonene 2.11 og 2.12, har vi under antagelsene $\Sigma^o = \emptyset$ og* **DEGEN**, *samt* **KONSERV** *og* **TK**, *ved konsistens:*

$$\simeq^\alpha = \simeq^\omega$$

Vi kan vise proposisjonen direkte. Imidlertid ønsker vi å etablere samt gjøre et poeng av at initialsemantikk og degenerert finalsemantikk *under konsistens* ivaretar intensjonene A side 30 og B side 31. Merk at for den her degenererte finalsemantikk forsvinner punkt 2 og typene $T_1^y, \ldots, T_m^y$ på side 30.

Intensjonene A og B forfekter (bl.a.) en modulær sammensetting av to komponenter; nemlig semantikk for generatorer og semantikk for definerte funksjonssymboler. I kortene ligger også et ønske om at den resulterende sammensetningen ikke skal gi annen semantikk til generatorene og de definerte funksjonssymboler enn den som spesifiseres (hver for seg) av komponentene. Sistnevnte har vi jo sett at ikke nødvendigvis er tilfellet, i og med at semantikk for definerte funksjonssymboler kan gi (ytterligere) semantikk til generatorer og altså gi det vi kaller inkonsistens. Tilfellet inkonsistens viser at en modulær sammensetting av semantikker kan gi semantikk som på et vis er mer enn summen av sine deler. Det vi altså nå skal vise er at under konsistens vil modulære sammensettinger som forfektet i intensjoner A og B, ikke gi annen semantikk enn den som spesifiseres av sine komponenter. Vi skal gå veien om en relasjon som vi viser har egenskapen å være en «ren» sum av sine deler og som vi derfor skal kalle ***separabel***. Denne relasjonen skal ved flere anledninger vise seg å spille en grunnleggende rolle.

Her kommer den: Betrakt relasjonen

$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* \quad \text{[den refleksiv-transitive tillukning av unionen]} \tag{2.13}$$

Vi skal vise at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$ gir nøyaktig den semantikk som spesifiseres av sine komponenter $\simeq^x$ og $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma}$. Følgende lemma sier at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$ under `KONSERV` alltid er «kjernebevarende».

Sats 2.16 *For en semantikk $\simeq^x$ på $\mathcal{G}_{\Sigma^x}$ og en ligningsmengde E, har vi under* `KONSERV`

$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*_{\mathcal{G}_{\Sigma^x}} = \simeq^x$$

Bevis: Anta tvert imot at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*_{\mathcal{G}_{\Sigma^x}} \neq \simeq^x$. Siden vi opplagt har $\simeq^x \subseteq (\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*_{\mathcal{G}_{\Sigma^x}}$, må det da finnes $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$ slik at $g_x \not\simeq^x g'_x$, men $g_x(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g'_x$. Men da må vi ha

$$g_x \simeq^x g_1 \overset{*}{\underset{E}{\leftrightarrow}} g'_1 \simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} g'_2 \simeq^x \cdots g_i \overset{*}{\underset{E}{\leftrightarrow}} g'_i \cdots \simeq^x g_n \overset{*}{\underset{E}{\leftrightarrow}} g'_n \simeq^x g'_x$$

for noen $g_i, g'_i \in \mathcal{G}_{\Sigma^x}; 1 \leq i \leq n$. Ved `KONSERV` må $g_i \simeq^x g'_i$ for hver $1 \leq i \leq n$. Da får vi $g_x \simeq^x g'_x$, som er en motsigelse, og satsen følger.
□

Vi viser så at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$ bevarer semantikk gitt ved $\overset{*}{\underset{E}{\leftrightarrow}}$. Først viser vi at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$ i en viss forstand er komplett for $\overset{*}{\underset{E}{\leftrightarrow}}$:

Sats 2.17 *For signaturer, semantikker og ligningsmengde som i definisjonene 2.9 og 2.7, har vi for $g \in \mathcal{G}_\Sigma \setminus \mathcal{G}_{\Sigma^x}$ og $g_x \in \mathcal{G}_{\Sigma^x}$*

$$g \overset{*}{\underset{E}{\leftrightarrow}} g' \;\Rightarrow\; g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g'$$

Bevis: Satsen holder trivielt siden $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma} \subseteq (\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$.
□

Vi viser så sunnhet av $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$ med hensyn til en konstruktiv funksjonsspesifikasjon E:

Sats 2.18 *For signaturer, semantikker og ligningsmengde som i definisjonene 2.9 og 2.7, har vi under* `TK` *for vilkårlige* $g \in \mathcal{G}_\Sigma$ *og* $g_x \in \mathcal{G}_{\Sigma^x}$

$$g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g_x \;\Rightarrow\; g \overset{*}{\underset{E}{\leftrightarrow}} g'_x$$

for en $g'_x \in \mathcal{G}_{\Sigma^x}$ *slik at* $g_x \simeq^x g'_x$.

Bevis: Anta $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g_x$ for vilkårlige $g \in \mathcal{G}_\Sigma, g_x \in \mathcal{G}_{\Sigma^x}$. Observer at g må være av samme type som g_x. Ved `TK` har vi da $g \overset{*}{\underset{E}{\leftrightarrow}} g'_x$ for en $g'_x \in \mathcal{G}_{\Sigma^x}$. Da får vi

$$g_x(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g \overset{*}{\underset{E}{\leftrightarrow}} g'_x$$

m.a.o. $g_x(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g'_x$. Men ved sats 2.16 har vi da $g_x \simeq^x g'_x$.
□

La $\mathcal{T}ype^x$ være mengden av typer som termer i $\mathcal{G}_{\Sigma^x}$ er av. Sats 2.18 sier i en viss forstand at enhver konstruktiv funksjonsspesifikasjon i $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$ i $\mathcal{T}ype^x$-typer finnes i E. Men med mindre vi antar `DEGEN`, kan det også finnes annen (konstruktiv) funksjonsbeskrivelse i E som ikke relaterer seg til typer i $\mathcal{T}ype^x$. Følgende sats sier at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$ er sunn også for slik funksjonsbeskrivelse, samt i tilfellet E fullstendig Σ-ukomplett mhp. $\mathcal{G}_{\Sigma^x}$ (beviset av satsen gir her innsikt):

Sats 2.19 *For signaturer, semantikker og ligningsmengde som i definisjonene 2.9 og 2.7, har vi for* E *fullstendig* Σ*-ukomplett mhp.* $\mathcal{G}_{\Sigma^x}$*, for vilkårlige* $g, g' \in \mathcal{G}_\Sigma$

$$g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g' \;\Rightarrow\; g \overset{*}{\underset{E}{\leftrightarrow}} g'$$

Bevis: Anta $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g'$ for vilkårlige $g, g' \in \mathcal{G}_\Sigma$. Observer at g må være av samme type som g', og at enhver komponent i en vilkårlig $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$-utledning $\langle g, \ldots, g' \rangle$ er av samme type som g, g'.

Observer også at ethvert $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$-utledningssteg er på formen $g_i = g_x \simeq^x g'_x = g_{i+1}$, for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$; eller på formen $g_i \overset{}{\underset{E}{\leftrightarrow}} g_{i+1}$ (for $g_i, g_{i+1} \in \mathcal{G}_\Sigma$).

La $\mathcal{T}ype^x$ være mengden av typer som termer i $\mathcal{G}_{\Sigma^x}$ er av. Anta g, g' ikke er av en type i $\mathcal{T}ype^x$. Men da kan vi ikke ha $g_i \simeq^x g_{i+1}$ for noen komponenter g_i, g_{i+1} i noen $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$-utledning $\langle g, \ldots, g' \rangle$. Følgelig har vi $g \overset{*}{\underset{E}{\leftrightarrow}} g'$, og satsen følger.

Anta så at g, g' er av en type i $\mathcal{T}ype^x$. Dersom en av g, g' er i $\mathcal{G}_{\Sigma^x}$, følger satsen fra sats 2.18 over. Anta derfor $g, g' \in \mathcal{G}_\Sigma \setminus \mathcal{G}_{\Sigma^x}$. Siden E er fullstendig Σ-ukomplett mhp. $\mathcal{G}_{\Sigma^x}$, kan vi heller ikke nå ha $g_i \simeq^x g_{i+1}$ for noen komponenter g_i, g_{i+1} i noen $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$-utledning $\langle g, \ldots, g' \rangle$. Følgelig har vi igjen $g \overset{*}{\underset{E}{\leftrightarrow}} g'$.
□

Satsene 2.16, 2.17, 2.18 og 2.19 uttrykker tilsammen at $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$ gir nøyaktig den semantikk som spesifiseres av sine komponenter $\simeq^x$ og $\overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma$ (separabel semantikk).

Vi viser nå at initialsemantikk og degenerert finalsemantikk gir separabel semantikk under konsistens.

Teorem 2.20 *Anta* `DEGEN`*. For initialsemantikken* $\simeq^\alpha$ *relativ til en kjerne* $\simeq^x$ *bestemt av en* Σ *og en ligningsmengde* E*, har vi under* `KONSERV` *og* `TK`

$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* = \simeq^\alpha$$
$$\Updownarrow$$
E er initielt konsistent relativt til $\simeq^x$

Bevis: Anta først at E er initielt inkonsistent relativt til $\simeq^x$. Da er $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* \neq \simeq^\alpha$ ved sats 2.16 over.

Anta så initiell konsistens. At $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* \subseteq \simeq^\alpha$ er innlysende. Vi konsentrerer oss derfor om tilfellet $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* \supseteq \simeq^\alpha$. Induksjon på lengden n av en vilkårlig $\simeq^\alpha$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_\Sigma$:

$n = 1$: Trivielt har vi $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g$.

$n = k + 1; k \geq 1$: Da har vi en utledning $\langle g, \ldots, g_k, g' \rangle$. Induksjonshypotesen gir $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g_k$. Anta $g_k \underset{E}{\leftrightarrow} g'$. Da har vi $g_k(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g'$ og teoremet følger trivielt. Anta $g_k = c[g'_k]$ og $c[g''] = g'$ og $g'_k \simeq^x g''$, for noen $g'_k, g'' \in \mathcal{G}_{\Sigma^x}$. Ved `DEGEN` med observasjon 2.13, gir `TK` at $c[g'_k] \overset{*}{\underset{E}{\leftrightarrow}} g_x$ og $c[g''] \overset{*}{\underset{E}{\leftrightarrow}} g'_x$ for noen $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$.

Vi har antatt initiell konsistens eller det ved sats 2.7 side 35 ekvivalente begrep final kjernebevaring. Vi har derfor

$$\neg \exists c' \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^x} \mid c'[g'_k] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c'[g'']$$

Men da får vi altså

$$g_k = c[g'_k] \overset{*}{\underset{E}{\leftrightarrow}} g_x \simeq^x g'_x \overset{*}{\underset{E}{\leftrightarrow}} c[g''] = g'$$

m.a.o. $g_k(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g'$ og teoremet følger.
□

Teorem 2.21 *Anta $\Sigma^h = \emptyset$ og* `DEGEN`*. For finalsemantikken $\simeq^\omega$ relativ til en kjerne $\simeq^x$ bestemt av en Σ og en ligningsmengde E, har vi under* `KONSERV` *og* `TK`

$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* = \simeq^\omega$$
$$\Updownarrow$$
E er finalt konsistent relativt til $\simeq^x$

Bevis: Anta først at E er finalt inkonsistent relativt til $\simeq^x$. Ved satsene 2.14 (side 40) og 2.12 (side 39) har vi at E er ikke finalt kjernebevarende. Da er $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* \neq \simeq^\omega$ ved sats 2.16 over.

Anta så final konsistens. Anta $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* g'$ for vilkårlige $g, g' \in \mathcal{G}_\Sigma$. At $g \simeq^\omega g'$ kan vises ved induksjon på lengden n av en vilkårlig $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$-utledning $\langle g, \ldots, g' \rangle$.

$n = 1$: Ved `KONSERV` og `TK` er $\simeq^\omega$ refleksiv ved sats 2.5 på side 28. Følgelig er $g \simeq^\omega g$.

$n = k + 1; k \geq 1$: Da har vi en utledning $\langle g, \ldots, g_k, g' \rangle$. Induksjonshypotesen gir $g \simeq^\omega g_k$. Ethvert $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$-utledningssteg er på formen $g_i = g_x \simeq^x g'_x = g_{i+1}$, for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$; eller på formen $g_i \underset{E}{\leftrightarrow} g_{i+1}$ (for $g_i, g_{i+1} \in \mathcal{G}_\Sigma$). Anta $g_k = g_x \simeq^x g'_x = g'$ for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$: Siden E er finalt konsistent og da ved satsene 2.14 og 2.12 finalt kjernebevarende, har vi $g_k = g_x \simeq^\omega g'_x = g'$, og teoremet følger.

Anta $g_k \underset{E}{\leftrightarrow} g'$: Anta $g_k \not\simeq^\omega g'$. Siden $g_k \underset{E}{\leftrightarrow} g'$, må g_k, g' være av samme type. For at $g_k \not\simeq^\omega g'$ må det derfor finnes $c \in \mathcal{G}_\Sigma$ og $g_1, g_2 \in \mathcal{G}_{\Sigma^x}$, slik at

$$c[g_k] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

Men siden $g_k \overset{*}{\underset{E}{\leftrightarrow}} g'$, har vi

$$c[g_k] \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

som gir $g_1 \overset{*}{\underset{E}{\leftrightarrow}} g_2$. Ved `KONSERV` er dette umulig; så vi må ha $g_k \simeq^\omega g'$. Dette konkluderer inklusjonen $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^* \subseteq \simeq^\omega$.

Anta så $g \simeq^\omega g'$. Ved `DEGEN` og `TK` har vi $g \overset{*}{\underset{E}{\leftrightarrow}} g_x$ og $g' \overset{*}{\underset{E}{\leftrightarrow}} g'_x$ for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$. Siden $g \simeq^\omega g'$, har vi

$$\neg \exists c \in \mathcal{G}_\Sigma; g_1, g_2 \in \mathcal{G}_{\Sigma^x} \mid c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

Spesielt har vi da

$$g \overset{*}{\underset{E}{\leftrightarrow}} g_x \simeq^x g'_x \overset{*}{\underset{E}{\leftrightarrow}} g'$$

og dermed $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g'$.
□

Vi har nå vist at initialsemantikk og degenerert finalsemantikk gir separabel semantikk under konsistens. Således har vi vist at initialsemantikk og degenerert finalsemantikk under konsistens bevarer semantikken til sine komponenter. I den grad ønsket generatorsemantikk er den gitt av kjerne-semantikken og ønsket semantikk til definerte funksjonssymboler er den (konstruktive) gitt av ligningsmengden E, er det derfor under konsistens rimelig å si at initialsemantikk og degenerert finalsemantikk gir ønsket semantikk.

Videre gir teoremene 2.20 og 2.21 proposisjon 2.15.

2.3.8 Inkonsistens og tolkninger

Et definert funksjonssymbol hvis semantikk gir inkonsistens i forhold til en tiltenkt semantikk $\simeq^x$ på et generatorunivers, kan ikke tolkes som en funksjon i noen algebra A, som er slik at $g_x \not\simeq^x g'_x \Rightarrow A \not\models g_x = g'_x$ (A sies å *begrense* denne tiltenkte semantikken. Se ellers avsnitt A.1 i tillegg A). F.eks. kan ikke symbolet f fra eksempel 18 (side 32) tolkes til noen funksjon f i en algebra A som begrenser $\overset{*}{\underset{E_{\mathcal{I}nt\,c}}{\leftrightarrow}}\mathcal{G}_\Sigma$. Dette siden succ(pred(0)) og 0 da tolkes til samme element a i A, mens succ(0) tolkes til et element $b \neq a$. Dermed har vi $f(a) = a$ men også $f(a) = b$, og f er følgelig ingen funksjon.

Således generaliserer våre inkonsistensbegrep inkonsistensbegrepet i predikatlogikk som er ekvivalent med at den inkonsistente predikatmengden ikke har noen modell. Predikatlogisk inkonsistens vil si at true = false er bevisbart i predikatkalkylen. Ved en predefinert standard tolkning av true og false, kan det da ikke finnes noen modell hvor 'sant' er lik 'usant'. Vi har således en predefinert semantikk på {true, false}. Hos oss er det kjernesemantikken $\simeq^x$ som spiller rollen som predefinert semantikk.

2.3.9 Inkonsistens sett som inkongruens

Det kan se ut som om sats 2.16 på side 41 viser at relasjonen $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$ er ufølsom for inkonsistens. Dette er ikke riktig. Selv om $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^*$ (alltid) bevarer sine komponenter, gir inkonsistens her andre symptomer; nemlig inkongruens:

Sats 2.22 *Anta* DEGEN. *Anta* KONSERV *og* TK. *La* $\simeq^\omega$ *være finalsemantikken relativ til en* $\simeq^x$ *bestemt av en* Σ *og en* E. *Vi har*

$$E \text{ er finalt kjernebevarende relativt til } \simeq^x$$
$$\Updownarrow$$
$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* \text{ er inkongruent}$$

Bevis: Anta E er ikke finalt kjernebevarende. Ved (2.10) side 24 og observasjon 2.3 side 27 må det da finnes $c \in \mathcal{G}_\Sigma$ og $g_x, g'_x, g_1, g_2 \in \mathcal{G}_{\Sigma^x}$ slik at

$$c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x] \quad \text{men} \quad g_x \simeq^x g'_x$$

Siden $g_x \simeq^x g'_x$ har vi ved sats 2.16 at $g_x(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* g'_x$. Anta så at

$$c[g_x](\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_\Sigma)^* c[g'_x]$$

Vi kan ikke ha $c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x]$ siden det ville gi $g_1 \overset{*}{\underset{E}{\leftrightarrow}} g_2$ som er umulig ved **KONSERV**. Enhver $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$-utledning $\langle c[g_x], \ldots, c[g'_x]\rangle$ i $\mathcal{G}_\Sigma$ må derfor ha formen

$$\begin{aligned} c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} g_{x1} \simeq^x g_{x1}' \overset{*}{\underset{E}{\leftrightarrow}} g_{x2} \simeq^x g_{x2}' \overset{*}{\underset{E}{\leftrightarrow}} \cdots \\ \vdots \\ \cdots \overset{*}{\underset{E}{\leftrightarrow}} g_{xi} \simeq^x g_{xi}' \overset{*}{\underset{E}{\leftrightarrow}} \cdots \\ \vdots \\ \cdots \overset{*}{\underset{E}{\leftrightarrow}} g_{xn} \simeq^x g_{xn}' \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x] \end{aligned} \tag{2.14}$$

for $n \geq 2$ og $g_{xi}, g'_{xi} \in \mathcal{G}_{\Sigma^x}$; $1 \leq i \leq n$. Ved **KONSERV** må $g'_{xi} \simeq^x g_{xi+1}$ for $1 \leq i \leq n$. Men siden $g_1 \overset{*}{\underset{E}{\leftrightarrow}} c[g_x]$ og $c[g'_x] \overset{*}{\underset{E}{\leftrightarrow}} g_2$ får vi da $g_1 \simeq^x g_2$ som er en motsigelse. Følgelig har vi

$$c[g_x](\simeq^x \not\emptyset \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* c[g'_x]$$

og $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$ er ikke kongruent.

Anta så at $g(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* g'$ men $c[g](\simeq^x \not\emptyset \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* c[g']$. Ved **DEGEN** og **TK** har vi $g \overset{*}{\underset{E}{\leftrightarrow}} g_x$ og $g' \overset{*}{\underset{E}{\leftrightarrow}} g'_x$ for $g_x, g'_x \in \mathcal{G}_{\Sigma^x}$. Vi har således

$$g_x(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* g'_x$$

og ved sats 2.16 på side 41

$$g_x \simeq^x g'_x$$

Ved **DEGEN** og **TK** har vi også $c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1$ og $c[g'] \overset{*}{\underset{E}{\leftrightarrow}} g_2$ for $g_1, g_2 \in \mathcal{G}_{\Sigma^x}$. Nå kan ikke $g_1 \simeq^x g_2$, ellers var jo $c[g](\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* c[g']$. Vi har altså

$$c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g']$$

og da

$$c[g_x] \overset{*}{\underset{E}{\leftrightarrow}} c[g] \overset{*}{\underset{E}{\leftrightarrow}} g_1 \not\simeq^x g_2 \overset{*}{\underset{E}{\leftrightarrow}} c[g'] \overset{*}{\underset{E}{\leftrightarrow}} c[g'_x]$$

Vi har altså $g_x \not\simeq^\omega g'_x$, men $g_x \simeq^x g'_x$. Følgelig er E ikke finalt kjernebevarende. □

Følgende korollar følger ved sats 2.7 (side 35):

Korollar 2.23 *Anta* **DEGEN**. *Anta* **KONSERV** *og* **TK**. *La* $\simeq^\alpha$ *være initialsemantikken relativ til en* $\simeq^x$ *bestemt av en* Σ *og en* E. *Vi har*

$$\begin{array}{c} E \text{ er initielt inkonsistent relativt til } \simeq^x \\ \Updownarrow \\ (\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* \text{ er inkongruent} \end{array}$$

Eksempel 23 Betrakt generatorsignaturen

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0}, \\ \mathsf{succ}, \\ \mathsf{pred} \end{array} \right\}$$

Betrakt E_{f} og $E_{\mathcal{I}nt_c}$ fra eksempel 18 på side 32.

La $\Sigma = \mathsf{Int} \cup \{\mathsf{f}\}$. La $\simeq^x$ være basis-initialsemantikken spesifisert av $E_{\mathcal{I}nt_c}$. La $\simeq^\alpha$ være initialsemantikken relativ til $\simeq^x$ spesifisert av E_{f} og Σ. Vi hadde

$$\mathsf{f}(\mathsf{succ}(\mathsf{pred}(\mathsf{0}))) \simeq^\alpha \mathsf{succ}(\mathsf{0}) \quad \text{og} \quad \mathsf{f}(\mathsf{0}) \simeq^\alpha \mathsf{0}$$

og dermed

$$\mathsf{succ}(0) \simeq^{\alpha} 0.$$

og dermed initial inkonsistens relativt til $\simeq^x$.

Betrakt så relasjonen $(\simeq^x \cup \overset{*}{\underset{E\,\mathsf{f}}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$. Vi har

$$\mathsf{succ}(\mathsf{pred}(0))\ (\simeq^x \cup \overset{*}{\underset{E\,\mathsf{f}}{\leftrightarrow}} \mathcal{G}_\Sigma)^*\ 0$$

men altså

$$\mathsf{f}(\mathsf{succ}(\mathsf{pred}(0)))\ (\simeq^x \not\cup \overset{*}{\underset{E\,\mathsf{f}}{\leftrightarrow}} \mathcal{G}_\Sigma)^*\ \mathsf{f}(0)$$

○

2.3.10 Modulær oppbygging av semantikk

I implementasjonssammenheng er det naturlig å assosiere kjernesemantikken i initial- og finalsemantikk til generatorsemantikk. Den relative delen av initial-/finalsemantikken er da ment å spesifisere definerte funksjonssymboler.

Kjernesemantikken kan igjen være en semantikk relativ til en annen kjerne. Således kan semantikker bygges opp fra enklere semantikker. Ved dette får vi en oppbygging av formelle datatyper fra enklere/andre formelle datatyper.

Nå fungerer formelle datatyper som «grensesnitt» mellom abstrakte datatyper og eventuelle implementasjoner i et programmeringsspråk. Oppbygging av formelle datatyper fra andre formelle datatyper, svarer da til oppbygging av moduler fra andre moduler.

For våre generelle initial- og finalsemantikker er en slik generell oppbygging av semantikker fra enklere semantikker alltid ved hjelp av ligningslogikk. Gitt et formelt grep om kjernen i en semantikk til en formell datatype, utgjør den formelle datatypen en formell *omgivelse* for formell resonnering; sett på «øverste nivå», bestående av et term-univers og et formelt ligningslogisk system relativ til en kjerne.

Den «innerste» kjerne i en semantikk $\simeq$ skal vi kalle den ***atomære*** semantikk i $\simeq$. En atomær semantikk er ikke definert relativt til en annen semantikk. For å ha et formelt ligningslogisk grep om hele semantikken, kan vi kreve at atomær semantikk skal defineres over en ligningsmengde og slik at ligningslogikk mer eller mindre umiddelbart har et grep om semantikken. F.eks. er basis-initialsemantikk og fri semantikk således velegnet til atomære semantikker. I neste kapittel skal vi se på en annen semantikk som er egnet som atomær semantikk.

Det er mange interessante sider knyttet til en modulær oppbygging av formelle datatyper. Hvilke kriterier må til for at konsistens overlever fra den atomære semantikken og oppover? Hvilke kriterier skal til for å unngå at inkonsistens skal bli skjult av en senere påbygging? Plasshensyn gjør at vi utsetter denne diskusjonen til en senere anledning.

Imidlertid definerer vi nå følgende nyttige begrep:

Definisjon 2.13 *Det er mulig å assosiere en* ***formell omgivelse*** *til en semantikk som følger:*

- *For en atomær semantikk $\simeq^a$ definert over en $E^a \subseteq \mathcal{E}(\mathcal{T}_{\Sigma^a}(\mathcal{V}))$, er*

$$\langle \mathcal{G}_{\Sigma^a}, E^a \rangle$$

den formelle omgivelse for $\simeq^a$.

- *La $\simeq^x$ være en semantikk, og la $\mathcal{X}$ være den formelle omgivelse for $\simeq^x$. For en semantikk $\simeq$ relativ til $\simeq^x$ bestemt av en Σ og en ligningsmengde $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$, er*

$$\mathcal{X} \cup \{\langle \mathcal{G}_\Sigma, E\rangle\}$$

den formelle omgivelse for $\simeq$.

En formell omgivelse for en semantikk består av bestanddeler for de grunnleggende ligningslogiske maskiner som er involvert i definisjonen av semantikken. Den *ytterste* eller *øverste* komponent i en formell omgivelse for en semantikk — i definisjon 2.13 $\langle \mathcal{G}_{\Sigma^a}, E^a\rangle$ for $\simeq^a$ og $\langle \mathcal{G}_\Sigma, E\rangle$ for $\simeq$ — skal vi i noen anledninger kalle den ***logiske omgivelse*** eller ***grensesnitt-omgivelsen*** for semantikken.

* * *

Vi har nå definert ulike typer semantikk og utfra disse, forskjellige typer formelle datatyper. Oppbyggingen av semantikkene er ved ligningslogikk. Gitt atomær semantikk definert direkte ved ligningslogikk, har vi derved et formelt grep på våre semantikker.

I neste avsnitt søker vi å videreføre dette formelle grep til mekaniserbarhet.

2.4 Mot mekanisk resonnering

Generaliserer vi diskusjonen for basis-initialsemantikk i avsnitt 2.2.9 (side 21) til generell initial- og finalsemantikk, er det på det rene at vi er interessert i påstander som

$$\forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \simeq t\sigma \tag{2.15}$$

for en gitt initial-/finalsemantikk $\simeq$ på en $\mathcal{G}_\Sigma$. (Påstanden (2.15) generaliserer (2.5) på side 22.) Vi har

$$\mathcal{G}_\Sigma/\!\simeq \models s = t \;\Leftrightarrow\; \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \simeq t\sigma$$

(punkt 2 på side 23), så vi er altså interessert i påstander som er sanne i den formelle datatypen $\mathcal{G}_\Sigma/\!\simeq$. Vi skal kalle det å finne ut for en vilkårlig ligning $s = t$ om $\mathcal{G}_\Sigma/\!\simeq \models s = t$, for ***resolusjon***[8] av $\simeq$ i $\mathcal{G}_\Sigma/\!\simeq$. I dette avsnittet presenterer vi noen i datatype-sammenheng, eksisterende sentrale tilnærmelser til *algoritmisk* resolusjon for basis-initial- og basis-finalsemantikker. Vi drøfter muligheten for resolusjon av våre generelle initial- og finalsemantikker. Vi ser også litt på algoritmisk etablering av (in)konsistens.

I forbindelse med resolusjon av semantikk skal vi imidlertid først påpeke en begrensning i beviskraft hos ligningslogikk i forholdet til formelle datatyper. Selv om det er assosiert en formell ligningslogisk omgivelse med enhver formell datatype, er ligningslogikk i seg selv generelt ikke «sterk nok» for resolusjon i formelle datatyper. Dette skal vi nå se i forbindelse med basis-initialsemantikk.

2.4.1 Induktive vs. logiske konsekvenser

For en vilkårlig $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ er en vilkårlig algebra A en ***modell*** for E dersom A tilfredstiller hver ligning i E. Vi betegner klassen av Σ-algebraer som er

[8] 'Resolusjon' brukes her *ikke* i betydningen forbundet med en bevisstrategi som ligger til grunn for implementasjoner (f.eks. Prolog) av såkalt *Logic Programming* ([Rob65] og f.eks. [Apt90]). Vi ønsker at 'resolusjon' her skal ha konnotasjonen 'løsning' (på et problem); evt. 'oppløsning', i den forstand at god oppløsning lar oss se en semantikks enkelte bestanddeler som er par av termer, m.a.o. ligninger.

modeller for E, med $\mathcal{M}od_\Sigma(E)$. For en vilkårlig $s = t \in \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ skriver vi $E \models s = t$ for å antyde at $s = t$ er tilfredstilt av samtlige elementer i $\mathcal{M}od_\Sigma(E)$. Dvs. at $s = t$ er tilfredstilt i alle tilfeller der alle ligningene i E er tilfredstilt; altså hvis alle ligningene i E er tilfredstilt, må *nødvendigvis* $s = t$ være tilfredstilt. Vi sier da at $s = t$ er en ***logisk konsekvens*** av E. Den ***logiske teori*** for E — mengden av alle logiske konsekvenser av E betegnes med $\mathcal{T}eo(E)$.

Ligningslogikk kan brukes til å utlede eller verifisere logiske konsekvenser av en gitt ligningsmengde. Følgende teorem stadfester at det formelle systemet ligningslogikk for en ligningsmengde E er *sunt*; i den forstand at enhver ligning utledet av systemet er en logisk konsekvens av E, og *komplett*; i den forstand at alle logiske konsekvenser av E er utledbare i systemet:

Teorem 2.24 (Birkhoff [Bir35]**)** *La Σ være en vilkårlig signatur. For en vilkårlig $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ har vi for alle $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$:*

$$E \models s = t \;\Leftrightarrow\; s \overset{*}{\underset{E}{\leftrightarrow}} t$$

For gitte Σ og E, er vi basis-initialsemantisk interessert i resolusjon i den basis-initielle datatypen $\mathcal{G}_\Sigma/E$. Algebraen $\mathcal{G}_\Sigma/E$ er en modell for E (vises i avsnitt A.2). Dette vil si at vi, snarere enn å være interessert i mengden av ligninger tilfredstilt av samtlige modeller for E, er interessert i mengden av ligninger tilfredstilt av én slik modell; nemlig $\mathcal{G}_\Sigma/E$. Mengden av ***induktive konsekvenser*** av E — ligninger tilfredstilt av $\mathcal{G}_\Sigma/E$ — betegnes med $\mathcal{I}nd(E)$. Mengden $\mathcal{I}nd(E)$ kalles den ***induktive teorien*** for E. Det er lettere å tilfredstille én modell enn mange modeller, så ikke uventet er *generelt* $\mathcal{T}eo(E)$ en ekte delmengde av $\mathcal{I}nd(E)$:

Eksempel 24 Nat^+-algebraen $\mathcal{N}at^{+a}$ fra eksempel 4 side 15 er en modell for Peano-aksiomene E_P i eksempel 9 side 20. Ligningen $\mathsf{x{+}y = y{+}x}$ er ikke tilfredstilt i $\mathcal{N}at^{+a}$ siden $a +^a 0 = a \neq 0 = 0 +^a a$. Ergo er $\mathsf{x{+}y = y{+}x} \notin \mathcal{T}eo(E_P)$.

I eksempel 10 side 20 så vi at $\overset{*}{\underset{E_P}{\leftrightarrow}}_{\mathcal{G}_{\mathsf{Nat}^+}} = \simeq^{\mathcal{N}at^+}_{\mathcal{G}_{\mathsf{Nat}^+}}$. Siden $\phi^{\mathcal{N}at^+}_{\mathcal{G}_{\mathsf{Nat}^+}}$ er surjektiv, har vi at $\mathcal{G}_{\mathsf{Nat}^+}/\simeq^{\mathcal{N}at^+}_{\mathcal{G}_{\mathsf{Nat}^+}} = \mathcal{G}_{\mathsf{Nat}^+}/E_P$ og $\mathcal{N}at^+$ er elementært ekvivalente. Ligningen $\mathsf{x{+}y = y{+}x}$ er tilfredstilt i $\mathcal{N}at^+$. Dermed er $\mathsf{x{+}y = y{+}x}$ også tilfredstilt i $\mathcal{G}_{\mathsf{Nat}^+}/E_P$. Altså er $\mathcal{T}eo(E_P) \neq \mathcal{I}nd(E_P)$.
○

Ligningsmengder med egenskapen $\mathcal{T}eo(E) = \mathcal{I}nd(E)$ kalles ***ω-komplette*** [Tar68]; en egenskap som altså eksempel 24 demonstrerer at ikke holder generelt.

Generelt unnslipper altså påstander som $\mathcal{G}_\Sigma/E \models s = t$ det formelle grepet til ligningslogikk. I avsnitt A.2 i tilleggskapittel A utdypes dette nærmere.

For formell resonnering i basis-initielle formelle datatyper, er derfor andre klasser av formelle systemer nødvendige. En tilnærming er å innføre en *induksjonsregel*, som formaliserer allkvantoriseringen i (2.5) på side 22. Dette effektuerer bevis ved induksjon over syntaktisk oppbygging av grunntermer, og kalles ofte *strukturell induksjon* [Bur69] eller også *generator induksjon* bla. [Gut75]. En nødvendig (men ikke tilstrekkelig) betingelse for kompletthet av slike induktive systemer, er et rekursivt tellbart term-univers.

Selv om ligningslogikk generelt er for svak for resolusjon i formelle datatyper, er ligningslogikk likefullt en del av formelle systemer rettet mot formelle datatyper. Ligningslogikk er dessuten sentral i oppbygningen av formelle datatyper.

For tilnærmelser til *mekanisk* formell resonnering — her algoritmisk resolusjon av (basis-)semantikk — spiller derfor en *deterministisk* form for ligningslogikk en sentral rolle:

2.4.2 Konvergente omskrivningssystemer

Vi betrakter en gitt $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$. Termuniverset for diskusjonen er her $\mathcal{T}_\Sigma(\mathcal{V})$. La $\mathcal{T} \subseteq \mathcal{T}_\Sigma(\mathcal{V})$ være en termmengde.

- R er $\mathcal{T}$-***terminerende*** dersom det ikke finnes noen uendelig utledning $\langle t, \ldots \rangle$ i $\mathcal{T}_\Sigma(\mathcal{V})$ for noen $t \in \mathcal{T}$.
- R er $\mathcal{T}$-***Church-Rosser*** dersom for alle $s, t \in \mathcal{T}$

$$s \overset{*}{\underset{R}{\leftrightarrow}} t \;\Rightarrow\; \exists u \in \mathcal{T}_\Sigma(\mathcal{V}) \mid s \overset{*}{\underset{R}{\rightarrow}} u \overset{*}{\underset{R}{\leftarrow}} t$$

For vilkårlige s, t skriver vi $s \overset{!}{\underset{R}{\rightarrow}} t$ dersom $s \overset{*}{\underset{R}{\rightarrow}} t$ og $t \not\underset{R}{\rightarrow} u$ for alle $u \in \mathcal{T}_\Sigma(\mathcal{V})$ og kaller t en ***normalform*** for s.

- R har ***entydige*** $\mathcal{T}$-***normalformer*** dersom for alle $t \in \mathcal{T}$; $u, v \in \mathcal{T}_\Sigma(\mathcal{V})$

$$u \overset{!}{\underset{R}{\leftarrow}} t \overset{!}{\underset{R}{\rightarrow}} v \;\Rightarrow\; u = v$$

Dersom det finnes nøyaktig en normalform for en vilkårlig $s \in \mathcal{T}_\Sigma(\mathcal{V})$, betegner vi denne med $s!R$.

Anta at R er $\mathcal{T}$-terminerende, har entydige $\mathcal{T}$-normalformer og er $\mathcal{T}$-Church-Rosser. Vi sier da at R er $\mathcal{T}$-***konvergent***. Da finnes $s!R$ og $t!R$ for vilkårlige $s, t \in \mathcal{T}$, og

$$s \overset{*}{\underset{R}{\leftrightarrow}} t \;\Leftrightarrow\; s!R = t!R \tag{2.16}$$

(Bevis i avsnitt A.3 i tilleggskapittel A).

Det er lett å se at omskrivningssystemet utgjort av R da kan brukes til å avgjøre hvorvidt $s \overset{*}{\underset{R}{\leftrightarrow}} t$, for vilkårlige $s, t \in \mathcal{T}$. M.a.o. er da mengden $\mathcal{T}eo(R)_\mathcal{T}$ avgjørbar.

I lys av ligningslogikk er det da interessant gitt en vilkårlig ligningsmengde E, å finne en $\mathcal{T}$-konvergent regelmengde R slik at $\overset{*}{\underset{R}{\leftrightarrow}}_\mathcal{T} = \overset{*}{\underset{E}{\leftrightarrow}}_\mathcal{T}$. En slik regelmengde kalles $\mathcal{T}$-***komplett*** for E. Dette kompletthetsbegrepet for en $\mathcal{T}$-konvergent R deles opp i

$\mathcal{T}$-***Sunt for*** E: $\overset{*}{\underset{R}{\leftrightarrow}}_\mathcal{T} \subseteq \overset{*}{\underset{E}{\leftrightarrow}}_\mathcal{T}$

$\mathcal{T}$-***Adekvat for*** E: $\overset{*}{\underset{E}{\leftrightarrow}}_\mathcal{T} \subseteq \overset{*}{\underset{R}{\leftrightarrow}}_\mathcal{T}$

Dersom $\mathcal{T}$ er identisk med termuniverset i diskusjonen, sløyfes prefikset $\mathcal{T}$- fra begrepene ovenfor. Dersom $\mathcal{T}$ består av grunntermer over en signatur gitt utfra diskusjonen, prefikser vi ofte begrepene med *grunn-*.

For bevis av (2.16) og mer om konvergens, se avsnitt A.3 i tilleggskapittel A.

Eksempel 25 Ligningsmengden $E_{\mathcal{I}nt^+} = E_{+_\mathbb{Z}} \cup E_{\mathcal{I}nt_c}$ fra eksempel 11 side 21 er, sett som regelmengde ($R_{\mathcal{I}nt^+}$), allerede konvergent og derfor trivielt komplett for $E_{\mathcal{I}nt^+}$.
○

Ligninger sammen med ligningslogikk utgjør den operasjonelle delen av en formell datatype. I lys av dette ser vi på ligninger som (ikke-deterministiske) abstrakte programmer. Konvergente omskrivningssystemer er da en ***deterministisk*** form for slike abstrakte programmer, i den forstand at konvergente omskrivningssystemer gir samme output for et gitt input.

Det er naturlig å tenke konstruktivt ved algebraisk funksjonsspesifikasjon. Dette fører ofte naturlig til at den algebraiske funksjonsspesifikasjonen er konvergent.

Generelt er det imidlertid ikke-trivielt å finne en komplett regelmengde for en gitt ligningsmengde. Knuth&Bendix-komplettering i form av Knuth&Bendix-prosesser [KB70] — abstrakte maskiner — gir i noen tilfeller en komplett regelmengde gitt en ligningsmengde E.

Eksempel 26 Betrakt de ligningslogiske *gruppe*-aksiomer:

$$E_G = \left\{ \begin{array}{l} 1 \oplus x = x, \\ x^{-1} \oplus x = 1, \\ (x \oplus y) \oplus z = x \oplus (y \oplus z) \end{array} \right\}$$

E_G er ikke konvergent. En mulig for E_G komplett regelmengde er denne ikke-intuitive R_G framkommet ved komplettering [KB70]:

$$R_G = \left\{ \begin{array}{l} 1 \oplus x \rightarrow x, \\ x^{-1} \oplus x \rightarrow 1, \\ (x \oplus y) \oplus z \rightarrow x \oplus (y \oplus z), \\ x^{-1} \oplus (x \oplus y) \rightarrow y, \\ x \oplus 1 \rightarrow x, \\ 1^{-1} \rightarrow 1, \\ (x^{-1})^{-1} \rightarrow x, \\ x \oplus x^{-1} \rightarrow 1, \\ x \oplus (x^{-1} \oplus y) \rightarrow y, \\ (x \oplus y)^{-1} \rightarrow x^{-1} \oplus y^{-1} \end{array} \right\}$$

○

I avsnitt 2.4.4 presenteres Knuth&Bendix-prosesser.

Det finnes ligningsmengder for hvilke ingen endelig komplett regelmengde finnes:

Eksempel 27 I forbindelse med implementasjon av den abstrakte datatypen 'mengder av naturlige tall', er vi interessert i den formelle basis-initielle datatypen spesifisert av følgende SetNat og E_{SetNat}:

$$\mathsf{SetNat} = \left\{ \begin{array}{l} 0 : \mathsf{nat}, \\ \mathsf{succ} : \mathsf{nat} \rightarrow \mathsf{nat}, \\ \emptyset : \mathsf{setnat}, \\ \mathsf{add} : \mathsf{setnat} \times \mathsf{nat} \rightarrow \mathsf{setnat} \end{array} \right\}$$

$$E_{\mathsf{SetNat}} = \left\{ \begin{array}{l} \mathsf{add}(\mathsf{add}(s,x),x) = \mathsf{add}(s,x), \\ \mathsf{add}(\mathsf{add}(s,x),y) = \mathsf{add}(\mathsf{add}(s,y),x) \end{array} \right\}$$

Imidlertid finnes ingen endelig regelmengde som er komplett for E_{SetNat}, ei heller $\mathcal{G}_{\mathsf{SetNat}}$-komplett.[9]

○

2.4.3 Resolusjonsmetoder for basis-semantikker

Vi nevner her noen tilnærmelser til algoritmisk resolusjon. Metoder som i ulik grad søker algoritmisk resolusjon, skal vi kalle ***resolusjonsmetoder***.

For ω-komplette ligningsmengder, er det klart at konvergente omskrivningssystemer umiddelbart gir full algoritmisk resolusjon. Men generelt trengs imidlertid en sterkere form for formelt system, og med dette sterkere resolusjonsmetoder.

[9] Det finnes dog varianter av omskrivningssystemer, f.eks. *betingete* omskrivningssystemer og *ordnede* omskrivningssystemer (se f.eks. [DJ90]), i hvilke det ville være mulig å restriktere anvendelsen av den andre ligningen til grunn-instanser på en slik måte at terminering garanteres.

Ligningslogikk styrket med strukturell- eller generator-induksjon er et formelt system. Men som for mange formelle systemer, kreves ofte en viss form for innsikt (intelligens) for å lykkes med et bevis i systemet. *Mekanisk* resonnering kan derfor ofte bare oppnås delvis; som en interaktiv prosess med en menneskelig bruker. Den mekaniske delen av strukturell-/generator-induksjonssystemer er naturlig ivaretatt delvis av grunn-konvergent omskriving. Resten av bevisprosessen skjer interaktivt med bruker. Se f.eks. [Lys93] og [Lys94b].

Implementerte eksempler på slike interaktive bevis-førere basert delvis på ligningslogikk, er LP [GG91] og RRL [HZ89]. En implementasjon av flere (andre) bevisstrategier er beskrevet i [BM88].

Det finnes metoder som søker en tilnærming til hel-mekanisk resolusjon av basis-semantikker. For basis-initialsemantikk finnes eksempelvis *induktiv komplettering* (*induksjonsløs induksjon*) [Mus80] og bl.a. [HH82, JK89, KM87], samt en videreutvikling av induktiv komplettering beskrevet i [Bac88]. Tilfellet basis-finalsemantikk (finalsemantikk med basis-initiell eller fri kjerne) er behandlet i [Lys92] og [Lys94a]. Alle disse resolusjonsmetoder er i sine generelle former avhengige av grunn-konvergens.

Resolusjon er generelt uavgjørbart, siden mengden $\mathcal{T}eo(E)$ generelt er uavgjørbar [Mar47, Pos47], så generell algoritmisk resolusjon er umulig.[10]

Men *refutering* er her algoritmisk for basis-semantikker, i det minste for konvergent spesifiserende ligningsmengde. M.a.o. dersom $\mathcal{G}_\Sigma/\simeq \not\models s = t$ for en ligning $s = t$ og en basis formell datatype $\mathcal{G}_\Sigma/\simeq$, så kan dette oppdages mekanisk og i endelig tid, gitt konvergent spesifiserende ligningsmengde E.

For basis-initialsemantikk vil det da nemlig finnes en $\tau \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$, slik at $s\tau!E \neq t\tau!E$; og for basis-finalsemantikk vil det finnes en $\rho \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$ og en kontekst $c \in \mathcal{G}_\Sigma$, slik at $c[s\sigma]!E \neq c[t\sigma]!E$.

Disse fakta kan oppdages ved *systematiske søk* i hhv. $\mathcal{S}bst^\Sigma$ og (essensielt) $\mathcal{S}bst^\Sigma \times \mathcal{G}_\Sigma$. Det er her essensielt at våre term-univers er rekursivt tellbare. (Kartesiske produkt over tellbare mengder er også tellbare.) Et systematisk søk kan uttrykkes ved *fairness*. For basis-initialsemantikk:

> *Søket er slik at dersom prosessen var uendelig, så oversees ingen* $\sigma \in \mathcal{S}bst^\Sigma$ *i det uendelige.*

og for basis-finalsemantikk:

> *Søket er slik at dersom prosessen var uendelig, så oversees intet par* $\langle c, \sigma\rangle$ *av kontekster* $c \in \mathcal{G}_\Sigma$ *og* $\sigma \in \mathcal{S}bst^\Sigma$ *i det uendelige.*

Metodene over som søker en tilnærming til hel-mekanisk resolusjon, er alle påbygginger på Knuth&Bendix-prosesser. Vi presenterer nå Knuth&Bendix-prosesser. Vi skal også presentere induktiv komplettering for å illustrere grunnprinsippet i utbygningene av Knuth&Bendix-prosesser som metodene over representerer.

2.4.4 Knuth&Bendix-komplettering

Hoved-mekanismen i Knuth&Bendix-komplettering er utleding av såkalte *kritiske par*. For å snakke om kritiske par, må vi imidlertid først si noe om *unifikasjon*.

[10] At resolusjon generelt er uavgjørbart, kan her *ikke* vises direkte via Gödels berømte ufullstendighetsteorem [Göd31], [EN58]. Dette teoremet etablerer at det finnes et utsagn i 1. ordens predikatlogikk som er sant i modellen (algebraen) $\mathcal{N}at^{+*}$; altså de naturlige tall med addisjon og multiplikasjon, men som ikke kan vises sant i noen konsistent predikatkalkyle for $\mathcal{N}at^{+*}$ (og heller ikke usant ved sunnhet). Man kan ikke overføre dette argumentet til den ligningslogiske verden, fordi ligningslogikk (gjerne utvidet med induksjonsregler) har et språk med (mye) mindre uttrykkskraft. Faktisk er resolusjon i en passende formell datatype for $\mathcal{N}at^{+*}$ avgjørbar! (Vises f.eks. ved transformasjon til problemet om avgjørbarhet av sannhet av polynomiske ligninger med naturlige tall som kvotienter.)

Unifikasjon

For vilkårlige termer s og t, er substitusjonen σ en ***unifikator*** for s og t, hvis

$$s\sigma = t\sigma$$

Termene s og t sies å være ***unifiserbare*** hvis s og t har en unifikator σ. To gitte termer kan ha ingen unifikator:

Eksempel 28 Det finnes ingen instansiering σ av x, slik at $(\mathsf{x}*\mathsf{x})\sigma = \mathsf{0}\sigma$.
○

To gitte termer kan ha flere unifikatorer:

Eksempel 29 Termene $(\mathsf{u}*\mathsf{v})+\mathsf{w}$ og $\mathsf{x}+\mathsf{y}$ har bl.a. unifikatorene σ og τ slik at

$\sigma(\mathsf{x}) = (\mathsf{u}*\mathsf{v})$, $\sigma(\mathsf{w}) = \mathsf{y}$ og identitet for u,v og y.
$\tau(\mathsf{x}) = \mathsf{1}*\mathsf{z}$, $\tau(\mathsf{u}) = \mathsf{1}$, $\tau(\mathsf{v}) = \mathsf{z}$, $\tau(\mathsf{w}) = \tau(\mathsf{y}) = \mathsf{0}$.

○

Merk at i eksemplet over, er τ en spesialisering av σ, i den forstand at det finnes en substitusjon ρ slik at $\sigma \circ \rho = \tau$; nemlig enhver substitusjon ρ slik at

$\rho(\mathsf{u}) = \mathsf{1}$, $\rho(\mathsf{v}) = \mathsf{z}$, $\rho(\mathsf{w}) = \mathsf{0}$.

(f.eks. τ selv.)

En unifikator σ for to termer s og t , er en ***mest generell unifikator*** for s og t, hvis enhver annen unifikator τ for s og t er slik at det finnes en ρ slik at

$$\sigma \circ \rho = \tau$$

Ethvert unifserbare term-par har en unik (opptil omnavning av variable) mest generelle unifikator. Å avgjøre om et term-par er unifiserbart og isåfall å finne den mest generelle unifikator, kan gjøres algoritmisk.

Kritiske par

Anta nå et omskrivningssystem $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$. La $v_i \to h_i$ og $v_j \to h_j$ være to vilkårlige (muligens like) regler i R. Vi antar at

1. enhver regel i R er slik at ingen høyreside har forekomster av variable som ikke forekommer i venstresiden. Denne antagelsen er nødvendig for terminering av $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$
2. Mengdene av variable forekommende i hhv. v_i og v_j er disjunkte; dvs. enhver variabel som forekommer i v_i forekommer ikke i v_j og omvendt. Dette kan tilfredstilles for en vilkårlig regelmengde, ved en passende omnavning av variable i reglene. Denne antagelsen endrer ingen egenskaper mht. omskriving i $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$.

Anta nå at det finnes en posisjon p i v_i slik at $v_i|p$ ikke er en variabel, og at $v_i|p$ og v_j er unifiserbare. La μ være den mest generelle unifikator for $v_i|p$ og v_j. Da kan termen $v_i\mu$ omskrives på to måter som involverer reglene $v_i \to h_i$ og $v_j \to h_j$:

$$v_i\mu[h_j\mu]_p \underset{R}{\leftarrow} v_i\mu \underset{R}{\rightarrow} h_i\mu$$

Paret $\langle v_i\mu[h_j\mu]_p, h_i\mu\rangle$ kalles et ***kritisk par*** for R. Dersom det finnes en u slik at

$$v_i\mu[h_j\mu]_p \overset{*}{\underset{R}{\rightarrow}} u \overset{*}{\underset{R}{\leftarrow}} h_i\mu$$

kalles $\langle v_i\mu[h_j\mu]_p, h_i\mu\rangle$ et ***trivielt*** kritisk par. Hvis ingen slik u finnes kalles det kritiske paret ***ekte***. (Grunnen til at $v_i|p$ over presiseres til ikke å være en variabel, er fordi i motsatt fall fås umiddelbart et trivielt kritisk par. Interessen ligger først og fremst i ekte kritiske par.)

Unifikasjon fungerer her som en slags dobbel matching: Det er her to regler i R involvert. Eksistensen av et kritisk par uttrykker at det eksisterer en term som kan omskrives (innebærer matching) på *to* måter i R. Denne termen uttrykkes i generell form som den mest generelle unifikasjonen av de to venstresidene i reglene (subterm av den ene); nemlig $v_i\mu|p$.

Et omskrivningssystem $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ er ***lokalt konfluent***, dersom

$$s \overset{*}{\underset{R}{\leftarrow}} w \overset{*}{\underset{R}{\rightarrow}} t \;\Rightarrow\; \exists u \mid s \overset{*}{\underset{R}{\rightarrow}} u \overset{*}{\underset{R}{\leftarrow}} t$$

Et omskrivningssystem $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ er ***globalt konfluent***, dersom

$$s \overset{*}{\underset{R}{\leftarrow}} w \overset{*}{\underset{R}{\rightarrow}} t \;\Rightarrow\; \exists u \mid s \overset{*}{\underset{R}{\rightarrow}} u \overset{*}{\underset{R}{\leftarrow}} t$$

Vi har:

Lemma 2.25 (Newman (avledet av et resultat i [New42])**)** *Et terminerende omskrivningssystem er lokalt konfluent hvis og bare hvis det er globalt konfluent.*

Lemma 2.26 *Et terminerende omskrivningssystem er konvergent hvis og bare hvis det er globalt konfluent.*

Det sentrale for hvordan Knuth&Bendix-komplettering fungerer er nå:

Lemma 2.27 ("Critical Pair Lemma" [KB70]**)** *Et omskrivningssystem er lokalt konfluent hvis og bare hvis det ikke har noen ekte kritiske par.*

Å finne alle ekte kritiske par i et omskrivningssystem kan gjøres algoritmisk. Det følger således fra lemmaene 2.25 og 2.26 at konvergens for terminerende omskrivningssystemer er avgjørbart.

Men merk at terminering generelt ikke er avgjørbart (vises ved uavgjørbarhet av *stoppeproblemet* [Tur36] og transformasjon fra Turing-maskiner til omskrivningssystemer).

Komplettering

Vi beskriver nå Knuth&Bendix-prosesser. Den opprinnelige beskrivelse finnes i [KB70]. Vår beskrivelse er inspirert av [Kir94].

Hensikten er altså å transformere en gitt muligens ikke-konvergent ligningsmengde E til en for E komplett regelmengde R.

Idéen er å generere for E sunne regler sålenge det finnes kritiske par for den (hittil) genererte regelmengden. Ligningsmengden E og en kandidat-regelmengde R opplever således transformasjoner gjennom kompletteringen. Invarianter gjennom kompletteringen er at $\overset{*}{\underset{E\cup R}{\leftrightarrow}}$ forblir uendret og at R alltid er terminerende.

Som sagt er terminering av omskrivningssystemer generelt ikke avgjørbart. Knuth&Bendix-prosesser forsynes av den grunn med en reduksjonsordning på termuniverset i tillegg til input-ligningsmengden. Reduksjonsordningen brukes til å sikre terminering av det genererte regelsettet. Knuth&Bendix-komplettering er beskrevet som et formelt system i figur 2.4.

Definer relasjonen $\vdash_{KB}$ over $\mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))\times\mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ slik at $\langle E, R\rangle \vdash_{KB} \langle E', R'\rangle$ hvis og bare hvis $\langle E', R'\rangle$ fås fra $\langle E, R\rangle$ i ett skritt ved å bruke inferensreglene i figur 2.4. Vi har:

Datastruktur:

$\langle E, R\rangle \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V})) \times \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$. Relasjonen $\succ$ er den gitte reduksjonsordning.

Inferensregler:

Init: For en $E_0 \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$:

$$\langle E_0, \emptyset\rangle$$

Forenkle1:

$$\frac{\langle E \cup \{s = t\}, R\rangle}{\langle E \cup \{r = t\}, R\rangle},\ s \underset{R}{\rightarrow} r$$

Forenkle2:

$$\frac{\langle E \cup \{s = t\}, R\rangle}{\langle E \cup \{s = r\}, R\rangle},\ t \underset{R}{\rightarrow} r$$

Slett:

$$\frac{\langle E \cup \{s = s\}, R\rangle}{\langle E, R\rangle}$$

Orienter1:

$$\frac{\langle E \cup \{s = t\}, R\rangle}{\langle E, R \cup \{s \rightarrow t\}\rangle},\ s \succ t$$

Orienter2:

$$\frac{\langle E \cup \{s = t\}, R\rangle}{\langle E, R \cup \{t \rightarrow s\}\rangle},\ t \succ s$$

Sammensett:

$$\frac{\langle E, R \cup \{s \rightarrow t\}\rangle}{\langle E, R \cup \{s \rightarrow r\}\rangle},\ t \underset{R}{\rightarrow} r$$

Kollaps:

$$\frac{\langle E, R \cup \{s \rightarrow t\}\rangle}{\langle E \cup \{r = t\}, R\rangle},\ \mathrm{s} \underset{R'}{\rightarrow} \mathrm{r}$$

for $R' \subseteq R \setminus \{s \rightarrow t\}$.

Utled:

$$\frac{\langle E, R\rangle}{\langle E \cup \{s = t\}, R\rangle}$$

for $\langle s, t\rangle$ et ekte kritisk par i R.

Figur 2.4: Inferensregler i Knuth&Bendix-komplettering.

Lemma 2.28 *Dersom*

$$\langle E, R\rangle \vdash^*_{KB} \langle E', R'\rangle$$

har vi at

- $\overset{*}{\underset{E\cup R}{\leftrightarrow}} = \overset{*}{\underset{E'\cup R'}{\leftrightarrow}}$
- *For reduksjonsordningen* $\succ$ *gitt til prosessen har vi* $\overset{*}{\underset{R}{\rightarrow}} \subseteq \succ \quad \Rightarrow \quad \overset{*}{\underset{R'}{\rightarrow}} \subseteq \succ$.

En Knuth&Bendix-prosess kan gå i det uendelige. Men dersom ingen inferensregel er anvendbar på en gitt $\langle E, R\rangle$, sier vi at prosessen er ***terminerende***.

Teorem 2.29 *Dersom*

$$\langle E, \emptyset\rangle \vdash^*_{KB} \langle \emptyset, R\rangle$$

og ingen inferensregel er anvendbar på $\langle \emptyset, R\rangle$*, har vi at* R *er komplett for* E.

Teorem 2.29 er nå enkel å verifisere: Ved lemma 2.28 har vi $\overset{*}{\underset{E}{\leftrightarrow}} = \overset{*}{\underset{R}{\leftrightarrow}}$ og at R er terminerende. Siden ingen regel er anvendbar på $\langle \emptyset, R\rangle$, er spesielt ikke **Utled** anvendbar. Følgelig finnes ingen ekte kritiske par i R. Ved lemmaene 2.27, 2.25 og 2.26 er da R konvergent. Ialt er altså R komplett for E. Vi kaller en Knuth&Bendix-prosess som terminerer i en konfigurasjon $\langle \emptyset, R\rangle$ for ***terminerende vellykket***.

En Knuth&Bendix-prosess kan også terminere i en konfigurasjon $\langle E \neq \emptyset, R\rangle$. Dette skjer dersom en ligning i E ikke kan orienteres ved noen av inferensreglene **Orienter1** eller **Orienter2**. Vi kaller en Knuth&Bendix-prosess som terminerer i en slik konfigurasjon for ***terminerende mislykket***.

Ikke-terminering

Knuth&Bendix-prosesser kan være ikke-terminerende, og likevel interessante. Betrakt en muligens uendelig sekvens

$$\langle\langle E_0, R_0\rangle, \ldots, \langle E_i, R_i\rangle, \ldots\rangle$$

slik at $\langle E_i, R_i\rangle \vdash_{KB} \langle E_{i+1}, R_{i+1}\rangle$. En slik sekvens kan sees å representere en særskilt Knuth&Bendix-prosess, dvs. en særskilt anvedelsesrekkefølge av inferensreglene i figur 2.4. Den ***induktive grense*** til en slik sekvens er definert som

$$E_\infty = \bigcup_{j\geq 0}^{\infty} \bigcap_{i\geq j}^{\infty} E_i \quad \text{og} \quad R_\infty = \bigcup_{j\geq 0}^{\infty} \bigcap_{i\geq j}^{\infty} R_i$$

Ligningsmengden E_∞ består av alle ligninger generert av denne særskilte Knuth&Bendix-prosessen, som ikke senere forsvinner fra en eller annen E_i. Vi kaller E_∞ mengden av ***vedvarende ligninger*** for prosessen. Likeledes er R_∞ mengden av ***vedvarende regler*** for prosessen.

En muligens uendelig Knuth&Bendix-prosess kalles ***vellykket*** dersom $E_\infty = \emptyset$ og R_∞ er konvergent. Tilfellet terminerende vellykket er her et spesialtilfelle for $\langle E_i, R_i\rangle = \langle E_n, R_n\rangle; i \geq n$ for et eller annet naturlig tall n.

For uendelige Knuth&Bendix-prosesser skal det for oss være interessant å avgjøre bl.a. følgende:

1. om en gitt generert regel er vedvarende
2. om R_∞ er endelig
3. hvis R_∞ er uendelig, om en endelig mengde regel-*skjemaer* kan konstrueres som uttrykker alle regler i R_∞

Arbeid under det siste punktet er presentert i bl.a. [Her92].

Et sentralt resultat i forbindelse med Knuth&Bendix-komplettering er:

Teorem 2.30 (Bachmair [Bac87]**)** *Anta en Knuth&Bendix-prosess med en assosiert muligens uendelig sekvens*

$$\langle \langle E_0, R_0 \rangle, \ldots, \langle E_i, R_i \rangle, \ldots \rangle$$

slik at $\langle E_i, R_i \rangle \vdash_{KB} \langle E_{i+1}, R_{i+1} \rangle$*. Anta prosessen er 'fair' mht. kritisk partesting, dvs. ingen regel i* R_∞ *blir oversett i det uendelige for anvendelse av inferensregelen* ***Utled****. Da har vi for vilkårlige termer* s *og* t*:*

$$s \overset{*}{\underset{E_0}{\leftrightarrow}} t \;\Leftrightarrow\; \exists j \mid s!R_j = t!R_j$$

2.4.5 Induktiv komplettering

La E være en vilkårlig ligningsmengde. Som sagt er generelt $\mathcal{T}eo(E)$ en ekte delmengde av $\mathcal{I}nd(E)$. Gitt en for E komplett regelmengde R, finnes en resolusjonsmetode for de ligninger som er logiske konsekvenser av E. Men generelt kan det altså finnes induktive konsekvenser som ikke er logiske konsekvenser, og vanlig omskriving i R blir da utilstrekkelig for resolusjon.

Vi beskriver nå meget kort induktiv komplettering som er et forsøk på å resolvere flere ligninger enn dem som også er logiske konsekvenser av E.

La $\mathcal{G}$ være en subalgebra av en $\mathcal{G}_\Sigma$. En term $t \in \mathcal{T}$ er $\mathcal{G}$***-(grunn)redusibel*** i et omskrivningssystem R dersom

$$\forall \sigma \in \mathcal{S}bst^{\mathcal{G}} \mid \exists u \mid t\sigma \overset{+}{\underset{R}{\rightarrow}} u$$

Grunnredusibilitet er avgjørbart [Pla85, KNZ87].

La $\mathcal{T}_\Sigma(\mathcal{V})$ være vårt term-univers. La E være en ligningsmengde og R en $\mathcal{G}_\Sigma$-komplett regelmengde for E. La $s = t$ være en vilkårlig ligning. Vi har da:

Teorem 2.31 *Anta en Knuth&Bendix-prosess slik at*

$$\langle \{s = t\}, R \rangle \vdash^*_{KB} \langle \emptyset, R' \rangle$$

Hvis enhver regel $v \rightarrow h$ *generert i prosessen er slik at* v *er* $\mathcal{G}_\Sigma$*-redusibel av* R*, så er* $s = t$ *en induktiv konsekvens av* E*.*

Teorem 2.32 *Anta en uendelig Knuth&Bendix-prosess gitt* $\langle \{s = t\}, R \rangle$*. Hvis enhver regel* $v \rightarrow h$ *generert i prosessen er slik at* v *er* $\mathcal{G}_\Sigma$*-redusibel av* R*, så er* $s = t$ *en induktiv konsekvens av* E*.*

Teorem 2.33 *Anta en Knuth&Bendix-prosess slik at*

$$\langle \{s = t\}, R \rangle \vdash^*_{KB} \langle E_k, R_k \rangle \vdash^*_{KB} \ldots$$

der R_k *inneholder en regel* $v \rightarrow h$ *slik at* v *ikke er* $\mathcal{G}_\Sigma$*-redusibel av* R*. Da er* $s = t$ *ikke en induktiv konsekvens av* E*.*

En viss intuisjon over teoremene 2.31, 2.32 og 2.33 kan fås ved å observere at dersom en regel $v \rightarrow h$ genereres og v ikke er $\mathcal{G}_\Sigma$-redusibel av R, så kollapser $v \rightarrow h$ kongruensklasser i $\mathcal{G}_\Sigma/R = \mathcal{G}_\Sigma/E$. I tilfellet tilstrekkelig kompletthet mhp. generatortermer, vil en slik kollaps kunne vise seg i inkonsistens relativt til generatorsemantikk i $\mathcal{G}_\Sigma$ gitt av E:

I konteksten av implementasjon av abstrakte datatyper, betrakt en signatur Σ disjunkt delt i generatorer Σ^c og definerte funksjonssymboler Σ^d. I implementasjonssammenheng er det naturlig å anta at

1. R er tilstrekkelig Σ^d-komplett mhp. $\mathcal{G}_{\Sigma^c}$.

2. reduksjonsordningen på termer i kompletteringsprosessen er slik at enhver $u \in \mathcal{T}_{\Sigma^d \cup \Sigma^c}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$.

Så anta altså at en regel $v \rightarrow h$ genereres og v ikke er $\mathcal{G}_\Sigma$-redusibel av R. Da finnes $\sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$ slik at $v\sigma = v\sigma!R$. Ved 1 må følgelig $v\sigma$ være en generatorterm, og ved 2 er da også $h\sigma$ en generatorterm. Ved 2 har vi videre $h\sigma \overset{*}{\underset{R}{\not\leftarrow}} v\sigma$, og siden $v\sigma$ ikke er redusibel av R, har vi også $v\sigma \overset{*}{\underset{R}{\not\leftarrow}} h\sigma$.

Generatortermene $v\sigma$ og $h\sigma$ er således i hver sin E-kongruensklasse. Ligningen $v = h$ — vitnet som viser at $s = t$ ikke er en induktiv konsekvens av E — er også et vitne på at $s = t$ innfører initiell inkonsistens relativt til generatorsemantikken spesifisert av E. Vi har altså initialsemantisk

$$v\sigma \; \overset{*}{\underset{E \cup \{s=t\}}{\leftrightarrow}} \; h\sigma \quad \text{til tross for at} \quad v\sigma \overset{*}{\underset{E}{\not\leftrightarrow}} h\sigma$$

Dersom den tilførte ligningen (hypotesen) $s = t$ vises ved induktiv komplettering å være en induktiv konsekvens, kan dette da sies å skje ved å vise at $s = t$ ikke fører til slik inkonsistens. Dette *bevis ved konsistens* prinsipp er gjennomgående i metodene beskrevet i [JK89], [Bac88], [Lys92] og [Lys94a].

2.4.6 Metoder som søker generell resolusjon

Hva så med metoder som søker resolusjon av generell initial- og finalsemantikk? I avsnitt 2.3.10 tok vi til orde for en modulær oppbygging av formelle datatyper. Kan det tenkes at en analog modulær sammensetting av resolusjonsmetoder er mulig? Dvs. kunne det tenkes at en resolusjonsmetode for en kjerne $\simeq^x$, og en resolusjonsmetode for $\overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma}$ for en ligningsmengde E, tilsammen gir en resolusjonsmetode for en semantikk relativ til $\simeq^x$ bestemt av Σ og E? Dette er ikke åpenbart; spesielt ikke for metoder som baserer seg på utleding av kritiske par.

En grunn til dette er at initial-/finalsemantikk er på sett og vis mer enn summen av sine deler: Initialsemantikk er eksempelvis ikke bare «summen» av kjernen og semantikken utgjort av den tilhørende ligningsmengden, for ved inkonsistens oppstår nye likheter i kjernen. Inkonsistens er således en egenskap i initialsemantikken, men ikke i delene som semantikken er bygget opp av. De forskjellige logiske maskiner i den formelle omgivelse for en semantikk må følgelig samarbeide på en eller annen *innfløkt* måte.

Gitt konsistens kan vi imidlertid ved teoremene 2.20 side 42 og 2.21 side 43, derimot betrakte den separable relasjonen

$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$$

som stedfortreder for initialsemantikk og degenerert finalsemantikk. Denne relasjonen er modulær i sin oppbygning i forhold til initial- og finalsemantikker. Denne relasjonen *kan* derfor være interessant i en videre diskusjon om muligheten av modulære metoder som søker resolusjon av generell initial- og finalsemantikk.

Vi kan iallefall under konsistens og under visse forutsetninger slå fast at refutering er algoritmisk for generell initialsemantikk og degenerert finalsemantikk.

Anta nemlig en initialsemantikk eller degenerert finalsemantikk $\simeq$ relativ til en kjerne $\simeq^x$ spesifisert av en Σ og en E. Under konsistens kan vi ved teoremene 2.20 side 42 og 2.21 side 43, altså betrakte relasjonen

$$(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}}_{\mathcal{G}_\Sigma})^*$$

Anta $\mathcal{G}_\Sigma/\simeq \not\models s = t$ for en vilkårlig ligning $s = t$. Da skulle vi altså ha

$$s\tau(\simeq^x \not\cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^* t\tau$$

for en substitusjon $\tau \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$. La oss anta at E er konvergent og at $\simeq^x$ er avgjørbar. Vi antar `KONSERV` og `TK`, og i tillegg at E-normalformer er i $\mathcal{G}_{\Sigma^x}$. Anta i tillegg at E er ***fri*** mhp. $\mathcal{G}_{\Sigma^x}$; dvs. E har ingen ligninger fra $\mathcal{E}(\mathcal{T}_{\Sigma^x}(\mathcal{V}))$.

Da må enhver $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$-utledning i $\mathcal{G}_\Sigma$ ha formen $g \overset{!}{\underset{E}{\rightarrow}} g_x \simeq^x g'_x \overset{!}{\underset{E}{\leftarrow}} g'$. Dette følger fra at enhver $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$-utledning under `KONSERV` har formen beskrevet i (2.14) på side 45, samt at alle $g_{x\,i}' \overset{*}{\underset{E}{\leftrightarrow}} g_{x\,i+1}$ forsvinner ved antagelsen om frihet mhp. $\mathcal{G}_{\Sigma^x}$.

For refutering under konsistens kan vi følgelig søke systematisk ('fair') i universet $\mathcal{S}bst^{\mathcal{G}_\Sigma}$ inntil

$$s\tau!E \not\simeq^x t\tau!E$$

En mulig innfallsvinkel til å finne generelle resolusjonsmetoder (uavhengig av konsistens), kan være gjennom *klasse-* og *utvidet (extended)* omskriving (se f.eks. [DJ90]). Klasse-omskrivning er omskriving modulo kongruensklasser. Utvidet omskriving er en beregningsmessig sterkere, men logisk svakere type omskriving enn klasse-omskriving, og brukes i *utvidet* komplettering [DJ90] og dessuten i forbindelse med resolusjon av basis-initialsemantikk [JK89] i noen tilfeller der det ikke er mulig å finne et komplett regelsett pga. uorienterbarhet. Det er klart at at klasse-omskriving i noen tilfeller kan overføres til omskriving modulo kjernesemantikk. Avhengig av hvordan en slik kjernesemantikk er spesifisert, kan det da tenkes at utvidet komplettering kan gi idéer til resolusjonsmetoder for våre generelle semantikker.

Det ville være interessant å gi en skikkeligere drøfting av generelle resolusjonsmetoder sammen med en mer utfyllende diskusjon om pragmatiske sider ved generell semantikk (modulær oppbygging, i hvilken grad modulær oppbygging av formelle datatyper samsvarer med fornuftige programmeringsstrategier osv.). Det har vi ikke plass til her. Linjen videre med hensyn til generell semantikk skal være i form av en anvendelse, hvor atomær semantikk spesifiseres på en ny måte (kaptittel 3). Det er ikke opplagt at utvidet komplettering kan danne grunnlaget for en resolusjonsmetode til slik semantikk. Istedet skal vi se at slik semantikk av og til kan reduseres til basis-semantikker med tilhørende eksisterende resolusjonsmetoder.

2.4.7 Algoritmisk oppdagbarhet av inkonsistens

Anta en initial- eller finalsemantikk $\simeq$ relativ til en kjerne $\simeq^x$ spesifisert av en Σ og en E. Dersom E er konvergent og $\simeq^x$ er avgjørbar, kan initial inkonsistens og mangel på final kjernebevaring oppdages algoritmisk. Vi antar `TK`, og at E-normalformer er i $\mathcal{G}_{\Sigma^x}$. Ikke final kjernebevaring vil da si at det finnes $c \in \mathcal{G}_\Sigma$ og $g_x, g'_x, g_1, g_2 \in \mathcal{G}_{\Sigma^x}$ slik at

$$c[g_x] \overset{!}{\underset{E}{\rightarrow}} g_1 \not\simeq^x g_2 \overset{!}{\underset{E}{\leftarrow}} c[g'_x] \quad \text{men} \quad g_x \simeq^x g'_x$$

Dette faktum kan nå, siden $\simeq^x$ er avgjørbar, oppdages algoritmisk ved et systematisk ('fair') søk i (essensielt) universet $\mathcal{G}_{\Sigma^x} \times \mathcal{G}_{\Sigma^x} \times \mathcal{G}_\Sigma$. Vi minner så om ekvivalensen mellom initiell konsistens og final kjernebevaring ved sats 2.7 (side 35).

Merk at dette ved sats 2.22 side 44 er det samme som å oppdage inkongruens i relasjonen $(\simeq^x \cup \overset{*}{\underset{E}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$.

På den annen side kan selvfølgelig ikke et slikt uttømmende søk i seg selv brukes generelt for å etablere konsistens: Generelt er termuniverset uendelig.

Eksempel 30 La oss betrakte en vilkårlig basis-initialsemantikk spesifisert ved en signatur Σ og en ligningsmengde E. Vi antar Σ disjunkt delt i Σ^c og Σ^d av hhv. generatorer og definerte funksjonssymboler. Likeledes antar vi E delt disjunkt i E^c og E^d som spesifiserer hhv. basis-initialsemantikk på $\mathcal{G}_{\Sigma^c}$ og semantikk til de definerte funksjonssymboler.

Anta E^c og E^d begge er hver for seg konvergente, på en slik måte at alle E^d-normalformer er i $\mathcal{G}_{\Sigma^c}$. Da kan vi anvende det naïve prinsipp om uttømmende søk over, for å oppdage inkonsistens. Gitt inkonsistens, må det nemlig finnes $g_c, g'_c, g_1, g_2 \in \mathcal{G}_{\Sigma^c}$ og $c \in \mathcal{G}_\Sigma$ slik at

$$g_c \xrightarrow[E^c]{!} g_c! = g'_c! \xleftarrow[E^c]{!} g'_c$$

til tross for at

$$c[g_c] \xrightarrow[E^d]{!} g_1 \xrightarrow[E^c]{!} g_1! \neq g_2! \xleftarrow[E^c]{!} g_2 \xleftarrow[E^d]{!} c[g'_c]$$

(Eller $(c[g_c]!E^d)E^c! \neq (c[g'_c]!E^d)E^c!$ til tross for at $g_c!E^c = g'_c!E^c$.)

Å etablere konsistens, blir på den annen side det samme som å etablere at

$$g_c \overset{*}{\underset{E^c}{\leftrightarrow}} g'_c \;\Rightarrow\; c[g_c] \xrightarrow[E^d]{!} g_1 \xrightarrow[E^c]{!} g_1! = g_2! \xleftarrow[E^c]{!} g_2 \xleftarrow[E^d]{!} c[g'_c]$$

for noen $g_1, g_2 \in \mathcal{G}_{\Sigma^c}$ for alle $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$ og $c \in \mathcal{G}_\Sigma$; som vil si det samme som å etablere kongruens for relasjonen

$$(\overset{*}{\underset{E^c}{\leftrightarrow}} \mathcal{G}_{\Sigma^c} \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$$

○

Eksempel 31 Betrakt initialsemantikken $\simeq^\alpha$ relativ til $\overset{*}{\underset{E^x}{\leftrightarrow}} \mathcal{G}_{\Sigma^x}$ spesifisert av $\Sigma = \Sigma^x \cup \Sigma^d$ og E^d som følger:

$$\Sigma^x = \mathsf{SetNat} = \left\{ \begin{array}{l} \mathsf{0 : nat}, \\ \mathsf{succ : nat \longrightarrow nat}, \\ \mathsf{\emptyset : setnat}, \\ \mathsf{add : setnat \times nat \longrightarrow setnat} \end{array} \right\}$$

$$\Sigma^d = \{\ \mathsf{mpc : setnat \longrightarrow nat}\ \}$$

$$E^x = E_{\mathsf{SetNat}} = \left\{ \begin{array}{l} \mathsf{add(add(s,x),x) = add(s,x)}, \\ \mathsf{add(add(s,x),y) = add(add(s,y),x)} \end{array} \right\}$$

$$E^d = \left\{ \begin{array}{l} \mathsf{mpc(\emptyset) = 0}, \\ \mathsf{mpc(add(s,x)) = succ(mpc(s))} \end{array} \right\}$$

Vi kan også betrakte finalsemantikken $\simeq^\omega$ relativ til, og spesifisert av samme. Siden

$$\mathsf{mpc(add(add(\emptyset,0),0))} \overset{*}{\underset{E^d}{\leftrightarrow}} \mathsf{succ(succ(0))} \overset{*}{\underset{E^x}{\not\leftrightarrow}} \mathsf{succ(0)} \overset{*}{\underset{E^d}{\leftrightarrow}} \mathsf{mpc(add(\emptyset,0))}$$

er E^d initielt og finalt inkonsistent relativt til $\overset{*}{\underset{E^x}{\leftrightarrow}} \mathcal{G}_{\Sigma^x}$. ($E^d$ er finalt inkonsistent mhp. $\Sigma^o = \emptyset$ (se definisjon 2.12 side 33). Vi har `DEGEN` og `TK` og `KONSERV`, så da er altså E^d heller ikke finalt kjernebevarende.)

Nå er E^d konvergent. Det er ikke E^x, men hadde det eksistert en for E^x komplett regelmengde R, kunne et systematisk søk oppdage inkonsistensen, på analog måte som i eksempel 30. I tråd med det som er sagt i begynnelsen av dette avsnittet, kan inkonsistensen også oppdages hvis $\overset{*}{\underset{E^x}{\leftrightarrow}} \mathcal{G}_{\Sigma^x}$ på annen måte kan gjøres avgjørbar. Dette kommer vi tilbake til i kapittel 3.

○

Konsistens er generelt uavgjørbart, siden konsistens for spesialtilfellet basis-initialsemantikk er generelt uavgjørbart [Gut77].

La oss imidlertid etablere et lite resultat for basis-initialsemantikk som knytter konvergens direkte til konsistens:

Sats 2.34 *Betrakt en vilkårlig basis-initialsemantikk spesifisert ved en signatur Σ og en ligningsmengde E, slik at Σ er disjunkt delt i Σ^c og Σ^d av hhv. generatorer og definerte funksjonssymboler. Anta E delt disjunkt i E^c og E^d som spesifiserer hhv. basis-initialsemantikk på $\mathcal{G}_{\Sigma^c}$ og semantikk til de definerte funksjonssymboler.*

Anta videre at E er konvergent på en slik måte at enhver $u \in \mathcal{T}_{\Sigma^d \cup \Sigma^c}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i reduksjonsordningen $\overset{*}{\underset{E}{\rightarrow}}$ enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$. (M.a.o. $u \overset{*}{\underset{E}{\rightarrow}} v$). Da er E initielt konsistent.

Bevis: Anta nemlig $g_c \overset{*}{\underset{E}{\leftrightarrow}} g'_c$. Da har vi $g_c \overset{!}{\underset{E}{\rightarrow}} g_c! \overset{!}{\underset{E}{\leftarrow}} g'_c$. Men da må vi ha ved antagelsen om reduksjonsordningen, at $g_c \overset{!}{\underset{E^c}{\rightarrow}} g_c! \overset{!}{\underset{E^c}{\leftarrow}} g'_c$.
□

Det at E her er konvergent er selvfølgelig meget forskjellig fra at E^c og E^d hver for seg er konvergente (jfr. eksempel 30).

Kommentar:

En konsekvens av Gödels ufullstendighetsteorem [Göd31], [EN58] sier at et hvis en 1. ordens predikatkalkyle for $\mathcal{N}at^{+*}$ er konsistent, så kan ikke dennes konsistens bevises i kalkylen selv. Dette innebærer dessuten at konsistensen heller ikke kan bevises ved et *meta-matematisk* bevis, dersom sistnevnte kan *representeres av et bevis i kalkylen*. Dette utelukker ikke eksistensen av meta-matematiske bevis for konsistens av 1. ordens predikatkalkyler for $\mathcal{N}at^{+*}$, men antyder at et slikt (meta-matematisk) bevis for konsistens må følge bevisprinsipper som er *mer kompliserte* enn dem som kan nedfelles i kalkylen som skal vises konsistent. Disse bevisprinsippers *egne* konsistens kan følgelig være under mer tvil enn konsistensen til den opprinnelige objekt-kalkylen.

Men kan det tenkes at det finnes *enklere* bevisprinsipper? Dette var Hilberts håp. Han søkte å å finne *finistiske* bevisprinsipper som kunne etablere konsistens; dvs. bevisprinsipper som kun involverer referanse til endelige strukturer og et endelig antall operasjoner på strukturer. Et slikt finistisk bevis vil typisk være en uttømende demonstrasjon for «alle mulige instanser». Konsistensen til finistiske bevisprinsipper kan i denne sammenheng ansees som så godt som hevet over enhver tvil. Men ved Gödels resonnement kan et slikt finistisk bevis for konsistens av en 1. ordens predikatkalkyle for $\mathcal{N}at^{+*}$ som kan representeres i kalkylen ikke eksistere. Og på den annen side er det vanskelig å forestille seg et slikt finistisk bevis som *ikke* kan representeres i kalkylen.

Flere meta-matematiske bevis finnes for konsistens som ikke oppfyller Hilberts program. De er likevel nyttige og er på sett og vis det beste man har. Se [EN58] for en meget god forklaring til både Gödels ufullstendighetsteorem og for mer om de tingene vi har snakket om her.

Hvilken relevans har dette til vår diskusjon? Gödels ufullstendighetsteorem er knyttet til 1. ordens predikatlogikk. Ligningslogikk

(evt. styrket med induksjonsregler) har som sagt svakere uttrykkskraft, og som nevnt også i fotnote 10 på side 51, kan vi ikke umiddelbart overføre Gödels resultater til spesialtilfellet ligningslogikk. Men Gödels resultater kaster et bevisstgjørende lys over oppgaven å etablere konsistensen til formelle kalkyler. Akkurat hvor mye av dette lys som også kastes over konsistensproblemet for ligningslogiske systemer, er ukjent for meg.

2.5 Oppsummering

I dette kapitlet har vi ved hjelp av universell algebra definert begrepene *abstrakt* og *formell datatype*. Vi har definert hva vi mener med at en formell datatype er en *implementasjon* av en abstrakt datatype.

Vi har så sett på forskjellige måter å definere formelle datatyper på ved ulike typer *semantikk*. Semantikker bygges opp ved hjelp av ligninger og ligningslogikk. Vi har presentert *algebraisk spesifikasjon* som en grunnleggende måte å spesifisere semantikk på. Algebraisk spesifikasjon gir såkalt *initial*semantikk. Vi har også nevnt en grunnleggende *final*semantikk presentert i [Lys92]. Vi har innført *generaliserte* semantikker som spesifiseres relativt til *kjerne*semantikker. Dette gir en mulighet for modulær oppbygging av formelle datatyper fra enklere/andre formelle datatyper.

Vi har ikke gått videre inn på slik oppbygging, men det at en semantikk kan spesifiseres relativt til en kjernesemantikk er essensielt for diskusjonen i neste kapittel.

Vi har introdusert forskjellige *konsistens*begreper. Spesielt interessant er *kjernebevaring*. Vi har knyttet kjernebevaring til en *intensjon* ved våre generelle semantikker i forbindelse med implementasjon. Denne intensjon forfekter en modulær sammensetting av delsemantikker på en slik måte at sammensetningen bevarer delsemantikkene nøyaktig. Videre har vi vist at en degeneret form for generell finalsemantikk under konsistens gir et annet uttrykk for generell initialsemantikk, samt at begge disse under konsistens ivaretar nevnte intensjon. Sentralt i utviklingen av teorien omkring konsistens har vært det vi har kalt *separabel* semantikk. Denne skal dukke opp igjen i diskusjonen senere.

Konsistens og inkonsistens er syntaktiske egenskaper. Vi har sett at våre begreper om inkonsistens kan sees i et modell-teoretisk perspektiv ved ikke-eksistens av tolkninger. Vi har også sett at inkonsistens kan sees som *inkongruens*.

Formell resonnering — her utledning av ligninger — er interessant. Dette som et steg mot presis og muligens mekanisk programverifikasjon. Vi har nevnt og så vidt presentert noen eksisterende *resolusjonsmetoder* for basis-semantikker. Vi har også antydet at resolusjonsmetoder for våre generelle semantikker ikke umiddelbart kan oppnås ved sammensetting av eksisterende resolusjonsmetoder.

Avslutningsvis har vi såvidt sett på muligheter for å etablere konsistens og inkonsistens mekanisk.

Kapittel 3

Semantikkgivende syntaktiske funksjoner

I dette kapitlet introduserer vi *semantikkgiving ved syntaktiske funksjoner*.

Dette overfører vi til en ny måte å spesifisere semantikk på ved hjelp av ligningslogikk; nemlig det vi skal kalle *indirekte spesifikasjon*. Vi skal bruke indirekte spesifikasjon som atomær kjernesemantikk. Semantikk med en slik indirekte spesifisert kjerne, er en reell anvendelse av våre generaliserte semantikker fra kapittel 2; idet slik semantikk ikke er basis-semantikk.

Siden vi ikke har utviklet resolusjonsmetoder for generell semantikk, utgjør resolusjon av semantikk med en indirekte spesifisert kjerne således et problem. Imidlertid viser vi at semantikk med en indirekte spesifisert kjerne under visse omstendigheter kan *reduseres* til basis-semantikker, og er på den måten tilgjengelig for eksisterende resolusjonsmetoder.

Vi anvender også våre inkonsistensbegreper utviklet i kapittel 2. Vi går mere i detalj og viser hvordan inkonsistens kan oppdages. Problemet med *inkongruens* oppsøkes også.

Vi introduserer dessuten begrepet *kunstig inkonsistens*. Dette er inkonsistens innført av *hjelpefunksjoner* som begrepsmessig burde være skjult for logikken. Vi viser hvordan slik kunstig inkonsistens på en operasjonell måte kan elimineres fra ligningslogikken. Denne *skjuling* fra logikken av hjelpefunksjoner i formelle datatyper svarer på sett og vis til skjuling av interne hjelpeprosedyrer/funksjoner i moduldeklarasjoner på programmeringsspråk-nivå.

Vi ser også hvordan eliminering av kunstig inkonsistens kan gjøres på spesifikasjonsnivå.

Våre indirekte spesifikasjoner viser seg å ha et sterkt *operasjonelt* programaktig særpreg. Vi diskuterer hvordan indirekte spesifikasjoner kan *verifiseres*.

3.1 Syntaktiske funksjoner

En ***syntaktisk funksjon*** er en funksjon hvis domene og kodomene er symbolmengder. I lys av abstrakte maskiner, kan konvergente omskrivningssystemer sies å *beregne* syntaktiske funksjoner: For et omskrivningssystem $\langle \mathcal{T}_\Sigma(\mathcal{V}), R\rangle$ konvergent på en term-mengde $\mathcal{T}$, betrakt relasjonen $\frac{\mathcal{T}\downarrow}{R} = \{\langle t, t!R\rangle \mid t \in \mathcal{T}\}$. Relasjonen $\frac{\mathcal{T}\downarrow}{R}$ er en funksjon i $(\mathcal{T} \rightarrow \mathcal{T}_\Sigma(\mathcal{V}))$. En syntaktisk funksjon *synt* på en term-mengde $\mathcal{T}$ er da her ***termomskrivningsberegnbar*** eller bare ***omskrivningsberegnbar***, hvis det fins et $\mathcal{T}$-konvergent omskrivningssystem R slik at

$$\frac{\mathcal{T}\downarrow}{R} = synt \tag{3.1}$$

Dersom $\overset{*}{\underset{R}{\leftrightarrow}}_{\mathcal{T}} = \simeq$ for en kongruensrelasjon $\simeq$ på $\mathcal{T}$ ($\mathcal{T}$ antas da å være en term-algebra), har vi videre

$$s \simeq t \;\Leftrightarrow\; s \overset{*}{\underset{R}{\leftrightarrow}} t \;\Leftrightarrow\; s!R = t!R$$

for alle $s, t \in \mathcal{T}$. M.a.o. verdiene av s og t under funksjonen $\overset{\mathcal{T}!}{\underset{R}{\rightarrow}}$ er identiske hvis og bare hvis s og t er i samme $\simeq$-kongruensklasse. Vi kan derfor si at $\overset{\mathcal{T}!}{\underset{R}{\rightarrow}}$ spesifiserer kongruensrelasjonen $\simeq$ ved at

$$\simeq \;=\; \overset{\mathcal{T}!}{\underset{R}{\rightarrow}} \circ \overset{\mathcal{T}!}{\underset{R}{\leftarrow}}$$

Generelt kan da *vilkårlige* syntaktiske funksjoner (ikke bare de som er omskrivningsberegnbare) brukes til å spesifisere kongruensrelasjoner på termer ved følgende prinsipp:

$$s \simeq t \;\Leftrightarrow\; synt(s) = synt(t) \tag{3.2}$$

En nødvendig (og tilstrekkelig) betingelse for at en syntaktisk funksjon skal kunne bestemme en kongruensrelasjon på denne måten, er monotonitet mhp. kontekstapplikasjon:

$$synt(s) = synt(t) \;\Rightarrow\; synt(c[s]) = synt(c[t]) \tag{3.3}$$

for alle kontekster c og termer s, t i domenet til $\simeq$. Vi skal kalle syntaktiske funksjoner som spesifiserer kongruensrelasjoner på termer ved prinsipp (3.2) ***semantikkgivende***.

Alle syntaktiske funksjoner på term-algebraer som er omskrivningsberegnbare, er såkalte *kanonisk-representant funksjoner.*

Definisjon 3.1 *For en Σ-algebra A og en kongruensrelasjon $\simeq$ på A, la $\delta \in (A \to A)$ være slik at det for alle $q \in A/\simeq$ finnes en unik $b \in q$ slik at $\delta(a) = b$ for alle $a \in q$. Vi kaller δ en **kanonisk-representant funksjon** for $\simeq$, og elementene i bildet $\delta(A)$ **kanoniske representanter** for $\simeq$ bestemt av δ.*

Eksempel 32 For

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0 : int}, \\ \mathsf{succ : int \longrightarrow int}, \\ \mathsf{pred : int \longrightarrow int} \end{array} \right\}$$

betrakt «standardtolkningen» $\mathcal{I}nt$ fra eksempel 1 side 14. La $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ være kongruensrelasjonen indusert av den unike surjektive homomorfi $\phi^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$. Det konvergente omskrivningssystemet utgjort av regelmengden:

$$R_{\mathcal{I}nt} = \left\{ \begin{array}{l} \mathsf{succ(pred(x))} \rightarrow \mathsf{x} \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

beregner følgende kanonisk-representant funksjon $\delta_{\mathcal{I}nt}$ for $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$: La n_g være differansen mellom antall succ'er og antall pred'er i en term $g \in \mathcal{G}_{\mathsf{Int}}$. La $\delta_{\mathcal{I}nt}$ være slik at for vilkårlig $g \in \mathcal{G}_{\mathsf{Int}}$ så er $\delta_{\mathcal{I}nt}(g)$ termen $\mathsf{succ}^{n_g}(\mathsf{0})$ dersom $n_g > 0$ og termen $\mathsf{pred}^{-n_g}(\mathsf{0})$ ellers.
○

En annen måte å karakterisere kanonisk-representant funksjoner på, er ved funksjoner hvis kodomene består kun av fikspunkter. En slik ***fikspunktfunksjon*** kan karakteriseres ved egenskapen

$$f(f(a)) = f(a) \quad \text{for alle } a \text{ i domenet til } f \tag{3.4}$$

Sats 3.1 *La synt være en syntaktisk funksjon på en $\mathcal{G}_\Sigma$, og la $\simeq$ være en kongruensrelasjon på $\mathcal{G}_\Sigma$. Vi har:*

$$\begin{array}{c} \textit{synt er en kanonisk-representant funksjon for } \simeq \\ \Updownarrow \\ \textit{synt er en fikspunktfunksjon, og synt er semantikkgivende og spesifiserer } \simeq \end{array}$$

Bevis: Vi minner om (2.6) på side 23.

Anta *synt* er en kanonisk-representant funksjon for $\simeq$. For hver kongruensklasse $q \in \mathcal{G}_\Sigma/\simeq$ fins da en unik $g^* \in q$ slik at $synt(g) = g^*$ for alle $g \in q$. M.a.o. har vi ved (2.6) for alle $g, g' \in \mathcal{G}_\Sigma$:

$$synt(g) = g^* = synt(g') \;\Leftrightarrow\; g \simeq g'$$

så *synt* er semantikkgivende og spesifiserer $\simeq$. Siden *synt* nå er semantikkgivende har vi videre at $synt(g^*) = synt(g)$ for $g^* = synt(g)$, siden g^* er i samme kongruensklasse som g. Altså har vi for alle $g, g' \in \mathcal{G}_\Sigma$:

$$synt(synt(g)) = synt(g)$$

så *synt* er en fikspunktfunksjon.

Anta *synt* er en fikspunktfunksjon og at *synt* er semantikkgivende og spesifiserer $\simeq$ ved prinsippet (3.2). Ved egenskapen (3.4), har vi for vilkårlige $g \in \mathcal{G}_\Sigma$, $synt(synt(g)) = synt(g)$ som da gir $synt(g) \simeq g$. Ved (2.6) er altså de to elementene $synt(g)$ og g alltid i samme kongruensklasse. Videre må alle g, g' i en klasse q ha samme (unike) verdi under *synt*, ved at $synt(g) = synt(g')$. Altså er *synt* en kanonisk-representant funksjon for $\simeq$.

□

Det er lett å se at enhver omskrivningsberegnbar syntaktisk funksjon på (bæremengden til) en term-algebra, er en kanonisk-representant funksjon. Men ikke alle syntaktiske kanonisk-representant funksjoner er omskrivningsberegnbare, siden omskrivningsberegnbarhet fordrer konvergens:

Eksempel 33 Betrakt basis-initialsemantikken $\overset{*}{\underset{E_{\mathsf{SetNat}}}{\longleftrightarrow}}\mathcal{G}_{\mathsf{SetNat}}$ spesifisert av signaturen SetNat og ligningsmengden

$$E_{\mathsf{SetNat}} = \left\{ \begin{array}{l} \mathsf{add(add(s,x),x) = add(s,x)}, \\ \mathsf{add(add(s,x),y) = add(add(s,y),x)} \end{array} \right\}$$

fra eksempel 27 side 50. La δ_{SetNat} være en kanonisk-representant funksjon for $\overset{*}{\underset{E_{\mathsf{SetNat}}}{\longleftrightarrow}}\mathcal{G}_{\mathsf{SetNat}}$ slik at δ_{SetNat} gir «sorterte» termer uten «repetisjon» og slik at δ_{SetNat} er identitet på termer av type nat. F.eks. er

$$\delta_{\mathsf{SetNat}}(\mathsf{add(add(add(\emptyset,succ(0)),0),succ(0))}) = \mathsf{add(add(\emptyset,0),succ(0))}$$

Det finnes intet omskrivningssystem som beregner δ_{SetNat} ved (3.1). (Vi har definert et omskrivningssystem til å ha en endelig regelmengde. Vi ser også bort fra mer avanserte former for omskrivningssystemer.)

○

Alle kanonisk-representant funksjoner for kongruensrelasjoner på termer er ved sats 3.1 semantikkgivende. Imidlertid behøver ikke en semantikkgivende funksjon være en kanonisk-representant funksjon:

Eksempel 34 La signaturen Int, algebraen $\mathcal{I}nt$ og kongruensrelasjonen $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ være som i eksempel 32. Vi definerer en semantikkgivende funksjon som definisjonsmessig bare avviker *litt* fra $\delta_{\mathcal{I}nt}$ i eksempel 32: La igjen n_g være differansen

mellom antall succ'er og antall pred'er i en term $g \in \mathcal{G}_{\mathsf{Int}}$. La imidlertid $\gamma_{\mathcal{I}nt}$ være slik at for vilkårlig $g \in \mathcal{G}_{\mathsf{Int}}$ så er $\gamma_{\mathcal{I}nt}(g)$ termen $\mathsf{succ}^{n_g+1}(\mathsf{0})$ dersom $n_g > 0$ og termen $\mathsf{pred}^{-n_g+1}(\mathsf{0})$ ellers. F.eks. er $\gamma_{\mathcal{I}nt}(\mathsf{succ}(\mathsf{0})) = \mathsf{succ}(\mathsf{succ}(\mathsf{0}))$. Det er lett å overbevise seg om at

$$g \simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}} g' \;\Leftrightarrow\; \gamma_{\mathcal{I}nt}(g) = \gamma_{\mathcal{I}nt}(g')$$

for alle $g, g' \in \mathcal{G}_{\mathsf{Int}}$, så $\gamma_{\mathcal{I}nt}$ er semantikkgivende. Legg merke til at $\gamma_{\mathcal{I}nt}(\gamma_{\mathcal{I}nt}(g)) \neq \gamma_{\mathcal{I}nt}(g)$ for alle g, så $\gamma_{\mathcal{I}nt}$ er ikke en fikspunktfunksjon, og da ved sats 3.1 ingen kanonisk-representant funksjon.
○

Vi skal nå generalisere begrepet kanonisk-representant funksjon som følger:

Definisjon 3.2 *For en Σ-algebra A og en kongruensrelasjon $\simeq$ på A, la $\gamma \in (A \to A)$ være slik at det for alle $q \in A/\simeq$ finnes en unik $b \in A$ slik at $\gamma(a) = b$ for alle $a \in q$, og slik at for $a, a' \in A$ og $q, q' \in A/\simeq$*

$$a \in q;\; a' \in q';\; q \neq q' \quad \Rightarrow \quad \gamma(a) \neq \gamma(a')$$

*Vi kaller γ en **klasserepresentant funksjon** for $\simeq$, og elementene i bildet $\gamma(A)$ **klasserepresentanter** for $\simeq$ bestemt av γ. Vi definerer **γ-reduksjonen** av $A/\simeq$, skrevet A/γ som Σ-algebraen med $\gamma(A)$ som bæremengde og med funksjonsmengde bestående av en funksjon $f_{A/\gamma}$ for hver f i Σ, slik at*

$$f_{A/\gamma}(\gamma(a_1), \ldots, \gamma(a_n)) = \gamma(f_A(a_1, \ldots, a_n))$$

for tolkningen f_A av f i A og vilkårlige $a_1, \ldots, a_n \in A$.

Dersom $\gamma(a) \in q \;\Leftrightarrow\; a \in q$ for alle $a \in A, q \in A/\simeq$ i definisjon 3.2, har vi spesialtilfellet kanonisk-representant funksjon.

Sats 3.2 *La synt være en syntaktisk funksjon på en $\mathcal{G}_\Sigma$, og la $\simeq$ være en kongruensrelasjon på $\mathcal{G}_\Sigma$. Vi har:*

synt er en semantikkgivende funksjon som spesifiserer $\simeq$
$$\Updownarrow$$
synt er en klasserepresentant-funksjon for $\simeq$

Bevis: Vi minner igjen om (2.6) på side 23.

Anta *synt* spesifiserer $\simeq$ ved prinsipp (3.2). Anta for vilkårlige $g, g' \in \mathcal{G}_\Sigma$ at g, g' er i samme $q \in \mathcal{G}_\Sigma/\simeq$. Ved (2.6) har vi da $g \simeq g'$, og da har vi $synt(g) = synt(g')$. Altså finnes en unik $g^* \in \mathcal{G}_\Sigma$ slik at $synt(g) = g^*$ for alle $g \in q$. Videre har vi ved (3.2) $g \not\simeq g' \;\Rightarrow\; synt(g) \neq synt(g')$, så ved (2.6) har vi direkte

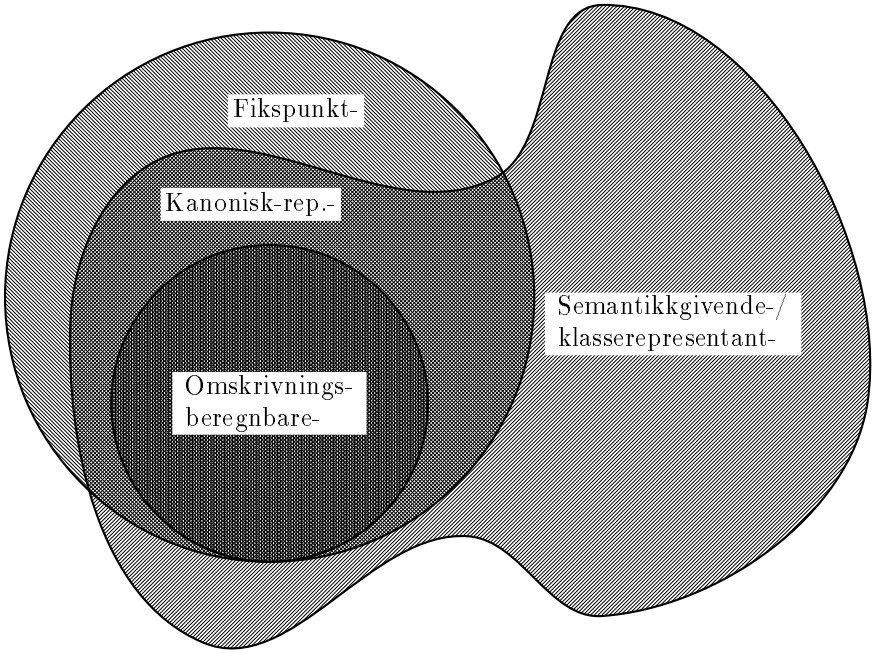
$$g \in q, g' \in g', q \neq q' \quad \Rightarrow \quad synt(g) \neq synt(g')$$

for $q, q' \in \mathcal{G}_\Sigma/\simeq$; så *synt* er en klasserepresentant-funksjon for $\simeq$.

Anta *synt* er en klasserepresentant-funksjon for $\simeq$. For vilkårlige $g, g' \in \mathcal{G}_\Sigma$, hvis g, g' er i samme $q \in \mathcal{G}_\Sigma/\simeq$, så finnes en unik $g^* \in \mathcal{G}_\Sigma$ slik at $synt(g) = g^* = synt(g')$. Hvis g, g' ikke er i samme $q \in \mathcal{G}_\Sigma/\simeq$, har vi derimot $synt(g) \neq synt(g')$. Ved (2.6) er da ialt *synt* en semantikkgivende funksjon som spesifiserer $\simeq$ ved prinsipp (3.2).
□

Litt av landskapet oppsummeres i figur 3.1.

Blant klasserepresentanter for en kongruensrelasjon, er de kanoniske representanter spesielle. De er fikspunkter til kanonisk-representant funksjoner, og da



Figur 3.1: Vår oppdeling av syntaktiske funksjoner. Av særlig interesse for oss er snittet mellom semantikkgivende-/klasserepresentantfunksjoner og fikspunktfunksjoner; nemlig kanonisk-representant funksjoner.

i noen henseende lettere å håndtere mekanisk. Men den egenskapen vi ønsker å fremheve nå, har med menneskelig håndtering av symbolikk å gjøre. Kanoniske representanter har den viktige egenskapen at de selv er i klassen de representerer. «De står for det de representerer.» Denne egenskapen må sies å være sentral for enhver naturlig representant av hva som helst. I vår kontekst kan vi i tillegg komme med to ytterlige krav til ***naturlige representanter*** for kongruensklasser: For en kanonisk-representant funksjon δ vil vi at de kanoniske representanter bestemt av δ er:

1. generator-termer
2. i tilfellet mange-til-en generatorunivers, de «enkleste» generator-termer i en eller annen praktisk forstand.[1]

I vårt oppsett ivaretas symbol-tolking av grunntermtolker. Tolkingen er det «semantiske sprang» fra syntaks til semantikk, og det er essensielt at tolkingen er umiddelbar eller direkte. Det ville f.eks. være ufornuftig om en lomme-kalkulator eller et kalkulator-program gitt symbolsekvensen 1+1, ga svaret $\mathsf{cos}(0)+\mathsf{sin}(\frac{\pi}{2})$, eller om den symbolske representasjonen for tallet 1 var $\mathsf{succ}(\mathsf{succ}(0))$ (sett bort fra at *unær* representasjon i seg selv er ufornuftig for nær sagt alt utenom formelle betraktninger).

Kommentar:

Slike naturlige klasserepresentanter kan brukes til å supplere begrepet ***konstruktiv*** funksjons-spesifikasjon (se tekstavsnittet før de-

[1] Kvalifikatoren 'kanonisk' henspeiler på noe som er et mål eller en målestokk for andre (ting i en klasse); en beste versjon.

finisjon 2.2 side 19). For at en formell datatype skal være implementasjonen (avsnitt 2.3.1, side 22) av en algebra A, kreves algebraisk spesifikasjon av alle funksjoner i A, samt at riktig semantikk må gis til generatortermer. Det er naturlig å forvente i tillegg at de algebraiske funksjons-spesifikasjonene er konvergente. Da har vi ved termomskrivning et abstrakt *deterministisk* program (algoritme) med hvilket verdien av et element under en funksjon kan «regnes» ut. Dessuten bør nå en slik funksjonsverdi presenteres på en enkleste form, i form av en naurlig representant. (Disse krav må ihvertfall innfris i det endelige (imperative) program.)

Anta da en kanonisk-representant funksjon δ for en semantikk $\simeq$ på en $\mathcal{G}_\Sigma$ som tilfredstiller punktene 1 og 2 over. Våre tilleggskrav for en formell datatype er da i lys av disse punktene, med andre ord:

> Gitt en vilkårlig term $g \in \mathcal{G}_\Sigma$ ønsker vi å beregne $\delta(g)$ (mekanisk).

Punktene 1 og 2 kan realiseres helt for tilfellet basis-initialsemantikk, dersom ligningsmengden er konvergent og beregner δ ved prinsipp (3.1) på side 63. For generelle semantikker som beskrevet i avsnitt 2.3, kan ihvertfall punkt 1 innfris.

*

Vi skal bruke semantikkgivende syntaktiske funksjoner til å definere atomær semantikk. Siden konvergente omskrivningssystemer kan sies å beregne kanonisk-representant funksjoner, har vi gjort dette på en implisitt måte tidligere; ved basis-initialsemantikk, gitt at den aktuelle ligningsmengden er konvergent. Vi skal imidlertid studere *algebraiske beskrivelser* av de semantikkgivende syntaktiske funksjoner. Disse i sin tur skal brukes til å spesifisere semantikk.

Men før vi gjør dette skal vi, som et sidespor til vår diskusjon om syntaktiske funksjoner som semantikkgivende, se på en annen anvendelse av klasserepresentant-funksjoner.

3.2 Funksjonsspesifikasjon over kanoniske representanter

Mange-til-en generatorunivers er litt problematiske. Funksjonsspesifikasjoner over mange-til-en generatorunivers kan gi inkonsistens, og i allefall må generatortermene på en eller annen måte gis semantikk. Vi har i avsnitt 2.3 i kapittel 2 sett et par måter å gjøre sistnevnte på, og vi skal senere også bruke algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner til å gi generatortermer semantikk.

I dette avsnittet ser vi kort på en-til-en «generatorunivers» bestående av klasserepresentanter. Vi ser på funksjonsspesifikasjoner *over klasserepresentanter*. 'Mange-til-en'-problematikken løses således ikke på spesifikasjonsnivå ved f.eks. basis-initialsemantikk, men snarere «bak kulissene» — før spesifikasjon tar til.

Vi fører diskusjonen her for kanoniske representanter. Det ville bryte sterkt mot symbolers brukervennlighet å prøve å spesifisere funksjoner over andre klasserepresentanter enn kanoniske.

3.2.1 Mekanisk generering av kanoniske representanter

Vi kan tenke oss at kanoniske representanter for en semantikk skal genereres «bak kulissene» av et mekanisk system på et «lavere» nivå enn det vi programmerer våre implementasjoner av abstrakte datatyper på. Hvordan genereres så en term-mengde bestående av kanoniske representanter mekanisk? Vi ser på to måter.

For den første måten, betrakt en formell datatype $\mathcal{G}_\Sigma/\simeq$ for en signatur Σ. La δ være en kanonisk-representant funksjon for $\simeq$. Betrakt så δ-reduksjonen $\mathcal{G}_\Sigma/\delta$ av $\mathcal{G}_\Sigma/\simeq$. Det er nå $\mathcal{G}_\Sigma/\delta$ som tar rollen $\mathcal{G}_\Sigma$ ellers har som utgangspunkt for «programmering». Men nå har funksjonene i «term»-algebraen «pre-programmert» semantikk og er langt mer avanserte enn funksjoner i en term-algebra. Bæremengden til $\mathcal{G}_\Sigma/\delta$ er mekanisk genererbar dersom δ f.eks. er beregnbar ved et konvergent omskrivningssystem. Se figur 3.2. Denne første måten er beslektet med begrepet *semantisk subtype* (se f.eks. [Dah92]).

Et alternativ til denne måten å generere en bæremengde bestående av kanoniske representanter på, er ved *syntaktiske subtyper*.

For typer T og U, la T være en ***subtype*** av U, skrevet $T \preceq U$. En term av type T er en term av type U, mens en term av type U kan være, men er ikke nødvendigvis, en term av type T. Dette bestemmer oppbyggingen av termer beskrevet i avsnitt 2.2 i lys av subtyper. En tolk av typer Φ_τ skal tilfredstille

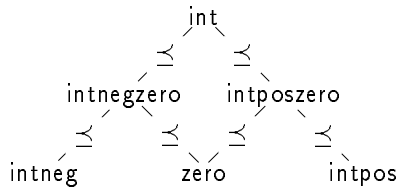
$$\Phi_\tau(T) \subseteq \Phi_\tau(U)$$

for $T \preceq U$. I noen tilfeller kan så kanoniske representanter «tvinges fram» syntaktisk:

Eksempel 35 Betrakt følgende signatur:

$$\mathsf{Int}' = \left\{ \begin{array}{l} \mathsf{0 : zero}, \\ \mathsf{succ : intposzero \longrightarrow intpos}, \\ \mathsf{pred : intnegzero \longrightarrow intneg} \end{array} \right\}$$

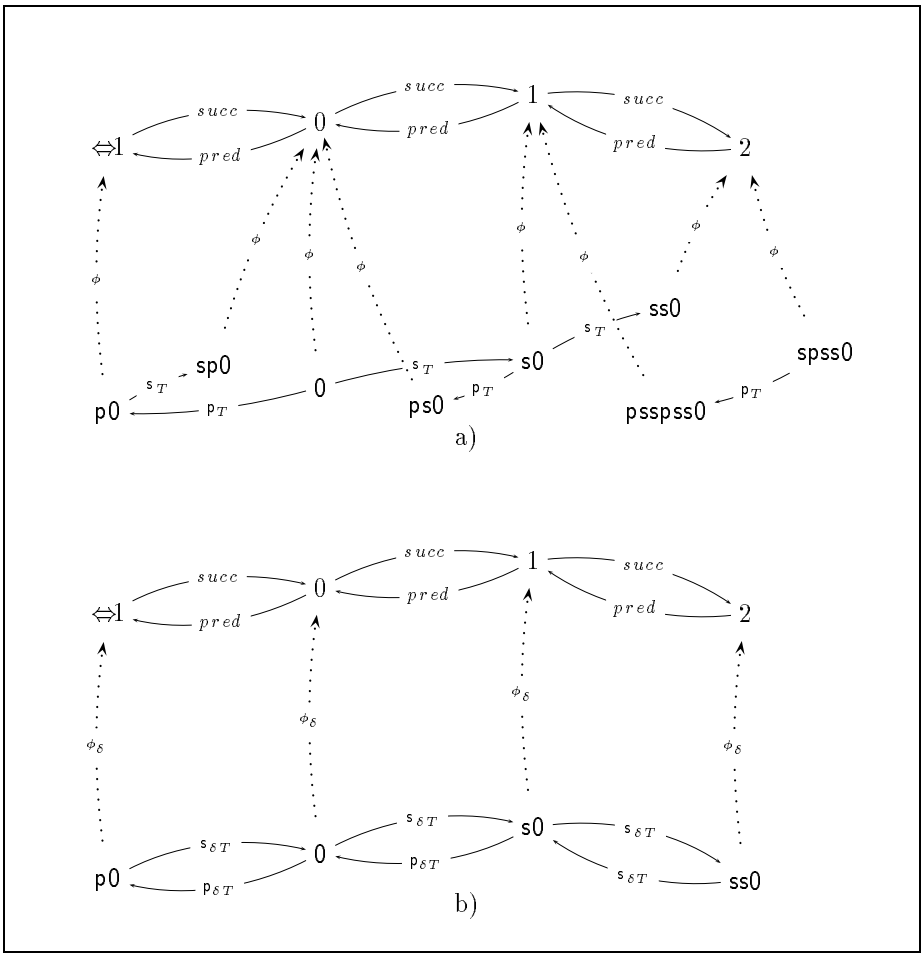
der subtyperelasjonen er som følger (den *maksimale supertypen* int er tatt med for estetikkens skyld):



Fra Int′ kan vi nå eksempelvis bygge termene succ(succ(0)) og pred(pred(0)), men ikke termen succ(pred(0)), siden pred(0) ikke er av type intposzero eller intpos. Det er lett å se at grunnterm-algebraen $\mathcal{G}_{\mathsf{Int}'}$ er isomorf med $\delta_{\mathcal{I}nt}$-reduksjonen til $\mathcal{G}_{\mathsf{Int}}/\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$, for $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ og kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32 side 64.
○

Ved syntaktiske subtyper overlates genereringen av kanoniske representanter til typingsalgoritmen som sørger for termoppbygging slik at uekte typegale «termer» ikke kan genereres. En slik typingsalgoritme tenkes å underligge enhver formell datatype.

Det er imidlertid ikke så lett i det generelle tilfelle å finne en signatur utfra hvilken ønskede kanoniske representanter genereres (tvinges frem). Hvordan skulle en slik signatur se ut for tilfellet 'mengder av naturlige tall'?



Figur 3.2: «Uprogrammert» og «pre-programmert» maskin. I nedre del av a) sees et utsnitt av (ikke)-strukturen i grunnterm-algebraen $\mathcal{G}_{\mathsf{Int}}$ for $\mathsf{Int} = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}\}$. Symbolene succ og pred er forkortet til s og p hhv., og parenteser er utelatt i termer. Tolkningene av succ og pred i $\mathcal{G}_{\mathsf{Int}}$ er betegnet s_T og p_T. I øvre del av a) sees det korresponderende utsnitt av algebraen $\mathcal{I}nt$. Den unike homomorfien fra $\mathcal{G}_{\mathsf{Int}}$ til $\mathcal{I}nt$ er betegnet ϕ.

I nedre del av b) sees til sammenligning et utsnitt av strukturen i $\delta_{\mathcal{I}nt}$-reduksjonen til $\mathcal{G}_{\mathsf{Int}}/\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$, for $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ og kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ for $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ fra eksempel 32 side 64. Tolkningene av succ og pred er her betegnet $\mathsf{s}_{\delta T}$ og $\mathsf{p}_{\delta T}$. (Vi minner om definisjon 3.2 side 66 for definisjonene av $\mathsf{s}_{\delta T}$ og $\mathsf{p}_{\delta T}$.) Funksjonene i $\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$ er avanserte i forhold til funksjonene i $\mathcal{G}_{\mathsf{Int}}$. F.eks. er $\mathsf{s}_{\delta T}(\mathsf{p0}) = \delta_{\mathcal{I}nt}(\mathsf{sp0})$. $\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$ er mekanisk genererbar hvis $\delta_{\mathcal{I}nt}$ er beregnbar. Den unike (lett å vise) homomorfien fra $\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$ til $\mathcal{I}nt$ er betegnet ϕ_δ.

3.2.2 Eksempler på funksjonsspesifikasjon

Vi skal gi noen eksempler på funksjonsspesifikasjon over kanoniske representanter. Først presenteres for referanse et par eksempler på funksjonsspesifikasjon over *mange-til-en generatorunivers*:

Eksempel 36 For den abstrakte datatypen 'mengder av naturlige tall' har vi i en passende formell datatype generatorene:

$$\mathsf{SetNat} = \left\{ \begin{array}{l} \mathsf{0 : nat}, \\ \mathsf{succ : nat \longrightarrow nat}, \\ \mathsf{\emptyset : setnat}, \\ \mathsf{add : setnat \times nat \longrightarrow setnat} \end{array} \right\}$$

For 'element-funksjonen' $\in$ på mengder av naturlige tall spesifiserer vi så:

$$E_{\in} = \left\{ \begin{array}{l} \mathsf{x \in \emptyset = false}, \\ \mathsf{x \in add(s,y) = eq(x,y) \vee x \in s} \end{array} \right\}$$

for

$$E_{\vee} = \left\{ \begin{array}{l} \mathsf{true \vee u = true}, \\ \mathsf{false \vee u = u} \end{array} \right\}$$

og

$$E_{=_{\mathcal{N}at}} = \left\{ \begin{array}{l} \mathsf{eq(0,0) = true}, \\ \mathsf{eq(0,succ(x)) = false}, \\ \mathsf{eq(succ(x),0) = false}, \\ \mathsf{eq(succ(x),succ(y)) = eq(x,y)} \end{array} \right\}$$

○

Eksempel 37 For addisjon på hele tall spesifiserer vi

$$E_{+_{\mathbb{Z}}} = \left\{ \begin{array}{l} \mathsf{x+0 = x}, \\ \mathsf{x+succ(y) = succ(x+y)}, \\ \mathsf{x+pred(y) = pred(x+y)} \end{array} \right\}$$

○

Her følger så noen eksempler på funksjonsspesifikasjon over kanoniske representanter:

Eksempel 38 For kanoniske representanter som definert av $\delta_{\mathcal{I}nt}$ fra eksempel 32, er spesifikasjonen av addisjon på hele tall $E_{+_{\mathbb{Z}}}$ i eksempel 37 ikke *kanonisk-representant-bevarende*, i den forstand at det finnes kanoniske representanter g og g', men g'' ikke kanonisk slik at $g{+}g' \stackrel{*}{\underset{E_{+_{\mathbb{Z}}}}{\longrightarrow}} g''$. Eksempelvis har vi

$$\mathsf{succ(0)+pred(0)} \stackrel{*}{\underset{E_{+_{\mathbb{Z}}}}{\longrightarrow}} \mathsf{pred(succ(0))}$$

En kanonisk-representant-bevarende spesifikasjon er følgende:

$$E'_{+_{\mathbb{Z}}} = \left\{ \begin{array}{l} \mathsf{x+0 = x}, \\ \mathsf{0+x = x}, \\ \mathsf{succ(x)+succ(y) = x+succ(succ(y))}, \\ \mathsf{pred(x)+pred(y) = x+pred(pred(y))}, \\ \mathsf{succ(x)+pred(y) = x+y}, \\ \mathsf{pred(x)+succ(y) = x+y} \end{array} \right\}$$

○

Eksempel 39 For kanoniske representanter som definert av δ_{SetNat} i eksempel 33 side 65, er det mulig å spesifisere 'element-funksjonen' $\in$ på mengder av naturlige tall beregningsmessig mer effektivt enn i eksempel 36, ved å benytte seg av strukturen til de kanoniske representanter:

$$E'_{\in} = \left\{ \begin{array}{lcl} \mathsf{x} \in \emptyset & = & \mathsf{false}, \\ \mathsf{x} \in \mathsf{add(s,y)} & = & \mathsf{eq(x,y)} \vee \\ & & \mathsf{if\ x} < \mathsf{y\ then\ x} \in \mathsf{s\ else\ false} \end{array} \right\}$$

for

$$E_{ite} = \left\{ \begin{array}{lcl} \mathsf{if\ true\ then\ b\ else\ b'} & = & \mathsf{b}, \\ \mathsf{if\ false\ then\ b\ else\ b'} & = & \mathsf{b'} \end{array} \right\}$$

og

$$E_{<_{\mathcal{N}at}} = \left\{ \begin{array}{l} 0 < 0 \ = \ \mathsf{false}, \\ 0 < \mathsf{succ(x)} \ = \ \mathsf{true}, \\ \mathsf{succ(x)} < 0 \ = \ \mathsf{false}, \\ \mathsf{succ(x)} < \mathsf{succ(y)} \ = \ \mathsf{x} < \mathsf{y} \end{array} \right\}$$

Effektiviseringen svarer til det å prøve å finne et gitt element i en sortert liste framfor å prøve å finne elementet i en usortert liste. Legg merke til at fokuseringen her på term-oppbygging til kanoniske representanter, trekker oss noe ut av abstraksjonsnivået vi ønsker å oppnå i formelle datatyper og formell resonnering.
○

Eksempel 40 Vi kan spesifisere 'mindre-enn' relasjonen (funksjonsvarianten av denne) på hele tall over kanoniske representanter som definert av $\delta_{\mathcal{I}nt}$ fra eksempel 32, som følger:

$$E_{<_{\mathcal{I}nt}} = \left\{ \begin{array}{l} 0 < 0 \ = \ \mathsf{false}, \\ 0 < \mathsf{succ(x)} \ = \ \mathsf{true}, \\ \mathsf{succ(x)} < 0 \ = \ \mathsf{false}, \\ \mathsf{succ(x)} < \mathsf{succ(y)} \ = \ \mathsf{x} < \mathsf{y}, \\ 0 < \mathsf{pred(x)} \ = \ \mathsf{false}, \\ \mathsf{pred(x)} < 0 \ = \ \mathsf{true}, \\ \mathsf{pred(x)} < \mathsf{pred(y)} \ = \ \mathsf{x} < \mathsf{y} \end{array} \right\}$$

○

Funksjonsspesifikasjoner som «utnytter» egenskaper ved kanoniske representanter, vil ofte ikke fungere korrekt i det korresponderende mange-til-en universet. Dette er ikke uventet, bl.a. siden mange-til-en univers er en grunnleggende kilde til inkonsistens.

Eksempel 41 Betrakt mange-til-en generatoruniverset over signaturen

$$\mathsf{Int} = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}\}.$$

Vi spesifiserer basis-initialsemantisk

$$E_{\mathcal{I}nt_c} = \left\{ \begin{array}{l} \mathsf{succ(pred(x))} = \mathsf{x}, \\ \mathsf{pred(succ(x))} = \mathsf{x} \end{array} \right\}$$

Relativt til den frie semantikk på $\{\mathsf{true}, \mathsf{false}\}$ gir spesifikasjonen $E_{<_{\mathcal{I}nt}} \cup E_{\mathcal{I}nt_c}$ for $E_{<_{\mathcal{I}nt}}$ fra eksempel 40, initial inkonsistens ved

$$\mathsf{true} \simeq^{\alpha} \mathsf{pred(0)} < 0 \simeq^{\alpha} \mathsf{succ(pred(pred(0)))} < 0 \simeq^{\alpha} \mathsf{false}$$

Således er ligninger som $0 < \mathsf{succ}(x) = \mathsf{true}$ og $\mathsf{succ}(x) < 0 = \mathsf{false}$, ikke «riktige» for x ikke kanonisk, men kan dersom x er kanonisk, effektivisere beregningen av $<$ som i eksempel 40.
○

At funksjonsspesifikasjoner således fungerer korrekt, men også at en funksjonsspesifikasjon er kanonisk-representant-bevarende, forutsetter dessuten generelt at enhver subterm av en kanonisk representant er en kanonisk representant. Denne ***subterm-egenskap*** for kanoniske representanter kan sees som en egenskap ved den aktuelle kanonisk-representant funksjon. Vi merker oss følgende:

Observasjon 3.3 *Enhver omskrivningsberegnbar syntaktisk funksjon har denne subterm-egenskapen. Anta nemlig det motsatte — at det finnes en $c[g]$ som er en kanonisk representant bestemt av en kanonisk-representant funksjon beregnbar av et omskrivningssystem R, men at g ikke er en kanonisk representant. Siden $c[g]$ er en kanonisk representant, har vi $c[g] = c[g]!R$. Men siden g ikke er en kanonisk representant har vi $g \xrightarrow[R]{+} g'$ for en eller annen g', og da kan ikke $c[g] = c[g]!R$. Dette er en motsigelse.*

Dersom vi velger å beskrive et en-til-en kanonisk-representant term-univers ved en semantisk subtype, kan altså subterm-egenskapen garanteres dersom den aktuelle kanonisk-representant funksjon er omskrivningsberegnbar.

Merk at ikke alle kanonisk-representant funksjoner som har subterm-egenskapen for kanoniske representanter, er omskrivningsberegnbare. Se f.eks. eksempel 33 på side 65. For å supplere til figur 3.1 på side 67, har vi da

Omskrivningsberegnbare $\subset$ subterm-egenskap- $\subset$ kanonisk-representant-

Dersom vi velger å beskrive et en-til-en kanonisk-representant term-univers ved syntaktiske subtyper, er det ved et induktivt argument lett å se at dette termuniverset har subterm-egenskapen for kanoniske representanter.

3.2.3 Utledning og resolusjon

Uten å gå nærmere inn på det her, kan det tenkes at eksisterende resolusjonsmetoder kan modifiseres til å håndtere resolusjon i formelle datatyper med en-til-en kanonisk-representant term-univers.

Vi har sett to måter å beskrive et en-til-en kanonisk-representant term-univers på: Som tolkningen av en semantisk subtype, og ved hjelp av syntaktiske subtyper. Vi skal her bare peke på at det ikke er vilkårlig hvilke av disse vi velger når vi snakker om tilfredsstillbarhet, grunnredusibilitet og kritiske par i forbindelse med resolusjonsmetoder.

La eksempelvis $\sigma \in \mathcal{H}om^{\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}}_{\mathcal{T}_{\mathsf{Int}}(\mathcal{V})}$ være slik at $\sigma(x) = \mathsf{pred}(0)$. Skjeler vi til eksempel 40, får vi ved definisjon av funksjonene i $\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$ (vi minner om definisjonen av homomorfi side 14 og definisjonen av klasserepresentant funksjoner 3.2 side 66)

$$(0 < \mathsf{succ}(x))\sigma = 0 < \delta_{\mathcal{I}nt}(\mathsf{succ}(\mathsf{pred}(0))) = 0 < 0$$

Følgelig er påstanden

$$\forall \sigma \in \mathcal{H}om^{\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}}_{\mathcal{T}_{\mathsf{Int}}(\mathcal{V})} \mid \mathcal{I}nt \models (0 < \mathsf{succ}(x))\sigma = \mathsf{true}$$

gal. Betrakt deriomt Int' fra eksempel 35 på side 69. Påstanden

$$\forall \tau \in \mathcal{H}om^{\mathcal{G}_{\mathsf{Int}'}}_{\mathcal{T}_{\mathsf{Int}'}(\mathcal{V})} \mid \mathcal{I}nt \models (0 < \mathsf{succ}(x))\tau = \mathsf{true}$$

er da riktig. (F.eks. er funksjonen τ slik at $\tau(x) = \mathsf{pred}(0)$ ikke en funksjon i $\mathcal{H}om^{\mathcal{G}_{\mathsf{Int}'}}_{\mathcal{T}_{\mathsf{Int}'}(\mathcal{V})}$, siden x her er av type intposzero.)

3.3 Algebraiske spesifikasjoner av syntaktiske funksjoner

Vi vender nå tilbake til det som skal være hovedtemaet i dette kapittel; nemlig *semantikkspesifikasjon* ved syntaktiske funksjoner. I særdeleshet skal generatorsemantikk spesifiseres på denne måten. Problematikken forbundet med mange-til-en generatorunivers skal ikke løses som i forrige avsnitt, ved underliggende mekanismer som typingsalgoritmer eller beregnbare klasserepresentant-funksjoner. 'Mange-til-en'-problematikken skal her søkes løst på spesifikasjonsnivå. Vi skal derfor ikke bruke semantikkgivende syntaktiske funksjoner *per se*, men betrakte algebraiske spesifikasjoner/beskrivelser av slike funksjoner.

For å lage en algebraisk spesifikasjon av en hvilken som helst funksjon, trengs en symbolsk representasjon av funksjonen og dens domene og kodomene. Dette gjelder selvsagt også for syntaktiske funksjoner. Siden domene og kodomene for syntaktiske funksjoner er termer, trengs i prinsipp termer som representerer termer. Vi begynner diskusjonen med noen betraktninger omkring representasjon av syntaktiske funksjoner. Selv om vi primært er opptatt av syntaktiske funksjoner på grunnterm-mengder, gjør vi, i håp om at visse poenger blir klarere, disse betraktningene vha. syntaktiske funksjoner på bæremengder til *vilkårlige* term-algebraer. Merk at teorien i dette avsnittet er mer for bevisstgjøring enn for bruk senere.

Definisjon 3.3 *La $A = \langle D_A, F_A \rangle$ være en Σ-algebra for en signatur Σ. La F være en mengde funksjoner på D_A. Betrakt algebraen $A' = \langle D_A, F_A \cup F \rangle$. Vi kaller A' F-**utvidelsen** av A.*

For en signatur Σ, la *Synt* være en mengde syntaktiske funksjoner over $\mathcal{T}_\Sigma(\mathcal{V})$. La T være *Synt*-utvidelsen av $\mathcal{T}_\Sigma(\mathcal{V})$. La Σ_S være en minste signatur som inneholder en passende funksjonsprofil for hver funksjon i *Synt*. Da er T en $\Sigma \cup \Sigma_S$-algebra. Merk at T imidlertid ikke er en term-algebra; bl.a. fordi symbolene i Σ_S ikke forekommer i noen elementer (altså termer) i bæremengden til T.

For å lage algebraiske spesifikasjoner/beskrivelser av funksjoner i *Synt* trengs termer som kan representere elementer og funksjonsapplikasjoner i T. For å gjøre dette fullstendig, trengs en Σ' slik at den unike grunnterm-tolken $\phi^T_{\mathcal{G}_{\Sigma'}}$ er surjektiv. Generelt holder ikke $\Sigma \cup \Sigma_S$ til dette formål: Dersom $\mathcal{V} \neq \emptyset$, vil enhver variabel i $\mathcal{V}$ være skrot i forhold til $\Sigma \cup \Sigma_S$. La derfor C være en minste signatur inneholdende en passende funksjonsprofil for hver variabel i $\mathcal{V}$. Dvs. hver variabel i $\mathcal{V}$ er tolkningen av en konstant i C. For $\Sigma' = \Sigma \cup C \cup \Sigma_S$ er da $\phi^T_{\mathcal{G}_{\Sigma'}}$ surjektiv.

Siden $\mathcal{T}_\Sigma(\mathcal{V})$ figurerer som bæremengde i T, vil hver term i $\mathcal{G}_{\Sigma'}$ som også er i $\mathcal{G}_\Sigma$ tolkes til seg selv av $\phi^T_{\mathcal{G}_{\Sigma'}}$. Vårt valg for Σ' innebærer at vi lar grunntermer være sin egen symbolske representasjon: *Det er unødvendig å lage nye termer for å representere termer.* I praksis skal vi da også la $C = \mathcal{V}$; dvs. profilene i $\mathcal{V}$ fungerer både som variabel-profiler og som konstantfunksjons-profiler. Da vil hver term i $\mathcal{G}_{\Sigma'}$ som også er i $\mathcal{T}_\Sigma(\mathcal{V})$ tolkes til seg selv av $\phi^T_{\mathcal{G}_{\Sigma'}}$. Merk dog at et symbol i $C = \mathcal{V}$ vil før tolking være en konstant, men etter tolking være et variabelsymbol.

Eksempel 42 Betrakt signaturene

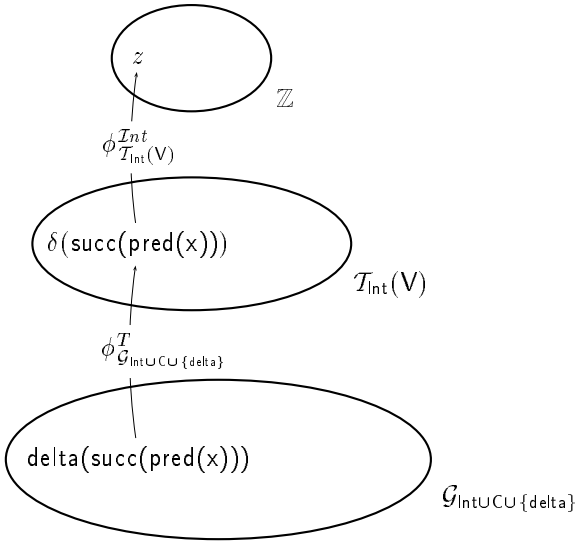
$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0 : int}, \\ \mathsf{succ : int \rightarrow int}, \\ \mathsf{pred : int \rightarrow int} \end{array} \right\}$$

$$\mathsf{C = V = \{x\}}$$

La δ være en syntaktisk funksjon på $\mathcal{T}_{\mathsf{Int}}(\mathsf{V})$, og la T være δ-utvidelsen av $\mathcal{T}_{\mathsf{Int}}(\mathsf{V})$. For et funksjonssymbol delta med passende profil, er T en IntUCU{delta}-algebra.

Termen delta(succ(pred(x))) er en *grunn*term i $\mathcal{G}_{\mathsf{IntUCU\{delta\}}}$, og tolkes av grunntermtolken $\phi^{T}_{\mathcal{G}_{\mathsf{IntUCU\{delta\}}}}$ til elementet δ((succ(pred(x)))) i T. Merk at termen (succ(pred(x))) nå *ikke* er en grunnterm i bæremengden $\mathcal{T}_{\mathsf{Int}}(\mathsf{V})$ til T.

For en $\phi^{\mathcal{I}nt}_{\mathcal{T}_{\mathsf{Int}}(\mathsf{V})} \in \mathcal{H}om^{\mathcal{I}nt}_{\mathcal{T}_{\mathsf{Int}}(\mathsf{V})}$ er $\phi^{\mathcal{I}nt}_{\mathcal{T}_{\mathsf{Int}}(\mathsf{V})}(\delta((\mathsf{succ}(\mathsf{pred}(\mathsf{x})))))$ et element i $\mathbb{Z}$. Se figur 3.3. En algebraisk beskrivelse av δ vil trolig måtte være en $E \subseteq \mathcal{E}(\mathcal{T}_{\mathsf{IntUCU\{delta\}}}(\mathcal{V}'))$ for en eller annen $\mathcal{V}' \neq \mathsf{C}$.



Figur 3.3: Illustrasjon til eksempel 42. Grunntermen delta(succ(pred(x))) i bæremengden til grunntermalgebraen $\mathcal{G}_{\mathsf{IntUCU\{delta\}}}$ tolkes til elementet δ(succ(pred(x))) i $\mathcal{T}_{\mathsf{Int}}(\mathsf{V})$, som er bæremengden til T. Merk at termen succ(pred(x)) tolkes til seg selv i T. Men i $\mathcal{G}_{\mathsf{IntUCU\{delta\}}}$ er succ(pred(x)) en grunnterm, mens den som element i bæremengden til T *ikke* er en grunnterm.
Elementet δ(succ(pred(x))) kan så tolkes (ikke-unikt) til et element z i bæremengden til $\mathcal{I}nt$.

○

Eksempel 42 illustrerer symbolsk representasjon av termer ved seg selv. Men eksemplet illustrerer også at utvidelser av term-algebraer med syntaktiske funksjoner befinner seg på et semantisk nivå mellom den syntaktiske verden og det «ekte» semantiske plan. Vi har to semantiske nivåer, men har valgt å bruke samme representasjon for begge der dette er mulig. Dette betyr at termer i dette tilfellet representerer både termer og «ekte» semantiske objekter.

Eksempel 42 (forts.) Termen succ(0) i $\mathcal{G}_{\mathsf{IntUCU\{delta\}}}$ kan tolkes både til elementet succ(0) i bæremengden til T og til elementet 1 i $\mathbb{Z}$
○

Ved å bruke samme representasjon for to semantiske nivåer på denne måten, kan en hvilken som helst algebraisk spesifikasjon av en funksjon også sees som en algebraisk spesifikasjon av en syntaktisk funksjon:

Eksempel 43 Peano-aksiomene for addisjon på naturlige tall (nå skrevet prefix)

$$\left\{ \begin{array}{l} +(\mathsf{x},0) = \mathsf{x}, \\ +(\mathsf{x},\mathsf{succ}(\mathsf{y})) = \mathsf{succ}(+(\mathsf{x},\mathsf{y})) \end{array} \right\}$$

kan sees som en algebraisk spesifikasjon av den syntaktiske funksjonen som gitt to termer $\mathsf{succ}^n(\mathsf{0})$ og $\mathsf{succ}^m(\mathsf{0})$ i $\mathcal{G}_{\{\mathsf{0},\mathsf{succ}\}}$ gir termen $\mathsf{succ}^{n+m}(\mathsf{0})$.
○

Det motsatte gjelder selvfølgelig også: Enhver algebraisk spesifikasjon av en syntaktisk funksjon kan også sees som en algebraisk spesifikasjon av en «ekte» semantisk funksjon.

Kommentar:

Det er naturlig å tenke på en abstrakt datatype som noe semantisk, som søkes representert symbolsk og implementert av en formell datatype. Term-algebraer er i denne sammenheng ikke semantiske, men hører til den syntaktiske verden som er gjenstand for tolking til semantiske objekter. Det er begrepsmessig tilfredstillende å ha et skille mellom syntaks og semantikk. Syntaks på den ene side er noe som tolkes, og semantiske objekter på den andre side er noe som er tolkninger av og gir mening til syntaks. Vi føler kanskje at vår matematiske verden straks blir litt mer ryddig.

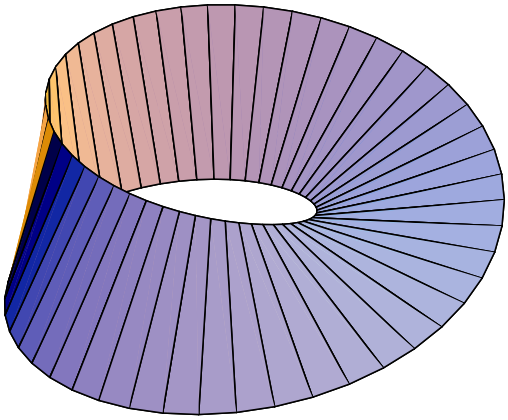
Men enhver algebra kan i prinsipp fungere som en abstrakt datatype; også syntaktiske algebraer som f.eks. term-algebraer — eller for oss mer interessant — utvidelser av term-algebraer med syntaktiske funksjoner. Termer kan således fungere i to roller: som *syntaks* og gjenstand for tolking; og som *semantikk* og selv fortolkninger av syntaks. Man skjønner at skillet mellom syntaks og semantikk avhenger av *rollen* de involverte objektene til enhver tid spiller.

Så videre er det ingenting i veien for å tilordne ikke-syntaktiske objekter en «sekundær» mening. Symboler kan forestilles å være «mening-løse». Symboler passer derfor naturlig inn i begreper som abstrakte maskiner og egner seg for å tilordnes «mening». Men det er ingenting i veien for å tolke f.eks. *tall* til f.eks. *symboler*.

Å arbeide på tvers av et kanskje etablert skille mellom syntaks og semantikk er en del av essensen i f.eks. Gödels bevis for ufullstendighet av elementære tallteorier [Göd31], [EN58]. Syntaks — predikatlogiske utsagn tolkes i utgangspunktet til matematiske påstander om naturlige tall. Med *Gödel-nummerering* kodet Gödel predikatlogiske utsagn ved tall. Tall representerer altså predikatlogiske utsagn og predikatlogiske utsagn representerer matematiske påstander om tall. Følgelig kan et predikatlogisk utsagn representere en matematisk påstand om predikatlogiske utsagn; som igjen representerer matematiske påstander om tall. Ved siden av å kunne tolkes som påstander om naturlige tall, kan altså predikatlogisk utsagn også tolkes som påstander om påstander om naturlige tall.

Et predikatlogisk utsagn G konstrueres så slik at dens tolkning som påstand om påstander om naturlige tall er «Det predikatlogiske utsagn med Gödel-nummer n er ikke utledbart fra aksiomer for elementær tallteori»; hvor utsagnet med Gödel-nummer n er G selv(!) Hverken G eller negasjonen $\neg G$ kan være utledbart i elementær tallteori. Imidlertid representerer G (også) et utsagn om naturlige tall som er sant (bevises enkelt, men selvfølgelig *utenfor* predikatkalkylen). Ergo er elementær tallteori ufullstendig.

Dette er juks, kan man fristes til å si. Det er ikke meningen at selvreferende påstander om systemet selv — og langt mindre påstander som endog snakker om seg selv — skal få være med i «leken»,



Figur 3.4: Möbius-bånd. Sammenbrudd av skillet mellom syntaks og semantikk. «Hvilken side er jeg på?» En flate i rommet med én side; uten evne til å dele rommet i to.

og følgelig kan hele ufullstendighetsbeviset forkastes ved å anta fornuftige meta-nivåer.

Poenget er imidlertid at meta-nivåene er der og overlever hele ufullstendighetsbeviset. Det er aldri snakk om noe annet enn lovlige predikatlogiske utsagn som er ment å tolkes til påstander (innen ett meta-nivå) om naturlige tall. Cruxet er imidlertid at det er ingen ting i veien for å definere en alternativ snedig avbildning (ved Gödel-nummerering) av predikatlogiske utsagn til påstander om påstander om naturlige tall, på en slik måte at altså ufullstendighet demonstreres.

Et annet «legendarisk» resultat hvis bevis tilsynelatende overtrer etablerte meta-nivåer, er uavgjørbarhet av *stoppeproblemet* [Tur36]. Her tar en Turing-maskin *kodinger av* andre Turing-maskiner og spesielt av seg selv som input.

Det er nyttig med rigide forestillinger om skiller mellom meta-nivåer, mellom syntaks og semantikk, mellom påstander og det som det påstås noe om, mellom maskiner og programmer osv. Men det er også viktig å kunne betrakte ting som et homogent landskap av mengder og avbildninger. Noen meta-nivåer kan dessuten være for strenge: Mange problematiske, men morsomme paradoks, f.eks. det før nevnte 'Russels paradoks', elimineres ved å innføre (u)passende metanivåer. Noen av skillene vi setter opp er kun for å hjelpe oss begrepsmessig i *gitte* situasjoner; som i betrakninger tilknyttet implementasjon av abstrakte datatyper.

Vi presenterer nå noen eksempler på algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner.

Eksempel 44 Betrakt kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32

på side 64. Den (konvergente) ligningsmengden $E_{\delta_{\mathcal{I}nt}}$ under er en en algebraisk spesifikasjon av $\delta_{\mathcal{I}nt}$.

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0 : int}, \\ \mathsf{succ : int \rightarrow int}, \\ \mathsf{pred : int \rightarrow int} \end{array} \right\}$$

$$\mathbf{\Delta}_{\mathsf{Int}} = \left\{ \begin{array}{l} \mathsf{delta : int \rightarrow int}, \\ \#_{\mathsf{s}} \mathsf{ : int \rightarrow int}, \\ \#_{\mathsf{p}} \mathsf{ : int \rightarrow int}, \\ \mathsf{- : int \times int \rightarrow int}, \\ \mathsf{+ : int \times int \rightarrow int} \end{array} \right\}$$

$$E_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{delta(x) = \#_s(x) - \#_p(x)}, \\ \mathsf{\#_s(0) = 0}, \\ \mathsf{\#_s(succ(x)) = succ(0) + \#_s(x)}, \\ \mathsf{\#_s(pred(x)) = \#_s(x)}, \\ \mathsf{\#_p(0) = 0}, \\ \mathsf{\#_p(pred(x)) = succ(0) + \#_p(x)}, \\ \mathsf{\#_p(succ(x)) = \#_p(x)} \\ \\ \mathsf{succ(x) - succ(y) = x - y}, \\ \mathsf{0 - succ(x) = pred(0 - x)}, \\ \mathsf{x - 0 = x}, \\ \\ \mathsf{x + succ(y) = succ(x + y)}, \\ \mathsf{x + 0 = x} \end{array} \right\}$$

En naturlig tolkning av $\#_{\mathsf{s}}$ er en funksjon som tar en term i $\mathcal{G}_{\mathsf{Int}}$ og gir antallet (et naturlig tall) succ'er i termen. Analogt for $\#_{\mathsf{p}}$ og antall pred'er. Differansen finnes så ved tolkningen av – til 'minus' på naturlige tall. Disse tolkningene gjør jo tolkningen av delta til en funksjon med kodomene de hele tall, og da ikke til en syntaktisk kanonisk-representant funksjon. For å forklare virkemåten av $E_{\delta_{\mathcal{I}nt}}$ er disse tolkningene hensiktsmessige. Over nevnte vi at enhver algebraisk spesifikasjon av en funksjon også kan sees som en algebraisk spesifikasjon av en syntaktisk funksjon. Så gitt intuisjonen for $E_{\delta_{\mathcal{I}nt}}$ fremskaffet ved tolkningene som over av $\#_{\mathsf{s}}$, $\#_{\mathsf{p}}$ osv., kan nå $E_{\delta_{\mathcal{I}nt}}$ sees som algebraiske beskrivelser av syntaktiske funksjoner. Bl.a. tolkes da de nevnte $\#_{\mathsf{s}}$, $\#_{\mathsf{p}}$ og – til passende syntaktiske funksjoner.
○

Eksempel 45 Vi presenterer en algebraisk spesifikasjon av δ_{SetNat} fra eksempel 33 side 65.

$$\mathsf{SetNat} = \left\{ \begin{array}{l} \mathsf{0 : nat}, \\ \mathsf{succ : nat \rightarrow nat}, \\ \mathsf{\emptyset : setnat}, \\ \mathsf{add : setnat \times nat \rightarrow setnat} \end{array} \right\}$$

$$\mathbf{\Delta}_{\mathsf{SetNat}} = \left\{ \begin{array}{l} \mathsf{delta : setnat \rightarrow setnat}, \\ \mathsf{mem : setnat \times setnat \rightarrow setnat}, \\ \mathsf{sort : nat \times setnat \times setnat \times nat \rightarrow setnat}, \\ \mathsf{close : setnat \times setnat \rightarrow setnat}, \\ \mathsf{- : nat \times nat \rightarrow nat}, \\ \mathsf{pred : nat \rightarrow nat} \end{array} \right\}$$

$$\mathsf{V} = \left\{ \begin{array}{l} \mathsf{s,t : setnat,} \\ \mathsf{w,x,y,z : nat} \end{array} \right\}$$

(Merk forkortet skrivemåte i variabelsignaturen V.)

$E_{\delta_{\mathsf{SetNat}}} =$

$$\left\{ \begin{array}{ll}
1: & \mathsf{delta(s) = mem(s,\emptyset),} \\
 & \\
2: & \mathsf{mem(add(s,0),\emptyset) = mem(s,sort(0,\emptyset,\emptyset,pred(0)–0)),} \\
3: & \mathsf{mem(add(s,succ(x)),\emptyset) = mem(s,sort(succ(x),\emptyset,\emptyset,0–succ(x))),} \\
4: & \mathsf{mem(add(s,x),add(t,y)) = mem(s,sort(x,add(t,y),\emptyset,y–x)),} \\
5: & \mathsf{mem(\emptyset,s) = s,} \\
 & \\
6: & \mathsf{sort(x,s,t,0) = close(t,s),} \\
7: & \mathsf{sort(x,s,t,pred(y)) = close(t,add(s,x)),} \\
8: & \mathsf{sort(w,add(add(s,x),y),t,succ(z)) = sort(w,add(s,x),add(t,y),x–w),} \\
9: & \mathsf{sort(w,add(\emptyset,x),t,succ(z)) = close(t,add(add(\emptyset,w),x)),} \\
 & \\
10: & \mathsf{close(\emptyset,s) = s,} \\
11: & \mathsf{close(add(s,x),t) = close(s,add(t,x)),} \\
 & \\
12: & \mathsf{x–0 = x,} \\
13: & \mathsf{0–succ(x) = pred(0–x),} \\
14: & \mathsf{0–pred(x) = succ(0–x),} \\
15: & \mathsf{succ(x)–succ(y) = x–y,} \\
16: & \mathsf{pred(x)–pred(y) = x–y}
\end{array} \right\}$$

Intuisjon for $E_{\delta_{\mathsf{SetNat}}}$: $E_{\delta_{\mathsf{SetNat}}}$ er konvergent, så vi behandler $E_{\delta_{\mathsf{SetNat}}}$ som et abstrakt deterministisk program.

En input-term til delta plasseres umiddelbart i hukommelsen mem (ligning 1). Prosesseringen av en term på formen

$$\mathsf{add(add(}\cdots\mathsf{(add(\emptyset,}g_1\mathsf{))}\cdots\mathsf{,}g_{n-1}\mathsf{),}g_n\mathsf{)}$$

i mem foregår ved å lese termen utenfra og inn, og over i mems 2. argument. Overføringen til mems 2. argument skjer via sort. Det 2. argument i mem er da invariant kanonisk (dvs. sortert).

Overføringen til mems 2. argument skjer vanligvis ved ligning 4. Ligningene 2 og 3 tar seg av initieringen som vi kommer tilbake til.

Subprosedyren sort fungerer slik: 1. argument inneholder en $\mathcal{G}_{\mathsf{Nat}}$-generatorterm, som skal plasseres i riktig posisjon i termen gitt i 2. argument. Termen gitt i 2. argument åpnes opp, dvs. leses over i 3. argument til riktig posisjon er lokalisert. Riktig posisjon bestemmes av beregningen gjort i 4. argument. Denne beregning er intuitivt beregningen av differansen mellom verdien av $\mathcal{G}_{\mathsf{Nat}}$-generatortermen i posisjonen som sees på for øyeblikket, og verdien av kandidattermen som skal plasseres. Når kandidat-termen er plassert, lukkes termen ved subprosedyren close.

Ligning 6: Differansen er 0. Altså fins verdien fra før, og vi kan lukke.

Ligning 7: Differansen er negativ. Kandidat-termen skal inn her, og vi lukker.

Ligning 8: Differansen er positiv. Vi må lete videre.

Ligning 9: Nå har vi lett gjennom hele. Kandidat-termens verdi er mindre enn alle andre verdier i mengden. Kandidat-termen skal inn innerst, og vi lukker.

Ligningene 2 og 3 initierer prosessering i sort. Spørsmålet er hvordan beregningen i sorts 4. argument skal initieres, siden det ikke ennå finnes noen for øyeblikket lest $\mathcal{G}_{\mathsf{Nat}}$-generatorterm å sammenligne den aller første kandidattermen med. Et naturlig og riktig valg for en fiktiv sammenligningsterm, er 0

(ligning 3); *bortsett* fra i tilfellet der kandidat-termen selv er $\mathsf{0}$. Isåfall løses problemet ved å velge $\mathsf{pred(0)}$ som fiktiv term (ligning 2). Ligningene 12 – 16 tar seg av beregningen i 4. argument av sort.

Vi har gjort et poeng her av å ikke innføre nye typer. Imidlertid kan en mer lesbar spesifikasjon skrives ved hjelp av ting som $\mathsf{if\ then\ else}$ og $<$. Merk at vi i dette eksemplet ikke anser pred som et generatorsymbol.
○

Eksempel 46 Det er lett å modifisere $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 til en algebraisk spesifikasjon $E_{\gamma_{\mathcal{I}nt}}$ av $\gamma_{\mathcal{I}nt}$ fra eksempel 34 side på 65. Dette kan gjøres på flere måter. Endre f.eks. ligningen $\#_{\mathsf{s}}(\mathsf{0}) = \mathsf{0}$ til $\#_{\mathsf{s}}(\mathsf{0}) = \mathsf{succ(0)}$.
○

Vi merker oss at disse spesifikasjonene av semantikkgivende syntaktiske funksjoner har involvert ***hjelpefunksjon(symbol)er*** ($\#_{\mathsf{s}}$, $\#_{\mathsf{p}}$, $-$, $+$, mem og sort for å nevne noen.) Mange funksjoner kan med fordel spesifiseres ved hjelp av hjelpefunksjoner, og dersom presis matematisk spesifikasjon (f.eks. algebraisk) er ønskelig, er det kanskje(?) noen ganger nødvendig med hjelpefunksjoner. Hjelpefunksjoner og hjelpeprosedyrer er imidlertid svært sentrale i programmering og bidrar vesentlig til oppdeling av programmeringsoppgaver («splitt og hersk»).

La nå *synt* være en syntaktisk funksjon som spesifiserer en kongruensrelasjon $\simeq$ på en $\mathcal{G}_\Sigma$ ved prinsipp (3.2) på side 64. For E_s en algebraisk spesifikasjon av *synt*, er det innlysende at

$$s(g) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g') \;\Leftrightarrow\; g \simeq g' \tag{3.5}$$

for alle $g, g' \in \mathcal{G}_\Sigma$, der $s \notin \Sigma$ er et funksjonsymbol med passende profil slik at *synt* er tolkningen av s (i *synt*-utvidelsen av $\mathcal{G}_\Sigma$). Det er dog ikke nødvendig at E_s er en (fullstendig) algebraisk spesifikasjon av *synt* for å tilfredstille 3.5:

Eksempel 47 Betrakt følgende ligningsmengde:

$$E_{\mathsf{synt}} = \left\{ \begin{array}{l} \mathsf{synt(succ(x))} = \mathsf{synt(succ(synt(x)))}, \\ \mathsf{synt(pred(x))} = \mathsf{synt(pred(synt(x)))}, \\ \mathsf{synt(succ(synt(pred(x))))} = \mathsf{synt(x)}, \\ \mathsf{synt(pred(synt(succ(x))))} = \mathsf{synt(x)} \end{array} \right\}$$

For signaturen Int og $E_{\mathcal{I}nt_c}$ fra eksempel 11 side 21, har vi at E_{synt} tilfredstiller

$$\mathsf{synt}(g) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(g') \;\Leftrightarrow\; g \overset{*}{\underset{E_{\mathcal{I}nt_c}}{\leftrightarrow}} g'$$

for alle $g, g' \in \mathcal{G}_{\mathsf{Int}}$.

Intuisjonen for E_{synt} er som følger: For to termer $\mathsf{synt}(g), \mathsf{synt}(g')$ for $g, g' \in \mathcal{G}_{\mathsf{Int}}$ slik at $\mathsf{synt}(g) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(g')$, er det mulig å omskrive begge termene til en term «mettet» med synt-forekomster; i den forstand at det ikke finnes to umiddelbart etterfølgende symboler fra $\mathcal{G}_{\mathsf{Int}}$ i termen. Vi har f.eks.

$$\begin{array}{l} \mathsf{synt(succ(succ(succ(pred(0)))))} \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \cdots \\ \qquad\qquad \vdots \\ \cdots \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt(succ(synt(succ(synt(0)))))} \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \cdots \\ \qquad\qquad\qquad\qquad \vdots \\ \qquad\qquad\qquad\qquad\qquad\qquad \cdots \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt(succ(succ(0)))} \end{array}$$

i tråd med at $\mathsf{succ(succ(succ(pred(0))))} \overset{*}{\underset{E_{\mathcal{I}nt_c}}{\leftrightarrow}} \mathsf{succ(succ(0))}$.

Men E_{synt} er ikke tilstrekkelig synt-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$. Dermed er ikke E_{synt} en algebraisk spesifikasjon av $\delta_{\mathcal{I}nt}$ fra eksempel 32. Men E_{synt} er dog en algebraisk *beskrivelse* av $\delta_{\mathcal{I}nt}$ (se definisjon 2.2 side 19).

Merk at E_{synt} ikke er en algebraisk beskrivelse av $\gamma_{\mathcal{I}nt}$ fra eksempel 34. Dette siden

$$\mathsf{synt(succ(0))} \overset{*}{\underset{E_{\mathsf{synt}}}{\longleftrightarrow}} \mathsf{synt(succ(synt(0)))}$$

men

$$G \not\models \mathsf{synt(succ(0))} = \mathsf{synt(succ(synt(0)))}$$

for $\gamma_{\mathcal{I}nt}$-utvidelsen G av $\mathcal{G}_{\mathsf{Int}}$, siden

$$\gamma_{\mathcal{I}nt}(\mathsf{succ(0)}) = \mathsf{succ(succ(0))}$$

mens

$$\gamma_{\mathcal{I}nt}(\mathsf{succ}(\gamma_{\mathcal{I}nt}(\mathsf{0}))) = \mathsf{succ(succ(succ(0)))}$$

og $\mathsf{succ(succ(0))}$ og $\mathsf{succ(succ(succ(0)))}$ jo er to (ikke identiske) elementer i bæremengden til G.
○

Definisjon 3.4 *La $\simeq$ være en kongruensrelasjon spesifisert av en syntaktisk funksjon synt. Vi kaller en ligningsmengde E_s som for $\simeq$ tilfredstiller (3.5), en* ***(algebraisk) spesifikasjon*** *av $\simeq$* ***mhp.*** *synt .*

En algebraisk spesifikasjon av en kongruensrelasjon mhp. en syntaktisk funksjon, gir tydelig et (nesten) like direkte ligningslogisk grep om kongruensrelasjonen som en (vanlig) algebraisk spesifikasjon ville ha gjort: Nå trengs bare symbolet s settes som rot før ligningslogikk kan slippes til. Algebraiske spesifikasjoner av kongruensrelasjoner mhp. syntaktiske funksjoner egner seg derfor som atomære semantikker.

Vi er istand til å gi algebraiske spesifkasjoner/beskrivelser for semantikkgivende funksjoner som *ikke* er omskrivningsberegnbare. Således gir algebraisk spesifikasjon mhp. syntaktiske funksjoner muligheten for et *deterministisk* ligningslogisk grep om flere semantikker (kongruensrelasjoner) enn dem som kan spesifiseres ved konvergente direkte algebraiske spesifkasjoner.

Eksempel 48 Betrakt

$$\mathsf{SetNat} = \left\{ \begin{array}{l} \mathsf{0 : nat}, \\ \mathsf{succ : nat \longrightarrow nat}, \\ \mathsf{\emptyset : setnat}, \\ \mathsf{add : setnat \times nat \longrightarrow setnat} \end{array} \right\}$$

$$E_{\mathsf{SetNat}} = \left\{ \begin{array}{l} \mathsf{add(add(s,x),x) = add(s,x)}, \\ \mathsf{add(add(s,x),y) = add(add(s,y),x)} \end{array} \right\}$$

La $\simeq^x$ være semantikken på $\mathcal{G}_{\mathsf{SetNat}}$ spesifisert direkte algebraisk av E_{SetNat}.

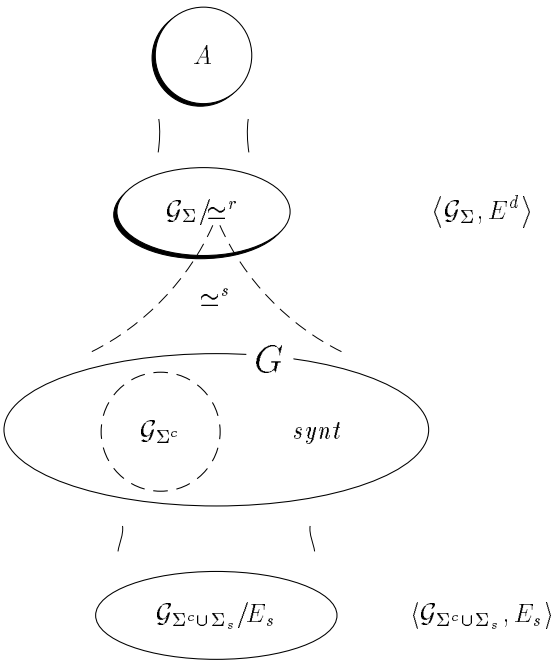
Siden ingen endelig regelmengde er komplett for E_{SetNat}, ei heller $\mathcal{G}_{\mathsf{SetNat}}$-komplett, er ikke $\simeq^x$ avgjørbar ved noen direkte algebraisk spesifikasjon.

Derimot er $\simeq^x$ avgjørbar ved den algebraiske spesifikasjon $E_{\delta_{\mathsf{SetNat}}}$ mhp. δ_{SetNat} fra eksempel 45 side 78.

Vi minner da om siste tekstavsnitt i eksempel 31 side 59.
○

Ved algebraisk spesifikasjon mhp. syntaktiske funksjoner spesifiseres semantikk begrepsmessig på to nivåer: En algebraisk beskrivelse spesifiserer (noe av) semantikken til en syntaktisk funksjon som i sin tur spesifiserer semantikken til termer. Figur 3.5 illustrerer situasjonen.

Vi har ikke utviklet metoder som søker resolusjon av generell initial- og finalsemantikk. Imidlertid skal vi se at semantikk relativ til semantikk spesifisert



Figur 3.5: Semantikkspesifikasjon på to nivåer. Her er det en Σ-algebra A som søkes implementert, og $\simeq^A_{\mathcal{G}_\Sigma}$ — kongruensrelasjonen indusert av $\phi^A_{\mathcal{G}_\Sigma}$ — søkes derfor spesifisert på en formelt tilnærmbar måte. Σ er her delt i Σ^c og Σ^d bestående av generatorer og definerte funksjonssymboler hhv.
Nedre nivå: $E_s \subseteq \mathcal{E}(\mathcal{T}_{\Sigma_s}(\mathcal{V}))$ er en algebraisk beskrivelse av en syntaktisk funksjon *synt* i *synt*-utvidelsen G av $\mathcal{G}_{\Sigma^c}$.
Øvre nivå: Generatorsemantikken $\simeq^s$ spesifiseres av *synt*. Semantikken $\simeq^r$ er en semantikk relativ til $\simeq^s$ bestemt av Σ og en ligningsmengde E^d som gir semantikk til symboler i Σ^d.
Den formelle omgivelse for $\simeq^s$ er $\langle \mathcal{G}_{\Sigma^c \cup \Sigma_s}, E_s \rangle$. Den formelle omgivelse for $\simeq^r$ er mengden av $\langle \mathcal{G}_{\Sigma^c \cup \Sigma_s}, E_s \rangle$ og $\langle \mathcal{G}_\Sigma, E^d \rangle$. Grensesnitt-omgivelsen/den logiske omgivelse for $\simeq^r$ er $\langle \mathcal{G}_\Sigma, E^d \rangle$.

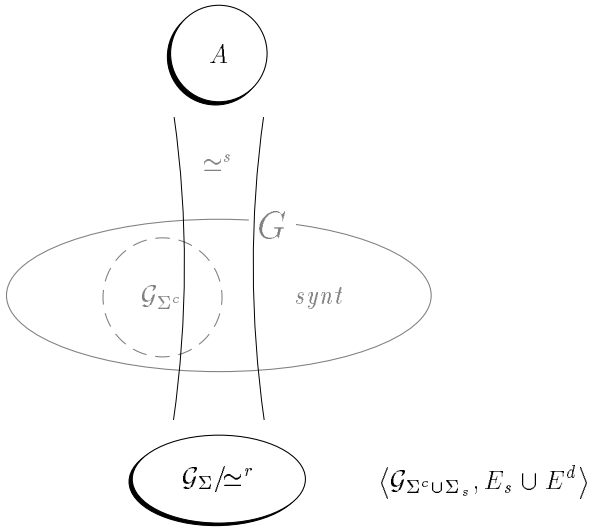
mhp. syntaktiske funksjoner i noen tilfeller kan reduseres til basis-initial/basis-finalsemantikk og er således tilgjengelig for eksisterende resolusjonsmetoder.

Det er mulig å delvis bruke en og samme representasjon på tvers av våre to semantikkgivende nivå. Det er derfor mulig å betrakte den algebraiske beskrivelsen av den syntaktiske funksjonen på samme nivå som algebraisk spesifikasjon av «ekte» semantiske funksjoner, og disse to algebraiske beskrivelser/spesifikasjoner kan så slås sammen. Dette skal vi utnytte i reduksjonen til basis-initial/basis-finalsemantikk. Figur 3.6 illustrerer situasjonen.

3.4 Reduksjon til basis-semantikker

I dette avsnittet viser vi at semantikker relative til semantikker spesifisert mhp. semantikkgivende syntaktiske funksjoner, under visse omstendigheter, kan reduseres til basis-semantikker.

Som et ledd i dette, betrakter vi algebraiske beskrivelser av semantikkgiven-



Figur 3.6: Semantikkspesifikasjon på ett nivå og sammenslåtte funksjonsspesifikasjoner/beskrivelser.
Den algebraiske beskrivelsen E_s av den semantikkgivende syntaktiske funksjonen *synt* og den algebraiske spesifikasjon E^d deler representasjon. Dermed er det mulig å se E_s og E^d som del av én formell (logisk) omgivelse $\langle \mathcal{G}_{\Sigma^c \cup \Sigma_s}, E_s \cup E^d \rangle$ for $\simeq^r$, der $\simeq^r$ er en basis-initial eller en basis-final semantikk.

de syntaktiske funksjoner på samme «spesifikasjonsnivå» som øvrig algebraisk beskrivelse/spesifikasjon.

Det er da av og til naturlig å «glemme» at algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner har noe med de syntaktiske funksjoner å gjøre, da sistnevnte for oss er ledd i en «to-nivå-spesifikasjon».

Istedet skal vi finne det hensiktsmessig å snakke om ligningsmengder som er ment å være semantikkspesifikasjoner mhp. syntaktiske funksjoner, på et helt og holdent syntaktisk nivå. For dette defineres først noen relevante begreper. Deretter setter vi opp rammen rundt selve diskusjonen om reduksjon til basis-semantikker.

3.4.1 Kongruens og indirekte spesifikasjon

Dersom E_s er en algebraisk spesifikasjon av en kongruensrelasjon $\simeq$ mhp. en syntaktisk funksjon *synt*, har vi nødvendigvis for alle kontekster c og termer g, g' i domenet til $\simeq$ og et passende symbol s ment å representere *synt*,

$$s(g) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g') \;\Rightarrow\; s(c[g]) \overset{*}{\underset{E_s}{\leftrightarrow}} s(c[g']) \tag{3.6}$$

for alle c, g, g' i domenet til $\simeq$ og et passende symbol s. (Se også (3.3) side 64.)
Å lage algebraiske spesifikasjoner/beskrivelser av semantikkgivende syntaktiske funksjoner er ikke alltid så lett, så i diskusjonen fremover skal vi ikke alltid ta for gitt at en ligningsmengde som er ment å være en spesifikasjon av en kongruensrelasjon $\simeq$ mhp. en syntaktisk funksjon, nødvendigvis er det. Spesifikt skal vi betrakte en «feil» som 'inkongruens' for mulig.

Eksempel 49 Betrakt følgende forsøk på en spesifikasjon av $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ mhp. kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32.

$$E'_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{delta(succ(pred(x))) = delta(x)}, \\ \mathsf{delta(pred(succ(x))) = delta(x)}, \\ \mathsf{delta(succ(succ(x))) = succ(delta(succ(x)))}, \\ \mathsf{delta(pred(pred(x))) = pred(delta(pred(x)))}, \\ \mathsf{delta(succ(0)) = succ(0)}, \\ \mathsf{delta(pred(0)) = pred(0)}, \\ \mathsf{delta(0) = 0} \end{array} \right\}$$

$E'_{\delta_{\mathcal{I}nt}}$ er konvergent, og vi har

$$\mathsf{delta(pred(succ(succ(0))))}_{E} \overset{!}{\underset{\delta_{\mathcal{I}nt}}{\mapsto}} \mathsf{succ(0)}_{E} \overset{!}{\underset{\delta_{\mathcal{I}nt}}{\leftarrow}} \mathsf{delta(succ(0))}$$

men

$$\mathsf{delta(pred(pred(succ(succ(0)))))}_{E} \overset{!}{\underset{\delta_{\mathcal{I}nt}}{\mapsto}} \mathsf{pred(succ(0)) \neq 0}_{E} \overset{!}{\underset{\delta_{\mathcal{I}nt}}{\leftarrow}} \mathsf{delta(pred(succ(0)))}$$

Altså oppfyller ikke E (3.6) og er således 'inkongruent' og ubrukelig for semantikkspesifikasjon etter prinsipp 3.5 side 80.
○

Definisjon 3.5 *La Σ være en signatur inneholdende et unært funksjonssymbol s. La $E_s \in \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ være en ligningsmengde. Betrakt relasjonen $\Re^s$ slik at for alle $g, g' \in \mathcal{G}_\Sigma$*

$$g \ \Re^s \ g' \ \Leftrightarrow \ s(g) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g')$$

*For ethvert redukt $\mathcal{G}$ av $\mathcal{G}_\Sigma$ er E_s **indirekte $\mathcal{G}$-kongruent (mhp. s)** dersom $\Re^s_{\mathcal{G}}$ er en kongruensrelasjon.*

*Isåfall kaller vi E_s en **indirekte (algebraisk) spesifikasjon** av $\Re^s_{\mathcal{G}}$. Termene i $\mathcal{G}$ sier vi gis **indirekte semantikk**, og symbolet s sier vi er **spesifiserende**.*

3.4.2 Formell omgivelse...

Vi definerer nå formelle datatyper ved bruk av indirekte spesifikasjon. Vi skal benytte oss av teorien i avsnitt 2.3 i kapittel 2. Vi skal følge intensjonene A side 30 og B side 31.

Vi antar i de følgende avsnitt signaturer og ligningsmengder som vist i figur 3.7.

Vi skal i det følgende foreløpig anta at

`INDKONG`: E_s er indirekte $\mathcal{G}_{\Sigma^c}$-kongruent.

`INDDEGEN`: Det fins ingen andre generatorer i $\mathcal{G}_{\hat{\Sigma}}$ enn dem i $\mathcal{G}_{\Sigma^c}$.

(Alle slike merkede antagelser finnes i registret bakerst.) Antagelsen `INDDEGEN` er en presisering for tilfellet her av `DEGEN` på side 39 i avsnitt 2.3.6. Ved `INDKONG` har vi at den indirekte semantikken $\simeq^s$ spesifisert av E_s finnes.

Andre kriterier vi kommer til å anta er:

`DISJ`: $E^d \subseteq \mathcal{E}(\mathcal{T}_{\Sigma^d \cup \Sigma^c}(\mathcal{V}))$ og $E_s \subseteq \mathcal{E}(\mathcal{T}_{\{s\} \cup \Sigma^h \cup \Sigma^c}(\mathcal{V}))$.

E_s`VARBEVAR`: Alle ligninger $v = h$ i E_s er ***variabelbevarende***, dvs. ingen variabel forekommer kun i en av v og h.

E^d`VARBEVAR`: Alle ligninger $v = h$ i E^d er variabelbevarende.

Vi antar gitt følgende signaturer og ligningsmengder:

$$\Sigma = \bigcup \begin{array}{lcl} \Sigma^c & : & \text{generatorer} \\ \Sigma^d & : & \text{definerte funksjonssymboler} \end{array}$$

$$\Sigma_s = \bigcup \begin{array}{lcl} \{s\} & : & \text{spesifiserende symbol} \\ \Sigma^h & : & \text{definerte hjelpefunksjonssymboler} \end{array}$$

$$\hat{\Sigma} = \Sigma \cup \Sigma_s$$

Samtlige $\Sigma^c, \Sigma^d, \{s\}, \Sigma^h$ er parvis disjunkte.

$$\begin{array}{lcl} E^d & : & \text{algebraisk beskrivelse for funksjonssymboler i } \Sigma^d \\ E_s & : & \text{formodet indirekte algebraisk spesifikasjon med} \\ & & \text{spesifiserende symbol } s \text{ av en generatorsemantikk } \simeq^s \end{array}$$

$$\hat{E} = E^d \cup E_s$$

Den formelle omgivelse er nå $\left\langle \mathcal{G}_{\hat{\Sigma}}, \hat{E} \right\rangle$.
Men: Det er $\mathcal{G}_\Sigma$ som først og fremst skal gis semantikk.

Figur 3.7: Formell omgivelse for reduksjon til basis-semantikker.

Antagelsen `DISJ` har betydning ved den parvise disjunkthet av samtlige Σ^c, Σ^d, $\{s\}$ og Σ^h som vi postulerte over, og sier da at E_s og E^d har kun generatorsymboler felles. Dette er en rimelig antagelse i lys av E^d og E_s som opprinnelig separate beskrivelser (på to separate «semantiske nivåer»).

Antagelsen E^d`VARBEVAR` er en rimelig antagelse i lys av konstruktiv funksjonsspesifikasjon. Antagelsen E_s`VARBEVAR` er ikke urimelig, men er ikke alltid naturlig oppfylt ved våre algebraiske spesifikasjoner av semantikkgivende syntaktiske funksjoner. F.eks. oppfyller ikke $E_{\delta_{\mathsf{SetNat}}}$ fra eksempel 45 på side 78 E_s`VARBEVAR` (f.eks. ligning 6).

Merk også at E_s`VARBEVAR` og E^d`VARBEVAR` er uorienterte versjoner av antagelsen 1 gjort på side 52 om regler i omskrivningssystemer i forbindelse med Knuth&Bendix-komplettering.

Andre antagelser er videre:

`TΣK:` $\hat{E}$ er tilstrekkelig Σ-komplett mhp. $\mathcal{G}_{\Sigma^c}$.

`TsK:` $\hat{E}$ er tilstrekkelig $\{s\}$-komplett mhp. $\mathcal{G}_{\Sigma^c}$.

`TΣ`^{h}`K:` $\hat{E}$ er tilstrekkelig Σ^h-komplett mhp. $\mathcal{G}_{\Sigma^c}$.

`TΣ̂K:` $\hat{E}$ er tilstrekkelig $\hat{\Sigma}$-komplett mhp. $\mathcal{G}_{\Sigma^c}$.

Kriteriet `TΣ̂K` er presiseringen her av `TK` på side 28 i avsnitt 2.3.3.

Vi minner om definisjonen 2.5 av tilstrekkelig kompletthet på side 25. Merk at eksempelvis `TsK` betyr at det for hver term $g \in \mathcal{G}_{\{s\}\cup\Sigma^c}$ finnes en term $g_c \in \mathcal{G}_{\Sigma^c}$ slik at $g \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_c$. `TsK` betyr f.eks. *ikke* at det finnes en $g_c \in \mathcal{G}_{\Sigma^c}$ for enhver term $s(\hat{g})$ for $\hat{g} \in \mathcal{G}_{\hat{\Sigma}}$, slik at $s(\hat{g}) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_c$.

Intuitivt bør `TΣK`, `TsK` og `TΣ`^{h}`K` tilsammen gi `TΣ̂K`. Følgende lemma bekrefter dette.

Lemma 3.4 *La Σ være disjunkt delt i Σ^x, Σ^d og $\Sigma^{d'}$. La $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ være tilstrekkelig Σ^d-komplett mhp. Σ^x og tilstrekkelig $\Sigma^{d'}$-komplett mhp. Σ^x. Da er E tilstrekkelig $\Sigma^d \cup \Sigma^{d'}$-komplett mhp. Σ^x.*

Bevis: Induksjon over antall n symboler fra $\Sigma^d \cup \Sigma^{d'}$ i en vilkårlig $g \in \mathcal{G}_\Sigma$.

$n = 0$: Trivielt.

$n = k + 1; k \geq 0$: Da er $g = c[f(g_1, \ldots, g_m)]$ for $c \in \mathcal{G}_\Sigma, g_1, g_m \in \mathcal{G}_{\Sigma^x}$ og en $f \in \Sigma^d \cup \Sigma^{d'}$. Vi betrakter altså et i g dypest forekommende symbol fra $\Sigma^d \cup \Sigma^{d'}$. Siden E er tilstrekkelig Σ^d-komplett mhp. $\mathcal{G}_{\Sigma^x}$ og tilstrekkelig $\Sigma^{d'}$-komplett mhp. $\mathcal{G}_{\Sigma^x}$, har vi $g = c[f(g_1, \ldots, g_m)] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g_0]$ for en $g_0 \in \mathcal{G}_{\Sigma^x}$. Induksjonshypotesen gir så $c[g_0] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_x$ for en $g_x \in \mathcal{G}_x$ og lemmaet følger.
□

To meget kraftige, men i spesifikasjonssammenheng rimelige antagelser er følgende:

KONVERG: $\hat{E}$ er konvergent

KONSTR: Enhver ligning $v = h \in \hat{E}$ er slik at v har formen

$$f(t_1, \ldots, t_n)$$

der f er et definert symbol i $\hat{\Sigma}$ og alle $t_i \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$; $1 \leq i \leq n$.

Merk dog at ligning 7 i eksempel 45 side 78 ikke tilfredstiller **KONSTR**.

Presiseringen her av **KONSERV** (side 28) blir:

INDKONSERV: $\overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\Sigma^c} \subseteq \simeq^s$.

M.a.o. $g_c \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g'_c \;\Rightarrow\; s(g_c) \overset{*}{\underset{E^s}{\leftrightarrow}} (g'_c)$ for alle $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$

Vi har umidelbart for alle $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$

$$g_c \overset{*}{\underset{E^s}{\leftrightarrow}} g'_c \;\Rightarrow\; s(g_c) \overset{*}{\underset{E^s}{\leftrightarrow}} (g'_c) \tag{3.7}$$

så det kan virke som om **INDKONSERV** ville være oppfylt dersom

$$g_c \overset{*}{\underset{E^d}{\leftrightarrow}} g'_c \;\Rightarrow\; s(g_c) \overset{*}{\underset{E^s}{\leftrightarrow}} (g'_c)$$

og da spesielt hvis $g_c \overset{*}{\underset{E^d}{\leftrightarrow}} g'_c \;\Rightarrow\; g_c = g'_c$. Dette er ikke tilfellet:

Eksempel 50 Anta

$\mathsf{succ(pred(0))} \simeq^s \mathsf{0}$

$\mathsf{succ(0)} \not\simeq^s \mathsf{0}$

$\mathsf{succ(pred(0))} \overset{*}{\underset{E^s}{\leftrightarrow}} \mathsf{0}$

$\mathsf{f(succ(pred(0)))} \overset{*}{\underset{E^d}{\leftrightarrow}} \mathsf{succ(0)}$

$\mathsf{f(0)} \overset{*}{\underset{E^d}{\leftrightarrow}} \mathsf{0}$

Vi får da

$$\mathsf{succ(0)} \overset{*}{\underset{E^d}{\leftrightarrow}} \mathsf{f(succ(pred(0)))} \overset{*}{\underset{E^s}{\leftrightarrow}} \mathsf{f(0)} \overset{*}{\underset{E^d}{\leftrightarrow}} \mathsf{0}$$

M.a.o. $\mathsf{succ(0)} \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathsf{0}$ selv om $\mathsf{succ(0)} \not\simeq^s \mathsf{0}$.
○

Betrakt da følgende antagelser:

E_s**FRI**^{c}: E_s er fri mhp. Σ^c; dvs. E_s har ingen ligninger fra $\mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$; utenom muligens ligninger på formen $v = v$.

E^d**FRI**^{c}: E^d er fri mhp. Σ^c; utenom muligens ligninger på formen $v = v$.

Antagelsene E_s**FRI**^{c} og E^d**FRI**^{c} er rimelige ved indirekte spesifikasjon; formodet at generatorsemantikk kun ønskes å bli gitt ved indirekte spesifikasjon.

Sats 3.5 **INDKONSERV** *følger fra* E_s**FRI**^{c}, E^d**FRI**^{c}, **KONVERG** *og* **KONSTR**

Bevis: Betrakt to vilkårlige $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$ slik at $g_c \neq g'_c$ og slik at $g_c \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g'_c$. Ved **KONVERG** har vi da $g_c \overset{!}{\underset{\hat{E}}{\rightarrow}} g_c! \overset{!}{\underset{\hat{E}}{\leftarrow}} g'_c$. Ved **KONSTR** må enhver komponent i vilkårlige ensrettede $\hat{E}$-utledninger $\langle g_c, \ldots, g_c! \rangle$ og $\langle g'_c, \ldots, g_c! \rangle$ være i $\mathcal{G}_{\Sigma^c}$. Men dette er umulig ved E_s**FRI**^{c} og E^d**FRI**^{c}. Altså må $g_c \overset{*}{\underset{\hat{E}}{\not\leftrightarrow}} g'_c$. **INDKONSERV** følger da trivielt. □

Dersom **KONVERG** ikke er oppfylt, blir det mer problematisk å etablere **INDKONSERV**. Vi skal ikke gå inn på detaljer, men betrakt følgende antagelser:

E_s**KONSERV+**: $g_c \overset{*}{\underset{E_s}{\leftrightarrow}} g'_c \;\Rightarrow\; g_c = g'_c$ for alle $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$.

E^d**KONSERV+**: $g_c \overset{*}{\underset{E^d}{\leftrightarrow}} g'_c \;\Rightarrow\; g_c = g'_c$ for alle $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$.

Antagelsene E_s**KONSERV+** og E^d**KONSERV+** er rimelige, formodet at generatorsemantikk kun ønskes å bli gitt ved indirekte spesifikasjon og i lys av konstruktiv funksjonsspesifikasjon. Merk at

$$E_s\textbf{KONSERV+} \;\Rightarrow\; E_s\textbf{FRI}^c \quad \text{og} \quad E^d\textbf{KONSERV+} \;\Rightarrow\; E^d\textbf{FRI}^c$$

E_s**KONSERV+** og E^d**KONSERV+** sammen med **DISJ**, gir at enhver subterm g^f av en komponent i en vilkårlig E^d- eller E_s-utledning $\langle g_c, \ldots \rangle$ i $\mathcal{G}_{\hat{E}}$ fra en vilkårlig $g_c \in \mathcal{G}_{\Sigma^c}$, har en unik $g^f_c \in \mathcal{G}_{\Sigma^c}$ slik at $g^f \overset{*}{\underset{E^d}{\leftrightarrow}} g^f_c$ eller $g^f \overset{*}{\underset{E_s}{\leftrightarrow}} g^f_c$.

Dette kan brukes sammen med forskjellige antagelser til å vise at enhver komponent $\hat{g}$ i en en vilkårlig $\hat{E}$-utledning $\langle g_c, \ldots \rangle$i $\mathcal{G}_{\hat{E}}$ fra en vilkårlig $g_c \in \mathcal{G}_{\Sigma^c}$, har en unik $\hat{g}_c \in \mathcal{G}_{\Sigma^c}$ slik at $\hat{g} \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \hat{g}_c$. Siden $\hat{g}_c$ er unik, har vi følgelig

$$g_c \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g'_c \;\Rightarrow\; g_c = g'_c$$

for alle $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$, og **INDKONSERV** er oppfylt. Siden **KONVERG** i vår sammenheng er en rimelig og avgjørbar egenskap, viser vi som sagt ikke detaljene her.

3.4.3 ...og formelle datatyper

Mens **INDDEGEN** og **INDKONG** i de nærmeste avsnitt vanligvis alltid skal antas å holde, skal vi eksplisitt si når de andre kriterier skal antas.

Fra elementene i figur 3.7 kan vi bygge opp initial- eller finalsemantikk relativ til indirekte semantikk.

Finalsemantikk relativ til en indirekte semantikk $\simeq^s$ kan uttrykkes som følger: For alle $g, g' \in \mathcal{G}_{\hat{\Sigma}}$

$$g \simeq^\omega g' \;\Leftrightarrow\; \left\{ \begin{array}{l} g, g' \textit{ er av samme type} \quad \textit{og} \\ \neg\exists c \in \mathcal{G}_{\hat{\Sigma}}; g_1, g_2 \in \mathcal{G}_{\Sigma^c} \mid c[g] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1 \not\simeq^s g_2 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g'] \end{array} \right. \tag{3.8}$$

«Ankerfestene» gis her ved den indirekte semantikk spesifisert av E_s på $\mathcal{G}_{\Sigma^c}$.

Det er også naturlig å benytte initialsemantikk, ved her å la $\simeq^\alpha$ være initialsemantikken relativ til $\simeq^s$ spesifisert av $\hat{\Sigma}$ og $\hat{E}$.

Antar vi **INDKONSERV** og **TΣK** og konsistens, følger det fra diskusjonen i avsnitt 2.3.7 (se slutten av avsnittet på side 44) at både $\simeq^\omega_{\mathcal{G}_\Sigma}$ og $\simeq^\alpha_{\mathcal{G}_\Sigma}$ gir oss ønsket

semantikk på $\mathcal{G}_\Sigma$, i den grad ønsket generatorsemantikk er den gitt av $\simeq^s$ og ønsket semantikk til definerte funksjonssymboler er den gitt i $\hat{E}$.

I forbindelse med indirekte spesifikasjon av generatorsemantikk, er initial- og finalsemantikk ikke de eneste alternativer for oppbygging av semantikk. Vi kunne f.eks. ganske enkelt la den indirekte spesifikasjon omfatte hele term-universet og ikke kun generator-termene: Dersom $\hat{E}$ er $\mathcal{G}_\Sigma$-kongruent, kan vi betrakte semantikken $\Re^s$ på $\mathcal{G}_{\hat{\Sigma}}$ slik at for alle $g, g' \in \mathcal{G}_{\hat{\Sigma}}$:

$$g \; \Re^s \; g' \;\Leftrightarrow\; s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g') \tag{3.9}$$

Men det er flere omstendigheter som hindrer $\mathcal{G}_\Sigma$-kongruens av $\hat{E}$. Dersom $\hat{E}$ ikke er tilstrekkelig Σ^d-komplett mhp. $\mathcal{G}_{\Sigma^c}$, kan vi ha for $c \in \Sigma^d$

$$s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g') \;\text{ men }\; s(c[g]) \overset{\not *}{\underset{\hat{E}}{\leftrightarrow}} s(c[g'])$$

hvis den indirekte spesifikasjon E_s inneholdt i $\hat{E}$ ikke (tilfeldigvis) er slik at den eksplisitt gir ønsket semantikk også til definerte funksjonssymboler. Dette er kanskje mindre trolig, ettersom E_s formodentlig opprinnelig var en algebraisk beskrivelse av en (semantikkgivende syntaktisk) funksjon på $\mathcal{G}_{\Sigma^c}$.

Dette opphavet til inkongruens her er også et opphav til inkongruens av final pseudosemantikk. Begge inkongruenssituasjoner reddes imidlertid av antagelser om tilstrekkelig kompletthet. Senere vil vi se andre mer tungtveiende grunner til at $\hat{E}$ ikke er $\mathcal{G}_\Sigma$-kongruent.

I de neste to avsnittene reduserer vi semantikkene $\simeq^\omega$ og $\simeq^\alpha$ til basis-semantikker.

3.4.4 Reduksjon til basis-finalsemantikk

Vi vil nå redusere finalsemantikken $\simeq^\omega$ i (3.8) til en basis-finalsemantikk. Betrakt følgende kriterium:

ASSS: E_s er en algebraisk spesifikasjon av en semantikkgivende syntaktisk funksjon *synt*.

Kriteriet ASSS medfører T*s*K.

Teorem 3.6 *La $\simeq^{\omega\prime}$ være relasjonen over $\mathcal{G}_{\hat{\Sigma}}$ slik at*

$$g \simeq^{\omega\prime} g' \;\Leftrightarrow\; \left\{ \begin{array}{l} g, g' \text{ er av samme type} \quad \text{og} \\ \neg\exists c \in \mathcal{G}_{\hat{\Sigma}}; g_1, g_2 \in \mathcal{G}_{\Sigma^c}/\mathit{synt} \mid c[g] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1 \neq g_2 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g'] \end{array} \right. \tag{3.10}$$

For spesialtilfellet ASSS*, har vi for $\simeq^\omega$ i (3.8):*

$$\simeq^{\omega\prime} = \simeq^\omega$$

Bevis: Vi antar g, g' av samme type, ellers følger teoremet trivielt. Merk at vi ved sats 3.2 (side 66) har at *synt* er en klasserepresentant-funksjon; og siden E_s er en algebraisk spesifikasjon av *synt*, spesifikt en klasserepresentant-funksjon for $\simeq^s$. Følgelig eksisterer *synt*-reduksjonen $\mathcal{G}_{\Sigma^c}/\mathit{synt}$.

Anta $g \not\simeq^{\omega\prime} g'$. Da finnes $c \in \mathcal{G}_{\hat{\Sigma}}$ og $g_1, g_2 \in \mathcal{G}_{\Sigma^c}/\mathit{synt}$ slik at

$$c[g] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1 \neq g_2 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g']$$

Nå er g_1 og g_2 klasserepresentanter for $\simeq^s$. Siden $g_1 \neq g_2$ må $g_1 \not\simeq^s g_2$, ellers var ikke *synt* en klasserepresentant-funksjon for $\simeq^s$. Men da har vi

$$c[g] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1 \not\simeq^s g_2 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g']$$

og dermed $g \not\simeq^{\omega} g'$.

Anta så $g \not\simeq^{\omega} g'$. M.a.o. det finnes $c \in \mathcal{G}_{\hat{\Sigma}}$ og $g_1, g_2 \in \mathcal{G}_{\Sigma^c}$ slik at

$$c[g] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1 \not\simeq^{s} g_2 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g']$$

Nå er ikke nødvendigvis $g_1, g_2 \in \mathcal{G}_{\Sigma^c}/\mathit{synt}$, men siden E_s er en spesifikasjon av *synt* og *synt* er en klasserepresentant-funksjon for $\simeq^s$, har vi at det finnes $g_1', g_2' \in \mathcal{G}_{\Sigma^c}/\mathit{synt}$ slik at $g_1' \neq g_2'$ og slik at $s(g_1) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1'$ og $s(g_2) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_2'$, og vi får

$$s(c[g]) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_1) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1' \neq g_2' \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_2) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(c[g'])$$

og dermed $g \not\simeq^{\omega\prime} g'$.

□

«Ankerfestene» for (3.10) er $\mathcal{G}_{\Sigma^c}/\mathit{synt}$ med fri (preprogrammert) semantikk. Resolusjon i den formelle datatypen $\mathcal{G}_{\Sigma}/\simeq^{\omega\prime}$ kan så gjøres, under `TΣ̂K` og `IND-KONSERV`, ved f.eks. metodene for basis-finalsemantikk beskrevet i [Lys92] og [Lys94a] for spesialtilfellet fri kjerne-semantikk.

«Ankerfestene» for (3.10) «fungerer» ikke med mindre $\hat{E}$ er tilstrekkelig $\{s\}$-komplett mhp. $\mathcal{G}_{\Sigma^c}$; altså med mindre vi antar `TsK`:

Eksempel 51 For E_{synt} fra eksempel 47 er «ankerfestene» i $\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$ til ingen nytte. Finalsemantikken på $\mathcal{G}_{\mathsf{Int}}$ relativ til den frie semantikk i $\mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$ blir den universelle.

○

Pga. den frie kjernesemantikken i (3.10), kan man videre bli fristet til å forenkle (3.10) til

$$g \simeq^{\omega\prime\prime} g' \;\Leftrightarrow\; \begin{cases} g, g' \text{ er av samme type} & og \\ \neg\exists c \in \mathcal{G}_{\hat{\Sigma}} \mid c[g] \overset{*}{\underset{\hat{E}}{\not\leftrightarrow}} c[g'] \end{cases} \tag{3.11}$$

Men for at (3.11) skal tilsvare (3.10), må $\hat{E}$ være tilstrekkelig Σ^d-komplett mhp. *synt-reduksjonen* $\mathcal{G}_{\Sigma^c}/\mathit{synt}$ for ikke umiddelbart å motsi $\simeq^s$:

Eksempel 52 La E være unionen av den algebraiske spesifikasjonen av $\delta_{\mathcal{I}nt}$ fra eksempel 32 på side 64, med f.eks. $E^+_{\mathbb{Z}}$ fra eksempel 37 side 71. Vi har da

$$\mathsf{succ(pred(0))+0} \overset{*}{\underset{E}{\leftrightarrow}} \mathsf{succ(pred(0))} \quad \text{og} \quad \mathsf{0+0} \overset{*}{\underset{E}{\leftrightarrow}} \mathsf{0}$$

som gir $\mathsf{succ(pred(0))}$ ikke semantisk lik $\mathsf{0}$ ifølge (3.11). Denne ulikheten spesifiseres *ikke* i følge (3.10), siden $\mathsf{succ(pred(0))} \notin \mathcal{G}_{\mathsf{Int}}/\delta_{\mathcal{I}nt}$. (Det hjelper ikke å istedet bruke $E'_{+_{\mathbb{Z}}}$ fra eksempel 38 side 71.)

○

Kommentar:

> Finalsemantikken i (3.10) er relativ til fri semantikk på en en-til-en termmengde av klasserepresentanter — bildet av en semantikk-givende syntaktisk funksjon. Det er flere måter å betrakte en slik mengde på:
>
> - Som et «skyggeunivers» bestående av kopier av klasserepresentantene og av en ny type. Hvis f.eks. E er en algebraisk beskrivelse av $\delta_{\mathcal{I}nt}$ fra eksempel 32, og $\mathsf{delta : int \rightarrow int}$ er profilen tolket til δ, definerer vi en type $\underline{\mathsf{int}} \neq \mathsf{int}$ og profiler

$\underline{0}$: $\underline{\mathsf{int}}$,
$\underline{\mathsf{succ}}$: $\underline{\mathsf{int}} \longrightarrow \underline{\mathsf{int}}$
$\underline{\mathsf{pred}}$: $\underline{\mathsf{int}} \longrightarrow \underline{\mathsf{int}}$

Profilen til delta endres til

delta : int$\longrightarrow\underline{\mathsf{int}}$

Evt. hjelpefunksjonprofiler samt E endres i tråd med dette. I forbindelse med finalsemantikken i (3.10), gjør dette her delta til en observator i tradisjonell forstand, som beskrevet i avsnitt 2.3.3. Merk at termer med nestinger av delta nå ikke er velformede.

- Som en delmengde av mange-til-en termuniverset. Fortsetter vi eksemplet over, innføres da en type int$' \preceq$ int, og profilen til delta endres til delta : int$\longrightarrow$int$'$. Nå er nesting av delta lovlig, men en ligning som succ(pred(0)) = 0 er ikke lovlig ifølge tradisjonelle typingsregler siden ligningen innebærer at succ(pred(0)) er av type int$'$, som jo er galt. Merk likevel at $\Phi_{\mathcal{T}}^{\mathcal{I}nt}(\mathsf{int}) = \Phi_{\mathcal{T}}^{\mathcal{I}nt}(\mathsf{int}')$, for standard-typetolken $\Phi_{\mathcal{T}}^{\mathcal{I}nt}$. Derfor gir ligningen succ(pred(0)) = 0 likevel semantisk mening.
- Klasserepresentantene kan spesifiserers til å være av en type-sum T+U. Vi har ikke snakket om type-sum og sum-algebraer, men motstykket i programmeringsspråk ivaretas av konstruksjonen ofte kalt *union*. Litt upresist betyr dette for vårt eksempel at f.eks. termen 0 kan sees som av type int *og* av type int$'$. Da er en ligning som succ(pred(0)) = 0 lovlig.
- Ingen ny type brukes. Profiler og den algebraiske beskrivelsen av den syntaktiske funksjonen endres ikke.

Disse fire punktene representerer forskjellige måter å betrakte mengder av klasserepresentanter på og går fra svært eksplisitt i første punkt til helt implisitt i siste punkt. Vi skal imidlertid være pragmatiske og kun være interessert i hvilke syntaktiske forskjeller disse betraktnings-måtene gir og hvordan disse forskjellene innvirker på spesifikasjon og resolusjon av semantikk. Vi har hittil ført diskusjonen i tråd med siste punkt. Dette skal vi foreløpig fortsette med.

3.4.5 Reduksjon til basis-initialsemantikk

En initialsemantikk relativ til en kjernesemantikk $\simeq^x$ er en omskrivningsrelasjon. Men en slik initialsemantikk er ikke generelt tilgjengelig for eksisterende resolusjons-metoder, da disse er for basis-initialsemantikk. Vårt tilfelle — initialsemantikk relativ til en indirekte semantikk $\simeq^s$ — er imidlertid slik at en enkel utvidelse av ligningsmengden involvert opplagt antyder muligheten for tilbakeføring til basis-initialsemantikk og eksisterende resolusjonsmetoder.

Vi antar fortsatt signaturer og ligningsmengder som definert i figur 3.7 på side 85. Vi antar også fortsatt `INDDEGEN` og `INDKONG`.

La $\simeq^\alpha$ være den initielle semantikk relativ til $\simeq^s$ spesifisert av $\hat{\Sigma}$ og $\hat{E}$ (se definisjon 2.9 av initialsemantikk side 31). Betrakt en $\simeq^\alpha$-utledning i $\mathcal{G}_{\hat{\Sigma}}$ på formen

$$\langle g, \ldots, c[g_1], c[g_2], \ldots, g' \rangle$$

der

$$g \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g_1], \quad g_1 \simeq^s g_2, \quad c[g_2] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g'$$

Nå betyr $g_1 \simeq^s g_2$ at $s(g_1) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_2)$. Betrakter vi nå ligningsmengden

$\hat{E}^{id} = \hat{E} \cup \{s(x) = x\}$ — ***s-id-utvidelsen*** av $\hat{E}$

kan vi umiddelbart lage $\hat{E}^{id}$-utledningen i $\mathcal{G}_{\hat{\Sigma}}$

$$\langle g, \ldots, c[g_1], c[s(g_1)], c[s(g_2)], c[g_2], \ldots, g' \rangle$$

Det er derfor ikke fullstendig unaturlig å tenke seg muligheten for at

$$\simeq^{\alpha} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}} \tag{3.12}$$

Resolusjon i $\mathcal{G}_{\hat{\Sigma}}/\simeq^{\alpha}$ kan i så fall overføres til resolusjon i basis-datatypen $\mathcal{G}_{\hat{\Sigma}}/\hat{E}^{id}$; og er således i prinsipp tilgjengelig for resolusjons-metoder for basis-initialsemantikk. Desverre skal vi se at (3.12) ikke holder generelt. Likevel kan (3.12) vises å holde i interessante spesialtilfeller og for interessante restriksjoner av $\simeq^{\alpha}$.

Det er kun den ene av inklusjonene i (3.12) som er problematisk. Vi har nemlig uten videre:

Lemma 3.7

$$\simeq^{\alpha} \subseteq \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$$

Bevis: Induksjon på lengden n av en vilkårlig $\simeq^{\alpha}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$.

$n = 1$: Trivielt.

$n = k + 1; k \geq 1$: Da har vi en utledning $\langle g, \ldots, g_k, g' \rangle$. Induksjonshypotesen gir $g \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} g_k$. Anta så $g_k \underset{\hat{E}}{\leftrightarrow} g'$. Trivielt får vi da $g \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} g'$. Anta $g_k = c[g_1]$ og $g' = c[g_2]$ og $g_1 \simeq^{s} g_2$. Vi har ved sistnevnte $s(g_1) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_2)$, og kan da lage $\hat{E}^{id}$-utledningen $\langle g_k = c[g_1], c[s(g_1)], c[s(g_2)], c[g_2] = g' \rangle$, og induksjonssteget følger ved transitivitet.

□

Resten av dette avsnittet tilegner seg til oppgaven å vise

$$\simeq^{\alpha}\mathcal{G}_{\Sigma'} \supseteq \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\Sigma'}$$

og følgelig ved lemma 3.7

$$\simeq^{\alpha}\mathcal{G}_{\Sigma'} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\Sigma'}$$

for forskjellige redukt $\mathcal{G}_{\Sigma'}$ av $\mathcal{G}_{\hat{\Sigma}}$, under forskjellige kriterier. Domenet $\mathcal{G}_{\Sigma'}$ for vår reduksjon til basis-initialsemantikk innskrenkes i takt med lettelse av kriteriene.

• Reduksjon på hele $\mathcal{G}_{\hat{\Sigma}}$:

Vi skal her vise under visse antagelser

$$\simeq^{\alpha}\mathcal{G}_{\hat{\Sigma}} \supseteq \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$$

Vi snakker her hele tiden om ligningen $s(x) = x$. Ikke overraskende trenger vi en antagelse om at den semantikkgivende syntaktiske funksjon som E_s er en beskrivelse av, er en kanonisk-representant funksjon. Det er flere måter dette kan tilkjennegi seg i E_s på, hvorav en er:

`KREP1:` For alle $g_c \in \mathcal{G}_{\Sigma^c}$:

$$s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} s(s(g_c))$$

Antagelsen `KREP1` fordrer at E_s er en algebraisk beskrivelse av en fikspunkt-funksjon (se (3.4) side 64). Antagelsen `INDKONG` fordrer at E_s er en algebraisk

beskrivelse av en semantikkgivende funksjon. Ved sats 3.1 (side 65), er en fikspunktfunksjon som er semantikkgivende også en kanonisk-representant funksjon. Følgelig er E_s en algebraisk beskrivelse av en kanonisk-representant funksjon.

Vi kan også se dette på følgende rent syntaktiske måte: Under $\mathtt{T}s\mathtt{K}$, har vi $s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} \tilde{g}_c$, for en $\tilde{g}_c \in \mathcal{G}_{\Sigma^c}$. Ifølge KREP1 har vi da $s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} s(\tilde{g}_c)$, eller m.a.o. under INDKONG:

$$g_c \simeq^s \tilde{g}_c \tag{3.13}$$

Merk at under $\mathtt{T}s\mathtt{K}$ medfører dessuten KREP1 at E_s er en algebraisk *spesifikasjon* av en kanonisk-representant funksjon (ikke bare en beskrivelse).

Antar man $\mathtt{T}\hat{\Sigma}\mathtt{K}$ kan i noen tilfeller KREP1 verifisereres ved å bekrefte den sterkere påstand at $s(\mathsf{x}) = s(s(\mathsf{x}))$ er en induktiv konsekvens av E_s. Resolusjonsmetoder for basis-initialsemantikk kan tas i bruk til dette. Vi kan nå vise:

Teorem 3.8 *Under antagelsene* $\mathtt{T}\hat{\Sigma}\mathtt{K}$ *og* KREP1 *har vi*

$$\simeq^{\alpha} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$$

Lemma 3.7 gir umiddelbart

$$\simeq^{\alpha} \subseteq \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$$

Den andre inklusjon er altså mer arbeidsom å vise. Vi skal trenge mer notasjon. For en $g \in \mathcal{G}_{\hat{\Sigma}}$, la

- $\overline{g}$ betegne en term i $\mathcal{G}_{\Sigma^c}$ slik at

$$g \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \overline{g}$$

- $\tilde{g}_c$ for en generatorterm g_c, betegne en generatorterm slik at $s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} \tilde{g}_c$.

Under antagelsen $\mathtt{T}\hat{\Sigma}\mathtt{K}$ finnes slike $\overline{g}$ og $\tilde{g}_c$ alltid.

Lemma 3.9 *Anta* $\mathtt{T}\hat{\Sigma}\mathtt{K}$ *og* KREP1*. Da har vi*

$$\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}} \subseteq \simeq^{\alpha}$$

Bevis: Induksjon over lengden n til en vilkårlig $\hat{E}^{id}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$.

$n = 1$: Trivielt har vi $g \simeq^{\alpha} g$.

$n = k + 1; k \geq 1$: Da har vi en $\hat{E}^{id}$-utledning $\langle g, \ldots, g_k, g' \rangle$. Induksjonshypotesen gir $g \simeq^{\alpha} g_k$. Anta $g_k \overset{}{\underset{\hat{E}}{\leftrightarrow}} g'$. Induksjonssteget følger da trivielt. Anta $g_k \overset{}{\underset{\{\overrightarrow{s(x)=x}\}}{\leftrightarrow}} g'$. Da er $g_k = c[g'_k]$ og $g' = c[s(g'_k)]$, eller $g_k = c[s(g'_k)]$ og $g' = c[g'_k]$. For begge tilfeller har vi ved (3.13)

$$g'_k \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \overline{g'_k} \simeq^s \overset{\sim}{\overline{g'_k}} \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(\overline{g'_k}) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g'_k) \tag{3.14}$$

og induksjonssteget følger ved monotonitet mhp. kontekstapplikasjon og transitivitet (og evt. symmetri).

□

Teorem 3.8 følger så ved lemma 3.7 og 3.9.

Er det rimelig å forvente at $\mathtt{T}\hat{\Sigma}\mathtt{K}$ er oppfylt? Det er rimelig å forvente i implementasjonssammenheng at $\mathtt{T}\Sigma\mathtt{K}$ i det minste er oppfylt. Gitt at den indirekte spesifikasjonen i $\hat{E}$ er en algebraisk spesifikasjon av en syntaktisk funksjon, vil også $\mathtt{T}s\mathtt{K}$ være oppfylt. Men det er ikke sikkert at $\mathtt{T}\Sigma^h\mathtt{K}$ gjelder, selv om $\mathtt{T}s\mathtt{K}$ er oppfylt:

Eksempel 53 Betrakt $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77. Her er $E_{\delta_{\mathcal{I}nt}}$ tilstrekkelig {delta}-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$ (dvs. $\mathtt{T}s\mathtt{K}$ er oppfylt). Men $E_{\delta_{\mathcal{I}nt}}$ er ikke tilstrekkelig {−}-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$ ($\mathtt{T}\Sigma^h\mathtt{K}$ er ikke oppfylt), siden f.eks. termen pred(0)−pred(0) ikke kan omskrives ved $E_{\delta_{\mathcal{I}nt}}$ til en term i $\mathcal{G}_{\mathsf{Int}}$. (Men merk at pred(0)−pred(0) aldri vil forekomme som komponent i en $E_{\delta_{\mathcal{I}nt}}$-utledning $\langle \mathsf{delta}(g), \ldots \rangle$.)

Imidlertid kan vi gjøre en for spesifikasjonen av $\delta_{\mathcal{I}nt}$ harmløs utvidelse av $E_{\delta_{\mathcal{I}nt}}$, ved å legge til ligningene:

$$E_{kompl} = \left\{ \begin{array}{l} \mathsf{0{-}pred(x)} = \mathsf{succ(0{-}x)}, \\ \mathsf{pred(x){-}pred(y)} = \mathsf{x{-}y}, \\ \mathsf{x{+}pred(y)} = \mathsf{pred(x{+}y)} \end{array} \right\}$$

Det er ikke vanskelig å overbevise seg om at $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$ er tilstrekkelig $\mathsf{Int} \cup \mathbf{\Delta}_{\mathsf{Int}}$-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$ (dvs. at $\mathtt{T}\hat{\Sigma}\mathtt{K}$ er oppfylt) (ved hjelp av lemma 3.4 side 85). Ved et langt induksjonsbevis med mange induksjonsnivåer, kan vi også vise at

$$\mathsf{delta}(g_c) \overset{*}{\underset{E_{\delta_{\mathcal{I}nt}}}{\leftrightarrow}} \mathsf{delta}(\mathsf{delta}(g_c))$$

(altså at KREP1 er oppfylt). Dermed har vi ved teorem 3.8, for delta-*id*-utvidelsen E^{id} av $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$, at

$$\overset{*}{\underset{E^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}} = \simeq^{\alpha}$$

der $\hat{\Sigma} = \mathsf{Int} \cup \mathbf{\Delta}_{\mathsf{Int}}$, og $\simeq^{\alpha}$ er den initielle semantikk spesifisert av $\hat{\Sigma}$ og $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$ relativ til $\simeq^{\mathsf{delta}}$, hvor $\simeq^{\mathsf{delta}}$ er indirekte spesifisert ved $E_{\delta_{\mathcal{I}nt}}$.
○

Problemet med at $\mathtt{T}\Sigma^h\mathtt{K}$ ikke var oppfylt, løstes i eksempel 53 ved å legge til ligninger slik at $\mathtt{T}\Sigma^h\mathtt{K}$ ble oppfylt. Imidlertid skal vi senere se eksempler der en slik løsning ikke er tilfredstillende. Vi skal derfor nå behandle reduksjon til basis-initialsemantikk for tilfellet $\mathtt{T}s\mathtt{K}$ oppfylt, men $\mathtt{T}\Sigma^h\mathtt{K}$ ikke oppfylt.

• $\mathtt{T}\Sigma^h\mathtt{K}$ ikke oppfylt. Reduksjon på $\mathcal{G}_{\{s\}\cup\Sigma} \subset \mathcal{G}_{\hat{\Sigma}}$:

Anta nå at $\mathtt{T}\Sigma^h\mathtt{K}$ ikke er oppfylt. Da gjelder ikke (3.12):

Eksempel 53 (forts.) Betrakt delta-*id*-utvidelsen $E^{id'}$ av $E_{\delta_{\mathcal{I}nt}}$. Vi har

$$\mathsf{pred(0){-}pred(0)} \underset{E^{id'}}{\leftrightarrow} \mathsf{delta(pred(0){-}pred(0))}$$

Men verken pred(0)−pred(0) eller delta(pred(0)−pred(0)) kan omskrives ved $E_{\delta_{\mathcal{I}nt}}$, så vi har

$$\mathsf{pred(0){-}pred(0)} \overset{*}{\underset{E_{\delta_{\mathcal{I}nt}}}{\not\leftrightarrow}} \mathsf{delta(pred(0){-}pred(0))}$$

Videre kan disse fakta brukes til å overbevise seg om at

$$\mathsf{pred(0){-}pred(0)} \not\simeq^{\alpha} \mathsf{delta(pred(0){-}pred(0))}$$

○

Hvis $\mathtt{T}\Sigma^h\mathtt{K}$ ikke er oppfyllt, er det altså generelt ikke samsvar mellom $\overset{*}{\underset{E^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$ og $\simeq^{\alpha}$; ihvertfall ikke i termer som inneholder symboler fra Σ^h. Det er ikke sikkert dette er så ille. Den semantikken vi her egentlig er interessert i, er jo restriksjonen av $\simeq^{\alpha}$ ($\simeq^{\omega}$) til

$$\simeq^{\alpha}_{\mathcal{G}_{\Sigma}} \quad (\simeq^{\omega}_{\mathcal{G}_{\Sigma}})$$

Vi skal nå vise en restriktert versjon av teorem 3.8 som dekker semantikken vi er interessert i. Vi skal vise under visse antagelser at

$$\overset{*}{\underset{E^{id}}{\leftrightarrow}} \mathcal{G}_{\{s\}\cup\Sigma} = \simeq^{\alpha}_{\mathcal{G}_{\{s\}\cup\Sigma}}$$

Vi skal anta $\mathtt{T}\Sigma\mathtt{K}$ og $\mathtt{T}s\mathtt{K}$, men ikke $\mathtt{T}\Sigma^h\mathtt{K}$. Men det er åpenbart at $\hat{E}$ ikke kan være fullstendig Σ^h-ukomplett, sålenge $\mathtt{T}s\mathtt{K}$ avhenger av symboler i Σ^h.

Eksempel 53 (forts.) Vi har jo for alle $g_c \in \mathcal{G}_{\mathsf{Int}}$ $\mathsf{delta}(g_c) \overset{*}{\underset{E_{\delta_{\mathcal{I}nt}}}{\longleftrightarrow}} g'_c$ for en $g'_c \in \mathcal{G}_{\mathsf{Int}}$ og $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77, og det er lett å se at enhver utledning $\langle \mathsf{delta}(g_c), \ldots, g'_c \rangle$ må ha komponenter med forekomster av symboler fra $\mathbf{\Delta}_{\mathsf{Int}}$. (f.eks. $\langle \mathsf{delta}(\mathsf{succ}(\mathsf{pred}(0))), \#_{\mathsf{s}}(\mathsf{succ}(\mathsf{pred}(0)))\text{–}\#_{\mathsf{p}}(\mathsf{succ}(\mathsf{pred}(0))), \ldots, 0 \rangle$). På den annen side forekommer aldri f.eks. $\mathsf{pred}(0)\text{–}\mathsf{pred}(0)$ i en $E_{\delta_{\mathcal{I}nt}}$-utledning der $\mathsf{delta}(g)$ for en eller annen g, er første komponent.
○

Gitt $\mathtt{T}s\mathtt{K}$, altså, må vi ihvertfall ha såpass «Σ^h-kompletthet» at det abstrakte program E_s fungerer for all riktig input. M.a.o.: Gitt $\mathtt{T}s\mathtt{K}$ må vi ha for $g_c \in \mathcal{G}_{\Sigma^c}$ og $g_h \in \mathcal{G}_{\hat{\Sigma}}$ med forekomster av symboler fra Σ^h:

$$s(g_c) \overset{*}{\underset{E_s}{\longleftrightarrow}} g_h \;\Rightarrow\; \exists g'_c \in \mathcal{G}_{\Sigma^c} \mid g_h \overset{*}{\underset{E_s}{\longleftrightarrow}} g'_c \tag{3.15}$$

Siden vi i en viss forstand har «Σ^h-kompletthet for alle praktiske formål», er det nærliggende å se om å bevisstrategien for beviset av lemma 3.9 på side 92 kan brukes under vår overskrift her. Betrakt nemlig en vilkårlig $\hat{E}^{id}$-utledning $\langle g, \ldots, g' \rangle$, for $g, g' \in \mathcal{G}_{\{s\}\cup\Sigma}$. Anta at

$$\langle \ldots, c[g_i], c[s(g_i)], \ldots \rangle$$

er et enkeltsteg i utledningen, der $c[g_i] \underset{\{s(x)=x\}}{\longleftrightarrow} c[s(g_i)]$. I tråd med (3.14) side 92, ser vi at vi bare må godtgjøre at enhver slik term g_i er omskrivbar i $\hat{E}$ til en generatorterm $\overline{g_i}$. Det kunne tenkes at (3.15) sammen med $\mathtt{T}\Sigma\mathtt{K}$ og $\mathtt{T}s\mathtt{K}$, er sterk nok til å garantere dette. Desverre stemmer dette ikke:

Eksempel 53 (forts.) Vi har

$$0 \overset{*}{\underset{E_{\delta_{\mathcal{I}nt}}}{\longleftrightarrow}} 0\text{–}0$$

$$\underset{\{\mathsf{delta}(x)=x\}}{\longleftrightarrow} 0\text{–}\mathsf{delta}(0) \overset{*}{\underset{E_{\delta_{\mathcal{I}nt}}}{\longleftrightarrow}} 0\text{–}\mathsf{delta}(\mathsf{succ}(\mathsf{pred}(0))) \underset{\{\mathsf{delta}(x)=x\}}{\longleftrightarrow}$$

$$0\text{–}\mathsf{succ}(\mathsf{pred}(0)) \underset{\{\mathsf{delta}(x)=x\}}{\longleftrightarrow} \mathsf{delta}(0\text{–}\mathsf{succ}(\mathsf{pred}(0)))$$

for $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77. Men termen $0\text{–}\mathsf{succ}(\mathsf{pred}(0)))$ er ikke omskrivbar i $E_{\delta_{\mathcal{I}nt}}$ til noen term i $\mathcal{G}_{\mathsf{Int}}$.
○

Vi må derfor benytte et mer raffinert resonnement. Vi skal fortsatt ha i sikte å bruke bevisstrategien for beviset av lemma 3.9.

Betrakt da en vilkårlig $\hat{E}^{id}$-utledning $\langle g, \ldots, g' \rangle$ for $g, g' \in \mathcal{G}_{\{s\}\cup\Sigma}$, med enkeltsteg på formen $\langle \ldots, c[g_i], c[s(g_i)], \ldots \rangle$, der $c[g_i] \underset{\{s(x)=x\}}{\longleftrightarrow} c[s(g_i)]$, og slik at at g_i *ikke* er omskrivbar i $\hat{E}$ til en generatorterm $\overline{g_i}$. Vi skal vise at utledningen $\langle g, \ldots, g' \rangle$ kan transformeres til en $\hat{E}^{id}$-utledning $\langle g, \ldots, g' \rangle'$, der enhver g'_i i et enkeltsteg på formen $\langle \ldots, c[g'_i], c[s(g'_i)], \ldots \rangle'$ *er* omskrivbar i $\hat{E}$ til en generatorterm $\overline{g'_i}$.

For å gjøre ting mer oversiktlig, skal vi begrense situasjonen ved følgende antagelse:

*h*ROT: Enhver ligning $l = r \in E_s$ er slik at l og r hver har høyst én forekomst av et symbol fra Σ^h, og dette symbolet er i så fall rot i l hhv. r.

Kriteriet *h*ROT gir en ganske kraftig innsnevring av den generelle situasjon. F.eks. tilfredstiller ikke $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77 *h*ROT. Imidlertid skal vi senere se andre eksempler på indirekte spesifikasjoner som tilfredstiller dette kriteriet (og for hvilke denne diskusjonen er interessant, siden de ikke tilfredstiller $\mathtt{T}\Sigma^h\mathtt{K}$). Disse spesifikasjoner vil dessuten tilfredstille

$\Sigma^h = \{h\}$: Σ^h inneholder kun én funksjonsprofil.

Vi skal også se at en med $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 ekvivalent indirekte spesifikasjon kan gis som tilfredstiller hROT (og $\Sigma^h = \{h\}$).

Før vi går videre, skal vi danne oss et nytt begrep: Vi skal ha nytte av å snakke om ***utledningsforekomster*** av funksjonssymboler i en utledning. Idéen er å holde rede på eller *merke* forekomster av funksjonssymboler gjennom en utledning på en slik måte som f.eks. å snakke om at en utledningsforekomst f^* av et funskjonssymbol f *oppstår* i en komponent g_i i utledningen for senere å *opphøre* i en komponent g_{i+k}. Det er da nødvendig å finne en måte å identifisere entydig utledningsforekomsten f^* i komponentene $g_{i+j}; i \leq j \leq i+k$. Dette er ikke problematisk, og flere merkingsstrategier er mulige. Hvilken vi for vårt formål velger er ikke viktig, men vi skal presisere til en viss grad:

Anta en E-utledning U med et enkeltsteg $\langle \ldots, g, g', \ldots \rangle$. Anta for et funksjonssymbol f at g' har flere forekomster av f enn g. Leser vi U fra venstre mot høyre, sier vi at utledningsforekomster av f har ***oppstått*** i g'. Leser vi U fra høyre mot venstre, sier vi at utledningsforekomster av f har ***opphørt*** i g. Den symmetriske variant der g har flere forekomster av f enn g' er opplagt.

Anta så at regelen $l = r \in E$ ble brukt ved omskrivingen mellom g og g'. I tillegg skal vi dersom l har flere forekomster av f enn r, si at utledningsforekomster av f har opphørt i g'. Merk at g og g' i dette tilfellet kan ha like mange forekomster av f; det har da oppstått like mange utledningsforekomster av f som det har opphørt i g'. Spesifikt *hvilke* utledningsforekomster som har opphørt og oppstått avhenger av den aktuelle merkingsstrategien. For E_s i forbindelse med hROT er dette entydig: Anta $g|p = l\sigma$ og $l = h(t_1, \ldots, t_n)$ for en $h \in \Sigma^h$. Da er $g = g[h^*(g_1, \ldots, g_n)]_p$ (for $g_i = t_i\sigma; 1 \leq i \leq n$), for en utledningsforekomst h^* av h. Dersom r ikke har noen forekomst av h, følger det da at utledningsforekomsten h^* i g opphører i g'.

I motsatt fall; altså hvis r har en forekomst av h, har r ifølge hROT sin ene forekomst av h som rot. En naturlig merkingsstrategi ville innebære at utledningsforekomsten $h^\dagger$ i $g' = g[h^\dagger(g'_1, \ldots, g'_n)]_p$ da sees å være den ***samme*** utledningsforekomst som h^*. Begrepet 'samme utledningsforekomst' kan defineres på mange måter avhengig av den aktuelle merkingsstrategi. Detaljer er ikke viktige for oss her.

Imidlertid har begrepet 'samme utledningsforekomst' for to forekomster av et funksjonssymbol i to nabokomponenter i en utledning, en opplagt transitiv forlengelse. Vi skal i lys av dette si at en utledningsforekomst h^* i en term g er ***opphørbar*** ved en $E' \subseteq E$, dersom det finnes en E'-utledning $\langle g, \ldots, g' \rangle$ for en g' slik at h^* opphører i g'. Dersom en utledningsforekomst i en term g ikke er opphørbar (ved E'), sier vi at forekomsten er en ***(E'-)kritisk*** forekomst i g.

*

Vi har nå som overordnet mål å vise:

Teorem 3.10 *Anta* TsK, TΣK *og* KREP1. *Anta i tillegg* E_sVARBEVAR, E^dVARBEVAR *og* DISJ, *samt* hROT. *Da har vi*

$$\overset{*}{\underset{E^{id}}{\longleftrightarrow}} \mathcal{G}_{\{s\} \cup \Sigma} = \simeq^{\alpha}_{\mathcal{G}_{\{s\} \cup \Sigma}}$$

Nå gir lemma 3.7 umiddelbart

$$\simeq^{\alpha}_{\mathcal{G}_{\{s\}\cup\Sigma}} \subseteq \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}\mathcal{G}_{\{s\}\cup\Sigma}$$

Det er den andre inklusjonen vi skal arbeide med. Vi viser først at den ovenfornevnte transformasjon av utledninger er mulig:

Lemma 3.11 *Anta* `TsK` *og* `TΣK`. *Anta i tillegg* E_s`VARBEVAR`, E^d`VARBEVAR` *og* `DISJ`, *samt* h`ROT`. *En vilkårlig* $\hat{E}^{id}$*-utledning* $\langle g, \ldots, g'\rangle$ *for* $g, g' \in \mathcal{G}_{\{s\}\cup\Sigma}$, *kan transformeres til en* $\hat{E}^{id}$*-utledning* $\langle g, \ldots, g'\rangle'$, *der alle* g_i *i enkeltsteg på formen*

$$\langle \ldots, c[g_i], c[s(g_i)], \ldots\rangle' \quad og \quad \langle \ldots, c[s(g_i)], c[g_i], \ldots\rangle'$$

der $c[g_i] \underset{\{s(x)=x\}}{\leftrightarrow} c[s(g_i)]$, *er omskrivbare i* $\hat{E}$ *til en generatorterm* $\overline{g_i}$.

Vi trenger følgende lemma:

Lemma 3.12 *Anta* E_s`VARBEVAR`, E^d`VARBEVAR` *og* `DISJ`. *Anta også* h`ROT`.

Anta

$$g = c[h(g_1, \ldots, g_n)] \underset{\{l=r\}}{\leftrightarrow} c'[h(g'_1, \ldots, g'_n)] = g'$$

ved en ligning $l = r \in \hat{E}^{id}$, *slik at de viste forekomstene av* h *i* g *og* g' *er samme utledningsforekomst* h^*. *Da har vi*
enten:

1.

$$h^*(g_1, \ldots, g_n) \underset{\{l=r\}}{\leftrightarrow} h^*(g'_1, \ldots, g'_n) \quad og \quad c = c'$$

eller:

2.

$$h^*(g_1, \ldots, g_n) = h^*(g'_1, \ldots, g'_n) \quad og \quad c[t] \underset{\{l=r\}}{\leftrightarrow} c'[t]$$

for en vilkårlig *term* t.

M.a.o.: Den omskriving som skjer mellom g *og* g', *skjer enten lokalt i subtermen* $h^*(g_1, \ldots, g_n)$ *og uten kraft av at* $h^*(g_1, \ldots, g_n)$ *er i konteksten* c, *eller omskrivingen skjer i konteksten* c *og uten kraft av at* $h^*(g_1, \ldots, g_n)$ *er en subterm i* g; *eller kan erstattes ved en omskrivning ved* $l = r$ *som er slik.*

Bevis: Det finnes altså en ligning $l = r$ $(r = l) \in \hat{E}$ slik at $g|p = l\sigma$ for en posisjon p i g og en substitusjon σ. La q være posisjonen slik at $g|q = h^*(g_1, \ldots, g_n)$.

Anta først at q er en disjunkt posisjon fra p; dvs. $g|p$ og $g|q$ er ikke subtermer av hverandre. Da er det opplagt at omskrivingen ikke berører $h^*(g_1, \ldots, g_n)$ og at 2 gjelder.

Anta så at $g|p$ er en ekte subterm av $g|q$. Da skjer omskrivingen i en av $g_1, \ldots, g_n$. Dvs. $g_i|p' = l\sigma$ for en $1 \leq i \leq n$ og en posisjon p', og $g'_i = g_i[r\sigma]_{p'}$. Det er da opplagt at 1 gjelder.

Anta at $g|q$ er en ekte subterm av $g|p$. Vi har to tilfeller: l har forekomster av h, og l har ikke forekomster av h. Anta først at l har forekomster av h. Ved `DISJ` må $l = r \in E_s$. Ved h`ROT` har l kun én forekomst av h, og den er rot i l. Siden $g|q$ er en ekte subterm av $g|p$, må derfor $l = c_l[x]$ for en variabel x og kontekst c_l, slik at $x\sigma = c_h[h(g_1, \ldots, g_n)]$, for en kontekst c_h. Altså

$$l\sigma = c'_l[c_h[h(g_1, \ldots, g_n)]] \tag{3.16}$$

der $c'_l = c_l\sigma$. Ved E_s`VARBEVAR` må da $r = c_r[x]$ for en kontekst c_r. Da er altså

$$r\sigma = c'_r[c_h[h(g_1, \ldots, g_n)]]$$

for $c'_r = c_r\sigma$. Vi har altså at $h^*(g_1,\ldots,g_n) = h^*(g'_1,\ldots,g'_n)$. Siden $h(g_1,\ldots,g_n)$ er substituert inn av σ, er det videre innlysende at vi for en eller annen substitusjon τ, har

$$l\tau = c'_l[c_h[t]] \quad \text{og} \quad c'_r[c_h[t]] = r\tau$$

og dermed $c[t] \underset{\{l=r\}}{\leftrightarrow} c'[t]$ for en vilkårlig term t. Altså er 2 oppfylt.

Anta så at l *ikke* har forekomster av h. Men da må $l = c_l[x]$ for en variabel x og kontekst c_l, slik at $x\sigma = c_h[h(g_1,\ldots,g_n)]$, for en kontekst c_h. Altså har vi (3.16). Vi kan så etablere at 2 er oppfylt, ved et analogt resonnement som over. Men nå er ikke nødvendigvis $l = r \in E_s$, så vi må argumentere med E^d og E^d-`VARBEVAR` i stedet for E_s og E_s`VARBEVAR`, eller med observasjonen at ligningen $s(x) = x$ er variabelbevarende.

Anta til slutt at $p = q$. Vi har pga. `DISJ` og h`ROT` to muligheter: Enten er $l = h(t_1,\ldots,t_n)$, eller $l = x$ for en variabel x. For sistnevnte kan 2 etableres, ved resonnementet over for c_h og c_l tomme.

Dersom nå $l = h(t_1,\ldots t_n)$, er da $l = r \in E_s$ ved `DISJ`. Dersom r ikke har noen forekomst av h, opphører h^* i g'. Dette er mot antagelse, så r må ha en forekomst av h. Ved h`ROT` må videre $r = h(t'_1,\ldots,t'_n)$. Følgelig har vi $h^*(g_1,\ldots,g_n) \underset{\{l=r\}}{\leftrightarrow} h^*(g'_1,\ldots,g'_n)$ og $c = c'$, så 1 er oppfylt.
□

Vi kan nå vise:

Lemma 3.13 *Anta* E_s`VARBEVAR`, E^d`VARBEVAR` *og* `DISJ`. *Anta også* h`ROT`.
Anta en vilkårlig $\hat{E}^{id}$*-utledning*

$$U = \langle c[h^*(g_1,\ldots,g_n)],\ldots,c'[h^*(g'_1,\ldots,g'_n)]\rangle$$

for en utledningsforekomst h^* *av* h. *La* $E(U) \subseteq \hat{E}^{id}$ *være mengden av ligninger brukt ved en omskriving representert av* U.
Da har vi $\hat{E}^{id}$*-utledninger*

$$U_1 = \langle h^*(g_1,\ldots,g_n),\ldots,h^*(g'_1,\ldots,g'_n)\rangle$$

og

$$U_2 = \langle c[t],\ldots,c'[t]\rangle$$

for en vilkårlig t, *på en slik måte at* U_1 *og* U_2 *representerer omskrivinger hvor nøyaktig ligninger fra* $E(U)$ *er brukt, og slik at* $len_{U_1} + len_{U_2} = len_U$. *Videre har samtlige komponenter i* U_1 h^* *som rot.*

Bevis: Induksjon på lengden n av U.

$n = 1$: Trivielt.

$n = k+1; k \geq 1$: Da har vi $\langle c[h^*(g_1,\ldots,g_n)],\ldots,g_k,c'[h^*(g'_1,\ldots,g'_n)]\rangle$. Siden de to viste forekomster av h er samme utledningsforekomst, må (også) $g_k = c_k[h^*(g_{1k},\ldots,g_{nk})]$.

La $E(U_k)$ være mengden av ligninger brukt ved en omskriving representert av delutledningen $U_k = \langle c[h^*(g_1,\ldots,g_n)],\ldots,g_k\rangle$. Induksjonshypotesen gir delutledninger

$$U_{k1} = \langle h^*(g_1,\ldots,g_n),\ldots,h^*(g_{1k},\ldots,g_{nk})\rangle$$

og

$$U_{k2} = \langle c[t],\ldots,c_k[t]\rangle$$

for en vilkårlig t, slik at U_{k1} og U_{k2} representerer omskrivinger hvor nøyaktig ligninger fra $E(U_k)$ er brukt, og slik at $len_{U_{k1}} + len_{U_{k2}} = len_{U_k}$. Samtlige komponenter i U_{k1} har dessuten h^* som rot.

Betrakt nå enkeltsteget

$$\langle \ldots, c_k[h^*(g_{1_k}, \ldots, g_{n_k})], c'[h^*(g'_1, \ldots, g'_n)] \rangle$$

La $l = r$ være ligningen brukt i omskrivingen representert av U, slik at

$$c_k[h^*(g_{1_k}, \ldots, g_{n_k})] \underset{\{l=r\}}{\leftrightarrows} c'[h^*(g'_1, \ldots, g'_n)]$$

Ved lemma 3.12 har vi nå: Enten

$$h^*(g_{1_k}, \ldots, g_{n_k}) \underset{\{l=r\}}{\leftrightarrows} h^*(g'_1, \ldots, g'_n) \quad \text{og} \quad c_k = c'$$

Da har vi umiddelbart ved induksjonshypotesen og transitivitet en utledning

$$U_1 = \langle h^*(g_1, \ldots, g_n), \ldots, h^*(g'_1, \ldots, g'_n) \rangle$$

med lengde $len_{U_{k_1}} + 1$, og slik at hver komponent i U_1 har h^* som rot.

Videre har vi umiddelbart en utledning

$$U_2 = \langle c[t], \ldots, c'[t] \rangle$$

med lengde $len_{U_{k_2}}$, for en vilkårlig t, ved induksjonshypotesen, og siden $c_k = c'$.

Vi har ved induksjonshypotesen nå ialt kun brukt $E(U_k) \cup \{l = r\} = E(U)$. Vi har dessuten $len_{U_1} + len_{U_2} = n = len_U$ og lemmaet følger.

Eller så gir lemma 3.12 at vi har

$$h^*(g_{1_k}, \ldots, g_{n_k}) = h^*(g'_1, \ldots, g'_n) \quad \text{og} \quad c_k[t] \underset{\{l=r\}}{\leftrightarrows} c'[t]$$

for en vilkårlig term t. Umiddelbart har vi da ved induksjonshypotesen

$$U_1 = \langle h^*(g_1, \ldots, g_n), \ldots, h^*(g'_1, \ldots, g'_n) \rangle$$

slik at $len_{U_1} = len_{U_{k_1}}$. Ved induksjonshypotesen har vi trivielt at hver komponent i U_1 har h^* som rot.

Videre gir induksjonshypotesen og transitivitet direkte en utledning

$$U_2 = \langle c[t], \ldots, c'[t] \rangle$$

med lengde $len_{U_{k_2}} + 1$. Igjen har vi ved induksjonshypotesen nå ialt kun brukt $E(U_k) \cup \{l = r\} = E(U)$, og $len_{U_1} + len_{U_2} = n = len_U$, så lemmaet følger.
□

Vi har nå kommet dithen at vi kan presentere hva som skal være et *transformasjons-steg* for lemma 3.11. Betrakt en vilkårlig $\hat{E}^{id}$-utledning $U = \langle g, \ldots, g' \rangle$ for $g, g' \in \mathcal{G}_{\{s\} \cup \Sigma}$. La n være lengden av U. Anta det finnes et enkeltsteg

$$\langle \ldots, c[g_i], c[s(g_i)], \ldots \rangle$$

i U, der $c[g_i] \underset{\{s(x)=x\}}{\leftrightarrows} c[s(g_i)]$, og slik at $g_i = c_{g_i}[h^*(g_1, \ldots, g_n)]$, der utledningsforekomsten h^* av et symbol $h \in \Sigma^h$ ikke er opphørbar ved $\hat{E}$; altså der h^* er en $\hat{E}$-kritisk forekomst i $c[g_i]$.

Siden siste komponent g' i U er i $\mathcal{G}_{\{s\} \cup \Sigma}$ og altså ikke har forekomster av noen $h \in \Sigma^h$, kan vi imidlertid *garantere at utledningsforekomsten h^* vil opphøre før eller siden i en komponent g_{i+k}; $1 \leq k \leq n \Leftrightarrow i$ av U.*

Det er åpenbart at en utledningsforekomst av en $h \in \Sigma^h$ ikke kan opphøre ved regelen $s(x)=x$. Dette innebærer at det finnes en komponent $g_l = c^*[h^*(g^*_1, \ldots, g^*_n)]$ for en $i < l < i + k$, der h^* kan opphøres *ved omskriving i $\hat{E}$*.

Kjernen i transformasjonen er nå denne: Vi har altså i U delutledningen U_{h^*}:

$$\langle \ldots, c[c_{g_i}[h^*(g_1, \ldots, g_n)]],$$
$$c[s(c_{g_i}[h^*(g_1, \ldots, g_n)])], \ldots, c^*[h^*(g_1^*, \ldots, g_n^*)], \ldots \rangle$$

Lemma 3.13 sier da at vi har delutledninger

$$\langle h^*(g_1, \ldots, g_n), \ldots, h^*(g_1^*, \ldots, g_n^*) \rangle$$

der hver komponent har h^* som rot, og

$$\langle c[s(c_{g_i}[h^*(g_1^*, \ldots, g_n^*)])], \ldots, c^*[h^*(g_1^*, \ldots, g_n^*)] \rangle$$

Fra disse delutledninger kan vi så umiddelbart lage delutledningen U'_{h^*}

$$\langle \ldots, c[c_{g_i}[h^*(g_1, \ldots, g_n)]], \ldots, c[c_{g_i}[h^*(g_1^*, \ldots, g_n^*)]],$$
$$c[s(c_{g_i}[h^*(g_1^*, \ldots, g_n^*)])], \ldots, c^*[h^*(g_1^*, \ldots, g_n^*)], \ldots \rangle$$

Vi kan altså i U bytte ut delutledningen U_{h^*} med delutledningen U'_{h^*}. La $E(U_{h^*}) \subseteq \hat{E}^{id}$ være mengden av ligninger brukt ved en omskriving representert av U_{h^*}. Lemma 3.13 gir da dessuten at U'_{h^*} representerer en omskriving hvor nøyaktig ligninger fra $E(U_{h^*})$ er brukt, samt at lengdene til U_{h^*} og U'_{h^*} er identiske. Argumentasjonen er analog for det symmetriske tilfellet

$$\langle \ldots, c[s(g_i)], c[g_i], \ldots \rangle$$

Vi leser da U mot venstre istedenfor. En slik delutlednings-substitusjon utgjør ett transformasjonssteg.

Vi skal nå vise at en prosess bestående av slike transformasjonssteg før eller siden vil eliminere samtlige kritiske forekomster av symboler $h \in \Sigma^h$. Prosessen terminerer når samtlige slike kritiske forekomster er eliminert.

Observasjon 3.14 *Anta en vilkårlig E-utledning $\langle g, \ldots, g' \rangle$ av lengde n, for en endelig ligningsmengde E. En vilkårlig komponent i utledningen har termdybde begrenset av*

$$Kn + d_0$$

hvor K er en konstant avhengig av E og d_0 er termdybden til den av g og g' med minst termdybde.

Betrakt så en vilkårlig $\hat{E}^{id}$-utledning $U_j = \langle g, \ldots, g' \rangle$ for $g, g' \in \mathcal{G}_{\{s\} \cup \Sigma}$. La n være lengden til U_j. La $E(U_j) \subseteq \hat{E}^{id}$ være mengden av ligninger brukt ved en omskriving representert av U_j. Siden U_j er endelig, er $E(U_j)$ endelig.

La max_d være den øvre grense for komponenters termdybde i en vilkårlig $E(U_j)$-utledning av lengde n. (Ved observasjon 3.14 finnes max_d.)

La så U_{j+1} være en $\hat{E}^{id}$-utledning framkommet ved ett transformasjonssteg som over. Merk at U_{j+1} representerer en omskriving hvor nøyaktig ligninger fra $E(U_j)$ er brukt, og merk at lengden til U_{j+1} er n.

Ved observasjon 3.14 er da max_d en øvre grense for komponenters termdybde i både U_j og U_{j+1}. (Merk at ved å anta med rimelighet at $\hat{E}$ er endelig, kunne vi her ha resonnert ved hjelp av en maksdybde utifra $\hat{E}$ istedenfor utifra $E(U_j)$.)

Betrakt så max_d-tuppelet

$$\left\langle n_{1j}, \ldots, n_{max_d j} \right\rangle$$

La nå n_{dj} være det totale antall kritiske forekomster av symboler fra Σ^h som forekommer på dybde d i komponenter $c[g_i]$ av U_j, der

$$c[g_i] \underset{\{s(x)=x\}}{\longleftrightarrow} c[s(g_i)]$$

Vi skal vise at

$$\left\langle n_{1j+1}, \ldots, n_{max_dj+1} \right\rangle \prec_{lex} \left\langle n_{1j}, \ldots, n_{max_dj} \right\rangle \tag{3.17}$$

der $\prec_{lex}$ er en ***leksikografisk ordning*** på $\mathbb{N}^{max_d}$, som følger:

$$\begin{gathered}\langle n_1, \ldots, n_{max_d} \rangle \prec_{lex} \langle n'_1, \ldots, n'_{max_d} \rangle \\ \Updownarrow \\ n_1 < n'_1 \\ \text{eller} \\ n_1 = n'_1, \ldots, n_i = n'_i; 1 \leq i < max_d \;\Rightarrow\; n_{i+1} < n'_{i+1}\end{gathered}$$

Det er velkjent at denne ordningen er en velfundert relasjon.

La oss betrakte transformasjonen over av delutledningen U_{h^*} til delutledningen U'_{h^*}. Vi antar at U_{h^*} er en delutledning i en U_j og at U'_{h^*} er en delutledning i U_{j+1}. Merk at U_j og U_{j+1} er identiske, utenom delutledningene U_{h^*} og U'_{h^*}.

Anta h^* forekommer på dybde d i $c[g_i]$. Det er da klart at

$$n_{dj+1} = n_{dj} \Leftrightarrow 1$$

Da vil (3.17) holde, dersom

$$n_{d'j+1} \leq n_{d'j} \tag{3.18}$$

for enhver $d' < d$. Vi godtgjør nå (3.18) ved å inspisere delen av U_{j+1} som er forskjellig fra U_j.

Anta at det er kritiske forekomster av symboler $h \in \Sigma^h$ i komponenter i delutledningen

$$\langle \ldots, c[c_{g_i}[h^*(g_1, \ldots, g_n)]], \ldots, c[c_{g_i}[h^*(g^*_1, \ldots, g^*_n)]], \ldots \rangle$$

av U'_{h^*}. Disse kritiske forekomster har korresponderende kritiske forekomster i U_{h^*}. Disse kan i U'_{h^*} ha fått mindre dybde, siden den viste anvendelsen av $s(x)=x$ er «forsinket». Men ved at delutledningen over er konstruert ved delutledningen

$$\langle h^*(g_1, \ldots, g_n), \ldots, h^*(g^*_1, \ldots, g^*_n) \rangle$$

der hver komponent har h^* som toppsymbol, har vi at de nevnte kritiske forekomster må befinne seg i ekte subtermer av disse komponentene. Følgelig vil disse kritiske forekomster ha dybde større enn d.

Betrakt så eventuelle kritiske forekomster i $g^*_1, \ldots, g^*_n$ introdusert ved steget

$$\langle c[c_{g_i}[h^*(g^*_1, \ldots, g^*_n)]], c[s(c_{g_i}[h^*(g^*_1, \ldots, g^*_n)])] \rangle$$

i U'_{h^*}. Disse har også dybde større enn d.

Anta til slutt at det er kritiske forekomster av symboler $h \in \Sigma^h$ i komponenter i delutledningen

$$\langle c[s(c_{g_i}[h^*(g^*_1, \ldots, g^*_n)])], \ldots, c^*[h^*(g^*_1, \ldots, g^*_n)] \rangle$$

Her foregår all omskriving alle andre steder enn i $h^*(g^*_1, \ldots, g^*_n)$. Enhver kritisk forekomst har da en korresponderende kritisk forekomst på samme dybde i delutledningen U_{h^*}. Altså har vi (3.18).

Vi har dermed vist (3.17). Det er da på det rene at en slik transformasjonsprosess vil terminere med en utledning U_m slik at

$$n_{1\,m} = \cdots = n_{max_d\,m} = 0$$

Vi har nå vist at vi kan eliminere alle kritiske forekomster av symboler $h \in \Sigma^h$. For å etablere lemma 3.11 på side 96, gjenstår det bare å vise følgende forbindelse:

Lemma 3.15 *Anta* `TsK` *og* `TΣK`*. Anta det finnes et enkeltsteg*

$$\langle \ldots, c[g_i], c[s(g_i)], \ldots \rangle \quad \textit{eller} \quad \langle \ldots, c[s(g_i)], c[g_i], \ldots \rangle$$

i en $\hat{E}$-utledning U, der $c[g_i] \underset{\{s(x)=x\}}{\longleftrightarrow} c[s(g_i)]$, og slik at g_i ikke er omskrivbar i $\hat{E}$ til en generatorterm $\overline{g_i}$. Da finnes en kritisk forekomst av en $h \in \Sigma^h$ i $c[g_i]$.

Bevis: Siden vi antar `TsK` og `TΣK`, må det finnes en forekomst av en $h \in \Sigma^h$ i g_i. Spesifikt må $g_i = c_{g_i}[h(g_1, \ldots, g_n)]$ for en kontekst c_{g_i}, slik at $h(g_1, \ldots, g_n)$ ikke er omskrivbar i $\hat{E}$ til en generatorterm $\overline{h(g_1, \ldots, g_n)}$. Det må videre finnes en (blant flere) dypeste slik h; dvs. vi kan anta at $g_1, \ldots, g_n$ alle er omskrivbare i $\hat{E}$ til generatortermer $\overline{g_1}, \ldots, \overline{g_n}$. Men da er den viste utledningsforekomsten av h ikke opphørbar ved $\hat{E}$. (Merk at den viste utledningsforekomsten av h godt kunne være opphørbar ved $\hat{E}$, dersom vi ikke betrakter en dypeste h.)
□

Med lemma 3.11 således etablert, kan vi gi oss i kast med beviset av teorem 3.10 på side 95. Teorem 3.10 følger ved neste lemma som følger strategien i beviset av lemma 3.9 side 92:

Lemma 3.16 *Anta* `TsK`*,* `TΣK` *og* `KREP1`*. Anta i tillegg E_s*`VARBEVAR`*, E^d*`VARBEVAR` *og* `DISJ`*, samt h*`ROT`*. Da har vi*

$$\overset{*}{\underset{\hat{E}^{id}}{\longleftrightarrow}} \mathcal{G}_{\Sigma \cup \{s\}} \subseteq \simeq^{\alpha} \mathcal{G}_{\Sigma \cup \{s\}}$$

Bevis: La $\langle g, \ldots, g' \rangle$ være en vilkårlig $\hat{E}$-utledning i $\mathcal{G}_{\hat{\Sigma}}$ for $g, g' \in \mathcal{G}_{\{s\} \cup \Sigma}$. Ved lemma 3.11 finnes en $\hat{E}^{id}$-utledning $\langle g, \ldots, g' \rangle'$, der alle g'_i i enkeltsteg

$$\langle \ldots, c[g'_i], c[s(g'_i)], \ldots \rangle' \quad og \quad \langle \ldots, c[s(g_i)], c[g_i], \ldots \rangle'$$

der $c[g_i] \underset{\{s(x)=x\}}{\longleftrightarrow} c[s(g_i)]$, er omskrivbare i $\hat{E}$ til en generatorterm $\overline{g'_i}$.

Vi induserer nå over lengden n til $\langle g, \ldots, g' \rangle'$.

$n = 1$: Trivielt har vi $g \simeq^{\alpha} g$.

$n = k + 1; k \geq 1$: Da har vi $\langle g, \ldots, g_k, g' \rangle'$. Induksjonshypotesen gir $g \simeq^{\alpha} g_k$. Anta $g_k \overset{}{\underset{\hat{E}}{\leftrightarrow}} g'$. Induksjonssteget følger da trivielt. Anta $g_k \underset{\{s(x)=x\}}{\longleftrightarrow} g'$. Da er $g_k = c[g'_k]$ og $g' = c[s(g'_k)]$, eller $g_k = c[s(g'_k)]$ og $g' = c[g'_k]$. For begge tilfeller har vi nå ved (3.13) på side 92:

$$g'_k \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \overline{g'_k} \simeq^{s} \overset{\sim}{\overline{g'_k}} \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(\overline{g'_k}) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g'_k)$$

og induksjonsteget følger ved monotonitet mhp. kontekstapplikasjon og transitivitet (og evt. symmetri).
□

Dette konkluderer vår behandling av reduksjon til basis-initialsemantikk for tilfellet `TΣ`^{h}`K` ikke oppfylt, men likevel `TsK` oppfylt.

• TsK ikke oppfylt. Reduksjon på $\mathcal{G}_\Sigma \subset \mathcal{G}_{\hat{\Sigma}}$:

Hva nå hvis TsK ikke er tilfredsstilt? Vi viser at (3.12) på side 91 da umiddelbart kan gjendrives ved å finne et eksempel på at

$$g \overset{}{\underset{E^{id}}{\leftrightarrow}} s(g) \quad \text{men} \quad g \not\simeq^{\alpha} s(g)$$

Eksempel 54 Betrakt $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77. La her

$$E_{\mathsf{delta}} = E_{\delta_{\mathcal{I}nt}} \setminus \{\mathsf{x{-}0} = \mathsf{x}, \mathsf{0{-}succ(x)} = \mathsf{pred(0{-}x)}\}$$

E_{delta} er altså $E_{\delta_{\mathcal{I}nt}}$ gjort fullstendig $\{\mathsf{delta}\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$. Vi påstår at E_{delta} er indirekte $\mathcal{G}_{\mathsf{Int}}$-kongruent, så den indirekte semantikk $\simeq^{\mathsf{delta}}$ finnes. La $\simeq^{\alpha}$ betegne initialsemantikken relativ til $\simeq^{\mathsf{delta}}$. Vi har

$$\mathsf{succ(pred(0))} \underset{E^{id}_{\mathsf{delta}}}{\leftrightarrow} \mathsf{delta(succ(pred(0)))}$$

For å se at

$$\mathsf{succ(pred(0))} \not\simeq^{\alpha} \mathsf{delta(succ(pred(0)))}$$

observer først at for alle $g_c \in \mathcal{G}_{\mathsf{Int}}$, må enhver g slik at $\mathsf{delta}(g_c) \overset{*}{\underset{E_{\mathsf{delta}}}{\leftrightarrow}} g$, ha delta eller – som rot. Dette gjelder også om generatortermer byttes med andre generatortermer underveis i utledningen; dvs. at dette gjelder for alle $g_c \in \mathcal{G}_{\mathsf{Int}}$ og g slik at $\mathsf{delta}(g_c) \simeq^{\alpha} g$.

Observer deretter at E_{delta} er fullstendig $\{\mathsf{delta}, -\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$. (Man blir altså ikke kvitt funksjonssymbolet delta.)
○

Hvis TsK ikke er oppfyllt er det altså generelt ikke samsvar mellom $\overset{*}{\underset{E^{id}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$ og $\simeq^{\alpha}$; ihvertfall ikke i termer som inneholder det spesifiserende symbol s. Igjen er dette ikke så ille: Vi er interessert i restriksjonen av $\simeq^{\alpha}$ til

$$\simeq^{\alpha}_{\mathcal{G}_\Sigma}$$

Vi skal igjen vise en restriktert versjon av teorem 3.8 som dekker semantikken vi er interessert i. Vi antar ikke TsK og heller ikke $\mathtt{T}\Sigma^h\mathtt{K}$. Vi antar dog TΣK. La oss først betrakte situasjonen for $\Sigma^h = \emptyset$.

- Tilfellet $\Sigma^h = \emptyset$

Vi trenger mer notasjon:

- La $g_{\restriction s}$ stå for termen identisk med g, men med alle forekomster av det spesifiserende symbol s fjernet.

Vi betrakter følgende antagelse:

KREP2: For alle $g, g' \in \mathcal{G}_{\Sigma^c \cup \{s\}}$:

$$g \overset{*}{\underset{E_s}{\leftrightarrow}} g' \;\Rightarrow\; s(g_{\restriction s}) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g'_{\restriction s})$$

Antagelsen KREP2 er som KREP1 en manifestasjon av at den syntaktiske funksjonen som E_s er en algebraisk beskrivelse av, er en kanonisk-representant funksjon.

Teorem 3.17 *Under antagelsene* TΣK, *samt* KREP2, DISJ *og* E_sVARBEVAR *har vi for* $\Sigma^h = \emptyset$

$$\simeq^{\alpha}_{\mathcal{G}_\Sigma} = \overset{*}{\underset{E^{id}}{\leftrightarrow}} \mathcal{G}_\Sigma$$

Inklusjonen $\simeq^{\alpha}_{\mathcal{G}_\Sigma} \subseteq \overset{*}{\underset{E^{id}}{\leftrightarrow}}\mathcal{G}_\Sigma$ følger fra lemma 3.7, og igjen er den andre inklusjonen mer vrien. Teorem 3.17 følger da fra lemma 3.7 og følgende lemma:

Lemma 3.18 *Anta* TΣK, KREP2, DISJ *og* E_s VARBEVAR. *Da har vi dersom* $\Sigma^h = \emptyset$

$$\overset{*}{\underset{E^{id}}{\leftrightarrow}}\mathcal{G}_\Sigma \subseteq \simeq^{\alpha}_{\mathcal{G}_\Sigma}$$

For å vise lemma 3.18, viser vi:

Lemma 3.19 *Anta* TΣK, KREP2, DISJ *og* E_s VARBEVAR. *Da har vi dersom* $\Sigma^h = \emptyset$

$$g \overset{*}{\underset{E}{\leftrightarrow}} g' \;\Rightarrow\; g_{\natural s} \simeq^{\alpha} g'_{\natural s}$$

for alle $g, g' \in \mathcal{G}_{\hat{\Sigma}}$

Bevis: Anta $g \overset{*}{\underset{E}{\leftrightarrow}} g'$ for vilkårlige $g, g' \in \mathcal{G}_{\hat{\Sigma}}$.

Anta $g \underset{E^d}{\leftrightarrow} g'$. Da har vi $g|p = v\sigma$ og $g' = g[h\sigma]_p$, for en $v{=}h$ eller $h{=}v \in E^d$, posisjon p i g og substitusjon $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\hat{\Sigma}}}$.

For en substitusjon τ, la nå $\tau_{\natural s}$ betegne substitusjonen slik at for alle $x \in \mathcal{V}$:

$$x\tau_{\natural s} = (x\tau)_{\natural s}$$

(husk at en substitusjon τ med domene bæremengden til en term-algebra $\mathcal{T}_\Sigma(\mathcal{V})$, er entydig bestemt av bildet $\tau(\mathcal{V})$). Merk at vi for en term t har

$$t\tau_{\natural s} = (t\tau)_{\natural s} \;\Leftrightarrow\; t_{\natural s} = t \tag{3.19}$$

Nå er altså $(g|p)_{\natural s} = (v\sigma)_{\natural s}$. Ved DISJ finnes ingen forekomster av s i v; m.a.o.: $v_{\natural s} = v$, så ved (3.19) får vi $(g|p)_{\natural s} = v\sigma_{\natural s}$. Det er videre innlysende at det finnes en posisjon q i $g_{\natural s}$ slik at $g_{\natural s}|q = (g|p)_{\natural s}$. Da har vi altså

$$g_{\natural s}|q = v\sigma_{\natural s} \tag{3.20}$$

Betrakt så termen $g_{\natural s}[h\sigma_{\natural s}]_q$. Ved DISJ finnes ingen forekomster av s i h; m.a.o.: $h_{\natural s} = h$, så ved (3.19) får vi $h\sigma_{\natural s} = (h\sigma)_{\natural s}$, og dermed $g_{\natural s}[h\sigma_{\natural s}]_q = g_{\natural s}[(h\sigma)_{\natural s}]_q$. Det er opplagt at $g_{\natural s}[(h\sigma)_{\natural s}]_q = (g[(h\sigma)_{\natural s}]_p)_{\natural s}$, som igjen er identisk med $(g[h\sigma]_p)_{\natural s} = g'_{\natural s}$. Altså har vi

$$g_{\natural s}[(h\sigma)_{\natural s}]_q = g'_{\natural s} \tag{3.21}$$

Tilsammen får vi ved (3.21) og (3.20) at $g_{\natural s} \overset{*}{\underset{E^d}{\leftrightarrow}} g'_{\natural s}$, og altså $g_{\natural s} \simeq^{\alpha} g'_{\natural s}$.

Anta så at $g \underset{E_s}{\leftrightarrow} g'$. Da har vi $g|p = v\sigma$ og $g' = g[h\sigma]_p$, for en $v{=}h$ $(h{=}v) \in E_s$, posisjon p i g og substitusjon $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\hat{\Sigma}}}$. Vi deler opp i de to tilfellene:

1. $g|p \in \mathcal{G}_{\Sigma^c \cup \{s\}}$
2. $g|p \in \mathcal{G}_{\hat{\Sigma}} \setminus \mathcal{G}_{\Sigma^c \cup \{s\}}$

1) Anta først $g|p = v\sigma \in \mathcal{G}_{\Sigma^c \cup \{s\}}$. Ved DISJ og siden $\Sigma^h = \emptyset$, må også $h\sigma \in \mathcal{G}_{\Sigma^c \cup \{s\}}$. Vi har jo $v\sigma \overset{*}{\underset{E_s}{\leftrightarrow}} h\sigma$, så KREP2 gir under INDKONG $(v\sigma)_{\natural s} \simeq^{s} (h\sigma)_{\natural s}$. Nå er $g_{\natural s} = c[(v\sigma)_{\natural s}]$ og $c[(h\sigma)_{\natural s}] = g'_{\natural s}$, for en $c \in \mathcal{G}_{\Sigma^c \cup \Sigma^d}$, og vi får

$$g_{\natural s} = c[(v\sigma)_{\natural s}] \simeq^{\alpha} c[(h\sigma)_{\natural s}] = g'_{\natural s}$$

2) Anta så $g|p \in \mathcal{G}_{\hat{\Sigma}} \setminus \mathcal{G}_{\Sigma^c \cup \{s\}}$. Da er

$$g|p = v\sigma = c[g_{f_1}, \ldots, g_{f_m}]$$

for en $c \in \mathcal{G}_{\Sigma^c \cup \{s\}}$ og $g_{f_1}, \ldots, g_{f_n} \in \mathcal{G}_{\hat{\Sigma}}$, slik at et funksjonssymbol $f_i \in \Sigma^d$ forekommer i g_{f_i} for alle $1 \leq i \leq m$. Husk at $\Sigma^h = \emptyset$. Vi presiserer her $c \in \mathcal{G}_{\Sigma^c \cup \{s\}}$; vi ser altså på termer g_{f_i} på tilstrekkelig liten dybde i $v\sigma$. Vi skal vise at

$$h\sigma = c'[g_{f_1}, \ldots, g_{f_m}]$$

også for en $c' \in \mathcal{G}_{\Sigma^c \cup \{s\}}$. Ved DISJ finnes ingen forekomster av f_i i v; $1 \leq i \leq m$. Derfor må $v = c_v[x_1, \ldots, x_m]$ for variabler $x_1, \ldots, x_m$ og kontekst c_v, slik at for $1 \leq i \leq m$, $x_i\sigma = c_{f_i}[g_{f_i}]$, for en kontekst c_{f_i}. Merk at hver $c_{f_i} \in \mathcal{G}_{\Sigma^c \cup \{s\}}$, siden $c = c_v[c_{f_i}, \ldots, c_{f_m}]$, og $c \in \mathcal{G}_{\Sigma^c \cup \{s\}}$.

Ved E_sVARBEVAR er $h = c_h[x_1, \ldots, x_m]$ for en kontekst c_h. Dermed er $h\sigma = c'_h[c_{f_1}[g_{f_1}], \ldots, c_{f_m}[g_{f_m}]]$ for $c'_h = c_h\sigma$. Ved DISJ er $c_h \in \mathcal{G}_{\Sigma^c \cup \{s\}}$. Følgelig er

$$h\sigma = c'[g_{f_1}, \ldots, g_{f_m}]$$

for en $c' \in \mathcal{G}_{\Sigma^c \cup \{s\}}$.

La nå $\overline{g_{f_i}} \in \mathcal{G}_{\Sigma^c}$ være termen slik at under TΣK, $g_{f_i \natural s} \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \overline{g_{f_i}}$, for $1 \leq i \leq m$. Betrakt substitusjonen $\sigma' \in \mathcal{H}om^{\mathcal{G}_{\Sigma^c \cup \{s\}}}_{\mathcal{T}_{\hat{\Sigma}}(\mathcal{V})}$ slik at $x_i\sigma' = c_{f_i}[\overline{g_{f_i}}]$; $1 \leq i \leq m$ (og identitet for andre variable). Vi har

$$v\sigma' = c_v[c_{f_1}[\overline{g_{f_1}}], \ldots, c_{f_n}[\overline{g_{f_n}}]] = c[\overline{g_{f_1}}, \ldots, \overline{g_{f_n}}]$$

og

$$h\sigma' = c_h[c_{f_1}[\overline{g_{f_1}}], \ldots, c_{f_n}[\overline{g_{f_n}}]] = c'[\overline{g_{f_1}}, \ldots, \overline{g_{f_n}}]$$

Videre har vi selvfølgelig $v\sigma' \overset{*}{\underset{\hat{E}^s}{\leftrightarrow}} h\sigma'$. Ved at $c, c' \in \mathcal{G}_{\Sigma^c \cup \{s\}}$, får vi da ved KREP2

$$(v\sigma')_{\natural s} \simeq^s (h\sigma')_{\natural s}$$

Ialt får vi da

$$(v\sigma)_{\natural s} = c_{\natural s}[g_{f_1 \natural s}, \ldots, g_{f_m \natural s}]$$

$$\overset{*}{\underset{\hat{E}}{\leftrightarrow}} c_{\natural s}[\overline{g_{f_1}}, \ldots, \overline{g_{f_n}}] = (v\sigma')_{\natural s} \simeq^s (h\sigma')_{\natural s} = c'_{\natural s}[\overline{g_{f_1}}, \ldots, \overline{g_{f_n}}] \overset{*}{\underset{\hat{E}}{\leftrightarrow}}$$

$$c'_{\natural s}[g_{f_1 \natural s}, \ldots, g_{f_m \natural s}] = (h\sigma)_{\natural s}$$

Ved kongruens følger da at

$$g_{\natural s} = g_{\natural s}[(v\sigma)_{\natural s}]_q \simeq^\alpha g_{\natural s}[(h\sigma)_{\natural s}]_q = g'_{\natural s}$$

□

Det er nå trivielt å vise:

Lemma 3.20 *Anta* TΣK, KREP2, DISJ *og* E_sVARBEVAR. *Da har vi dersom* $\Sigma^h = \emptyset$

$$g \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} g' \;\Rightarrow\; g_{\natural s} \simeq^\alpha g'_{\natural s}$$

for alle $g, g' \in \mathcal{G}_{\hat{\Sigma}}$

Bevis: Induksjon over lengden n til en vilkårlig $\hat{E}^{id}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$.

$n = 1$: Trivielt har vi $g_{\natural s} \simeq^\alpha g_{\natural s}$.

$n = k + 1; k \geq 1$: Da har vi en $\hat{E}^{id}$-utledning $\langle g, \ldots, g_k, g' \rangle$. Induksjonshypotesen gir $g_{\natural s} \simeq^\alpha g_{k \natural s}$. Anta $g_k \underset{\hat{E}}{\leftrightarrow} g'$. Ved lemma 3.19 har vi $g_{k \natural s} \simeq^\alpha g'_{\natural s}$, og induksjonssteget følger ved transistivitet. Anta $g_k \underset{\{s(x)=x\}}{\longleftrightarrow} g'$. Da er $g_{k \natural s} = g'_{\natural s}$, så induksjonssteget følger trivielt.

□

Lemma 3.18 følger så som et spesialtilfelle av lemma 3.20, og teorem 3.17 følger så ved lemmata 3.7 og 3.18.

Flere av antagelsene vi gjør i dette avsnittet er lette å verifisere og er naturlig oppfylt ved konstruktiv algebraisk spesifikasjon av funksjoner. Antagelsene `KREP1` og `KREP2` kan imidlertid begge innebære bevisbyrder; som dog ikke nødvendigvis er spesielt vanskelige, men gjerne litt langtekkelige. Vi viser et eksempel hvor en mindre langtekkelig godtgjørelse av `KREP2` inngår.

Eksempel 55 Betrakt E_{synt} fra eksempel 47 side 80. La her $\hat{E}$ være E_{synt} i union med f.eks. følgende E^d:

$$E^d = \left\{ \begin{array}{ll} \#_{\mathsf{s}}(0) = 0, & \mathsf{x}{+}0 = \mathsf{x}, \\ \#_{\mathsf{s}}(\mathsf{succ}(\mathsf{x})) = \mathsf{succ}(0){+}\#_{\mathsf{s}}(\mathsf{x}), & \mathsf{x}{+}\mathsf{succ}(\mathsf{y}) = \mathsf{succ}(\mathsf{x}{+}\mathsf{y}) \end{array} \right\}$$

Vi lar $\hat{\Sigma}$ være mengden av alle (profiler til) funskjonssymboler forekommende i $\hat{E}$.

Betrakt initialsemantikken $\simeq^{\alpha}$ relativ til den indirekte semantikk spesifisert av E_{synt} bestemt av $\hat{\Sigma}$ og $\hat{E}$. Vi ønsker å vise at $\simeq^{\alpha}$, eller en fornuftig restriksjon av denne, kan reduseres til en basis-initialsemantikk. Vi lar

$$\Sigma^c = \{0, \mathsf{succ}, \mathsf{pred}\}$$

og

$$\Sigma^d = \Sigma \setminus (\Sigma^c \cup \{\mathsf{synt}\})$$

Det er ingen hjelpe-funksjonssymboler til `synt`, så vår $\Sigma^h = \emptyset$. Siden $\hat{E}$ ikke tilfredstiller `TsK` skal vi appellere til teorem 3.17. Vi «ser lett at» $\hat{E}$ tilfredstiller `TΣK`, `DISJ` og E_s `VARBEVAR`. Vi skal derimot godtgjøre at E_{synt} tilfredstiller `KREP2`:

Godtgjørelse: Siden $\hat{E}$ tilfredstiller `DISJ` og $\Sigma^h = \emptyset$ samt E_s `VARBEVAR`, vil enhver E_{synt}-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$ hvor $g, g' \in \mathcal{G}_{\Sigma^c \cup \{\mathsf{synt}\}}$, være en E_{synt}-utledning i $\mathcal{G}_{\Sigma^c \cup \{\mathsf{synt}\}}$. Vi kan derfor se helt bort ifra E^d og kan enkelt vise at E_{synt} tilfredstiller `KREP2` ved induksjon på lengden av en slik utledning. Induksjonsbasisen er triviell. Induksjonssteget består i å vurdere hver regel i E_{synt}.

Så la $g_k, g' \in \mathcal{G}_{\Sigma^c \cup \{\mathsf{synt}\}}$ slik at $g_k \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} g'$. Anta regelen

$$\mathsf{synt}(\mathsf{succ}(\mathsf{x})) = \mathsf{synt}(\mathsf{succ}(\mathsf{synt}(\mathsf{x})))$$

er brukt. Da har vi $g_k = c[\mathsf{synt}(\mathsf{succ}(g''))]$ og $g' = c[\mathsf{synt}(\mathsf{succ}(\mathsf{synt}(g'')))]$ (eller analogt omvendt). Men da er $g_{k\,\natural_{\mathsf{synt}}} = g'_{\natural_{\mathsf{synt}}}$, så trivielt har vi

$$\mathsf{synt}(g_{k\,\natural_{\mathsf{synt}}}) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(g'_{\natural_{\mathsf{synt}}})$$

Tilfellet for regelen

$$\mathsf{synt}(\mathsf{pred}(\mathsf{x})) = \mathsf{synt}(\mathsf{pred}(\mathsf{synt}(\mathsf{x})))$$

er analogt. Anta regelen

$$\mathsf{synt}(\mathsf{succ}(\mathsf{synt}(\mathsf{pred}(\mathsf{x})))) = \mathsf{synt}(\mathsf{x})$$

er brukt. Da har vi $g_k = c[\mathsf{synt}(\mathsf{succ}(\mathsf{synt}(\mathsf{pred}(g''))))]$ og $g' = c[\mathsf{synt}(g'')]$ (eller analogt omvendt), og vi har

$$g_{k\,\natural_{\mathsf{synt}}} = c_{\natural_{\mathsf{synt}}}[\mathsf{succ}(\mathsf{pred}(g''))] \quad \text{og} \quad g'_{\natural_{\mathsf{synt}}} = c_{\natural_{\mathsf{synt}}}[g'']$$

Vi fortsetter nå ved induksjon på posisjonen p i $c_{\natural \mathsf{synt}}$ hvori subtermene $\mathsf{succ}(\mathsf{pred}(g''))$ og g'' forekommer.

$p = \varepsilon$: Vi må vise $\mathsf{synt}(\mathsf{succ}(\mathsf{pred}(g''))) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(g'')$. Det er greit; vi har:

$$\mathsf{synt}(\mathsf{succ}(\mathsf{pred}(g''))) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(\mathsf{succ}(\mathsf{synt}(\mathsf{pred}(g')))) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(g'')$$

$p = i.q, 1 \leq i$: Siden $g_k, g' \in \mathcal{G}_{\Sigma^c \cup \{\mathsf{synt}\}}$, og alle ikke-konstanter i $\mathcal{G}_{\Sigma^c \cup \{\mathsf{synt}\}}$ har aritet 1, er $g_{k\natural \mathsf{synt}} = f(c'[\mathsf{succ}(\mathsf{pred}(g''))]_q)$ og $g'_{\natural \mathsf{synt}} = f(c'[g'']_q)$, for en $f \in \Sigma^c \cup \{\mathsf{synt}\}$. Vi må vise $\mathsf{synt}(f(c'[\mathsf{succ}(\mathsf{pred}(g''))]_q)) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(f(c'[g'']_q))$.

Dersom $f = \mathsf{synt}$ har vi ved induksjonshypotesen direkte at

$$f(c'[\mathsf{succ}(\mathsf{pred}(g''))]_q) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} f(c'[g'']_q)$$

så induksjonssteget følger trivielt ved monotonitet mhp. kontekstapplikasjon.

For tilfellet $f = \mathsf{succ}$ har vi

$$\mathsf{synt}(\mathsf{succ}(c'[\mathsf{succ}(\mathsf{pred}(g''))]_q)) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(\mathsf{succ}(\mathsf{synt}(c'[\mathsf{succ}(\mathsf{pred}(g''))]_q)))$$

og

$$\mathsf{synt}(\mathsf{succ}(c'[g'']_q)) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(\mathsf{succ}(\mathsf{synt}(c'[g'']_q)))$$

Induksjonshypotesen gir da $\mathsf{synt}(c'[\mathsf{succ}(\mathsf{pred}(g''))]_q) \overset{*}{\underset{E_{\mathsf{synt}}}{\leftrightarrow}} \mathsf{synt}(c'[g'']_q)$, og induksjonssteget følger ved monotonitet mhp. kontekstapplikasjon. Tilfellet $f = \mathsf{pred}$ er analogt. Dette avslutter p-induksjonen.

Situasjonen for regelen

$$\mathsf{synt}(\mathsf{pred}(\mathsf{synt}(\mathsf{succ}(\mathsf{x})))) = \mathsf{synt}(\mathsf{x})$$

er analog med situasjonen for regelen $\mathsf{synt}(\mathsf{succ}(\mathsf{synt}(\mathsf{pred}(\mathsf{x})))) = \mathsf{synt}(\mathsf{x})$.

◇

Dermed har vi ved teorem 3.17 at

$$\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\Sigma^c \cup \Sigma^d} = \simeq^{\alpha}_{\mathcal{G}_{\Sigma^c \cup \Sigma^d}}$$

○

Merk at dersom vi *vet* at E_s er en algebraisk spesifikasjon av en kanonisk-representant funksjon, så vet vi også at `KREP1` og `KREP2` er oppfylt. Da vet vi dessuten at `INDKONG` er oppfylt. Vi diskuterer verifikasjon av algebraiske spesifikasjoner av semantikkgivende syntaktiske funksjoner i avsnitt 3.9.

- Tilfellet $\Sigma^h \neq \emptyset$

Vi ser nå avslutningsvis i dette avsnittet kort på tilfellet $\Sigma^h \neq \emptyset$. I tråd med at `TsK` ikke antas å holde, må vi rimeligvis anta at $\hat{E}$ ikke nødvendigvis er tilstrekkelig Σ^h-komplett mhp. $\mathcal{G}_{\Sigma^c}$. I eksempel 54 side 102 viste vi at (3.12) side 91 ikke generelt gjelder for termer i $\mathcal{G}_{\Sigma^c \cup \{s\}}$. Ligningsmengden E_{delta} i eksempel 54, er imidlertid også et eksempel på at (3.12) ikke gjelder generelt for termer i $\mathcal{G}_{\Sigma^c \cup \Sigma^h}$. Ved en videreføring av argumentasjonen for at $\mathsf{succ}(\mathsf{pred}(0)) \not\simeq^{\alpha} \mathsf{delta}(\mathsf{succ}(\mathsf{pred}(0)))$, kan vi vise at

$$\mathsf{succ}(\mathsf{pred}(0)) \not\simeq^{\alpha} \#_{\mathsf{s}}(\mathsf{succ}(\mathsf{pred}(0))) – \#_{\mathsf{p}}(\mathsf{succ}(\mathsf{pred}(0)))$$

til tross for at

$$\mathsf{succ}(\mathsf{pred}(0)) \overset{*}{\underset{E^{id}_{\mathsf{delta}}}{\leftrightarrow}} \#_{\mathsf{s}}(\mathsf{succ}(\mathsf{pred}(0))) – \#_{\mathsf{p}}(\mathsf{succ}(\mathsf{pred}(0)))$$

Igjen er dette ikke så farlig, siden vi igjen primært er opptatt av restriksjonen $\simeq^{\alpha}_{\mathcal{G}_{\Sigma}}$. Vi kunne være fristet til å fremføre følgende påstand:

Påstand 3.21 *Under visse interessante betingelser, men uten å anta* `TsK` *eller* $\Sigma^h = \emptyset$ *eller* $\mathtt{T}\Sigma^h\mathtt{K}$*, gjelder*

$$\simeq^{\alpha}_{\mathcal{G}_\Sigma} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_\Sigma \tag{3.22}$$

Imidlertid skal vi senere i forbindelse med konsistens, se eksempler der

- $\Sigma^h \neq \emptyset$
- $\hat{E}$ er fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$

og der (3.22) ikke holder.

* * *

I dette avsnittet har vi vist at semantikk relativ til indirekte semantikk under forskjellige rimelige antagelser, kan reduseres til basis-initialsemantikk. Dette har vi gjort ved å betrakte *id*-utvidelser av indirekte spesifikasjoner.

Vi har sett at lettelser av antagelsene naturlig nok resulterer i innsnevringer av domenet reduksjonen kan foretas på. De reduksjonene vi har vist, er likevel alle på domener som inkluderer termuniverset som i utgangspunktet skal gis semantikk.

Videre beskriver de antagelser vi har gjort, interessante tilfeller som vi har sett, eller skal se eksempler på i den videre diskusjon.

Vi nevnte i begynnelsen av dette avsnittet at restriksjonene $\simeq^{\alpha}_{\mathcal{G}_\Sigma}$ og $\simeq^{\omega}_{\mathcal{G}_\Sigma}$ av $\simeq^{\alpha}$ og $\simeq^{\omega}$ under konsistens gir oss ønsket semantikk på $\mathcal{G}_\Sigma$, i den grad ønsket generatorsemantikk er den gitt av $\simeq^{s}$ og ønsket semantikk til definerte funksjonssymboler er den gitt i $\hat{E}$. For initialsemantikk følger dette bl.a. fra teorem 2.20 side 42 under bl.a. antagelsen $\mathtt{T}\hat{\Sigma}\mathtt{K}$. Men vi har jo ikke nødvendigvis $\mathtt{T}\hat{\Sigma}\mathtt{K}$ i alle våre reduksjoner av initialsemantikk. Imidlertid er det mulig å verifisere en generisk form av teorem 2.20 for restriksjoner av initialsemantikk ved forskjellige grader av tilstrekkelig kompletthet. Se også kommentaren på side 38. Det er ikke hensiktsmessig å gjøre dette i detalj her, men vi kan slå fast at $\simeq^{\alpha}_{\mathcal{G}_\Sigma}$ under konsistens *er* den ønskede semantikk i den grad ønsket generatorsemantikk er den gitt av $\simeq^{s}$ og ønsket semantikk til definerte funksjonssymboler er den (konstruktive) gitt i $\hat{E}$.

3.5 Inkonsistens i tilknytning til indirekte algebraisk spesifikasjon

Vi skal nå se på inkonsistens i tilknytning til indirekte algebraisk spesifikasjon. Nærmere bestemt skal vi se på bevaring av kjernen i semantikk definert relativt til indirekte semantikk. Vi skal altså se på initiell konsistens; eller det ved sats 2.7 ekvivalente begrep, final kjernebevaring (ekvivalent under `KONSERV` og `TK`).

I avsnitt 2.3.6 så vi hvordan spesifikasjoner over mange-til-en generatorunivers av definerte funksjoner kunne gi inkonsistens mhp. en gitt semantikk. Således er selvfølgelig faren for initiell inkonsistens relativt til indirekte semantikk også tilstede.

Men det er i tillegg en mulighet for inkonsistens som så og si er iboende hos indirekte spesifikasjon.

Anta at en algebraisk beskrivelse E_s av en syntaktisk funksjon gir indirekte semantikk til et mange-til-en generatorunivers. En konsekvens av kollapsen av to semantikkgivende nivåer til ett nivå, er at spesifikasjonen E_s nå blir en spesifikasjon/beskrivelse av funksjoner *over et mange-til-en termunivers*. (Våre algebraiske beskrivelser av syntaktiske funksjoner, er til sammenligning, alltid

over en-til-en univers. Vi har ikke to termer som representerer samme term.) Muligheten for inkonsistens ved selve spesifikasjonen/beskrivelsen E_s er således introdusert.

Inkonsistens relativt til en indirekte semantikk kan ikke ha opphav i det aktuelle spesifiserende definerte symbol: Det er jo nettopp spesifikasjonen av semantikken til det spesifiserende symbol som fungerer som spesifikasjon av indirekte semantikk ved (3.5) side 80.

Men dersom den underliggende semantikkgivende syntaktiske funksjon som den indirekte spesifikasjonen E_s er en beskrivelse av, er definert vha. hjelpefunksjoner, vil disses beskrivelse i E_s kunne gi inkonsistens mhp. $\simeq^s$:

Eksempel 56 Betrakt $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 på side 77. Siden

$$\mathsf{delta(succ(pred(0)))}\, {}_E\overset{*}{\leftrightarrow}_{\delta_{\mathcal{I}nt}}\, \mathsf{delta(0)}$$

har vi

$$\mathsf{succ(pred(0))} \simeq^{\mathsf{delta}} \mathsf{0}$$

for $\simeq^{\mathsf{delta}}$ den indirekte semantikk spesifisert av $E_{\delta_{\mathcal{I}nt}}$. Men vi har også

$$\#_{\mathsf{s}}\mathsf{(succ(pred(0)))}\, {}_E\overset{*}{\leftrightarrow}_{\delta_{\mathcal{I}nt}}\, \mathsf{succ(0)} \quad \text{og} \quad \#_{\mathsf{s}}\mathsf{(0)}\, {}_E\overset{*}{\leftrightarrow}_{\delta_{\mathcal{I}nt}}\, \mathsf{0}$$

Relativt til $\simeq^{\mathsf{delta}}$, gir dette initialsemantisk $\mathsf{succ(0)} \simeq^{\alpha} \mathsf{0}$ og finalsemantisk $\mathsf{succ(pred(0))} \not\simeq^{\omega} \mathsf{0}$. Siden $\mathsf{succ(0)} \not\simeq^{\mathsf{delta}} \mathsf{0}$ har vi altså at $E_{\delta_{\mathcal{I}nt}}$ er initielt inkonsistent relativt til $\simeq^{\mathsf{delta}}$, og ved «inversjon», ikke finalt kjernebevarende relativt til $\simeq^{\mathsf{delta}}$.

Følgende spesifikasjon er derimot initielt konsistent og finalt kjernebevarende relativt til $\simeq^{\mathsf{delta}}$:

$$E_{\mathsf{delta}} = \left\{ \begin{array}{l} \mathsf{delta(x) = mem(x,0)}, \\ \mathsf{mem(succ(x),0) = mem(x,succ(0))}, \\ \mathsf{mem(pred(x),0) = mem(x,pred(0))}, \\ \mathsf{mem(succ(x),pred(y)) = mem(x,y)}, \\ \mathsf{mem(pred(x),succ(y)) = mem(x,y)}, \\ \mathsf{mem(succ(x),succ(y)) = mem(x,succ(succ(y)))}, \\ \mathsf{mem(pred(x),pred(y)) = mem(x,pred(pred(y)))}, \\ \mathsf{mem(0,x) = x} \end{array} \right\}$$

Intuisjon for E_{delta}: E_{delta} er konvergent, så vi behandler E_{delta} som et deterministisk program.

En grunntermterm i $\mathcal{G}_{\mathsf{Int}}$ «prosesseres» funksjonssymbol for funksjonssymbol og leses initielt inn i «hukommelsen» mem. Prosesseringen skjer ved å flytte symboler fra 1. til 2. argument av mem. Det 2. argument er invariant kanonisk.

Således er E_{delta} en algebraisk spesifikasjon av kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32 på side 64.

Merk at vi her «inkorporerer» $\#_{\mathsf{s}}$ og $\#_{\mathsf{p}}$ inn i én funksjon mem. Således unngås inkonsistens innført av $\#_{\mathsf{s}}$ (og $\#_{\mathsf{p}}$).

Merk også at i motsetning til $E_{\delta_{\mathcal{I}nt}}$, er E_{delta} en indirekte spesifikasjon som tilfredstiller $\Sigma^h = \{h\}$ og $h\mathtt{ROT}$ på side 94.

○

Eksempel 57 Den samme historien som i eksempel 56 fortaltes om $E_{\delta_{\mathcal{I}nt}}$, kan også fortelles om $E_{\gamma_{\mathcal{I}nt}}$ fra eksempel 46 på side 80. $E_{\gamma_{\mathcal{I}nt}}$ er inkonsistent relativt til $\simeq^{\mathsf{gamma}}$ for $\simeq^{\mathsf{gamma}}$ den indirekte semantikk spesifisert av $E_{\gamma_{\mathcal{I}nt}}$. (Vi antar gamma som spesifiserende symbol i $E_{\gamma_{\mathcal{I}nt}}$). Vi har f.eks.

$$\mathsf{gamma(succ(pred(0)))}\, {}_E\overset{*}{\leftrightarrow}_{\gamma_{\mathcal{I}nt}}\, \mathsf{gamma(0)}$$

så

$$\mathsf{succ(pred(0))} \simeq^{\mathsf{gamma}} \mathsf{0}$$

Men vi har også

$$\#_{\mathsf{s}}(\mathsf{succ(pred(0))})_{E} \overset{*}{\underset{\gamma_{\mathcal{I}nt}}{\leftrightarrow}} \mathsf{succ(succ(0))} \quad \text{og} \quad \#_{\mathsf{s}}(\mathsf{0})_{E} \overset{*}{\underset{\gamma_{\mathcal{I}nt}}{\leftrightarrow}} \mathsf{succ(0)}$$

Relativt til $\simeq^{\mathsf{gamma}}$, gir dette initialsemantisk $\mathsf{succ(succ(0))} \simeq^{\alpha} \mathsf{succ(0)}$ og dermed initiell inkonsistens; og finalsemantisk $\mathsf{succ(pred(0))} \not\simeq^{\omega} \mathsf{0}$ og dermed ikke final kjernebevaring.

Følgende spesifikasjon er derimot initielt konsistent og finalt kjernebevarende relativt til $\simeq^{\mathsf{gamma}}$:

$$E_{\mathsf{gamma}} = \left\{ \begin{array}{l} \mathsf{gamma(x) = mem(succ(x),0)}, \\ \mathsf{mem(succ(x),0) = mem(x,succ(0))}, \\ \mathsf{mem(pred(x),0) = mem(x,pred(0))}, \\ \mathsf{mem(succ(x),pred(y)) = mem(x,y)}, \\ \mathsf{mem(pred(x),succ(y)) = mem(x,y)}, \\ \mathsf{mem(succ(x),succ(y)) = mem(x,succ(succ(y)))}, \\ \mathsf{mem(pred(x),pred(y)) = mem(x,pred(pred(y)))}, \\ \mathsf{mem(0,x) = x} \end{array} \right\}$$

Her er alle ligninger bortsett fra den første, identiske med ligningene i E_{delta} fra eksempel 56.
○

En indirekte spesifikasjon av en kjernesemantikk kan altså (selv) introdusere inkonsistens relativt til kjernesemantikken. Vi antydet innledningsvis i dette avsnittet at inkonsistens ikke kan ha opphav i det spesifiserende symbol alene. Vi gir her én presisering av hva som menes med dette (i avsnitt 4.2 kapittel 4 skal vi komme med en ytterligere presisering):

Sats 3.22 *For en $\hat{\Sigma} = \Sigma^c \cup \{s\}, s \notin \Sigma^c$, anta at $E_s \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$ er en indirekte spesifikasjon med spesifiserende symbol s av $\simeq^s$ på $\mathcal{G}_{\Sigma^c}$ (vi antar altså dermed* `INDKONG`*). Da er E_s finalt kjernebevarende relativt til $\simeq^s$.*

Bevis: Vi minner om (3.7) på side 86.

Anta så tvert imot at det finnes en $c \in \mathcal{G}_{\hat{\Sigma}}$ og $g_c, g'_c, g_1, g_2 \in \mathcal{G}_{\Sigma^c}$ slik at

$$c[g_c] \overset{*}{\underset{E_s}{\leftrightarrow}} g_1 \not\simeq^s g_2 \overset{*}{\underset{E_s}{\leftrightarrow}} c[g'_c] \tag{3.23}$$

til tross for at $g_c \simeq^s g'_c$. Nå kan ikke $c \in \mathcal{G}_{\Sigma^c}$, siden vi da får $c[g_c] \simeq^s c[g'_c]$, og ved (3.7) også $g_1 \simeq^s c[g_c]$ og $c[g'_c] \simeq^s g_2$. Ialt får vi da $g_1 \simeq^s g_2$ som er en motsigelse.

Det må derfor være en forekomst av s i c. Merk da at dersom E_s er fullstendig $\{s\}$-ukomplett mhp. Σ^c, så har vi umiddelbart en motsigelse til (3.23), og satsen følger.

Anta dog for det generelle tilfellet at $c = c'[s(c'')]$ for en $c'' \in \mathcal{G}_{\Sigma^c}$. (Vi ser altså på en dypest forekommende s i c.) Ved `INDKONG` får vi da $c''[g_c] \simeq^s c''[g'_c]$, dvs. $s(c''[g_c]) \overset{*}{\underset{E_s}{\leftrightarrow}} s(c''[g'_c])$, som ved monotonitet gir $c[g_c] \overset{*}{\underset{E_s}{\leftrightarrow}} c[g'_c]$. Men da har vi $g_1 \overset{*}{\underset{E_s}{\leftrightarrow}} g_2$. Da gir (3.7) $g_1 \simeq^s g_2$, som er en motsigelse, og satsen følger.
□

Vi kan altså ved sats 3.22 i noen tilfeller direkte slå fast kjernebevaring:

Eksempel 58 Betrakt beskrivelsen E_{synt} fra eksempel 47 side 80. La $\simeq^{\mathsf{synt}}$ være semantikken på $\mathcal{G}_{\mathsf{Int}}$ spesifisert indirekte ved E_{synt} (med spesifiserende symbol synt). Ved sats 3.22 er E_{synt} finalt kjernebevarende relativt til $\simeq^{\mathsf{synt}}$.

Dette fordi det i E_{synt} ikke forekommer andre definerte funksjonssymboler enn det spesifiserende symbol. Men vi kan også bruke bemerkningen om fullstendig $\{s\}$-ukompletthet i beviset for sats 3.22, siden E_{synt} er fullstendig $\{\mathsf{synt}\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$

○

Eksempel 59 Imidlertid kan også sats 2.8 side 37 benyttes for å slå fast kjernebevaring for E_{synt} fra forrige eksempel. Siden E_{synt} er fullstendig $\{\mathsf{synt}\}$-ukomplett (eller $\mathsf{Int} \cup \{\mathsf{synt}\}$-ukomplett) mhp. $\mathcal{G}_{\mathsf{Int}}$, er E_{synt} finalt kjernebevarende og initielt konsistent relativt til $\simeq^{\mathsf{synt}}$.

○

Avslutningsvis i dette avsnitt ser vi kort på relasjonen $\Re^s$ definert i (3.9) på side 88. Vi adopterer den formelle omgivelse gitt i figur 3.7 på side 85. Vi skal bare antyde at inkonsistens gjør at $\Re^s$ ikke uten videre spesifiserer semantikk:

Sats 3.23 *Anta* `INDKONSERV` *og* `DISJ`. *For $\hat{E}$ og E_s som i figur 3.7, anta at E_s er indirekte $\mathcal{G}_{\Sigma^c}$-kongruent (*`INDKONG`*). At $\hat{E}$ er indirekte $\mathcal{G}_{\hat{\Sigma}}$-kongruent medfører at $\hat{E}$ er finalt kjernebevarende relativt til $\simeq^s$*

Bevis: Anta $\hat{E}$ er ikke finalt kjernebevarende mhp. $\simeq^s$. Da finnes $g_c, g'_c, g_1, g_2 \in \mathcal{G}_{\Sigma^c}$ og $c \in \mathcal{G}_{\hat{\Sigma}}$ slik at

$$c[g_c] \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1 \not\simeq^s g_2 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} c[g'_c]$$

selv om $g_c \simeq^s g'_c$. Nå betyr $g_1 \not\simeq^s g_2$ at $s(g_1) \overset{*}{\underset{E_s}{\not\leftrightarrow}} s(g_2)$, og ved `INDKONSERV` altså $g_1 \overset{*}{\underset{\hat{E}}{\not\leftrightarrow}} g_2$. Siden nå verken $s(g_1) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g_2)$ eller $g_1 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_2$, kan ved `DISJ` heller ikke $s(g_1) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_2)$. Altså har vi $s(g_1) \overset{*}{\underset{\hat{E}}{\not\leftrightarrow}} s(g_2)$. Ved monotonitet mhp. kontekstapplikasjon får vi da

$$s(c[g_c]) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_1) \overset{*}{\underset{\hat{E}}{\not\leftrightarrow}} s(g_2) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(c[g'_c])$$

som gir $s(c[g_c]) \overset{*}{\underset{\hat{E}}{\not\leftrightarrow}} s(c[g'_c])$, og dermed mangel på indirekte $\mathcal{G}_{\hat{\Sigma}}$-kongruens, siden

$$g_c \simeq^s g'_c \;\Leftrightarrow\; s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g'_c) \;\Rightarrow\; s(g_c) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g'_c)$$

□

At $\hat{E}$ ikke er finalt kjernebevarende relativt til $\simeq^s$, medfører altså at relasjonen $\Re^s$ på $\mathcal{G}_{\hat{\Sigma}}$ slik at

$$g \;\Re^s\; g' \;\Leftrightarrow\; s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g')$$

ikke er en kongruensrelasjon på $\mathcal{G}_{\hat{\Sigma}}$. Det kan dog tenkes at $\Re^s$ restriktert til $\mathcal{G}_{\Sigma}$ likevel er en kongruensrelasjon. Vi skal komme tilbake til relasjonen $\Re^s$ i avsnitt 3.8.

* * *

Vi har introdusert *id*-utvidelser av indirekte spesifikasjoner. Vi har vist reduksjon til basis-initialsemantikk ved *id*-utvidelser, og vi har demonstrert at indirekte spesifikasjon i seg selv kan være en kilde til inkonsistens relativt til semantikken indirekte spesifisert.

Vi skal nå i de nærmeste avsnitt studere *id*-utvidelser av indirekte spesifikasjoner videre. Vi skal introdusere begrepet kjernebevaring også for *id*-utvidelser.

Ved samsvar mellom *id*-utvidelse og basis-initiell semantikk, vil kjernebevaring for *id*-utvidelser kunne overføres til initiell konsistens i basis-initial semantikken.

Vi skal se på noen måter for å etablere kjernebevaring og for å oppdage ikke kjernebevaring i *id*-utvidelser.

3.6 Komplettering av *id*-utvidelser.

Initiell inkonsistens relativt til en semantikk indirekte spesifisert ved en ligningsmengde E_s med spesifiserende symbol s, kan av og til tilkjennegi seg i reglene produsert av en Knuth&Bendix-prosess gitt s-*id*-utvidelsen av E_s. At selve semantikken indirekte spesifisert er riktig, kan også tilkjennegi seg i slike regler.

Vi skal derfor studere *id*-utvidelser videre i sammenheng med Knuth&Bendix-komplettering. Vi minner om at 'vellykket Knuth&Bendix-prosess' betyr at mengden av vedvarende ligninger er tom og at mengden av vedvarende omskrivningsregler er komplett for initialligningene (se avsnitt 2.4.4).

Definisjon 3.6 *For en s-id-utvidelse E_s^{id} av en under* `INDKONG` *indirekte spesifikasjon E_s med spesifiserende symbol s, kaller vi $\simeq^s$ **kjernen** til E_s^{id}. Videre kalles E_s^{id} **kjernebevarende** dersom*

$$\overset{*}{\underset{E_s^{id}}{\longleftrightarrow}} \mathcal{G}_{\Sigma^c} = \simeq^s$$

for $\mathcal{G}_{\Sigma^c}$ — domenet til $\simeq^s$.

Det viser seg at en vellykket Knuth&Bendix-komplettering av s-*id*-utvidelsen E_s^{id} av en indirekte spesifikasjon E_s med spesifiserende symbol s, kan gi en spesifikasjon av $\overset{*}{\underset{E_s^{id}}{\longleftrightarrow}} \mathcal{G}_{\Sigma^c}$ i form av en regelmengde med regler utelukkende fra $\mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$. Dette skal være vårt grunnleggende resultat i dette avsnittet.

Definisjon 3.7 *La R være en vilkårlig regelmengde. La r være en vilkårlig regel.*

- *R er **høyreredusert med hensyn på** r dersom for hver regel $v \rightarrow h \in R$ så kan h ikke omskrives ved r.*
- *R er **innbyrdes høyreredusert** dersom R er høyreredusert med hensyn på alle regler i R.*

Lemma 3.24 *Anta 'fairness' mht. inferensregelanvendelse i Knuth&Bendix-prosesser. Dvs. ingen inferens oversees i det uendelige i en prosess. Dersom en Knuth&Bendix-prosess er vellykket, så er mengden av vedvarende regler innbyrdes høyreredusert.*

Godtgjørelse: Dette er lett å se ved inspeksjon av inferensreglene for Knuth-&Bendix-prosesser i figur 2.4 på side 54, og da i særdeleshet inferensregelen **Sammensett**.
◇

Teorem 3.25 *La Σ være en vilkårlig signatur, og la $s \notin \Sigma$ være et funksjonssymbol med aritet 1. La $\hat{\Sigma} = \Sigma \cup \{s\}$. La $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$.*

Anta at en Knuth&Bendix-prosess er vellykket gitt s-id-utvidelsen $\hat{E}^{id}$ av $\hat{E}$, og la R være mengden av vedvarende ligninger til denne prosessen. Dersom semantikken $\overset{*}{\underset{\hat{E}^{id}}{\longleftrightarrow}} \mathcal{T}_{\Sigma}(\mathcal{V})$ ikke er fri, finnes en konvergent $R' \subseteq R$ slik at

$$R' \subseteq \mathcal{E}(\mathcal{T}_{\Sigma}(\mathcal{V})) \qquad \textit{(dvs. } R' \textit{ har ingen forekomster av } s\textit{)}$$

og

$$\overset{*}{\underset{R'}{\leftrightarrow}} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{T}_\Sigma(\mathcal{V})$$

Bevis: Anta $t \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} u$ for vilkårlige $t, u \in \mathcal{T}_\Sigma(\mathcal{V})$. Siden R er komplett for $\hat{E}^{id}$, har vi $t \overset{*}{\underset{R}{\leftrightarrow}} u$. Ved at $\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{T}_\Sigma(\mathcal{V})$ ikke er fri, er det interessante tilfellet $t \neq u$; og dermed $t \overset{+}{\underset{R}{\leftrightarrow}} u$.

Anta nå at en slik $R' \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ ikke finnes. Siden $t, u \in \mathcal{T}_\Sigma(\mathcal{V})$ og $t \neq u$, må det da finnes en regel $v \to h \in R$ slik at $v \in \mathcal{T}_\Sigma(\mathcal{V})$ og $h = c[s(h')]$ for en $c \in \mathcal{T}_{\hat{\Sigma}}(\mathcal{V})$ og en $h' \in \mathcal{T}_\Sigma(\mathcal{V})$. (Vi ser altså på en dypeste forekomst av s i h.) Men siden R er komplett for $\hat{E}^{id}$, har vi $s(h') \overset{*}{\underset{R}{\to}} \tilde{h} \overset{*}{\underset{R}{\leftarrow}} h'$ for $s(h)! = \tilde{h} = h'!$. Nå kan ikke $\tilde{h} = s(h')$, siden det ville medføre $h' \overset{+}{\underset{R}{\to}} s(h')$ (siden $s(h') \neq h'$), og altså ikke-terminerende R. Altså må vi ha $s(h') \overset{+}{\underset{R}{\to}} \tilde{h}$. Men da er ikke regelen $v \to h$ høyreredusert, og det strider mot lemma 3.24.
□

Eksempel 60 Betrakt E_{synt} fra eksempel 47 side 80. I overensstemmelse med teorem 3.25 gir her en terminerende vellykket Knuth&Bendix-prosess gitt $E_{\mathsf{synt}} \cup \{\mathsf{synt(x)=x}\}$, regelmengden

$$R^{KB}_{\mathsf{synt}} = \left\{ \begin{array}{l} \mathsf{synt(x)} \to \mathsf{x}, \\ \mathsf{succ(pred(x))} \to \mathsf{x}, \\ \mathsf{pred(succ(x))} \to \mathsf{x} \end{array} \right\}$$

○

I det generelle tilfelle der den aktuelle ligningsmengden $\hat{E}$ inneholder spesifikasjoner av flere funksjoner, skal vi gjøre en for oss ikke urimelig antagelse om reduksjonsordningen på termuniverset for Knuth&Bendix-prosessen i tillegg:

Observasjon 3.26 *La Σ og Σ' være vilkårlige signaturer slik at $\Sigma \cap \Sigma' = \emptyset$. La $\hat{\Sigma} = \Sigma \cup \Sigma'$. La $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$. Anta at en Knuth&Bendix-prosess startes med $\hat{E}$ og en reduksjonsordning som er slik at enhver $u \in \mathcal{T}_{\hat{\Sigma}}(\mathcal{V}) \setminus \mathcal{T}_\Sigma(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_\Sigma(\mathcal{V})$. Da vil ingen regel $v \to h$ slik at $v \in \mathcal{T}_\Sigma(\mathcal{V})$ og $h \in \mathcal{T}_{\hat{\Sigma}}(\mathcal{V}) \setminus \mathcal{T}_\Sigma(\mathcal{V})$ noensinne bli generert av prosessen.*

Vi kan nå forsterke teorem 3.25:

Teorem 3.27 *La Σ og Σ' være vilkårlige signaturer slik at $\Sigma \cap \Sigma' = \emptyset$. La $s \notin \Sigma \cup \Sigma'$ være et funksjonssymbol med aritet 1. La $\hat{\Sigma} = \Sigma \cup \Sigma' \cup \{s\}$. La $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$.*

Anta at en Knuth&Bendix-prosess er vellykket gitt s-id-utvidelsen $\hat{E}^{id}$ av $\hat{E}$, og la R være mengden av vedvarende ligninger til denne prosessen. Anta dessuten at reduksjonsordningen i prosessen er slik at enhver $u \in \mathcal{T}_{\Sigma' \cup \Sigma}(\mathcal{V}) \setminus \mathcal{T}_\Sigma(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_\Sigma(\mathcal{V})$. Dersom $\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{T}_\Sigma(\mathcal{V})$ ikke er fri, finnes en konvergent $R' \subseteq R$ slik at

$$R' \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V})) \qquad \textit{(dvs. } R' \textit{ har ingen forekomster av symboler fra } \{s\} \cup \Sigma'\textit{)}$$

og

$$\overset{*}{\underset{R'}{\leftrightarrow}} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{T}_\Sigma(\mathcal{V})$$

Bevis: Anta $t \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} u$ for vilkårlige $t, u \in \mathcal{T}_\Sigma(\mathcal{V})$. Siden R er komplett for $\hat{E}^{id}$, har vi $t \overset{*}{\underset{R}{\leftrightarrow}} u$. Ved at $\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{T}_\Sigma(\mathcal{V})$ ikke er fri, er det interessante tilfellet $t \neq u$; og dermed $t \overset{+}{\underset{R}{\leftrightarrow}} u$.

Anta nå at en slik $R' \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ ikke finnes. Siden $t, u \in \mathcal{T}_\Sigma(\mathcal{V})$ og $t \neq u$, må det da finnes en regel $v \to h \in R$ slik at $v \in \mathcal{T}_\Sigma(\mathcal{V})$ og $h \in \mathcal{T}_{\hat{\Sigma}}(\mathcal{V})$. Men ved

teorem 3.25 kan ikke s forekomme i h, og ved observasjon 3.26 kan heller ikke noen symboler fra Σ' forekomme i h, så en slik regel $v \rightarrow h \in R$ finnes ikke, og vi har en motsigelse.
□

Definisjon 3.8 *La $\hat{\Sigma}$ og Σ være signaturer slik at $\Sigma \subseteq \hat{\Sigma}$. La $\hat{E}$ være en vilkårlig ligningsmengde. For en id-utvidelse $\hat{E}^{id}$ av $\hat{E}$ og en ligningsmengde $E' \in \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$, anta at*

$$\overset{\leftrightarrow}{\underset{E'}{*}}\mathcal{G}_\Sigma = \overset{\leftrightarrow}{\underset{\hat{E}^{id}}{*}}\mathcal{G}_\Sigma$$

*Da sier vi at E' er et Σ-**manifest** for $\hat{E}^{id}$. Dersom $\mathcal{G}_\Sigma$ er domenet til kjernen til $\hat{E}^{id}$, kaller vi E' et **kjerne-manifest** for $\hat{E}^{id}$.*

For Σ, $\hat{E}^{id}$ og R' i teoremene 3.25 og 3.27, er således R' et Σ-manifest for $\hat{E}^{id}$.

Eksempel 60 (forts.) Delmengden av R^{KB}_{synt}

$$R^{man}_{\mathsf{synt}} = \left\{ \begin{array}{l} \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

er et $\mathcal{G}_{\mathsf{Int}}$-manifest for $E_{\mathsf{synt}} \cup \{\mathsf{synt(x)=x}\}$.
○

Eksempel 61 Betrakt E_{delta} fra eksempel 56 på side 108. La

$$\Sigma^h = \{\mathsf{mem}\}$$
$$\Sigma^c = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}\}$$

Vår eksekvering av en Knuth&Bendix-prosess gitt $E_{\mathsf{delta}} \cup \{\mathsf{delta(x)=x}\}$ og en reduksjonsordning der enhver $u \in \mathcal{T}_{\Sigma^h \cup \Sigma^c}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$, terminerer ikke, men er vellykket og gir den vedvarende regelmengden

$$R^{KB}_{\mathsf{delta}} = \left\{ \begin{array}{l} \mathsf{delta(x)} \rightarrow \mathsf{x}, \\ \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x}, \\ \\ \mathsf{mem(x,0)} \rightarrow \mathsf{x}, \\ \mathsf{mem(x,succ}^i\mathsf{(0))} \rightarrow \mathsf{succ}^i\mathsf{(x)}; 1 \leq i, \\ \mathsf{mem(x,pred}^i\mathsf{(0))} \rightarrow \mathsf{pred}^i\mathsf{(x)}; 1 \leq i, \\ \mathsf{mem(succ(x),y)} \rightarrow \mathsf{mem(x,succ(y))}, \\ \mathsf{mem(pred(x),y)} \rightarrow \mathsf{mem(x,pred(y))}, \\ \mathsf{mem(0,x)} \rightarrow \mathsf{x} \end{array} \right\}$$

(Det er lett å se eksempelvis, at det uendelige antall regler

$$\mathsf{mem(x,succ}^i\mathsf{(0))} \rightarrow \mathsf{succ}^i\mathsf{(x)}; 1 \leq i$$

vil bli generert: Reglene

$$\mathsf{mem(succ(x),y)} \rightarrow \mathsf{mem(x,succ(y))} \quad \text{og} \quad \mathsf{mem(x,succ}^i\mathsf{(0))} \rightarrow \mathsf{succ}^i\mathsf{(x)}$$

gir det ekte kritiske par $\mathsf{mem(x,succ}^{i+1}\mathsf{(0))} \rightarrow \mathsf{succ}^{i+1}\mathsf{(x)}$. Følgelig terminerer ikke prosessen.

Det er også lett å se at R^{KB}_{delta} må være vedvarende: Regelskjemaene

$$\mathsf{mem}(\mathsf{x},\mathsf{succ}^i(0)) \rightarrow \mathsf{succ}^i(\mathsf{x}); 1 \leq i,$$
$$\mathsf{mem}(\mathsf{x},\mathsf{pred}^i(0)) \rightarrow \mathsf{pred}^i(\mathsf{x}); 1 \leq i$$

uttrykker alle nye regler som vil genereres. Ingen andre ekte kritiske par er mulige, enn dem som gir opphav til regler uttrykt ved disse skjemaene.) Delmengden av R^{KB}_{delta}

$$R^{man}_{\mathsf{delta}} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{pred}(\mathsf{x})) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) \rightarrow \mathsf{x} \end{array} \right\}$$

er da ifølge teorem 3.27 et $\mathcal{G}_{\mathsf{Int}}$-manifest for $E_{\mathsf{delta}} \cup \{\mathsf{delta}(\mathsf{x})=\mathsf{x}\}$. (Dette rettferdiggjøres i dette ikke-terminerende tilfellet ved at det jo ikke vil bli generert nye regler i $\mathcal{E}(\mathcal{T}_{\mathsf{Int}}(\mathcal{V}))$.)
○

Eksempel 62 Betrakt $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 på side 77. La

$$\Sigma^h = \{\#_{\mathsf{s}}, \#_{\mathsf{p}}, +, -\}$$
$$\Sigma^c = \{0, \mathsf{succ}, \mathsf{pred}\}$$

En terminerende vellykket eksekvering av en Knuth&Bendix-prosess gitt $E_{\delta_{\mathcal{I}nt}} \cup \{\mathsf{delta}(\mathsf{x})=\mathsf{x}\}$ og en reduksjonsordning der enhver $u \in \mathcal{T}_{\Sigma^h \cup \Sigma^c}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$, gir regelmengden

$$R^{KB}_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{delta}(\mathsf{x}) \rightarrow \mathsf{x}, \\ \\ \mathsf{succ}(\mathsf{x}) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{x}) \rightarrow \mathsf{x}, \\ \\ \#_{\mathsf{s}}(0) \rightarrow 0, \\ \#_{\mathsf{p}}(0) \rightarrow 0, \\ \#_{\mathsf{s}}(\mathsf{x}) - \#_{\mathsf{p}}(\mathsf{x}) \rightarrow \mathsf{x}, \\ \mathsf{x} - 0 \rightarrow \mathsf{x}, \\ \mathsf{x} + 0 \rightarrow \mathsf{x}, \\ 0 + \#_{\mathsf{p}}(\mathsf{x}) \rightarrow \#_{\mathsf{s}}(\mathsf{x}), \\ 0 + \#_{\mathsf{s}}(\mathsf{x}) \rightarrow \#_{\mathsf{s}}(\mathsf{x}) \end{array} \right\}$$

Ved teorem 3.27 er delmengden av $R^{KB}_{\delta_{\mathcal{I}nt}}$

$$R^{man}_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{x}) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{x}) \rightarrow \mathsf{x} \end{array} \right\}$$

et $\mathcal{G}_{\mathsf{Int}}$-manifest for $E_{\delta_{\mathcal{I}nt}} \cup \{\mathsf{delta}(\mathsf{x})=\mathsf{x}\}$.

En terminerende vellykket Knuth&Bendix-prosess gitt delta-id-utvidelsen av $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$ (se eksempel 53 side 93) og en reduksjonsordning som over, gir også $\mathcal{G}_{\mathsf{Int}}$-manifestet $R^{man}_{\delta_{\mathcal{I}nt}}$.
○

Vi kan således bruke Knuth&Bendix-komplettering som et verktøy for lettere å *se* generatorsemantikken spesifisert av id-utvidelsen av en indirekte spesifikasjon, i form av et kjerne-manifest. Et slikt kjerne-manifest kan gjøre det enklere å etablere hvorvidt en id-utvidelse er kjernebevarende eller ikke.

Gitt at en id-utvidelse kan reduseres til basis-initialsemantikk (se avsnitt 3.4.5), kan så kjernebevaring av id-utvidelsen direkte overføres til initiell konsistens.

Dersom en *id*-utvidelse er kjernebevarende, er det også mulig å bruke kjerne-manifest til lettere å se om kjernesemantikken er den man faktisk ønsker. Disse ting skal vi se på i de nærmeste avsnitt.

En forutsetning for at Knuth&Bendix-komplettering av *id*-utvidelser skal gi et kjerne-manifest for en *id*-utvidelse, er selvfølgelig eksistensen av et et slikt kjerne-manifest. Et kjerne-manifest R generert ved Knuth&Bendix-komplettering må dessuten, siden Knuth&Bendix-prosesser er basert på søk av kritiske par — m.a.o. utleding av logiske konsekvenser — være slik at

hver regel $v \to h$ i R er en logisk konsekvens av id-utvidelsen.

Et kjerne-manifest for en *id*-utvidelse består ikke nødvendigvis av logiske konsekvenser av *id*-utvidelsen:

Eksempel 63 Betrakt den indirekte spesifikasjonen $E_{\delta_{\mathsf{SetNat}}}$ fra eksempel 45 på side 78. Betrakt så den direkte spesifikasjon

$$E_{\mathsf{SetNat}} = \left\{ \begin{array}{l} \mathsf{add(add(s,x),x) = add(s,x)}, \\ \mathsf{add(add(s,x),y) = add(add(s,y),x)} \end{array} \right\}$$

fra eksempel 27 side 50.

Anta en Knuth&Bendix-prosess startes gitt $E_{\delta_{\mathsf{SetNat}}} \cup \{\mathsf{delta(x) = x}\}$ Det fins ingen komplett regelmengde for E_{SetNat}, så vi kan med en gang si at manifestet E_{SetNat} ikke vil bli generert som regelmengde.

Det kunne dog tenkes at E_{SetNat} vil dukke opp som ligninger under prosessens gang. Men dette vil heller ikke skje. E_{SetNat} er nemlig ikke logiske konsekvenser av $E_{\delta_{\mathsf{SetNat}}} \cup \{\mathsf{delta(x) = x}\}$.

Imidlertid er E_{SetNat} induktive konsekvenser av $E_{\delta_{\mathsf{SetNat}}} \cup \{\mathsf{delta(x) = x}\}$. (Fullstendig sortering fungerer kun på grunntermer.) Termene

$$\mathsf{delta(add(add(s,x),y))} \quad \text{og} \quad \mathsf{delta(add(add(s,y),x))}$$

er eksempelvis altså ikke $E_{\delta_{\mathsf{SetNat}}}$-like, selv om alle grunninstanser av disse to termene er det.

○

3.6.1 *id*-utvidelser og mangel på kjernebevaring

Vi ser her på muligheter for å oppdage/etablere mangel på kjernebevaring i *id*-utvidelser.

Først et resultat som bekrefter noe som sikkert allerede er innlysende; nemlig at kjerne-manifest for *id*-utvidelser kan brukes til å vise mangel på kjernebevaring for *id*-utvidelser.

Sats 3.28 *La igjen Σ^c være en signatur av generatorer. Anta en vilkårlig $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$ inneholdende en indirekte spesifikasjon E_s av en semantikk $\simeq^s$ på $\mathcal{G}_{\Sigma^c}$ med spesifiserende symbol s. Anta vi har tilgjengelig et Σ^c-manifest $\hat{E}^{man}$ av $\hat{E}^{id}$.*

Anta det finnes en ligning/regel $v = h$ i $\hat{E}^{man}$ og en $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}}$, slik at

$$s(v\sigma) \overset{*}{\underset{\hat{E}^{id}_s}{\leftrightarrow}} s(h\sigma)$$

Da er $\hat{E}^{id}$ ikke kjernebevarende.

Bevis: Vi har jo selvsagt $v\sigma \overset{*}{\underset{\hat{E}^{man}}{\leftrightarrow}} h\sigma$, og siden $\hat{E}^{man}$ er et Σ^c-manifest for $\hat{E}^{id}$, har vi $v\sigma \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} h\sigma$. Men dermed er $\hat{E}^{id}$ ikke kjernebevarende.

□

Og siden Knuth&Bendix-komplettering kan generere kjerne-manifest for *id*-utvidelser, kan altså mangel på kjernebevaring i *id*-utvidelser av og til etableres vha. Knuth&Bendix-komplettering.

Eksempel 64 Betrakt $E_{\delta_{\mathcal{I}nt}}$ (evt. $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$) fra eksempel 44 side 77. Betrakt $\mathcal{G}_{\mathsf{Int}}$-manifestet

$$R^{man}_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{succ(x)} \rightarrow \mathsf{x}, \\ \mathsf{pred(x)} \rightarrow \mathsf{x} \end{array} \right\}$$

for delta-*id*-utvidelsen $E_{\delta_{\mathcal{I}nt}} \cup \{\mathsf{delta(x)=x}\}$ ($E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$) fra eksempel 62 side 114. Vi kan da straks konstatere at delta-*id*-utvidelsen av $E_{\delta_{\mathcal{I}nt}}$ ($E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$) ikke er kjernebevarende. Vi har nemlig $\mathsf{delta(succ(0))}_{E \not\delta_{\mathcal{I}nt}} \mathsf{delta(0)}$. $E_{\delta_{\mathcal{I}nt}}$ er konvergent, så dette vises ved å observere at

$$\mathsf{delta(succ(0))}!E_{\delta_{\mathcal{I}nt}} \neq \mathsf{delta(0)}!E_{\delta_{\mathcal{I}nt}}$$

○

Eksempel 65 Betrakt en abstrakt datatype $\mathcal{I}nt^{pos}_n$ med bæremengde

$$\{\ldots, \Leftrightarrow 4, \Leftrightarrow 3, \Leftrightarrow 2, \Leftrightarrow 1, 0, 1, \ldots, n\}; n \geq 1$$

dvs. bestående av samtlige negative tall med 0 og de positive tall $1, \ldots, n$. Vi har altså *modulo* $n+1$ på de positive tall. En mulig formell datatype er den initielle bestemt av

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0 : Int}, \\ \mathsf{succ : Int \rightarrow Int}, \\ \mathsf{pred : Int \rightarrow Int} \end{array} \right\}$$

og

$$E_n = \left\{ \begin{array}{ll} 1: & \mathsf{delta(x) = mem(x,0,0)}, \\ 2: & \mathsf{mem(succ(x),y,0) = mem(x,succ(y),0)}, \\ 3: & \mathsf{mem(pred(x),y,0) = mem(x,pred(y),0)}, \\ 4: & \mathsf{mem(0,pred(x),0) = mem(0,x,pred(0))}, \\ 5_0: & \mathsf{mem(0,succ(x),0) = mem(0,x,succ(0))}, \\ 5_1: & \mathsf{mem(0,succ(x),succ(0)) = mem(0,x,succ(succ(0)))}, \\ \vdots & \vdots \\ 5_i: & \mathsf{mem(0,succ(x),succ^i(0)) = mem(0,x,succ^{i+1}(0))}, \\ \vdots & \vdots \\ 5_n: & \mathsf{mem(0,succ(x),succ^n(0))) = mem(0,x,0)}, \\ 6: & \mathsf{mem(0,succ(x),pred(y)) = mem(0,x,y)}, \\ 7: & \mathsf{mem(0,pred(x),succ(y)) = mem(0,x,y)}, \\ 8: & \mathsf{mem(0,pred(x),pred(y)) = mem(0,x,pred(pred(y)))}, \\ 9: & \mathsf{mem(0,0,x) = x} \end{array} \right\}$$

og relativ til den indirekte semantikk spesifisert av E_n.

Betrakt kongruensrelasjonen $\simeq^{\mathcal{I}nt^{pos}_n}_{\mathsf{Int}}$ indusert av den unike $\phi^{\mathcal{I}nt^{pos}_n}_{\mathsf{Int}}$. Bl.a. for å gi intuisjon om E_n, godtgjør vi at E_n er en spesifikasjon av en kanonisk-representant funksjon for $\simeq^{\mathcal{I}nt^{pos}_n}_{\mathsf{Int}}$, der de kanoniske representanter er termer uten forekomster av både succ og pred.

E_n er konvergent, så vi behandler E_n som et deterministisk program. En grunnterm i $\mathcal{G}_{\mathsf{Int}}$ «prosesseres» funksjonssymbol for funksjonssymbol og leses initielt inn i «hukommelsen» mem (ligning 1). Pga. asymmetri, er det her ikke likegyldig i hvilken rekkefølge succ'er og pred'er prosesseres. F.eks. ønskes at

$\mathsf{succ}(\mathsf{pred}(\mathsf{succ}^n(0)))$ representerer n,

men at

$\mathsf{pred}(\mathsf{succ}(\mathsf{succ}^n(0)))$ representerer $\Leftrightarrow 1$

(Antallet succ'er og pred'er er identisk i disse to termene.)

Prosesseringen av en term må derfor foregå «innenfra og ut». (Som matematisk funksjonsapplikasjons-evaluering.) Ligningene 2 og 3 «reverserer» derfor termen.

Anta så at en reversert term foreligger i 2. argument av mem. Det 3. argument av mem er invariant kanonisk, og er den kanoniske «historie» til subtermen prosessert hittil. Dette er lett å verifisere for ligninger som fører symboler fra 2. til 3. argument.

Betrakt så reglene $5_0 - 5_n$. Leses en succ og den kanoniske historie er $\mathsf{succ}^n(0)$, sløyfes alle $n+1$ succ'er i tråd med *modulo* $n+1$ på de positive tall.

De øvrige regler skulle da være greie.

Siden E_n er en spesifikasjon av en kanonisk-representant funksjon for $\simeq^{\mathcal{I}nt^{pos}_n}_{\mathsf{Int}}$, er E_n en indirekte spesifikasjon av $\simeq^{\mathcal{I}nt^{pos}_n}_{\mathsf{Int}}$.

En terminerende vellykket Knuth&Bendix-prosess gitt delta-id-utvidelsen av E_n for $n = 2$, gir følgende $\mathcal{G}_{\mathsf{Int}}$-manifest:

$$R_2^{man} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{succ}(\mathsf{succ}(\mathsf{x}))) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{x}) \rightarrow \mathsf{succ}(\mathsf{succ}(\mathsf{x})) \end{array} \right\}$$

Vi kan så vise mangel på kjernebevaring ved

$$\mathsf{delta}(\mathsf{pred}(0))!E_2 = \mathsf{pred}(0) \neq \mathsf{succ}(\mathsf{succ}(0)) = \mathsf{delta}(\mathsf{succ}(\mathsf{succ}(0)))!E_2$$

○

Eksempel 66 Betrakt basis-initialsemantikken $\simeq$ på $\mathcal{G}_{\mathsf{Int}}$ spesifisert av den ene ligning

$$E_a = \left\{ \; \mathsf{succ}(\mathsf{pred}(\mathsf{x})) = \mathsf{x} \; \right\}$$

Vi presenterer en indirekte spesifikasjon av $\simeq$:

$$E_b = \left\{ \begin{array}{ll} 1: & \mathsf{delta}(\mathsf{x}) = \mathsf{mem}(\mathsf{x},0,0,0), \\ 2: & \mathsf{mem}(\mathsf{succ}(\mathsf{x}),\mathsf{y},0,0) = \mathsf{mem}(\mathsf{x},\mathsf{succ}(\mathsf{y}),0,0), \\ 3: & \mathsf{mem}(\mathsf{succ}(\mathsf{x}),\mathsf{y},\mathsf{pred}(\mathsf{z}),0) = \mathsf{mem}(\mathsf{x},\mathsf{succ}(\mathsf{y}),\mathsf{pred}(\mathsf{z}),0), \\ 4: & \mathsf{mem}(\mathsf{pred}(\mathsf{x}),0,\mathsf{z},0) = \mathsf{mem}(\mathsf{x},0,\mathsf{pred}(\mathsf{z}),0), \\ 5: & \mathsf{mem}(\mathsf{pred}(\mathsf{x}),\mathsf{succ}(\mathsf{y}),\mathsf{z},0) = \mathsf{mem}(\mathsf{x},\mathsf{y},\mathsf{z},0), \\ 6: & \mathsf{mem}(0,\mathsf{succ}(\mathsf{y}),\mathsf{z},\mathsf{w}) = \mathsf{mem}(0,\mathsf{y},\mathsf{z},\mathsf{succ}(\mathsf{w})), \\ 7: & \mathsf{mem}(0,0,\mathsf{pred}(\mathsf{z}),\mathsf{w}) = \mathsf{mem}(0,0,\mathsf{z},\mathsf{pred}(\mathsf{w})), \\ 8: & \mathsf{mem}(0,0,0,\mathsf{w}) = \mathsf{w} \end{array} \right\}$$

Intuisjon for E_b: E_b er konvergent, så vi behandler E_b som et deterministisk program.

Termer «leses» sekvensielt utenfra og inn, og plasseres initielt i 1. argument av mem. Enhver succ havner i 2. argument av mem. Enhver pred havner i 3. argument av mem.

Betrakt ligninger 4 og 5: Leses en pred og det er er en overskudds-succ i 2. argument, kan pred'en og succ'en sløyfes. Dette ivaretar rollen til ligningen i E_a.

Betrakt ligninger 2 og 3: Leses en succ skal ingenting sløyfes, til tross for at det er en overskudds-pred i 3. argument. (Merk at vi i tillegg lett kan «implementere» basis-initialligningen $\mathsf{pred}(\mathsf{succ}(\mathsf{x})) = \mathsf{x}$ ved å endre ligning 3. Men her og nå kan ligning 2 og 3 slås sammen til $\mathsf{mem}(\mathsf{succ}(\mathsf{x}),\mathsf{y},\mathsf{z},0) = \mathsf{mem}(\mathsf{x},\mathsf{succ}(\mathsf{y}),\mathsf{z},0)$.)

Betrakt ligninger 6, 7 og 8: Ligningsmengden E_a er en spesifikasjon av en kanonisk-representant funksjon for $\simeq$. De kanoniske representanter er termer på formen $\mathsf{pred}^n\mathsf{succ}^m\mathsf{0}$. Annet argument i mem består nå kun av succ'er, og 3. argument bare av pred'er. Ligningene 6, 7 og 8 sørger da for kanoniske termer.

Hva skjer så ved komplettering av $E_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$? Vår komplettering terminerer ikke (ved et ligninende argument som det parentetiske i eksempel 61), men følgende regler blir blant andre, generert:

$$R_b = \left\{ \begin{array}{l} \vdots \\ \mathsf{delta(x)} \rightarrow \mathsf{x}, \\ \vdots \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \\ \vdots \end{array} \right\}$$

Det følger fra lemma 2.28 side 55 at enhver regel generert under en Knuth&-Bendix-komplettering vil være en logisk konsekvens av initialligningene. Følgelig vil den induktive teorien for et fullstendig $\mathcal{G}_{\mathsf{Int}}$-manifest inkludere den induktive teorien for

$$R_b^{man} = \left\{ \begin{array}{l} \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

Altså kan vi vise mangel på kjernebevaring ved

$$\mathsf{delta(pred(succ(0)))}!E_b = \mathsf{pred(succ(0))} \neq \mathsf{0} = \mathsf{delta(0)}!E_b$$

○

Kjerne-manifest kan altså brukes til å vise mangel på kjernebevaring for *id*-utvidelser.

Faktisk er det slik at dersom en *id*-utvidelse ikke er kjernebevarende, så vil et kjerne-manifest for *id*-utvidelsen *alltid* inneholde informasjon som viser dette faktum; gitt at manifestet er variabelbevarende:

Sats 3.29 *La Σ^c være en signatur av generatorer inneholdt i en signatur $\hat{\Sigma}$. Anta en vilkårlig $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$ inneholdende en indirekte spesifikasjon E_s av en semantikk $\simeq^s$ på $\mathcal{G}_{\Sigma^c}$ med spesifiserende symbol s. Anta vi har tilgjengelig et Σ^c-manifest $\hat{E}^{man}$ for $\hat{E}^{id}$. Anta at $\hat{E}^{man}$ er variabelbevarende.*

Anta $\hat{E}^{id}$ ikke er kjernebevarende. Da finnes en ligning/regel $v = h$ i $\hat{E}^{man}$ og en $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}}$, slik at

$$s(v\sigma) \not\overset{*}{\underset{E_s}{\leftrightarrow}} s(h\sigma)$$

Bevis: Siden $\hat{E}^{id}$ ikke er kjernebevarende, finnes $g_c, g_c' \in \mathcal{G}_{\Sigma^c}$ slik at

$$g_c \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} g_c' \quad \text{men} \quad s(g_c) \not\overset{*}{\underset{E_s}{\leftrightarrow}} s(g_c')$$

Ved at $\hat{E}^{man}$ er et Σ^c-manifest for $\hat{E}^{id}$, har vi også

$$g_c \overset{*}{\underset{\hat{E}^{man}}{\leftrightarrow}} g_c'$$

Anta så tvert imot at for alle ligninger $l = r$ i $\hat{E}^{man}$:

$$\forall \tau \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}} \mid s(l\tau) \overset{*}{\underset{E_s}{\leftrightarrow}} s(r\tau)$$

Siden E_s er en indirekte spesifikasjon av $\simeq^s$, er E_s indirekte $\mathcal{G}_{\Sigma^c}$-kongruent. Det er da lett å se at

$$s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g_c')$$

ved induksjon på lengden n av en vilkårlig $\hat{E}^{man}$-utledning $\langle g_c, \ldots, g'_c \rangle$:

$n = 1$: Trivielt.

$n = k + 1; k \geq 1$: Da har vi en $\hat{E}^{man}$-utledning $\langle g_c, \ldots, g_k, g'_c \rangle$. Merk at $g_k \in \mathcal{G}_{\Sigma^c}$, siden $\hat{E}^{man} \in \mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$ og $g_c \in \mathcal{G}_{\Sigma^c}$ og $\hat{E}^{man}$ er variabelbevarende. Induksjonshypotesen gir $s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g_k)$. Vi har så $g_k = c[l\rho]$ og $g'_c = c[r\rho]$, for en regel $l = r$ $(r = l) \in \hat{E}^{man}$, en $\rho \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}}$ og en kontekst $c \in \mathcal{G}_{\Sigma^c}$. Men siden vi antar $s(l\rho) \overset{*}{\underset{E_s}{\leftrightarrow}} s(r\rho)$, får vi siden E_s er indirekte $\mathcal{G}_{\Sigma^c}$-kongruent, at $s(c[l\rho]) \overset{*}{\underset{E_s}{\leftrightarrow}} s(c[r\rho])$. Dermed har vi

$$s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g'_c)$$

Vi har altså en motsigelse, og satsen følger.

□

Så hvis en s-id-utvidelse $\hat{E}^{id}$ ikke er kjernebevarende, finnes ved sats 3.29 i ethvert variabelbevarende kjerne-manifest for id-utvidelsen en ligning/regel $v = h$, slik at

$$s(v\sigma) \overset{*}{\underset{E_s}{\not\leftrightarrow}} s(h\sigma)$$

for en $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}}$.

Det å oppdage en slik ligning/regel og substitusjon kan gjøres algoritmisk dersom kjerne-manifestet $\hat{E}^{man}$ er endelig og E_s er konvergent. Dette kan gjøres ved et *systematisk* søk i universet av alle par $\langle l = r, \tau \rangle$ for ligninger/regler $l = r \in \hat{E}^{man}$ og $\mathcal{G}_{\Sigma^c}$-substitusjoner τ. Det er her essensielt at våre term-univers er rekursivt tellbare. Et systematisk søk kan uttrykkes ved *fairness*:

> *Søket er slik at dersom prosessen er uendelig, så oversees intet par $\langle l = r, \tau \rangle$ i det uendelige.*

Sats 3.29 sammen med en slik systematisk gjennomgang av universet av alle par $\langle l = r, \tau \rangle$ kan selvfølgelig generelt ikke alene brukes for å verifisere at en $\hat{E}^{id}$ *er* kjernebevarende, siden søkerommet generelt er uendelig.

Men finnes det muligheter for å avskjære et slikt søk etter endelig tid med visshet om kjernebevaring? En slik avskjæring kunne være bestemt av kompleksiteten på grunntermene substituert inn av $\mathcal{G}_{\Sigma^c}$-substitusjonene τ, forutsatt at det systematiske søket foregår med stigende slik kompleksitet.

Våre eksempler 64, 65 og 66 på ikke kjernebevarende id-utvidelser, er alle slik at det finnes en regel i de ved Knuth&Bendix-komplettering genererte kjerne-manifestene, slik at den minst komplekse $\mathcal{G}_{\Sigma^c}$-substitusjonen er tilstrekkelig for å etablere mangel på kjernebevaring.

Man kan spekulere om det alltid er slik for Knuth&Bendix-komplettering genererte kjerne-manifest. Isåfall ville kjernebevaring av id-utvidelser være avgjørbart, gitt endelige kjerne-manifest generert ved komplettering og konvergent indirekte spesifikasjon.

Merk at det finnes tenkbare kjerne-manifest hvor den minst komplekse substitusjon ikke er tilstrekkelig for å vise mangel på kjernebevaring. Vi kan gi et eksempel; men manifestet er *ikke* generert ved komplettering og har her heller ikke opphav i en spesifik indirekte spesifikasjon:

Eksempel 67 Anta en konvergent indirekte spesifikasjon E'_n for $\simeq^{\mathcal{I}nt^{pos}_n}_{\mathsf{Int}}$ fra eksempel 65. Anta så følgende $\mathcal{G}_{\mathsf{Int}}$-manifest for den aktuelle id-utvidelse av E'_n:

$$E_n^{man'} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{pred}(\mathsf{x})) = \mathsf{x}, \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) = \mathsf{x}, \\ \mathsf{succ}^n(\mathsf{x}) = \mathsf{x} \end{array} \right\}$$

Det er her ikke tilstrekkelig å instansiere x med 0. For en gitt n kan likevel id-utvidelsen av E'_n vises å være ikke kjernebevarende, ved å konstatere at

$$\mathsf{delta}(\mathsf{pred}(\mathsf{succ}(\mathsf{succ}^n(0))))!E'_n \neq \mathsf{delta}(\mathsf{succ}^n(0))!E'_n$$

Men merk også at vi kan sette grensen for en slik test vilkårlig høyt ved å velge vilkårlig stor n.
○

Uansett om hypotesen om tilstrekkelighet av minst komplekse substitusjon for å vise mangel på kjernebevaring vha. Knuth&Bendix-komplettering genererte manifest holder eller ikke, kan man undres om det finnes en *dynamisk* øvre grense på søketiden som kun avhenger av den aktuelle indirekte spesifikasjon og det genererte manifest.

Spesielt interessant er dette, hvis en slik øvre grense kan bestemmes algoritmisk utfra f.eks. syntaktiske egenskaper ved de aktuelle ligningsmengder.

Merk at vi her snakker om avgjørbarhet av kjernebevaring av id-utvidelser, gitt *endelige kjerne-manifest generert ved komplettering og konvergent indirekte spesifikasjon*, og på ingen måte *generell* avgjørbarhet av kjernebevaring av id-utvidelser.

3.6.2 Synliggjøring av kjernebevaring

Et kjerne-manifest kan i en viss forstand også åpenbare direkte at en id-utvidelse er kjernebevarende. Dette er tilfellet dersom vi vet at kjernen til id-utvidelsen er identisk med semantikken spesifisert av manifestet. Altså at vi vet for en kjerne $\simeq^s$ på en $\mathcal{G}_{\Sigma^c}$, en id-utvidelse $\hat{E}^{id}$ og et kjerne-manifest $\hat{E}^{man}$ for $\hat{E}^{id}$, at

$$\simeq^s = \overset{*}{\underset{\hat{E}^{man}}{\leftrightarrow}} \mathcal{G}_{\Sigma^c}$$

Ialt har vi da

$$\simeq^s = \overset{*}{\underset{\hat{E}^{man}}{\leftrightarrow}} \mathcal{G}_{\Sigma^c} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}} \mathcal{G}_{\Sigma^c}$$

som gir at $\hat{E}^{id}$ er kjernebevarende.

Viten om at $\simeq^s = \overset{*}{\underset{\hat{E}^{man}}{\leftrightarrow}} \mathcal{G}_{\Sigma^c}$ er ingen selvfølgelighet. Denne viten fordrer at man vet hva kjernen $\simeq^s$ faktisk er, dvs. at man i en viss forstand har verifisert den aktuelle indirekte spesifikasjon.

Eksempel 68 Betrakt E_{delta} fra eksempel 56 på side 108.

Ved den intuisjon gitt for E_{delta} i eksemplet, har vi mer eller mindre godtgjort at E_{delta} er en algebraisk spesifikasjon av kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32 på side 64. I sistnevnte eksempel konstaterte vi også at

$$R_{\mathcal{I}nt} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{pred}(\mathsf{x})) \rightarrow \mathsf{x} \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) \rightarrow \mathsf{x} \end{array} \right\}$$

beregner $\delta_{\mathcal{I}nt}$ ved prinsipp (3.1) på side 63. Altså spesifiserer den indirekte spesifikasjon E_{delta} og den direkte spesifikasjon $R_{\mathcal{I}nt}$ den samme semantikk (kall den her $\simeq^{\mathsf{delta}}$) på $\mathcal{G}_{\mathsf{Int}}$.

Betrakt nå delta-id-utvidelsen E^{id}_{delta} av E_{delta}. Fra eksempel 61 side 113 har vi følgende $\mathcal{G}_{\mathsf{Int}}$-manifest for E^{id}_{delta}:

$$R^{man}_{\mathsf{delta}} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{pred}(\mathsf{x})) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) \rightarrow \mathsf{x} \end{array} \right\}$$

Men nå vet vi jo at

$$\simeq^{\mathsf{delta}} = \overset{*}{\underset{R_{\mathcal{I}nt}}{\leftrightarrow}} \mathcal{G}_{\mathsf{Int}} = \overset{*}{\underset{R^{man}_{\mathsf{delta}}}{\leftrightarrow}} \mathcal{G}_{\mathsf{Int}}$$

Og R^{man}_{delta} er jo et $\mathcal{G}_{\mathsf{Int}}$-manifest for E^{id}_{delta}:

$$\overset{*}{\underset{R^{man}_{\mathsf{delta}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}} = \overset{*}{\underset{E^{id}_{\mathsf{delta}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}}$$

Følgelig må E^{id}_{delta} være kjernebevarende.
○

3.6.3 Kjernebevaring av *id*-utvidelser og inkonsistens

Om en *id*-utvidelse er kjernebevarende er her først og fremst interessant, dersom *id*-utvidelsen er reduserbar til basis-initialsemantikk (avsnitt 3.4.5). Isåfall kan nemlig kjernebevaring av *id*-utvidelsen direkte overføres til initiell konsistens, siden vi da har

$$\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}\mathcal{G}_{\Sigma^c} = \simeq^s \;\Leftrightarrow\; \simeq^{\alpha}_{\mathcal{G}_{\Sigma^c}} = \simeq^s$$

for Σ^c en signatur av generatorer og $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$ inneholdende en indirekte spesifikasjon av en $\simeq^s$ på $\mathcal{G}_{\Sigma^c}$ med spesifiserende symbol s.

Da representerer dessuten eventuelle metoder for å etablere kjernebevaring/ oppdage mangel på kjernebevaring av *id*-utvidelser alternativer og suppleringer til metoder for å etablere/oppdage initiell konsistens/inkonsistens.

Eksempel 69 Betrakt signaturene Int og $\mathbf{\Delta}_{\mathsf{Int}}$ og spesifikasjonen $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77. La $\hat{\Sigma} = \mathsf{Int} \cup \mathbf{\Delta}_{\mathsf{Int}}$.

I eksempel 53 side 93 argumenterte vi for at $\overset{*}{\underset{E^{id}}{\leftrightarrow}}\mathcal{G}_{\hat{\Sigma}} = \simeq^{\alpha}$, for delta-*id*-utvidelsen $\overset{*}{\underset{E^{id}}{\leftrightarrow}}$ av $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$, og for den initielle semantikk $\simeq^{\alpha}$ relativ til $\simeq^{\mathsf{delta}}$ bestemt av $\hat{\Sigma}$ og $E_{\delta_{\mathcal{I}nt}}$. Her er $\simeq^{\mathsf{delta}}$ på $\mathcal{G}_{\mathsf{Int}}$ indirekte spesifisert ved $E_{\delta_{\mathcal{I}nt}}$.

Da kan kjernebevaring av $\overset{*}{\underset{E^{id}}{\leftrightarrow}}$ overføres til initiell konsistens av $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$ relativt til $\simeq^{\mathsf{delta}}$.

I eksempel 64 side 116 konstaterte vi at $\overset{*}{\underset{E^{id}}{\leftrightarrow}}$ ikke er kjernebevarende. Ergo er ikke $E_{\delta_{\mathcal{I}nt}} \cup E_{kompl}$ initielt konsistent relativt til $\simeq^{\mathsf{delta}}$.
○

Eksempel 70 Betrakt den indirekte spesifikasjonen E_n fra eksempel 65 på side 116. Vi konstaterte mangel på kjernebevaring for delta-*id*-utvidelsen av E_n (for $n = 2$). Vi kan overføre dette til initiell inkonsistens, ved å vise at delta-*id*-utvidelsen av E_n kan reduseres til basis-initialsemantikk:

Merk at E_n er delta-komplett, men *ikke* mem-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$: Termen $\mathsf{mem(succ(0),succ(0),succ(0))}$ er eksempelvis ikke omskrivbar i E_n til noen term i $\mathcal{G}_{\mathsf{Int}}$. Dette er i tråd med vår her sekvensielle tenkemåte at et steg i prosesseringen av en term gjøres unna før neste steg settes igang: En term reverseres fullstendig før videre prosessering.

Ønsker vi å vise reduksjon av delta-*id*-utvidelsen av E_n til basis-initialsemantikk, kan vi ty til teorem 3.10 på side 95, siden E_n oppfyller kriteriene for satsen. Merk spesielt her det «strenge» kriteriet $h\mathtt{ROT}$.

Vi kunne i prinsipp også modifisere E_n til å være mem-komplett og bruke teorem 3.8 på side 92 istedet. En slik modifikasjon er her ikke så lett (i motsetning til modifikasjonen gjort i eksempel 53 side 93). Vi kunne forsøke en rett

fram modifikasjon av E_n til en mem-komplett $E_{n\,kompl}$ som følger:

$$E_{n\,kompl} = \left\{ \begin{array}{ll} 1: & \mathsf{delta(x) = mem(x,0,0)}, \\ 2: & \mathsf{mem(succ(x),0,y) = mem(x,succ(0),y)}, \\ 3: & \mathsf{mem(pred(x),0,y) = mem(x,pred(0),y)}, \\ 4: & \mathsf{mem(w,pred(x),0) = mem(w,x,pred(0))}, \\ 5_0: & \mathsf{mem(w,succ(x),0) = mem(w,x,succ(0))}, \\ 5_1: & \mathsf{mem(w,succ(x),succ(0)) = mem(w,x,succ(succ(0)))}, \\ \vdots & \vdots \\ 5_i: & \mathsf{mem(w,succ(x),succ^{\mathit{i}}(0)) = mem(w,x,succ^{\mathit{i+1}}(0))}, \\ \vdots & \vdots \\ 5_n: & \mathsf{mem(w,succ(x),succ^{\mathit{n}}(0))) = mem(w,x,0)}, \\ 6: & \mathsf{mem(w,succ(x),pred(y)) = mem(w,x,y)}, \\ 7: & \mathsf{mem(w,pred(x),succ(y)) = mem(w,x,y)}, \\ 8: & \mathsf{mem(w,pred(x),pred(y)) = mem(w,x,pred(pred(y)))}, \\ 9: & \mathsf{mem(0,0,x) = x} \end{array} \right\}$$

Dette er katastrofalt, siden vi da får $E_{n\,kompl}$-utledningen

$$\begin{array}{l} \langle \mathsf{delta(pred(succ(succ(succ(0)))))}, \\ \qquad \mathsf{mem(pred(succ(succ(succ(0)))))},0,0), \\ \qquad \mathsf{mem(succ(succ(succ(0))))},\mathsf{pred(0)},0), \\ \qquad \mathsf{mem(succ(succ(succ(0))))},0,\mathsf{pred(0)}), \\ \qquad \mathsf{mem(succ(succ(0)))},\mathsf{succ(0)},\mathsf{pred(0)}), \\ \qquad\quad \mathsf{mem(succ(succ(0)))},0,0), \ldots \\ \qquad\qquad\qquad\qquad\qquad\qquad \ldots, \mathsf{succ(succ(0))} \rangle \end{array}$$

Det er nemlig essensielt for idéen bak E_n, at termer reverseres fullstendig før videre prosessering. (Vi kan ikke åpne opp for parallell prosessering.) En mer gjennomtenkt utvidelse av E_n; eller en annen strategi enn E_n, må altså til for å oppnå mem-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$. Men heldigvis har vi altså teorem 3.10.
○

Eksempel 71 Betrakt den indirekte spesifikasjonen E_b fra eksempel 66 på side 117. Vi konstaterte mangel på kjernebevaring for delta-*id*-utvidelsen av E_b. Vi kan overføre dette til initiell inkonsistens, ved å vise at delta-*id*-utvidelsen av E_b kan reduseres til basis-initialsemantikk:

Merk også her at E_b er et tilfelle der teorem 3.10 på side 95 med stor fordel kan benyttes for å vise reduksjon til basis-initialsemantikk. Eksempelvis er termen $\mathsf{mem(succ(0),succ(0),succ(0),succ(0))}$ ikke omskrivbar i E_b til noen term i $\mathcal{G}_{\mathsf{Int}}$. Forsøker vi igjen en rett fram utvidelse for å oppnå mem-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$, får vi problemer lignende dem som oppsto i eksempel 70.
○

3.6.4 Synliggjøring av kjernesemantikk

Dersom vi vet at en *id*-utvidelse av en indirekte spesifikasjon er kjernebevarende, kan et kjerne-manifest bidra til å synliggjøre kjernesemantikken i form av en direkte algebraisk spesifikasjon uten «fremmede» symboler. Dette er på sett og vis speilvendingen av diskusjonen i avsnitt 3.6.2.

La Σ^c være en signatur av generatorer. Anta $\hat{E} \subseteq \mathcal{E}(\mathcal{T}_{\hat{\Sigma}}(\mathcal{V}))$ inneholder en indirekte spesifikasjon av en $\simeq^s$ på $\mathcal{G}_{\Sigma^c}$ med spesifiserende symbol s. Anta et Σ^c-manifest $\hat{E}^{man}$ for $\hat{E}^{id}$. Anta altså at $\hat{E}$ er kjernebevarende. Da har vi

$$\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}_{\mathcal{G}_{\Sigma^c}} = \simeq^s$$

På den annen side har vi

$$\overset{*}{\underset{\hat{E}}{\leftrightarrow}}{}^{id}\mathcal{G}_{\Sigma^c} = \overset{*}{\underset{\hat{E}}{\leftrightarrow}}{}^{man}\mathcal{G}_{\Sigma^c}$$

og dermed

$$\overset{*}{\underset{\hat{E}}{\leftrightarrow}}{}^{man}\mathcal{G}_{\Sigma^c} = \simeq^s$$

Vi kan da betrakte $\hat{E}^{man}$, og det kan muligens bli åpenbart hvorvidt $\simeq^s$ er den generator-/kjernesemantikk vi er ute etter:

Eksempel 72 Vi godtgjorde i eksempel 55 side 105 at $\hat{E} = E_{\mathsf{synt}} \cup E^d$ tilfredstiller `KREP2`. Men ved `DISJ` er denne godtgjørelse også en godtgjørelse for at E_{synt} alene tilfredstiller `KREP2`. Dessuten oppfylte $\hat{E}$ kriteriene `DISJ` og E_s`VAR-BEVAR`, samt $\Sigma^h = \emptyset$. Disse ting er selvfølgelig også oppfylt for E_{synt} alene. For $\mathsf{Int} = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}\}$ er derfor ved teorem 3.17 side 102

$$\overset{*}{\underset{E^{id}_{\mathsf{synt}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}} = \simeq^{\alpha}_{\mathcal{G}_{\mathsf{Int}}}$$

for den initielle semantikken $\simeq^{\alpha}$ spesifisert av $\mathsf{Int} \cup \{\mathsf{synt}\}$ og E_{synt} relativ til $\simeq^{\mathsf{synt}}$, hvor $\simeq^{\mathsf{synt}}$ er den indirekte semantikken spesifisert av E_{synt}.

Fra eksempel 59 side 110 har vi at E_{synt} er initielt konsistent relativt til $\simeq^{\mathsf{synt}}$. Altså har vi at E^{id}_{synt} er kjernebevarende, dvs.

$$\overset{*}{\underset{E^{id}_{\mathsf{synt}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}} = \simeq^{\mathsf{synt}}$$

Nå vet vi altså at $\overset{*}{\underset{E^{id}_{\mathsf{synt}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}} = \overset{*}{\underset{R^{man}_{\mathsf{synt}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}}$ for

$$R^{man}_{\mathsf{synt}} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{pred}(\mathsf{x})) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) \rightarrow \mathsf{x} \end{array} \right\}$$

fra eksempel 60 side 113.

Vi kan nå si at E_{synt} er en korrekt indirekte spesifikasjon av semantikk på $\mathcal{G}_{\mathsf{Int}}$, i den grad vi vet at $\overset{*}{\underset{R^{man}_{\mathsf{synt}}}{\leftrightarrow}}\mathcal{G}_{\mathsf{Int}}$ er semantikken vi ønsker.
○

Vi sier mer om dette i avsnitt 3.9.

3.6.5 *id*-utvidelser og inkongruens

Vi har til nå, både i teori og i eksempler, antatt `INDKONG`; altså at den aktuelle E_s faktisk *er* en indirekte spesifikasjon av en eller annen kongruensrelasjon $\simeq^s$. Anta en E_s ment å spesifisere en indirekte semantikk på en $\mathcal{G}_{\Sigma^c}$, *ikke* er indirekte $\mathcal{G}_{\Sigma^c}$-kongruent. Likevel er selvfølgelig $\overset{*}{\underset{E^{id}_s}{\leftrightarrow}}$ en kongruensrelasjon. Spesielt er da også restriksjonen $\overset{*}{\underset{E^{id}_s}{\leftrightarrow}}\mathcal{G}_{\Sigma^c}$ en kongruensrelasjon. Hvis `INDKONG` ikke er etablert, kan man da i en viss forstand ikke stole på det man ser:

Eksempel 73 Betrakt $E'_{\delta_{\mathcal{I}nt}}$ fra eksempel 49 side 83. Vi husker at $E'_{\delta_{\mathcal{I}nt}}$ ikke er indirekte $\mathcal{G}_{\mathsf{Int}}$-kongruent. Men en terminerende vellykket komplettering av $E'_{\delta_{\mathcal{I}nt}} \cup \{\mathsf{delta}(\mathsf{x}) = \mathsf{x}\}$, gir regelmengden

$$R'_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{delta}(\mathsf{x}) \rightarrow \mathsf{x}, \\ \mathsf{succ}(\mathsf{pred}(\mathsf{x})) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{succ}(\mathsf{x})) \rightarrow \mathsf{x} \end{array} \right\}$$

○

Eksempel 74 Betrakt følgende

$$E = \left\{ \begin{array}{l} \mathsf{delta}(\mathsf{succ}(\mathsf{pred}(\mathsf{x}))) = \mathsf{x}, \\ \mathsf{delta}(\mathsf{pred}(\mathsf{succ}(\mathsf{x}))) = \mathsf{x} \end{array} \right\}$$

E er ikke indirekte $\mathcal{G}_{\mathsf{Int}}$-kongruent. Men en terminerende vellykket komplettering av $E \cup \{\mathsf{delta(x) = x}\}$, gir igjen regelmengden

$$R = \left\{ \begin{array}{l} \mathsf{delta(x)} \rightarrow \mathsf{x}, \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

○

Disse eksemplene illustrerer viktigheten av å kunne etablere `INDKONG`. Visste vi ikke bedre kunne vi i farten tro at manifestet

$$\left\{ \begin{array}{l} \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

for id-utvidelsene av $E'_{\delta_{\mathcal{I}nt}}$ og E i eksemplene 73 og 74 synliggjør kjernebevaring for disse id-utvidelser. Merk dessuten at inkonsistens i disse eksemplene i prinsipp kan utelukkes ved sats 3.22 side 109. Da kunne man også tro at manifestet over synliggjør kjernesemantikken. Dette er selvfølgelig uten mening, siden det her ikke finnes noen kjerne.

Men merk likevel at id-utvidelsene av disse «feilaktige» indirekte spesifikasjonene faktisk gir en (fornuftig) semantikk.

Mangel på indre kongruens kan oppdages algoritmisk; gitt konvergent indirekte spesifikasjon som er tilstrekkelig s-komplett for spesifiserende symbol s. Dette kan gjøres ved å beregne og lagre normalformer $s(g)!$ på en systematisk måte helt til følgende inntreffer:

$$s(c[g])! \neq s(c[g'])! \quad \text{og} \quad s(g)! = s(g')!$$

Det systematiske søket i termuniverset må være slik:

> *Ingen term må oversees uendelig lenge, og dersom $s(g)!$ for en term g beregnes, må $s(g')!$ for alle subtermer g' av g allerede være beregnet.*

Igjen er det essensielt at våre term-univers er rekursivt tellbare.

Og igjen er det selvfølgelig umulig generelt å bruke utelukkende et slikt systematisk søk til å etablere et *positivt* resultat; dvs. at en indirekte spesifikasjon *er* indre kongruent.

3.6.6 Nytten av *id*-utvidelser som basis-initialsemantikk spesifikatorer

Flere (semi-)mekaniske resolusjonsmetoder for basis-initial- og basis-finalsemantikker forutsetter at ligningsmengden som spesifiserer semantikken er grunnkonvergent.

Vi har redusert initialsemantikk relativ til indirekte semantikk til basis-initialsemantikk ved hjelp av id-utvidelser. For å bruke resolusjonsmetoder som fordrer grunnkonvergente spesifikasjons-ligninger, må da en for id-utvidelsen grunnkomplett og altså grunnkonvergent regelmengde fremskaffes.

Ved Knuth&Bendix-komplettering av id-utvidelser under de forutsetninger beskrevet i sats 3.27 på side 112, vil ifølge sats 3.27 en komplett regelmengde for en id-utvidelse *kun genereres dersom det finnes en direkte algebraisk spesifikasjon av den indirekte semantikken.*

I lys av resolusjonsmetoder som fordrer konvergente spesifikasjons-ligninger, og dersom Knuth&Bendix-komplettering brukes for å fremskaffe slike konvergente ligninger, er således bruken av id-utvidelser noe søkt: Man kunne heller brukt en (vanlig) direkte algebraisk spesifikasjon.

Indirekte spesifikasjon og *id*-utvidelser er i denne sammenheng likevel hensiktsmessig dersom det er lettere å spesifisere en semantikk indirekte enn direkte. Vår indirekte spesifikasjon ser ut til å representere en *operasjonell* stil innen spesifikasjon. Dette er ikke rart, siden vi stort sett har tatt utgangspunkt i spesifikasjoner/beskrivelser av syntaktiske funksjoner, og syntaktiske funksjoner opererer på term-struktur. Det er en smaksak, men har man vendt seg til den operasjonelle måten å spesifisere semantikk på som indirekte spesifikasjon ser ut til å representere, kan man *i noen situasjoner* synes indirekte spesifikasjon er lettere enn direkte spesifikasjon.

Eksempel 75 Betrakt den indirekte spesifikasjonen E_n i eksempel 65 side 116. En tilsvarende direkte algebraisk spesifikasjon er

$$E_n^{dir} = \left\{ \begin{array}{l} \mathsf{pred(succ(0))} = \mathsf{0}, \\ \vdots \\ \mathsf{pred(succ}^n\mathsf{(0))} = \mathsf{succ}^{n-1}\mathsf{(0)}, \\ \mathsf{succ}^{n+1}\mathsf{(0)} = \mathsf{0}, \\ \mathsf{succ(pred(x))} = \mathsf{x} \end{array} \right\}$$

Personlig synes jeg at det falt lettere å spesifisere den indirekte spesifikasjonen E_n enn den direkte spesifikasjonen E_n^{dir}. Det er muligens også lettere å se at den indirekte spesifikasjonen er korrekt, enn å se at den direkte spesifikasjonen er korrekt. (Nå er vel i utgangspunktet ingen av disse to spesifikasjoner spesielt kompliserte.)

Merk dog at *id*-utvidelsen av den indirekte spesifikasjonen jo desverre ikke er kjernebevarende (eksempel 65).
○

Det er også verd å merke seg at at slike operasjonelle indirekte spesifikasjoner også naturlig er tilgjengelige for metoder innen programverifikasjon. Se avsnitt 3.9.

Indirekte spesifikasjoner (alene eller i *id*-utvidelse) er også spesielt hensiktsmessige dersom man kan spesifisere flere, eller andre semantikker indirekte enn man kan ved direkte spesifikasjon.

3.7 Skjuling av hjelpefunksjoner

Utgangspunktet for indirekte spesifikasjon har vært algebraiske spesifikasjoner/beskrivelser av semantikkgivende syntaktiske funksjoner. Disse beskrivelser av syntaktiske funksjoner har ofte involvert beskrivelser av hjelpefunksjoner til den semantikkgivende syntaktiske funksjonen, og vi har sett at disses beskrivelser kan innføre inkonsistens relativt til den indirekte semantikk.

Den inkonsistens som på denne måten blir innført pga. disse hjelpefunksjoner, er på en måte *kunstig*. Hjelpefunksjonene er opprinnelig tenkt som implementatoriske tekniske hjelpemidler, og burde være skjult fra logikken. Hjelpefunksjoner er ikke nødvendigvis kun knyttet til syntaktiske funksjoner, men vår tanke om at hjelpefunksjoner burde skjules i en passende forstand er spesielt nærliggende i lys av den operasjonelle stilen vi har sett i våre algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner.

La oss igjen adoptere den formelle omgivelse som definert i figur 3.7 på side 85. Vi ønsker å gi semantikk til $\mathcal{G}_\Sigma$ for $\Sigma = \Sigma^c \cup \Sigma^d$. Vi ønsker å spesifisere generatorsemantikk ved $\simeq^s$ og semantikk til definerte symboler i Σ^d ved E^d. Det er da naturlig å betrakte

initialsemantikken $\simeq^{\alpha\prime}$ relativ til $\simeq^s$ spesifisert av E^d og Σ

med grensesnitt-/logisk omgivelse

$$\langle \mathcal{G}_\Sigma, E^d \rangle$$

I våre reduksjoner til basis-final- og basis-initialsemantikker benyttet vi oss av en utvidet formell omgivelse $\langle \mathcal{G}_{\hat{\Sigma}}, \hat{E} \rangle$, og så på semantikker ($\simeq^\alpha$ og $\simeq^\omega$) relative til $\simeq^s$ spesifisert av $\hat{E}$ og $\hat{\Sigma}$. Den formelle omgivelsen $\langle \mathcal{G}_{\hat{\Sigma}}, \hat{E} \rangle$ gjør oss istand til å utføre reduksjonene; bl.a. kan vi definere *id*-utvidelser. Men i forhold til den logiske omgivelse $\langle \mathcal{G}_\Sigma, E^d \rangle$ innføres en ny mulighet for inkonsistens, ved at definerte symboler i E_s nå deltar i spesifiseringen av semantikken på $\mathcal{G}_\Sigma$. I lys av tanken om at slik inkonsistens er kunstig, er da den utvidete omgivelse $\langle \mathcal{G}_{\hat{\Sigma}}, \hat{E} \rangle$ ikke den logiske omgivelse vi ønsker. (Se også figurene 3.5 side 82 og 3.6 side 83.)

Vi ser i dette avsnittet på to måter å «avvæpne», eller skjule, hjelpefunksjonssymboler med hensyn på inkonsistens. Den ene måten er operasjonell. Den andre måten skal utføres på spesifikasjonsnivå.

3.7.1 Operasjonell skjuling

Vi skal innføre en operasjonell restriksjon i basis-initialsemantikk spesifisert av *id*-utvidelser av indirekte spesifikasjoner. Vi skal dessuten arbeide mer med relasjonen definert i (2.13) på side 41. Vi fører diskusjonen med signaturer, ligningsmengder osv. som i figur 3.7 side 85.

Vi skal vise hvordan *s-id*-utvidelser sammen med en restriktert form for omskrivning, kan gi initialsemantikken $\simeq^{\alpha\prime}$ relativ til $\simeq^s$ spesifisert av E^d og Σ.

Betrakt først relasjonen

$$(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$$

Dette er en refleksiv-transitiv relasjon, og et steg $g_i (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}) g_{i+1}$ i en $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$-utledning er slik at enten

$$g_i \simeq^s g_{i+1}$$

eller

$$g_i \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_{i+1}$$

Ingen kontekstapplikasjon er mulig. (Substistusjon er selvfølgelig uaktuell på grunntermer.)

Vi skal nå definere *delvis monoton tillukning*:

Definisjon 3.9 *For en relasjon $\Re$ på en $\mathcal{T}_\Sigma(\mathcal{V})$ definerer vi den* ***Σ'-monotone tillukning***

$$\Re^{\overset{*}{\cdot\Sigma'\cdot}}$$

av $\Re$ for en $\Sigma' \subseteq \Sigma$ som den minste relasjon som er

1. *monoton mhp. substitusjon for alle substitusjoner i $\mathcal{S}bst^{\mathcal{T}_{\Sigma'}(\mathcal{V})}$.*
2. *monoton mhp. kontekstapplikasjon for alle kontekster $c \in \mathcal{T}_{\Sigma'}(\mathcal{V})$.*

Anvendt på $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$ kan delvis monotonitet således gi forskjellige grader av kontekstapplikasjon; sagt billedlig: forskjellige grader av samarbeid mellom $\simeq^s$ og $\overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$.

Eksempelvis er full monotonitet $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\cdot\hat{\Sigma}\cdot}}$ av $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$ identisk med initialsemantikk relativ til $\simeq^s$ spesifisert av $\hat{E}$ og $\hat{\Sigma}$. Se definisjon 2.9 på side 31.

Vi definerer så

Definisjon 3.10 *For en ligningsmengde E, betrakt s-id-utvidelsen $E^{id} = E \cup \{s(x) = x\}$. Vi definerer Σ'-s-id-**restriksjonen***

$$\overset{\Sigma'}{\underset{E^{id}}{\rightsquigarrow^{s}}}$$

av $\overset{*}{\underset{E^{id}}{\leftrightarrow}}$, som relasjonen $\overset{*}{\underset{E^{id}}{\leftrightarrow}}$, men med den restriksjon som følger av at ligningen $s(x) = x$ kun anvendes i kontekster $c \in \mathcal{T}_{\Sigma'}(\mathcal{V})$ ved omskriving.

Følgende sats er sentral:

Sats 3.30 *Anta* `TΣ̂K` *og* `KREP1`. *La $\Sigma' \subseteq \hat{\Sigma}$. Vi har*

$$\overset{\Sigma'}{\underset{\dot{E}^{id}}{\rightsquigarrow^{s}}} \mathcal{G}_{\hat{\Sigma}} = (\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'}$$

Bevis: For inklusjonen

$$\overset{\Sigma'}{\underset{\dot{E}^{id}}{\rightsquigarrow^{s}}} \mathcal{G}_{\hat{\Sigma}} \subseteq (\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'}$$

betrakt en vilkårlig $\overset{\Sigma'}{\underset{\dot{E}^{id}}{\rightsquigarrow^{s}}}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$, for vilkårlige $g, g' \in \mathcal{G}_{\hat{\Sigma}}$. Vi induserer over lengden på en slik utledning. Basisen er triviell. For induksjonssteget gir induksjonshypotesen

$$g \; (\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'} \; g_k$$

for en g_k i utledningen slik at $\langle g, \ldots, g_k, g' \rangle$. Anta $g_k \overset{*}{\underset{\dot{E}}{\leftrightarrow}} g'$. Da har vi trivielt

$$g_k \; (\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'} \; g'$$

og induksjonssteget følger.

Anta $g_k \overset{}{\underset{\{s(x)=x\}}{\longleftrightarrow}} g'$. Da er $g_k = c[g'_k]$ og $g' = c[s(g'_k)]$, eller $g_k = c[s(g'_k)]$ og $g' = c[g'_k]$, for en $c \in \mathcal{G}_{\Sigma'}$. Resten går på samme måte som i beviset for lemma 3.9 på side 92: Siden vi antar `TΣ̂K` og `KREP1`, og siden vi implisitt antar `INDKONG` ved å nevne $\simeq^{s}$, har vi ved (3.13) side 92:

$$g'_k \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \overline{g'_k} \simeq^{s} \overset{\sim}{\overline{g'_k}} \overset{*}{\underset{\dot{E}}{\leftrightarrow}} s(\overline{g'_k}) \overset{*}{\underset{\dot{E}}{\leftrightarrow}} s(g'_k)$$

Vi kan applisere konteksten c rundt dette og vi får:

$$c[g'_k] \overset{*}{\underset{\dot{E}}{\leftrightarrow}} c[\overline{g'_k}]$$

og

$$c[\overset{\sim}{\overline{g'_k}}] \overset{*}{\underset{\dot{E}}{\leftrightarrow}} c[s(\overline{g'_k})] \overset{*}{\underset{\dot{E}}{\leftrightarrow}} c[s(g'_k)]$$

der altså $\overline{g'_k} \simeq^{s} \overset{\sim}{\overline{g'_k}}$. Nå var $c \in \mathcal{G}_{\Sigma'}$, så vi har altså (evt. ved symmetri)

$$g_k \; (\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'} \; g'$$

og induksjonssteget følger. Inklusjonen

$$\overset{\Sigma'}{\underset{\dot{E}^{id}}{\rightsquigarrow^{s}}} \mathcal{G}_{\hat{\Sigma}} \supseteq (\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'}$$

er også grei å vise: Vi induserer over lengden n av en vilkårlig $(\simeq^{s} \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$. Basis er triviell. For induksjonssteget gir induksjonshypotesen

$$g \; \overset{\Sigma'}{\underset{\dot{E}^{id}}{\rightsquigarrow^{s}}} \; g_k$$

for en $g_k \in \mathcal{G}_{\hat{\Sigma}}$. Anta

$$g_k \overset{*}{\underset{\dot{E}}{\leftrightarrow}} g'$$

Da er trivielt $g_k \overset{\Sigma' * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}} g'$, og induksjonssteget følger. Anta så at

$$g_k = c[g'_k] \;\text{ og }\; c[g''] = g' \;\text{ og }\; g'_k \simeq^s g''$$

for en vilkårlig (muligens tom) kontekst $c \in \mathcal{G}_{\Sigma'}$. Vi har da, siden $g'_k \simeq^s g'' \;\Leftrightarrow\; s(g'_k) \overset{*}{\underset{\dot{E}\,s}{\leftrightarrow}} s(g'')$

$$g_k = c[g'_k]\,{}_{\{s(x)=x\}}\!\longleftrightarrow c[s(g'_k)] \overset{*}{\underset{\dot{E}\,s}{\leftrightarrow}} c[s(g'')]\,{}_{\{s(x)=x\}}\!\longleftrightarrow c[g''] = g'$$

Konteksten c er i $\mathcal{G}_{\Sigma'}$, så vi har $g_k \overset{\Sigma' * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}} g'$, og induksjonssteget følger.
□

Merk at vi kan vise sats 3.30 for antagelsene `TsK`, *h*`ROT`, samt E_s`VARBEVAR`, E^d`-VARBEVAR` og `DISJ`, istedenfor antagelsen `T`$\hat{\Sigma}$`K`. Der linjen i beviset for sats 3.30 følger tråden i beviset for lemma 3.9 på side 92, skal heller tråden i beviset for lemma 3.16 på side 101 følges. Man må dessuten forsikre seg om at relasjonen $\overset{\Sigma' * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}}$ er lukket mhp. transformasjonen for lemma 3.11 på side 96. Dette gjøres enkelt ved å observere at ved et transformasjonssteg forblir kontekstene rundt anvendelser av ligningen $s(x)=x$ uforandrede, eller blir delkontekster av de opprinnelige. Vi kan altså slå fast:

Sats 3.30b *Anta* `TsK` *og* `KREP1`*. Anta i tillegg* *h*`ROT`*, samt* E_s`VARBEVAR`*,* E^d`-VARBEVAR` *og* `DISJ`*. La* $\Sigma' \subseteq \hat{\Sigma}$*. Vi har*

$$\overset{\Sigma' * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}} \mathcal{G}_{\hat{\Sigma}} = (\simeq^s \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\Sigma'}$$

—

For initialsemantikken $\simeq^{\alpha\prime}$ relativ til $\simeq^s$ spesifisert av E^d og Σ, skal vi så vise vårt hovedresultat i dette avsnittet:

Teorem 3.31 *Anta* `T`$\hat{\Sigma}$`K`*,* `DISJ` *og* `KREP1`*. Anta i tillegg* `KONVERG` *og* `KONSTR`*. Da har vi*

$$\overset{\Sigma * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}} \mathcal{G}_{\Sigma} = \simeq^{\alpha\prime} \tag{3.24}$$

Antagelsene `KONVERG` og `KONSTR` gjør ting svært mye lettere for oss. Vi kan også vise

Teorem 3.31b *Anta* `TsK` *og* `KREP1`*. Anta i tillegg* *h*`ROT`*, samt* E_s`VARBEVAR`*,* E^d`VARBEVAR` *og* `DISJ`*. Anta også* `KONVERG` *og* `KONSTR`*. Da har vi*

$$\overset{\Sigma * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}} \mathcal{G}_{\Sigma} = \simeq^{\alpha\prime}$$

Når det er sagt, viser vi likevel kun teorem 3.31. Hvordan teorem 3.31b vises, vil fremgå svært tydelig.

Teorem 3.31 følger fra følgende tre lemmata:

Lemma 3.32 *Anta* `T`$\hat{\Sigma}$`K` *og* `KREP1`*. Da har vi*

$$\overset{\Sigma * \hat{s}}{\underset{\dot{E}\,i\,d}{\rightsquigarrow}} \mathcal{G}_{\hat{\Sigma}} \subseteq (\simeq^s \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\hat{\Sigma}}$$

Bevis: Lemma 3.32 følger fra sats 3.30.
□

Lemma 3.33 *Anta* `KONVERG`, `KONSTR` *og* `DISJ`. *Da har vi*

$$(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}_{\mathcal{G}_\Sigma} \subseteq (\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{*}$$

Bevis: Betrakt en vilkårlig $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{*}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$ for $g, g' \in \mathcal{G}_\Sigma$. En slik utledning består av

1) enkeltsteg $\langle \ldots, g_i, g_{i+1}, \ldots \rangle$ slik at

$$g_i = c[g_i'] \;\text{ og }\; c[g_{i+1}'] = g_{i+1} \;\text{ og }\; g_i' \simeq^s g_{i+1}'$$

for vilkårlige (muligens tomme) kontekster $c \in \mathcal{G}_\Sigma$; samt

2) delsekvenser $\langle g_i, \ldots, g_{i+k} \rangle$ slik at

$$g_i \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_{i+k}$$

Enkeltstegene under 1) er umiddelbart også $(\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{*}$-steg.

Vi konsentrerer oss om 2): Betrakt en vilkårlig delsekvens $\langle g_i, \ldots, g_{i+k} \rangle$ slik at $g_i \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_{i+k}$ og slik at eventuelle g_{i-1}, g_{i+k+1} er slik at enkeltstegene $\langle \ldots, g_{i-1}, g_i, \ldots \rangle$ og $\langle \ldots, g_{i+k}, g_{i+k+1}, \ldots \rangle$ ikke er $\underset{\hat{E}}{\leftrightarrow}$-steg. Merk at g_i og g_{i+k} da må være i $\mathcal{G}_\Sigma$, ved den Σ-monotone tillukning.

Ialt følger derfor lemmaet nå ved å vise at

$$\overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_\Sigma \subseteq \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma$$

Anta så at $g \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g'$ for vilkårlige $g, g' \in \mathcal{G}_\Sigma$. Ved `KONVERG` har vi $g \overset{!}{\underset{\hat{E}}{\rightarrow}} g! \overset{!}{\underset{\hat{E}}{\leftarrow}} g'$. Vi viser under `DISJ` og `KONSTR` at enhver ensrettet $\hat{E}$-utledning $\langle g, \ldots, g! \rangle$ ($\langle g', \ldots, g! \rangle$) må være en (ensrettet) E^d-utledning i $\mathcal{G}_\Sigma$: Induksjon på lengden n av $\langle g, \ldots, g! \rangle$.

$n = 1$: Trivielt, siden $g \in \mathcal{G}_\Sigma$.

$n = k+1; k \geq 1$: Induksjonshypotesen gir $g \overset{*}{\underset{E^d}{\leftrightarrow}} g_k$ for en $g_k \in \mathcal{G}_\Sigma$. Ved `KONSTR` er hver venstreside av ligninger i E_s på formen $f(t_1, \ldots, t_n)$, der f er et definert symbol i $\hat{\Sigma}$ og alle $t_i \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$; $1 \leq i \leq n$. Ved `DISJ` må $f \in \Sigma^h$, eller f er det spesifiserende symbol s. Siden $g_k \in \mathcal{G}_\Sigma$, fins da ingen ligning $v = h \in E_s$ slik at $g_k|p = v\sigma$ for noen posisjon p eller substitusjon σ. Følgelig må $g_k \overset{*}{\underset{E^d}{\leftrightarrow}} g!$; og ved `DISJ` og `KONSTR` må $g! \in \mathcal{G}_\Sigma$.

□

Nå er pr. definisjon

$$(\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{*} = \simeq^{\alpha\prime}$$

så det gjenstår bare å vise:

Lemma 3.34

$$\simeq^{\alpha\prime} \subseteq \overset{\Sigma^* s}{\underset{\hat{E} \cup d}{\leftrightsquigarrow}} \mathcal{G}_\Sigma$$

Bevis: Anta $g \simeq^{\alpha\prime} g'$ for vilkårlige $g, g' \in \mathcal{G}_\Sigma$. Vi induserer over lengden n av en vilkårlig $\simeq^{\alpha\prime}$-utledning $\langle g, \ldots, g' \rangle$ i $\mathcal{G}_\Sigma$. Basis er triviell. For induksjonssteget gir induksjonshypotesen $g \overset{\Sigma^* s}{\underset{\hat{E} \cup d}{\leftrightsquigarrow}} g_k$ for en $g_k \in \mathcal{G}_\Sigma$. Anta

$$g_k \overset{}{\underset{E^d}{\leftrightarrow}} g'$$

Da er trivielt $g_k \overset{\Sigma^* s}{\underset{\hat{E} \cup d}{\leftrightsquigarrow}} g'$, og induksjonssteget følger. Anta så at

$$g_k = c[g_k'] \;\text{ og }\; c[g''] = g' \;\text{ og }\; g_k' \simeq^s g''$$

for en vilkårlig (muligens tom) kontekst $c \in \mathcal{G}_\Sigma$. Vi har da, siden $g_k' \simeq^s g''$ $\Leftrightarrow$ $s(g_k') \overset{*}{\underset{E_s}{\leftrightarrow}} s(g'')$

$$g_k = c[g_k'] \underset{\{s(x)=x\}}{\longleftrightarrow} c[s(g_k')] \overset{*}{\underset{E_s}{\leftrightarrow}} c[s(g'')] \underset{\{s(x)=x\}}{\longleftrightarrow} c[g''] = g'$$

Konteksten c er i $\mathcal{G}_\Sigma$, så vi har $g_k \overset{\Sigma^{*\mathring{s}}}{\underset{E^{id}}{\rightsquigarrow}} g'$, og induksjonssteget følger.
□

Lemmata 3.32, 3.33 og 3.34 gir da ialt teorem 3.31. Teorem 3.31 sier altså at $\overset{\Sigma^{*\mathring{s}}}{\underset{E^{id}}{\rightsquigarrow}}\mathcal{G}_\Sigma$ er identisk med initialsemantikken $\simeq^{\alpha'}$ relativ til $\simeq^s$ spesifisert av E^d og Σ. Inkonsistens innført ved symboler i Σ^h er utelukket, *siden* Σ^h *ikke finnes i* $\simeq^{\alpha'}$ *sin logiske omgivelse.*

Nå er ikke $\overset{\Sigma^{*\mathring{s}}}{\underset{E^{id}}{\rightsquigarrow}}\mathcal{G}_\Sigma$ noen basis-initialsemantikk i vanlig forstand. Men siden $\overset{\Sigma^{*\mathring{s}}}{\underset{E^{id}}{\rightsquigarrow}}\mathcal{G}_\Sigma$ er en *operasjonell* restriksjon av basis-initialsemantikken $\overset{*}{\underset{E^{id}}{\leftrightarrow}}$, er det likevel tenkelig at resolusjonsmetoder for basis-initialsemantikk kan modifiseres til å være anvendbare på slike *id*-restriksjoner. I såfall kan resolusjonsmetoder for basis-initialsemantikk brukes for (generell) initialsemantikk med indirekte kjerne, der semantikkens formelle omgivelse er den som er i tråd med den abstrakte datatypen som skal implementeres, og *ikke* en utvidet formell omgivelse (en utvidelse som altså i en viss forstand kan innføre kunstig inkonsistens).

Merk at vår operasjonelle skjuling av hjelpefunksjonssymboler i formelle datatyper, kan sies å svare til skjuling av interne hjelpeprosedyrer/funksjoner i moduldeklarasjoner på programmeringsspråk-nivå.

*

Denne idéen om operasjonell skjuling av hjelpefunksjon(symbol)er synes jeg kunne være interessant for algebraisk spesifikasjon generelt; ikke bare for vårt tilfelle her med *id*-utvidelser. Det kan av og til være fordelaktig (og kanskje nødvendig) å spesifisere funksjoner ved hjelp av hjelpefunksjoner; våre semantikkgivende syntaktiske funksjoner er jo et godt eksempel på dette. I programmering er hjelpefunksjoner en vesentlig del av fornuftige (modulære) utviklingsstrategier. Slike hjelpefunksjoner mener vi hører til på det implementatoriske syntaktiske nivå og ikke på det semantiske nivå (jf. figur 1.2 side 6). Ved formell resonnering om abstrakte datatyper bør i dette lys interne hjelpefunksjoner kun sees som implementatoriske hjelpemidler og på en eller annen måte skjules fra logikken.

Følgende er et eksempel på hjelpefunksjoner brukt i definisjonen av en (vanlig) definert funksjon.

Eksempel 76 Vi skal lage en algebraisk funksjonsdefinisjon av *absoluttverdi*-funksjonen *abs* på de hele tall, der

$$abs(x) = \begin{cases} x & \text{for } x \geq 0 \\ \Leftrightarrow x & \text{for } x < 0 \end{cases}$$

Vi skal ta en *operasjonell* tilnærming: Vi skal bevege oss ned et abstraksjonsnivå til det semantiske nivå der syntaktiske funksjoner befinner seg. (Det nivået som er mellom den syntaktiske verden og det «ekte» semantiske nivå. Jf. diskusjonen rundt eksempel 42 side 74.) Vi skal spesifisere en syntaktisk funksjon $synt_{abs}$ som tar en term g i $\mathcal{G}_{\mathsf{Int}}$ og gir en term som representerer (iflg. grunnterm-tolken $\phi^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$ absolutt-verdien til verdien som g representerer. Vi definerer $synt_{abs}$ vha. syntaktiske hjelpefunksjoner; nærmere bestemt kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32 side 64, og funksjonen *pos* som gitt en kanonisk representant bestemt av $\delta_{\mathcal{I}nt}$ skifter alle pred'er til succ'er. Altså:

$$synt_{abs}(g) = pos(\delta_{\mathcal{I}nt}(g))$$

La altså

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0} : \mathsf{int}, \\ \mathsf{succ} : \mathsf{int} \rightarrow \mathsf{int}, \\ \mathsf{pred} : \mathsf{int} \rightarrow \mathsf{int} \end{array} \right\}$$

$$\mathsf{Abs} = \left\{ \begin{array}{l} \mathsf{abs : int \longrightarrow int,} \\ \mathsf{pos : int \longrightarrow int} \end{array} \right\} \cup \mathbf{\Delta}_{\mathsf{Int}} = \left\{ \begin{array}{l} \mathsf{delta : int \longrightarrow int,} \\ \mathsf{:} \end{array} \right\}$$

$$E_{abs} = \left\{ \begin{array}{l} \mathsf{abs(x) = pos(delta(x)),} \\ \\ \mathsf{pos(0) = 0,} \\ \mathsf{pos(succ(x)) = succ(x),} \\ \mathsf{pos(pred(x)) = succ(pos(x))} \end{array} \right\} \cup E_{\delta}$$

der $\mathbf{\Delta}_{\mathsf{Int}}$ og E_{δ} er en signatur og en spesifikasjon for og av $\delta_{\mathcal{I}nt}$.

Selvfølgelig er E_{abs} en algebraisk spesifiksjon av abs hvis E_{abs} er en algebraisk spesifiksjon av $synt_{abs}$. (Resonneringen i forbindelse med konstruksjon av algebraiske spesifikasjoner er ofte fokusert på syntaktisk manipulering. Vi gjør dette her helt eksplisitt ved å snakke via spesifikasjoner av syntaktiske funksjoner.) Poenget er nå at vi har

$$\mathsf{succ(succ(0))}_{E^{\overleftrightarrow{*}}_{abs}} \mathsf{pos(pred(succ(0)))} \simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}} \mathsf{pos(0)}_{E^{\overleftrightarrow{*}}_{abs}} \mathsf{0}$$

som gir (kunstig) initiell inkonsistens relativt til $\simeq^{\mathcal{I}nt}_{\mathcal{G}_{\mathsf{Int}}}$. Det er naturlig å betrakte hjelpefunksjonene $\delta_{\mathcal{I}nt}$ og pos og deres spesifikasjoner som implementatoriske hjelpemidler, og ikke tilhørende den abstrakte datatype $\langle \mathbb{Z}, \{0, succ, pred, abs\}\rangle$ som søkes implementert. Disse hjelpefunksjoner er jo også best forstått og forklart som syntaktiske funksjoner, og faktisk er det jo slik at ihvertfall pos ikke kan tolkes til noen funksjon over $\mathbb{Z}$. Jf. avsnitt 2.3.8 (Inkonsistens og tolkninger) side 44. Likevel faller det naturlig — eller vi kunne ønske — å bruke pos og $\delta_{\mathcal{I}nt}$ til hjelp for å implementere abs.

○

Hjelpefunksjoner tilbyr implementatoriske fordeler, men kan samtidig innføre inkonsistens. Vi ønsker å dra nytte av de implementatoriske sidene, uten at hjelpefunskjoner spiller noen rolle i logikken. Dette ønsket søker vi å imøtekomme ved skjuling av hjelpefunksjoner.

Den operasjonelle skjuling for id-utvidelser over, kan på en lignende måte gjøres for algebraisk spesifikasjon generelt.

3.7.2 Operasjonell skjuling i resolusjonsmetoder

I dette avsnitt skal vi foreslå hvordan vår operasjonelle skjuling kan inlemmes i resolusjonsmetoder. Stringente bevis for påstandene som fremsettes er ikke gjennomført, og diskusjonen i dette avsnittet skal kun ansees som fornuftig spekulasjon og reising av åpne spørsmål.

For resolusjonsmetoder bygget på komplettering (f.eks. induktiv komplettering avsnitt 2.4.5 side 56) kan det se ut til at den operasjonelle skjuling beskrevet over så og si direkte kan overføres til en restriksjon på hvilke kritiske par som betraktes under komplettering. I forbindelse med id-utvidelser foreslår vi at skjuling kanskje kan implementeres ved følgende endring:

> For inferensregelen **Utled** fra figur 2.4 side 54 gjør vi følgende restriksjon (E_0 er initial-ligningsmengden) :
>
> **RI-Utled:**
>
> $$\frac{\langle E, R\rangle}{\langle E \cup \{s = t\}, R\rangle}$$
>
> for $\langle s,t\rangle = \langle v_i\mu[h_j\,\mu]_p, h_i\mu\rangle$ et ekte kritisk par i R fra $v_i\mu$ for to regler $v_i \rightarrow h_i$ og $v_j \rightarrow h_j$, der ikke både $v_i\mu[\Box]_p$ har forekomster av hjelpefunksjonssymboler og samtidig $v_j \rightarrow h_j \notin {}_{E_0 \setminus \{\overleftrightarrow{s(x)}=x\}}^{\overleftrightarrow{*}}$.

Ved et induktivt resonnement kan vi fastslå at et kritisk par ikke blir generert med mindre det er i relasjonen $\overset{\Sigma' * \hat{s}}{\underset{E_0}{\rightsquigarrow}}$, der Σ' er Σ uten hjelpefunksjonssymboler. Hvordan det kan avgjøres hvorvidt $v_j \rightarrow h_j \notin {}_{E_0 \setminus \{\hat{s}(x)=x\}}\overset{*}{\longleftrightarrow}$ skal vi se i kapittel 4. Vi antar at hjelpefunksjonssymboler kun finnes i forbindelse med den indirekte spesifikasjonen.

La oss kalle en Knuth&Bendix-prosess med denne restriksjonen på **Utled** for en *RI-restriktert* Knuth&Bendix-prosess.

Påstand 3.35 *La signaturer osv. være som i figur 3.7 side 85. La R_{RI} være resultatet av en vellykket RI-restriktert Knuth&Bendix-komplettering gitt en ligningsmengde $\hat{E}^{id}$ inneholdende en indirekte spesifikasjon med spesifiserende symbol s.*

Anta at reduksjonsordningen i prosessen er slik at enhver $u \in \mathcal{T}_{\hat{\Sigma}}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$. Da finnes en konvergent $R'_{RI} \subseteq R_{RI}$ slik at

$$R'_{RI} \subseteq \mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$$

og

$${}_{R'_{RI}}\overset{*}{\longleftrightarrow} = \overset{\Sigma * \hat{s}}{\underset{\hat{E}^{id}}{\rightsquigarrow}} \Sigma^c$$

Videre finnes en operasjonell restriksjon ${}_{R\overset{*}{\longleftrightarrow}_{RI}op}$ av relasjonen ${}_{R}\overset{*}{\longleftrightarrow}_{RI}$ som er terminerende og slik at

$${}_{R\overset{*}{\longleftrightarrow}_{RI}op} = \overset{\Sigma * \hat{s}}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$$

og

$$s \; {}_{R\overset{*}{\longleftrightarrow}_{RI}op} \; t \;\; \Leftrightarrow \;\; s!_{op} = t!_{op}$$

der $s!_{op}$ betegner en entydig normalform i den velfunderte ordningen ${}_{R\overset{*}{\longleftrightarrow}_{RI}op}$.

En skisseaktig begrunnelse for vår fremsetting av påstand 3.35, gis i avsnitt 4.2.2 i kapittel 4.

Påstand 3.35 hevder at en vellykket RI-restriktert komplettering gir det vi kan kalle et ***kjernemanifest*** for $\hat{E}^{id}$ ***i henhold til*** $\overset{\Sigma * \hat{s}}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$. Dersom påstand 3.35 er riktig kan nå et slikt kjernemanifest sammenholdes med diskusjonen forøvrig om kjernemanifest (synliggjøring av kjernebevaring, bevis av mangel på kjernebevaring og synliggjøring av kjernesemantikk); men nå relatert, ved teorem 3.31 side 128, til initialsemantikken $\simeq^{\alpha'}$ relativ til $\simeq^s$ spesifisert av E^d og Σ (og ikke til initialsemantikken $\simeq^{\alpha}$ relativ til $\simeq^s$ spesifisert av $\hat{E}$ og $\hat{\Sigma}$.) Kunstig inkonsistens grunnet symboler i Σ^h er således holdt skjult i genereringen av kjernemanifestet.

Eksempel 77 Vi har ingen implementasjon av RI-restriktert komplettering, men for delta-*id*-utvidelsen av $E_{\delta_{\mathcal{I}nt}}$ fra eksempel 44 side 77, ville RI-restriktert komplettering muligens gi følgende kjernemanifest i henhold til $\overset{\mathsf{Int} * \mathsf{delta}}{\underset{E^{id}_{\delta_{\mathcal{I}nt}}}{\rightsquigarrow}}$:

$$R^{man}_{\delta_{\mathcal{I}nt}\,RI} = \left\{ \begin{array}{l} \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

○

Eksempel 78 RI-restriktert komplettering av delta-*id*-utvidelsen av E_n fra eksempel 65 side 116 kunne gi kjernemanifestet

$$R^{man}_{nRI} = \left\{ \begin{array}{l} \mathsf{pred(succ(0))} \rightarrow \mathsf{0}, \\ \vdots \\ \mathsf{pred(succ}^n\mathsf{(0))} \rightarrow \mathsf{succ}^{n-1}\mathsf{(0)}, \\ \mathsf{succ}^{n+1}\mathsf{(0)} \rightarrow \mathsf{0}, \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

i henhold til $\overset{Int*delta}{\underset{E^{id}_n}{\leftrightsquigarrow}}$.
○

Like, eller kanskje mer interessant, er at påstand 3.35 hevder en avgjørbar relasjon $_R\overset{*}{\underset{RI}{\longleftrightarrow}}_{op}$ som kan brukes som grunnlag i f.eks. RI-restriktert *induktiv* komplettering.

Når det gjelder direkte spesifikasjon, antar vi hjelpefunksjoner i forbindelse med definerte funksjoner, og ikke i forbindelse med spesifisering av generatorsemantikk. La oss anse initial-ligningsmengden E_0 delt disjunkt i generatorligninger E_0^c og andre ligninger E_0^d. Det kan se ut til at skjuling kan implementeres i kompletteringsbaserte metoder ved følgende endring:

> For inferensregelen **Utled** gjør vi følgende restriksjon:
>
> **RD-Utled:**
>
> $$\frac{\langle E, R\rangle}{\langle E \cup \{s = t\}, R\rangle}$$
>
> for $\langle s,t\rangle = \langle v_i\mu[h_j\,\mu]_p, h_i\mu\rangle$ et ekte kritisk par i R fra $v_i\mu$ for to regler $v_i \rightarrow h_i$ og $v_j \rightarrow h_j$, der ikke både $v_i\mu[\Box]_p$ har forekomster av hjelpefunksjonssymboler og samtidig $v_j \rightarrow h_j \notin {}_{E_0}\overset{*}{\underset{E_0^c}{\longleftrightarrow}}$.

Ved et induktivt resonnement kan vi fastslå at et kritisk par ikke blir generert med mindre det er i relasjonen $(\overset{*}{\underset{E_0^c}{\longleftrightarrow}} \mathcal{T}_{\Sigma^c}(\mathcal{V}) \cup \overset{*}{\underset{E_0^d}{\longleftrightarrow}} \mathcal{T}_{\Sigma}(\mathcal{V}))\restriction^*_{\Sigma'}$, der Σ^c er generatorene i Σ og Σ' er Σ uten hjelpefunksjonssymboler.

Vi skal ikke gå videre inn på *RD-restriktert* komplettering her, men lar påstand 3.35 over antyde en lignende kurs for kjernemanifest og RD-restriktert *induktiv* komplettering.

Videre bør resonnementene omkring RI-restriktert komplettering kunne utvides til å omfatte hjelpefunksjonssymboler også utenom de involvert i den indirekte spesifikasjonen.

Ved å inlemme skjuling av hjelpefunksjoner i resolusjonsmetoder, får man en ny frihet i algebraisk spesifikasjon, idet hjelpefunksjoner som innfører kunstig inkonsistens likevel kan benyttes i spesifisering og i formell mekanisk resonnering om programmer. Dette er et tema som synes meget interessant. Vi må som sagt heller ta tid til å utrede dette tema mer helhetlig en annen gang.

I de følgende avsnitt skal vi se på andre måter å skjule kunstig inkonsistens på. Det ser ut som disse først får praktisk interesse når resolusjonsmetoder for *generell* initial- og finalsemantikk forefinnes.

3.7.3 Skjuling på spesifikasjonsnivå

I sats 2.8 side 37 etablerte vi at fullstendig ukompletthet garanterer kjernebevaring/konsistens. Vi skal her la oss inspirere av dette til å avvæpne hjelpefunskjonssymboler med hensyn til inkonsistens, ved å la vår $\hat{E}$ være fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$.

La oss først se at det gir mening for oss å snakke om indirekte spesifikasjoner som er fullstendig $\{s\} \cup \Sigma^h$-ukomplette mhp. $\mathcal{G}_{\Sigma^c}$:

Eksempel 79 Betrakt den indirekte spesifikasjon E_n fra eksempel 65 på side 116.

Ved å fjerne ligning 9 får vi en ekvivalent indirekte spesifikasjon E_n^{fu} som er fullstendig $\{\mathsf{delta}\} \cup \{\mathsf{mem}\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$. At denne amputerte spesifikasjon er ekvivalent med den opprinnelige, sees ved å observere at en vilkårlig $g_c \in \mathcal{G}_{\mathsf{Int}}$ er en E_n-normalform hvis og bare hvis $\mathsf{mem}(0,0,g_c)$ er en E_n^{fu}-normalform. (Både E_n og E_n^{fu} er konvergente.)
○

Eksempel 80 Betrakt den indirekte spesifikasjon E_b fra eksempel 66 på side 117.

Ved å fjerne ligning 8 får vi en ekvivalent indirekte spesifikasjon E_b^{fu} som er fullstendig $\{\mathsf{delta}\} \cup \{\mathsf{mem}\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$. En vilkårlig $g_c \in \mathcal{G}_{\mathsf{Int}}$ er en E_b-normalform hvis og bare hvis $\mathsf{mem}(0,0,0,g_c)$ er en E_b^{fu}-normalform.
○

Følgende sats viser at inkonsistens innført ved symboler i Σ^h under visse omstendigheter kan forhindres, ved å la den indirekte spesifikasjon E_s være fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$:

Sats 3.36 *Anta* `DISJ`, E_s`KONSERV+` *og* `TΣK`. *Anta E_s er fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$. La $\simeq^{\alpha'}$ være initialsemantikken relativ til $\simeq^s$ spesifisert av E^d og Σ. La på den annen side $\simeq^{\alpha}$ være initialsemantikken relativ til $\simeq^s$ spesifisert av $\hat{E}$ og $\hat{\Sigma}$. Da har vi*

$$\simeq^{\alpha}{}_{\mathcal{G}_\Sigma} = \simeq^{\alpha'}$$

M.a.o.:

$$(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\hat{\Sigma}}\rightarrow}_{\mathcal{G}_\Sigma} = (\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{\overset{*}{\Sigma}\rightarrow}$$

Bevis: At

$$(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\hat{\Sigma}}\rightarrow}_{\mathcal{G}_\Sigma} \supseteq (\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{\overset{*}{\Sigma}\rightarrow}$$

er innlysende.

For den motsatte inklusjon betrakt først en vilkårlig $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\hat{\Sigma}}\rightarrow}$-utledning $\langle g_c, \ldots, g_c' \rangle$ i $\mathcal{G}_{\hat{\Sigma}}$ for generatortermer $g_c, g_c' \in \mathcal{G}_{\Sigma^c}$.

Siden E_s er fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$, og vi antar `DISJ`, er $\hat{E}$ også fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$. Men da må $\langle g_c, \ldots, g_c' \rangle$ være en utledning i $\mathcal{G}_\Sigma$. Ved E_s`KONSERV+` og igjen `DISJ`, må i tillegg ethvert $\hat{E}$-steg i $\langle g_c, \ldots, g_c' \rangle$ være et E^d-steg. Følgelig har vi

$$g_c \ (\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{\overset{*}{\Sigma}\rightarrow} \ g_c'$$

Anta nå

$$g \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\hat{\Sigma}}\rightarrow} \ g'$$

for vilkårlige $g, g' \in \mathcal{G}_\Sigma$.

Ved `TΣK` har vi jo $g \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_1$ og $g' \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_2$ for noen $g_1, g_2 \in \mathcal{G}_{\Sigma^c}$. Da har vi

$$g_1 \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\hat{\Sigma}}\rightarrow} \ g' \overset{*}{\underset{\hat{E}}{\leftrightarrow}} g_2$$

og følgelig $g_1 \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\overset{*}{\hat{\Sigma}}\rightarrow} \ g_2$, og da altså ved argumentet over $g_1 \ (\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{\overset{*}{\Sigma}\rightarrow} \ g_2$.

Ved `DISJ` og E_s`KONSERV+` har vi endog $g \overset{*}{\underset{E^d}{\leftrightarrow}} g_1$ og $g' \overset{*}{\underset{E^d}{\leftrightarrow}} g_2$. Dette gir da på den annen side

$$g \overset{*}{\underset{E^d}{\leftrightarrow}} g_1 \ (\simeq^s \cup \overset{*}{\underset{E^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^{\overset{*}{\Sigma}\rightarrow} \ g_2 \overset{*}{\underset{E^d}{\leftrightarrow}} g'$$

og dermed

$$g\ (\simeq^s \cup_{E^d}^{\overleftrightarrow{*}} \mathcal{G}_\Sigma)^{\overleftrightarrow{*}_\Sigma}\ g'$$

og satsen følger.
□

Vi kan altså under rimelige antagelser skjule hjelpefunksjonssymboler med hensyn til inkonsistens, ved å la den indirekte spesifikasjon E_s være fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$.

Dette er vel og bra, men i neste avsnitt skal vi se at antagelsen om fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$ kan ødelegge korrespondansen mellom basis-initialsemantikk og *id*-utvidelser av indirekte spesifikasjoner.

3.7.4 Skjuling på spesifikasjonsnivå og reduksjon til basis-initialsemantikk

La oss igjen betrakte (3.22) på side 107. Vi skal som vi lovte på side 107, nå vise et par eksempler der

- $\Sigma^h \neq \emptyset$
- $\hat{E}$ er fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$

og der (3.22) ikke holder.

Eksempel 79 (forts.) Ved å fjerne ligning 9 fra den indirekte spesifikasjon E_n fra eksempel 65 på side 116 fikk vi en ekvivalent indirekte spesifikasjon E_n^{fu} som er fullstendig $\{\mathsf{delta}\} \cup \{\mathsf{mem}\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$.

La $\simeq^\alpha$ være initialsemantikken spesifisert av E_n^{fu} og $\mathcal{G}_{\mathsf{Int} \cup \{\mathsf{mem}\}}$, relativ til semantikken $\simeq^{\mathsf{delta}}$ på $\mathcal{G}_{\mathsf{Int}}$ spesifisert indirekte av E_n^{fu}. M.a.o.:

$$\simeq^\alpha = (\simeq^{\mathsf{delta}} \cup_{E_n^{fu}}^{\overleftrightarrow{*}} \mathcal{G}_{\mathsf{Int} \cup \{\mathsf{mem}\}})^{\overleftrightarrow{*}_{\mathsf{Int} \cup \{\mathsf{mem}\}}}$$

Betingelsene for sats 3.36 side 134 er oppfylt (eller mer passende i dette tilfellet: sats 2.8 side 37). Vi har altså initiell konsistens, m.a.o.:

$$\simeq^\alpha \mathcal{G}_{\mathsf{Int}} = \simeq^{\mathsf{delta}}$$

Imidlertid er delta-*id*-utvidelsen $E_n^{fu\ id}$ av E_n^{fu} *ikke kjernebevarende*. En terminerende vellykket Knuth&Bendix-prosess gitt delta-*id*-utvidelsen av E_n^{fu} for $n = 2$, gir Int-manifestet

$$R_2^{fu\ man} = \left\{ \begin{array}{l} \mathsf{succ}(\mathsf{succ}(\mathsf{succ}(\mathsf{x}))) \rightarrow \mathsf{x}, \\ \mathsf{pred}(\mathsf{x}) \rightarrow \mathsf{succ}(\mathsf{succ}(\mathsf{x})) \end{array} \right\}$$

Vi kan analogt som i eksempel 65 på side 116, vise mangel på kjernebevaring ved

$$\mathsf{delta}(\mathsf{pred}(0))!E_2^{fu} \neq \mathsf{delta}(\mathsf{succ}(\mathsf{succ}(0)))!E_2^{fu}$$

Altså har vi

$$\simeq^\alpha \mathcal{G}_{\mathsf{Int}} \neq \overleftrightarrow{E_n^{fu\ id}}^{*} \mathcal{G}_{\mathsf{Int}}$$

og (3.22) på side 107 holder ikke.
○

Eksempel 80 (forts.) Betrakt den indirekte spesifikasjon E_b fra eksempel 66 på side 117.

Ved å fjerne ligning 8 fra den indirekte spesifikasjon E_b fra eksempel 66 på side 117, får vi en ekvivalent indirekte spesifikasjon E_b^{fu} som er fullstendig $\{\mathsf{delta}\} \cup \{\mathsf{mem}\}$-ukomplett mhp. $\mathcal{G}_{\mathsf{Int}}$.

Vår komplettering av delta-id-utvidelsen av E_b^{fu} terminerer ikke (på en analog måte som i eksempel 66 side 117), men følgende regler blir blant andre, generert:

$$R_b = \left\{ \begin{array}{l} \vdots \\ \mathsf{delta(x)} \rightarrow \mathsf{x}, \\ \vdots \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \\ \vdots \end{array} \right\}$$

Det følger fra lemma 2.28 side 55 at enhver regel generert under en Knuth&-Bendix-komplettering vil være en logisk konsekvens av initialligningene. Følgelig kan vi analogt som i eksempel 66 side 117, vise mangel på kjernebevaring ved

$$\mathsf{delta(pred(succ(0)))}!E_b^{fu} \neq \mathsf{delta(0)}!E_b^{fu}$$

På den annen side gir sats 2.8 side 37 for initialsematikken

$$\simeq^{\alpha} = (\simeq^{\mathsf{delta}} \cup_{E_b^{\overset{*}{\leftarrow}fu}} \mathcal{G}_{\mathsf{Int} \cup \{\mathsf{mem}\}})^{\dashv \mathsf{Int} \cup \overset{*}{\{\mathsf{mem}\}} \vdash}$$

der $\simeq^{\mathsf{delta}}$ på $\mathcal{G}_{\mathsf{Int}}$ er spesifisert indirekte av E_b^{fu}, at

$$\simeq^{\alpha} \mathcal{G}_{\mathsf{Int}} = \simeq^{\mathcal{G}_{\mathsf{delta}}}$$

Igjen har vi da at

$$\simeq^{\alpha} \mathcal{G}_{\mathsf{Int}} \neq_{E_b^{\overset{*}{\leftarrow}fu\ id}} \mathcal{G}_{\mathsf{Int}}$$

og (3.22) på side 107 holder ikke.
○

Vi ser her i eksempler 79 og 80 at id-utvidelsene av de fullstendig $\{s\} \cup \Sigma^h$-ukomplette versjonene av de indirekte spesifikasjonene oppfører seg, i det minste på generatorene, identisk med id-utvidelsene av de originale indirekte spesifikasjonene. Det har for id-utvidelser ingen virkning å innføre fullstendig $\{s\} \cup \Sigma^h$-ukompletthet. Forklaringen på dette er, løst forklart, at ligningen $s(x) = x$ (gjen-)innfører tilstrekkelig $\{s\} \cup \Sigma^h$-kompletthet mhp. $\mathcal{G}_{\Sigma^c}$.

Hjelpefunksjonssymboler kan altså skjules med hensyn til inkonsistens, ved å la den indirekte spesifikasjon E_s være fullstendig $\{s\} \cup \Sigma^h$-ukomplett mhp. $\mathcal{G}_{\Sigma^c}$. Men skjuling på denne måten mister generelt sin effekt ved overføring til id-utvidelsen. Således har vi altså at denne fullstendige $\{s\} \cup \Sigma^h$-kompletthet kan ødelegge korrespondansen mellom basis-initialsemantikk og id-utvidelser av indirekte spesifikasjoner.

Dersom skjuling av hjelpefunksjoner på spesifikasjonsnivå gjøres som her, kan altså ikke id-utvidelser generelt brukes som en tilnærming av indirekte spesifikasjon til basis-initialsemantikk og tilhørende resolusjonsmetoder.

3.7.5 Skjuling ved innføring av typer

La oss vende tilbake et øyeblikk til kommentaren på side 89 angående måter å se en mengde av klasserepresentanter på. Vi har hittil diskutert i henhold til siste punkt i denne kommentaren. Vi skal her se at kunstig inkonsistens kan elimineres ved å et syn i henhold til første punkt i kommentaren. Vi nøyer oss med å gi et eksempel.

Eksempel 81 La

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0 : int}, \\ \mathsf{succ : int \rightarrow int}, \\ \mathsf{pred : int \rightarrow int} \end{array} \right\}$$

$$\mathbf{\Delta}_{\mathsf{Int}} = \left\{ \begin{array}{l} \mathsf{delta : int \rightarrow int}, \\ \#_{\mathsf{s}} \mathsf{ : int \rightarrow int}, \\ \#_{\mathsf{p}} \mathsf{ : int \rightarrow int}, \\ \mathsf{- : int \times int \rightarrow int}, \\ \mathsf{+ : int \times int \rightarrow int} \end{array} \right\}$$

Vi husker fra eksempel 56 side 108 at $E_{\delta_{\mathcal{I}nt}}$ er initielt inkonsistent og ikke finalt kjernebevarende relativt til $\simeq^{\mathsf{delta}}$. Vi innfører nå imidlertid en ny type $\underline{\mathsf{int}}$ og modifiserer signaturene Int og $\mathbf{\Delta}_{\mathsf{Int}}$ slik:

$$\mathsf{Int}' = \left\{ \begin{array}{l} \mathsf{0 : int}, \\ \mathsf{succ : int \rightarrow int}, \\ \mathsf{pred : int \rightarrow int}, \\ \underline{\mathsf{0}} \mathsf{ : } \underline{\mathsf{int}}, \\ \underline{\mathsf{succ}} \mathsf{ : } \underline{\mathsf{int}} \rightarrow \underline{\mathsf{int}}, \\ \underline{\mathsf{pred}} \mathsf{ : } \underline{\mathsf{int}} \rightarrow \underline{\mathsf{int}} \end{array} \right\}$$

$$\mathbf{\Delta}'_{\mathsf{Int}} = \left\{ \begin{array}{l} \mathsf{delta : int \rightarrow } \underline{\mathsf{int}}, \\ \#_{\mathsf{s}} \mathsf{ : int \rightarrow } \underline{\mathsf{int}}, \\ \#_{\mathsf{p}} \mathsf{ : int \rightarrow } \underline{\mathsf{int}}, \\ \mathsf{- : } \underline{\mathsf{int}} \times \underline{\mathsf{int}} \rightarrow \underline{\mathsf{int}}, \\ \mathsf{+ : } \underline{\mathsf{int}} \times \underline{\mathsf{int}} \rightarrow \underline{\mathsf{int}} \end{array} \right\}$$

I henhold til denne typingen modifiseres $E_{\delta_{\mathcal{I}nt}}$ til $E'_{\delta_{\mathcal{I}nt}}$. Merk at semantikken $\simeq^{\mathsf{delta}}$ på $\mathcal{G}_{\mathsf{Int}}$ er uendret. Typene Int og $\underline{\mathsf{Int}}$ antas *disjunkte*; dvs. de står ikke i noe subtype-forhold til hverandre.

Vi har nå

$$\#_{\mathsf{s}}(\mathsf{succ}(\mathsf{pred}(0)))_{E \overset{*}{\underset{\delta_{\mathcal{I}nt}}{\mapsto}}} \underline{\mathsf{succ}(0)} \quad \text{og} \quad \#_{\mathsf{s}}(0)_{E \overset{*}{\underset{\delta_{\mathcal{I}nt}}{\mapsto}}} \underline{0}$$

Relativt til $\simeq^{\mathsf{delta}}$ gir dette initialsemantisk $\underline{\mathsf{succ}(0)} \simeq^{\alpha} \underline{0}$, men det gjør ikke noe: Vi er interessert i semantikken gitt på $\mathcal{G}_{\mathsf{Int}}$. Det vesentlige er at vi nå har

$$\mathsf{succ}(0) \not\simeq^{\alpha} 0$$

La nå $\delta'_{\mathcal{I}nt}$ være som $\delta_{\mathcal{I}nt}$, men med kodomene $\mathcal{G}_{\underline{\mathsf{Int}}}$. For finalsemantiske betraktninger må vi nå inkludere en semantikk på $\mathcal{G}_{\underline{\mathsf{Int}}}/\delta'_{\mathcal{I}nt}$ i kjernen $\simeq^{\mathsf{delta}}$. Lar vi denne semantikken på $\mathcal{G}_{\underline{\mathsf{Int}}}$ være den frie (som umiddelbart er naturlig), får vi stadig finalsemantisk $\mathsf{succ}(\mathsf{pred}(0)) \not\simeq^{\omega} 0$ og mangel på kjernebevaring. Vi kan imidlertid spesifisere universell semantikk på $\mathcal{G}_{\underline{\mathsf{Int}}}/\delta'_{\mathcal{I}nt}$ (f.eks. ved ligningen $\mathsf{x} = \mathsf{y}$). Merk at dette ikke ødelegger (delen av) kjernesematikken på $\mathcal{G}_{\mathsf{Int}}$. Da får vi (for eksempel)

$$\mathsf{succ}(\mathsf{pred}(0)) \simeq^{\omega} 0$$

○

Ved innføring av en «skygge-»type som i dette eksemplet, kan hjelpefunksjonssymboler avvæpnes mht. inkonsistens.

Kan vi si at slik typing i en viss forstand implementerer den operasjonelle restriksjonen på omskrivning over *id*-utvidelser fra avsnitt 3.7.1? Vel, det kommer an på hvordan man ser det, men ett er sikkert: Ved tradisjonelle typingsregler er nå ikke *id*-ligninger (for eksemplet over: $\mathsf{delta}(\mathsf{x}) = \mathsf{x}$) lenger lovlig.

Sambandet til *id*-utvidelser er derfor ikke tilstede ved innføring av en slik skyggetype. Det er ikke umiddelbart mulig å redde situasjonen ved subtyping.

*

For initialsemantikk relativ til indirekte kjerner er foreløpig vår eneste kontakt til resolusjonsmetoder via *id*-utvidelser. I lys av dette er skjulingsmetodene beskrevet i dette avsnittet og i avsnitt 3.7.3 (Skjuling på spesifikasjonsnivå) foreløpig kun av teoretisk interesse. I en drøfting av resolusjonsmetoder for generell semantikk, vil disse skjulingsmetodene imidlertid være av praktisk interesse.

3.8 Alternativ til *id*-utvidelser under konsistens

Vi skal som et apropos til begrepet 'delvis monoton tillukning' se litt på relasjonen $\Re^s$ som definert i (3.9) på side 88.

Vi skal igjen adoptere den formelle omgivelse som definert i figur 3.7 på side 85.

Vi nevnte i avsnitt 3.7.1 (Operasjonell skjuling) at delvis monoton tillukning anvendt på $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$, på sett og vis gir forskjellige grader av samarbeid mellom $\simeq^s$ og $\overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}}$. Full monoton tillukning gir rett og slett initialsemantikk. Vi skal nå se på den andre enden av skalaen; nemlig ingen monoton tillukning.

På side 41 i avsnitt 2.3.7 introduserte vi begrepet 'separabel semantikk', og viste at relasjonen definert i (2.13) side 41 uttrykker slik semantikk.

Vi skal her vise at separabel semantikk også kan uttrykkes ved relasjonen $\Re^s$ som definert i (3.9).

Teorem 3.37 *Anta* $\mathtt{T\hat{\Sigma}K}$ *og* `KREP1`*. Vi har da*

$$s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g') \;\Leftrightarrow\; g\ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*\ g'$$

for alle $g, g' \in \mathcal{G}_{\hat{\Sigma}}$.

Bevis: Anta

$$s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g')$$

for vilkårlige $g, g' \in \mathcal{G}_{\hat{\Sigma}}$. Da har vi

$$g \underset{\{s(x)=x\}}{\longleftrightarrow} s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g') \underset{\{s(x)=x\}}{\longleftrightarrow} g'$$

og altså

$$g \overset{*}{\underset{\hat{E}}{\overset{s}{\underset{id}{\leftrightarrow}}}} g'$$

Ved sats 3.30 på side 127 har vi da

$$g\ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*\ g'$$

For den andre inklusjonen, betrakt en vilkårlig $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$-utledning $\langle g, \ldots, g' \rangle$ i $\hat{E}$. Vi induserer på lengden n av denne.

$n = 1$: Trivielt har vi $s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g)$.

$n = k + 1; k \geq 1$: Da har vi $\langle g, \ldots, g_k, g' \rangle$. Induksjonshypotesen gir

$$s(g) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g_k)$$

Anta $g_k \simeq^s g'$. Dette betyr $s(g_k) \overset{*}{\underset{\hat{E}_s}{\leftrightarrow}} s(g')$, som medfører $s(g_k) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} s(g')$, og induksjonssteget følger.

Anta $g_k \overset{}{\underset{\dot{E}}{\leftrightarrow}} g'$. Da har vi ved monotonitet mhp. kontekstapplikasjon umiddelbart $s(g_k) \overset{*}{\underset{\dot{E}}{\leftrightarrow}} s(g')$.
□

Teorem 3.37 gir altså at separabel semantikk kan uttrykkes ved $\Re^s$.

Relasjonen $\Re^s$ er, på linje med indirekte semantikk, en umiddelbar avledning av ren ligningslogikk, og gjør derfor semantikken $(\simeq^s \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\dot{\Sigma}})^*$ tilgjengelig for ligningslogiske metoder.

Merk at ved sats 3.30, er også $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}}$ et uttrykk for $(\simeq^s \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\dot{\Sigma}})^*$. Forøvrig er da ved teorem 3.37 og sats 3.30, $\Re^s$ identisk med $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}}$. Relasjonen $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}}$ er en operasjonell restriksjon av ligningslogikk: Ligningen $s(x) = x$ kan kun anvendes i posisjon ε.

I tråd med skjuling av hjelpefunksjoner, kan man videre vise under `KONVERG`, `KONSTR` og `DISJ`, at

$$(\simeq^s \cup \overset{*}{\underset{\dot{E}}{\leftrightarrow}} \mathcal{G}_{\dot{\Sigma}})^*_{\mathcal{G}_\Sigma} = (\simeq^s \cup \overset{*}{\underset{\dot{E}^d}{\leftrightarrow}} \mathcal{G}_\Sigma)^*$$

(på analog måte som vi viste lemma 3.33 på side 129). Vi har da at restriksjonen $\Re^s_{\mathcal{G}_\Sigma}$ av $\Re^s$ (og således også restriksjonen $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}} \mathcal{G}_\Sigma$ av $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}}$) uttrykker separabel semantikk *bestemt av* Σ *og* E^d.

Merk til slutt at under konsistens gir teoremene 2.20 side 42 og 2.21 side 43, at $\Re^s$ ($\Re^s_{\mathcal{G}_\Sigma}$) og $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}}$ ($\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}} \mathcal{G}_\Sigma$) gir initialsemantikk og degenerert finalsemantikk.

Under konsistens er altså $\Re^s$ — en umiddelbar avledning av ligningslogikk, og $\overset{*s}{\underset{\dot{E}}{\rightsquigarrow}}$ — en enkel restriksjon på ligningslogikk, mulige erstatninger for *id*-utvidelser i tilnærminger til eksisterende resolusjonsmetoder.

3.9 Verifikasjon av indirekte algebraiske spesifikasjoner

Å avgjøre om en indirekte spesifiksjon i seg selv gir ønsket semantikk, er interessant. Dette kanskje enda mer, nå som det er mulig å avvæpne den iboende mulighet for inkonsistens i en indirekte spesifikasjon.

Vi vender oss nå et øyeblikk mot oppgaven å verifisere om en indirekte algebraisk spesifikasjon gir oss den semantikk vi ønsker. Vi kommenterer kort følgende metoder:

1. Program-analyse.
2. Verifikasjon relativt til en direkte algebraisk spesifikasjon, herunder
   (a) sammenligning av indirekte- med direkte spesifikasjon
   (b) generering av direkte spesifikasjon fra indirekte spesifikasjon.

3.9.1 Program-analyse

Konstruktiv algebraisk funksjonsspesifikasjon fører ofte naturlig til at den algebraiske funksjonsspesifikasjonen er konvergent. Det er derfor mulig å se på konstruktive algebraiske funksjonsspesifikasjoner som abstrakte deterministiske programmer. *I prinsipp* kan da metoder for programverifikasjon tas ibruk for å verifisere algebraiske funksjonsspesifikasjoner.

Å eksplisitt ta ibruk programverifikasjonsmetoder er ikke vanligvis naturlig for algebraisk funksjonsspesifikasjon. Men vi har i eksempler antydet bruk av slike metoder for algebraisk spesifikasjon av klasserepresentant-funksjoner. Vi har snakket om invarianter, subprosedyrer osv.

Våre algebraiske spesifikasjoner av syntaktiske klasserepresentant-funksjoner har en mye mer operasjonell stil enn det som vanligvis er tilfellet for algebraisk spesifikasjon. Dette er ikke så rart: Våre syntaktiske funksjoner opererer på termer; på en lignende måte som abstrakte maskiner gjør. En delmengde av kanonisk-representant funksjoner er endog beregnbare/implementerbare av konvergente omskrivningssystem (se figur 3.1 side 67). Dette har gjenspeilt seg i våre algebraiske spesifikasjoner av syntaktiske funksjoner.

Således er det naturlig å ta ibruk programverifikasjonsmetoder i full kraft. Vi kan f.eks. benytte «Hoare-logikk» eller andre metoder. Vi har ikke gjort, og kommer ikke til å gjøre dette i detalj. Men vi har som sagt i eksempler antydet i grove trekk bruken av programverifikasjonsmetoder på spesifikasjoner av syntaktiske funksjoner.

Vi biter oss tilsynelatende i halen. Vi har havnet i programmeringsverdenen igjen, mens vi jo søkte å resonnere *om* programmer i en mer abstrakt formell verden. Begrepsmessig er dette ikke noe problem, og våre algebraiske beskrivelser av syntaktiske funksjoner kan fortsatt sies å være på et mer abstrakt nivå enn konkret programmering. Men det er likevel verdt å merke seg at indirekte spesifikasjon ved algebraiske beskrivelser av syntaktiske funksjoner kan være mindre abstrakt enn (vanlig) algebraisk spesifikasjon.

3.9.2 Verifikasjon relativt til en direkte algebraisk spesifikasjon

Her fungerer en direkte algebraisk spesifikasjon som en fasit for hvilken semantikk som ønskes.

Sammenligning med direkte spesifikasjon

Anta en indirekte spesifikasjon E_s med spesifiserende symbol s. Anta en direkte spesifikasjon E, som gir den semantikk vi ønsker at den indirekte spesifikasjon skal gi på — si — $\mathcal{G}_\Sigma$. Vi vil med andre ord at

$$s(g) \overset{*}{\underset{E_s}{\leftrightarrow}} s(g') \;\Leftrightarrow\; g \overset{*}{\underset{E}{\leftrightarrow}} g' \tag{3.25}$$

for alle $g, g' \in \mathcal{G}_\Sigma$. Kravet (3.25) representerer en betydelig bevisbyrde. Det er ikke utenkelig at programverifikasjonsmetoder som beskrevet i forrige avsnitt, er på sin plass i tilknytning til E_s. Det er heller ikke utenkelig at (semi)-mekaniske metoder kan appliseres for å verifisere (3.25).

Generering av direkte spesifikasjon

En annen mulighet er å generere en ekvivalent direkte spesifikasjon fra den indirekte spesifikasjonen. Den genererte direkte spesifikasjon kan så verifiseres; eller man kanskje vet at denne gir den ønskede semantikk.

I avsnitt 3.6.4 på side 122 så vi at Knuth&Bendix-komplettering av *id*-utvidelsen av en indirekte spesifikasjon kunne gi oss en slik ekvivalent direkte spesifikasjon.

Vi husker at dette forutsetter at vi kan etablere kjernebevaring for denne *id*-utvidelsen. Vi husker også fra diskusjonen på side 115 og eksempel 63 på side 115 at slik generering ved Knuth&Bendix-komplettering av en direkte spesifikasjon, forutsetter eksistensen av en direkte spesifikasjon hvor hver ligning er en logisk konsekvens av *id*-utvidelsen.

Vi kunne nå ønske å fri oss fra disse forutsetningene. Den siste forutsetningen er grunnet i måten Knuth&Bendix-prosesser fungerer på (generering av logiske konsekvenser ved å finne kritiske par). Og videre, siden det er mulig

å avvæpne den iboende mulighet for inkonsistens i indirekte spesifikasjoner, er nå en indirekte spesifikasjon interessant i spesifikasjonssammenheng, selv om dens *id*-utvidelse ikke er kjernebevarende; og dermed ikke oppfyller den første forutsetningen over. Nå hevder påstand 3.35 side 132 generering av direkte spesifikasjon i henhold til skjuling av hjelpefunksjonssymboler. Men selv om denne påstanden viser seg å holde, vil den siste forutsetningen over fortsatt være bindende.

Ialt er det derfor interessant å finne andre metoder å generere direkte spesifikasjoner fra indirekte spesifikasjoner på, enn ved Knuth&Bendix-komplettering.

Jeg har i et arbeidsnotat [Han94] skissert en naïv metode som bygger på generalisering utfra eksempler. Tanken er at det skal være mulig å sette en øvre grense for eksempler som må sees, utfra syntaktiske egenskaper i den indirekte spesifikasjonen. Denne metoden er her ikke utarbeidet eller verifisert på noen måte, og det virker vanskelig å finne *generelle* egenskaper i ligningsmengder som kan overføres til en slik øvre grense. Det er likevel ikke utenkelig at visse heuristiske regler kan utarbeides for visse forutsetninger. Vi kan bare nevne eksempelsvis at metoden gitt E_n for $n = 2$ fra eksempel 65 på side 116 på et tidspunkt gir

$$E_2^{dir} = \left\{ \begin{array}{l} \mathsf{pred(succ(0))} = \mathsf{0}, \\ \mathsf{pred(succ(succ(0)))} = \mathsf{succ(0)}, \\ \mathsf{succ(succ(succ(0)))} = \mathsf{0}, \\ \mathsf{succ(pred(x))} = \mathsf{x} \end{array} \right\}$$

(Dette er E_n^{dir} for $n = 2$ fra eksempel 75 på side 125).

E_2^{dir} er generert på et tidspunkt hvor det virker opplagt at ingen flere eksempler behøver å sees. Dette er for metoden grunnet i at ingen regel i E_2 «betrakter/ser forskjell på» større deler av generator-uttrykk enn dem av lengde 3. Det kunne her tenkes at det utfra dette var mulig å bevise en øvre grense for lengden av termer som må prøves i eksempler.

Det er også nærliggende å tenke seg muligheten for etablering av indre kongruens ved en variant av den naïv metoden.

Generering av direkte spesifikasjon fra indirekte spesifikasjon kan sees som en form for *programsyntese* eller *programtransformasjon*. Det kan være fruktbart å se på idéer innen disse felt. Se f.eks. [DR91].

3.10 Andre temaer

Vi avslutter dette kapittel med noen tanker som viderefører noen av temaene vi har diskutert:

3.10.1 Generaliserte *id*-utvidelser

Vi reduserte i avsnitt 3.4.5 initialsemantikk relativ til indirekte semantikk til basis-initialsemantikk. Dette gjorde vi ved hjelp av *id*-utvidelser av indirekte spesifikasjoner. En forutsetning var da at den indirekte spesifikasjonen kunne sees som en algebraisk beskrivelse av en kanonisk-representant funksjon. (Husk antagelsene `KREP1` og `KREP2`.)

Hva nå med indirekte spesifikasjoner som er algebraiske beskrivelser av klasserepresentant funksjoner andre enn kanonisk-representant funksjoner? Det er ting som tyder på at diskusjoner analoge til dem ført for algebraiske beskrivelser av kanonisk-representant funksjoner og *id*-utvidelser, kan føres for algebraiske beskrivelser av generelle klasserepresentant funksjoner og *generaliserte id*-utvidelser.

Slike generaliserte *id*-utvidelser skal simulere normale *id*-utvidelser. Dvs. dersom for spesifiserende symbol s

$$s(g) = f(g)$$

for alle (generator-)grunntermer g, må vi finne et funksjonssymbol h slik at

$$s(h(x)) = x$$

Et slikt funksjonssymbol h finnes kanskje allerede i term-universet, men må også kanskje innføres som et nytt symbol med tilhørende spesifikasjon.

Eksempel 82 Betrakt E_{gamma} fra eksempel 57 på side 108. Vi har

$$\mathsf{gamma}(g) \underset{E_{\mathsf{gamma}}}{\overset{*}{\leftrightarrow}} \mathsf{succ}(g)$$

for hver $g \in \mathcal{G}_{\mathsf{Int}}$.

Den generaliserte γ-*id*-utvidelse av E_{gamma} fremkommer da ved ligningen

$$\mathsf{gamma}(\mathsf{pred}(\mathsf{x})) = \mathsf{x}$$

○

Vi forfølger ikke dette temaet videre her. Vi viser bare følgende eksempel på komplettering av en en slik generalisert *id*-utvidelse.

Eksempel 83 Betrakt E_{gamma} fra eksempel 57 på side 108. La

$$\Sigma^h = \{\mathsf{mem}\}$$
$$\Sigma^c = \{\mathsf{0}, \mathsf{succ}, \mathsf{pred}\}$$

Vår eksekvering av en Knuth&Bendix-prosess gitt $E_{\mathsf{gamma}} \cup \{\mathsf{gamma(pred(x))=x}\}$ og en term-ordning der enhver $u \in \mathcal{T}_{\Sigma^h \cup \Sigma^c}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$, terminerer ikke, men gir den vedvarende regelmengden

$$R^{KB}_{\mathsf{gamma}} = \left\{ \begin{array}{l} \mathsf{gamma(x)} \rightarrow \mathsf{succ(x)}, \\ \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x}, \\ \\ \mathsf{mem(x,0)} \rightarrow \mathsf{x}, \\ \mathsf{mem(x,succ}^i\mathsf{(0))} \rightarrow \mathsf{succ}^i\mathsf{(x)}; 1 \leq i, \\ \mathsf{mem(x,pred}^i\mathsf{(0))} \rightarrow \mathsf{pred}^i\mathsf{(x)}; 1 \leq i, \\ \mathsf{mem(succ(x),y)} \rightarrow \mathsf{mem(x,succ(y))}, \\ \mathsf{mem(pred(x),y)} \rightarrow \mathsf{mem(x,pred(y))}, \\ \mathsf{mem(0,x)} \rightarrow \mathsf{x} \end{array} \right\}$$

(Argument for ikke-terminering og vedvarenhet som i eksempel 61 på side 113.) Alle regler uten den første er identiske med dem i R^{KB}_{delta} i eksempel 61 på side 113. Merk spesielt «Int-manifestet».

○

3.10.2 *id*-utvidelser og finalsemantikk

Hva skjer hvis ligningen $h(x) = x$ tilføres for et definert funksjonssymbol h ment å spesifisere semantikk ved prinsipp (2.11) på side 26?

Kan noe av vår diskusjon rundt *id*-utvidelser overføres til final-semantikk på denne måten? Kan final inkonsistens (se definisjon 2.12 side 33) oppdages og etableres ved metodene vi har nevnt for *id*-utvidelser? Er det naturlig å snakke

om *kunstig inkonsistens* i samband med final-semantikk, og kan isåfall kunstig inkonsistens elimineres ved f.eks. en operasjonell restriksjon av ligningslogikken for «finale id-utvidelser»?

Merk at hvis h er en observator, må typer droppes. Det er ikke sikkert at dette er problemfritt.

3.10.3 Fikspunkt-semantikk

Betrakt en funksjon f på et termunivers $\mathcal{G}_\Sigma$ slik at det for alle $g \in \mathcal{G}_\Sigma$ finnes et positivt naturlig tall n slik at

$$f^{n+i}(g) = f^n(g)$$

for alle $i \geq n$. Elementet $f^n(g)$ er et fikspunkt for f. Videre kan det her for hvert element g i domenet til f identifiseres et unikt slikt fikspunkt $g_{f_{fix}}$. Betrakt relasjonen

$$\text{fix}_f = \{\langle g, g_{f_{fix}} \rangle \mid g \in \mathcal{G}_\Sigma\}$$

Relasjonen fix_f er ikke bare en funksjon på $\mathcal{G}_\Sigma$, men dessuten en fikspunkt-funksjon på $\mathcal{G}_\Sigma$ (se (3.4) side 64), siden

$$\text{fix}_f(\text{fix}_f(g)) = \text{fix}_f(g_{f_{fix}}) = \text{fix}_f(g)$$

Merk at selve f ikke nødvendigvis er en fikspunktfunksjon.

Vi kan nå, dersom fix_f er semantikkgivende, spesifisere semantikk som ved prinsipp (3.2) på side 64:

$$g \simeq g' \;\Leftrightarrow\; \text{fix}_f(g) = \text{fix}_f(g')$$

La oss kalle en syntaktisk funksjon f for hvilket det finnes en slik semantikkgivende fix_f, en ***semantikkantydende*** funksjon. Merk at fix_f er en kanonisk-representant funksjon (se figur 3.1 side 67).

Det vi skal forslå her, er at algebraiske spesifikasjoner/beskrivelser av semantikk*antydende* syntaktiske funksjoner muligens kan brukes sammen med ligningslogikk i definisjon og resolusjon av semantikk.

Det analoge til indirekte spesifikasjon blir da for en E_s med 'spesifiserende symbol s':

$$g \simeq^s g' \;\Leftrightarrow\; \exists n, m \mid s^n(g) \overset{*}{\underset{E_s}{\leftrightarrow}} s^m(g')$$

Dersom E_s er konvergent kan høyresiden avgjøres ved følgende prosess:

La $g_0 = g$

Beregn $g_{i+1} = s(g_i)!E_s$

  inntil $s(g_{i+1})!E_s = s(g_i)!E_s$, for en $i = k$

La $g'_0 = g'$

Beregn $g'_{i+1} = s(g'_i)!E_s$

  inntil $s(g'_{i+1})!E_s = s(g'_i)!E_s$, for en $i = l$

Test $g_{k+1} = g'_{l+1}$

Merk at dersom E_s er en algebraisk spesifikasjon/beskrivelse av en semantikk-antydende syntaktisk funksjon, så må denne prosessen terminere; dvs k og l i prosess-skissen over finnes.

Eksempel 84 Betrakt $E'_{\delta_{\mathcal{I}nt}}$ fra eksempel 73 side 123 og eksempel 49 side 83:

$$E'_{\delta_{\mathcal{I}nt}} = \left\{ \begin{array}{l} \mathsf{delta(succ(pred(x))) = delta(x)}, \\ \mathsf{delta(pred(succ(x))) = delta(x)}, \\ \mathsf{delta(succ(succ(x))) = succ(delta(succ(x)))}, \\ \mathsf{delta(pred(pred(x))) = pred(delta(pred(x)))}, \\ \mathsf{delta(succ(0)) = succ(0)}, \\ \mathsf{delta(pred(0)) = pred(0)}, \\ \mathsf{delta(0) = 0} \end{array} \right\}$$

Vi husker at $E'_{\delta_{\mathcal{I}nt}}$ ikke er indirekte $\mathcal{G}_{\mathsf{Int}}$-kongruent. Siden $E'_{\delta_{\mathcal{I}nt}}$ er tilstrekkelig $\{\mathsf{delta}\}$-komplett mhp. $\mathcal{G}_{\mathsf{Int}}$, er det lett å se at $E'_{\delta_{\mathcal{I}nt}}$ ikke kan være en algebraisk beskrivelse av en semantikkgivende syntaktisk funksjon.

Imidlertid er $E'_{\delta_{\mathcal{I}nt}}$ en spesifikasjon av en semantikk*antydende* syntaktisk funksjon f. Videre er

$$\mathrm{fix}_f = \{\langle g, g_{f_{fix}} \rangle \mid g \in \mathcal{G}_{\mathsf{Int}}\}$$

identisk med kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32 side 64.

$E'_{\delta_{\mathcal{I}nt}}$ er konvergent, og vi har f.eks.

$$\mathsf{delta(delta(pred(pred(succ(succ(0))))))}_E \xrightarrow[\delta_{\mathcal{I}nt}]{!} \mathsf{0} = \mathsf{0}_E \xleftarrow[\delta_{\mathcal{I}nt}]{!} \mathsf{delta(pred(succ(0)))}$$

og

$$\mathsf{delta(succ(pred(succ(pred(succ(pred(0)))))))}_E \xrightarrow[\delta_{\mathcal{I}nt}]{!} \mathsf{0} = \mathsf{0}_E \xleftarrow[\delta_{\mathcal{I}nt}]{!} \mathsf{delta(succ(pred(0)))}$$

○

Eksempel 85 Betrakt følgende utvidelse av E fra eksempel 74 på side 123:

$$E' = \left\{ \begin{array}{l} \mathsf{delta(succ(pred(x))) = x}, \\ \mathsf{delta(pred(succ(x))) = x}, \\ \mathsf{delta(succ(succ(x))) = succ(delta(succ(x)))}, \\ \mathsf{delta(pred(pred(x))) = pred(delta(pred(x)))}, \\ \mathsf{delta(succ(0)) = succ(0)}, \\ \mathsf{delta(pred(0)) = pred(0)}, \\ \mathsf{delta(0) = 0} \end{array} \right\}$$

E' er en spesifikasjon av en semantikkantydende syntaktisk funksjon h, slik at

$$\mathrm{fix}_h = \{\langle g, g_{h_{fix}} \rangle \mid g \in \mathcal{G}_{\mathsf{Int}}\}$$

er identisk med kanonisk-representant funksjonen $\delta_{\mathcal{I}nt}$ fra eksempel 32. E' er konvergent, og vi har f.eks.

$$\mathsf{delta}^n\mathsf{(succpred}^n\mathsf{(0))} \xrightarrow[E']{!} \mathsf{0}$$

○

3.11 Oppsummering

I dette kapitlet har vi introdusert en ny måte å spesifisere atomær semantikk på; nemlig *indirekte algebraisk spesifikasjon*. Som grunnlag og bakgrunn for indirekte algebraisk spesifikasjon, har vi semantikkgivende syntaktiske funksjoner. *Kanonisk-representant funksjoner* er her spesielt interessante, ved at bildet

til kanonisk-representant funksjoner er *naturlige klasserepresentanter*. Algebraiske beskrivelser av kanonisk-representant funksjoner utgjør naturlig indirekte algebraiske spesifikasjoner, men vi har sett at algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner andre enn kanonisk-representant funksjoner også kan fungere som indirekte algebraisk spesifikasjon.

Teorien utviklet i kapittel 2 anvendes, ved at initial- og final-semantikk *relativ til indirekte spesifisert kjernesemantikk* betraktes. Slik initial- og final-semantikk kan i utgangspunktet ikke sees som basis-semantikker. I mangel av resolusjonsmetoder for *generell* initial- og final-semantikk, har vi således et problem.

Imidlertid har vi under visse rimelige og interessante forutsetninger *redusert* semantikk relativ til indirekte kjerne, til basis-semantikker. Vi har viet spesiell oppmerksomhet til reduksjon til basis-initialsemantikk. Denne reduksjonen er gjort ved *id-utvidelser* av indirekte spesifikasjon. En grunnleggende forutsetning for benyttelsen av *id*-utvidelser i reduksjonen, er at den indirekte spesifikasjon er en algebraisk beskrivelse av en kanonisk-representant funksjon.

Vi har videre sett at indirekte spesifikasjon kan ha en *iboende* kilde til *inkonsistens*. Dette fordi algebraiske beskrivelser av semantikkgivende funksjoner naturlig har beskrivelser av hjelpefunksjoner.

Vi har i den sammenheng studert *id*-utvidelser videre, og har introdusert *kjernebevaring* for *id*-utvidelser. Kjernebevaring for *id*-utvidelser kan ved reduksjon, overføres til initial-konsistens relativt til indirekte spesifisert kjerne. Vi har sett at Knuth&Bendix-komplettering på *id*-utvidelser kan brukes for å oppdage mangel på slik kjernebevaring, og for å synliggjøre kjernebevaring, gitt viss annen informasjon.

Vi har også sett at Knuth&Bendix-komplettering av *id*-utvidelser, gitt kjernebevaring, også kan synliggjøre kjernesemantikken i form av en direkte algebraisk spesifikasjon. Dette kan sees som *program-transformasjon*. Vi har ettersøkt andre friere metoder å transformere indirekte spesifikasjoner til direkte spesifikasjoner på, enn ved den noe restriktive Knuth&Bendix-komplettering.

En forutsetning for indirekte spesifikasjon er *indirekte kongruens*. (Et analogt kongruenskrav må også stilles ved aktiv final-spesifikasjon ved prinsipp (2.11) på side 26, f.eks. ved spesifikasjon ved observatorer ved såkalt *observator-basis*. I [Dah92] beskrives denne teknikken, men kongruenskravet ansees som umiddelbart oppfylt under konvergent konstruktiv funksjonsspesifikasjon (TGI). At dette ikke holder demonstreres f.eks. av spesifikasjonene $E'_{\delta_{\mathcal{I}nt}}$ og E' fra eksempler 84 og 85 hhv., dersom delta sees som en observator og en skyggetype innføres.) En algebraisk beskrivelse som *ikke* er indirekte kongruent kan likevel, ved sin *id*-utvidelse, spesifisere fornuftig semantikk. Men da gir ikke våre metoder for synliggjøring noen mening.

Vi har videre innført begrepet *kunstig inkonsistens* for den inkonsistens som skyldes algebraisk beskrivelse av hjelpefunksjoner. Dette begrep har vi grunnet i at hjelpefunksjon(symbol)ene er *implementatoriske* hjelpemidler, og burde være skjult fra logikken. Vi har vist at kunstig inkonsistens kan elimineres, ved faktisk å *skjule* disse hjelpefunksjon(symbol)ene. Vi har vist dette ved en *operasjonell restriksjon* av ligningslogikk. Vi har spekulert at det er mulig å implementere denne operasjonelle skjuling i basis-resolusjonsmetoder, og vi har antydet noen resonnementer i den forbindelse. Vi har dessuten eliminert kunstig inkonsistens ved skjuling på spesifikasjonsnivå ved å innføre fullstendig ukompletthet, og ved å innføre typer. De to sistnevnte skaper imidlertid problemer i forbindelse med reduksjon til basis-initialsemantikk.

Våre resultater om Knuth&Bendix-komplettering av *id*-utvidelser antyder også at *basis-semantikk* spesifisering ved *id*-utvidelser i forbindelse med basis-semantisk resolusjon kan være litt søkt, for resolusjons-metoder som fordrer konvergente spesifikasjonsligninger. Ved vellykket komplettering av *id*-utvidelser,

fås jo ved vår kompletteringsstrategi, direkte algebraiske spesifikasjoner.

Hvorvidt man kan spesifisere andre semantikker med indirekte spesifikasjon enn med direkte spesifikasjon, er uklart. Det er også uklart om klassen av semantikker som kan spesifiseres indirekte ved algebraiske beskrivelser av kanonisk-representant funksjoner, er mindre enn klassen av semantikker som kan spesifiseres indirekte ved algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner generelt.

Det er likevel klart at indirekte spesifikasjon utgjør et interessant alternativ og supplement til direkte spesifikasjon. Vår indirekte spesifikasjon er, siden den er grunnet i syntaktiske funksjoner, generelt mer *operasjonell* i stil enn direkte spesifikasjon. Verifikasjon av indirekte spesifikasjoner kan således gjøres ved metoder utviklet for mer konkrete programmer. Om dette i seg selv er en fordel har vi ikke tatt virkelig stilling til.

Vi har også sett at indirekte spesifikasjon gir et *deterministisk* ligningslogisk grep om flere semantikker enn det direkte spesifikasjon gir. Således utvider indirekte spesifikasjon klassen av semantikker som er avgjørbare ved termomskriving i enkleste form.

Avslutningsvis har vi antydet noen videreføringer av våre idéer i dette avsnittet. Spesielt interessant er kanskje om resultater for *id*-utvidelser kan overføres til andre områder av spesifikasjon.

Kapittel 4

Sekvens-utvidet Knuth& Bendix-komplettering

Konvergente ligningsmengder er essensielle i flere ligningslogikk-baserte metoder som søker mekanisk-formell resonnering. Dersom en gitt ligningsmengde E ikke er konvergent, kan i noen tilfeller en Knuth&Bendix-prosess gitt E, gi en konvergent ligningsmengde R som er komplett for E. Det kan f.eks. da avgjøres algoritmisk hvorvidt en gitt ligning $s = t$ er en logisk konsekvens av E ved omskriving i R til unike normalformer $s!R$ og $t!R$. Ligningen $s = t$ er en logisk konsekvens av E hvis og bare hvis $s!R = t!R$.

Vi kan da ved omskrivingen også *konstruere* R-utledninger (syntaktiske objekter) $\langle s, \ldots, s!R\rangle$ og $\langle t, \ldots, t!R\rangle$. Vi kan således få et konstruktivt bevis/motbevis i R for at $s = t$ er en logisk konsekvens av E.

Imidlertid kan det av og til være ønskelig å se slike bevis i den opprinnnelige ligningsmengden E framfor i (den muligens svært forskjellig fra E) regel-/ligningsmengde R. Vi ønsker m.a.o. av og til E-utledninger framfor R-utledninger.

Selvfølgelig oppfylles dette ønsket hvis E i seg selv er konvergent, men ved ikke-konvergent E er det generelt ikke uten videre så lett å konstruere E-utledninger (mekanisk).

Vi presenterer i dette kapittel en metode som gjør det mulig å konstruere E-utledninger mekanisk utfra de korresponderende R-utledninger; for R komplett for E og R generert ved Knuth&Bendix-komplettering.

Kjernen i metoden er en enkel utvidelse av Knuth&Bendix-komplettering som består i å legge inn ekstra datastruktur som «husker» E-utledningshistorien til hver regel generert i løpet av kompletteringsprosessen. Hver regel generert vil således ha en E-utledning tilknyttet seg. Ved en vellykket prosess resulterende i en for E komplett R, kan da en E-utledning konstrueres mekanisk utfra enhver omskriving i R, ved de tilknyttede E-utledningene, og siden enhver omskriving i R er terminerende.

4.1 Definisjon av sekvens-utvidet Knuth&Bendix-komplettering

I dette avsnittet beskrives utvidelsen av Knuth&Bendix-komplettering. For en gitt ligningsmengde E og en regelmengde R komplett for E, er hensikten med utvidelsen å gjøre det mulig å konstruere E-utledninger fra R-utledninger. Vi trenger et kraftigere begrepsapparat omkring utledningssekvenser.

4.1.1 Utledningssekvenser

For en term-mengde $\mathcal{T}_\Sigma(\mathcal{V})$, la $\mathcal{S}eq_\Sigma$ betegne mengden av alle sekvenser over $\mathcal{T}_\Sigma(\mathcal{V})$. La videre $\mathcal{S}eq_\Sigma^+$ betegne mengden av alle ikke-tomme sekvenser over $\mathcal{T}_\Sigma(\mathcal{V})$. Vi skal betrakte mengdene $\mathcal{S}eq_\Sigma$ og $\mathcal{S}eq_\Sigma^+$ som generert av generator-funksjonene ε og $\vdash$, og $\langle\rangle$ og $\vdash$ hhv.:

Generatorfunksjoner:

$$\begin{array}{lll} \varepsilon & \in & \mathcal{S}eq_\Sigma \\ \langle\rangle & \in (\mathcal{T}_\Sigma(\mathcal{V}) & \rightarrow \mathcal{S}eq_\Sigma^+) \\ \vdash & \in (\mathcal{S}eq_\Sigma \times \mathcal{T}_\Sigma(\mathcal{V}) & \rightarrow \mathcal{S}eq_\Sigma^+) \end{array}$$

Vi skal ha behov for følgende funksjoner:

$$\begin{array}{lll} lt & \in (\mathcal{S}eq_\Sigma^+ & \rightarrow \mathcal{T}_\Sigma(\mathcal{V})) \\ rt & \in (\mathcal{S}eq_\Sigma^+ & \rightarrow \mathcal{T}_\Sigma(\mathcal{V})) \\ rr & \in (\mathcal{S}eq_\Sigma^+ & \rightarrow \mathcal{S}eq_\Sigma) \\ \vdash\!\!\!\vdash & \in (\mathcal{S}eq_\Sigma \times \mathcal{S}eq_\Sigma & \rightarrow \mathcal{S}eq_\Sigma) \\ \dashv & \in (\mathcal{T}_\Sigma(\mathcal{V}) \times \mathcal{S}eq_\Sigma & \rightarrow \mathcal{S}eq_\Sigma^+) \\ rev & \in (\mathcal{S}eq_\Sigma & \rightarrow \mathcal{S}eq_\Sigma) \end{array}$$

$$\begin{array}{lcl} lt(x \vdash t) & = & \begin{cases} t & \text{for } x = \varepsilon \\ lt(x) & \text{for } x \neq \varepsilon \end{cases} \\ rt(x \vdash t) & = & t \\ rr(x \vdash t) & = & \begin{cases} \varepsilon & \text{for } x = \varepsilon \\ rr(x) \vdash t & \text{for } x \neq \varepsilon \end{cases} \\ x \vdash\!\!\!\vdash y & = & \begin{cases} x & \text{for } y = \varepsilon \\ (x \vdash\!\!\!\vdash z) \vdash t & \text{for } y = z \vdash t \end{cases} \\ t \dashv x & = & \langle t \rangle \vdash\!\!\!\vdash x \\ rev(x) & = & \begin{cases} \varepsilon & \text{for } x = \varepsilon \\ t \dashv rev(y) & \text{for } x = y \vdash t \end{cases} \end{array}$$

Følgende to funksjoner er utvidelser til term-sekvenser av funksjonen $[]_p$ på termer og av substitusjoner på termer.

$$\begin{array}{lll} \{\}_p & \in (\mathcal{T}_\Sigma(\mathcal{V}) \times \mathcal{S}eq_\Sigma^+ & \rightarrow \mathcal{S}eq_\Sigma^+) \\ \sigma_\mathcal{S} & \in (\mathcal{S}eq_\Sigma^+ & \rightarrow \mathcal{S}eq_\Sigma^+) \end{array}$$

Funksjonen $\{\}_p$ legger en kontekst rundt hver komponent i en term-sekvens:

$$t\{x \vdash u\}_p \quad = \quad \begin{cases} \langle t[u]_p \rangle & \text{for } x = \varepsilon \\ t\{x\}_p \vdash t[u]_p & \text{for } x \neq \varepsilon \end{cases}$$

Funksjonen $\sigma_{\mathcal{S}}$ utfører en substitusjon på hver komponent i en termsekvens:

For en substitusjon $\sigma \in \mathcal{S}bst^{\mathcal{T}_\Sigma(\mathcal{V})}$:
$(x \vdash t)\sigma_{\mathcal{S}} \quad = \quad x\sigma_{\mathcal{S}} \vdash t\sigma$

Vi lar S_s^t betegne en sekvens x i $\mathcal{S}eq_\Sigma$ med $lt(x) = s$ og $rt(x) = t$.

Gitt en ligningsmengde E, betegner vi mengden av alle E-utledninger i $\mathcal{T}_\Sigma(\mathcal{V})$ med $\mathcal{S}eq_\Sigma(E)$. Vi har $\mathcal{S}eq_\Sigma(E) \subseteq \mathcal{S}eq_\Sigma^+$.

Anta det finnes en E-utledning $S_s^t \in \mathcal{S}eq_\Sigma(E)$. Siden relasjonen $\overset{*}{\underset{E}{\leftrightarrow}}$ i $\mathcal{T}_\Sigma(\mathcal{V})$ er monoton mhp. substitusjon og monoton mhp. kontekstapplikasjon, vet vi at også $S_{c[s\sigma]}^{c[t\sigma]} \in \mathcal{S}eq_\Sigma(E)$ finnes, for vilkårlige $\sigma \in \mathcal{S}bst^{\mathcal{T}_\Sigma(\mathcal{V})}$ og kontekst $c \in \mathcal{T}_\Sigma(\mathcal{V})$.

Imidlertid er vi i dette avsnittet opptatt av å *konstruere* utledninger. Det skal i den sammenheng være interessant å vite hvordan vi fra en utledning $S_s^t \in \mathcal{S}eq_\Sigma(E)$ kan konstruere (mekanisk) en utledning $S_{c[s\sigma]}^{c[t\sigma]} \in \mathcal{S}eq_\Sigma(E)$.

Vi presenterer derfor følgende lemmata som det er svært lett å innse riktigheten av. For ordens skyld gjennomfører vi likevel bevis for dem. Det første lemma beskriver konstruksjon ved $\sigma_{\mathcal{S}}$:

Lemma 4.1 *La Σ være en vilkårlig signatur. La E være en vilkårlig ligningsmengde. La σ være en vilkårlig substitusjon i $\mathcal{S}bst^{\mathcal{T}_\Sigma(\mathcal{V})}$.*

Anta gitt en E-utledning $S_s^t \in \mathcal{S}eq_\Sigma(E)$. Da har vi $S_s^t\sigma_{\mathcal{S}} \in \mathcal{S}eq_\Sigma(E)$, og $lt(S_s^t\sigma_{\mathcal{S}}) = s\sigma$ og $rt(S_s^t\sigma_{\mathcal{S}}) = t\sigma$.

Bevis: Induksjon på lengden n av en vilkårlig $S_s^t \in \mathcal{S}eq_\Sigma(E)$.

$n = 1$: Da er $s = t$ og $S_s^s\sigma_{\mathcal{S}} = \langle s\sigma \rangle$ er trivielt en E-utledningssekvens i $\mathcal{T}_\Sigma(\mathcal{V})$. Videre er $lt(S_s^s\sigma_{\mathcal{S}}) = s\sigma$ og $rt(S_s^s\sigma_{\mathcal{S}}) = s\sigma$.

$n = k + 1; k \geq 1$: Da er $S_s^t = \langle s, \ldots, s_k, t \rangle$, og da ved definisjonen av σ_S:

$$S_s^t\sigma_S = S_s^{s_k}\sigma_S \vdash t\sigma$$

Induksjonshypotesen gir $S_s^{s_k}\sigma_S \in \mathcal{S}eq_\Sigma(E)$ og $lt(S_s^{s_k}\sigma_S) = s\sigma$ og $rt(S_s^{s_k}\sigma_S) = s_k\sigma$. Siden $s_k \overset{*}{\underset{E}{\leftrightarrow}} t$ og ved monotonitet mhp. substitusjon, har vi $s_k\sigma \overset{*}{\underset{E}{\leftrightarrow}} t\sigma$. Følgelig er $S_s^t\sigma_S \in \mathcal{S}eq_\Sigma(E)$. Vi har $lt(S_s^t\sigma_S) = lt(S_s^{s_k}\sigma_S \vdash t\sigma) = lt(S_s^{s_k}\sigma_S)$ som ved induksjonshypotesen er identisk med $s\sigma$. Videre er $rt(S_s^t\sigma_S) = rt(S_s^{s_k}\sigma_S \vdash t\sigma) = t\sigma$.

□

Neste lemma beskriver konstruksjon ved $\{\}_p$:

Lemma 4.2 *La Σ være en vilkårlig signatur. La E være en vilkårlig ligningsmengde. La $c \in \mathcal{T}_\Sigma(\mathcal{V})$ være en vilkårlig konstekst.*

Anta gitt en E-utledning $S_s^t \in \mathcal{S}eq_\Sigma(E)$. Da har vi $c\{S_s^t\} \in \mathcal{S}eq_\Sigma(E)$, og $lt(c\{S_s^t\}) = c[s]$ og $rt(c\{S_s^t\}) = c[t]$.

Bevis: Induksjon på lengden n av en vilkårlig $S_s^t \in \mathcal{S}eq_\Sigma(E)$.

$n = 1$: Da er $s = t$ og $c\{S_s^t\} = \langle c[s] \rangle$ er trivielt en E-utledningssekvens i $\mathcal{T}_\Sigma(\mathcal{V})$. Videre er $lt(c\{S_s^s\}) = c[s]$ og $rt(c\{S_s^s\}) = c[s]$.

$n = k + 1, k \geq 1$: Da er $S_s^t = \langle s, \ldots, s_k, t \rangle$, og da ved definisjonen av $\{\}_p$:

$$c\{S_s^t\} = c\{S_s^{s_k}\} \vdash c[t]$$

Induksjonshypotesen gir $c\{S_s^{s_k}\} \in \mathcal{S}eq_\Sigma(E)$ og $lt(c\{S_s^{s_k}\}) = c[s]$ og $rt(c\{S_s^{s_k}\}) = c[s_k]$. Siden $s_k \overset{*}{\underset{E}{\leftrightarrow}} t$ og ved monotonitet, har vi $c[s_k] \overset{*}{\underset{E}{\leftrightarrow}} c[t]$. Følgelig er $c\{S_s^t\} \in$

$\mathcal{S}eq_\Sigma(E)$. Vi har $lt(c\{S_s^t\}) = lt(c\{S_s^{s_k}\} \vdash c[t]) = lt(c\{S_s^{s_k}\})$ som ved induksjonshypotesen er identisk med $c[s]$. Videre er $rt(c\{S_s^t\}) = rt(c\{S_s^{s_k}\} \vdash c[t]) = c[t]$.
□

Funksjonene $\sigma_\mathcal{S}$ og $\{\}_p$ er mekaniserbare. Lemma 4.1 og lemma 4.2 sier derfor at en utledning $S_{c[s\sigma]}^{c[t\sigma]} \in \mathcal{S}eq_\Sigma(E)$ kan konstrueres mekanisk fra en utledning S_s^t.

Funksjonene $\mapsto$ og rev er også opplagt mekaniserbare, så andre «konstruksjonsmetoder» vi får bruk for opppsummeres i følgende observasjon:

Observasjon 4.3 *La Σ være en vilkårlig signatur. La E være en vilkårlig ligningsmengde. La S_s^t og S_t^u være to vilkårlige utledninger i $\mathcal{S}eq_\Sigma(E)$. Da er*

1. *$rev(S_s^t) \in \mathcal{S}eq_\Sigma(E)$, og $lt(rev(S_s^t)) = t$ og $rt(rev(S_s^t)) = s$.*
2. *$S_s^t \mapsto rr(S_t^u) \in \mathcal{S}eq_\Sigma(E)$, og $lt(S_s^t \mapsto rr(S_t^u)) = s$ og $rt(S_s^t \mapsto rr(S_t^u)) = u$.*

Det sentrale for oss er nå:

Sats 4.4 *La Σ være en vilkårlig signatur. La E være en vilkårlig ligningsmengde, og la R være en for E komplett regelmengde. Anta $s \underset{R}{\rightarrow} t$ for vilkårlige $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$. Dvs. det fins en regel $v \rightarrow h$ i R, en posisjon p i s og en substitusjon σ slik at $s|p = v\sigma$ og $t = s[h\sigma]_p$.*

Anta en $S_v^h \in \mathcal{S}eq_\Sigma(E)$. Da er $s\{S_v^h\sigma_\mathcal{S}\}_p \in \mathcal{S}eq_\Sigma(E)$. Videre er $lt(s\{S_v^h\sigma_\mathcal{S}\}_p) = s$ og $rt(s\{S_v^h\sigma_\mathcal{S}\}_p) = t$. Altså kan en S_s^t konstrueres fra S_v^h.

Bevis: Det finnes en $S_v^h \in \mathcal{S}eq_\Sigma(E)$, siden R er komplett for E. Resten er enkelt: Ved lemma 4.1 kan en $S_{v\sigma}^{h\sigma} \in \mathcal{S}eq_\Sigma(E)$ konstrueres ved $S_v^h\sigma_\mathcal{S}$. Ved lemma 4.2 kan videre en $S_{s[v\sigma]_p}^{s[h\sigma]_p} \in \mathcal{S}eq_\Sigma(E)$ konstrueres ved $s\{S_{v\sigma}^{h\sigma}\}_p$; dvs. ved $s\{S_v^h\sigma_\mathcal{S}\}$. Men nå er $s[v\sigma]_p = s$ og $s[h\sigma]_p = t$, så vi kan altså konstruere en S_s^t.
□

Det er enkelt å utvide sats 4.4 fra ett enkelt R-omskrivningssteg $\underset{R}{\rightarrow}$ til $\overset{*}{\underset{R}{\leftrightarrow}}$:

Sats 4.5 *La Σ være en vilkårlig signatur. La E være en vilkårlig ligningsmengde, og la R være en for E komplett regelmengde. Anta $s \overset{*}{\underset{R}{\leftrightarrow}} t$ for vilkårlige $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$.*

Da kan en utledningssekvens $S_s^t \in \mathcal{S}eq_\Sigma(E)$ konstrueres, gitt utledningssekvenser $S_v^h \in \mathcal{S}eq_\Sigma(E)$ for hver regel $v \rightarrow h \in R$.

Bevis: Induksjon over lengden n til en vilkårlig R-utledning $\langle s, \ldots, t\rangle$ i $\mathcal{T}_\Sigma(\mathcal{V})$.

$n = 1$: Trivielt.

$n = k+1; k \geq 1$: Da har vi en R-utledning $\langle s, \ldots, s_k, t\rangle$. Induksjonshypotesen gir at en $S_s^{s_k}$ kan konstrueres. Anta $s_k \underset{R}{\rightarrow} t$. Da kan ved sats 4.4 $S_{s_k}^t$ konstrueres og satsen følger ved observasjon 4.3 punkt 2. Anta $s_k \underset{R}{\leftarrow} t$. Da kan ved sats 4.4 $S_t^{s_k}$ konstrueres. Da kan en $S_{s_k}^t$ konstrueres ved $rev(S_t^{s_k})$ (observasjon 4.3 punkt 1), og satsen følger ved observasjon 4.3 punkt 2.
□

Så anta en ikke-konvergent ligningsmengde E, og en regelmengde R komplett for E. Fra en R-utledning for $s \overset{*}{\underset{R}{\leftrightarrow}} t$, kan altså en E-utledning S_s^t konstrueres mekanisk utfra E-utledninger for reglene i R brukt i R-utledningen. Eller snarere: Ved mekanisk R-omskrivning av to termer s og t til felles normalform, kan samtidig en E-utledning S_s^t konstrueres; *gitt at det forefinnes E-utledninger for hver av reglene i R.*

Konstruksjonen av E-utledninger for hver av reglene i R, er nettopp det vår utvidelse av Knuth&Bendix-komplettering tar seg av.

4.1.2 Utvidelsen

Vi beskriver nå *sekvens-utvidelse* av Knuth&Bendix-komplettering. Datastrukturen i vanlig Knuth&Bendix-komplettering består av en ligningsmengde og en regelmengde. Vi utvider datastrukturen slik at det assosieres en utledningssekvens til hver ligning og regel. Datastrukturen blir således en $\langle ES, RS\rangle$ hvor $ES \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V})) \times \mathcal{S}eq_\Sigma$, og $RS \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V})) \times \mathcal{S}eq_\Sigma$. For en $\langle ES, RS\rangle$ definerer vi:

$$E_\partial(ES) = \left\{s = t \mid \left\langle s = t, S_s^t\right\rangle \in ES\right\}$$

$$R_\partial(RS) = \left\{v \rightarrow h \mid \left\langle v \rightarrow h, S_v^h\right\rangle \in RS\right\}$$

$$S_\partial(ES) = \left\{S_s^t \mid \left\langle s = t, S_s^t\right\rangle \in ES\right\}$$

$$S_\partial(RS) = \left\{S_v^h \mid \left\langle v \rightarrow h, S_v^h\right\rangle \in RS\right\}$$

Sekvens-utvidet Knuth&Bendix-komplettering er beskrevet som et formelt system i figur 4.1.

4.1.3 Godtgjørelse for sekvens-utvidet Knuth&Bendix-komplettering

Vår utvidelse av Knuth&Bendix-komplettering er rett fram; den består kun i utvidet struktur og utvidelser i inferensreglene mhp. på den utvidete strukturen. Tilleggsstrukturen og utvidelsene i inferensreglene interfererer ikke med den opprinnelige struktur og dennes utvikling gjennom de opprinnelige inferensregler. Altså: $\langle E, R\rangle = \langle E_\partial(ES), R_\partial(RS)\rangle$ for en $\langle ES, RS\rangle$ i det utvidete inferensregelsettet er urørt av utvidelsen. Følgelig gjelder invarianter og satser om $\langle E, R\rangle$ vist for det opprinnelige inferensregelsettet, derfor også for $\langle E, R\rangle$ i det utvidete settet.

Vi skal nå vise følgende teorem som beskriver utledningene i en inferenssekvens i sekvens-utvidet Knuth&Bendix-komplettering.

Teorem 4.6 *La*

$$\langle\langle ES_0, RS_0\rangle, \ldots, \langle ES_k, RS_k\rangle, \ldots\rangle$$

være en inferens-sekvens i sekvens-utvidet Knuth-Bendix komplettering. For en vilkårlig $\langle ES_i, RS_i\rangle$ i sekvensen er

$$RS_i = \{\left\langle v \rightarrow h, S_v^h\right\rangle \mid v \rightarrow h \in R_\partial(RS_i)\}$$

$$ES_i = \{\left\langle s = t, S_s^t\right\rangle \mid s = t \in E_\partial(ES_i)\}$$

der S_v^h og S_s^t er $E_0 = E_\partial(ES_0)$-utledninger. Dvs.

$$S_\partial(RS_i), S_\partial(ES_i) \subseteq \mathcal{S}eq_\Sigma(E_0)$$

Godtgjørelse: (Induksjon på i)

$i = 0$: Opplagt ved inspeksjon av **S-Init**.

$i = l; l > 0$: Ved induksjonshypotesen holder teoremet for $\langle ES_{l-1}, RS_{l-1}\rangle$. En av reglene utenom **S-Init** må være brukt for å oppnå $\langle ES_l, RS_l\rangle$.

Anta **S-Forenkle1** er brukt. Induksjonshypotesen gir at en S_s^t er konstruert. Vi må godtgjøre at det er mulig å konstruere S_s^r. La $R = (R_\partial(RS_{l-1}))$.

Datastruktur:

$\langle ES, RS\rangle$ hvor $ES \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V})) \times \mathcal{S}eq_\Sigma$, og $RS \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V})) \times \mathcal{S}eq_\Sigma$. $R = (R_\partial(RS))$. Relasjonen $\succ$ er den gitte reduksjonsordning.

Inferensregler:

S-Init: For en $E_0 \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$:

$$\langle \{\langle v = h, \langle v, h\rangle\rangle | v = h \in E_0\}, \emptyset\rangle$$

S-Forenkle1:

$$\frac{\langle ES \cup \{\langle s = t, S_s^t\rangle\}, RS\rangle}{\langle ES \cup \{\langle r = t, rev(S_s^r) \mapsto rr(S_s^t)\rangle\}, RS\rangle}, \; s \underset{R}{\rightarrow} r$$

S-Forenkle2:

$$\frac{\langle ES \cup \{\langle s = t, S_s^t\rangle\}, RS\rangle}{\langle ES \cup \{\langle s = r, S_s^t \mapsto rr(S_t^r)\rangle\}, RS\rangle}, \; t \underset{R}{\rightarrow} r$$

S-Slett:

$$\frac{\langle ES \cup \{\langle s = s, S_s^s\rangle\}, RS\rangle}{\langle ES, RS\rangle}$$

S-Orienter1:

$$\frac{\langle ES \cup \{\langle s = t, S_s^t\rangle\}, RS\rangle}{\langle ES, RS \cup \{\langle s \rightarrow t, S_s^t\rangle\}\rangle}, \; s \succ t$$

S-Orienter2:

$$\frac{\langle ES \cup \{\langle s = t, S_s^t\rangle\}, RS\rangle}{\langle ES, RS \cup \{\langle t \rightarrow s, rev(S_s^t)\rangle\}\rangle}, \; t \succ s$$

S-Sammensett:

$$\frac{\langle ES, RS \cup \{\langle s \rightarrow t, S_s^t\rangle\}\rangle}{\langle ES, RS \cup \{\langle s \rightarrow r, S_s^t \mapsto rr(S_t^r)\rangle\}\rangle}, \; t \underset{R}{\rightarrow} r$$

S-Kollaps:

$$\frac{\langle ES, RS \cup \{\langle s \rightarrow t, S_s^t\rangle\}\rangle}{\langle ES \cup \{\langle r = t, rev(S_s^r) \mapsto rr(S_s^t)\rangle\}, RS\rangle}, \; s \underset{R'}{\rightarrow} r$$

for $R' \subseteq R \setminus \{s \rightarrow t\}$.

S-Utled:

$$\frac{\langle ES, RS\rangle}{\langle ES \cup \{\langle s = t, rev(S_u^s) \mapsto rr(S_u^t)\rangle\}, RS\rangle}$$

for $\langle s, t\rangle$ et ekte kritisk par fra u i R.

Figur 4.1: Inferensregler i sekvens-utvidet Knuth&Bendix-komplettering.

Regelmengden R er en del av den opprinnelige struktur. I vanlig Knuth&Bendix-komplettering har vi invariant at R er terminerende og endelig. Betingelsen s $\overrightarrow{}_R$ r for regelen kan derfor bestemmes algoritmisk.

Ved induksjonshypotesen har vi at en $S_v^h \in S_\partial(RS_{l-1})$ er konstruert for regelen $v \rightarrow h$ i R brukt ved omskrivingen $s \overrightarrow{}_R r$. Det følger av sats 4.4 at en algoritme som bestemmer om $s \overrightarrow{}_R r$, på veien også kan konstruere S_s^r. Derfor gir det mening å snakke om en konstruert S_s^r kun utfra betingelsen $s \overrightarrow{}_R r$. Induksjonssteget følger så ved inspeksjon av **S-Forenkle1** og ved induksjonshypotesen.

Analoge argumenter kan brukes for reglene **S-Forenkle2**, **S-Sammensett** og **S-Kollaps**.

Anta så at regelen **S-Utled** er brukt. Her må det godtgjøres at vi kan snakke om konstruerte S_u^s og S_u^t. Betingelsen $\langle s,t\rangle$ et ekte kritisk par fra u i R kan igjen bestemmes algoritmisk. En algoritme for å finne kritiske par vil kunne, gitt at den har funnet et kritisk par $\langle s,t\rangle$, finne en u som kan omskrives (i ett skritt) i R til s og t. Ved induksjonshypotesen har vi da at $S_v^h, S_{v'}^{h'} \in S_\partial(RS_{l-1})$ for reglene $v \rightarrow h$ og $v' \rightarrow h'$ i R brukt ved omskrivingene $u \overrightarrow{}_R s$ og $u \overrightarrow{}_R t$. Det følger så av sats 4.4 at S_u^s og S_u^t kan konstrueres.

Induksjonen for de øvrige regler er opplagt.

$\diamond$

Anta at sekvens-utvidet Knuth&Bendix-komplettering er vellykket, gitt en $E_0 \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$. La $\langle ES_\infty, RS_\infty\rangle$ være den induktive grense for denne vellykkete kompletteringen. Det følger fra resultater for standard Knuth&Bendix-komplettering at $R_\infty = R_\partial(RS_\infty)$ er et komplett omskrivningssystem for E_0. Videre følger det fra teorem 4.6 at $S_\partial(RS_\infty)$ vil for hver regel i R_∞, inneholde en korresponderende E_0-utledningssekvens — et E_0-*bevis* for regelen.

*

Idéen bak vår måte å konstruere E-utledninger fra R-utledninger for R en regelmengde komplett for E, er på ingen måte raffinert. Den gir dog en mulighet for å konstruere E-utledninger mekanisk for ikke-konvergent E.

En umiddelbar observasjon man kan komme med for vår metode, er at måten hver S_v^h for hver regel $v \rightarrow h$ generert i prosessen bygges opp på er «blind»; i den forstand at utviklingen av hver S_v^h slavisk følger utviklingen av «sin» $v \rightarrow h$. Dessuten er metoden svært *lokal* — dette er bra beregningsmessig — men gir helt sikkert ikke de mest elegante bevisene i E. Tvert imot har man følelsen av at bevisene blir svært lange med mange identiske deler.

En opplagt utfordring i tilknytning til konstruksjon av utledninger eller bevis er konstruksjon av korteste bevis. Vi går ikke videre med dette her, men henviser til diskusjoner innen feltet *bevisteori*.

Ligningslogikk kan, ved siden av å presenteres ved term-omskrivning som vi har gjort, også presenteres som et inferenssystem. «Bevisene» i et slikt inferenssystem er ikke utledningssekvenser, men *bevis-trær*. Det er mulig å definere *tre-utvidet* Knuth&Bendix-komplettering, som konstruerer bevis-trær istedenfor utledningssekvenser for hver regel generert i kompletteringsprosessen. Dette er gjort i [Han92] på en analog måte som for sekvens-utvidet Knuth&Bendix-komplettering.

4.2 Anvendelser av sekvens-utvidet Knuth&Bendix-komplettering

Vi skal antyde noen anvendelser av sekvens-utvidet Knuth&Bendix-komplettering i relasjon til diskusjonen forøvrig. Først tar vi igjen opp temaet *id*-utvidelser og kjernebevaring fra avsnitt 3.6.1 i kapittel 3. Deretter ser vi på sekvens-utvidet Knuth&Bendix-komplettering i tilknytning til diskusjonen om basis-initialsemantikk og induktive konsekvenser fra kapittel 2.

4.2.1 *id*-utvidelser og kjernebevaring igjen

Vi tar nå opp igjen temaet fra avsnitt 3.6.1 og diskuterer *id*-utvidelser av indirekte spesifikasjoner og kjernebevaring. Vi husker at *id*-utvidelser av indirekte spesifikasjoner under visse forutsetninger er identisk med initialsemantikk relativ til indirekte spesifisert kjerne. Kjernebevaring for *id*-utvidelser overføres da til *initiell konsistens*.

Som argumentert i avsnitt 3.6.1 side 115, er det mulig å oppdage mangel på kjernebevaring av *id*-utvidelser mekanisk og i endelig tid. Forutsetningene i vårt argument var et endelig kjerne-manifest for *id*-utvidelsen og konvergent indirekte spesifikasjon av kjernen.

Men det virker ikke så greit å etablere *tilstedeværelse* av kjernebevaring for *id*-utvidelser. Vår strategi i avsnitt 3.6.2 side 120 fordrer i en passende forstand verifikasjon av den indirekte spesifikasjon. Vi skal her se om sekvens-utvidet Knuth&Bendix-komplettering kan bidra med noe for å etablere kjernebevaring.

Anta $\hat{E}$ inneholder en indirekte spesifikasjon E_s med spesifiserende symbol s, av en semantikk på en $\mathcal{G}_{\Sigma^c}$. Anta sekvens-utvidet Knuth&Bendix-komplettering anvendes på s-*id*-utvidelsen $\hat{E}^{id}$ av $\hat{E}$, og at prosessen gir et vedvarende Σ^c-manifest R for $\hat{E}^{id}$. Ved sekvens-utvidet Knuth&Bendix-komplettering, kan nå en $\hat{E}^{id}$-utledning $\langle v, \ldots, h\rangle$ konstrueres for hver regel $v \rightarrow h$ i R.

Idéen i dette avsnittet er at det kanskje er mulig å å finne syntaktiske kjennetegn i disse $\hat{E}^{id}$-utledningene som etablerer kjernebevaring for $\hat{E}^{id}$. Dette er en besnærende tanke. Vi kan ihvertfall komme med følgende enkle resultat:

Teorem 4.7 *Anta enhver regel $v \rightarrow h$ i Σ^c-manifestet R har assosiert en $\hat{E}^{id}$-utledning på formen*

$$\begin{array}{ccc} s(v) & \xleftarrow[E_s]{*} & s(h) \\ {\scriptstyle id}\nwarrow & & \nearrow{\scriptstyle id} \\ v & & h \end{array}$$

Da er $\hat{E}^{id}$ kjernebevarende.

Bevis: Denne formen gir direkte at

$$s(v) \overset{*}{\underset{E_s}{\leftrightarrow}} s(h)$$

for alle regler $v \rightarrow h$ i R. Siden vi ved monotonitet også har $s(v)\sigma \overset{*}{\underset{E_s}{\leftrightarrow}} s(h)\sigma$; m.a.o. $s(v\sigma) \overset{*}{\underset{E_s}{\leftrightarrow}} s(h\sigma)$ for vilkårlig substitusjon σ, gir sats 3.29 på side 118 da at $\hat{E}^{id}$ kjernebevarende.

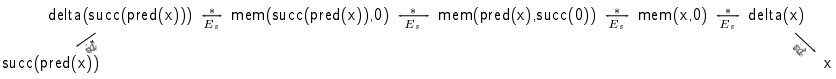
□

Eksempel 86 Betrakt E_{delta} fra eksempel 56 på side 108. I eksempel 61 side 113 konstaterte vi at

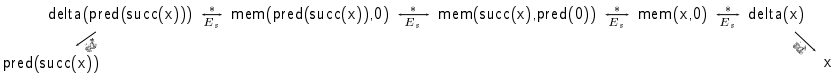
$$R^{man}_{\mathsf{delta}} = \left\{ \begin{array}{l} \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \end{array} \right\}$$

er et vedvarende Int-manifest for delta-*id*-utvidelsen av E_{delta}.

Videre er R^{man}_{delta} generert ved Knuth&Bendix-komplettering. *Sekvensutvidet* Knuth&Bendix-komplettering gir for reglene i R^{man}_{delta} her utledningssekvensene



og



Ved teorem 4.7 er derfor delta-*id*-utvidelsen av E_{delta} kjernebevarende.

To vesentlige poeng: Merk at kjernebevaring her kan slås fast uten å verifisere E_{delta}. På den annen side kan nå denne viten om kjernebevaring brukes for å verifisere E_{delta}. Se avsnitt 3.6.4 side 122.
○

Faktisk kan vi under visse antagelser komme med en sterkere påstand enn teorem 4.7. Vi benytter signaturer, ligningsmengder osv. som vist i figur 3.7 side 85:

Teorem 4.8 *Anta* $\mathtt{T\hat{\Sigma}K}$, `DISJ` *og* `KREP1`. *Anta enhver regel* $v \to h$ *i* Σ^c*-manifestet* R *har assosiert en* $\hat{E}^{id}$*-utledning der enhver anvendelse av ligningen* $s(x) = x$ *forekommer kun i kontekster* $c \in \mathcal{G}_{\Sigma^c \cup \{s\}}$. *Da er* $\hat{E}^{id}$ *kjernebevarende.*

Teorem 4.8 følger fra følgende to lemmata:

Lemma 4.9 *Anta* $\mathtt{T\hat{\Sigma}K}$ *og* `KREP1`.

$$\overleftrightarrow{\hat{E}^{id}}^{*\,s}_{\Sigma^c \cup \{s\}} \mathcal{G}_{\hat{\Sigma}} = (\simeq^s \cup \overleftrightarrow{\hat{E}}^{*} \mathcal{G}_{\hat{\Sigma}})^{\leftarrow_{\Sigma^c \cup \{s\}}^{*} \rightarrow}$$

Bevis: Lemmaet følger fra sats 3.30 på side 127.
□

Lemma 4.10 *Anta* $\mathtt{T\hat{\Sigma}K}$ *og* `KREP1`.

$$(\simeq^s \cup \overleftrightarrow{\hat{E}}^{*} \mathcal{G}_{\hat{\Sigma}})^{\leftarrow_{\Sigma^c \cup \{s\}}^{*} \rightarrow} = (\simeq^s \cup \overleftrightarrow{\hat{E}}^{*} \mathcal{G}_{\hat{\Sigma}})^{*}$$

Bevis: Lemmaet følger fra følgende påstand (siden forskjellen i de to relasjoner kun er i grad av tillukning): Dersom

$$c[g]\ (\simeq^s \cup \overleftrightarrow{\hat{E}}^{*} \mathcal{G}_{\hat{\Sigma}})^{\leftarrow_{\Sigma^c \cup \{s\}}^{*} \rightarrow}\ c[g']$$

slik at

$$g \simeq^s g'$$

for en vilkårlig (muligens tom) kontekst $c \in \mathcal{G}_{\Sigma^c \cup \{s\}}$, så er

$$c[g]\ (\simeq^s \cup \overleftrightarrow{\hat{E}}^{*} \mathcal{G}_{\hat{\Sigma}})^{*}\ c[g']$$

Påstanden vises ved induksjon på kompleksiteten av konteksten c: Merk at $g, g' \in \mathcal{G}_{\Sigma^c}$.

Tom kontekst c: Trivielt, siden $g \simeq^s g'$.

Anta $c = c'[s(\Box)]$. Vi minner nå om (3.13) side 92. Vi antar implisitt `INDKONG` (ved å nevne $\simeq^s$), så vi har ved $\mathtt{T\hat{\Sigma}K}$ og `KREP1`:

$$s(g) \overleftrightarrow{\hat{E}_s}^{*} \tilde{g} \simeq^s g \simeq^s g' \simeq^s \tilde{g'} \overleftrightarrow{\hat{E}_s}^{*} s(g')$$

for $\tilde{g}, \tilde{g'} \in \mathcal{G}_{\Sigma^c}$. Så ved induksjonshypotesen får vi

$$c'[s(g)] \overset{*}{\underset{E_s}{\leftrightarrow}} c'[\tilde{g}] \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^* \ c'[\tilde{g'}] \overset{*}{\underset{E_s}{\leftrightarrow}} c'[s(g')]$$

og vi har da at $c[g] \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^* \ c[g']$.

Anta $c = c'[f(g_1, \ldots, \Box, \ldots, g_n)]$, for et generatorsymbol f og $g_1, \ldots, g_n \in \mathcal{G}_{\Sigma^c \cup \{s\}}$. Ved $\mathtt{T\hat{\Sigma}K}$ har vi

$$f(g_1, \ldots, \Box, \ldots, g_n) \overset{*}{\underset{\hat{E}}{\leftrightarrow}} f(\tilde{g_1}, \ldots, \Box, \ldots, \tilde{g_n})$$

for $\tilde{g_1}, \ldots, \tilde{g_n} \in \mathcal{G}_{\Sigma^c}$. Vi har så ved kongruens:

$$f(\tilde{g_1}, \ldots, g, \ldots, \tilde{g_n}) \simeq^s f(\tilde{g_1}, \ldots, g', \ldots, \tilde{g_n})$$

La $\overline{g} = f(\tilde{g_1}, \ldots, g, \ldots, \tilde{g_n})$ og $\overline{g_{'}} = f(\tilde{g_1}, \ldots, g', \ldots, \tilde{g_n})$.

Dermed gir induksjonshypotesen

$$c'[f(g_1, \ldots, g, \ldots, g_n)] \overset{*}{\underset{\hat{E}}{\leftrightarrow}}$$

$$c'[\overline{g}] \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^* \ c'[\overline{g_{'}}]$$

$$\overset{*}{\underset{\hat{E}}{\leftrightarrow}} c'[f(g_1, \ldots, g', \ldots, g_n)]$$

og vi har da at $c[g] \ (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^* \ c[g']$.

□

Bevis for teorem 4.8:

Vi har antatt at enhver regel $v \to h$ i Σ^c-manifestet R har assosiert en $\hat{E}^{id}$-utledning med egenskapen at enhver anvendelse av ligningen $s(x) = x$ forekommer kun i kontekster $c \in \mathcal{G}_{\Sigma^c} \cup \{s\}$.

Observer at da finnes en $\hat{E}^{id}$-utledning $\langle v\sigma, \ldots, h\sigma \rangle$ med denne egenskap, for hver substitusjon $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}}$. Altså har vi

$$v\sigma \overset{\Sigma^c \cup \{s\}^{*s}}{\underset{\hat{E}^{id}}{\rightsquigarrow}} h\sigma$$

Lemma 4.9 og 4.10 gir så

$$\overset{\Sigma^c \cup \{s\}^{*s}}{\underset{\hat{E}^{id}}{\rightsquigarrow}} \mathcal{G}_{\hat{\Sigma}} = (\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*$$

Ved lemma 2.16 på side 41, er

$$(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^*_{\mathcal{G}_{\Sigma^c}} = \simeq^s$$

Følgelig har vi

$$v\sigma \simeq^s h\sigma$$

eller

$$s(v\sigma) \overset{*}{\underset{E_s}{\leftrightarrow}} s(h\sigma)$$

for vilkårlige $v \to h$ i R og $\sigma \in \mathcal{S}bst^{\mathcal{G}_{\Sigma^c}}$. Ved sats 3.29 på side 118, er da $\hat{E}^{id}$ kjernebevarende.

□

Lemmata 4.9 og 4.10 samt teorem 4.8 har varianter 4.9b og 4.10b og 4.8b for erstatning av antagelsen $\mathtt{T\hat{\Sigma}K}$ med $h\mathtt{ROT}$, samt $E_s\mathtt{VARBEVAR}$, $E^d\mathtt{VARBEVAR}$ og `DISJ`. Det er lett å verifisere dette.

Merk dessuten at lemma 4.10 er en ytterligere presisering av påstanden i avsnitt 3.5 om at inkonsistens ikke kan ha opphav i det spesifiserende symbol

alene (se sats 3.22 side 109). Ifølge lemma 4.10 er nemlig $(\simeq^s \cup \overset{*}{\underset{\hat{E}}{\leftrightarrow}} \mathcal{G}_{\hat{\Sigma}})^{\rightarrow_{\Sigma^c \cup \{s\}}^*\rightarrow}$ identisk med separabel semantikk, som under konsistens er identisk med generell initial- og degenerert finalsemantikk (teorem 2.20 side 42 og teorem 2.21 side 43). M.a.o.: Den (delvise) $\Sigma^c \cup \{s\}$-monotone tillukningen er ikke kilde til inkonsistens.

Vi kan ved teorem 4.8 (4.8b) etablere kjernebevaring av *id*-utvidelser. Satsen sier også at mangel på kjernebevaring vil, i et Σ^c-manifest, måtte tilkjennegi seg ved en regel hvis assosierte $\hat{E}^{id}$-utledningssekvens fremkommer ved anvendelser av $s(x) = x$ i $\Sigma^d \cup \Sigma^h$-kontekster.

Men vi kan ikke fange opp alle tilfeller av kjernebevaring, gitt Σ^c-manifest, på denne måten. Vi kan nemlig konstruere *kjernebevarende* *id*-utvidelser hvis Σ^c-manifest har regler med assosiert $\hat{E}^{id}$-utledningssekvens fremkommet ved anvendelser av $s(x) = x$ i $\Sigma^d \cup \Sigma^h$-kontekster.

Dette kan f.eks. gjøres ved å observere at premissen i teorem 4.7 fordrer at

> *hver regel $v \rightarrow h$ i Σ^c-manifestet R er slik at $s(v) = s(h)$ er en logisk konsekvens av E_s.*

Vi kan da prøve å konstruere en indirekte spesifikasjon hvis *id*-utvidelse er kjernebevarende, og slik at

1. Et Σ^c-manifest for *id*-utvidelsen eksisterer. Da må hver regel $v \rightarrow h$ i R være en logisk konsekvens av *id*-utvidelsen (side 115).

2. Σ^c-manifestet inneholder en regel $v \rightarrow h$ slik at $s(v) = s(h)$ *ikke* er en logisk konsekvens av E_s.

Eksempel 87 Vi betrakter generator-universet over

$$\mathsf{Int} = \left\{ \begin{array}{l} \mathsf{0 : Int}, \\ \mathsf{succ : Int \longrightarrow Int}, \\ \mathsf{pred : Int \longrightarrow Int} \end{array} \right\}$$

For semantikken på $\mathcal{G}_{\mathsf{Int}}$ spesifisert direkte av den ene ligning

$$E_a = \{\ \mathsf{succ(pred(x)) = x}\ \}$$

har vi fra eksempel 66 på side 117 følgende indirekte spesifikasjon:

$$E_b = \left\{ \begin{array}{ll} 1: & \mathsf{delta(x) = mem(x,0,0,0)}, \\ 2: & \mathsf{mem(succ(x),y,0,0) = mem(x,succ(y),0,0)}, \\ 3: & \mathsf{mem(succ(x),y,pred(z),0) = mem(x,succ(y),pred(z),0)}, \\ 4: & \mathsf{mem(pred(x),0,z,0) = mem(x,0,pred(z),0)}, \\ 5: & \mathsf{mem(pred(x),succ(y),z,0) = mem(x,y,z,0)}, \\ 6: & \mathsf{mem(0,succ(y),z,w) = mem(0,y,z,succ(w))}, \\ 7: & \mathsf{mem(0,0,pred(z),w) = mem(0,0,z,pred(w))}, \\ 8: & \mathsf{mem(0,0,0,w) = w} \end{array} \right\}$$

Vi skal svakt modifisere E_b til en E_b' slik at E_b' blir en indirekte spesifikasjon av semantikken spesifisert direkte av

$$E_a' = \left\{ \begin{array}{l} \mathsf{succ(pred(x)) = x}, \\ \mathsf{pred(succ(x)) = x} \end{array} \right\}$$

For å oppnå E_b' legger vi ligningen

$$5b: \quad \mathsf{mem(0,succ(y),pred(z),0) = mem(0,y,z,0)}$$

til E_b. Ligningene 3 og 5b tar nå hånd om «pred(succ(x)) = x-delen» av semantikken. Legg merke til at prosesseringen av denne del, gjøres først når hele «input»-termen er prosessert ferdig i mems 1. argument. Dette stikker kjepper i hjulene for at delta(pred(succ(x))) = delta(x) skal kunne være en logisk konsekvens av E'_b: En term med variabel blir nemlig aldri ferdigprosessert i mems 1. argument, og blokkerer endog all videre prosessering. (Altså må input-termen x til delta være en grunnterm for at delta(x) skal være omskrivbar til en kanonisk representant.)

I tillegg må ligning 6 endres til

$$6' : \quad \mathsf{mem(0,succ(y),0,w)} = \mathsf{mem(0,y,0,succ(w))}$$

(siden kanoniske representanter nå er rene succ- eller rene pred-termer.)

Vi ser at E'_b er en indirekte spesifikasjon av semantikken spesifisert direkte av E'_a. (Det 4. argument til mem er for E'_b ikke nødvendig spesifikasjonsmessig, men er sentralt for vårt argument her.)

Komplettering av $E_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ gir blant andre regler følgende:

$$R_b = \left\{ \begin{array}{l} \vdots \\ \mathsf{delta(x)} \rightarrow \mathsf{x}, \\ \vdots \\ \mathsf{succ(pred(x))} \rightarrow \mathsf{x}, \\ \mathsf{pred(succ(x))} \rightarrow \mathsf{x} \\ \vdots \end{array} \right\}$$

Dette avslører at $E_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ ikke er kjernebevarende.

Vår komplettering av $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ terminerer ikke, men gir også blant andre regler, reglene i R_b. Det er enkelt å overbevise seg om at ingen flere regler i $\mathcal{E}(\mathcal{T}_{\mathsf{Int}}(\mathcal{V}))$ vil bli generert, og R_b utgjør derfor et Int-manifest for $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$.

På måten vist i avsnitt 3.6.2 side 120, fastslår vi derfor at $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ er kjernebevarende.

Da er Int-manifestene til begge $E_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ og $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ identiske. Poenget er nå at de ved sekvensutvidet Knuth&Bendix-komplettering assosierte $E_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$-utledningssekvenser og $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$-utledningssekvenser også er identiske; til tross for at $E_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ og $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$ ikke er like mht. kjernebevaring. Se figur 4.2.
○

Det er mulig å etablere noe heuristikk og muligens andre syntaktiske «signaler» i utledninger, med hvilke det er mulig å fange inn flere tilfeller av kjernebevaring enn gjør teoremene 4.7 og 4.8.

Det er også mulig å bruke sekvens-utvidet Knuth&Bendix-komplettering til å si mer i tilknytning til eliminasjon av kunstig inkonsistens. Dette *kan* gjøres ved typing eller annen syntaktisk kontroll. Velges typing mistes, som i avsnitt 3.7.5, sambandet med *id*-utvidelser.

Vi kan også ved å studere *id*-utvidelse-utledningssekvenser gi kriterier som *garanterer* kjernebevaring av *id*-utvidelser (og ved reduksjon: initiell konsistens).

Av plasshensyn må en videre diskusjon om disse ting heller gjøres annensteds.

4.2.2 Skjuling av hjelpefunksjoner

Som nok en interessant anvendelse av sekvens-utvidet Knuth&Bendix-komplettering skal vi nå gi en skisseaktig begrunnelse for vår fremsetting av påstand 3.35

Figur 4.2: Figur til eksempel 87. Øverst vises $E_b \cup \{\mathsf{delta(x) = x}\}$-/$E_b' \cup \{\mathsf{delta(x) = x}\}$-utledningssekvensen fremkommet ved vår eksekvering av sekvensutvidet Knuth&Bendix-komplettering for regelen $\mathsf{succ(pred(x))} \longrightarrow \mathsf{x}$. Omskriving ved E_b/E_b' er merket med E_s. Nederst vises $E_b \cup \{\mathsf{delta(x) = x}\}$-/$E_b' \cup \{\mathsf{delta(x) = x}\}$-utledningssekvensen for $\mathsf{pred(succ(x))} \longrightarrow \mathsf{x}$.

på side 132. Vi gir på dette tidspunkt *på ingen måte* noe bevis for påstanden, kun en begrunnelse for spekulasjonen som motiverer den.

Begrunnelsesskisse for påstand 3.35: Vi skal bruke sekvens-utvidet Knuth-&Bendix-komplettering. La altså R_{RI} være resultatet av den vellykkete RI-restrikterte-sekvensutvidete Knuth&Bendix-komplettering gitt $\hat{E}^{id}$.

Det er lett å innse at hver regel i R_{RI} vil være element i $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$. Videre vil det til hver regel $v \to h$ i R_{RI} forefinnes en konstruert $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$-utledningsekvens $\langle v, \ldots, h\rangle$. Siden $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}} \subseteq \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}$, vil de konstruerte utledningsekvensene selvfølgelig også være $\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}$-utledningsekvenser.

Det er klart at R_{RI} er terminerende. Det er også klart at $\overset{*}{\underset{R_{RI}}{\leftrightarrow}} = \overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}$. For vilkårlige s,t slik at $s \overset{*}{\underset{R_{RI}}{\leftrightarrow}} t$ kan det da konstrueres en $\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}$-utledningssekvens $\langle s, \ldots, t\rangle$ (men som ikke nødvendigvis er en $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\leftrightarrow}}$-utledningssekvens).

Betrakt nå den operasjonelle restriksjon

$$\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$$

av $\overset{*}{\underset{R_{RI}}{\leftrightarrow}}$, som er som $\overset{*}{\underset{R_{RI}}{\leftrightarrow}}$, men med den restriksjon som følger av kravet om at de konstruerte utledningssekvensene forbundet med en $\overset{*}{\underset{R_{RI}}{\leftrightarrow}}$-omskrivning er $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$-utledningssekvenser. M.a.o.: $s \overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}} t$ hvis og bare hvis $s \overset{*}{\underset{R_{RI}}{\leftrightarrow}} t$, og den konstruerte $\overset{*}{\underset{\hat{E}^{id}}{\leftrightarrow}}$-utledningssekvens $\langle s, \ldots, t\rangle$ også er en $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$-utledningssekvens. Det synes nå mulig å vise at

$$\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}} = \overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}} \tag{4.1}$$

som er en av delpåstandene i påstand 3.35.

Vi skal nå skissere hvordan det muligens kan vises at $\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$ er konvergent. Kompletteringsprosessen er vellykket, og la oss for enkelhets skyld anta den er terminerende. I R_{RI} finnes da kun kritiske par som er hindret i å bli eliminert ved restriksjonen på regelen **Utled**. Dvs. de kritiske par som finnes er ikke med i $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$ og ved (4.1) ikke med i $\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$. Lemma 2.27 ("Critical Pair Lemma") side 53 kan forsterkes (se beviset for lemmaet i [Kir94]) til å gi lokal konfluens for $\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$. Videre ser det ut til at lemma 2.25 (Newman) og lemma 2.26 kan forsterkes til å gi global konfluens og til sist konvergens (sees ved å studere bevisene for disse lemmata i [Kir94]). Fra dette vises så at

$$s \overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}} t \;\Leftrightarrow\; s!\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}} = t!\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$$

Ved at $\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$ er konvergent og ved den gitte reduksjonsordning, kan det nå argumenteres for de to siste del-påstandene om eksistensen av en $R'_{RI} \subseteq \mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$ slik at $\overset{*}{\underset{R'_{RI}}{\leftrightarrow}} = \overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\leftrightarrow}}_{\mathcal{T}_{\Sigma^c}(\mathcal{V})}$.
◇

Det er mulig å sjekke mekanisk om en sekvens er en $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$-sekvens. Omskriving i $\overset{*}{\underset{R_{RI}\hat{E}^{id}}{\rightsquigarrow}}$ er derfor mekaniserbar.

Merk dessuten at kriteriet $v_j \to h_j \in_{E_0 \setminus \{\overset{*}{\underset{s(x)=x}{\leftrightarrow}}\}}$ i inferensregelen **RI-Utled** lett kan sjekkes gitt den assosierte E_0- (her $\hat{E}^{id}$-) utledningssekvensen for $v_j \to h_j$ konstruert ved sekvens-utvidet komplettering.

Et apropos til eksempel 87 side 157: Påstand 3.35 hevder at en vellykket RI-restriktert komplettering gir et kjernemanifest for $\hat{E}^{id}$ i henhold til $\overset{\Sigma^*\S}{\underset{\hat{E}^{id}}{\rightsquigarrow}}$. Slike kjernemanifest vil for id-utvidelsene av E_b og E'_b i eksempel 87 selvfølgelig ikke lenger være identiske! (Men merk at regelen $\mathsf{pred(succ(x))} \to \mathsf{x}$ ikke vil bli generert for id-utvidelsen av E'_b, siden regelens assosierte $E'_b \cup \{\mathsf{delta(x)} = \mathsf{x}\}$-utledningssekvens ikke kan være en $\overset{\mathsf{Int}*\mathsf{delta}}{\underset{E'_b\,id}{\rightsquigarrow}}$-utledningssekvens. Se også figur 4.2. Ved (4.1) må likevel regler mellom grunntermer sannsynligvis på formen

$$\mathsf{pred(succ(0))} \to \mathsf{0},\;\; \mathsf{pred(succ(succ(0)))} \to \mathsf{succ(0)} \;\; \text{o.l.}$$

genereres (i det uendelige).)

4.2.3 Basis-initialsemantikk og induktive konsekvenser

Anta for en gitt ligningsmengde E en vellykket sekvens-utvidet Knuth&Bendix-prosess resulterende i en regelmengde R komplett (innebærer konvergent) for E. Gitt at $s = t$ er en logisk og dermed induktiv konsekvens av E, finnes altså et omskrivningsbevis $s \xrightarrow[R]{!} \circ \xleftarrow[R]{!} t$ i R for dette faktum. Ved vår utvidelse, kan så et konstruktivt E-bevis i form av et syntaktisk objekt — en utledningssekvens — konstrueres fra de lagrede E-utledninger for hver regel i R.

For induktive konsekvenser som *ikke også* er logiske konsekvenser, er saken ikke så grei. Anta vi bruker induktiv komplettering. La R være en for E komplett regelmengde. En ligning $s = t$ er en induktiv konsekvens av E bl.a. dersom en Knuth&Bendix-prosess gitt $\langle \{s = t\}, R\rangle$ terminerer vellykket, og alle regler generert underveis har av R grunnredusible venstresider. Så anta at $s = t$ således er vist å være en induktiv konsekvens av E, og la R' være sluttproduktet av prosessen. Vi antar at sekvens-utvidet Knuth&Bendix-komplettering er brukt i forbindelse med den induktive komplettering.

Dersom regelen $s \rightarrow t$ eller $t \rightarrow s$ finnes i R', kan vi konstruere et ligningsbevis som over. Imidlertid behøver ikke $s \rightarrow t$ eller $t \rightarrow s$ finnes i R'. Men det må nødvendigvis finnes et omskrivningsbevis $s \xrightarrow[R']{!} \circ \xleftarrow[R']{!} t$ i R'; så igjen kan et korresponderende ligningsbevis konstrueres.

Saken er imidlertid at de konstruerte bevisene selvfølgelig ikke er E-bevis. De er $E \cup \{s = t\}$-bevis, og ikke en eller annen form for induktive bevis i E for at $s = t$ er en induktiv konsekvens av E. Slike $E \cup \{s = t\}$-bevis er i dette lys følgelig ikke så interessante.

Dersom det under induktiv komplettering dukker opp en regel med en av R ikke grunnredusibel venstreside, er $s = t$ *ikke* en induktiv konsekvens av E. Denne regel må nødvendigvis ha en anvendelse av $s = t$ i sin tilknyttede $E \cup \{s = t\}$-utledningssekvens. Ved *refutering* av en hypotese som induktiv konsekvens kan derved et konstruktivt E-motbevis konstrueres:

La oss anta en signatur Σ disjunkt delt i generatorer Σ^c og definerte funksjonssymboler Σ^d, og anta at

1. R er tilstrekkelig Σ^d-komplett mhp. $\mathcal{G}_{\Sigma^c}$.
2. term-ordningen i kompletteringsprosessen er slik at enhver $u \in \mathcal{T}_{\Sigma^d \cup \Sigma^c}(\mathcal{V}) \setminus \mathcal{T}_{\Sigma^c}(\mathcal{V})$ er større i ordningen enn alle $v \in \mathcal{T}_{\Sigma^c}(\mathcal{V})$.

La $v \rightarrow h$ være en regel generert ved vår induktive komplettering, slik at v ikke er grunnredusibel i R. Dvs. det finnes en $\sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$ slik at $v\sigma!R = v\sigma$.

På side 57 i avsnitt 2.4.5, nevnte vi at *generatortermene* $v\sigma$ og $h\sigma$ da vil være i hver sin E-kongruensklasse. Vi har altså

$$v\sigma \xleftrightarrow[E \cup \{s=t\}]{*} h\sigma \quad \text{og} \quad v\sigma \not\xleftrightarrow[E]{*} h\sigma$$

Vi kan ved sekvensutvidet Knuth&Bendix-/induktiv komplettering konstruere en $E \cup \{s = t\}$-utledningssekvens $\langle v\sigma, \ldots, h\sigma\rangle$. En slik utledningssekvens gir en konstruktiv *demonstrasjon* på at $s = t$ gir inkonsistens, og derfor ikke kan være en induktiv konsekvens av E. Dette gir et konstruktivt $E \cup \{s = t\}$-bevis for at $s = t$ ikke er en induktiv konsekvens av E.

Et rent E-motbevis kan konstrueres slik: At $s = t$ ikke er en induktiv konsekvens av E, medfører jo at $s = t$ heller ikke er en logisk konsekvens av E. Et konstruktivt E-bevis for at $s = t$ ikke er en logisk konsekvens av E kan da konstrueres ved hjelp av omskrivingen

$$s \xrightarrow[R]{!} s! \neq t! \xleftarrow[R]{!} t$$

Et konstruktivt E-bevis for at $s = t$ ikke er en *induktiv* konsekvens av E kan så skaffes ved omskrivingen

$$s\tau \xrightarrow[R]{!} g \neq g' \xleftarrow[R]{!} t\tau$$

for en $\tau \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$. Det er algoritmisk å finne en slik τ. (Spesielt er den σ over som vitner ikke-grunnredusibilitet ved $v\sigma!R = v\sigma$, også slik at $s\sigma \xrightarrow[R]{!} g \neq g' \xleftarrow[R]{!} t\sigma$. Anta nemlig det motsatte. Omskrivingen $v\sigma_{E\cup\overleftrightarrow{\{s=t\}}}h\sigma$ må nødvendigvis anvende ligningen $s = t$. Dvs. det finnes et enkeltsteg på formen $\langle \ldots, c[s\sigma], c[t\sigma], \ldots\rangle$ i enhver utledning $\langle v\sigma, \ldots, h\sigma\rangle$. Dersom $s\sigma \overset{*}{\underset{E}{\leftrightarrow}} t\sigma$, ville vi også ha $c[s\sigma] \overset{*}{\underset{E}{\leftrightarrow}} c[t\sigma]$. Slik kunne ethvert omskrivningssteg ved $s = t$ erstattes ved et E-steg, og vi ville ha $v\sigma \overset{*}{\underset{E}{\leftrightarrow}} h\sigma$; som er en motsigelse.)

Slike konstruktive (mot)bevis kan gi større innsikt i såvel den induktive teorien som den logiske teorien for en ligningsmengde E. Konstruktive motbevis demonstrerer i en viss forstand *hvordan/hvorfor* en ligning ikke er en logisk eller induktiv konsekvens. I implementasjonssammenheng kan dette så brukes i streben mot en mer korrekt eller fullstendig implementasjon av den aktuelle abstrakte datatype. Se avsnitt 2.2.9 side 21.

4.2.4 Basis-initialsemantikk og konsistens

I avsnitt 2.4.7 fastslo vi at inkonsistens er algoritmisk oppdagbart gitt konvergens og avgjørbarhet av kjernesemantikk. Vi nevnte også et tilfelle der *konsistens* kan etableres. I tillegg kan Knuth&Bendix-komplettering av og til brukes for å etablere konsistens:

Sats 4.11 *Anta en signatur Σ disjunkt delt i generatorer Σ^c og definerte funksjonssymboler Σ^d. La R være resultatet av en vellykket Knuth&Bendix-komplettering gitt en ligningsmengde E, under antagelse av punkt 2 over.*

Observasjon 3.26 på side 112 gir en $R' \subseteq R$ slik at $R' \subseteq \mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$, og slik at $\overset{*}{\underset{R'}{\leftrightarrow}}\mathcal{T}_{\Sigma^c}(\mathcal{V}) = \overset{*}{\underset{E}{\leftrightarrow}}\mathcal{T}_{\Sigma^c}(\mathcal{V})$. La $E^c = E \cap \mathcal{E}(\mathcal{T}_{\Sigma^c}(\mathcal{V}))$. Da er E basis-initielt konsistent hvis hver $v \to h \in R'$ er slik at $v \overset{*}{\underset{E^c}{\leftrightarrow}} h$.

Bevis: Vi må vise at $\overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_{\Sigma^c} = \overset{*}{\underset{E^c}{\longleftrightarrow}}\mathcal{G}_{\Sigma^c}$. At $\overset{*}{\underset{E}{\leftrightarrow}}\mathcal{G}_{\Sigma^c} \supseteq \overset{*}{\underset{E^c}{\longleftrightarrow}}\mathcal{G}_{\Sigma^c}$ er opplagt. Anta $g_c \overset{*}{\underset{E}{\leftrightarrow}} g'_c$ for vilkårlige $g_c, g'_c \in \mathcal{G}_{\Sigma^c}$. Vi har da $g_c \overset{*}{\underset{R'}{\leftrightarrow}} g'_c$. Betrakt en vilkårlig R'-utledning $\langle g_c, \ldots, g'_c\rangle$, og betrakt to nabo-komponenter g_i, g_{i+1} i denne. Vi har $g_i = g_i[v\sigma]_p$ og $g_i[h\sigma]_p = g_{i+1}$ for en $v \to h$ i R'. Pr. antagelse har vi $v \overset{*}{\underset{E^c}{\leftrightarrow}} h$ og derfor $v\sigma \overset{*}{\underset{E^c}{\leftrightarrow}} h\sigma$. Videre har vi selvfølgelig $g_i[v\sigma]_p \overset{*}{\underset{E^c}{\leftrightarrow}} g_i[h\sigma]_p$. Således fins en E^c-utledning $\langle g_c, \ldots, g'_c\rangle'$. Det følger at E er basis-initielt konsistent.
□

Kriteriet i sats 4.11 om at hver $v \to h \in R'$ er slik at $v \overset{*}{\underset{E^c}{\leftrightarrow}} h$, kan sjekkes algoritmisk dersom $\overset{*}{\underset{E^c}{\leftrightarrow}}$ er avgjørbar (f.eks. ved konvergens). Dersom E^c her ikke er avgjørbar, kan sekvensutvidet Knuth&Bendix-komplettering være til hjelp for etablering av konsistens. F.eks. er E konsistent dersom den til hver regel i R' assosierte E-utledningssekvens er en E^c-utledningssekvens. Sistnevnte kan synliggjøres ved bruk av sekvensutvidet Knuth&Bendix-komplettering.

Mer sofistikerte måter å bruke sekvensutvidet Knuth&Bendix-komplettering på i denne forbindelse, er mulige, men vi følger ikke dette opp her.

Kapittel 5

Konklusjon og videre arbeid

Hovedlinjen/grunnstrukturen i denne hovedoppgaven har vært

1) å utvikle et *generalisert semantikk*-begrep, for så å anvende dette i tilknytning til

2) å spesifisere semantikk på en ny måte; ved *indirekte* algebraisk spesifikasjon.

På dette rammeverket har vi så lagt vår diskusjon. Vårt generaliserte semantikkbegrep tillater spesifisering av semantikk relativ til en *kjerne*-semantikk. Vi har kunnet utvikle relevante begreper for slik semantikk uten viten om hvordan kjernen er spesifisert. Dette åpner for *modulær* oppbygging av formelle datatyper.

En naturlig anvendelse av generalisert semantikk uttrykkes i intensjonen om at kjernesemantikk er semantikk til generatorer og den relative del av semantikken er semantikk til definerte funksjonssymboler. Det at vi kan spesifisere og resonnere om semantikk uten å vite hvordan kjernesemantikken er spesifisert, er da en imøtekomming av at det finnes ulike teknikker for å spesifisere generatorsemantikk.

Generalisert semantikk åpner også for modulær oppbygging av spesifikasjoner og formelle datatyper i flere nivåer utover en to-nivå oppbygning der kjernen direkte er *atomær*. Dette har i seg selv ikke vært vårt hovedtema, og vi har heller ikke tatt stilling til i hvilken grad en slik flernivå spesifikasjonsstil i formelle datatyper er fornuftig, eller i hvilken grad og på hvilken måte slik flernivå spesifikasjon svarer til oppbygging av moduler i (konkrete) programmeringsspråk. Vi har antydet at disse og tilknyttede spørsmål er interessante og bør kunne utredes videre.

Tradisjonell spesifikasjon ved ligninger og ligningslogikk involverer én (homogen) ligningsmengde. Dette utgjør spesialtilfellet *basis-semantikk* av våre generelle semantikker. En egenskap ved vår generelle semantikkspesifikasjon som for hovedlinjen i vår diskusjon har vært vesentlig, er den at generell semantikk uttrykker spesifikasjon, der deler av semantikken ikke er spesifisert som del av en slik homogen ligningsmengde. Denne del isoleres da i kjernen. Denne egenskap har vi brukt for å «plugge inn» kjernesemantikk spesifisert på en ny måte; nemlig *indirekte algebraisk*.

Grunnlaget for indirekte algebraisk spesifikasjon har vært *semantikkgivende syntaktiske funksjoner* eller det ekvivalente begrep *klasserepresentant funksjoner*. Av disse har vi først og fremst betraktet *kanonisk-representant funksjoner*, men vi har også betraktet andre semantikkgivende syntaktiske funksjoner. Våre eksempler på indirekte spesifikasjoner har stort sett vært algebraiske beskrivelser av slike funksjoner. *Indirekte kongruens* er en forutsetning for indirekte algebraisk spesifikasjon, og vi har sett eksempler på at indirekte kongruens ver-

ken er umiddelbart tilfredstilt eller nødvendigvis er lett å oppnå. Vi har som et apropos påpekt at et lignende kongruenskrav også eksplisitt må stilles for spesifikasjon ved observator-basis, og at et slikt kongruenskrav ikke trivielt er oppfylt. Indirekte semantikk egner seg som atomær semantikk.

Semantikk spesifisert relativ til en indirekte kjerne er et (ekte) tilfelle der basis-semantikk ikke er tilstrekkelig, og der en generalisert form for semantikk-spesifikasjon må brukes. Men med det sagt, har vi vist at den heterogene situasjonen som semantikk relativ til indirekte kjerne i utgangspunktet representerer, under visse interessante omstendigheter kan *reduseres* til basis-semantikker. For final-semantikk kan dette gjøres mer eller mindre direkte. For initial-semantikk har vi gjort reduksjonen ved å hjelp av *id-utvidelser*.

Slik reduksjon er foreløpig essensiell: En hovedhensikt ved formell spesifikasjon og formelle datatyper er i formell og endog mekanisk resonnering om programmer. Resolusjonsmetoder eksisterer for basis-semantikker i basis-formelle datatyper, men vi har foreløpig ikke utviklet resolusjonsmetoder for våre generaliserte semantikker.

Å utvikle resolusjonsmetoder for generalisert semantikk er en interessant oppgave. Slike resolusjonsmetoder vil være spesielt hensiktsmessige dersom det viser seg at en flernivå spesifikasjonsstil i formelle datatyper som nevnt over, er fornuftig. Vi har antydet at vår modulære oppbygging av semantikker og formelle datatyper ikke umiddelbart kan overføres til en modulær oppbygging av korresponderende resolusjonsmetoder. Vi har dog nevnt at en mulig innfallsvinkel til å finne resolusjonsmetoder for generalisert semantikk kan være gjennom klasse- og utvidet omskriving. *Refutering* er under konvergens algoritmisk for basis-semantikker.

Et vesentlig gjennomgangstema har vært *konsistens*. Konsistens må alltid sees relativt til *forutfatninger* i semantikkspesifikasjon. *In*konsistens utgjør da en intern motsigelse i spesifikasjonen i forhold til en slik forutfatning. Vi har sammen med generaliseringen av semantikk, utviklet tilhørende konsistensbegrep. Våre konsistensbegrep generaliserer konsistensbegrepet i predikatlogikk, hvor forutfatningen uttrykkes ved predefinert tolkning av symbolene true og false, og der inkonsistens er ekvivalent med at $\mathsf{true} = \mathsf{false}$ er bevisbart. I vår kontekst ivaretas forutfatninger ved håndheving av *kjerne-bevaring*. Intensjonen om at kjernesemantikk skal være semantikk til generatorer og den relative del av semantikken skal være semantikk til definerte funksjonssymboler, ivaretas således ved konsistens. Sentralt i det mer stringente argumentet for dette, var det vi har kalt *separabel* semantikk.

I forbindelse med indirekte spesifikasjon, har vi innført begrepet *kunstig inkonsistens*. Dette er inkonsistens som skyldes hjelpefunksjoner og som vi har argumentert godt kan skjules fra logikken, siden de sannsynligvis ikke hører hjemme i den semantiske abstrakte datatypen som søkes implementert. Vi har vist måter å eliminere kunstig inkonsistens fra logikken. Dette har vi bl.a. gjort ved å legge en *operasjonell restriksjon* inn i ligningslogikk. Vi har argumentert at en ny frihetsgrad i semantikkspesifikasjon oppnås, dersom det også lykkes å skjule hjelpefunksjoner mhp. kunstig inkonsistens i resolusjonsmetoder. Vi fremsatte en påstand om at dette er mulig for kompletteringsbaserte resolusjonsmetoder. En utfordring til videre arbeid ligger da i å bevise eller motbevise påstanden. Teorien utviklet i avsnitt 3.7, samt sekvens-utvidet komplettering i kapittel 4 kan i denne forbindelse være aktuell. En implementasjon av resolusjonsmetoder for basis-semantikk, f.eks. induktiv komplettering, der skjuling er innlemmet burde være meget interessant.

Sentralt i teorien utviklet i forbindelse med skjuling av hjelpefunksjoner er igjen separabel semantikk som nå tillukkes mhp. monotonitet. Vi har funnet at separabel semantikk sammen med variable monotone tillukninger er kraftige

uttrykksredskap. Disse har vært sentrale i all teorien vi har utviklet omkring konsistens og kjernebevaring.

Vi har videre vurdert muligheter for å *etablere konsistens* og for å *oppdage inkonsistens*. Under visse antagelser er inkonsistens algoritmisk oppdagbart ved prinsippet om systematisk søk. Vi har nevnt et tilfelle for basis-initialsemantikk der konvergens umiddelbart fører til konsistens (sats 2.34 side 60). Vi har dessuten knyttet vårt konsistensbegrep til *kongruens* i den separable semantikk.

Kjernebevaring i *id*-utvidelser kan ved reduksjon overføres til initiell konsistens. Det viser seg at en vellykket Knuth&Bendix-prosess gitt en *id*-utvidelse, gir et *kjerne-manifest* for *id*-utvidelsen. Et slikt kjerne-manifest kan brukes for å oppdage mangel på kjernebevaring for *id*-utvidelser av indirekte spesifikasjoner og dessuten for å *synliggjøre* kjernebevaring, gitt positiv kjennskap om hvilken kjernesemantikk den indirekte spesifikasjonen spesifiserer. Vi har også nevnt et tilfelle der Knuth&Bendix-komplettering kan brukes for å etablere konsistens i basis-initial semantikk (sats 4.11 side 162).

Med konsistens etablert, kan separabel semantikk betraktes i stedet for initial- og det vi kalte degenerert finalsemantikk. Separabel semantikk *kan* være en åpning til modulære sammensetninger av resolusjonsmetoder. Vi har også vist at relasjonene $\Re^s$ — en umiddelbar avledning av ligningslogikk, og $\overset{*}{\underset{E}{\rightsquigarrow}}$ — en enkel restriksjon på ligningslogikk, utrykker separabel semantikk. Under konsistens er de således mulige erstatninger for *id*-utvidelser i tilnærminger til eksisterende resolusjonsmetoder.

Kjerne-manifest for *id*-utvidelser av indirekte spesifikasjoner er, under kjernebevaring, ekvivalente *direkte* spesifikasjoner. En slik direkte spesifikasjon kan brukes for å *verifisere* en indirekte spesifikasjon. Dette er en form for *programtransformasjon*. Knuth&Bendix-komplettering som program-transformator fordrer flere noe restriktive forutsetninger. Andre metoder for å generere direkte spesifikasjoner fra indirekte spesifikasjoner ville derfor være av stor interesse. En lettelse i restriksjonene her forbundet med komplettering oppnås, dersom skjuling av hjelpefunksjonssymboler i den indirekte algebraiske spesifikasjon kan bygges inn i Knuth&Bendix-komplettering. Dette henger sammen med problemstillingen over om skjuling av hjelpefunksjoner for kompletteringsbaserte resolusjonsmetoder.

Vår indirekte spesifikasjon har tatt utgangspunkt i algebraiske beskrivelser av semantikkgivende syntaktiske funksjoner. Dette har vist seg å gi indirekte spesifikasjoner en *operasjonell* karakter. Dette fjerner oss en smule fra det abstraksjonsnivået vi i utgangspunktet søker for formell resonnering. Den operasjonelle stilen gjør indirekte spesifikasjoner tilgjengelige for programverifikasjons-metoder. Vi har antydet at indirekte spesifikasjon kanskje i noen tilfeller er enklere enn direkte spesifikasjon. Det gjenstår imidlertid en reell analyse av *pro et contra* for indirekte spesifikasjon i forhold til direkte og annen spesifikasjon. En slik analyse, føler vi, må komme først når et større erfaringsgrunnlag i bruken av indirekte spesifikasjon er tilstede. Vi har i denne rapporten lagt et teoretisk grunnlag for nettopp bruken av indirekte spesifikasjon.

Vi har sett at indirekte spesifikasjon utvider klassen av semantikker som er avgjørbare ved termomskriving i enkleste form. Dette har direkte konsekvenser for algoritmiske metoder som bygger på avgjørbarhet av semantikk. Eksempelvis så vi for tilfellet 'mengder av naturlig tall' at en naïv metode for algoritmisk oppdagbarhet av inkonsistens appliserer dersom kjernen spesifiseres indirekte i motsetning til direkte.

Om man kan spesifisere andre semantikker med indirekte spesifikasjon enn med direkte spesifikasjon, er uklart. Det er også uklart om algebraiske beskrivelser av *generelle* semantikkgivende syntaktiske funksjoner utvider klassen av

semantikker som kan spesifiseres indirekte ved algebraiske beskrivelser av kanonisk-representant funksjoner. Disse er høyst interessante spørsmål.

Det kunne være interessant å se om indirekte spesifikasjon kan anvendes også på termer med variable. Dette ville utvide anvendelsen av semantikkgivende syntaktiske funksjoner til å omfatte f.eks. logiske teorier i tillegg til induktive teorier. Kanskje kan indirekte spesifikasjon dermed også bidra med økt avgjørbarhet i forbindelse med klasseomskrivning. (Jeg har såvidt forsøkt å spesifisere semantikkgivende syntaktiske funksjoner på termer med variable med sikte på å spesifisere en interessant teori. Det virker ikke særlig enkelt.)

Et annet «rand»-eksperiment var *fikspunkt-semantikk* med *semantikkantydende* funksjoner. Vi så at ligningsmengder som ikke er indirekte kongruente, kan være algebraiske beskrivelser av semantikkantydende funksjoner, og kan da spesifisere semantikk ved en slags *fikspunkt-omskriving*.

Vi har også spekulert om noe av teorien om *id*-utvidelser – alene og i forbindelse med Knuth&Bendix-komplettering — kan inspirere til resultater for final-semantikk.

I forbindelse med *konstruktive bevis* har vi definert en enkel utvidelse av Knuth&Bendix-komplettering, som vi har kalt *sekvensutvidet Knuth&Bendix-komplettering*. Konstuktive bevis og sekvensutvidet Knuth&Bendix-komplettering, har mange anvendelsesområder. Vi har sett at sekvensutvidet Knuth&Bendix-komplettering har vist seg å være interessant i forbindelse med oppdaging og etablering av inkonsistens og kjernebevaring i forbindelse med *id*-utvidelser. Spesielt interessant er kanskje det faktum at syntaktiske kjennetegn i de konstruktive bevis for ligninger i kjerne-manifest for *id*-utvidelser kan bidra til å etablere kjernebevaring for *id*-utvidelser (og konsistens ved reduksjon). Kjernebevaring etableres her *uten* positiv kjennskap om hvilken kjernesemantikk den indirekte spesifikasjonen spesifiserer. Dette i motsetning til synliggjøring av kjernebevaring for *id*-utvidelser ved vanlig Knuth&Bendix-komplettering som over. På grunnlag av sådant etablert kjernebevaring, kan på den annen side positiv kjennskap om hvilken kjernesemantikk den indirekte spesifikasjonen spesifiserer, nå etableres via kjerne-manifest.

Vi har også sett litt på anvendelser av sekvensutvidet Knuth&Bendix-komplettering tilknyttet basis-initialsemantikk. Konstruktive *mot*-bevis kan gi innsikt i hvorfor/hvordan en ligning ikke er en logisk eller induktiv konsekvens. I implementasjonssammenheng kan dette så brukes på veien mot en mer korrekt eller fullstendig implementasjon av den aktuelle abstrakte datatype. Sekvensutvidet Knuth&Bendix-komplettering kan dessuten være til hjelp ved etablering av basis-initiell konsistens.

Mye mer kan sies om anvendelser av sekvensutvidet Knuth&Bendix-komplettering på *id*-utvidelser og på basis-initialsemantikk samt om andre anvendelser. Raffinementer av sekvensutvidet Knuth&Bendix-komplettering eller andre bedre måter å generere konstruktive bevis på, kan også være en interessant oppgave.

*

Vår diskusjon har altså utviklet seg rundt rammeverket beskrevet av de to punktene i begynnelsen av dette kapitlet. Den teorien som har vært utviklet underveis er imidlertid grunnlag for en rekke andre diskusjoner — noen har vi tatt tid til her; andre diskusjoner, og muligens svært interessante sådane, har vi av plass- og tidshensyn ikke kunnet annet enn antyde.

Takk for oppmerksomheten.

Tillegg A

Surjektivitet og grunnterm-algebraer

A.1 Mer om formelle datatyper og semantikk

Dette avsnittet er ment å utdype noen av de for oss sentrale begreper i algebra. Mye i dette avsnittet har ingen direkte relevans til den øvrige diskusjon.

A.1.1 Bevis av (2.2) side 17

Vi viser her mer presist (2.2) på side 17. Vi bruker følgende lemma [Lys92]:

Lemma A.1 *La A og B være to Σ-algebraer slik at det fins en surjektiv homomorfi κ fra A til B. Da kan enhver homomorfi ϕ fra $\mathcal{T}_\Sigma(\mathcal{V})$ til B, utrykkes ved sammensetningen $\rho \circ \kappa$, hvor ρ er en homomorfi fra $\mathcal{T}_\Sigma(\mathcal{V})$ til A.*

Bevis: Betrakt en vilkårlig homomorfi ϕ fra $\mathcal{T}_\Sigma(\mathcal{V})$ til B, og betrakt $\phi(x)$ for en vilkårlig variabel x. Siden κ er surjektiv, fins det en $a \in A$ slik at $\kappa(a) = \phi(x)$. La ρ være homomorfien slik at $\rho(x) = a$. Således er ρ entydig bestemt av verdiene den på denne måten tilordner hver variabel. Vi har altså $\kappa(\rho(x)) = \phi(x)$. Sammensetningen av homomorfier er en homomorfi, så homomorfien $\rho \circ \kappa$ er entydig bestemt på samme måte som ϕ. Følgelig må $\rho \circ \kappa$ og ϕ være samme homomorfi.
□

Vi kan umiddelbart bekrefte (2.2) på side 17. Påstanden $A \models s = t$ betyr jo $\forall \phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})} \mid \phi(s) = \phi(t)$. Betrakt en slik ϕ. Vi har antatt en surjektiv homomorfi $\phi^A_{\mathcal{G}_\Sigma}$ fra $\mathcal{G}_\Sigma$ til A. Lemma A.1 sier da at $\phi = \sigma \circ \phi^A_{\mathcal{G}_\Sigma}$ for en homomorfi (substitusjon) $\sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma}$. Altså er $\forall \phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})} \mid \phi(s) = \phi(t)$ samme påstand som $\forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid \phi^A_{\mathcal{G}_\Sigma}(s\sigma) = \phi^A_{\mathcal{G}_\Sigma}(t\sigma)$, siden $\phi^A_{\mathcal{G}_\Sigma}$ er unik. Men dette betyr jo $\forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid A \models s\sigma = t\sigma$, ved (2.1) på side 16.

A.1.2 Semantikk, initialalgebra og finalalgebra

Dette avsnittet bør leses i forbindelse med avsnitt 2.3.

La $\simeq$ være en semantikk på $\mathcal{G}_\Sigma$. Betrakt klassen av Σ-algebraer

$$\lfloor \simeq \rfloor_\Sigma = \{A \mid g \simeq g' \;\Rightarrow\; A \models g = g'\}$$

La oss si at algebraene i $\lfloor\simeq\rfloor_\Sigma$, ***realiserer*** semantikken gitt av $\simeq$. Algebraene i klassen av Σ-algebraer

$$\begin{aligned}\lceil\simeq\rceil_\Sigma &= \{A \mid g \simeq g' \Leftarrow A \models g = g'\}\\ &= \{A \mid g \not\simeq g' \Rightarrow A \not\models g = g'\}\end{aligned}$$

kan vi på den annen side si ***begrenser*** semantikken gitt av $\simeq$.

Betrakt restriksjonen $\lceil\simeq\rceil_\Sigma^{sur}$ av $\lceil\simeq\rceil_\Sigma$ til de algebraer som er homomorfe bilder av $\mathcal{G}_\Sigma$. Vi skal vise at $\mathcal{G}_\Sigma/\simeq$ er initiell i $\lfloor\simeq\rfloor_\Sigma$ og final i $\lceil\simeq\rceil_\Sigma^{sur}$. Altså

Sats A.2 *Vi har*

1. $\mathcal{G}_\Sigma/\simeq$ *er initiell i* $\lfloor\simeq\rfloor_\Sigma$
2. $\mathcal{G}_\Sigma/\simeq$ *er final i* $\lceil\simeq\rceil_\Sigma^{sur}$

Godtgjøring av punkt 1 i sats A.2 fordrer at vi viser at $\mathcal{G}_\Sigma/\simeq \in \lfloor\simeq\rfloor_\Sigma$, og at det finnes en unik homomorfi fra $\mathcal{G}_\Sigma/\simeq$ til alle A i $\lfloor\simeq\rfloor_\Sigma$. At $\mathcal{G}_\Sigma/\simeq \in \lfloor\simeq\rfloor_\Sigma$ er innlysende siden (2.6) på side 23 gir $\mathcal{G}_\Sigma/\simeq \models g = g' \Leftrightarrow g \simeq g'$.

Vi skal vise eksistensen av en unik homomorfi ved hjelp av induktiv implikasjon. Vi trenger først et generelt resultat:

Lemma A.3 *La A, B, C være tre Σ-algebraer. La ρ være en homomorfi fra A til B og φ en avbildning fra B til C slik at sammensetningen $\rho\circ\varphi$ er en homomorfi. La φ' være φ restriktert til $\rho(A)\times C$ (funksjoner er relasjoner), der $\rho(A) \subseteq B$ er bildet av A under ρ. Da er φ' en homomorfi fra algebraen $\rho(A) = \langle\rho(A), F'_B\rangle$ til C, der F'_B består av alle funksjoner i funksjonsmengden F_B til B men restriktert til $\rho(A)\times\rho(A)$, og slik at hver funksjon i F'_B er en tolkning av et funksjonssymbol i Σ.*

Bevis: Først viser vi at algebraen $\rho(A)$ er en Σ-algebra. Det holder å vise at kodomenet til hver $f'_B \in F'_B$ er inkludert i $\rho(A)$, m.a.o. at restriksjonen f'_B av f_B finnes. Betrakt derfor $f'_B(b_1,\ldots,b_n)$ for en vilkårlig $f'_B \in F'_B$ og vilkårlig $\langle b_1,\ldots,b_n\rangle$ i domenet til f'_B. Domenet til f'_B er inkludert i $\rho(A)$, så det finnes $a_i \in A$ slik at $\rho(a_i) = b_i; 1 \le i \le n$. Pr. antagelse finnes et funksjonsymbol f i Σ av hvilket f'_B er en tolkning. La f_A være tolkningen av f i A. Vi har $f'_B(b_1,\ldots,b_n) = f'_B(\rho(a_1),\ldots,\rho(a_n)) = \rho(f_A(a_1,\ldots,a_n))$ som viser at $f'_B(b_1,\ldots,b_n) \in \rho(A)$.

For å vise at φ' er en homomorfi, betrakt $\varphi'(f'_B(b_1,\ldots,b_n))$ for en vilkårlig $f'_B \in F'_B$. Siden domenet til f'_B er inkludert i $\rho(A)$, fins det $a_i \in A$ slik at $\rho(a_i) = b_i$ og det gir dessuten mening å snakke om $\varphi'(b_i); 1 \le i \le n$. Merk at $\varphi'(b_i) = \varphi(b_i); 1 \le i \le n$. Vi har altså $\varphi(\rho(a_i)) = \varphi(b_i); 1 \le i \le n$. Siden $\rho\circ\varphi$ er en homomorfi får vi da

$$\begin{aligned}\varphi(\rho(f_A(a_1,\ldots,a_n))) &= f_C(\varphi(\rho(a_1),\ldots,\varphi(\rho(a_n))\\ &= f_C(\varphi(b_1),\ldots,\varphi(b_n))\\ &= f_C(\varphi'(b_1),\ldots,\varphi'(b_n))\end{aligned}$$

for tolkninger f_A og f_C av funksjonssymbolet av hvilket f'_B er en tolkning. Men vi har også

$$\begin{aligned}\varphi(\rho(f_A(a_1,\ldots,a_n))) &= \varphi'(\rho(f_A(a_1,\ldots,a_n)))\\ &= \varphi'(f'_B(\rho(a_1),\ldots,\rho(a_n)))\\ &= \varphi'(f'_B(b_1,\ldots,b_n))\end{aligned}$$

Altså har vi

$$\varphi'(f'_B(b_1,\ldots,b_n)) = f_C(\varphi'(b_1),\ldots,\varphi'(b_n))$$

□

Sats A.4 *La A og B være to Σ-algebraer slik at den unike homomorfien fra $\mathcal{G}_\Sigma$ til A er surjektiv, og slik at B er induktivt implisert av A. Da finnes en unik homomorfi fra A til B.*

Bevis: La κ være den surjektive homomorfi fra $\mathcal{G}_\Sigma$ til A og ϕ den unike homomorfi fra $\mathcal{G}_\Sigma$ til B. Betrakt et vilkårlig element $a \in A$. Siden κ er surjektiv, fins minst en $g \in \mathcal{G}_\Sigma$ slik at $\kappa(g) = a$. Velg en slik g og kall den g_a. For alle $a \in A$ og på denne måte assosierte g_a slik at $\kappa(g_a) = a$, definerer vi så φ fra A til B slik at $\varphi(a) = \phi(g_a)$.

Vi viser så at $\kappa \circ \varphi = \phi$. For en vilkårlig $g \in \mathcal{G}_\Sigma$ betrakt $\varphi(\kappa(g))$. Nå er $\kappa(g) = a$ for en $a \in A$. Videre fins en g_a slik at $\varphi(a) = \phi(g_a)$ og slik at $\kappa(g_a) = a$. Men da har vi $A \models g = g_a$. Siden B er induktivt implisert av A, har vi $B \models g = g_a$ og altså $\phi(g) = \phi(g_a)$. Altså har vi $\varphi(\kappa(g)) = \varphi(a) = \phi(g_a) = \phi(g)$.

Da er $\kappa \circ \varphi$ lik homomorfien ϕ, så ifølge lemma A.3 er φ en homomorfi fra A til B. Homomorfien $\kappa \circ \varphi$ er unik fra $\mathcal{G}_\Sigma$ til B (ϕ er unik). Siden κ er unik fra $\mathcal{G}_\Sigma$ til A, må φ være unik fra A til B.
□

Det er lett å se at homomorfien fra $\mathcal{G}_\Sigma$ til $\mathcal{G}_\Sigma/\simeq$ er surjektiv, så det følger fra sats A.4 at $\mathcal{G}_\Sigma/\simeq$ er initiell i $\lfloor\simeq\rfloor_\Sigma$ for vilkårlige Σ og $\simeq$ på $\mathcal{G}_\Sigma$, siden hver $A \in \lfloor\simeq\rfloor_\Sigma$ er induktivt implisert av $\mathcal{G}_\Sigma/\simeq$. Punkt 1 i sats A.2 er dermed vist.

For punkt 2 i sats A.2, gir (2.6) på side 23 at $\mathcal{G}_\Sigma/\simeq \in \lceil\simeq\rceil_\Sigma^{sur}$. Ved sats A.4 følger det at det finnes en unik homomorfi fra enhver A i $\lceil\simeq\rceil_\Sigma^{sur}$ til $\mathcal{G}_\Sigma/\simeq$, siden $\mathcal{G}_\Sigma/\simeq$ er induktivt implisert av hver slik A. Følgelig er $\mathcal{G}_\Sigma/\simeq$ final i $\lceil\simeq\rceil_\Sigma^{sur}$.

Vi ser på unike homomorfier som *abstrakte tolker*. En homomorfi tolker et element i en algebra til et tilsvarende (ifølge den felles signatur) element i en (annen) algebra. At $\mathcal{G}_\Sigma/\simeq$ er initiell i $\lfloor\simeq\rfloor_\Sigma$ betyr i en viss forstand at $\mathcal{G}_\Sigma/\simeq$ er en *minste* algebra i $\lfloor\simeq\rfloor_\Sigma$ slik at det finnes en abstrakt tolk til alle algebraer i $\lfloor\simeq\rfloor_\Sigma$. Betrakt nemlig en $\simeq'$ slik at $\simeq \subset \simeq'$. Da vil $\mathcal{G}_\Sigma/\simeq' \in \lfloor\simeq\rfloor_\Sigma$, men det finnes ingen homomorfi fra $\mathcal{G}_\Sigma/\simeq'$ til f.eks. $\mathcal{G}_\Sigma/\simeq$. Vi har nemlig for noen $g, g' \in \mathcal{G}_\Sigma$ at $g \simeq' g'$ men $g \not\simeq g'$. En homomorfi ϕ fra $\mathcal{G}_\Sigma/\simeq'$ til $\mathcal{G}_\Sigma/\simeq$ må være slik at $\phi([g]_{\simeq'}) = [g]_\simeq$ og $\phi([g']_{\simeq'}) = [g']_\simeq$. Men $[g]_{\simeq'} = [g']_{\simeq'}$, så da må $[g]_\simeq = [g']_\simeq$. Sistnevnte er ikke mulig siden $g \not\simeq g'$.

A.1.3 Elementær ekvivalens og isomorfi

En homomorfi som er bijektiv kalles en ***isomorfi***. En isomorfi har, siden den er bikjektiv, en entydig invers. Denne inversen er også en isomorfi. Dersom det finnes en isomorfi mellom to algebraer A og B, sier vi at A og B er ***isomorfe***. Sammensetningen av homomorfier er en homomorfi. Siden sammensetningen av bijektive funksjoner er bijektiv, har vi at sammensetningen av isomorfier er en isomorfi.

Anta to Σ-algebraer A og B som er isomorfe ved en isomorfi $\phi \in \mathcal{H}om_A^B$. Bæremengdene til A og B er da like store (har lik kardinalitet), og med ethvert element a i A kan det assosieres et unikt element $b = \phi(a)$ i B som da igjen kan assosieres med elementet $a = \phi^{-1}(b)$.

Med hver funksjon f_A i A som er en tolkning av et symbol f i Σ, kan det videre assosieres en f_B og omvendt.

La f være et vilkårlig funksjonssymbol (med aritet n) i Σ, og la f_A og f_B være tolkningene av f i hhv. A og B. Enhver operasjon i A utført av operatoren f_A slik at f_A tar et n-tuppel $\langle a_1, \ldots, a_n\rangle$ og gir et element a kan da simuleres i B ved operasjonen der operatoren f_B tar n-tuplet $\langle \phi(a_1), \ldots, \phi(a_n)\rangle = \langle b_1, \ldots, b_n\rangle$ og gir $\phi(a) = b$. Likeledes simuleres operasjonen $f_B(b_1, \ldots, b_n) = b$ av operasjonen av $f_A(\phi^{-1}(b_1), \ldots, \phi^{-1}(b_n)) = \phi^{-1}(b)$, dvs. av operasjonen $f_A(a_1, \ldots, a_n) = a$.

Algebraer som er isomorfe er derfor *formlike* til en slik grad at den eneste essensielle matematiske forskjell er navnene på bestandelene av algebraene (sett bort fra «ekstra» funksjoner og skrot ikke representert i Σ).

For oss er det imidlertid en pragmatisk vesensforskjell. Denne vesensforskjell er grunnet i forskjellen mellom syntaks og semantiske størrelser, og gir seg utslag i følgende:

> *Betrakt en kvotientalgebra $\mathcal{G}_\Sigma/\simeq$. Anta at vi har et formelt mekanisk grep på $\simeq$. Dersom $\mathcal{G}_\Sigma/\simeq$ er isomorf med en algebra A, har vi dermed et formelt mekanisk grep på A.*

Ved vår intuisjon om isomorfi er dette formelle grepet nettopp det vi er ute etter ved implementasjon av abstrakte datatyper.

Hoveddiskusjonen i avsnitt 2.3 etablerer at korrekt implementasjon av en abstrakt datatype A oppnås f.eks. ved algebraisk spesifikasjon av samtlige funksjoner i A, samt implementasjon av identitetsrelasjonen på A. Dette uttryktes ved (2.3) på side 20.

For å fange inn dette, samt andre måter å spesifisere semantikk formelt vha. ligningslogikk på, uttrykkes 'korrekt implementasjon' ved kongruensrelasjonen $\simeq^A_{\mathcal{G}_\Sigma}$ indusert av unik grunntermtolk $\phi^A_{\mathcal{G}_\Sigma}$. 'Korrekt implementasjon' kan videre uttrykkes ved induktiv ekvivalens mellom en $\mathcal{G}_\Sigma/\simeq$ og A (A en Σ-algebra).

Poenget her i dette avsnittet, og det vi etterhvert skal vise, er at under surjektiv $\phi^A_{\mathcal{G}_\Sigma}$ så er induktiv ekvivalens, elementær ekvivalens og isomorfi mellom $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ og A ekvivalente begreper. Dette vil si følgende:

- En implementasjon $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ av en Σ-algebra A er isomorf med A. Således møtes den ved (2.3) mer operasjonelle måten å utrykke 'korrekt implementasjon' på, med en ren strukturell matematisk betraktning om isomorf simulering.

- En implementasjon $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ av en Σ-algebra A er elementært ekvivalent med A. Anta vi var interessert i å utlede sanne påstander om A. I den grad vi har et formelt mekanisk grep om $\simeq^A_{\mathcal{G}_\Sigma}$, har vi således et formelt mekanisk middel for å utlede sanne påstander om A.

La oss altså betrakte den spesielle kongruensrelasjon $\simeq^A_{\mathcal{G}_\Sigma}$ indusert av $\phi^A_{\mathcal{G}_\Sigma}$ for en Σ-algebra A. Muligens er $\phi^A_{\mathcal{G}_\Sigma}$ ikke injektiv. Betrakt så kvotientalgebraen $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$. Kvotientalgebraen $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ består nå av kongruensklasser av termer som tolkes til det samme i A av $\phi^A_{\mathcal{G}_\Sigma}$. Homomorfien ρ fra $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ til A er da en slags «injektiv versjon» av $\phi^A_{\mathcal{G}_\Sigma}$, og intuitivt er ikke-injektivitets-slakken dratt (akkurat passe) inn. Hvis $\phi^A_{\mathcal{G}_\Sigma}$ var surjektiv, skjønner vi intuitivt at $\mathcal{G}_\Sigma/\simeq^A_{\mathcal{G}_\Sigma}$ og A er isomorfe.

Imidlertid skal vi nå presist vise sammenhenger mellom induktiv ekvivalens, elementær ekvivalens og isomorfi.

Lemma A.5 *La A og B være to Σ-algebraer slik at det fins en surjektiv homomorfi fra A til B. Da er B elementært implisert av A.*

Bevis: La κ være en surjektiv homomorfi fra A til B. Anta $A \models u = v$ for en $u = v \in \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$. Betrakt så en vilkårlig homomorfi ϕ fra $\mathcal{T}_\Sigma(\mathcal{V})$ til B. Ved lemma A.1 har vi $\phi(s) = \kappa(\rho(s))$ for en homomorfi ρ fra $\mathcal{T}_\Sigma(\mathcal{V})$ til A. Ved antagelse er $\psi(u) = \psi(v)$ for alle homomorfier ψ fra $\mathcal{T}_\Sigma(\mathcal{V})$ til A, og spesielt er $\rho(u) = \rho(v)$. Det følger at

$$\phi(u) = \kappa(\rho(u)) = \kappa(\rho(v)) = \phi(v)$$

□

Sats A.6 *La A og B være to Σ-algebraer som er isomorfe. Da er A og B elementært ekvivalente.*

Bevis: Isomorfien ϕ fra A til B er en bijektiv homomorfi. Enhver bijeksjon har en entydig invers som også er en bijeksjon. Så vi får fra lemma A.5

$$A \models u = v \;\Rightarrow\; B \models u = v \;\text{ og }\; B \models u = v \;\Rightarrow\; A \models u = v$$

□

Resultatet i sats A.6 skulle bare mangle. Motsatsen gjelder dog ikke generelt. Her er et moteksempel:

Eksempel 88 Betrakt en signatur {f}. La A og K være to {f}-algebraer slik at

$$A = \langle \{a, b, c\}, \{Id\} \rangle$$

$$K = \langle \{k, l\}, \{Id\} \rangle$$

Funksjonssymbolet f tolkes til identitetsfunksjonen Id slik at $Id(x) = x$, i begge A og K. Både A og K tilfredstiller ligningene $\{\mathsf{f(x) = x}, \mathsf{f(f(x)) = f(x)}, \ldots\}$ og bare disse. Så A og K er elementært ekvivalente. Men A og K er ikke isomorfe.
○

Dette moteksemplet er mulig siden det ikke fins noen felles signatur Σ slik at det finnes surjektive homomorfier fra $\mathcal{G}_\Sigma$ til A og K. Vi skal se at dette er den eneste måten å lage et slikt moteksempel på. Dette gjør vi ved å utforske sammenhengen mellom induktiv implikasjon og unike homomorfier videre. Følgende to lemmata er påbygninger til sats A.4.

Lemma A.7 *La A og B være to Σ-algebraer slik at begge de unike homomorfiene fra $\mathcal{G}_\Sigma$ til A og fra $\mathcal{G}_\Sigma$ til B er surjektive, og slik at B er induktivt implisert av A. Da finnes en (unik) surjektiv homomorfi fra A til B.*

Bevis: Fra sats A.4 har vi at det finnes en unik homomorfi φ fra A til B. La κ være den surjektive homomorfi fra $\mathcal{G}_\Sigma$ til A og ϕ den unike homomorfi fra $\mathcal{G}_\Sigma$ til B. I beviset for sats A.4 viste vi at $\phi = \kappa \circ \varphi$. Nå er ϕ surjektiv, og det er lett å se at φ da må være surjektiv.
□

Lemma A.8 *La A og B være to Σ-algebraer slik at den unike homomorfien fra $\mathcal{G}_\Sigma$ til A er surjektiv, og slik at A er induktivt implisert av B. Da er avbildningen φ som definert i beviset for sats A.4 injektiv.*

Bevis: Avbildning φ fra A til B var definert slik: La κ være den surjektive homomorfi fra $\mathcal{G}_\Sigma$ til A og ϕ den unike homomorfi fra $\mathcal{G}_\Sigma$ til B. Betrakt et vilkårlig element $a \in A$. Siden κ er surjektiv, fins minst en $g \in \mathcal{G}_\Sigma$ slik at $\kappa(g) = a$. Velg en slik g og kall den g_a. For alle $a \in A$ og alle på denne måte assosierte g_a slik at $\kappa(g_a) = a$, definerer vi så φ fra A til B slik at $\varphi(a) = \phi(g_a)$.

Anta så at det fins $a, a' \in A$ slik at $a \neq a'$. Ved definisjonen av φ kan vi identifisere $g_a, g_{a'} \in \mathcal{G}_\Sigma$ slik at $\varphi(a) = \phi(g_a)$ og $\varphi(a') = \phi(g_{a'})$. Siden $A \not\models g_a = g_{a'}$ og vi har A induktivt implisert av B, har vi $B \not\models g_a = g_{a'}$ og dermed $\phi(g_a) \neq \phi(g_{a'})$. Men dette gir $\varphi(a) \neq \varphi(a')$ og dermed injektivitet av φ.
□

Sats A.9 *La A og B være to Σ-algebraer slik at de to unike homomorfiene fra $\mathcal{G}_\Sigma$ til A og fra $\mathcal{G}_\Sigma$ til B er surjektive og slik at A og B er induktivt ekvivalente. Da er A og B isomorfe.*

Bevis: Fra sats A.4 har vi at det finnes en unik homomorfi φ fra A til B. Fra lemma A.7 har vi at φ er surjektiv. Fra lemma A.8 har vi at φ er injektiv.
□

Vi oppsummerer (grovt).

Teorem A.10 *La A og B være to Σ-algebraer slik at homomorfiene fra $\mathcal{G}_\Sigma$ til A og fra $\mathcal{G}_\Sigma$ til B er surjektive. Da har vi:*
A og B er isomorfe hvis og bare hvis A og B er elementært ekvivalente.

Bevis: Anta A og B er isomorfe. Ved sats A.6 har vi at A og B er elementært ekvivalente. Anta A og B er elementært ekvivalente. Elementær ekvivalens medfører induktiv ekvivalens, og A og B er da isomorfe ved sats A.9.
□

For en Σ-algebra A, anta $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv. Betrakt så $\mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}}$. Siden den unike homomorfi fra $\mathcal{G}_\Sigma$ til $\mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}}$ er surjektiv, har vi ved sats A.9 og siden $\mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}}$ og A er induktivt ekvivalente, at

- $\mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}}$ og A er isomorfe

og dermed ved sats A.6 at

- $\mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}}$ og A er elementært ekvivalente.

(Sistnevnte har vi også etablert på side 23.)

A.1.4 Semantikk på termer med variable

Det er selvsagt også mulig å gi semantikk til termer med variable. Relasjonen $\simeq^A$ slik at $s \simeq^A t \;\Leftrightarrow\; A \models s = t$, for alle $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$ og A en Σ-algebra, er av en viss interesse for oss, siden denne også er mengden av ligninger tilfredstilt i A. Restriksjonen av $\simeq^A$ til $\mathcal{G}_\Sigma \times \mathcal{G}_\Sigma$ er den kjente kongruensrelasjon indusert av $\phi^A_{\mathcal{G}_\Sigma}$.

Merk at dersom $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv, kan $\simeq^A$ utrykkes ved $\simeq^A_{\mathcal{G}_\Sigma}$:

$$s \simeq^A t \;\Leftrightarrow\; A \models s = t \;\Leftrightarrow\; \mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}} \models s = t \;\Leftrightarrow\; \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \simeq^A_{\mathcal{G}_\Sigma} t\sigma$$

for alle $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$. Dette ved elementær ekvivalens mellom $\mathcal{G}_\Sigma/{\simeq^A_{\mathcal{G}_\Sigma}}$ og A og ved (2.2) side 17, kombinert med (2.6) side 23.

Vi viser nå at $\simeq^A$ er en kongruensrelasjon.

Sats A.11 *La Σ være en vilkårlig signatur. La A være en vilkårlig Σ-algebra. Relasjonen $\simeq^A$ på $\mathcal{T}_\Sigma(\mathcal{V})$ slik at $s \simeq^A t \;\Leftrightarrow\; A \models s = t$, er en kongruensrelasjon på $\mathcal{T}_\Sigma(\mathcal{V})$.*

Bevis: For vilkårlige $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$ betyr $s \simeq^A t$ at $\forall \phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})} \mid \phi(s) = \phi(t)$. Da er $\simeq^A$ en ekvivalensrelasjon ved at identitetsrelasjonen på A er en ekvivalensrelasjon. La nå f_T være en vilkårlig n-ær funksjon i $\mathcal{T}_\Sigma(\mathcal{V})$. Anta $s_i \simeq^A t_i$ for vilkårlige $s_i, t_i \in \mathcal{T}_\Sigma(\mathcal{V}); 1 \leq i \leq n$. Dvs. $\forall \phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})} \mid \phi(s_i) = \phi(t_i); 1 \leq i \leq n$. Men vi har $\phi(f_T(s_1,\ldots,s_n)) = f_A(\phi(s_1),\ldots,\phi(s_n))$ og $\phi(f_T(t_1,\ldots,t_n)) =$

$f_A(\phi(t_1),\ldots,\phi(t_n))$ for alle $\phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})}$, og ved funksjonsapplikasjon av identitiske elementer får vi så $\forall \phi \in \mathcal{H}om^A_{\mathcal{T}_\Sigma(\mathcal{V})} \mid \phi(f_T(s_1,\ldots,s_n)) = \phi(f_T(t_1,\ldots,t_n))$, og dermed $f_T(s_1,\ldots,s_n) \simeq^A f_T(t_1,\ldots,t_n)$.
□

Følgende resultat er dessuten viktig for oss:

Sats A.12 *La Σ være en vilkårlig signatur. La A være en vilkårlig Σ-algebra og B et vilkårlig redukt av A. La $\simeq$ være en kongruensrelasjon på A. Restriksjonen $\simeq_B$ av $\simeq$ til $B \times B$ er en kongruensrelasjon på B.*

Bevis: Siden $\simeq_B \subseteq \simeq$ er det innlysende at $\simeq_B$ er en ekvivalensrelasjon på B. La så f_B være en vilkårlig funksjon i B. Anta f_B er n-ær. Anta $b_i \simeq_B b'_i$ for vilkårlige $b_i, b'_i \in B; 1 \leq i \leq n$. Siden $\simeq_B \subseteq \simeq$ har vi også $b_i \simeq b'_i$. Ved definisjon av redukt er f_B en restriksjon av en funksjon i A. Siden $\simeq$ er en kongruensrelasjon (på A), får vi derfor $f_B(b_1,\ldots,b_n) \simeq f_B(b'_1,\ldots,b'_n)$. Siden B er en algebra er $f_B(\beta_1,\ldots,\beta_n) \in B$ for vilkårlige $\beta_i \in B; 1 \leq i \leq n$. Da får vi, siden $\simeq_B \subseteq \simeq$, $f_B(b_1,\ldots,b_n) \simeq_B f_B(b'_1,\ldots,b'_n)$.
□

At $\simeq^A_{\mathcal{G}_\Sigma}$ er en kongruensrelasjon følger fra A.12; men dette er også godtgjort direkte i sats 2.1 på side 23.

Vi utvider notasjonen $\lfloor\simeq\rfloor_\Sigma$ til å omfatte vilkårlige kongruensrelasjoner på vilkårlige Σ-term-algebraer. Altså $\lfloor\simeq\rfloor_\Sigma = \{A \mid s \simeq t \Rightarrow A \models s = t\}$ eksempelvis. For en kongruensrelasjon $\simeq$ på en $\mathcal{T}_\Sigma(\mathcal{V})$, har vi ved sats A.4 at det finnes en unik homomorfi fra $\mathcal{G}_\Sigma/{\simeq_{\mathcal{G}_\Sigma}}$ til alle $A \in \lfloor\simeq\rfloor_\Sigma$, siden hver A er induktivt implisert av $\mathcal{G}_\Sigma/{\simeq_{\mathcal{G}_\Sigma}}$. Imidlertid er det ikke sikkert at $\mathcal{G}_\Sigma/{\simeq_{\mathcal{G}_\Sigma}} \in \lfloor\simeq\rfloor_\Sigma$, så $\mathcal{G}_\Sigma/{\simeq_{\mathcal{G}_\Sigma}}$ er ikke nødvendigvis initiell i $\lfloor\simeq\rfloor_\Sigma$.

A.2 Induktive konsekvenser og surjektivitet

Dette avsnittet bør leses i forbindelse med avsnitt 2.4.1 på side 47.

Mengden $\mathcal{T}eo(E)$ er mengden av ligninger som følger ved *logisk nødvendighet* fra en ligningsmengde $E \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ for en Σ; m.a.o. de ligninger som er tilfredstilt i enhver modell for E. Nå er ikke modeller for E nødvendigvis homomorfe bilder av $\mathcal{G}_\Sigma$. Vi skal se at forskjellen mellom $\mathcal{T}eo(E)$ og $\mathcal{I}nd(E)$ er betinget av nettopp disse fra $\mathcal{G}_\Sigma$ «ikke-kontrollerbare» modeller for E.

Først viser vi at $\mathcal{G}_\Sigma/E \in \mathcal{M}od_\Sigma(E)$. Ved teorem 2.24 side 48 kan $\mathcal{M}od_\Sigma(E)$ uttrykkes ved $\lfloor\overset{*}{\underset{E}{\leftrightarrow}}\rfloor_\Sigma = \{A \mid s \overset{*}{\underset{E}{\leftrightarrow}} t \Rightarrow A \models s = t\}$. Anta så $s \overset{*}{\underset{E}{\leftrightarrow}} t$ for vilkårlige $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$. Spesielt har vi da ved monotonitet mhp. substitusjon $\forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \overset{*}{\underset{E}{\leftrightarrow}} t\sigma$. Men vi har jo $\mathcal{G}_\Sigma/E \models s = t \Leftrightarrow \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid s\sigma \overset{*}{\underset{E}{\leftrightarrow}} t\sigma$. Ergo er $\mathcal{G}_\Sigma/E \in \mathcal{M}od_\Sigma(E)$. (Ved diskusjonen i avsnitt A.1 er da også $\mathcal{G}_\Sigma/E$ initiell i $\mathcal{M}od_\Sigma(E)$.)

Definisjon A.1 *La $\mathcal{K}$ være en klasse algebraer. Vi lar $\mathcal{K} \models s = t$ bety at $A \models s = t$ for alle $A \in \mathcal{K}$ for en ligning $s = t$. For en signatur Σ er algebraen Q en* ***induktiv representant*** *for $\mathcal{K}$ dersom*

$$Q \models g = g' \Leftrightarrow \mathcal{K} \models g = g'$$

for alle $g, g' \in \mathcal{G}_\Sigma$ og en ***elementær representant*** *for $\mathcal{K}$ dersom*

$$Q \models s = t \Leftrightarrow \mathcal{K} \models s = t$$

for alle $s,t \in \mathcal{T}_\Sigma(\mathcal{V})$

Vi har altså sett at $\mathcal{G}_\Sigma/E$ generelt *ikke* er en elementær representant for klassen $\mathcal{M}od_\Sigma(E)$.

Lemma A.13 *Betrakt $\mathcal{G}_\Sigma$ for en vilkårlig signatur Σ. La $\mathcal{K}$ være en klasse av algebraer slik at $\phi^A_{\mathcal{G}_\Sigma}$ er surjektiv for alle $A \in \mathcal{K}$. La Q være en induktiv representant for $\mathcal{K}$. Anta videre at $Q \in \mathcal{K}$. Da er Q en elementær representant for $\mathcal{K}$.*

Bevis: Observer at alle elementene i $\mathcal{K}$ er Σ-algebraer. La $A \in \mathcal{K}$ være vilkårlig. Q er en induktiv representant for $\mathcal{K}$ mhp. Σ, så $Q \models g = g' \Rightarrow A \models g = g'$ for alle $g, g' \in \mathcal{G}_\Sigma$. Nå er $Q \in \mathcal{K}$, så $\phi^Q_{\mathcal{G}_\Sigma}$ er surjektiv, og lemma A.7 gir at det finnes en surjektiv homomorfi fra Q til A. Og da gir lemma A.5 at $Q \models s = t \Rightarrow A \models s = t$ for alle $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$. Vi har altså $Q \models s = t \Rightarrow \mathcal{K} \models s = t$ for alle $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$. Observer til slutt at siden $Q \in \mathcal{K}$ har vi trivielt $\mathcal{K} \models s = t \Rightarrow Q \models s = t$ for alle $s, t \in \mathcal{T}_\Sigma(\mathcal{V})$.
□

Teorem A.14 *La Σ være en vilkårlig signatur. Betrakt en vilkårlig ligningsmengde E. La $\mathcal{M}od_\Sigma(E)_{sur}$ betegne snittet av $\mathcal{M}od_\Sigma(E)$ med mengden av homomorfe bilder av $\mathcal{G}_\Sigma$. La $E_{sur} \models s = t$ bety at $s = t$ er tilfredstilt av alle $A \in \mathcal{M}od_\Sigma(E)_{sur}$. Vi har:*

$$\mathcal{G}_\Sigma/E \models u = v \;\Leftrightarrow\; E_{sur} \models u = v$$

for alle $u, v \in \mathcal{T}_\Sigma(\mathcal{V})$.

Bevis: $\mathcal{G}_\Sigma/E$ er en modell for E, og det fins en surjektiv homomorfi fra $\mathcal{G}_\Sigma$ til $\mathcal{G}_\Sigma/E$, så $\mathcal{G}_\Sigma/E \in \mathcal{M}od_\Sigma(E)_{sur}$. Altså har vi $E_{sur} \models u = v \Rightarrow \mathcal{G}_\Sigma/E \models u = v$ for alle $u, v \in \mathcal{T}_\Sigma(\mathcal{V})$ og spesielt

$$E_{sur} \models g = g' \;\Rightarrow\; \mathcal{G}_\Sigma/E \models g = g'$$

for alle $g, g' \in \mathcal{G}_\Sigma$.

Videre har vi ved (2.2) side 17, (2.6) side 23 og Birkhoffs teorem 2.24 side 48

$$\mathcal{G}_\Sigma/E \models u = v \;\Leftrightarrow\; \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid E \models u\sigma = v\sigma$$

for alle $u, v \in \mathcal{T}_\Sigma(\mathcal{V})$. Spesielt har vi da $\mathcal{G}_\Sigma/E \models g = g' \Rightarrow E \models g = g'$, og siden $\mathcal{M}od_\Sigma(E)_{sur} \subset \mathcal{M}od_\Sigma(E)$ har vi

$$E \models g = g' \;\Rightarrow\; E_{sur} \models g = g'$$

for alle $g, g' \in \mathcal{G}_\Sigma$. Altså er $\mathcal{G}_\Sigma/E$ en induktiv representant for $\mathcal{M}od_\Sigma(E)_{sur}$. Siden $\mathcal{G}_\Sigma/E \in \mathcal{M}od_\Sigma(E)_{sur}$ gir lemma A.13 at $\mathcal{G}_\Sigma/E$ er en elementær representant for $\mathcal{M}od_\Sigma(E)_{sur}$.
□

Ser vi altså bort fra modeller som ikke er homomorfe bilder av grunnterm-algebraer, har vi på et vis, og sagt litt sleivete, at den «logiske» teori og den induktive teori er identisk.

De modeller som betraktes som abstrakte datatyper i implementasjons-/spesifikasjons-/programmerings-sammenheng er nettopp homomorfe bilder av grunnterm-algebraer (jfr avsnitt 2.2.4 på side 15).

$\mathcal{M}od_\Sigma(E)$ har også en elementær representant mhp. Σ; nemlig $\mathcal{T}_\Sigma(\mathcal{V})/E$ ved (2.6) side 23. Vi oppsummerer: For en vilkårlig Σ har vi for vilkårlige $u, v \in \mathcal{T}_\Sigma(\mathcal{V})$:

- $E \models u = v \;\Leftrightarrow\; \mathcal{T}_\Sigma(\mathcal{V})/E \models u = v \;\Leftrightarrow\; u \overset{*}{\underset{E}{\leftrightarrow}} v$
- $E_{sur} \models u = v \;\Leftrightarrow\; \mathcal{G}_\Sigma/E \models u = v \;\Leftrightarrow\; \forall \sigma \in \mathcal{S}bst^{\mathcal{G}_\Sigma} \mid u\sigma \overset{*}{\underset{E}{\leftrightarrow}} v\sigma$

A.3 Omskrivningssystemer og konvergens

Dette avsnittet inneholder beviset av (2.16) på side 49, samt en bemerkning om definisjonen av $\mathcal{T}$-konvergens når $\mathcal{T}$ er hele termuniverset i den aktuelle diskusjon.

A.3.1 Bevis for (2.16) side 49

Vi skal altså vise

Sats A.15 *La $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ være et omskrivningssystem for en signatur Σ. La $\mathcal{T}$ være en term-mengde slik at $\mathcal{T} \subseteq \mathcal{T}_\Sigma(\mathcal{V})$. Anta R er $\mathcal{T}$-konvergent. Da finnes $s!R$ og $t!R$ for vilkårlige $s,t \in \mathcal{T}$, og*

$$s \overset{*}{\underset{R}{\leftrightarrow}} t \;\Leftrightarrow\; s!R = t!R$$

Bevis: Anta $s!R = t!R$ for vilkårlige $s,t \in \mathcal{T}$. Da har vi trivielt $s \overset{*}{\underset{R}{\leftrightarrow}} t$.

Anta så at $s \overset{*}{\underset{R}{\leftrightarrow}} t$. Siden R er $\mathcal{T}$-Church-Rosser finnes en $u \in \mathcal{T}_\Sigma(\mathcal{V})$ slik at $s \overset{*}{\underset{R}{\rightarrow}} u \overset{*}{\underset{R}{\leftarrow}} t$. Nå må u ha en normalform ellers fantes en uendelig $\underset{R}{\rightarrow}$-utledning $\langle s, \ldots \rangle$ og R var ikke $\mathcal{T}$-terminerende. Siden R har entydige $\mathcal{T}$-normalformer må videre u ha en *entydig* normalform $u!R$ ellers ville $s \in \mathcal{T}$ ikke ha en entydig normalform. Men $u!R$ er også en normalform for t, og siden R har entydige $\mathcal{T}$-normalformer har vi at $s!R = t!R$.

□

A.3.2 Delvis konvergens vs. full konvergens

La $\mathcal{T}_\Sigma(\mathcal{V})$ være vårt termunivers. La $\mathcal{T}$ være en term-mengde slik at $\mathcal{T} \subseteq \mathcal{T}_\Sigma(\mathcal{V})$. Vi har i på side 49 definert $\mathcal{T}$-konvergens for en vilkårlig $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ ved at R

- er $\mathcal{T}$-terminerende
- har entydige $\mathcal{T}$-normalformer
- er $\mathcal{T}$-Church-Rosser

Dersom $\mathcal{T} = \mathcal{T}_\Sigma(\mathcal{V})$ kan kravet om $\mathcal{T}_\Sigma(\mathcal{V})$-Church-Rosser utelates i definisjonen for $(\mathcal{T}_\Sigma(\mathcal{V}))$-konvergens, siden:

Lemma A.16 *Hvis et omskrivningssystem $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ er $(\mathcal{T}_\Sigma(\mathcal{V}))$-terminerende og har entydige $(\mathcal{T}_\Sigma(\mathcal{V}))$-normalformer så er det $(\mathcal{T}_\Sigma(\mathcal{V}))$-Church-Rosser.*

At et omskrivningssystem $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$ generelt er $\mathcal{T}$-Church-Rosser hvis det er $\mathcal{T}$-terminerende og har entydige $\mathcal{T}$-normalformer holder ikke:

Eksempel 89 La $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ være en signatur av konstanter. La $\mathcal{T} = \{\mathsf{a}, \mathsf{b}\}$. Betrakt følgende $R \subseteq \mathcal{E}(\mathcal{T}_\Sigma(\mathcal{V}))$:

$$\{\mathsf{c} \rightarrow \mathsf{a}, \mathsf{c} \rightarrow \mathsf{b}\}$$

R er $\mathcal{T}$-terminerende og har entydige $\mathcal{T}$-normalformer, men er ikke $\mathcal{T}$-Church-Rosser, siden

$$\mathsf{a} \underset{R}{\leftarrow} \mathsf{c} \underset{R}{\rightarrow} \mathsf{b}$$

○

Symboltabell

A_T (Delmengde (bæremengde) av domenet A til en algebra som er tolkningen i algebraen av en type T), 14

$A/\simeq$ (Kvotientalgebra av A over $\simeq$), 23

$\mathbb{F}$ (Mengden av alle funksjoner), 11

$\mathcal{G}_\Sigma$ (Grunntermalgebraen over Σ), 15

$\mathcal{GT}ype$ (Symbolmengde av grunntyper), 12

$\mathcal{H}om_A^B$ (Mengden av homomorfier fra A til B), 14

$\mathbb{N}$ (Mengden av naturlige tall), 14

$\mathcal{P}(A)$ (Mengden av alle delmengder av A), 11

$\Phi_{\mathcal{F}}$ (Funksjonsprofiltolk), 13

$\Phi_{\mathcal{T}}$ (Typetolk), 13

$\mathbb{R}$ (Mengden av reelle tall), 12

$\Re_{A'_1 \times \cdots \times A'_n}$ (Restriksjonen av relasjonen $\Re$ til $A'_1 \times \cdots \times A'_n$), 10

$\Re_{A'}$ (Restriksjonen av relasjonen $\Re$ til $A'_1 \times \cdots \times A'_n$ for $A'_1 = \cdots = A'_n = A'$), 10

$\mathbb{S}$ (Mengden av alle mengder), 11

$\mathcal{S}bst^{\mathcal{T}}$ (Mengden av substitusjoner med kodomene $\mathcal{T}$), 17

$\mathcal{T}_\Sigma(\mathcal{V})$ (Term-algebraen over Σ og $\mathcal{V}$), 15

$\mathcal{T}ype$ (Symbolmengde av typer), 12

$\mathbb{Z}$ (Mengden av hele tall), 14

ϕ_A^B (En homomorfi fra A til B), 14

$g_{\Downarrow s}$ (Termen identisk med g, men med alle forekomster av spesifiserende symbol s fjernet), 102

$\overline{g}$ (En generatorterm slik at $g \overset{*}{\underset{E}{\leftrightarrow}} \overline{g}$), 92

$\tilde{g}_c$ (En generatorterm slik at $s(g_c) \overset{*}{\underset{E_s}{\leftrightarrow}} \tilde{g}_c$, for en generatorterm g_c og spesifiserende symbol s), 92

$\Box$ (Hull i en kontekst.), 18

$\simeq^A_{\mathcal{G}_\Sigma}$ (Kongruensrelasjonen indusert i $\mathcal{G}_\Sigma$ av den unike $\phi^A_{\mathcal{G}_\Sigma}$), 23

$\simeq^\alpha$ (initialsemantikk), 31

$\simeq^\omega$ (finalsemantikk), 27

$\simeq^s$ (indirekte semantikk), 85

$\simeq^x$ (kjernesemantikk), 27

len_s (Lengden av sekvensen s), 10

$s!R$ (Den unike normalformen i R av s), 49

Referanser

[Apt90] K.R. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 10, pages 491–574. Elsevier, Amsterdam, 1990.

[Bac87] L. Bachmair. *Proof methods for equational theories.* PhD thesis, University of Illinois, Urbana-Champaign, 1987.

[Bac88] L. Bachmair. Proof by consistency in equational theories. In *Proceedings 3rd IEEE Symposium on Logic in Computer Science, Edinburgh (UK)*, pages 228–233, 1988.

[Bir35] G. Birkhoff. On the structure of abstract algebras. In *Proceedings Cambridge Philosophical Society*, 31, pages 433–454, 1935.

[BM88] R. S. Boyer and J. S. Moore. *A Computational Logic Handbook.* Academic Press, Inc., 1988.

[Bur69] R. Burstall. Proving properties of programs by structural induction. *Computer Journal*, 12(1):41–48, 1969.

[Dah92] O.-J. Dahl. *Verifiable Programming.* Prentice Hall International Series in Computer Science; C.A.R. Hoare, Series Editor. Prentice-Hall, UK, 1992.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6. Elsevier, Amsterdam, 1990.

[DMN70] O.-J. Dahl, B. Myhrhaug, and K. Nygaard. Simula 67 Common Base Language. Publikasjon N. S-22, Norsk Regnesentral, Oslo, okt. 1970.

[DO91] O.-J. Dahl and O. Owe. Formal development with ABEL. In *Proceedings 4th International Symposium of VDM Europe*, volume 552 of *Lecture Notes in Computer Science*, 1991.

[DR91] N. Dershowitz and U.S. Reddy. Deductive and inductive synthesis of equational programs. *Journal of Symbolic Computation*, 15:467–494, 1991.

[EN58] E.Nagel and J.R. Newman. *Gödel's Proof.* New York University Press, New York, N.Y., 1958.

[GG91] S.J. Garland and J.V. Guttag. A guide to LP, the Larch Prover. Technical report, Digital Equipment Corporation Systems Research Center, desember 1991.

[GJ67] G.Kreisel and J.L.Krivine. *Elements of Mathematical Logic.* North-Holland, 1967.

[Göd31] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[GR83] A. Goldberg and D. Robson. *Smalltalk-80: The Language and its Implementation*. Addison-Wesley, Reading, Mass., 1983.

[Gut75] J.V. Guttag. *The Specification and Application to Programming of Abstract Data Types*. PhD thesis, Computer Science Department, University of Toronto, 1975.

[Gut77] J.V. Guttag. Abstract datatypes and the development of data structures. *Communications of the ACM*, 20(6):396–404, 1977.

[GW88] J.A. Goguen and T. Winkler. Introducing OBJ3. Technical report, SRI International, Computer Science Lab, 1988.

[Han92] J.E. Hannay. Konstruksjon. Arbeidsnotat, november 1992.

[Han94] J.E. Hannay. Generering av (direkte) algebraiske spesifikasjoner fra indirekte algebraiske spesifikasjoner. Arbeidsnotat, mai 1994.

[Her92] M. Hermann. On the relation between primitive recursion, schematization, and divergence. In Giorgio Levi, editor, *Proceedings of the 3rd International Conference on Algebraic and Logic Programming, Volterra, Italy*, volume 632 of *Lecture Notes in Computer Science*, pages 115–127. Springer-Verlag, September 1992.

[HH82] G. Huet and J.-M. Hullot. Proofs by induction in equational theories with constructors. *Journal of Computer and System Sciences*, pages 239–266, 1982.

[Hoa69] C.A.R. Hoare. An Axiomatic Approach to Computer Programming. *Communications of the ACM*, 12:576–580, 1969.

[HZ89] X. Hua and H. Zhang. An overwiew of RRL: Rewrite Rule Laborotory. In *Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (North Carolina, USA)*, volume 355 of *Lecture Notes in Computer Science*, pages 513–529. Springer-Verlag, april 1989.

[JK89] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82(1):1–33, July 1989.

[KB70] D.E. Knuth and P.B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[Kir94] B. Kirkerud. Omskrivningssystemer. Forelesningsnotater til IN307, Institutt for informatikk, Universitetet i Oslo, 1994.

[KM87] D. Kapur and R. Musser. Proof by consistency. *Artificial Intelligence*, 31:125–157, 1987.

[KNZ87] D. Kapur, P. Narendran, and H. Zhang. On sufficient completeness ans related properties of term rewriting systems. *Acta Informatica*, 24(4):395–415, 1987.

[Lys92] O. Lysne. Proof by consistency in constructive systems with final algebra semantics. In *Proceedings 3rd International Conference on Algebraic and Logic Programming, Pisa (Italy)*, volume 632 of *Lecture Notes in Computer Science*, pages 276–290. Springer-Verlag, 1992.

[Lys93] O. Lysne. The equational part of proofs by structural induction. *BIT*, 33:596–618, 1993.

[Lys94a] O. Lysne. Extending Bachmair's method for proof by consistency to the final algebra. *Information Processing Letters*, 51:303–310, 1994.

[Lys94b] O. Lysne. Heuristics for completion in automatic proofs by structural induction. *Nordic Journal of Computing*, 1:135–156, 1994.

[Mar47] A.A. Markov. The impossibility of some algorithms in the theory of associative systems. *Dokl. Akad. Nauk SSSR*, 55(7):587–590, 1947.

[Mos90] P.D. Mosses. Denotational semantics. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 11, pages 575–631. Elsevier, Amsterdam, 1990.

[Mus80] D.L. Musser. On proving inductive properties in abstract data types. In *Proceedings of the 7th Annual ACM Symposium on Principles of Programming Languages*, pages 154–162, January 1980.

[New42] M.H.A. Newman. On theories with a combinatorial definition of 'equvalence'. *Ann. Math.*, 43(2):223–243, 1942.

[Pla85] D. Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65:182–215, 1985.

[Pos47] E.L. Post. Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic*, 13:1–11, 1947.

[Rob65] J.A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, 12(1):23–41, 1965.

[Tar68] A. Tarski. Equational logic and equational theories of algebras. In K. Schutte, editor, *Contributions to Mathematical Logic*. North-Holland, Amsterdam, 1968.

[Tur36] A.M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *London Mathematical Society*, 2(42, 43):230–265, 544–546, 1936.

Register

ASSS, 88
DEGEN, 39
DISJ, 84
E^dFRI^c, 87
E^dKONSERV+, 87
E^dVARBEVAR, 84
E_sFRI^c, 87
E_sKONSERV+, 87
E_sVARBEVAR, 84
hROT, 94
INDDEGEN, 84
INDKONG, 84
INDKONSERV, 86
KONSERV, 28
KONSTR, 86
KONVERG, 86
KREP1, 91
KREP2, 102
$\Sigma^h = \{h\}$, 95
TK, 28
TΣK, 85
TΣ^hK, 85
T$\hat{\Sigma}$K, 85
TsK, 85

abstrakt deterministisk program, *se* ligninger
abstrakt ikke-deterministisk program, *se* ligninger
abstrakte maskiner, 3
adekvat regelmengde, 49
algebra, 13
  bæremengde til, 14
  domene til, 14
  funksjonsmengde til, 14
  utvidelse av, 74
algebraisk funksjonsbeskrivelse, 20
algoritmisk, 12
avgjørbar, 12
  effektivt, 12

bæremengde, *se* algebra
begrense semantikk, *se* semantikk
bijektiv, *se* funksjon

Church-Rosser, 49

datatype, 1
  abstrakt, 2
  formell, 2
    basis-final, 27
    basis-initiell, 24
    final, 28
    initiell, 31
definert funksjonssymbol, 25
delvis monoton tillukning, *se* relasjon
direkte spesifikasjon, *se* spesifikasjon
domene
  algeba-, *se* algebra
  funksjons-, *se* funksjon
  relasjons-, *se* relasjon

elementært ekvivalent, 17
elementært implisert, 17

fikspunktfunksjon, 64
final, 14
finistiske bevisprinsipper, 60
formell omgivelse, 46
  grensesnitt-, 47
  logisk, 47
fri ligningsmengde, 58
fri semantikk, *se* semantikk
full uttrykkbarhet, 16
fullstendig definerthet, 25
funksjon, 11
  applikasjon av, 11
  bijektiv, 11
  bildet av, 11
  domene til, 11
  fikspunkt til, 11
  injektiv, 11
  kodomene til, 11
  konstant-, 11
  surjektiv, 11
  verdi av, 11
funksjonsmengde, *se* algebra
funksjonsvariant, 11

funksjonsymbol, 12

generator, 25
  -funksjon, 3
  -termer, 4
  -univers, 4
grensesnitt-omgivelse, *se* formell omgivelse
grunnredusibel, 56

hjelpefunksjon(symbol)er, 80
homomorfi, 14
homomorft bilde, 14
hull, 18

id-utvidelse, 91
  generalisert, 141
identitet, 9
id-restriksjon, 127
implementasjon
  definisjon av, 23
  intuitivt, 12
    operasjonell del av, 12
    representasjonell del av, 12
indirekte $\mathcal{G}$-kongruent, 84
indirekte spesifikasjon, *se* spesifikasjon
induktiv grense, 55
induktiv komplettering, 56
  RD-restriktert, 133
  RI-restriktert, 133
  sekvens-utvidet, 161
induktiv teori, 48
induktive konsekvenser, 48
induktivt implisert, 17
induktivt ekvivalent, 17
initiell, 14
injektiv, *se* funksjon
instansiering, 16
invers
  til funksjon, *se* funksjon
  til relasjon, *se* binær-relasjon
isomorfi, 169

kanonisk representant, *se* klasserepresentant
kanonisk-representant funksjon, 64
kjernebevarende
  finalt, 33
  initielt, *se* konsistens
klasserepresentant, 66
  kanonisk, 64
  naturlige, 67
klasserepresentant funksjon, 66
Knuth&Bendix-prosess
  RD-restriktert, 133
  RI-restriktert, 132
  terminerende, 55
  terminerende mislykket, 55
  terminerende vellykket, 55
  vellykket, 55
kodomene, *se* funksjon
komplett regelmengde, 49
komplettering, *se* Knuth&Bendix-prosess
konfluens
  global, 53
  lokal, 53
kongruensrelasjon, 22
  indusert av homomorfi, 23
kongruent, *se* relasjon
konsistens
  final, 33
    indre, 33
  initiell, 32
    relativ til kjerne, 33
konstant, 12
kontekst, 18
  tom, 18
konvergent, 49
korrekthet, 1
kritisk par
  ekte, 53
  trivielt, 53
kunstig inkonsistens, 125

leksikografisk ordning, 100
ligninger, 16
  ω-komplette, 48
  som abstrakte programmer, 4
    deterministiske, 5, 49
    ikke-deterministiske, 5, 21
ligningslogikk, 19
logisk omgivelse, *se* formell omgivelse
logisk teori, 48
lukket mengde, *se* mengde

manifest, 113
  i henhold til skjuling, 132
  kjerne, 113
matching, 18
mengde, 9
  avgjørbar, 12
  lukket, 11
modul, 2
monoton, *se* relasjon

n-foldige kartesiske produkt, 10
n-tuppel, 10
normalform, 49
normalformer
    entydige, 49

observator, 28
omskrivbar, 19
omskrivningsberegnbar, 63
omskrivningsrelasjon, 18
omskrivningssystem, 18
operator, 11
    gi, 11
    ta, 11

posisjon, 17
profil
    funksjons-, 12
    variabel-, 13

RD-restriktert komplettering, *se* Knuth&Bendix-prosess
realisere semantikk, *se* semantikk
reduksjon, 66
reduksjonsordning, 19
redukt, 14
relasjon, 10
    binær, 10
      delvis monoton tillukning av, 126
      invers av, 10
      monoton, 18
      monotone tillukning av, 19
      refleksiv, 10
      refleksiv tillukning av, 10
      refleksiv-transitiv tillukning av, 10
      sammensetning, 10
      symmetrisk, 10
      symmetrisk tillukning av, 10
      transitiv, 10
      transitiv tillukning av, 10
      velfundert, 10
    domene til, 10
    ekvivalens-, 10
    restriksjon av, 10
    universell, 10
resolusjon, 47
    metoder, 50
resonnering
    formell, 3
    mekanisk, 3
    ustrukturert, 1
RI-restriktert komplettering, *se* Knuth-&Bendix-prosess
Russels paradox, 11

sann, 16
sekvens, 10
sekvens-utvidet komplettering, *se* Knuth&Bendix-prosess, 151
semantikk, 24
    atomær, 46
    basis-final, 27
    basis-initial, 24
    begrense, 44, 168
    final pseudo-, 27
    final-, 28
    for funksjonssymboler, 24
    fri, 24
    fullstendig, 30
    indirekte, 84
    initial, 31
    kjerne-, 27
    løs, 31
    på termer, 24
    realisere, 26, 168
    separabel, 41
    ønsket, 44
semantikkantydende, *se* syntaktiske funksjoner
semantikkgivende, *se* syntaktiske funksjoner
semantisk sprang, 2
semantisk subtype, 69
signatur
    funksjons-, 12
    variabel-, 13
skrot, 15
spesifikasjon
    av kongruensrelasjon
      direkte, 25
      indirekte, 84
      mhp. syntaktisk funksjon, 81
    funksjons-, 19
    konstruktiv, 19, 67
spesifiserende symbol, 84
subalgebra, 14
substitusjon, 16
    grunn-, 16
    $\mathcal{T}$-, 17
subterm-egenskap
    kanonisk representant, 73
sunn regelmengde, 49
surjektiv, *se* funksjon
syntaktisk funksjon, 63
syntaktiske funksjoner

semantikkantydende, 143
semantikkgivende, 64

term, 13
dybde, 18
grunn-, 13
rot i, 18
som tre, 18
subterm, 18
ekte, 18
term-algebra, 14
terminerende Knuth&Bendix-prosess, *se* Knuth&Bendix-prosess
terminerende omskrivningssystem, 49
terminerende mislykket Knuth&Bendix-prosess, *se* Knuth&Bendix-prosess
terminerende vellykket Knuth&Bendix-prosess, *se* Knuth&Bendix-prosess
termomskrivningsberegnbar, *se* omskrivningsberegnbar
termomskrivningssystem, *se* omskrivningssystem
termunivers, 19
tilfredstille, 16
tilstrekkelig komplett, 25
tolk
funksjonsprofil-, 13
grunnterm-, 15
type-, 13
type, 12
grunn-, 12

unifikator, 52
mest generell, 52
unifiserbar, 52
utledbar, 19
utledning
M-, 19
$\Re$-, 18
ensrettet M-, 19
utledningsforekomst, 95
kritisk, 95
opphørbar, 95
opphørt, 95
oppstå, 95
samme, 95

variabelbevarende, 84
variabelsymbol, 13
vedvarende regler, 55
vedvarende ligninger, 55
velfundert, *se* relasjon