UNIVERSITETET I OSLO
Institutt for lingvistiske fag

Mekanisert snittsøk i elementær tallteori

Hovedfagsoppgave

Gyrd Brændeland

Mars 2003

Tittelen på avsnitt 10.3.1 i innholdsfortegnelsen er rettet i forhold til den trykte utgaven.

Sammendrag

Jeg viser at man for enhver funksjon f i en delmengde av de Kalmárelementære funksjonene kan konstruere bevis for utsagn på formen $N(f(m_1, \ldots, m_n))$ med høyde lineær i $m_1, \ldots, m_n$. Delmengden består av de elementære funksjonene, unntatt modifisert subtraksjon og funksjoner hvor modifisert subtraksjon inngår i definisjonen. For enkelhets skyld kaller jeg delmengden av de elementære funksjonene strengt voksende. Uttrykket $N(f(m_1, \ldots, m_n))$ uttrykker at f er definert for argumentene $m_1, \ldots, m_n$. De korte utledningene oppnås ved å bruke induktive predikater om funksjonene, introdusert av Jervell og Zhang [15], som snittformler. Uten snitt har en utledning av $N(f(m_1, \ldots, m_n))$ minst like stor høyde som tallverdien av $f(m_1, \ldots, m_n)$. Jeg gir en algoritme for å søke etter snitt ut i fra definisjonen av den funksjonen som forekommer i utsagnet man vil bevise. Algoritmen er integrert i en modifisert versjon av den automatiske teorembeviseren PESCA.

Videre viser jeg at hvis det finnes et bevis av $N(f(m_1, \ldots, m_n))$ med høyde lineær i $m_1, \ldots, m_n$, så er f elementær. I dette beviset benytter jeg at snitteliminasjonsteoremet gir et elementært bånd på snitteliminasjon, når snittkompleksiteten er kjent.

For å kunne benytte snitteliminasjonsteoremet har jeg definert et bevissystem for første ordens aritmetikk hvor aksiomer er omskrevet til ikkelogiske slutningsregler. Metoden, introdusert av Negri og von Plato [22], gir systemer som bevarer snitteliminasjon, i motsetning til tradisjonelle aksiomsystemer hvor full snitteliminasjon ikke er mulig. Systemet inneholder ikke induksjonsaksiomet.

Takksigelser

Det kan både være en fordel og en ulempe å velge et tema for hovedfagsoppgaven som ens veileder er engasjert i. Veilederen har god kjennskap til området, men kan ha sterke meninger om hvordan oppgaven skal løses. I arbeidet med denne oppgaven har jeg bare opplevd fordelene. Min veileder Herman Ruge Jervell har ingen kjepphester i forhold til eget materiale. Han er åpen for å høre på nye idéer og innfallsvinkler. I tillegg har han god oversikt over logikkfaget. Man kan alltid komme til Jervell med spørsmål. Da vet han som regel svaret på det man lurer på og har gjerne en bok med et kapittel om akkurat dét stående i sitt rikholdige bibliotek.

Delen om subrekursjonsteori i oppgaven kom i havn takket være Lars Kristiansen som i praksis har fungert som biveileder på dette feltet. Kristiansens bidrag begynte med at han på strak arm leverte et innføringskompendium i subrekursjonsteori da han hørte at noen av Jervells studenter var interessert i emnet. Han foreslo en tilnærming til startpunktet for oppgaven – å vise at de primitivt rekursive funksjonene er bevisbart beregnbare i en elementær tallteori (avsnitt 3.3). Kristiansen forklarte sentrale bevismetoder i subrekursjonsteori og skisserte gangen i et bevis for den andre delen av hovedresultatet i oppgaven: Hvis en funksjon er induktiv, så er den elementær (kapittel 7 og 8).

Sara Negri har også hatt avgjørende betydning for vinklingen på oppgaven. Jervell introduserte meg for Negri og Jan von Platos system for å overføre aksiomer til ikkelogiske regler. Deres metode gir systemer som bevarer snitteliminasjon. Denne egenskapen ved bevissystemer med regler gjør at de egner seg for strukturell analyse, i motsetning til tradisjonelle aksiomsystemer hvor snitteliminasjon ikke er mulig. Negri hjalp meg å tilpasse systemet deres til en teori for primitiv rekursiv aritmetikk. Hun har dessuten lest og gitt grundige tilbakemeldinger på avsnitt 2.2 og kapittel 3. Derfor er disse kapitlene skrevet på engelsk. Hun kom inn i arbeidet på et tidspunkt hvor jeg allerede hadde begynt å skrive resten av oppgaven på norsk. Jeg vil også takke også Negri og von Plato for deres gjestfrihet da jeg besøkte dem i Riihimaki sommeren 2001.

Bjarte M. Østvold har bidratt med typografi og layout i LaTeX. Han har vært min lokale hacker i tekniske problemer med LaTeX og det funksjonelle programmeringsspråket Haskell. Østvold har dessuten lest igjennom og gitt grundig tilbakemelding på alle kapitlene. Han har også deltatt i mange givende diskusjoner om emner relatert til oppgaven. Østvold har støttet meg gjennom hele arbeidet.

Takk til Wenhui Zhang for innblikk i hans tidlige upubliserte arbeider om induktive predikater; Stål Aandera for et interessant spørsmål å bryne predikatene på; Aarne Ranta for hjelp til å komme i gang med teorembeviseren PESCA; personer tilknyttet logikkmiljøet ved Universitetet i Oslo som har svart på spørsmål relatert til oppgavens tema, samtidige hovedfagstudenter ved Språk, logikk og informasjon for givende diskusjoner, venner og Pappa for oppmuntring og støtte og lille Stål som har holdt meg med selskap i siste del av arbeidet.

Innhold

Figurer

Kapittel 1

Introduksjon

. . . vissheten om at det av og til, etter at man har møtt veggen i dagevis, plutselig kommer et slikt stort øyeblikk hvor man ser løsningen og får lyst til å reise seg på lesesalen og rope "Yeah!".

– Elise Øby, Matemagiske øyeblikk, *Apollon* 3-4, 2000

I bevisteori har det pågått et arbeid for å avgrense hvilke funksjoner som er "bevisbart totale i praksis". Oppgaven min bygger videre på dette arbeidet. Idéen er å bruke snittformler for å korte ned utledninger i predikatlogikk. Snittintroduksjon i klassisk logikk kan sammenlignes med bruk av lemmaer eller hjelpesetninger i matematiske bevis. Temaet for oppgaven ligger i skjæringsfeltet mellom bevisteori og subrekursjonsteori. Et formelt bevis er typisk gitt i et formelt system som for eksempel sekventkalkyle, beskrevet i avsnitt 2.2. Enkelt sagt består et formelt system av en mengde regler for å regne på symboler. Rekursjonsteori handler om de beregnbare funksjonene, det vil si funksjoner som kan beregnes av en algoritme. Selv om en funksjon er beregnbar i prinsippet er den ikke nødvendigvis beregnbar i praksis. Eksempel på en funksjon som bare er beregnbar i prinsippet, er en som raskt krever flere skritt enn antall atomer i universet på å beregne resultatet.

1.1 Motivasjon

Motivasjonen bak arbeidet for å avgrense de praktisk bevisbart beregnbare funksjonene er dels bevisteoretisk og dels filosofisk. Den bevisteoretiske motivasjonen er å gi en karakteristikk av de Kalmárelementære funksjonene i en elementær tallteori, det vil si en første ordens tallteori om de elementære

funksjonene. Ved å bruke hjelpesetninger eller snitt kan man korte ned formelle bevis om utrykk på formen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, hvor f er elementær og N er predikatet for at noe er et naturlig tall. En karakteristikk går to veier. For det ene må man vise at hvis f er elementær, så er det mulig å finne korte bevis for $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, for alle $m \in N$. For det andre må man vise at hvis dette er mulig, så er f elementær.

En annen motivasjon er å finne metoder for å mekanisere snittsøk i induktive strukturer. Bruk av snitt har vist seg vanskelig å få til i automatisk bevissøk, fordi det innebærer et element av gjetting. Snittformelen kan være en hvilken som helst utledbar formel. En fordel med snitt er at det kan redusere lengden på bevis dramatisk. Derfor kan det være nyttig å se på metoder for å finne snitt.

Det filosofiske aspektet handler om å anskueliggjøre hvordan store tall må beskrives indirekte for å bli håndterlige eller begripelige om man vil. Jervell [14] trekker paralleller til Archimedes indirekte metode for å telle antall sandkorn i universet. Archimedes benytter seg av myriader ($= 10^4$) – antall maur i en maurtue. Ved å bruke myriadene som byggesteiner kommer han fram til et anslag på antall sandkorn i universet. Jeg har i denne oppgaven kun befattet meg med de bevisteorietiske aspektet og vil ikke komme ytterligere inn på de filosofiske motivene.

1.2 Litt historikk

På 1930-tallet ble det gjort flere forsøk på å finne en enkel matematisk karakteristikk av klassen av beregnbare funksjoner. Det vil si man ønsket å klassifisere de funksjonene som det er mulig å beregne for en maskin, ved at det finnes en entydig oppskrift som maskinen kan følge. Dette arbeidet resulterte i flere modeller for beregnbarhet, deriblant λ-kalkyle (Alonzo Church) Kleene-rekursjon (Stephen Cole Kleene) og Turingmaskiner (Alan M. Turing). De ulike modellene viste seg å karakterisere den samme klassen av funksjoner. Dette fikk Church til å fremsette tesen om at enhver beregnbar funksjon kan beregnes av en Turing-maskin (og dermed innen enhver av de andre formalismene), kjent som Church-Turings tese.

Selv om modellene til Church, Kleene og Turing gir en avgrensning av de prinsipielt beregnbare funksjonene, fins det funksjoner som vokser så raskt at de ikke er praktisk beregnbare med noen kjente metoder. Det er ikke mulig i det generelle tilfellet å avgjøre hvorvidt en funksjon f med et input m terminerer eller ikke. Dette resultatet er kjent som stoppeproblemet og viser en viktig begrensning på hva som er mulig å beregne mekanisk.

Siden Church framla sin tese har har mye arbeid vært lagt ned i å dele

de beregnbare funksjonene inn i mindre klasser etter som hvor komplekse de er. Det finnes ulike måter å beskrive kompleksitet av en funksjon på. Hvis man beskriver funksjoner ved hjelp av teorien om Turingmaskiner kan man måle kompleksitet enten ved rom; hvor mye tape som brukes til å beregne en funksjon, eller tid; hvor tid måles i antall skritt beregningen tar. Noen kjente kompleksitetsklasser er LINSPACEF (klassen av funksjoner som kan beregnes av Turingmaskiner i lineært rom) P, NP, EXP og så videre. Innen rekursjonsteori har man arbeidet med å inndele funksjonene i hierarkier, hvor hver klasse gjerne genereres av en mengde definisjonsskjemaer. Den grunnleggende idéen bak hierarkiteori beskrives av Rose [30]. Man konstruerer en muligens transfinit sekvens av klasser av funksjoner slik at hver klasse er inneholdt i den neste. Sekvensen begynner med de primitivt rekursive funksjonene eller de elementære funksjonene eller en liknende klasse, og unionen av hele sekvensen er klassen av rekursive funksjoner. Grzegorczyks hierarki for de primitivt rekursive funksjonene blir beskrevet nærmere i kapittel 2.

I tillegg til å dele de ulike beskrivelsene av beregnbare funksjoner inn i mindre grupper har man vært opptatt av å undersøke om ulike undergrupperinger beskriver samme klasse. I 1963 viste Ritchie [28] at klassen $\mathcal{E}^2$ i Gregorczyk-hierarkiet karakteriserer klassen LINSPACEF. Cobham [4] har gitt en maskinuavhengig karakteristikk av klassen PTIME. For en grundigere gjennomgang av forholdet mellom subrekursjonsteoretiske funksjonsklasser og kompleksitetsklasser kan man lese Clote [3].

Arbeidet med å gi rekursjonsteoretiske karakteristikker av kompleksitetsklasser har inspirert liknende arbeid i bevisteori. Ved å se på hva som kan utledes og hva som ikke kan utledes i en gitt første ordens tallteori kan subrekursjonsteori benyttes til å karakterisere uttrykkskraften til teorien. Man sier gjerne at man har en bevisteoretisk karakteristikk av en funksjonsklasse. Tidligere har man sett på sammenhengen mellom kompleksitetsklasser og bundet aritmetikk. Bundet aritmetikk er svakere utgaver av Peano-aritmetikk hvor man har induksjon over bundne kvantorer. Peano-aritmetikk beskrives i avsnitt 2.3. For eksempel er det vist at de primitivt rekursive funksjonene er nøyaktig de bevisbart beregnbare i teorien $I\Sigma_1$ [8] (Robinson-aritmetikk pluss induksjon for Σ_1-formler, definisjon 2.12). I avsnitt 10.2 om relatert arbeid beskrives nyere arbeid som er gjort for å gi rekursjonsteoretiske karakteristikker av kompleksitetsklasser og bevisteoretiske karakteristikker av rekursjons- og kompleksitetsklasser.

1.3 Oversikt

Kapittel 2 gir en kort innføring i grunnleggende begreper i rekursjonsteori og bevisteori. Kapitlet gir også en oversikt over noen kjente resultater som brukes senere i oppgaven. Bevissystemet defineres i kapittel 3.

I de neste tre kapitlene presenteres idéen bak induktive predikater (kapittel 4) og beviset for den ene delen av hovedresultatet i oppgaven: Hvis funksjonen f er elementær kan man finne utledninger av utsagn på formen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, med høyde lineær i $m_1, \ldots, m_n$ (kapittel 5). De korte utledningene oppnås ved å bruke induktive predikater om funksjonene, introdusert av Jervell og Zhang [15], som snittformler. Ifølge deres definisjon er et predikat P induktivt hvis følgende er utledbart:

$$P(0)$$
$$\forall x(P(x) \supset P(\mathcal{S}(x)))$$

Uten snitt har en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ minst like stor høyde som tallverdien av $f(\bar{m}_1, \ldots, \bar{m}_n)$. Jeg gir en algoritme for å søke etter snitt. Algoritmen bruker definisjonen av den funksjonen som forekommer i utsagnet man vil utlede. Videre viser jeg at algoritmen finner snittformler for alle de elementære strengt voksende funksjonene. I kapittel 6 gis et eksempel på forskjellen på direkte og indirekte bevis av at fakultetsfunksjonen er veldefinert.

Beviset for den andre delen av hovedresultatet: Hvis en funksjon er induktiv, så er den elementær, presenteres i de to neste kapitlene. Først viser jeg at bevistrær i PRA kan kodes som tall (kapittel 7 og vedlegg C). Jeg definerer en variant av Kleenes $\mathcal{T}$-predikat for å sjekke om et tall koder en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$. Definisjonen og beviset er nokså omstendelig. Kapittel 8 inneholder selve beviset for at f er elementær hvis den er induktiv. Beviset går ut på å konstruere en elementær algoritme for å beregne $f(m_1, \ldots, m_n)$ fra en utledning $\mathcal{D}$ av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, når høyden av $\mathcal{D}$ er lineær i $m_1, \ldots, m_n$. Her benytter jeg blant annet at snitteliminasjonsteoremet gir et elementært bånd på eliminasjon av snitt når snittkompleksiteten er kjent.

Algoritmen definert i kapittel 5 er integrert i en modifisert versjon av den automatiske teorembeviseren PESCA, kalt PESCA*. Teorembeviseren beskrives i kapittel 9. Jeg har også brukt PESCA* til å generere de fleste formelle utledningene i denne oppgaven, se figurer 6.1, 6.2 og 6.3 for direkte utledninger av henholdsvis $N(+mn)$, $N(\times mn)$ og $N(n!)$ og figur 6.7 for en indirekte utledning av $N(n!)$. Tillegg A gir utledninger av snittformler over noen induktive predikater, generert i PESCA*. Det er omstendelig å lese og skrive utledninger i sekventkalkyle, men de er enkle å verifisere maskinelt.

Skriving av utledninger i sekventkalkyle er derfor en oppgave som egner seg bra for maskiner. Utledninger i sekventkalkyle kan imidlertid fort ta stor plass. Jeg har derfor nøyd meg med å gjengi deler av bevisene som er generert i PESCA*, mens repetisjoner av delbevis og overganger som jeg har vist er mulige, er angitt med metanotasjon.

I kapittel 10 tar jeg opp et spørsmål som har vært stilt i forbindelse med arbeidet, presenterer relatert arbeid og vurderer retninger for videre arbeid.

1.4 Resultat

Som nevnt i avsnitt 1.1 var det et mål i oppgaven å vise at man kan finne korte bevis for utsagn av typen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ hvis og bare hvis f er elementær. Dette viste seg å ikke være mulig ved metodene til Jervell og Zhang. Isteden viser jeg 1) Hvis f er elementær og f ikke er funksjonen for modifisert subtraksjon og denne ikke inngår i definisjonen av f, så er f induktiv i en utvidet versjon av bevissystemet PRA, definert i kapittel 3. Jeg har ikke klart å gi en god betegnelse av denne delmengden av de elementære funksjonene. For enkelhets skyld vil jeg heretter si at en funksjon f er induktiv hvis den er elementær og strengt voksende (definisjon 2.5). Da får jeg med for få funksjoner, siden for eksempel projeksjonsfunksjonen også er induktiv (lemma 5.2), men jeg får i hvert fall ikke med for mange. At en funksjon er induktiv innebærer at man kan gi et kort bevis for $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ ved hjelp av snitt. 2) Hvis funksjonen f er induktiv så er den elementær. Dette resultatet er ikke symmetrisk og dermed svakere enn det som var ønsket.

Funksjonen modifisert subtraksjon skaper trøbbel for det ønskede hovedresultatet. For å vise at denne er induktiv legger Jervell og Zhang til aksiomene:

$$\begin{aligned} \dot{-}0y &= 0 \\ \dot{-}Sxy = 0 \vee \dot{-}Sxy &= S\dot{-}xy \end{aligned}$$

Problemet er at de ved å legge til disse aksiomene får med alt for mange funksjoner. I avsnitt 5.4 viser jeg at det finnes en funksjon f som ikke er elementær og hvor man kan finne en utledning med konstant lengde av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ i PRA utvidet med disse aksiomene. Faktisk finnes det uendelig mange slike ikke-elementære funksjoner, som har korte bevis for at de er beregnbare.

I konklusjonen (kapittel 10) foreslår jeg en alternativ framgangsmåte for å oppnå en bevisteoretisk karakteristikk av de elementære funksjonene.

1.5 Hva er nytt?

Bevissystemet jeg bruker er definert ved sekventkalkyle utvidet med definerende likninger for de primitivt rekursive funksjonene omskrevet til ikkelogiske slutningsregler, etter en ny metode introdusert av Negri og von Plato [22]. Systemet inneholder ikke induksjonsaksiomet. Dette gjør systemet svakt men har den fordelen at man kan oppnå full snitteliminasjon på endelige utledninger. Kreisel [16] har vist at den vanlige prosedyren for snitteliminasjon ikke fungerer i første ordens aritmetikker med full induksjon. For å vise konsistens av Peano aritmetikk må man overføre en utledning som bruker induksjonsaksiomet til en uendelig utledning. Deretter kan man eliminere snitt fra den uendelige utledningen. Snitteliminasjonsteoremet holder heller ikke for utledninger i sekventkalkyle utvidet med ikkelogiske aksiomer, som er en vanlig måte å definere tallteori på. Reglene i det nye systemet defineres imidlertid på en måte som tillater full snitteliminasjon. Dette har nyttige strukturelle anvendelser for systemet, blant annet bevares det øvre båndet på snittfrie utledninger som følger fra snitteliminasjonsteoremet.

Jeg gir et detaljert bevis for at enhver strengt voksende elementær funksjon f er induktiv i en første ordens tallteori (PRA+$AddAss$,$TimesAss$) med regler for likhet, definerende likninger for de primitivt rekursive funksjonene og assosiativitet av addisjon og multiplikasjon. Jervell og Zhang [15] gir en strategi for å finne induktive predikater for alle Kalmárelementære funksjoner, men de definerer ikke en fullstendig tallteori. De definerer predikater for addisjon, multiplikasjon, eksponensialfunksjonen og modifisert subtraksjon og viser lukningsegenskaper under de ulike skjemaene. Jeg bygger videre på deres strategi, men enkelte av predikatene definert i kapittel 5 har en litt annen form, se avsnitt 4.2. Jeg viser også at aksiomene Jervell og Zhang legger til for modifisert subtraksjon fører til at for mange funksjoner kan gis korte bevis for at de er totale.

Beviset for at hvis en funksjon f er induktiv så er den elementær, er gitt i detalj. Jervell og Zhang [15] gir ikke dette beviset, men antyder at det kan vises ved hjelp av snitteliminasjonsteoremet.

Videre har jeg etterstrebet en mest mulig mekanisk strategi for å generere induktive predikater. Dette har gjort det enkelt å implementere strategien i et Haskell-program og integrere det i en interaktiv teorembeviser. Programmet som genererer predikater for funksjoner har en semantisk begrensning: Hvis funksjonen f er definert ved rekursjon over en funksjon g må systemet inneholde aksiomer for assosiativitet av h. For øvrig genereres predikatene ut i fra syntaktiske kriterier og har lik syntaktisk form.

Kapittel 2

Teoretisk bakgrunn

I denne oppgaven gir jeg et øvre bånd på høyden av utledninger av bestemte utsagn. Utsagnene dreier seg om typetilhørighet av en klasse av subrekursive funksjoner. Kjernen av oppgaven grenser mellom bevisteori og subrekursjonsteori.

Dette kapitlet inneholder noen sentrale resultater i subrekursjonsteori og bevisteori, som blir referert til i senere kapitler.

2.1 Subrekursjonsteori

Rekursjonsteori handler om de beregnbare funksjonene. En beregnbar funksjon kan uformelt beskrives som en funksjon beregnbar av en algoritme. Rogers [29] gir en uformell beskrivelse av begrepet algoritme. Det finnes noen opplagte kriterier som det hersker enighet om blant matematikere.

- En algoritme gis som en endelig mengde av instruksjoner.
- Det finnes en regne-agent, vanligvis et menneske, som kan handle i tråd med instruksjonene og utføre beregningene.
- Man har fasiliteter for å utføre, lagre og sjekke opp tidligere skritt i beregningen.
- Beregningen er diskret og gjennomføres uten kontinuerlige eller analoge redskaper.
- Beregningen er deterministisk og utføres uten bruk av vilkårlige metoder, som for eksempel terningkast.

Videre kan en legge til at beregningen må være mulig å gjennomføre i et endelig antall skritt. I tillegg finnes kriterier som det ikke er opplagt om

må oppfylles for at noe skal kunne kalles en algoritme. Et kriterium som det kanskje kan være uenighet om mellom databehandlere og matematikere er om det skal finnes noen bånd på antall skritt i beregningen. Det vil si at man på forhånd kan avgjøre hvor mange skritt en algoritme vil bruke på å beregne en funksjon med hensyn på et gitt argument. For klassen av beregnbare funksjoner er ikke dette kriteriet oppfylt.

Definisjon 2.1 (Turingberegnbar funksjon) *At en funksjon f er Turingberegnbar vil si at det finnes en maskin M slik at for alle argumenter x i domenet D til f, $M(x) = f(x)$. Det vil si for alle $x \in D$ så stopper M med input x etter et endelig antall skritt med output $f(x)$ på tapen.*

Stoppeproblemet for Turingmaskiner sier at det ikke er mulig å avgjøre i det generelle tilfellet hvorvidt en maskin stopper på et vilkårlig input. Av dette følger at det ikke er mulig å avgjøre generelt om en funksjon er turingberegnbar og dermed heller ikke om den er beregnbar. Hadde man krevd at alle beregninger skulle stanse etter et bestemt antall skritt, så kunne man avgjort hvorvidt en funksjon er beregnbar.

2.1.1 Primitiv rekursjon

I denne oppgaven ser jeg på en delmengde av de rekursive funksjonene. En funksjon kalles rekursiv hvis den beregnes av en algoritme som er definert ved et kall på seg selv. De subrekursive eller primitivt rekursive funksjonene skiller seg fra de beregnbare funksjonene ved at det alltid er mulig å beregne hvor mange skritt som trengs for å beregne f i et gitt argument. De er totale og beregnbare funksjoner over heltall. Men det finnes totale og beregnbare funksjoner som ikke er primitivt rekursive.

I dette avsnittet introduseres en del grunnleggende begreper og egenskaper ved de primitivt rekursive funksjonene som jeg får bruk for i de senere kapitlene. Grunnleggende resultater som allerede er kjent vises i liten grad her. Beviser for lemmaer og påstander i dette avsnittet finnes blant annet i Rose [30], Sudkamp [32] og Kristiansen [17].

Nullfunksjonen $\mathcal{O}$, etterfølgerfunksjonen $\mathcal{S}$ og projeksjonsfunksjonene $\mathcal{I}_i^n$ er de primitivt rekursive initialfunksjonene. En funksjon f definert ved hjelp av funksjoner g, h og j og skjemaene for komposisjon eller primitiv rekursjon, er primitivt rekursiv hvis g, h og j er primitivt rekursive. Skjemaene er definert som følger:

Komposisjon:

$$f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

Primitiv rekursjon:

$$f(x_1, \ldots, x_n, 0) = g(y_1, \ldots, y_n)$$
$$f(x_1, \ldots, x_n, Sy) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$$

Definisjon 2.2 (Klassen $\mathcal{P}$ av primitivt rekursive funksjoner) *er tillukkingen av de rekursive initialfunksjonene under definisjonsskjemaene komposisjon og primitiv rekursjon.*

Definisjon 2.3 *En funksjon f er definert ved begrenset rekursjon over g, h og j hvis f er definert ved primitiv rekursjon og*

$$f(x_1, \ldots, x_n, y) \leq j(x_1, \ldots, x_n, y)$$

Lemma 2.4 *Subtraksjonsfunksjonen $\dot{-}$ er primitivt rekursiv.*

Bevis. Modifisert subtraksjon defineres ved primitiv rekursjon over forgjengerfunksjonen, som kan defineres ved primitiv rekursjon over projeksjonsfunksjonen.

$$P(0) = \mathcal{O}$$
$$P(Sy) = \mathcal{I}_1^2(y, P(y))$$
$$\dot{-}x0 = \mathcal{I}_1^1(x)$$
$$\dot{-}xSy = P(\dot{-}xy)$$

□

Definisjon 2.5 (Strengt voksende funksjon) *En n+1-ær funksjon f kalles strengt voksende hvis for alle tall $m_1, \ldots, m_n, n, p \in N$*

$$n < p \Rightarrow f(m_1, \ldots, m_n, n) < f(m_1, \ldots, m_n, p).$$

Notasjon

En rekke vanlige aritmetiske funksjoner kan konstrueres ved primitiv rekursjon. For eksempel er funksjonen + primitivt rekursiv:

$$+(0, y) = \mathcal{I}_1^1(y)$$
$$+(Sx, y) = h(+(x, y), x, y), h(x, y, z) = \mathcal{S}(\mathcal{I}_1^3(x, y, z))$$

Hvis man følger skjemaene for komposisjon og primitiv rekursjon helt presist, slik som her, kan definisjonene bli veldig omstendelige og vanskelige å lese. Alle funksjonene i eksemplene som følger kan skrives primitivt rekursivt på tilsvarende måte, men jeg vil ofte bruke standardnotasjon for enkelhets skyld. Funksjonene skrives med prefiksnotasjon, slik som i eksemplene over.

2.1.2 Elementære funksjoner

Den viktigste metoden for å konstruere subrekursive funksjoner er ved repetert iterasjon av en operasjon som allerede er definert. Addisjon er iterasjon av etterfølgerfunksjonen, multiplikasjon er iterasjon av addisjon og så videre.

Definisjon 2.6 *La f være en primitivt rekursiv funksjon. Da kan man definere en ny funksjon ved bundet sum eller bundet produkt over f:*

$$\textit{bundet sum: } \sum_{i\leq n} f(\vec{x},i) = f(\vec{x},0)+\cdots+f(\vec{x},n)$$
$$\textit{bundet produkt: } \prod_{i\leq 0} f(\vec{x},i) = f(\vec{x},0)+\cdots+f(\vec{x},n)$$

Lemma 2.7 *De primitivt rekursive funksjonene er lukket under bundet sum og bundet produkt.*

Dette kan man forsikre seg om ved å sjekke at følgende definisjoner er primitivt rekursive:

$$\sum_{i\leq 0} f(\vec{x},i) = f(\vec{x},0)$$
$$\sum_{i\leq Sy} f(\vec{x},i) = f(\vec{x},Sy)+\sum_{i\leq y} f(\vec{x},i)$$
$$\prod_{i\leq 0} f(\vec{x},i) = f(\vec{x},0)$$
$$\prod_{i\leq Sy} f(\vec{x},i) = f(\vec{x},Sy)+\prod_{i\leq y} f(\vec{x},i)$$

Ved hjelp av disse skjemaene kan man definere en ny, viktig klasse av funksjoner.

Definisjon 2.8 (De Kalmárelementære funksjonene, $\mathcal{E}'$) *De Kalmárelementære initialfunksjonene er nullfunksjonen $\mathcal{O}$, etterfølgerfunksjonen $\mathcal{S}$, projeksjonsfunksjonene $\mathcal{I}_i^n$, addisjon $+$ og modifisert subtraksjon $\dot{-}$. Klassen av Kalmárelementære funksjoner $\mathcal{E}'$ er tillukkingen av de Kalmárelementære initialfunksjonene under bundet sum, bundet produkt og komposisjon.*

De Kalmárelementære funksjonene har sitt navn etter den ungarske matematikeren László Kalmár. Han var den første til å introdusere denne klassen av funksjoner, i 1943. $\mathcal{E}'$ inneholder de fleste nyttige tallteoretiske funksjonene over heltall. De Kalmárelementære funksjonene kalles heretter stort sett elementære.

Teorem 2.9 *[30, s. 11] Hvis f er en primitivt rekursiv funksjon og f er bundet av en elementær funksjon g og dens hjelpefunksjoner er elementære, så er f elementær.*

Regning med logiske operatorer

Relasjoner over heltall kan representeres ved funksjoner med verdiområde 0,1.

Definisjon 2.10 *La R være en relasjon over naturlige tall. f kalles den karakteristiske funksjonen til R dersom*

$$f(x_1,\ldots,x_n) = 1 \quad \textit{når } R(x_1,\ldots,x_n) \textit{ tolkes som sann i standardmodellen}$$
$$f(x_1,\ldots,x_n) = 0 \quad \textit{når } R(x_1,\ldots,x_n) \textit{ tolkes som usann i standardmodellen}$$

På denne måten kan man regne på logiske utsagn, noe som blant annet kan brukes i søkealgoritmer[1].

Lemma 2.11 *De elementære relasjonene er lukket under de utsagnslogiske operatorene.*

Bevis. La p og q være de karakteristiske funksjonene til relasjonene P og Q. Følgende logiske operasjoner kan defineres ved komposisjon over de elementære funksjonene $\dot{-}, +$ og $\times$.

$$\begin{aligned} \neg P &= \dot{-}(1,p) \\ P \vee Q &= \dot{-}(1,\dot{-}(1,+(p,q))) \\ P \wedge Q &= \times(p,q) \end{aligned}$$

Alle andre utsagnslogiske operatorer kan defineres ved $\neg, \vee$ og komposisjon. □

På samme måte kan man definere elementære relasjoner. Relasjonen *sign* gir 0 hvis argumentet er 0 og 1 ellers. Relasjonen *cosg* gir 0 hvis argumentet er 0, 1 ellers. Relasjonene $<$ og $>$ er de vanlige ordningsrelasjonene og *eq* er likhetsrelasjonen.

$$\begin{aligned} sign(x) &= \dot{-}(1,\dot{-}(1,x)) \\ cosg(x) &= \dot{-}(1,x) \\ <xy &= sign(\dot{-}(y,x)) \\ >xy &= sign(\dot{-}(x,y)) \\ eqxy &= cosg(+(<xy, >xy)) \end{aligned}$$

[1] Da matematikerne Haskell B. Curry og Reuben Louis Goodstein viste at konnektivene i utsagnslogikk kan defineres i en elementær aritmetikk valgte de å la 0 representere sann og 1 falsk. Jeg velger imidlertid å holde meg til den mer moderne tradisjonen innen teoretisk databehandling hvor 0 er falsk og 1 er sann.

Definisjon 2.12 (Bundne kvantorer) *1.*

$\exists i \leq n P(i) \equiv$ *Det eksisterer en i mindre enn n slik at $P(i)$ holder.*
$\forall i \leq n P(i) \equiv$ *For alle i mindre enn n holder $P(i)$.*

2. *(a) En formel A er Δ_0 hvis den inneholder bare bundne kvantorer.*
(b) $\Sigma_0 = \Pi_0 = \Delta_0$
(c) Hvis en formel A er Π_n er $\exists x A$ Σ_{n+1}.
(d) Hvis en formel A er Σ_n er $\forall x A$ Π_{n+1}.

Lemma 2.13 *De elementære relasjonene er lukket under de bundne første ordens kvantorene $(\exists i \leq n)$ og $(\forall i \leq n)$.*

Bevis. La p være den karakteristiske funksjonen til den elementære relasjonen $P(\vec{x}, y)$. De karakteristiske funksjonene for bundet $\exists$ og bundet $\forall$ defineres ved:

$$(\forall i \leq n)[P(\vec{x}, i)] = \prod_{i \leq n} p(\vec{x}, i)$$
$$(\exists i \leq n)[P(\vec{x}, i)] = \neg(\forall i \leq n)[\neg P(\vec{x}, i)]$$ □

Begrenset søk

For å kunne vise at totale divisjonsfunksjoner er elementære må jeg introdusere en ny operator, μ-operatoren som brukes til begrenset søk. Det å dele et heltall på et annet er en mer komplisert operasjon enn å legge to tall sammen. Det første innebærer en form for søk, fordi man spør; hvor mange ganger kan jeg dele x på y? Hvis man skulle programmere en funksjon for heltallsdivisjon i et funksjonelt programmeringsspråk som Scheme, ville man kanskje trekke y fra x helt til x ble mindre enn y og samtidig telle antall subtraksjoner. Når man skal overholde skjemaet for primitiv rekursjon kan man imidlertid ikke bruke en slik tilnærming. Antall kall på en funksjon er gitt på forhånd av rekursjonsargumentet. I stedet regner man på verdier fra passende karakteristiske funksjoner ved hjelp av bundet minimalisering.

Definisjon 2.14 (Bundet minimalisering) *La P være en elementær relasjonen og p være den karakteristiske funksjonen til P. Da er*

$$\mu i \leq y[p(x_1, \ldots, x_n, i)]$$

det minste tallet i slik at $p(x_1, \ldots, x_n, i) = 1$. Det vil si $P(x_1, \ldots, x_n, i)$ er sann.

Lemma 2.15 *De elementære funksjonene er lukket under bundet minimalisering.*

Bevis. Bundet minimalisering kan defineres ved:

$$\mu i \leq y[p(i, x_1, \ldots, x_n)] = \sum_{z \leq y} \prod_{i \leq z} cosg(p(i, x_1, \ldots, x_n))$$

Siden *cosg* er elementær og de elementære funksjonene er lukket under bundet sum og bundet produkt, er $\mu i \leq y[p(i, \vec{x})]$ elementær når p er det. □

Ved hjelp av begrenset minimalisering kan man definere en rekke divisjonsfunksjoner. For å gjøre heltallsdivisjon total kan man sette $x/0 = 0$.

Eksempler på noen elementære divisjonsfunksjoner:

$$\begin{aligned}
quo(x, y) &= \times(sign(y), \mu i \leq y[> \times(+(z, 1), y)x]) \\
rem(x, y) &= \dot{-}(x, \times(y, quo(x, y))) \\
divides(x, y) &= \begin{cases} 1 \text{ hvis } x > y, y > 0, \text{ og } y \text{ ikke deler } x \\ 0 \text{ ellers} \end{cases} \\
&= eq(rem(x, y), 0) \wedge sign(x) \\
ndivisors(y) &= \sum_{i=0}^{y} divides(y, i) \\
prime(y) &= eq(ndivisors(y), 2)
\end{aligned}$$

For å få alle de beregnbare funksjonene kan man legge skjemaet for ubegrenset minimalisering (eller bare minimalisering) til de primitivt rekursive funksjonene.

Primtallskoding

Ved hjelp av operatorene definert over kan en definere relasjoner og funksjoner til bruk i koding og dekoding av bevis. Fra lemmaene over kan man vise at følgende relasjoner og funksjoner er elementære:

$nprimes(y) = \Sigma_{i \leq y} prime(i)$, gir antall primtall mindre enn eller lik y.
$p_n = (\mu i \leq 2^{n+1})[nprimes(i) = n + 1]$ som gir det n'te primtallet.

Definisjon 2.16 (Sekvenstall) *La p_i være det i'te primtallet. Sekvenstallet $\langle x_1, \ldots, x_n \rangle$ er gitt ved $p_0^n \times p_1^{x_1} \times \cdots \times p_n^{x_n}$.*

Eksponenten til k i dekomposisjonen av y fås fra funksjonen:

$$exp(y, k) = \mu x_{\leq y}[k^x | y \wedge \neg(k^{x+1} | y)]$$

I kapittel 7 defineres et dekodingssystem som gjør bruk av følgende elementære funksjoner og predikater:

$$(x)_n = exp(x, p_n);\ \text{den } n\text{'te komponenten til } x$$
$$ln(x) = (x)_0;\ \text{antall koordinater i } x, \text{ hvis } x \text{ er et sekvenstall, 0 ellers}$$
$$Seq(x) = (\forall n \leq x)[n > 0 \wedge (x)_n \neq 0 \Rightarrow n \leq ln(x)].$$

Jeg får også bruk for en begrenset versjon av skjemaet for verdiforløprekursjon. La n og k være faste tall og la $j_i(y) \leq y$ for $i = 1, \ldots, k$. Da defineres f over funksjonene $g, h, j_1, \ldots, j_k$ ved skjemaet for verdiforløprekursjon:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$
$$f(x_1, \ldots, x_n, Sy) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, j_1(y)), \ldots, f(x_1, \ldots, x_n, j_k(y)))$$

Definisjon 2.17 (Bundet verdiforløprekursjon) *En funksjon f er definert ved bundet verdiforløprekursjon over funksjoner $g, h, j_1, \ldots, j_k$ og l hvis f er definert ved verdiforløprekursjon og*

$$f(x_1, \ldots, x_n, y) \leq l(x_1, \ldots, x_n, y)$$

Teorem 2.18 *De elementære funksjonene er lukket under bundet verdiforløprekursjon.*

Bevis. Anta at $g, h, j_1, \ldots, j_k$ og l er elementære og at $j_i(y) \leq y$ for $i = 1, \ldots, k$. La $\vec{x}$ stå for $x_1, \ldots, x_n$ og la f være gitt som i definisjonen av begrenset verdiforløprekursjon. Definer en ny funksjon

$$F(\vec{x}, 0) = p(0)^{g(\vec{x})}$$
$$F(\vec{x}, Sy) = F(\vec{x}, y) \times p(Sy)^{h(\vec{x}, F(\vec{x},y)[j_1(y)], \ldots, F(\vec{x},y)[j_k(y)])}.$$

Det følger av lemmaene over at F er primitivt rekursiv. Merk at alle hjelpefunksjonene er elementære. Hvis jeg kan vise at F er begrenset av en elementær funksjon følger det fra teorem 2.9 at F er elementær. Siden $j_i(y) \leq y$ for $i = 1, \ldots, k$, så følger det at

$$F(\vec{x}, y) = p(0)^{f(\vec{x},0)} \times p(1)^{f(\vec{x},1)} \times \cdots \times p(y)^{f(\vec{x},y)} = \prod_{i \leq y} p^{f(\vec{x},i)}$$

Ved antakelsen er $f(\vec{x}, y) \leq l(x_1, \ldots, x_n, y)$, så $\prod_{i \leq y} p^{f(\vec{x},i)} \leq \Pi_{i \leq y} p^{l(\vec{x},i)}$. La $m(\vec{x}, y) = \Pi_{i \leq y} p^{l(\vec{x},i)}$. Da følger $F(\vec{x}, y) \leq m(\vec{x}, y)$. Funksjonen m er elementær og F er dermed definert ved begrenset rekursjon over elementære funksjoner, så F er elementær. Funksjonen f er gitt ved komposisjonen $f(\vec{x}, y) = F(\vec{x}, y)[y]$. Dermed er f elementær. □

De elementære funksjonene er også lukket under bundne versjoner av andre definisjonsskjemaer, men det blir kun bruk for bundet verdiforløprekursjon i denne oppgaven.

2.1.3 Gregorczyk-hierarkiet

Mye aktivitet innen matematisk logikk siden Church framla sin tese har vært rettet inn mot å klassifisere de beregnbare funksjonene etter ulike kriterier. Ideen er å konstruere en sekvens av klasser av funksjoner med økende kompleksitet slik at hver klasse er inneholdt i den neste. Sekvensen begynner med de primitivt rekursive funksjonene eller de elementære eller en liknende klasse og unionen av alle klassene er de beregnbare funksjonene.

Den polske matematikeren Andrzej Gregorczyk introduserte i 1953 et hierarki over de primitivt rekursive funksjonene. Hans arbeid bidro til utviklingen av mer omfattende hierarkier som omtales av Rose [30]. Gregorczyk introduserte klassene $\mathcal{E}^r (r = 0, 1, 2 \ldots)$. Hver av klassene i hierarkiet er ekte inkludert i den påfølgende. Hierarkiet defineres ved:

$$G = \bigcup_{r<\omega} \mathcal{E}^r$$

G er da klassen av primitivt rekursive funksjoner.

Definisjon 2.19 *Sekvensen av primitivt rekursive funksjoner E_k for $k = 0, 1, 2, \ldots$ er gitt ved*

$$\begin{aligned} E_0(x, y) &= +xy \\ E_1(x) &= +x^2 2 \\ E_{k+2}(0) &= 2 \\ E_{k+2}(x+1) &= E_{k+1}(E_{k+2}(x)) \end{aligned}$$

Definisjon 2.20 *Sekvensen av subrekursive klasser $\mathcal{E}^r$ for $r = 0, 1, 2, \ldots$ er gitt ved*

$\mathcal{E}^0$ er tillukkingen av funksjonene $\mathcal{O}, \mathcal{S}, \mathcal{I}_i^n$ under komposisjon og begrenset rekursjon.

$\mathcal{E}^{r+1}$ er tillukkingen av funksjonene $\mathcal{O}, \mathcal{S}, \mathcal{I}_i^n$ og E_0 under komposisjon og begrenset rekursjon. $E_r, \mathcal{O}, \mathcal{S}, \mathcal{I}_i^n$ og E_0 kalles initialfunksjonene i $\mathcal{E}^{r+1}$.

Teorem 2.21 *[17, s. 19] La f være en funksjon i $\mathcal{E}^0$ med aritet n. Da finnes det faste tall i og k (der $1 \leq i \leq n$) slik at*

$$f(x_1, \ldots, x_n) \leq x_i + k.$$

Lemma 2.22 *[30, s. 36] $f \in \mathcal{E}^n \Rightarrow f^y \in \mathcal{E}^{n+1}$ for y en variabel.*

Det vil si at hvis en funksjon f er i en Gregorczyk-klasse, så kommer man et nivå opp ved å rekursere ubegrenset over f. For eksempel er funksjonen $+ \in \mathcal{E}^1$, mens funksjonen $\times$ som rekurserer over $+$ er i $\mathcal{E}^2$.

Et korollar av dette resultatet er at når g og h er i $\mathcal{E}^n$ og f er definert ved hjelp av ubegrenset rekursjon over g og h, så er $f \in \mathcal{E}^{n+1}$.

Eksempelvis kan man definere hypereksponensialfunksjonen ved hjelp av ubegrenset primitiv rekursjon over eksponensialfunksjonen:

Definisjon 2.23 (Hypereksponensialfunksjonen)

$$hyp(0, y) = y$$
$$hyp(Sx, y) = y^{hyp(x,y)}$$

Toertårn er et spesialtilfelle av hypereksponensialfunksjonen som det blir bruk for blant annet i snitteliminasjonsteoremet.

Definisjon 2.24 (Toertårn)

$$\beth^0(x) = x$$
$$\beth^{k+1}(x) = 2^{\beth^k(x)}$$

Toertårn gir et tårn av toere, hvor eksponeneroingen gjøres den vrange veien. Eksempelvis er

$$\beth^4(3) = 2^{(2^{(2^{(2^3)})})}.$$

Det kan vises at enhver elementær funksjon majoriseres av hypereksponensialfunksjonen.

Lemma 2.25 *[26, s. 100-103] For enhver elementær funksjon $f(m_1, \ldots, m_n)$ finnes en gitt m slik at for $\max(m_1, \ldots, m_n) > 1$ holder ulikheten*

$$f(m_1, \ldots, m_n) \leq hyp(m, \max(m_1, \ldots, m_n)).$$

Teorem 2.26 *[30, s. 33] Klassen $\mathcal{E}^3$ er klassen av Kalmárelementære funksjoner $\mathcal{E}'$.*

Til senere bruk tar jeg med noen funksjoner og deres respektive klassetilhørighet.

Teorem 2.27

i. Funksjonen $\dot{-}$ (modifisert subtraksjon) er med i $\mathcal{E}^0$. Det er også konstantfunksjonen, c_i^n, og funksjonen $x + k$ der k er et vilkårlig fast tall.

ii. Funksjonen max *er med i $\mathcal{E}^1$ [17, s. 23].*

iii. Funksjonen $\beth^k(x)$ er med i $\mathcal{E}^3$ der k er et vilkårlig fast tall.

iv. $\mathcal{E}^i$ er lukket under bundet sum for $i \geq 2$ [17, s. 23].

v. $\mathcal{E}^i$ er lukket under bundet produkt for $i \geq 3$ [17, s. 23].

Bevis. (Bevis av teorem 2.27 punkt i og iii) Punkt i: Konstantfunksjonen c_i^n er gitt ved $c_i^n = (x_1, \ldots, x_n) = i$ for alle $i, n \in N$. Funksjonen c_i^k kan defineres ved komposisjon over etterfølgerfunksjonen, nullfunksjonen og projeksjonsfunksjonen:

$$c_i^k = \underbrace{\mathcal{S} \ldots \mathcal{S}}_{\text{i ganger}} \mathcal{O} \mathcal{I}_1^k(x_1, \ldots, x_k)$$

$\mathcal{O}, \mathcal{S}$ og $\mathcal{I}_i^n$ er alle med i $\mathcal{E}^0$ og $\mathcal{E}^0$ er lukket under komposisjon, så $c_i^n \in \mathcal{E}^0$. Punkt iii: Funksjonen $\beth^k(x)$ for gitt k kan skrives ved komposisjon over eksponensialfunksjonen. □

Korrolar 2.28 (Korrolar til teorem 2.27 og definisjon 2.20) *La σ være en n-ær funksjon i $\mathcal{E}^r, r \geq 3$. Da er funksjonen g definert ved $g(x_1, \ldots, x_n) = \beth^k(\sigma(x_1, \ldots, x_n))$, for vilkårlig fast k, også i $\mathcal{E}^r$.*

Bevis. Funksjonen g er definert ved komposisjon over funksjonene σ og $\beth^k(x)$ som begge er i $\mathcal{E}^r$. □

2.2 A proof system for first order logic

According to Schwichtenberg and Troelstra [33] proof theory may be roughly divided into two parts: structural proof theory and interpretational proof theory. In structural proof theory one analyse the properties and structure of derivations in formal systems. The central methods are cut elimination and normalization. In interpretational proof theory, the tools are syntactical translations of one formal theory into another. I will only be concerned with the methods of structural proof theory.

To avoid confusion I adopt the convention of Schwichtenberg and Troelstra to use the word derivation about formal arguments, whereas the word proof is used only on the meta-level.

As the formal system for the derivations in this thesis, I use a variant of sequent calculus for classical logic (LK). Sequent calculus was introduced by the German logician Gerhard Gentzen in the nineteen thirties and has later been refined by other logicians.

The variant I use is a classical Gentzen system with shared context in the premise, called **G3c**+Cut in conformity with Schwichtenberg and Troelstra [33] and Negri and von Plato [22].

Later **G3c**+Cut is extended with non-logical rules for equality and primitive recursive functions. The idea of extending proof systems with nonlogical rules of inference was first introduced by Sara Negri [20] for the intuitionistic theories of apartness and order. It has been developed further by Negri and Jan von Plato [21, 22] and is also described in Schwichtenberg and Troelstra [33].

First I give some preliminary definitions for later use.

2.2.1 Notation and definitions

A logical system consists of a formal language with an alphabet and a system of axioms and rules for making logical inferences.

Alphabet:

- The logical connectives, $\wedge, \vee, \neg, \supset$
- Quantifiers, $\forall, \exists$
- A set of variables: $x_1, x_2, \ldots$
- Function symbols $f, g, \ldots$ for all primitive recursive functions. Every functions symbol has an arity $n \geq 1$, which decides the number of arguments that f can take. Constants are functions with no arguments.
- Relation symbols, $R, S, T, \ldots$. Unary relations are called predicates. In particular the language contains the binary relation symbol $=$, written infix.

Language: The language consists of terms and propositions defined as follows:

- Terms are defined inductively by the clauses:
  1. Variables and names are terms.
  2. If f is a function symbol with arity n and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.
- Formulas:
  1. $\perp$ is a formula.

2. An atomic proposition is an expression of type $R(t_1, \ldots, t_n)$ where R is a relation symbol with arity n and each t_i is a term.
3. If A and B are propositions, then $\neg A$, $A \vee B$, $A \wedge B$, $A \supset B$, $\forall x A$ and $\exists x A$ are propositions.

Definition 2.29 *The depth of a formula is defined inductively as the maximum length of a branch in its construction tree:*

$|P| = 0$, *for* P *atomic.*

$|A \circ B| = max(|A|, |B|) + 1$ *for binary operators* $\circ$, $|\circ A| = |A| + 1$ *for unary operators* $\circ$.

Definition 2.30 *The height of a derivation* $|\mathcal{D}|$ *is defined inductively by:*

1. $|\mathcal{D}| = 0$ *if* $\mathcal{D}$ *is an axiom;*
2. *Let* $\mathcal{D}_1, \ldots, \mathcal{D}_n$ *be derivations with height* $d_1, \ldots, d_n$, R *be a rule with* n *premises. Let* $\mathcal{D}$ *be the derivation:*

$$\frac{\mathcal{D}_1, \quad \ldots \quad \mathcal{D}_n}{\Gamma \Rightarrow \Delta} \; R$$

Then $|\mathcal{D}| = max(d_1, \ldots, d_n) + 1$.[2]

Notation 2.31 *For sequents derived in a proof system* S *the following notation is used*

$$\vdash^S \Gamma \Rightarrow \Delta$$

If I want to indicate that a deduction $\mathcal{D}$ *derives* $\Gamma \Rightarrow \Delta$ *I write* $\mathcal{D} \vdash^S \Gamma \Rightarrow \Delta$ *(or* $\mathcal{D} \vdash \Gamma \Rightarrow \Delta$ *if* S *is given from the context).*

That $\mathcal{D}$ *derives a sequent in a system* S *in at most* n *steps is denoted*

$$\mathcal{D} \vdash^S_n \Gamma \Rightarrow \Delta$$

If I want to state that a proposition A *is derivable in a system* S *I write*

$$\vdash^S \Rightarrow A \qquad \textit{or just} \; \vdash^S A$$

[2]Several results in the coming sections are proven by induction on the heights of derivations. For some results I refer to proofs in Negri and von Plato [22] and Schwichtenberg and Troelstra [33]. Their definitions of the height of derivations look slightly different from the one given here, but the measures of height are equivalent in all three definitions.

Definition 2.32 (Rule types) *There are two main types of rules in a proof system G. The* ***primitive*** *rules that are part of the system and the rules that are either* ***derivable*** *or* ***admissible****:*

A rule with premises $S_1, \ldots, S_n$ and conclusion S is said to be ***derivable*** *in G if for each instance of $S_1, \ldots, S_n$ there is a deduction of S from S_i by the rules of G.*

It is said to be ***admissible*** *if, whenever an instance of $S_1, \ldots, S_n$ is derivable in G, the corresponding instance of S is derivable in G.*

2.2.2 Sequent calculus for classical logic

Sequent calculus derives sequents of the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets. Γ is called the antecedent and Δ the succedent. Let $\Gamma = P_1, \ldots, P_n$ $\Delta = Q_1, \ldots, Q_m$ where $n, m \geq 0$. The intuitive interpretation of validity in sequent calculus is that $\Gamma \Rightarrow \Delta$ is valid iff $P_1 \wedge \cdots \wedge \ldots P_n \supset Q_1 \vee \cdots \vee Q_m$ is true. The logical rules of sequent calculus for classical logic give a formal theory of the derivability relation. It is now common to use the symbol $\Rightarrow$ inside the system, instead of $\vdash$ which is a meta-level expression. Deductions in sequent calculus are finite trees with a single root and axioms at the top nodes.

Sequent calculus for classical logic has a semantical motivation in which the system is obtained by analyzing truth conditions for formulas. Each rule breaks composed formulas into simpler ones. The problem of checking the truth value of a composed formula is reduced to checking the truth values of its subformulas. Moreover these transitions are done locally in a derivation. Thus one obtains a guided way to search for a derivation of a sequent.

The usual way of searching for a derivation in sequent calculus is root-first. One starts with the theorem one wants to prove and sees if it is possible to arrive at axioms in the top nodes by application of the rules.

The root-first proof search procedure runs contrary to the formal definition of a derivation which is given inductively: Instances of axioms are derivations, and if instances of premises of a rule are conclusions of derivations, an application of the rule gives a derivation.

Thus when proofs are given by induction on the height of the proof tree one often use the "top-down" approach to derivations. Whereas when I give a constructive proof that there exists derivations of certain expressions, the derivation is constructed root-first.

Sequent calculus for classical logic has properties that are useful in the analysis of derivations. The most important is the existence of cut-free derivations corresponding to the existence of normal form in natural deduction.

Gentzen's original formalization of sequent calculus contained the Cut rule:

$$\frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'} \; Cut$$

The general idea is that if you can show a formula A from the axioms of a system, then you can use A together with the axioms to show Δ. The use of Cuts in classical logic can be compared to the use of lemmas in mathematical proofs. One start with showing some intermediate results and use these to prove the main theorem.

The problem with the Cut rule is that it allows infinite branching of the search space. The cut formula can be any derivable formula.

To meet this problem Gentzen showed that it is always possible to eliminate cuts in a derivation in sequent calculus. He gave an effective procedure for eliminating cut that always terminates.

The existence of cut-free derivations has several applications. It is used to show the subformula property and consistency of the system LK. The subformula property says that in a cut-free derivation all the formulas are subformulas of the end-sequent.

2.2.3 The proof system G3c+Cut

I describe the calculus **G3c**+Cut and some of its properties. In the rules of **G3c**+Cut the sets Γ and Δ are finite multisets, that is, lists with multiplicity but no order. Γ and Δ can be empty. The formula with the connectives in a rule is called the **principal** formula and its components in the premises are the **active** formulas. Γ, Δ are called the context.

The name **G3c**+Cut indicates the basic properties of the proof system. The letter G indicates that it is a sequent system (Gentzen system) and the letter c says that it is a classical system.

The number 3 means that the contexts in both premises of two-premise rules are the same. This is called context-sharing. Rules that are not context-sharing are called context-independent. Example of context-independent versus context-sharing rules:

$$\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \wedge B} \; R\wedge \qquad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B} \; R\wedge$$

In the former rule the contexts in the premises are independent of each other, whereas in the latter the contexts in the premises are the same. Context-sharing has advantages for the proof search. In systems with shared contexts one can uniquely determine the premises, once it is decided which formula of the end-sequent is the principal one.

Extension of a system by a rule R is indicated through writing +Rule. Thus the +Cut says that **G3c** is extended with the Cut rule.

$$\frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma' \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'} \; Cut$$

Note that this variant of the cut rule has context-independent premises.

Definition 2.33 *The axioms and rules for* ***G3c****:*

Axiom:

$$P, \Gamma \Rightarrow \Delta, Pax \qquad (P \; atomic)$$

Logical rules:

$$\frac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta} \; L\wedge \qquad \frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B} \; R\wedge$$

$$\frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta} \; L\vee \qquad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B} \; R\vee$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \supset B, \Gamma \Rightarrow \Delta} \; L\supset \qquad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \supset B} \; R\supset$$

$$\frac{A[t/x], \forall x A, \Gamma \Rightarrow \Delta}{\forall x A, \Gamma \Rightarrow \Delta} \; L\forall \qquad \frac{\Gamma \Rightarrow \Delta, A[y/x]}{\Gamma \Rightarrow \Delta, \forall x A} \; R\forall$$

$$\frac{A[y/x], \Gamma \Rightarrow \Delta}{\exists x A, \Gamma \Rightarrow \Delta} \; L\exists \qquad \frac{\Gamma \Rightarrow \Delta, \exists x A, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x A} \; R\exists$$

$$\frac{}{\bot, \Gamma \Rightarrow \Delta} \; L\bot$$

In $R\forall$ and $L\exists$, y cannot occur free in $\Gamma, \Delta, \forall x A$ or $\exists x A$.

An important property of G3 systems is that the structural rules of weakening and contraction are absorbed into the the rules and axioms. They are admissible and need not be part of the system as such. To obtain admissibility of contraction the principal formulas of the conclusion in $L\forall$ and $R\exists$ must be repeated in the premise.

Lemma 2.34 (Height preserving inversion in G3c) *[22, p. 75]*

i. If $\vdash_n A \wedge B, \Gamma \Rightarrow \Delta$ then $\vdash_n A, B, \Gamma \Rightarrow \Delta$.

ii. If $\vdash_n \Gamma \Rightarrow \Delta, A \vee B$ then $\vdash_n \Gamma \Rightarrow \Delta, A, B$.

iii. If $\vdash_n A_0 \vee A_1, \Gamma \Rightarrow \Delta$ then $\vdash_n A_i \Gamma \Rightarrow \Delta (i \in 0, 1)$.

iv. If $\vdash_n \Gamma \Rightarrow \Delta, A_0 \wedge A_1$ then $\vdash_n \Gamma \Rightarrow \Delta, A_i (i \in 0, 1)$.

v. *If* $\vdash_n \Gamma \Rightarrow \Delta, A \supset B$ *then* $\vdash_n A, \Gamma \Rightarrow \Delta, B$.

vi. *If* $\vdash_n A \supset B, \Gamma \Rightarrow \Delta$ *then* $\vdash_n \Gamma \Rightarrow \Delta, A$ *and* $\vdash_n B, \Gamma \Rightarrow \Delta$.

vii. *If* $\vdash_n \Gamma \Rightarrow \Delta, \forall x A$ *then* $\vdash_n \Gamma \Rightarrow \Delta, A[x/y]$, *for any* y *such that* $y \notin FV(\Gamma, \Delta, A)$.

viii. *If* $\vdash_n \exists x A, \Gamma \Rightarrow \Delta$ *then* $\vdash_n A[x/y], \Gamma \Rightarrow \Delta$, *for any* y *such that* $y \notin FV(\Gamma, \Delta, A)$.

Theorem 2.35 (Height-preserving weakening for G3c) *[22, p. 75]*

i. *If* $\vdash_n \Gamma \Rightarrow \Delta$, *then* $\vdash_n D, \Gamma \Rightarrow \Delta$.

ii. *If* $\vdash_n \Gamma \Rightarrow \Delta$, *then* $\vdash_n \Gamma \Rightarrow \Delta, D$.

Theorem 2.36 (Height-preserving contraction for G3c) *[22, p. 75]*

i. *If* $\vdash_n D, D, \Gamma \Rightarrow \Delta$, *then* $\vdash_n D, \Gamma \Rightarrow \Delta$.

ii. *If* $\vdash_n \Gamma \Rightarrow \Delta, D, D$, *then* $\vdash_n \Gamma \Rightarrow \Delta, D$.

Theorem 2.37 *[22, p. 81-86] The sequent* $\Gamma \Rightarrow \Delta$ *is derivable in G3c iff it is valid.*

2.2.4 Cut elimination and upper bounds

Definition 2.38 *The* level *of a cut is defined as the sum of the depths of the deductions of the premises. The* rank *of a cut on* A *is* $|A| + 1$. *The* cutrank *of a deduction* D, *is the maximum of the ranks of the cutformulas occurring in* D.

Notation 2.39 *A derivation with conclusion* $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$ *obtained by applying height-preserving weakening to* $\mathcal{D}$ *with conclusion* $\Gamma \Rightarrow \Delta$ *is denoted* $\mathcal{D}[\Gamma' \Rightarrow \Delta']$. *In this case* $\mathcal{D}$ *is said to be* weakened with $\Gamma' \Rightarrow \Delta'$.

Notation 2.40 *Derivations of the premises in an* n*-premise rule are denoted* $\mathcal{D}_0, \ldots, \mathcal{D}_n$. *The depth of a derivation* $\mathcal{D}$ *is* d, *the depth of* $\mathcal{D}_0, \ldots, \mathcal{D}_n$ *are denoted* $d_0, \ldots, d_n$.

Theorem 2.41 *[33, p. 94] Cut elimination holds for* ***G3c****+Cut.*

I outline the course of the proof given in Schwichtenberg and Troelstra [33][3]. Proofs that the cut elimination theorem holds for extensions of **G3c** are extensions of the proof outlined here. The strategy used by Schwichtenberg and Troelstra [33] is to remove cuts that are topmost among all cuts with rank equal to the rank of the whole deduction. It provides a constructive method for replacing a sub-deduction $\mathcal{D}$ of the form

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ \Gamma \Rightarrow \Delta, A & A, \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta}\ Cut_{CS}$$

where $cr(\mathcal{D}_i) \leq |A| = cr(\mathcal{D}) - 1$ for $i \in 0, 1$, by a deduction $\mathcal{D}^*$ with the same conclusion, such that $cr(\mathcal{D}^*) \leq |A|$. This is proved by a main induction on the cutrank, with a sub-induction on the level of the cut at the bottom of $\mathcal{D}$.

Three cases are considered:

1. at least one of $\mathcal{D}_0$, $\mathcal{D}_1$ is an axiom;
2. $\mathcal{D}_0$ and $\mathcal{D}_1$ are not axioms, and in at least one of the premises the cutformula is not principal;
3. the cutformula is principal on both sides.

I do not go through the various instances of the three cases, but refer to the proof in Troelstra and Schwichtenberg [33].

Theorem 2.42 (Hyper exponential bounds on cut elimination) *[33, p. 149] For every deduction $\mathcal{D}$ in **G3c**+Cut of cutrank k there is a cut-free deduction $\mathcal{D}^*$ with the same conclusion, achieved by eliminating cuts from $\mathcal{D}$, such that*

$$|D^*| \leq \beth^k(|D|)$$

2.3 First order structures and theories

In the previous section I gave rules for a calculus of pure logic. It contains only logical axioms and rules. The propositions derivable in **G3c**+Cut are logical tautologies as for example $A \wedge B \supset A$. In order to be able to derive propositions about functions one must add non-logical axioms or rules. Axioms are syntactical descriptions of functions and relations, but they can also be regarded as interpretations of elements in the *structure* of the language.

[3]Schwichtenberg and Troelstra show eliminability of context-sharing Cut, Cut_{CS}, and then show the eliminability of Cut as a consequence.

Definition 2.43 (First order structure) *Let L be a first order language as defined in the previous section.*

- *A structure, $\mathcal{A}$ for L contains a non-empty set A and interpretations of the names, relations and function symbols in L over A.*
- *A closed proposition C is said to be valid in $\mathcal{A}$, written $\mathcal{A} \models C$, if it is true in $\mathcal{A}$.*

For example the axiom $\forall xy(x + y) = (y + x)$ give an interpretation of the function plus, denoted $'+'$ as having the property of being commutative. Together a proof system and a set of axioms constitutes a theory:

Definition 2.44 (First order theory) *Let L be a first order language as defined in the previous section.*

1. *A first order theory T consists of*
   - *The language L.*
   - *A set of propositions from L called the nonlogical axioms (rules) of T.*
   - *The proof system for first order logic as described in the previous section.*
2. *If T is a first order theory with language L and $\mathcal{A}$ is an L-structure then $\mathcal{A}$ is a model for T if all nonlogical axioms in T are valid in A.*
3. *A proposition C is said to be a logical consequence of T, written $T \models C$, if C is valid in all models for T.*

In 1889 the Italian logician Guiseppe Peano gave five axioms to characterize the natural numbers[4].

$$N(0) \tag{2.1}$$

$$\forall x(N(x) \supset N(Sx)) \tag{2.2}$$

$$\forall xy(x = y \supset Sx = Sy) \tag{2.3}$$

$$\forall x(Sx \neq 0) \tag{2.4}$$

$$(0 : K \wedge \forall x(K(x) \supset K(Sx))) \supset \forall x(K(x)) \tag{2.5}$$

[4]In Peano's original axiomatization he used 1 as the first number instead of 0. Axioms 2.1 to 2.5 are written in conformity with modern tradition where the first number is 0.

The two first axioms of Peano arithmetic generate an inductive set that consists of the natural numbers. The remaining axioms ensures that the terms satisfying the N predicate coincide with the natural numbers. In the next chapter I extend **G3c**+Cut to a theory of primitive arithmetic, called PRA, without the induction axiom. This is a much weaker theory than Peano arithmetic.

Gödel's completeness theorem for first order theories exists in several formulations. The following is given by Fenstad and Normann [6]:

Theorem 2.45 (Completeness of first order theories) *[6, s. 148] Let T be a first order theory and C a proposition in the language of T. Then the following equivalence holds:*

$$T \vdash C \Leftrightarrow T \models C$$

To readers familiar with Gödel's incompleteness theorem this may be a bit confusing. The incompleteness theorem states that if T is an axiomatisable theory with N as one of its models there is a closed proposition C that is true in N but not derivable in T. Remember the meaning of $T \models C$ for theories. It denotes that C is a logical consequence of T. Furthermore a proposition is a logical consequence of a theory if it is valid in *all* it's models. Hence if a proposition C is true in N but false in a non-standard model of T it is not a logical consequence of T and therefore not derivable in T.

The notion of a non-standard model was first introduced by Skolem end Löwenheim. In the early nineteen thirties they found that all theories that have an infinitely enumerable model isomorphic to N has models with cardinalities larger than N. In other words all theories that has N as their model has a non-standard model. This is called the upward Skolem-Löwenheim Theorem [5].

Chapter 3

A theory of primitive recursive arithmetic

In this chapter **G3c**+Cut is extended to a theory for primitive recursive functions. Normally a theory is defined as a logical system that contains an explicitly stated set of axioms from which theorems can be derived. In sequent calculus one way to obtain a theory for equality and functions is to add axioms in the form of sequents of type $\Rightarrow A$. Nonlogical theorems of the theory can be derived by introducing cuts on the axioms. The problem with this method is that it leads to failure of cut elimination. Negri and von Plato [21] presents a simple example given by Girard: The axioms have the forms $\Rightarrow A \supset B$ and $\Rightarrow A$. The sequent B is derived from these axioms by

$$\dfrac{\Rightarrow A \qquad \dfrac{\Rightarrow A \supset B \qquad \dfrac{A \Rightarrow A \quad B \Rightarrow B}{A, A \supset B \Rightarrow B}\, L\supset}{A \Rightarrow B}\, Cut}{\Rightarrow B}\, Cut$$

Inspection of the rules for sequent calculus described in section 2.2.3 show that there can be no cut-free derivation of $\Rightarrow B$, which leads Girard to conclude that Gentzen's cut elimination theorem fail for proper axioms. Negri states more generally that "the cut elimination theorem does not apply to sequent calculus derivations having premises that are not logical axioms."

This problem with axiomatic systems makes them unsuitable for the structural analysis of derivations in primitive recursive arithmetic, which is the issue of this thesis. Negri and von Plato [22] have developed a method for transforming nonlogical axioms into inference rules. They show that all classical theories axiomatized by purely universal axioms can be transformed into rule systems that allow full cut elimination. A statement is **universal** if it is in prenex normal form with only universal quantifiers in the prefix.

3.1 Extensions of G3c+Cut with nonlogical rules

My application of Negri and von Plato's rule scheme differ from the examples they give in that I keep the cut rule as part of the system. This is because I give upper bounds on derivations D of certain arithmetic expressions. To obtain these bounds it is necessary to use cuts.

Negri and von Plato show that all structural rules, including the cut rule are admissible in extensions of **G3c** that follow their rule scheme. Thus they obtain cut-free systems for several axiomatic theories.

The method of transforming axioms into rules can also be used to obtain a system with cut, and then show that the cut rule can be eliminated from derivations. To show that a rule R is admissible in a system S is equivalent to showing that S is closed under R. Thus a proof that a rule R is admissible in S is also a proof that if a sequent is derivable in S, the same sequent is derivable in S without the use of R. In fact Schwichtenberg and Troelstra [33] give a proof that one has full cut elimination in **G3c**+Cut extended with rules of the type Negri and von Plato describe.

The basis proof systems of this section and proofs of their various properties are found in Schwichtenberg and Troelstra [33] and Negri and von Plato [22]. Where new axioms or rules are added I verify that they satisfy the rule scheme of Negri and von Plato.

3.1.1 Transforming a classical theory into a rule system

Nonlogical rules added to a system in the place of axioms must satisfy the following principle.

Principle 3.1 *In nonlogical rules, the premises and conclusion are sequents that have atoms as active and principal formulas in the antecedent and an arbitrary context in the succedent.*

It follows from the principle that the inversion lemma 2.34 holds for extensions of **G3c** with nonlogical rules. Following this principle one can convert theories axiomatized by universal axioms into rule systems: First eliminate all the quantifiers. Then one can use the existence of conjunctive normal form in classical propositional logic. This states that each formula is equivalent to a conjunction of disjunctions of atoms and negations of atoms. Each conjunct can be converted into the classical equivalent form $P_1 \wedge \cdots \wedge P_m \supset Q_1 \vee \cdots \vee Q_n$ which is representable as a rule of inference:

$$\frac{Q_1, \Gamma \Rightarrow \Delta \quad \ldots \quad Q_n, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_m, \Gamma \Rightarrow \Delta} \; \mathit{Reg}$$

Γ, Δ are arbitrary multi-sets. Due to the arbitrary context in the succedent right contraction is still admissible for extensions of **G3c**. To obtain admissibility of left contraction it is necessary to repeat the principal formulas of the conclusion in the premises:

$$\frac{Q_1, P_1, \ldots, P_m, \Gamma \Rightarrow \Delta \quad \ldots \quad Q_n, P_1, \ldots, P_m, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_m, \Gamma \Rightarrow \Delta} \; Reg$$

Here $P_1, \ldots, P_m$ in the conclusion are principal in the rule, and $P_1, \ldots, P_m$ and $Q_1, \ldots, Q_n$ are active in the premises. To take care of the case in which a substitution produces two identical formulas that are both principal in a nonlogical rule, Negri and von Plato have included the condition:

Closure Condition 3.2 *Given a system with nonlogical rules, if it has a rule where a substitution instance in the atoms produces a rule of the form*

$$\frac{Q_1, P_1, \ldots, P_{m-2}, P, P, \Gamma \Rightarrow \Delta \quad \ldots \quad Q_n, P_1, \ldots, P_{m-2}, P, P, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_{m-2}, P, P, \Gamma \Rightarrow \Delta} \; Reg$$

then it also has to contain the rule

$$\frac{Q_1, P_1, \ldots, P_{m-2}, P, \Gamma \Rightarrow \Delta \quad \ldots \quad Q_n, P_1, \ldots, P_{m-2}, P, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_{m-2}, P, \Gamma \Rightarrow \Delta} \; Reg$$

Here are some examples of axioms that are covered by the rule scheme and their transformation into rules. Axioms of the type $P \wedge Q, P \vee Q$ and $P \supset Q$ become:

$$\frac{P \Rightarrow \Delta,}{\Rightarrow \Delta} \qquad \frac{Q \Rightarrow \Delta}{\Rightarrow \Delta} \qquad \frac{P \Rightarrow \Delta \quad Q \Rightarrow \Delta}{\Rightarrow \Delta} \qquad \frac{Q, P \Rightarrow \Delta}{P \Rightarrow \Delta}$$

In conformity with Negri and von Plato [22] I denote an extension of **G3c** with rules that satisfy the given conditions as **G3c***

Theorem 3.3 *The rules of weakening and contraction are admissible in* ***G3c*****.*

The proof is an extension of the results for the purely logical calculus, theorems 2.35 and 2.36. Negri and von Plato [22, p. 131] give proofs of height-preserving admissibility of weakening and contraction for **G3im***[1]. The new cases that arise from the nonlogical rules are analogous for the classical and intuitionistic extensions.

[1] An extension of an intuitionistic multi-succedent Gentzen-system.

3.1.2 Cut elimination in G3c*+Cut and its applications

As mentioned extensions of **G3c** with nonlogical rules, following the scheme of Negri and von Plato, allows for full cut elimination. This has some useful applications for **G3c***+Cut. It allows one to show a restated version of the sub-formula property of cut-free derivations and it preserves the upper bound on cut-elimination that follows from the classical cut elimination theorem. Both these properties of **G3c***+Cut are exploited in later chapters.

Theorem 3.4 (Cut elimination) *The Cut rule*

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ \Gamma \Rightarrow \Delta, D & D, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\ Cut$$

*can be eliminated in **G3c***+Cut.*

The proof for **G3c**+Cut extends to **G3c***+Cut. For the new rules it is sufficient to show that they all commute with the cut rule. This is shown both in Negri and von Plato [22, p. 133] and in Schwichtenberg and Troelstra [33, p. 132-133] for an extended version of the Negri and von Plato-system.

Since all the nonlogical rules permute with the cut rule, they do not interfere with the upper bound on cut elimination given in theorem 2.42.

Corollary 3.5 *For every deduction D in **G3c***+Cut of cut-rank k there is a cut-free deduction D^* with the same conclusion, achieved by eliminating cuts from $\mathcal{D}$, such that*

$$|D^*| \leq 2_k^{|D|}.$$

From the cut elimination theorem for **G3c***+Cut and the form of nonlogical rules the usual sub-formula property of cut free derivations can now be restated.

Theorem 3.6 *If $\Gamma \Rightarrow \Delta$ is a cut-free derivation in **G3c***+Cut then all formulas in the derivation are either sub-formulas of the end-sequent or atomic formulas.*

Proof. Only the cut rule and nonlogical rules can make formulas disappear in a derivation $\mathcal{D}$. Since all principal formulas of nonlogical rules are atomic, the weaker sub-formula property follows for cut-free derivations. □

By the sub-formula property, only nonlogical rules are used in cut-free derivations of an atomic expression, A. Thus, the task of proof search is divided into a logical and a mathematical part.

3.1.3 The cost of eliminating nonlogical rules

As mentioned one can distinguish between two main types of rules in a proof system: The primitive rules that are part of the system, and the ones that are either derivable or admissible. In a system with nonlogical rules, none of the primitive nonlogical rules can be eliminated without loosing expressive power. However one can show that certain nonlogical rules are admissible. If an admissible rule is used in a derivation, this rule can later be eliminated. It is of interest to know the cost of eliminating such a rule.

Following the notation introduced in this chapter a rule that introduces an atomic formula P is called Reg_P.

For use in later proofs I show the following:

Proposition 3.7 *If P is an atomic formula and there is a cut-free derivation $\vdash_m \Rightarrow P$ in **G3c***+Cut, then the rule*

$$\frac{P, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\, Reg_P$$

*is admissible in **G3c***+Cut.*

Proof. A rule can be eliminated by replacing it with a cut on the introduced proposition and then eliminating the cut. □

The upper bound on cut elimination from corollary 3.5 is $2_1^{|\mathcal{D}|} = 2^{|\mathcal{D}|}$ when $\mathcal{D}$ contains only atomic cuts. The derivation grows exponentially in the height of the original derivation. However when there are only atomic cuts in the derivation, one can get a better measure of the cost.

Proposition 3.8 (Bounds on elimination of atomic cuts) *Let $\mathcal{D}$ be a deduction in* PRA *ending in a cut on an atomic, mathematical expression P:*

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1 \\ \Gamma' \Rightarrow P, \Delta' & P, \Gamma \Rightarrow \Delta\end{array}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\, Cut$$

It is essential that Γ' and Δ' are assumed empty. For the rest of the section I therefore let out Γ' and Δ' in the examples.

Let $\mathcal{D}_0$ and $\mathcal{D}_1$ be cut-free derivations of height d_0 and d_1 respectively. Then $\mathcal{D}$ can be transformed to a cut-free deduction $\mathcal{D}^*$ with height $d^* = d_0 + d_1$.

Proof. The proof is by induction on d_0 and makes use of closure under height preserving contraction and weakening.

Since Γ' is empty, the base case becomes $d_0 = 1$. Then the formula P must have been introduced in the left premise by a nonlogical rule and the cut is unnecessary:

$$\dfrac{\dfrac{\dfrac{}{P \Rightarrow P}\, ax}{\Rightarrow P}\, R \qquad \begin{matrix}\mathcal{D}_1 \\ P, \Gamma \Rightarrow \Delta\end{matrix}}{\Gamma \Rightarrow \Delta}\, Cut$$

is transformed to

$$\dfrac{\begin{matrix}\mathcal{D}_1 \\ P, \Gamma \Rightarrow \Delta\end{matrix}}{\Gamma \Rightarrow \Delta}\, R$$

By definition $d^* = d_1 + 1 = d_1 + d_0$

Induction hypothesis: The proposition is true when $d_0 < k$.

Induction step: It is necessary to show that the proposition holds when $d_0 = k$.

To show this I look at the last rule in $\mathcal{D}_0$. From the assumption that Γ', Δ' are empty and the sub-formula property of cut-free derivations in PRA, it follows that $\mathcal{D}_0$ contains nonlogical rules only. Since all nonlogical rules have arbitrary formulas in the succedent, P can not be principal in the rule.

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_{01} \\ Q_1 \Rightarrow P\end{matrix} \quad \ldots \quad \begin{matrix}\mathcal{D}_{0m} \\ Q_n \Rightarrow P\end{matrix}}{\Rightarrow P}\, R \qquad \begin{matrix}\mathcal{D}_1 \\ P, \Gamma \Rightarrow \Delta\end{matrix}}{\Gamma \Rightarrow \Delta}\, Cut$$

where $d_0 = max(d_{01}, \ldots, d_{0m}) + 1$ is transformed to

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_{01} \\ Q_1 \Rightarrow P\end{matrix} \quad \begin{matrix}\mathcal{D}_1 \\ P, \Gamma \Rightarrow \Delta\end{matrix}}{Q_1, \Gamma \Rightarrow \Delta}\, Cut \quad \ldots \quad \dfrac{\begin{matrix}\mathcal{D}_{0m} \\ Q_m \Rightarrow P\end{matrix} \quad \begin{matrix}\mathcal{D}_1 \\ P, \Gamma \Rightarrow \Delta\end{matrix}}{Q_m, \Gamma \Rightarrow \Delta}\, Cut}{\Gamma \Rightarrow \Delta}\, R$$

Now the induction hypothesis can be used on $d_{01}, \ldots, d_{0m}$ to obtain cut free sub-deductions $\mathcal{D}^*_{01}, \ldots, \mathcal{D}^*_{0m}$ with $d^*_{0i} = d_{0i} + d_1, 1 \leq i \leq m$:

$$\dfrac{\begin{matrix}\mathcal{D}^*_{01} \\ Q_1, \Gamma \Rightarrow \Delta\end{matrix} \quad \ldots \quad \begin{matrix}\mathcal{D}^*_{0m} \\ Q_m, \Gamma \Rightarrow \Delta\end{matrix}}{\Gamma \Rightarrow \Delta}\, R$$

This gives the above derivation $\mathcal{D}^*$ with height decided by the equation:

$$\begin{aligned}
\text{(by definition) } d^* &= max(d^*_{01}, \ldots, d^*_{0m}) + 1 \\
\text{(by induction hypothesis) } &= max(d_{01} + d_1, \ldots, d_{0m} + d_1) + 1 \\
&= max(d_{0i}) + 1 + d_1 \\
&= d_0 + d_1
\end{aligned}$$

Proposition 3.9 *If there is a cut free derivation $\vdash_n \Gamma \Rightarrow \Delta$ with one instance of the admissible rule Reg_P, then there is a derivation of the same sequent with no occurrences of Reg_P with height $d = m + n$ where m is the height of the derivation of P.*

Proof. Let $\mathcal{D}$ be the cut-free derivation

$$\begin{array}{ll} \mathcal{D}_1 \left\{ \begin{array}{l} \dfrac{}{P, \Gamma' \Rightarrow \Delta'}\, ax \\ \dfrac{}{\Gamma, \Gamma' \Rightarrow \Delta'}\, Reg_P \end{array} \right. \\ \mathcal{D}_0 \left\{ \begin{array}{l} \vdots \\ \Gamma \Rightarrow \Delta \end{array} \right. \end{array}$$

Replace the sub-deduction $\mathcal{D}_1$ with a deduction $\mathcal{D}'_1$ where Reg_P is replaced with a cut on P. Now $\mathcal{D}'_1$ is the deduction

$$\dfrac{\begin{array}{cc} \mathcal{D}'_{10} & \mathcal{D}'_{11} \\ \Gamma \Rightarrow P & P, \Gamma' \Rightarrow \Delta' \end{array}}{\Gamma, \Gamma' \Rightarrow \Delta'}\, Cut$$

By proposition 3.8 $\mathcal{D}'_1$ can be transformed into a cut-free deduction $\mathcal{D}'^*_1$ with the same conclusion such that $d'^*_1 = d'_{10} + d'_{11} = d_1 + m$. This gives a cut-free derivation $\mathcal{D}^*$ of $\Gamma \Rightarrow \Delta$ such that

$$\begin{aligned} d^* &= d_0 + d'^*_1 \\ &= d_0 + d_1 + m \qquad \text{by proposition 3.8} \\ &= d + m \end{aligned}$$

Proposition 3.10 *If there is cut free derivation $\vdash_n \Gamma \Rightarrow \Delta$ with admissible rules $Reg_{P1}, \ldots, Reg_{Pn}$, where the height of derivations of each P_i is $m_i, 1 \leq i \leq n$, then there is a derivation of the same sequent with no admissible rules with height $d = n + m_1 + \cdots + m_n$.*

Proof. First replace all instances of Reg_{Pi} with cuts. Eliminate the uppermost cut. The new derivation is now $n + m_1$ and there are $n - 1$ cuts left. For each cut eliminated m_i is added to the height of the tree. This gives the required height:

$$n + m_1 + \cdots + m_n$$

3.2 Application to number theory

In this section I give an application of the rule scheme described above for a formal theory of numbers. The theory used in this thesis is axiomatized by universal axioms only. First I define a basis theory that later is extended with the axioms necessary to prove that all the strictly growing elementary functions are inductive. It includes the two first axioms of Peano arithmetic, defining axioms for equality, transitivity, equality between functions and defining axioms for all primitive recursive functions. The axioms constitute a theory for primitive recursive arithmetic.

The choice of having a strictly universal theory puts limitations on what kind of system one can use. Full Peano arithmetic 2.3 is excluded because the induction axiom 2.5 cannot be transformed to universal form. This choice is motivated first by the wish to investigate which functions can be proven to be inductive from *as few* axioms or rules as possible. Second a universal axiomatized theory can be transformed into a system of nonlogical rules that satisfies the rule scheme of Negri and von Plato.

Without the induction axiom there are very few general properties of function terms that are derivable. For example it is not possible to derive within the system that $\forall xy(x + y = y + x)$, see section 5.3.

I now show that the axioms of PRA can be transformed into nonlogical rules that satisfy the rule scheme of section 3.1 and the closure condition.
Axioms:

$$
\begin{aligned}
&N(0)\\
&N(a) \supset N(Sa)\\
&a = a\\
&a = b \wedge a = c \supset b = c\\
&\vec{a} = \vec{b} \supset f(\vec{a}) = f(\vec{b})
\end{aligned}
$$

I also have defining axioms for all primitive recursive functions. For example for addition I have axioms

$$
\begin{aligned}
+a0 &= a\\
+aSb &= S(+ab)
\end{aligned}
$$

Negri and von Plato give an extension of **G3c** to a theory of equality with the following rules:

$$\frac{a = a, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ Ref \qquad \frac{a = b, P(a), P(b), \Gamma \Rightarrow \Delta}{a = b, P(a), \Gamma \Rightarrow \Delta}\ Repl$$

For transitivity they give the rule:

$$\frac{b = c, a = b, a = c, \Gamma \Rightarrow \Delta}{a = b, a = c, \Gamma \Rightarrow \Delta}\ Trans$$

They show that the closure condition and principle 3.1 are satisfied by the rules for equality and transitivity [22].

The axiom of equality between functions becomes:

$$\frac{f(a_1, \ldots, a_n) = f(b_1, \ldots, b_n), a_1 = b_1, \ldots, a_n = b_n, \Gamma \Rightarrow \Delta}{a_1 = b_1, \ldots, a_n = b_n, \Gamma \Rightarrow \Delta}\ Eqf$$

where $n \geq 1$.

Without loss of generality (and to avoid clutter) I assume $n = 2$ in the following definitions. In the case where one of the a_i and $b_i s$ are duplicated in $Eqf1$, I get multiple instances of $a_i = b_i$:

$$\frac{f(a, a) = f(b, b), a = b, a = b, \Gamma \Rightarrow \Delta}{a = b, a = b, \Gamma \Rightarrow \Delta}\ Eqf$$

In order to satisfy the closure condition I therefore need the contracted rule

$$\frac{f(a, a) = f(b, b), a = b, \Gamma \Rightarrow \Delta}{a = b, \Gamma \Rightarrow \Delta}\ Eqf_c$$

For functions defined by primitive recursion I get the following inference rules:

$$\frac{f(a, \bar{0}) = g(a), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ f0 \qquad \frac{f(a, Sb) = h(a, b, f(a, b)), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ frec$$

and for functions defined by composition:

$$\frac{f(a, b) = h(g(a, b), j(a, b)), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ fcomp$$

The rule of symmetry is derivable in PRA:

$$\dfrac{\dfrac{\dfrac{b = a, \Gamma \Rightarrow \Delta}{b = a, a = a, a = b, \Gamma \Rightarrow \Delta}\ W, W}{a = a, a = b, \Gamma \Rightarrow \Delta}\ Trans}{a = b, \Gamma \Rightarrow \Delta}\ Ref$$

These rule schemes cannot lead to duplications of principal formulas in the conclusion. Thus they satisfy the closure condition.

The addition of the above rules to **G3c**+Cut gives the system **G3c**+PRA $+Cut$. For simplicity I refer to this system simply as PRA in the preceding section.

Since all of the added rules follow the rule scheme and satisfy the closure condition I may conclude that the structural rules of weakening and contraction are admissible in PRA. I may also conclude that cuts may be eliminated from derivations in PRA.

PRA is a consistent theory since all of its axioms are true in the standard interpretation N. It follows from the upward Skolem-Löwenheim Theorem[5] that PRA has non-standard models, since it has N as a model.

3.3 Provably total functions in PRA

A function denoted by f is total over the natural numbers N if $\forall x(N(x) \supset \exists y(N(y) \wedge f(x) = y))$. This notion is captured by the expression $\forall x(N(x) \supset N(f(x)))$. PRA is not strong enough to derive such statements for primitive recursive functions f. Thus totality of a function f is not derived within the system itself, but I show by induction on the height of derivations that all primitive recursive functions f are total with regard to PRA. That is all primitive recursive functions f and for all $m_1, \ldots, m_n \in N$, the proposition $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ is derivable in PRA.

Notation 3.11 *Let m be a number. Then $\bar{m}$ is a term corresponding to m.*

3.3.1 Elementary bounds on derivations

In order to obtain the main result of this chapter I first need to show that for all $m_1, \ldots, m_k \in N$ there exists an $n \in N$ such that $f(m_1, \ldots, m_k) = n$ and there exists a corresponding derivation $\mathcal{D}$ in PRA of $f(\bar{m}_1, \ldots, \bar{m}_k) = \bar{n}$. In the course of the proof I verify that $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}^*$ with only nonlogical rules of the system. Furthermore there exists a function σ, such that $\sigma(m_1, \ldots, m_n)$ gives an upper bound on the height of $\mathcal{D}^*$ and σ is in $\mathcal{E}^r$.

Proposition 3.12 *Let f be an n-ary primitive recursive function in Gregorczyk class $\mathcal{E}^r$, $r \geq 2$. Then for all $m_1, \ldots, m_n \in N$ there exists a $k \in N$, and there exists a derivation $\mathcal{D} \in$ PRA, that contains only nonlogical rules, and a primitive recursive function $\sigma \in \mathcal{E}^3$ such that:*

$$\mathcal{D} \vdash^{\mathrm{PRA}}_{\sigma(m_1,\ldots,m_n)} f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$$

Proof. The proposition is shown by induction on the construction of f.

Basis step: I first show that the proposition holds for the initial functions. All the initial functions are in $\mathcal{E}^0$, thus they are also in $\mathcal{E}^2$.

1. The null-function can be defined for any number n of arguments by composition over the constant-function:

$$\mathcal{O}_n(x_1,\ldots,x_n) = c_0^n(x_1,\ldots,x_n) = 0$$

One instance of the inference rule for $\mathcal{O}_n$ gives the derivation:

$$\frac{\mathcal{O}_n(\bar{m}_1,\ldots,\bar{m}_n) = \bar{0} \Rightarrow \mathcal{O}_n(\bar{m}_1,\ldots,\bar{m}_n) = \bar{0}}{\Rightarrow \mathcal{O}_n(\bar{m}_1,\ldots,\bar{m}_n) = \bar{0}}\ \mathcal{O}$$

2. $f = \mathcal{S}(x)$. $\mathcal{S}(m) = m+1$ which is a number in N. For all numerals in N one can show

$$\frac{\mathcal{S}(\bar{m}) = \mathcal{S}(\bar{m}) \Rightarrow \mathcal{S}(\bar{m}) = \mathcal{S}(\bar{m})}{\Rightarrow \mathcal{S}(\bar{m}) = \mathcal{S}(\bar{m})}\ Ref$$

3. One instance of the inference rule for $f = \mathcal{I}_i^n$ gives:

$$\frac{\mathcal{I}_i^n(\bar{m}_1,\ldots,\bar{m}_n) = \bar{m}_i \Rightarrow \mathcal{I}_i^n(\bar{m}_1,\ldots,\bar{m}_n) = \bar{m}_i}{\Rightarrow \mathcal{I}_i^n(\bar{m}_1,\ldots,\bar{m}_n) = \bar{m}_i}$$

For the basis steps let $\sigma(\bar{m}) = c_1^n(\bar{m}) = 1, n \geq 0$. The σ is in $\mathcal{E}^2$.

Main induction hypothesis: Assume the proposition is true for functions $g_1,\ldots,g_m$ and h.

Induction step: It must be shown that the proposition is true also for a function f, when f is defined by $g_1,\ldots,g_m,h$ and the schemes for primitive recursion or composition.

Composition: Assume that $g_1,\ldots,g_m$ and $h \in \mathcal{E}^r$. Function f is defined by

$$f(l_1,\ldots,l_n) = h(g_1(l_1,\ldots,l_n),\ldots,g_m(l_1,\ldots,l_n))$$

Without loss of generality it can be assumed that $m = n = 2$. The function f must be in $\mathcal{E}^r$, since $\mathcal{E}^r$ is closed under composition. It must be shown that for all $l,m \in N$ there is a $q \in N$ such that $\vdash^{\mathrm{PRA}}_{\sigma(l,m)} \Rightarrow f(\bar{l},\bar{m}) = \bar{q}$ and $\sigma \in \mathcal{E}^r$.

Let $l,m \in N$. By IH 1 it follows that for all $l,m \in N$ there is a $k_i \in N$ and a derivation $\mathcal{D}_i, i \in 0,1$, such that $\mathcal{D}_i \vdash^{\mathrm{PRA}}_{\delta_i(l,m)} \Rightarrow g_i(\bar{l},\bar{m}) = \bar{k}_i$ and $\delta_i \in \mathcal{E}^r$.

By proposition 3.7 the rule

$$\frac{g_i(\bar{l},\bar{m}) = \bar{k}_i, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ Reg_{g_i}$$

is admissible in PRA.

By admissibility of the rules Reg_{g_1} and Reg_{g_2} and some instances of the equality rules $g_1(\bar{l},\bar{m})$ and $g_2(\bar{l},\bar{m})$ can be replaced by terms in h:

$$\dfrac{\dfrac{\dfrac{h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=h(\bar{k_1},\bar{k_2}),g_2(\bar{l},\bar{m})=\bar{k_2},g_1(\bar{l},\bar{m})=\bar{k_1}\Rightarrow\Delta}{g_2(\bar{l},\bar{m})=\bar{k_2},g_1(\bar{l},\bar{m})=\bar{k_1}\Rightarrow\Delta}\,Eqf}{g_1(\bar{l},\bar{m})=\bar{k_1}\Rightarrow\Delta}\,Reg_{g_2}}{\Rightarrow\Delta}\,Reg_{g_1}$$

By proposition 3.9 the admissible rules for Reg_{g_i} can be eliminated with cost $\delta_i(l,m)$.

By IH 1 there is a number $q\in N$ and a derivation $\mathcal{D}_2$ such that $\mathcal{D}_2\vdash^{\mathrm{PRA}}_{\gamma(k_1,k_2)}\Rightarrow h(\bar{k_1},\bar{k_2})=\bar{q},\gamma\in\mathcal{E}^r$. Again admissibility of the derivable rule for h can be used to obtain $h(\bar{k_1},\bar{k_2})=\bar{q}$ in the antecedent. Then by the derivable rule of symmetry and one instantiation of transitivity the formula $h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=\bar{q}$ is obtained in the antecedent. The inference rule for f and two new instances of symmetry and transitivity yields $f(\bar{l},\bar{m})=\bar{q}$ in the antecedent. Thus a derivation of $f(\bar{l},\bar{m})=\bar{q}$ can be obtained:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f(\bar{l},\bar{m})=\bar{q},h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=f(\bar{l},\bar{m}),h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=\bar{q}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}{h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=f(\bar{l},\bar{m}),h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=\bar{q}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Trans}{f(\bar{l},\bar{m})=h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m})),h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=\bar{q}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Sym+LW}{h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=\bar{q},h(\bar{k_1},\bar{k_2})=h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m})),h(\bar{k_1},\bar{k_2})=\bar{q}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,fcomp+LW}{h(\bar{k_1},\bar{k_2})=h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m})),h(\bar{k_1},\bar{k_2})=\bar{q}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Trans}{h(\bar{k_1},\bar{k_2})=\bar{q},h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=h(\bar{k_1},\bar{k_2})\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Sym+LW}{h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=h(\bar{k_1},\bar{k_2})\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,IH/Reg_h}{h(g_1(\bar{l},\bar{m}),g_2(\bar{l},\bar{m}))=h(\bar{k_1},\bar{k_2}),g_2(\bar{l},\bar{m})=\bar{k_2},g_1(\bar{l},\bar{m})=\bar{k_1}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,LW}{g_2(\bar{l},\bar{m})=\bar{k_2},g_1(\bar{l},\bar{m})=\bar{k_1}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Eqf}{g_1(\bar{l},\bar{m})=\bar{k_1}\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Reg_{g_2}}{\Rightarrow f(\bar{l},\bar{m})=\bar{q}}\,Reg_{g_1}$$

By proposition 3.10 $\mathcal{D}$ can be transformed into a derivation $\mathcal{D}^*$ where all the admissible rules are eliminated. $\mathcal{D}^*$ is bounded by a function $\sigma(l,m)=k+\delta_0(l,m)+\delta_1(l,m)+\gamma(g_1(l,m),g_2(l,m))$.

σ is composed of the functions $+,\delta_0,\delta_1,\gamma,g_1,g_2$ that all are in $\mathcal{E}^r, r\geq 2$. Since $\mathcal{E}^r$ is closed under composition σ is also in $\mathcal{E}^r$.

Primitive recursion: I show that the proposition is true for a primitive recursive function f with a sub-induction on the recursion variable n.

$$f(m,0)=g(m)$$
$$f(m,n+1)=h(n,m,f(n,m))$$

Sub basis step: $n = 0$. Assume $g \in \mathcal{E}^r$. By the main induction hypothesis there is a number $p_0 \in N$ and a derivation $\mathcal{D}_{00}$ such that $\mathcal{D}_{00} \vdash^{\mathrm{PRA}}_{\delta(m)} \Rightarrow g(\bar{m}) = \bar{p_0}$. By proposition 3.7 the rule

$$\frac{g(\bar{m}) = \bar{p_0}, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ Reg_g$$

is admissible in PRA. By Reg_g, the inference rule for $f0$, one instance of the derivable rule of symmetry and one instance of transitivity one obtains a derivation $\mathcal{D}_0$ of $f(\bar{m}, \bar{0}) = \bar{p_0}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{f(0,\bar{m}) = \bar{p_0}, g(\bar{m}) = f(0,\bar{m}), g(\bar{m}) = \bar{p_0} \Rightarrow f(\bar{m},\bar{0}) = \bar{p_0}}{g(\bar{m}) = f(0,\bar{m}), g(\bar{m}) = \bar{p_0} \Rightarrow f(\bar{m},\bar{0}) = \bar{p_0}}\ Trans}{f(0,\bar{m}) = g(\bar{m}), g(\bar{m}) = \bar{p_0} \Rightarrow f(\bar{m},\bar{0}) = \bar{p_0}}\ Sym, LW}{g(\bar{m}) = \bar{p_0} \Rightarrow f(\bar{m},\bar{0}) = \bar{p_0}}\ f0}{\Rightarrow f(\bar{m},\bar{0}) = \bar{p_0}}\ Reg_g$$

Sub induction hypotheses (IH 2): Suppose the proposition is true for $n = k$.

Sub induction step: I show that the proposition is true for $k + 1$. Let f be in $\mathcal{E}^r$. By definition $f(m, k+1) = h(m, k, f(m, k))$. By IH 2 there is a $p_1 \in N$ such that $f(m, k) = p_1$ and $\mathcal{D}_{10} \vdash^{\mathrm{PRA}}_{\sigma(m,k)} \Rightarrow f(\bar{m}, \bar{k}) = \bar{p_1}, \sigma \in \mathcal{E}^r$.

By the admissibility of Reg_f and instances of the equality rules $f(\bar{m}, \bar{k})$ can be replaced by $\bar{p_1}$ in h:

$$\dfrac{\dfrac{\dfrac{\dfrac{h(\bar{m},\bar{k},f(\bar{m},\bar{k})) = h(\bar{m},\bar{k},\bar{p_1}), \bar{m} = \bar{m}, \bar{k} = \bar{k}, f(\bar{m},\bar{k}) = \bar{p_1} \Rightarrow \Delta}{\bar{m} = \bar{m}, \bar{k} = \bar{k}, f(\bar{m},\bar{k}) = \bar{p_1} \Rightarrow \Delta}\ Eqf}{\bar{k} = \bar{k}, f(\bar{m},\bar{k}) = \bar{p_1} \Rightarrow \Delta}\ Ref}{f(\bar{m},\bar{k}) = \bar{p_1} \Rightarrow \Delta}\ Ref}{\Rightarrow \Delta}\ Reg_f$$

By the main induction hypothesis there is a $p_2 \in N$ and a derivation $\mathcal{D}_{11}$: $\mathcal{D}_{11} \vdash^{\mathrm{PRA}}_{\gamma(k,m,p_1)} \Rightarrow h(\bar{m}, \bar{k}, \bar{p_1}) = \bar{p_2}, \gamma \in \mathcal{E}^r$. By the admissibility of Reg_h, the inference rule for $frec$ and some instances of the equality rules a derivation $\mathcal{D}_1$ of $f(\bar{m}, k+1) = \bar{p_2}$ is obtained.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f(\bar{m},k\dot{+}1)=\bar{p_2},h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=f(\bar{m},k\dot{+}1),h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=\bar{p_2}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}{h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=f(\bar{m},k\dot{+}1),h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=\bar{p_2}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Trans}{f(\bar{m},k\dot{+}1)=h(\bar{m},\bar{k},f(\bar{m},\bar{k})),h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=\bar{p_2}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Sym,LW}{h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=\bar{p_2}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,frec}{h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=\bar{p_2},h(\bar{m},\bar{k},\bar{p_1})=\bar{p_2},h(\bar{m},\bar{k},\bar{p_1})=h(\bar{m},\bar{k},f(\bar{m},\bar{k}))\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,LW}{h(\bar{m},\bar{k},\bar{p_1})=\bar{p_2},h(\bar{m},\bar{k},\bar{p_1})=h(\bar{m},\bar{k},f(\bar{m},\bar{k}))\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Trans}{h(\bar{m},\bar{k},f(\bar{m},\bar{k}))=h(\bar{m},\bar{k},\bar{p_1}),\bar{m}=\bar{m},\bar{k}=\bar{k},f(\bar{m},\bar{k})=\bar{p_1}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Reg_h}{\bar{m}=\bar{m},\bar{k}=\bar{k},f(\bar{m},\bar{k})=\bar{p_1}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Sym,LW}{\bar{k}=\bar{k},f(\bar{m},\bar{k})=\bar{p_1}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Eqf}{f(\bar{m},\bar{k})=\bar{p_1}\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Ref}{\Rightarrow f(\bar{m},k\dot{+}1)=\bar{p_2}}\,Ref\,Reg_f$$

Let $|\mathcal{D}_0| = c$ and $|\mathcal{D}_1| = c'$. By proposition 3.10 these can be transformed into derivations $\mathcal{D}_0^*$ and $\mathcal{D}_1^*$ with height $c + \delta(m)$ and $c' + \sigma(k,m) + \gamma(k,m,f(k,m))$ respectively. This gives the following primitive recursive definition of σ:

$$\begin{aligned}\sigma(m,0) &= c + \delta(m)\\ \sigma(m,k+1) &= c' + \sigma(m,k) + \gamma(m,k,f(m,k))\end{aligned}$$

Through some fiddling one can rewrite the definition to fit the schema for primitive recursion. In order to show that σ is in $\mathcal{E}^r$, It must be shown that σ is also defined by limited recursion, see figure 3.1. By

$$\begin{array}{lllll}\sigma(m,k+1) & = c' + \sigma(m,k) & + & \gamma(m,k,f(m,k)\\ & = c'\sigma(k-1,m) & + & \gamma(k-1,m,f(k-1,m)\\ & & + & \gamma(m,k,f(m,k))\\ & \vdots & & \\ & = c' + \sigma(0,m) & + & \gamma(0,mf(0,m))\\ & & + & \gamma(1,mf(1,m))\\ & \vdots & & \\ & & + & \gamma(m,kf(m,k))\end{array}$$

In other words:

$$\sigma(m,k) = c + \delta(m) + \sum_{i=0}^{k-1}\big(c' + \gamma(m,i,f(m,i))\big)$$

Figure 3.1: Definition of σ by limited recursion.

assumption $+, \delta, \gamma, f \in \mathcal{E}^r, r \geq 2$. $\mathcal{E}^r$ is closed under bounded sum for $r \geq 2$. Thus σ is defined by limited recursion over functions in $\mathcal{E}^r$.

This concludes the proof that for all primitive recursive functions f and for all natural numbers $m_1, \ldots, m_n$ there exists a number n such that $\Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{n}$ is derivable in PRA. Furthermore I have shown that the height of such derivations are bounded by a function in the same Gregorczyk class as f. It is now easy to show that all primitive recursive functions are well defined in PRA:

Corollary 3.13 *Let f be an n-ary primitive recursive function in Gregorczyk class $\mathcal{E}^r$, $r \geq 2$. Then for all $m_1, \ldots, m_n \in N$ there exists a derivation $\mathcal{D} \in$* PRA*, that contains only nonlogical rules, and a primitive recursive function $\sigma \in \mathcal{E}^r$ such that*

$$\mathcal{D} \vdash^{\mathrm{PRA}}_{\sigma(m_1,\ldots,m_n)} \Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n)).$$

Proof. The result follows from proposition 3.12. It is already shown that if f is a primitive recursive function, then for all $m_1, \ldots, m_n$ in N there is a k in N and a derivation $\mathcal{D} \vdash^{\mathrm{PRA}}_{\sigma(m_1,\ldots,m_n)} \Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$.

When the sequent $\Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$ is derivable in PRA the rule

$$\frac{f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ Reg_f$$

is admissible in PRA by proposition 3.7.

By using one instance of NN, k instances of RN, one instance of Reg_f and *Repl*, one obtains a derivation $\mathcal{D}'$ of $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, where $\bar{k}$ is denoted by $\mathcal{S}^k(0)$. Without loss of generality it can be assumed that $n = 2$.

$$\dfrac{\dfrac{\dfrac{\dfrac{N(f(\bar{l},\bar{m})), f(\bar{l},\bar{m}) = S^k\bar{0}, N(S^k\bar{0}), \ldots, N(\bar{0}) \Rightarrow N(f(\bar{l},\bar{m}))}{f(\bar{l},\bar{m}) = S^k\bar{0}, N(S^k\bar{0}), \ldots, N(\bar{0}) \Rightarrow N(f(\bar{l},\bar{m}))}\ Repl}{N(S^k\bar{0}), \ldots, N(\bar{0}) \Rightarrow N(f(\bar{l},\bar{m}))}\ Reg_f}{\vdots\ \}k-1}}{\dfrac{N(S\bar{0}), N(\bar{0}) \Rightarrow N(f(\bar{l},\bar{m}))}{N(\bar{0}) \Rightarrow N(f(\bar{l},\bar{m}))}\ RN}}{\Rightarrow N(f(\bar{l},\bar{m}))}\ NN$$

$\mathcal{D}'$ has height $c + f(m_1, \ldots, m_n)$. By proposition 3.10 $\mathcal{D}'$ can be transformed into a derivation $\mathcal{D}'^*$ with only nonlogical rules with height bounded by the function $\sigma'(m_1, \ldots, m_n) = c + f(m_1, \ldots, m_n) + \sigma(m_1, \ldots, m_n)$.

Since σ' is composed of the functions $+, \sigma$ and f, and these are in $\mathcal{E}^r$, σ' is in $\mathcal{E}^r$. □

I have shown that for all primitive recursive functions f and for all numbers $m_1, \ldots, m_n$ there exists a derivation $\mathcal{D} \vdash \Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$ in PRA. I have also shown that the height of such a derivation $\mathcal{D}$ is bounded by a

function in the same Gregorczyk class as f. In the following chapters I show that when f is a strictly growing elementary function, it is possible to yield a derivation of $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$ with height bounded by a function linear in the arguments of f. The derivation is obtained by the use of cut formulas.

Kapittel 4

Søk etter snitt i induktive strukturer

Hvis man har en induktivt definert mengde kan man vise egenskaper som vil gjelde for alle elementer i mengden. Et matematisk induksjonbevis vil gjerne være på formen: Du har en egenskap P som du vil vise gjelder for alle elementer i en mengde. Først må du finne og vise et eller flere basistilfeller med egenskapen P, for eksempel tallet 0 hvis mengden er de naturlige tallene. Anta så, induksjonshypotese (IH), at P gjelder for et vilkårlig tall n. Hvis du klarer ved hjelp av IH å vise at P gjelder for $n + 1$ kan du slå fast at P vil gjelde for alle elementene i mengden.

Matematisk induksjon er vanskelig å mekanisere. Det innebærer et element av gjetting å finne ut akkurat hva induksjonshypotesen skal være. Ofte viser det seg at man må sikte mot en mer generell egenskap enn den man ønsker å vise. Videre er det ikke alltid opplagt å vite på hvilket punkt i beviset man trenger induksjonshypotesen.

Hvis man skal verifisere at resultatet av å kalle en funksjon f med et argument m gir et naturlig tall kan man gjøre dette ved å regne ut funksjonverdien. I en formell teori over de unære tallene vil bevisene etterhvert bli veldig store for raskt voksende funksjoner slik som fakultetsfunksjonen. For store nok verdier kan de bli uhåndterlige, selv om de prinsipielt sett er løsbare.

Resultatet i denne oppgaven bygger videre på en idé introdusert av Jervell og Zhang [15] om å konstruere hjelpesetninger eller snitt som har innbakt strukturen fra et induksjonsbevis. Dermed kan man oppnå noe av det samme som man oppnår med induksjon ved å bruke snitt istedenfor. Eksempelvis ville det skisserte induksjonsbeviset over kunne uttrykkes som

$$P(0) \wedge \forall x(P(x) \rightarrow P(Sx))$$

hvor Sx står for etterfølgeren til x. Snittformelene som introduseres senere i dette kapitlet vil være på denne formen. Hva man klarer å vise ved hjelp av denne typen snitt avhenger av hvordan P er formulert. I de følgende kapitlene viser jeg at man kan definere predikater med en bestemt struktur som kan brukes i snitt av denne typen. Predikatene formuleres ut i fra den primitivt rekursive definisjonen av funksjonstermen i påstanden man vil utlede. Påstandene jeg ønsker å vise er på formen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$. Dette kan leses som "funksjonen f anvendt på argumentene $m_1, \ldots, m_n$ er et naturlig tall". Det vil si det samme som at f er definert i argumentene $m_1, \ldots, m_n$.

4.1 Abstraher med snitt

Snittregelen i sekventkalkyle for klassisk logikk (LK) beskrives i avsnitt 2.2.3. Som nevnt kan snittintroduksjon i klassisk logikk sammenlignes med bruk av lemmaer i matematiske bevis. Problemet med snittregelen er at den kan lede til en uendelig forgrening i søkerommet. Dette har gjort det vanskelig å bruke snittregelen i automatisk bevissøk. Snittformelen kan være en hvilken som helst utledbar formel.

For å møte problemene snittregelen skaper viste Gentzen at det alltid lar seg gjøre å eliminere snitt i en (LK-)utledning. Men i noen tilfeller er en interessert i å bruke snittregelen. I kapittel 6 viser jeg at en direkte utledning $\mathcal{D}$ i PRA av et utsagn på formen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, når $f(m_1, \ldots, m_n) = n$, har høyde $|\mathcal{D}| \geq n$. Hvis f vokser eksponensielt i $m_1, \ldots, m_n$ kan $\mathcal{D}$ fort bli veldig høy. Ved bruk av passende snittformler kan man redusere høyden på utledningen drastisk. I neste kapittel gis en strategi for å introdusere snittformler i slike utledninger når f er elementær og strengt voksende.

Hvis en fjerner alle lemmaer i et større matematisk bevis, ville beviset bli uforholdsmessig stort og uhåndterlig. Likeledes gir prosessen med å fjerne snitt i verste fall hypereksponensiell vekst i høyden av den opprinnelige utledningen ved korrolar 3.5.

Boolos [2] argumenterer for å bruke bevissystemer som bevarer snittregelen. Han viser et eksempel på en utledning som eksploderer i høyde hvis man eliminerer snitt. Aksiomene under beskriver det unære tallsystemet. Symbolet + står for sammensetning av 1-ere, d står for dobling og Lx står for "x er et tall".

$$\forall x \forall y \forall z + x + ys = + + xyz$$
$$\forall x dx = + + xx$$
$$L1$$
$$\forall x(Lx \supset L + x1)$$

La H_n være en forkortelse for

$$H_n = L\underbrace{dd \ldots d}_{2^n}1$$

Da er påstanden $H_3 = Ldddddddd1$. Boolos viser at en utledning i et system uten snitt av H_7 inneholder i overkant av 10^{38} symboler, "more symbols than there are nanoseconds between Big Bangs".

Den korteste utledningen i naturlig deduksjon *med* snitt inneholder færre enn 3280 symboler som er et langt mer overkommelig antall.

4.2 Induktive predikater

Introduksjonen av snitt kan ses på som en form for abstrahering. Jervell og Zhang [15] viderefører ideen fra Boolos [2] med å bruke snitt i utledninger av typen "$f(\bar{m}_1, \ldots, \bar{m}_n)$ er et tall". De ønsker å vise at ved å legge til noen hjelpeaksiomer til en tallteori kan man alltid finne snitt for å korte ned en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ når f er elementær. For å få til dette introduserer Jervell og Zhang induktive predikater og termer. Et predikat P kalles induktivt hvis det har tilsvarende egenskaper som N. Det må være mulig å utlede $P(0) \wedge \forall x(P(x) \supset P(Sx))$. De induktive predikatene brukes i snittformler.

En direkte, det vil si snittfri, utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ krever like mange skritt som tallverdien av uttrykket, se kapittel 6. Med snitt kan man får utledninger med lengde lineær i argumentene til f. Dette vises i neste kapittel. Hvis f vokser eksponensielt innebærer dette en dramatisk forbedring. Zhang [34] har vist at de mest effektive snittene vil ha høy kompleksitet målt i form av nøstede kvantorer. Med andre ord kan man forvente at jo mer kompleks en funksjon er, det vil her si jo raskere den vokser, jo flere nøstede kvantorer trengs i predikatet som brukes i snittformelen for funksjonen.

Jervell og Zhang [15] gir konkrete eksempler på induktive predikater for noen elementære funksjoner. De gir også en strategi for å konstruere et induktiv predikat for en funksjon f, når f er definert ved henholdsvis komposisjon bundet sum eller bundet produkt over induktive funksjoner. Jervell og Zhang

definerer ikke en fullstendig tallteori. Jeg har definert et formelt system for å kunne etterprøve Jervell og Zhangs påstander om induktive predikater. Videre har jeg villet undersøke om det går an å generere induktive predikater mekanisk, ut i fra rent syntaktiske kriterier. Jeg bygger videre på strategien deres, men enkelte av predikatene har en litt annen form, se avsnitt 4.3.

Jeg gir en mer formell definisjon av induktive predikater enn den til Jervell og Zhang. Definisjonen har både en syntaktisk og en semantisk del. Ved å gi en induktiv definisjon kan jeg senere bevise egenskaper ved predikatene over strukturen deres.

Definisjon 4.1 (Induktive predikater) *La S angi et bevissystem som inneholder de to første Peano-aksiomene $N(0)$ og $N(x) \supset N(Sx)$ og definerende likninger for de primitivt rekursive funksjonene. Induktive predikater i S defineres induktivt:*

1. *N er induktivt.*

2. *La $F_0, \ldots, F_{n+1}$ være induktive predikater, f være en $n+1$-ær primitiv rekursiv funksjon. F er induktiv hvis*

   (a) *F er definert ved:*

   $$\begin{aligned} F(y) = & F_0(y) \wedge \forall x_1, \ldots, x_n (F_1(x_1) \wedge \cdots \wedge F_n(x_n) \\ & \supset F_{n+1}(f(x_1, \ldots, x_n, y))) \end{aligned}$$

   (b) $\vdash_S F(0) \wedge \forall x(F(x) \supset F(Sx))$ *og*
   $\vdash_S F(x) \supset N(x)$

Merk at når f er unær blir predikatet:

$$F(x) = F_0(x) \wedge F_1(f(x))$$

For at et predikat F skal være induktivt, må det altså ha de samme egenskapene som N og for alle x, hvis $F(x)$ må man kunne utlede $N(x)$. Når F er definert som over sier jeg at F beskriver termen $f(x_1, \ldots, x_n)$.

Definisjon 4.2 (Induktive termer) *En term $f(\bar{m}_1, \ldots, \bar{m}_n)$ er induktiv hvis den beskrives av et induktivt predikat.*

Når man ønsker å vise at bestemte påstander kan utledes er det viktig å presisere hvilket bevissystem de kan utledes i. Heretter snakker jeg derfor om induktive predikater med hensyn på et gitt bevissystem.

4.3 Mekanisk generering av predikater

Jeg har undersøkt muligheten for å generere induktive predikater mekanisk, ut i fra rent syntaktiske kriterier. Målet er å lage en algoritme som søker etter snittformler for utsagn av typen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ ut i fra definisjonen av f, samt eventuelle tilleggsaksiomer om f. Programmet som søker etter snitt er integrert i den automatiske teorembeviseren PESCA*. Begge deler beskrives i kapittel 9.

Ønsket er å formulere predikater som fanger opp de elementære funksjonenes totale natur, det vil si at de er definert for alle naturlige tall. Ved hjelp av induktive predikater kan man slutte at $f(\bar{m}_1, \ldots, \bar{m}_n)$ er i mengden av naturlige tall, uten å regne ut tallet. For funksjoner som vokser så raskt som fakultet er det nødvendig å abstrahere vekk tallverdien for å få en håndterlig utledning når argumentet bli stort nok.

4.3.1 Bemerkning om argumentplassering

I definisjonsskjemaet for primitiv rekursjon

$$\begin{aligned} f(x_1, \ldots, x_n, 0) &= g(y_1, \ldots, y_n) \\ f(x_1, \ldots, x_n, Sy) &= h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n)) \end{aligned}$$

kalles argumentet lengst til høyre rekursjonsargumentet. Det styrer antall kall på funksjonen h. Definisjonsskjemaet krever at rekursjonsargumentet og det rekursive kallet begge alltid er på argumentplassen lengst til høyre. Merk at hvis h også er primitivt rekursiv kan man være sikker på at $f(x_1, \ldots, x_n)$ substitueres inn på plassen til rekursjonsargumentet i h. Predikatet for addisjon, $A(x) = N(x) \wedge \forall y(N(y) \supset N(+yx))$ omtaler rekursjonsargumentet i addisjonsfunksjonen. Dette er et viktig poeng. Hadde x og y byttet plass, det vil si: $A(x) = N(x) \wedge \forall y(N(y) \supset N(+xy))$, så hadde det ikke gått å vise at A er induktivt.

For å få en entydig algoritme som enkelt lar seg mekanisere krever jeg at alle rekursivt definerte funksjoner følger det vanlige skjemaet for primitiv rekursjon[1]. Hvis funksjonen h er kommutativ, det vil si $h(a, b) = h(b, a)$ så trenger man strengt tatt ikke følge skjemaet gitt over, da rekkefølgen

[1] I PESCA* brukes et definisjonsskjema hvor rekursjonsargumentet er lengst til venstre. Dette er fordi jeg i utgangspunktet brukte notasjon som lå tettere opp til Jervell og Zhang [15]. Så lenge kallet på f i h også konsekvent gjøres på plassen lengst til venstre utgjør dette kun en syntaktisk forskjell og har ingenting å si for utledningene. Jeg har derfor valgt å ikke endre på dette i PESCA* eller i beviser generert i PESCA*, se for eksempel figur 6.1.

på argumentene ikke spiller noen rolle for resultatet. Jervell og Zhang [15] benytter seg av dette. De definerer multiplikasjon slik:

$$\begin{aligned} \times 0y &= y \\ \times Sxy &= +y \times xy \end{aligned}$$

hvor det rekursive kallet ikke gjøres på plassen til rekursjonsvariabelen i addisjonsfunksjonen. Definisjonen over går bra så lenge f er kommutativ, siden $+xy = +yx$. Men når f ikke er kommutativ kan man ikke velge rekkefølgen på argumentene vilkårlig, for eksempel $\neg\forall x, y(x^y = y^x)$. Med denne definisjonen klarer Jervell og Zhang å utlede et induktivt predikat for multiplikasjon

$$M(x) = A(x) \wedge \forall A(y) \supset N(\times xy).$$

uten en regel for assosiativitet. Problemet er at med et slikt predikat for multiplikasjon klarer man ikke å definere et predikat for den vanlige eksponensialfunksjonen.

Ved å definere $\times$ slik

$$\begin{aligned} \times x0 &= x \\ \times xSy &= +x \times xy \end{aligned}$$

får man ikke til å utlede et induktivt predikat for multiplikasjonsfunksjonen i PRA. Som nevnt i kapittel 3 er PRA mindre uttrykkskraftig enn Peanoaritmetikk, siden jeg ikke har med induksjonsaksiomet. De definerende likningene for elementære funksjoner kan ses på som regneregler. De kan brukes til å vise påstander som $2+2 = 4$ og ikke noe særlig mer. For eksempel er det ikke mulig å utlede en såpass basal egenskap ved addisjon som $x+y = y+x$, se avsnitt 5.3. Jervell påviste at ved å legge til en regel for assosiativitet av addisjon kan predikatet

$$M(x) = A(x) \wedge \forall A(y) \supset A(\times xy)$$

vises å være induktivt også når multiplikasjon defineres ved det vanlige skjemaet for primitiv rekursjon. Kravet om å følge definisjonsskjemaet for primitiv rekursjon tydeliggjør derfor hvilke egenskaper ved f som trengs for å definere et induktivt predikat. Det nye predikatet M kalles flatt (definisjon 5.9). Når predikatet M er flatt kan man også utlede et induktivt predikat for eksponensialfunksjonen. Mer generelt viser jeg i avsnitt 5.3 at hvis en funksjon f er definert ved rekursjon over en assosiativ funksjon med flatt predikat, så kan man konstruere et induktivt predikat for f. Dermed får enkelte av predikatene større kompleksitet med hensyn på kvantornøsting enn dem som

Jervell og Zhang definerer. Et flatt predikat for M får dybde $3A + 3$, se definisjon 2.29.

Jeg kunne også ha utledet et induktivt predikat for multiplikasjon ved å legge til kommutativitet for addisjon i stedet for assosiativitet. Men da hadde jeg ikke fått et predikat for bundet produkt. På samme måte kunne jeg lagt til assosiativitet for addisjon og kommutativitet for multiplikasjon for å få et predikat for bundet produkt. Da hadde utledningen av at predikatet for bundet produkt er induktivt blitt noe kortere. For å få en mest mulig generell algoritme velger jeg å kun legge til aksiomer for assosiativitet for pluss og gange.

Kapittel 5

Elementære strengt voksende funksjoner er induktive

Jeg viser at en delmengde av de Kalmárelementære funksjonene er induktive i en utvidet versjon av bevissystemet PRA, definert i kapittel 3. Delmengden består av alle elementære funksjoner unntatt modifisert subtraksjon og funksjoner f hvor modifisert subtraksjon inngår i definisjonen. Som nevnt i introduksjonen (avsnitt 1.4) sier jeg for enkelhets skyld at en funksjon f er induktiv hvis den er elementær og strengt voksende (definisjon 2.5). Resultatet vises ved å gi induktive predikater for de Kalmárelementære initialfunksjonene unntatt modifisert subtraksjon. Videre viser jeg at de induktive predikatene er lukket under komposisjon, bundet sum og bundet produkt, jamfør definisjon 2.8.

Algoritmen for å konstruere induktive predikater er integrert i den automatiske teorembeviseren PESCA*, beskrevet i kapittel 9. I tillegg til definisjonsskjemaene nevnt over kan programmet ta funksjoner definert ved primitiv rekursjon. Teorem 2.26 gir at klassen $\mathcal{E}^3$ er klassen av Kalmárelementære funksjoner $\mathcal{E}'$ og $\mathcal{E}^3$ er lukket under begrenset rekursjon. For å vise at en funksjon $f \in \mathcal{E}^r$ er definert ved begrenset rekursjon må man vise at den majoriseres av en funksjon i samme klasse. Programmet gjør ingen slik sjekk, men det har en semantisk begrensning: Hvis funksjonen f er definert ved rekursjon over en funksjon g må systemet inneholde aksiomer for assosiativitet av h. Senere viser jeg (korrolar 8.11) at hvis en funksjon f er definert ved primitiv rekursjon over induktive funksjoner g og h, hvor g er en av initialfunksjonene og PRA inneholder en regel for assosiativitet av h, så er f elementær. Dermed kan ikke programmet tildele predikater til funksjoner som ikke er elementære.

Flere av bevisene for at bestemte elementære funksjoner er induktive gis i sekventkalkyle. Utledningene er generert i den automatiske teorembeviseren

PESCA* beskrevet i kapittel 9. Det er enkelt å sjekke maskinelt at beviser utført i sekventkalkyle er korrekte fordi det er et formelt system, men de kan være nokså omstendelige å lese. For lesbarhetens skyld har jeg skrevet ut noen av bevisene i dette kapitlet med tekst. De likner ganske mye på hverandre, så jeg tar kun med de jeg anser som viktigst. Utledningene i sekventkalkyle finnes i tillegg A.

I de følgende bevisene ser jeg kun på de elementære funksjonene, så jeg klarer meg med en delmengde av aksiomene i PRA. I tillegg til aksiomer for likhet trengs aksiomene definert i figur 5.1.

$$(5.1) \quad N(0)$$
$$(5.2) \quad N(x) \supset N(Sx)$$
$$(5.3) \quad \mathcal{O}(x) = 0$$
$$(5.4) \quad \mathcal{I}_i^n(x_1, \ldots, x_n) = x_i$$
$$(5.5) \quad P(0) = \mathcal{O}$$
$$(5.6) \quad P(Sx) = x$$
$$(5.7) \quad \dot{-}x0 = x$$
$$(5.8) \quad \dot{-}Sxy = p\mathcal{I}_3^3(x, y, \dot{-}(x, y))$$
$$(5.9) \quad +x0 = x$$
$$(5.10) \quad +xSy = S(+xy)$$
$$(5.11) \quad \times x0 = 0$$
$$(5.12) \quad \times xSy = +y(\times xy)$$
$$(5.13) \quad \sum_{i \leq 0} g(x_1, \ldots, x_n, i) = g(x_1, \ldots, x_n, 0)$$
$$(5.14) \quad \sum_{i \leq Sy} g(x_1, \ldots, x_n, i) = +g(x_1, \ldots, x_n, Sy) \sum_{i \leq y} g(x_1, \ldots, x_n, i)$$
$$(5.15) \quad \prod_{i \leq 0} g(x_1, \ldots, x_n, i) = g(x_1, \ldots, x_n, 0)$$
$$(5.16) \quad \prod_{i \leq Sy} g(x_1, \ldots, x_n, i) = \times g(x_1, \ldots, x_n, Sy) \prod_{i \leq y} g(x_1, \ldots, x_n, i)$$
$$(5.17) \quad Chg_1, \ldots, g_m(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

Figur 5.1: Definerende likninger for de elementære funksjonene

I aksiomer 5.13 til 5.17 er h og g_i funksjonsvariable. Ved hjelp av disse aksiomene og aksiomskjemaene kan man konstruere alle de elementære funksjonene, jamfør definisjon 2.8.

I tråd med regelskjemaet til Negri og von Plato definert i kapittel 3 skrives

aksiomene om til ikkelogiske slutningsregler.

5.1 Predikater for initialfunksjonene

Først viser jeg at de Kalmárelementære initialfunksjonene, definisjon 2.8, unntatt modifisert subtraksjon $\dot{-}$ er induktive.

Aksiomene $N(0)$ og $\forall x(N(x) \supset N(Sx))$ kan til sammen ses på som et induktivt predikat for etterfølgerfunksjonen. Det sier at for ethvert naturlig tall n gir etterfølgerfunksjonen anvendt på n et nytt element i N. Dette predikatet er nødvendig for å kunne konstruere predikater om de andre elementære funksjonene. Funksjonen $+$ er definert ved iterasjon over etterfølgerfunksjonen. Det er lett å se at denne er total over de naturlige tallene; å legge sammen to hele tall gir et helt tall. Det induktive predikatet for addisjon er heller ikke så veldig komplisert. Det kan utledes kun fra de ikkelogiske reglene for N og addisjon i PRA.

Lemma 5.1 *Funksjonen $+$ er induktiv i* PRA.

Bevis. Predikatet for addisjon er definert ved:

$$A(x) = N(x) \wedge \forall y(N(y) \supset N(+yx))$$

Det er nødvendig å vise at $A(0)$ og at hvis $A(x)$ så følger det at $A(Sx)$.

Bevis for $A(0)$

$N(0)$ ved aksiom 5.1.

Anta $N(b)$. Ved aksiom 5.9 er $+b0 = b$. Ved reglene *Sym* og *Repl* kan $+b0$ substitueres i N for b. Dermed får jeg $N(+b0)$. Siden b var vilkårlig valgt kan det konkluderes at $\forall y(N(y) \supset N(+y0))$. Det vil si $A(0)$.

Bevis for $A(x) \supset A(Sx)$

Anta $A(a)$. Må vise at $A(Sa)$. Det vil si $N(Sa)$ og $\forall y(N(y) \supset N(+ySa))$. $N(a)$ og $\forall y(N(y) \supset N(+ay))$ følger fra antakelsen. Fra $N(a)$ følger $N(Sa)$ ved aksiom 5.2.

Anta $N(b)$. Fra antakelsen følger $N(+ba)$. Dermed følger også $N(S(+ba))$ ved aksiom 5.2. $+bSa = S(+ba)$ ved aksiom 5.10 Substitusjon gir $N(+bSa)$ Siden b var vilkårlig valgt kan det konkluderes at $\forall y(N(y) \supset N(+ySa))$ og dermed er $A(Sa)$. Siden a var vilkårlig valgt kan man slutte $\forall x A(x) \supset A(Sx)$.

Dette beviset er også generert i PESCA*, figurer A.1, A.2 og A.3.

Det er nødvendig å vise at hvis $A(x)$ så $N(x)$. Dette følger fra definisjonen av A. □

Lemma 5.2 *Nullfunksjonen $\mathcal{O}$ og projeksjonsfunksjonene $\mathcal{I}_i^n$ er induktive i* PRA.

Bevis. For nullfunksjonen kan man definere følgende predikat:

$$O(x) = N(x) \wedge N(\mathcal{O}(x))$$

$O(0)$ følger opplagt.

$O(x) \supset O(Sx)$

Anta $O(a)$

Det vil si $N(a) \wedge N(\mathcal{O}(a))$

$N(Sa)$ følger fra aksiom 5.2 og

$N(\mathcal{O}(Sa))$ følger fra aksiom 5.2 siden $N(\mathcal{O}(y)) = 0$ for alle y.

Predikatet for $\mathcal{I}_2^3$

$$I_2^3(x) = N(x) \wedge \forall yz(N(y) \wedge N(z) \supset N(\mathcal{I}_2^3(y, x, z)))$$

$\mathcal{I}_i^n(x_1, \ldots, x_n) = x_i$ så for 0 får man $N(0) \wedge N(0)$ som opplagt er utledbart i PRA.

Anta $N(a)$ Da får man ved litt omskrivning $N(a) \wedge N(a) \supset N(Sa) \wedge N(Sa)$ som opplagt kan utledes i PRA.

Man kan definere tilsvarende predikater for alle varianter av $\mathcal{I}_i^n, 1 \leq i \leq n$. □

I praksis sjelden har man sjelden bruk for predikatet til $\mathcal{I}_i^n$ og $\mathcal{O}$. Jeg tar de med for argumentets skyld.

5.2 Komposisjon over induktive predikater

Jeg må nå vise at de induktive predikatene er lukket under komposisjon. Dette gjør jeg ved å gi en framgangsmåte for å konstruere et predikat for en funksjon f når f er definert ved komposisjon over funksjoner $h, j_1, \ldots, j_m$. Først gis en induktiv definisjon for komposisjon av induktive predikater. Dermed kan jeg vise egenskaper ved komposisjon av predikater ved induksjon over oppbyggingen av predikatene.

Definisjon 5.3 (Komposisjon over induktive predikater) *La F og G være induktive predikater. Komposisjonen $F[G]$ hvor G erstatter N i F defineres induktivt:*

1. *$F[G] = G$ når $F = N$*

2. La F være definert ved

$$F(y) = F_0(y) \wedge \forall x_1, \ldots, x_n(F_1(x_1) \wedge \cdots \wedge F_n(x_n) \supset F_{n+1}(f(x_1, \ldots, x_n, y)))$$

Da er $F[G]$ definert ved

$$F[G](y) = F_0[G](y) \wedge \forall x_1, \ldots, x_n(F_1[G](x_1) \wedge \cdots \wedge F_n[G](x_n) \supset F_{n+1}[G](f(x_1, \ldots, x_n, y)))$$

I det følgende viser jeg at de induktive predikatene er lukket under skjemaet over. Det vil si hvis F og G er induktive, så er komposisjonen $F[G]$ induktiv. Jeg må vise at $F[G](0)$ og at $\forall x(F[G](x) \supset F[G](Sx))$ er utledbart. Gangen i beviset er som følger: Ikkelogiske regler som introduserer induktive predikater er admisible i PRA. Disse er like som reglene for N bortsett fra at N er byttet ut med en annen bokstav, for eksempel G. Ved å bytte ut reglene for N med regler for G i utledningen av $F(0) \wedge \forall x(F(x) \supset F(Sx))$ får jeg en utledning av $F[G](0) \wedge \forall x(F[G](x) \supset F[G](Sx))$. Intuisjonen bak dette er at siden de induktive predikatene er like i form som reglene for N så kan jeg bytte ut N med G alle steder i en utledning av et utsagn A og få en utledning av $A[G/N]$. Uttrykket $A[R/P]$ står for utsagnet A hvor predikatet P er erstattet med predikatet R. Først etablerer jeg formelt at denne intuisjonen holder:

Lemma 5.4 *La G være et G3c*-system, P være et unært predikat og A være et førsteordens utsagn hvor P forekommer. La $S \subseteq G$ slik at S inneholder alle ikkelogiske slutningsregler $R_1, \ldots, R_n$ hvor P forekommer. La S' være S hvor P er erstattet med Q overalt hvor P forekommer og la A' være A med Q erstattet for P. Da holder følgende:*

$$\textit{Hvis } \mathcal{D} \vdash^S \Gamma \Rightarrow A \textit{ så } \mathcal{D}' \vdash^{S'} \Gamma' \Rightarrow A'$$

hvor Γ er en muligens tom mengde med utsagn og Γ' er Γ hvor alle forekomster av P er erstattet med Q.

Bevis. Lemmaet vises ved induksjon over oppbyggingen av A med subinduksjon på høyden n av utledningen. Ved snitteliminasjonsteoremet for G3c* kan utledningene antas å være snittfrie.
Basistilfelle: A er atomær.

1. Anta $\mathcal{D} \vdash^S \Gamma \Rightarrow A$ og $|\mathcal{D}| = 0$.

   Ved definisjon 2.33 av **G3c** inneholder Γ enten $\bot$ eller A. Hvis Γ inneholder $\bot$, så inneholder Γ' $\bot$ og A' kan utledes i S'.

Anta Γ inneholder A. Siden A er atomær er A på formen $P(t_1,\ldots,t_n)$ og $A' = Q(t_1,\ldots,t_n)$. Siden Γ' er Γ hvor P er erstattet med Q inneholder $\Gamma' A'$, så kan $\Gamma' \Rightarrow A'$ utledes i S'.

2. Anta lemmaet holder for utledning med høyde $n = k$. Det er nødvendig å vise at den holder for $n = k+1$. I induksjonssteget må man se på den siste regelen som er brukt i utledningen. Siden A er atomær må den siste regelen enten være ikkelogisk eller en venstre-regel.

   (a) Siste regel er en logisk venstre-regel: La $\mathcal{R}$ være $L\vee$.

   $$\frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ C,\Gamma^* \Rightarrow A & D,\Gamma^* \Rightarrow A\end{array}}{C \vee D,\Gamma^* \Rightarrow A}\ L\vee$$

   $\Gamma^* = \Gamma \setminus C \vee D$. Induksjonshypotesen gir $\mathcal{D}_1' \vdash C',\Gamma^{*'} \Rightarrow A'$ og $\mathcal{D}_2' \vdash d',\Gamma^{*'} \Rightarrow A'$. $L\vee$ gir $C' \vee D',\Gamma^{*'} \Rightarrow A'$. De andre tilfellene hvor $\mathcal{R}$ er en logisk venstre-regel kan vises på tilsvarende måte.

   (b) Siste regel er en ikkelogisk regel:

   $$\frac{\begin{array}{ccc}\mathcal{D}_{01} & & \mathcal{D}_{0m}\\ R_1 \Rightarrow A & \ldots & R_n \Rightarrow A\end{array}}{\Rightarrow A}\ Reg_R$$

   Ved induksjonshypotesen for n finnes det utledninger i S' av $R_1' \Rightarrow A',\ldots,R_n' \Rightarrow A'$. S' inneholder Reg_R hvor P er erstattet med Q i $R_1,\ldots,R_n$. Siden alle ikkelogiske regler har vilkårlige formler i suksedenten følger det at A' kan utledes fra $Reg_R[Q/P]$ i S'.

Induksjonstilfelle: La $A = B \wedge C$. De andre tilfellene blir tilsvarende. Anta $\mathcal{D} \vdash^S \Gamma \Rightarrow B \wedge C$. Ved inversjonslemmaet 2.34 er $\Gamma \Rightarrow B$ og $\Gamma \Rightarrow C$ utledbart i S. Induksjonshypotesen gir at $\Gamma' \Rightarrow B'$ og $\Gamma' \Rightarrow C'$ er utledbart i S'. Høyre og-regelen gir $\mathcal{D} \vdash^{S'} \Gamma' \Rightarrow B' \wedge C'$. □

Lemma 5.5 *Reglene*

$$\frac{P(0),\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad \frac{P(Sa),P(a),\Gamma \Rightarrow \Delta}{P(a),\Gamma \Rightarrow \Delta}$$

er admisible i PRA *for alle induktive predikater P.*

Bevis. Ved definisjon 4.1 er $\Rightarrow P(0)$ og $\Rightarrow P(a) \supset P(Sa)$ utledbart. Ved å bruke snitt på $P(0)$ følger $\Gamma \Rightarrow \Delta$. Venstre implikasjon og snitt gir $P(a),\Gamma \Rightarrow \Delta$:

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \\ \Rightarrow P(0) & P(0),\Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta}\ Cut$$

$$\dfrac{\dfrac{\mathcal{D}_0}{\Rightarrow P(a) \supset P(Sa)} \qquad \dfrac{P(a), \Gamma \Rightarrow \Delta, P(a) \quad P(Sa), P(a), \Gamma \Rightarrow \Delta}{P(a) \supset P(Sa), P(a), \Gamma \Rightarrow \Delta}\, L\supset}{P(a), \Gamma \Rightarrow \Delta}\, Cut$$

Fra lemmaene over kan jeg vise at $F[G](0) \wedge \forall x(F[G](x) \supset F[G](Sx))$ er utledbart i PRA, når F og G er induktive. Men jeg er ennå ikke helt i havn. For at de induktive predikatene skal være lukket under skjemaet for komposisjon, må jeg også vise at $F[G](x) \supset N(x)$ er utledbart i PRA. Hvis jeg kan vise at $F[G](x)$ medfører $G(x)$ vil det følge at $N(x)$, siden G er induktiv per antakelse. Denne sammenhengen kan vises ved komposisjon over oppbyggingen av predikatene, så her får jeg bruk for den induktive definisjonen av komposisjon over induktive predikater.

Lemma 5.6 $\vdash_D F[G](\bar{m}) \Rightarrow \vdash_D G(\bar{m}),$ *F, G induktive predikater*

Bevis. Lemmaet vises ved induksjon over oppbyggingen av F.
Basissteg: $F = N$
Ved definisjonen av komposisjon er $N[G] = G$, så $N[G](x)$ gir $G(x)$.
Induksjonshypotese: Lemma 5.6 holder for et induktivt predikat H.
Induksjonssteg: Må vise at lemmaet holder for predikatet F når

$$F(x_1) = F_1(x_1) \wedge \forall x_2, \ldots, x_n(F_2(x_2) \wedge \cdots \wedge F_n(x_n) \supset F_{n+1}(f(x_1, \ldots, x_n))).$$

Anta $F[G](\bar{m})$. Ved definisjonen av komposisjon er

$$\begin{aligned} F[G](\bar{m}) = & F_1[G](\bar{m}) \wedge \forall x_2, \ldots, x_n(F_2[G](x_2) \wedge \cdots \wedge F_n[G](x_n) \\ & \supset f_{n+1}[G](f(m, x_2, \ldots, x_n)) \end{aligned}$$

Ved induksjonshypotesen følger $G(\bar{m})$ fra $F_1[G](\bar{m})$. □

Nå er jeg klar for å vise at de induktive predikatene er lukket under skjemaet for komposisjon:

Teorem 5.7 *La F, G være induktive predikater i et bevissystem S. Da er komposisjonen $F[G]$ også induktiv.*

Bevis. Det er nødvendig å vise at $F[G](0)$ og $\forall x(F[G](x) \supset F[G](Sx))$ er utledbart i S og at $F[G](x) \supset N(x)$ er utledbart i S.

1. Siden F er induktivt er $F(0)$ og $\forall x(F(x) \supset F(Sx))$ utledbart i S. Ved lemma 5.5 er reglene

$$\frac{G(0), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad \frac{G(Sa), G(a), \Gamma \Rightarrow \Delta}{G(a), \Gamma \Rightarrow \Delta}$$

admisible i S. Ved å la disse reglene erstatte reglene for N følger det fra lemma 5.4 at man kan finne utledninger $\Rightarrow F[G/N](0) = F[G](0)$ og $\Rightarrow \forall x(F(x) \supset F(Sx))[G/N] = \forall x(F[G](x) \supset F[G](Sx))$.

2. $F[G](\bar{m}) \overset{\text{lemma } 5.6}{\Rightarrow} G(\bar{m}) \overset{\text{def. av ind.pred}}{\Rightarrow} N(\bar{m})$. □

Nå har jeg vist at induktive predikater er lukket under skjemaet i definisjon 5.3. Dette skjemaet kan brukes til å definere induktive predikater for funksjoner definert ved skjemaet for komposisjon over primitivt rekursive funksjoner:

Teorem 5.8 *La $g_1, \ldots, g_m, h$ være elementære funksjoner og definer f ved*

$$f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)).$$

La følgende være induktive predikater for henholdsvis $g_1, \ldots, g_m, h$:

$$\begin{aligned}
G_1(x_1) &= G_{11}(x_1) \\
&\quad \wedge \forall x_2, \ldots, x_n(G_{12}(x_2) \wedge \cdots \wedge G_{1n}(x_n) \supset G_{1n+1}(g(x_1, \ldots, x_n))) \\
&\vdots \\
G_m(x_1) &= G_{m1}(x_1) \\
&\quad \wedge \forall x_2, \ldots, x_n(G_{m2}(x_2) \wedge \cdots \wedge G_{mn}(x_n) \supset G_{mn+1}(g(x_1, \ldots, x_n))) \\
x_1 : H &= H_1(x_1) \\
&\quad \wedge \forall x_2, \ldots, x_n(H_2(x_2) \wedge \cdots \wedge H_m(x_m) \supset H_{m+1}(h(x_1, \ldots, x_m)))
\end{aligned}$$

Da er følgende også induktivt:

$$\begin{aligned}
F(x_1) = & G_1[H](x_1) \wedge G_2[H_2](x_1) \wedge \cdots \wedge G_m[H_m](x_1) \wedge \\
& \forall x_2, \ldots . x_n(G_{12}[H](x_2) \wedge G_{22}[H_2](x_2) \wedge \cdots \wedge G_{m2}[H_m](x_2) \\
& \qquad \wedge \cdots \wedge G_{12}[H](x_m) \wedge G_{22}[H_2](x_m) \\
& \qquad \wedge \cdots \wedge G_{m2}[H_m](x_m) \supset H_{m+1}(f(x_1, \ldots, x_n)))
\end{aligned}$$

Bevis. Det er nødvendig å vise $F(0)$ og at hvis $F(x)$ så følger det at $F(Sx)$. Uten tap av generalitet la $n = m = 2, g_1 = g, g_2 = j, G_1 = G, G_2 = J$.

Først vises $F(0)$

Fra teorem 5.7 følger det at $G[H](0)$ og $J[H_1](0)$ når $G[H]$ og $J[H_1]$ er induktive. Det vil si $G_1[H](0) \wedge \forall y(G_2[H](y) \supset G_3[H](gy0))$ og $J_1[H_2](0) \wedge \forall y(J_2[H_2](y) \supset J_3[H_2](jy0))$.

Anta $b : G_1[H]$ og $b : J_1[H_1]$. Da er : $G_3[H](g(b0))$ og $J_3[H_2](j(b0))$. Fra lemma 5.6 følger det at $H(g(b0))$ og $H_2(j(b0))$. Dette gir $H_1(g(b0)) \wedge$

$\forall y(H_2(y) \supset H_3(hg(b0)y))$. Siden $H_1(j(b0))$ får man $H_3(h(g(b0), j(b0)))$. Ved definisjon er $f(b0) = h(g(b0), j(b0))$. Ved å substituere $f(b0)$ for $h(g(b0), j(b0))$ får man $H_2(f(b0))$. Siden b var vilkårlig valgt kan man konkludere $G[H](0) \wedge J[H_2](0) \wedge \forall y((G_2[H](y) \wedge J_2[H_2](y) \supset H_3(f(y, 0)))$. Dermed er det vist at $H(0)$.

Bevis for $F(x) \supset F(Sx)$

Anta $F(a)$. Ved antakelsen om at F er induktiv er $F(Sa)$. Fra definisjonen av F følger det at $G[H](Sa)$ og $J[H_2](Sa)$. Anta $G_2[H](b)$ og $J_2[H_2](b)$. Da følger det at $G_3[H](g(bSa))$ og $J_3[H_2](j(bSa))$ som i beviset over. Videre følger $H(g(bSa))$ og $H_2(j(bSa))$ fra lemma 5.6.

Dermed får man $H_3(h(g(bSa), j(bSa)))$.

Ved substitusjon får man $H_2(f(Sab))$. Siden b var vilkårlig kan det konkluderes at $\forall y(y : G_2[H] \wedge y : J_2[H_3]) \supset H_3(f(y, Sa))$. Dermed er $F(Sa)$. Siden a også var vilkårlig kan det konkluderes at $\forall F(x) \supset F(Sx)$.

$F(x) \overset{\text{ved def.}}{\Rightarrow} G[H](x) \overset{\text{lemmma 5.6}}{\Rightarrow} H(x) \overset{\text{ved ant.}}{\Rightarrow} N(x)$

5.3 Et hierarki av induktive predikater

Som nevnt ligger definisjonen av et induktivt predikat for en funksjon f tett opp til definisjonen av f. Jeg undersøker her om man kan definere predikater etter et mønster som ligner på definisjonsskjemaene for primitivt rekursive funksjoner. Sekvensen etterfølger, addisjon, multiplikasjon og eksponensiering utgjør et naturlig hierarki av monotont økende funksjoner og de er alle elementære. Addisjon er iterasjon over etterfølger, multiplikasjonsfunksjonen oppnås ved ubegrenset rekursjon over + og så videre.

Siden multiplikasjon er et rekursjonsnivå over addisjon er det naturlig å tenke seg at et predikat for multiplikasjon er et nøstingsnivå opp med hensyn på kvantorer. Da kan predikatet A brukes som en referanse til eller en forkortelse for uttrykket $N(x) \wedge \forall y(N(y) \supset N(+yx))$:

$$M(x) = A(x) \wedge \forall y(A(y) \supset A(\times yx))$$

Merk at alle predikater P har en normalform P_n, der definisjonen ikke inneholder noen andre predikater enn N. Normalformen til M oppnås ved å ekspandere A og

$$\begin{aligned} M_{\text{nf}}(x) =&((N(x) \wedge \forall y(N(y) \supset N(+yx))) \wedge \forall z((N(z) \wedge \forall w(N(w) \supset N(+wz))) \\ &\supset (N(+zx) \wedge \forall u(N(u) \supset N(+u + zx))))) \end{aligned}$$

Definisjon 5.9 *La H være et induktivt predikat for h:*

$$H(x) = H_1(x) \wedge \forall y(H_2(y) \supset H_3(h(y, x)))$$

H kalles ***flatt*** *hvis* $H_1 = H_2 = H_3$.

Når f er definert ved primitiv rekursjon over en induktiv funksjon h, hvor h er assosiativ og H er et *flatt* predikat for h, så kan det konstrueres et induktivt predikat for f.

Assosiativitet av addisjonsfunksjonen $(\forall xyz(+(+xy)z = +x(+yz)))$ og multiplikasjonsfunksjonen $(\forall xyz(\times(\times xy)z = \times x(\times yz)))$ er ikke utledbart i PRA. Det er et kjent resultat at disse egenskapene ikke er utledbare i Robinson-aritmetikk, R, [13]. Måten å vise dette på er å konstruere en modell for R hvor assosiativitet for addisjons- og multiplikasjonsfunksjonen ikke er gyldig. PRA er ikke prinsipielt sterkere enn R, så et tilsvarende argument kan gis for PRA.

Ved å utvide PRA med ikkelogiske slutningsregler for assosiativitet kan man vise at induktive funksjoner som rekurserer over assosiative funksjoner er induktive. I tråd med notasjonen i kapittel 3 kalles en ikkelogisk regel om assosiativitet av en funksjon h for Ass_h. PRA $+Reg$ benevner PRA utvidet med regelen Reg.

Påstand 5.10 *La f være definert ved primitiv rekursjon over g og h:*

$$f(y,0) = g(y)$$
$$f(y,Sx) = h(y, f(y,x))$$

La g være en av initialfunksjonene $\mathcal{O}, \mathcal{S}$ eller $\mathcal{I}_i^n$. La h være assosiativ og la bevissystemet inneholde følgende regel:

$$\frac{h(h(a,b),c) = h(a,h(b,c)), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ Ass_h$$

La H være et induktivt predikat for h

$$H(x) = G(x) \wedge \forall y(G(y) \supset G(h(x,y))$$

Da er F induktiv i PRA $+Ass_h$.

Bevis. Anta h er assosiativ. Definer et predikat for f ved: $F(x) = H(x) \wedge \forall y(H(y) \supset H(h(y,x)))$.

For å vise at $F(0)$ er det nødvendig å vise at $H(0) \wedge \forall y(H(y) \supset H(f(y,0)))$. Siden H er induktiv følger at $H(0)$.

Anta $H(b)$. Ved definisjonen av f er $f(b,0) = g(b)$. Tre tilfeller må sjekkes: $g = \mathcal{O}, g = \mathcal{S}$ eller $g = \mathcal{I}_i^n$. Fra $\mathcal{O}(b) = 0$ og $N(0)$ følger at$H(\mathcal{O}(b))$. $H(\mathcal{S}(b))$ siden $H(b)$ og H er induktiv. $H(\mathcal{I}_1^1(b))$ siden $\mathcal{I}_1^1(b) = b$.

Siden b var vilkårlig valgt kan det konkluderes at $\forall y(H(y) \supset H(f(y,0)))$ som konkluderer beviset for at $F(0)$

Det må også vises at $F(x) \supset F(Sx)$. Det vil si $(H(x) \wedge \forall y(H(y) \supset H(f(y,x)))) \supset (H(Sx) \supset \forall z(H(z) \supset H(f(z,Sx))))$.

Anta $F(a)$ og $H(b)$. Dette gir $H(f(b,a))$. Funksjonen f er definert ved $f(b,Sa) = h(b,f(b,a))$. Man må vise

$$G(h(b,f(b,a))) \wedge \forall z(G(z) \supset G(h(z,h(b,f(b,a))))).$$

Fra $H(f(b,a))$ og $G(b)$ følger $G(h(b,f(b,a)))$.

Anta $G(c)$. Fra antakelsen om at $H(b)$ følger $G(h(c,b))$. Fra dette og $H(f(b,a))$ følger $G(h(h(c,b),f(b,a)))$. Fra antakelsen om at h er assosiativ følger at $h(h(c,b),f(b,a)) = h(c,h(b,f(b,a)))$. Ved en anvendelse av regelen *Repl* følger $G(h(c,h(b,f(b,a))))$ som er det ønskede resultat.

Ved å velge $F(x) = H(x) \wedge \forall y(H(y) \supset H(h(y,x)))$ klarer man altså å vise $F(x) \supset F(Sx)$.

Til slutt må det vises at $F(x) \supset N(x)$. Dette følger fra definisjonen av F og antakelsen om at H er induktiv.

Hvis f er definert ved primitiv rekursjon over g og h og h ikke er en av de primitivt rekursive initialfunksjonene eller er assosiativ, så går det trolig ikke å utlede at f er induktiv. Man kan følge den samme gangen som i beviset for 5.10 til det punktet hvor det må vises $h(z,h(y,f(y,x))) : G$. For å få til dette må man ha $h(y,f(y,x)) : H$. I så fall må $f(y,x) : H[H]$ det vil si $F_3 = H[H]$. Men da får man $F_3 = H[H]$ og $F_3 = H$ og så videre. Det ser ikke ut til å la seg ikke gjøre. I avsnitt 5.4 viser jeg at modifisert subtraksjon, som rekurserer over en ikke-assosiativ funksjon, ikke er induktiv i PRA. Hvorvidt noe liknende holder i det generelle tilfellet vises ikke her.

5.3.1 Lukningsegenskaper med assosiativitet

Beviset for påstand 5.10 gir en algoritme for å konstruere et induktivt predikat for en funksjon f, når f er definert ved rekursjon over g og h, og h er assosiativ. For å kunne konstruere predikater for multiplikasjon og eksponensiering legges aksiomer om assosiativitet av addisjon og multiplikasjon til PRA:

$$+(+ab)c = +a(+bc) \tag{5.18}$$

$$\times(\times ab)c = \times a(\times bc) \tag{5.19}$$

Aksiomene gjøres om til ikkelogiske regler i tråd med framgangsmåten definert i kapittel 3. Også disse reglene tilfredsstiller lukningsbetingelsen 3.2.

$$\frac{+(+ab)c = +a(+bc), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ AssAdd \qquad \frac{\times(\times ab)c = \times a(\times bc), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ AssTimes$$

Korrolar 5.11 *[Korrolar til påstand 5.10] Funksjonen $\times$ er induktiv i PRA-+AddAss,TimesAss.*

Bevis. Korrolar 5.11 følger fra aksiom 5.18 og påstand 5.1 og 5.10. □

For å finne induktive predikater for bundet sum $\Sigma_{i\leq y} f(\vec{x}, i)$ og bundet produkt $\Pi_{i\leq y} f(\vec{x}, i)$ kan man bruke en kombinasjon av komposisjon over induktive predikater og prosedyren angitt i beviset for påstand 5.10.

Teorem 5.12 *Bundet sum og bundet produkt er induktive i PRA+AddAss,-TimesAss.*

Bevis. La G være et induktivt predikat:

$$G(x) = J(x) \wedge \forall J(y) \supset J(g(y, x))$$

Da er følgende induktivt:

$$Su(x) = G[A](x) \wedge \forall J[A](y) \supset A(\sum_{i\leq x} g(y, i))$$
$$Pr(x) = G[M](x) \wedge \forall J[M](y) \supset M(\prod_{i\leq x} g(y, i))$$

Utledninger av induktive predikater for fakultet- og eksponensialfunksjonen, som er definert ved hjelp av bundet produkt, er gitt i tillegg A. Disse kan enkelt utvides til generelle beviser for at de induktive predikatene er lukket under skjemaet for bundet produkt. Et bevis for at de induktive predikatene er lukket under skjemaet for bundet sum blir tilsvarende.

Jeg må også verifisere at $N(x)$ følger når henholdsvis $Su(x)$ og $Pr(x)$:

$$Su(x) \overset{\text{ved def. av } Su}{\Rightarrow} G[A](x) \overset{\text{lemma 5.6}}{\Rightarrow} A(x) \overset{\text{ved def. av A.}}{\Rightarrow} N(x).$$

Beviset for Pr er tilsvarende. □

Nå har jeg vist at de Kalmárelementære initialfunksjonene unntatt modifisert subtraksjon er elementære. Videre har jeg vist at de induktive predikatene er lukket under komposisjon, bundet sum og bundet produkt. Dermed kan jeg konkludere:

Teorem 5.13 *Elementære strengt voksende funksjoner er induktive i PRA-+AddAss,TimesAss.*

Bevis. Teoremet følger fra lemma 5.1 og 5.2 og teorem 5.8 og 5.12. □

Korrolar 5.14 *Eksponensialfunksjonen er induktiv i PRA+AddAss,TimesAss.*

Bevis. Eksponensialfunksjonen er elementær, så korrolar 5.14 følger fra teorem 5.13. □

Eksponensialfunksjonen *exp* kan defineres ved hjelp av bundet produkt:

$$exp(y,x) = \Pi_{i\leq x}\mathcal{I}_2^2(y,i)$$

Predikatet for *exp* blir da:

$$E(x) = A(x) \wedge \forall y(A(y) \supset A(exp(y,x)))$$

Ved hjelp av predikatet til produkt kan man også konstruere et predikat for bundet minimalisering

$$\mu i \leq y[p(i,x_1,\ldots,x_n)] = \sum_{z\leq y}\prod_{i\leq z} cosg(p(i,x_1,\ldots,x_n)).$$

La g være en primitiv rekursiv funksjon med predikat $G(x) = J(x)\wedge\forall y(J(y) \supset J(g(y,x)))$. Da er følgende induktivt:

$$My(x) = Pr(x) \wedge \forall y(J[M](y) \supset A(\sum_{z\leq x}\prod_{i\leq z} g(y,i)))$$

5.4 Modifisert subtraksjon er ikke induktiv

Modifisert subtraksjon er definert ved primitiv rekursjon over forgjengerfunksjonen:

$$\dot{-}0y = y$$
$$\dot{-}Sxy = P(\dot{-}xy)$$

Forgjengerfunksjonen er ikke assosiativ. Resultatet i avsnitt 5.3 indikerer at modifisert subtraksjon ikke er induktiv i PRA, noe jeg viser i dette avsnittet.

Påstand 5.15 *La* $\mathrm{MS}(x) = N(x) \wedge \forall y(N(y) \supset N(\dot{-}yx))$ *være et predikat for modifisert subtraksjon. Da er sekventen*

$$(\ddagger) \qquad \Rightarrow \mathrm{MS}(0) \wedge \forall x(\mathrm{MS}(x) \supset \mathrm{MS}(Sx))$$

ikke utledbar i PRA.

Beviset for påstand 5.15 er todelt. Først konstruerer jeg en ikkestandardmodell for PRA hvor ($\ddagger$) ikke er sant. Denne måten å definere ikkestandardmodeller på brukes av Jeffrey [13]. Deretter bruker jeg kompletthetsteoremet til å vise at ($\ddagger$) ikke kan være utledbart i PRA.

Bevis. La N' være N utvidet med elementene a, b, c.

La $N(a), N(b), N(\dot{-}ba), \neg N(c)$. La komposisjonen av forgjenger og modifisert subtraksjon ha følgende tolkning i N':.

	n	a	b	c
m	pred(m min n)	pred(m min a)	pred(m min b)	c
a	pred(a min n)	pred(a min a)	pred(a min b)	c
b	pred(b min n)	c	pred(b min b)	c
c	c	c	c	c

De nye elementenes egenskaper er ikke i konflikt med de ikkelogiske reglene i PRA, så N' er en modell for PRA. I N' er $N(a), N(b)$ og $N(\dot{-}ba)$ sann men ikke $S(\dot{-}ba) = P(\dot{-}ba) = N(c)$. Det vil si uttrykket $\exists x(\mathrm{MS}(x) \wedge \neg \mathrm{MS}Sx)$ er sant i N'. Dette er negasjonen av $\forall x(\mathrm{MS}(x) \supset \mathrm{MS}(Sx))$, men da kan ikke (‡) være sant i N'.

Anta at (‡) er utledbart i PRA. Ved kompletthetsteoremet for første ordens teorier 2.45 må (‡) være sant i alle modeller for PRA. Men jeg har nettopp vist at det finnes en modell for PRA hvor (‡) ikke er sant. Antakelsen om at (‡) er utledbart i PRA leder til en selvmotsigelse og må derfor være feil. □

Jervell og Zhang [15] gir et argument for at en funksjon er induktiv hviss den er elementær. Klarer man å vise denne påstanden begge veier får man en bevisteoretisk beskrivelse av de elementære funksjonene. Dette er interessant fordi en da har to alternative karakteristikker av en klasse av funksjoner ved hjelp av ulike formelle modeller. Jeg viser her at det ikke går å gi en slik avgrensing ved hjelp av de tilleggsaksiomene som Jervell og Zhang foreslår for å vise at funksjonen $\dot{-}$ er induktiv:

$$\dot{-}0y = 0$$
$$\dot{-}Sxy = 0 \vee \dot{-}Sxy = S\dot{-}xy$$

Problemet er at de ved å legge til disse aksiomene får med alt for mange funksjoner. Jeg viser at det finnes en ikke-elementær funksjon f og utledninger med konstant lengde av $N(f(m))$ for alle tall $m \in N$ i PRA utvidet med reglene:

$$\frac{\Gamma \Rightarrow \Delta, \dot{-}0b = 0}{\Gamma \Rightarrow \Delta}\ Sub0 \qquad \frac{\Gamma \Rightarrow \Delta, \dot{-}Sab = 0 \quad \Gamma \Rightarrow \Delta, \dot{-}Sab = S(\dot{-}ab)}{\Gamma \Rightarrow \Delta}\ Sub\vee$$

I beviset gjør jeg bruk av følgende definisjon og teoremer:

Definisjon 5.16 *En funksjon f er en universalfunksjon for en klasse av funksjoner $\mathcal{H}$ dersom det for enhver unær funksjon g finnes et fast tall m slik at $f(m, x) = g(x)$.*

Teorem 5.17 *[17, s. 37] La $\mathcal{H}$ være en subrekursiv funksjonsklasse. Da inneholder ikke $\mathcal{H}$ en universalfunksjon for $\mathcal{H}$.*

Teorem 5.18 *[17, s. 40] La $i \geq 3$. Det finnes en universalfunksjon p for $\mathcal{E}^i$ slik at $p \in \mathcal{E}^{i+1}$.*

Fra disse teoremene kan jeg vise hovedresultatet i dette avsnittet:

Påstand 5.19 *Det finnes en funksjon f slik at $f \notin \mathcal{E}^3$ og for alle $m \in N$ finnes en utledning $\mathcal{D} \vdash_k^{\mathrm{PRA}+sub} \Rightarrow N(f(m))$, for et fast tall k.*

Bevis. Teorem 5.18 gir at det finnes universalfunksjoner for alle subrekursive funksjonsklasser fra de elementære og oppover. Teorem 5.17 gir at en universalfunksjon for en subrekursiv funksjonsklasse $\mathcal{H}$ ikke er inneholdt i klassen selv. La p være en universalfunksjon for de elementære funksjonene. Definer f ved

$$f(x) = not(p(x, x))$$

Jeg viser ved et standard diagonaliseringsbevis at f ikke kan være i $\mathcal{E}^3$. Anta for en selvmotsigelse at $f \in \mathcal{E}^3$. Ved definisjonen av universalfunksjon må det finnes en fast m slik at $f(x) = p(m, x)$. Dermed følger

$$f(m) = not(p(m, m)) \text{ (ved def. av } f) \neq p(m, m) = f(m)$$

Antakelsen om at $f \in \mathcal{E}^3$ leder til en selvmotsigelse, så f kan ikke være i $\mathcal{E}^3$.

Figurer 5.2 og 5.3 gir en strategi for å utlede $\Rightarrow N(f(m))$ med høyde begrenset av et fast tall k for vilkårlige tall m, n. Den lengste greina i utledningen i figuren krever 14 skritt. Dette avslutter beviset for påstand 5.19. □

Utledningen av $\Rightarrow N(f(m))$ i beviset for påstand 5.19 gjør bruk av reglene *Sub0* og *Sub*$\vee$. Det kan derfor ikke være elementært å eliminere disse reglene fra systemet. Problemet er at man med disse reglene kan vise at $\dot{-}(1, p) = 0 \vee \dot{-}(1, p) = 1$. Siden $not(p)$ er definert som $\dot{-}(1, p)$ kan man for alle funksjoner f komponert over not gi en kort utledning av at $N(f(m))$. Dette er opplagt, siden f er en $\{0, 1\}$-funksjon, men det gir oss ikke noe informasjon.

Merk at liknende funksjoner kan konstrueres for alle de subrekursive funksjonsklassene fra de elementære og oppover. Dermed finnes det uendelig mange ikke-elementære funksjoner hvor en kan konstruere utledninger med konstant lengde for at de er av type N, ved hjelp av reglene *Sub0* og *Sub*$\vee$. Å legge disse reglene til systemet fører til at man får med flere induktive funksjoner enn det som er ønskelig. Eksemplet med modifisert subtraksjon

illustrerer at det ikke er likegyldig hvilke regler man legger til et bevissystem, man kan risikere at det blir for uttrykkskraftig.

Korrolar 3.13 gir at for enhver funksjon $f \in \mathcal{E}^r$ finnes en utledning i PRA av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ med høyde begrenset av en funksjon i samme Gregorczyk-klasse. Funksjonen $\dot{-} \in \mathcal{E}^0$. Teorem 2.21 gir at alle funksjoner f i $\mathcal{E}^0$ majoriseres av en funksjon som er lineær i et av argumentene til f. Dermed følger det at man for alle $m, n \in N$ kan finne utledninger av $N(\dot{-}(m,n))$ med høyde lineær i m, n. Av samme grunn kan man finne utledninger med lineær høyde for at de logiske operatorene er totale. Problemet er at siden det ikke går å konstruere et induktivt predikat for modifisert subtraksjon så følger det ikke at funksjoner definert ved komposisjon over modifisert subtraksjon og eventuelt andre elementære funksjoner er induktive. Dermed kan man ikke slutte at for enhver elementær funksjonen f så kan man utlede $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ med høyde lineær i argumentene til f.

5.4.1 Ingen avgrensing

På grunn av problemene med modifisert subtraksjon har jeg valgt å vise et svakere resultatet enn det Jervell og Zhang [15] ønsker å vise. I dette kapitlet har jeg vist at de strengt voksende elementære funksjonene er induktive i en utvidet versjon av bevissystemet PRA, definert i kapittel 3. I kapittel 8 viser jeg at hvis en funksjon er induktiv i PRA$+AddAss, TimesAss$ så er den elementær. Beviset går ut på å konstruere en elementær algoritme for å beregne $f(m_1, \ldots, m_n)$ fra en utledning $\mathcal{D}$ av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, når høyden av $\mathcal{D}$ er lineær i $m_1, \ldots, m_n$. Jeg har dermed en bevisteoretisk beskrivelse av en subrekursiv funksjonklasse og en subrekursiv beskrivelse av et bevissystem som ikke "treffer" hverandre. Det er ikke mulig å vise at de induktive funksjonene er elementære og strengt voksende for å oppnå en reell avgrensning av en funksjonklasse. Algoritmen som beregner $f(m_1, \ldots, m_n)$ fra en utledning av $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$ når f er induktiv må bruke $\{0,1\}$-funksjoner som er definert ved hjelp av modifisert subtraksjon.

Ved å utelate modifisert subtraksjon går en glipp av en rekke viktige funksjoner. I kapittel 2 viste jeg at modifisert subtraksjon brukes til å definere de karakteristiske funksjonene for de logiske operatorene og i delingsfunksjoner. Man mister dermed søkefunksjoner som defineres ved komposisjon over begrenset søk og $\{0,1\}$-funksjoner.

5.5 Induksjon oppover

Det er naturlig å lese en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ i PRA fra rota og oppover. I de neste kapitlene benytter jeg meg av egenskaper ved de ikkelogiske reglene i PRA$+AddAss, TimesAss$ til å gjøre induksjon over høyden av beviser oppover. Det innebærer at i induksjonstilfellet betraktes den øverste regelen i treet, istedenfor den nederste som ofte er vanlig. Formelt sett er en utledning lukket kun hvis den har et logisk aksiom på formen $A, \Gamma \Rightarrow \Delta, A$ i alle toppnodene. Men på grunn av egenskaper ved de ikkelogiske reglene kan man observere følgende:

Observasjon 5.20 *La $\mathcal{D}$ være en snittfri utledning i PRA$+AddAss, TimesAss$ av et atomært utsagn med tom Γ i konklusjonen. Da er alle formler $C_1, \ldots, C_n$ som forekommer på venstre side av sekventpila utledbare i PRA$+AddAss$,-$TimesAss$. For alle utledninger $\mathcal{D}_1 \vdash\Rightarrow C_1, \ldots, \mathcal{D}_n \vdash\Rightarrow C_n$ gjelder at $|\mathcal{D}_i| \leq |\mathcal{D}|$.*

Ved delformelegenskapen vist i teorem 3.6 er alle regler i en snittfri utledning av et atomært utsagn ikkelogiske når Γ er tom i konklusjonen. Ingen av de ikkelogiske reglene i PRA$+AddAss, TimesAss$ leder til forgreninger av bevistreet. Dette er en nødvendig forutsetning for at observasjonen over skal holde. Hadde PRA$+AddAss, TimesAss$ blitt utvidet med regler som fører til forgrening av den typen Jervell og Zhang gir for modifisert subtraksjon i forrige avsnitt, kunne jeg ikke gjort induksjon over bevistrærne nedenfra og opp. De ikkelogiske reglene endrer ikke på suksedenten i en sekvent. Dermed kan man bytte ut formlene i suksedenten i en utledning uten at dette påvirker reglene som er brukt. Ved å bytte ut formelen i suksedenten med en formel som finnes i antecedenten i toppnoden får man en utledning av denne.

$$\frac{}{N(f(m,n)), S0 = f(m,n), N(S0), N(0) \Rightarrow N(f(m,n))}\ ax$$

$$\frac{N(f(m,n)), S0 = f(m,n), N(S0), N(0) \Rightarrow N(f(m,n))}{S0 = f(m,n), N(S0), N(0) \Rightarrow N(f(m,n))}\ Repl$$

$$\frac{S0 = f(m,n), N(S0), N(0) \Rightarrow N(f(m,n))}{N(0), S0 = f(m,n) \Rightarrow N(f(m,n))}\ RN$$

$$\frac{N(0), S0 = f(m,n) \Rightarrow N(f(m,n))}{S0 = f(m,n), S(\dot{-}(0,p(m,n))) = S0, S(\dot{-}(0,p(m,n))) = f(m,n) \Rightarrow N(f(m,n))}\ NN, LW$$

$$\frac{S0 = f(m,n), S(\dot{-}(0,p(m,n))) = S0, S(\dot{-}(0,p(m,n))) = f(m,n) \Rightarrow N(f(m,n))}{S(\dot{-}(0,p(m,n))) = S0, S(\dot{-}(0,p(m,n))) = f(m,n), \dot{-}(0,p(m,n)) = 0 \Rightarrow N(f(m,n))}\ Trans, LW$$

$$\frac{S(\dot{-}(0,p(m,n))) = S0, S(\dot{-}(0,p(m,n))) = f(m,n), \dot{-}(0,p(m,n)) = 0 \Rightarrow N(f(m,n))}{\dot{-}(0,p(m,n)) = 0, S(\dot{-}(0,p(m,n))) = f(m,n) \Rightarrow N(f(m,n))}\ EqS$$

$$\frac{\dot{-}(0,p(m,n)) = 0, S(\dot{-}(0,p(m,n))) = f(m,n) \Rightarrow N(f(m,n))}{S(\dot{-}(0,p(m,n))) = f(m,n), \dot{-}(S0,p(m,n)) = S(\dot{-}(0,p(m,n))), \dot{-}(S0,p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}\ Sub0, LW$$

$$\frac{S(\dot{-}(0,p(m,n))) = f(m,n), \dot{-}(S0,p(m,n)) = S(\dot{-}(0,p(m,n))), \dot{-}(S0,p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}{\dot{-}(S0,p(m,n)) = S(\dot{-}(0,p(m,n))), \dot{-}(S0,p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}\ Trans$$

$$\overset{5.3}{\dot{-}(S0,p(m,n)) = 0, \dot{-}(S0,p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}$$

$$\frac{\dot{-}(S0,p(m,n)) = 0, \dot{-}(S0,p(m,n)) = f(m,n) \Rightarrow N(f(m,n)) \qquad \dot{-}(S0,p(m,n)) = S(\dot{-}(0,p(m,n))), \dot{-}(S0,p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}{\dot{-}(S0,p(m,n)) = f(m,n), not(p(m,n)) = \dot{-}(S0,p(m,n)), not(p(m,n)) = f(m,n), f(m,n) = not(p(m,n)) \Rightarrow N(f(m,n))}\ Sub\vee, LW$$

$$\frac{\dot{-}(S0,p(m,n)) = f(m,n), not(p(m,n)) = \dot{-}(S0,p(m,n)), not(p(m,n)) = f(m,n), f(m,n) = not(p(m,n)) \Rightarrow N(f(m,n))}{not(p(m,n)) = \dot{-}(S0,p(m,n)), not(p(m,n)) = f(m,n), f(m,n) = not(p(m,n)) \Rightarrow N(f(m,n))}\ Trans$$

$$\frac{not(p(m,n)) = \dot{-}(S0,p(m,n)), not(p(m,n)) = f(m,n), f(m,n) = not(p(m,n)) \Rightarrow N(f(m,n))}{f(m,n) = not(p(m,n)), not(p(m,n)) = \dot{-}(S0,p(m,n)) \Rightarrow N(f(m,n))}\ Sym$$

$$\frac{f(m,n) = not(p(m,n)), not(p(m,n)) = \dot{-}(S0,p(m,n)) \Rightarrow N(f(m,n))}{f(m,n) = not(p(m,n)) \Rightarrow N(f(m,n))}\ not$$

$$\frac{f(m,n) = not(p(m,n)) \Rightarrow N(f(m,n))}{\Rightarrow N(f(m,n))}\ fcomp$$

Figur 5.2: Kort utledning av $N(f(m,n))$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{N(f(m,n)), 0 = f(m,n), N(0), \dot{-}(S0, p(m,n)) = 0, \dot{-}(S0, p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}\, ax}{0 = f(m,n), N(0), \dot{-}(S0, p(m,n)) = 0, \dot{-}(S0, p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}\, Repl}{0 = f(m,n), \dot{-}(S0, p(m,n)) = 0, \dot{-}(S0, p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}\, NN}{\dot{-}(S0, p(m,n)) = 0, \dot{-}(S0, p(m,n)) = f(m,n) \Rightarrow N(f(m,n))}\, Trans$$

Figur 5.3: Kort utledning av $N(f(m,n))$: venstre side

Kapittel 6

Forskjellen på direkte og indirekte argument

Her viser jeg forskjellen i høyden på utledninger av $\Rightarrow N(fak(m))$ henholdsvis med og uten snitt. For å illustrere hvor raskt direkte utledninger om aritmetiske uttrykk vokser i PRA, viser jeg framgangsmåter for å finne utledninger av termene: $N(+\bar{m}\bar{n}), N(*\bar{m}\bar{n})$ og $N(fak(\bar{m}))$, for vilkårlig m, n. Høyden på utledningene i disse eksemplene er avhengig av at det brukes en tallteori med unær notasjon. Man kan spare inn på høyden ved å bruke binær notasjon. Men prinsippet blir det samme. Poenget her er å vise hvor mye man kan spare på høyden i et system som PRA ved hjelp av snitt.

Hvis man følger framgangsmåtene vist i figurer 6.1, 6.2, 6.3, 6.4, 6.5 og 6.6 bestemmes antall skritt i utledningene av følgende funksjoner.

Addisjon:	$\sigma(m,n) = 5m + n + 4$
Multiplikasjon:	$\gamma(m,n) = (\sigma(m,n) + a)m + mn + b$
Fakultet:	$\delta(m) = \sum_{i=1}^{m-1} \gamma(m-i, \prod_{j=0}^{i-1}(m-j)) + fak(m) + am + b$

Siden fakultet er sammensatt av addisjon og multiplikasjon, inkluderer utledninger for fakultet utledninger for disse termene som delutledninger. Alle utledningene består av flere repetisjoner av instansieringer av de samme reglene. Jeg skriver derfor kun opp strukturen til de delutledningene som blir repetert.

Disse algoritmene for bevissøk er ikke spesielt effektive. Mens $fak(60) \approx 8,3{*}10^{82}$ gir funksjonen som beregner høyden på utledningen et tall i overkant av 10^{160} for $m = 60$, når man setter inn verdiene $a = 15$ og $b = 3$. Det er imidlertid ikke mulig å finne en algoritme for bevissøk som er prinsipielt mer effektiv. En utledning av et uttrykk $f(\bar{m}_1, \ldots, \bar{m}_n)$ uten snitt tilsvarer et direkte argument for uttrykket. Utledningen inneholder en utregning av

funksjonsuttrykket. Ved påstand 6.1 har utledningen minst like stor høyde som tallverdien til uttrykket. Derfor må utledningen av $N(fak(60))$ inneholde flere trinn enn 10^{80}, som er mer enn det vanlige anslaget på antall atomer i universet.

Påstand 6.1 *En snittfri utledning $\mathcal{D}$ av $\Rightarrow N(\bar{t})$ har høyde større eller lik t.*

Bevis. Påstanden vises ved induksjon over høyden av utledningen. Ved observasjon 5.20 kan jeg gjøre induksjonen oppover. Det innebærer at i induksjonstilfellet betraktes den øverste regelen i treet.

Basistilfelle: $|\mathcal{D}| = 1$. Da må $\bar{t} = \mathcal{O}$ og $\mathcal{D}$ er:

$$\dfrac{\dfrac{}{N(\mathcal{O}) \Rightarrow N(\mathcal{O})}\, ax}{\Rightarrow N(\mathcal{O})}\, NN$$

$|\mathcal{D}| \geq 0$.

Induksjonshypotese: Påstanden holder for utledninger med høyde mindre enn n.

Induksjonssteg: Må vise at påstanden holder for utledninger med høyde n. En utledning av $\Rightarrow N(\bar{t})$ må ha $N(\bar{t})$ som hovedformel i antecedenten i toppnoden. Se på den øverste regelen $\mathcal{R}$ i utledningen:

$$\dfrac{\dfrac{}{N(\bar{t}), \Gamma \Rightarrow \Delta}\, ax}{\begin{array}{c}\Gamma \Rightarrow \Delta \\ \mathcal{D}_0\end{array}}\, \mathcal{R}$$

De eneste reglene hvor $N(\bar{t})$ er hovedformel i antecedenten er RN og *Repl*. Hvis $\mathcal{R}$ er RN må $\bar{t}$ være på formen $S(\bar{k})$.

$$\dfrac{\dfrac{}{N(\mathcal{S}(\bar{k})), N(\bar{k}), \Gamma \Rightarrow \Delta}\, ax}{\begin{array}{c}N(\bar{k}), \Gamma \Rightarrow \Delta \\ \mathcal{D}_0\end{array}}\, RN$$

$|\mathcal{D}| = |\mathcal{D}_0| + 1$. Ved induksjonshypotesen er høyden av $\mathcal{D}_0$ større eller lik k. Siden $\bar{t} = \mathcal{S}(\bar{k})$ er $t = Sk$ ved kompletthetsteoremet for første ordens teorier (teorem 2.45) som er $k + 1$ ved definisjonen av $\mathcal{S}$. Til sammen gir dette $|\mathcal{D}| = |\mathcal{D}_0| + 1 \geq k + 1 = t$.

La $\mathcal{R}$ være *Repl* med $N(\bar{t})$ som hovedformel.

$$\dfrac{\dfrac{}{N(\bar{t}), \bar{k} = \bar{t}, N(\bar{k}), \Gamma \Rightarrow \Delta}\, ax}{\begin{array}{c}\bar{k} = \bar{t}, N(\bar{k}), \Gamma \Rightarrow \Delta \\ \mathcal{D}_0\end{array}}\, Repl$$

$|\mathcal{D}| = |\mathcal{D}_0|+1$. Ved induksjonshypotesen er høyden av $\mathcal{D}_0$ større eller lik k. Siden $\bar{t} = \bar{k}$ er $t = k$ ved kompletthetsteoremet for første ordens teorier (teorem 2.45). Til sammen gir dette $|\mathcal{D}| = |\mathcal{D}_0|+1 \geq k$ (ved IH) $= t$. □

6.1 Et indirekte argument med snitt

I forrige kapittel definerte jeg det nødvendige verktøyet for å konstruere et indirekte argument for $\Rightarrow N(\mathit{fakm})$. Fakultetsfunksjonen kan defineres ved hjelp av komposisjon og begrenset rekursjon:

$$\begin{aligned} SP(a) &= S(P(a)) \\ \mathit{fak}(0) &= SP(0) \\ \mathit{fak}(Sa) &= \times(\mathit{fak}(a), SP(Sa)) \end{aligned}$$

Dette er ikke den vanlige definisjonen av fakultet. Fordelen med den er at den følger definisjonsskjemaene for elementære funksjoner. Dermed kan predikatet for fakultet konstrueres etter framgangsmåten beskrevet i forrige kapittel:

$$F(x) = Su[M](x) \wedge M(\mathit{fak}(x)),$$

se figur 6.7 og 6.8.

Ved å bruke det induktive predikatet for fakultet som snittformel, får man en utledning av $N(\mathit{fak}(\bar{m}))$ med høyde lineær i m. Se figurer 6.7 og 6.8.

Merk at høyden på en utledning regnes som høyden på den lengste greina i bevistreet ved definisjon 2.30. Høyden på utledningen av $F(0) \wedge \forall x(F(x) \supset F(Sx))$ er konstant. Høyregreina i bevistreet består hovedsaklig av m instansieringer av $L\forall-$ og $L\supset$-reglene samt tre instansieringer av $L\wedge$-regelen. Dermed kan høyden av utledningen beregnes av funksjonen

$$\sigma(m) = \max(27, 2m+5) = \begin{cases} 27 \text{ hvis } m \leq 11 \\ 2m+5 \text{ ellers.} \end{cases}$$

en lineær funksjon. Mens den snittfrie utledningen av $N(\mathit{fak}(60))$ inneholdt flere antall trinn enn det vanlige anslaget på antall atomer i universet, krever en utledning for den samme påstanden med snitt høyst 125 skritt. En utledning som får plass på noen få sider.

Teorem 6.2 *La f være en n-ær induktiv funksjon. For alle $m_1, \ldots, m_n \in N$ eksisterer det en utledning $\mathcal{D} \in$ PRA, som kan inneholde snitt, og en funksjon σ slik at*

$$\mathcal{D} \vdash^{\mathrm{PRA}}_{\sigma(m_1,\ldots,m_n)} \Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$$

og σ er lineær i $m_1, \ldots, m_n$.

Bevis. La f beskrives av predikatet

$$F(x_1) = F_1(x_1) \wedge \forall x_2, \ldots, x_n(F_2(x_2) \wedge \cdots \wedge F_n(x_n) \supset F_{n+1}(f(x_1, \ldots, x_n)))$$

Ved å bruke predikatene $F, F_1, \ldots, F_n$ i snittformler får man en utledning av $N(f(m_1, \ldots, m_n))$ med høyde lineær i $m_1, \ldots, m_n$ på samme måte som for fakultet. Se figurer 6.9, 6.10 og 6.11. Høyden av utledningen kan beregnes av funksjonen $\sigma(m_1, \ldots, m_n) = max(k, k_0, \ldots, k_{n-1}, (m_1 + \cdots + m_n + c))$. □

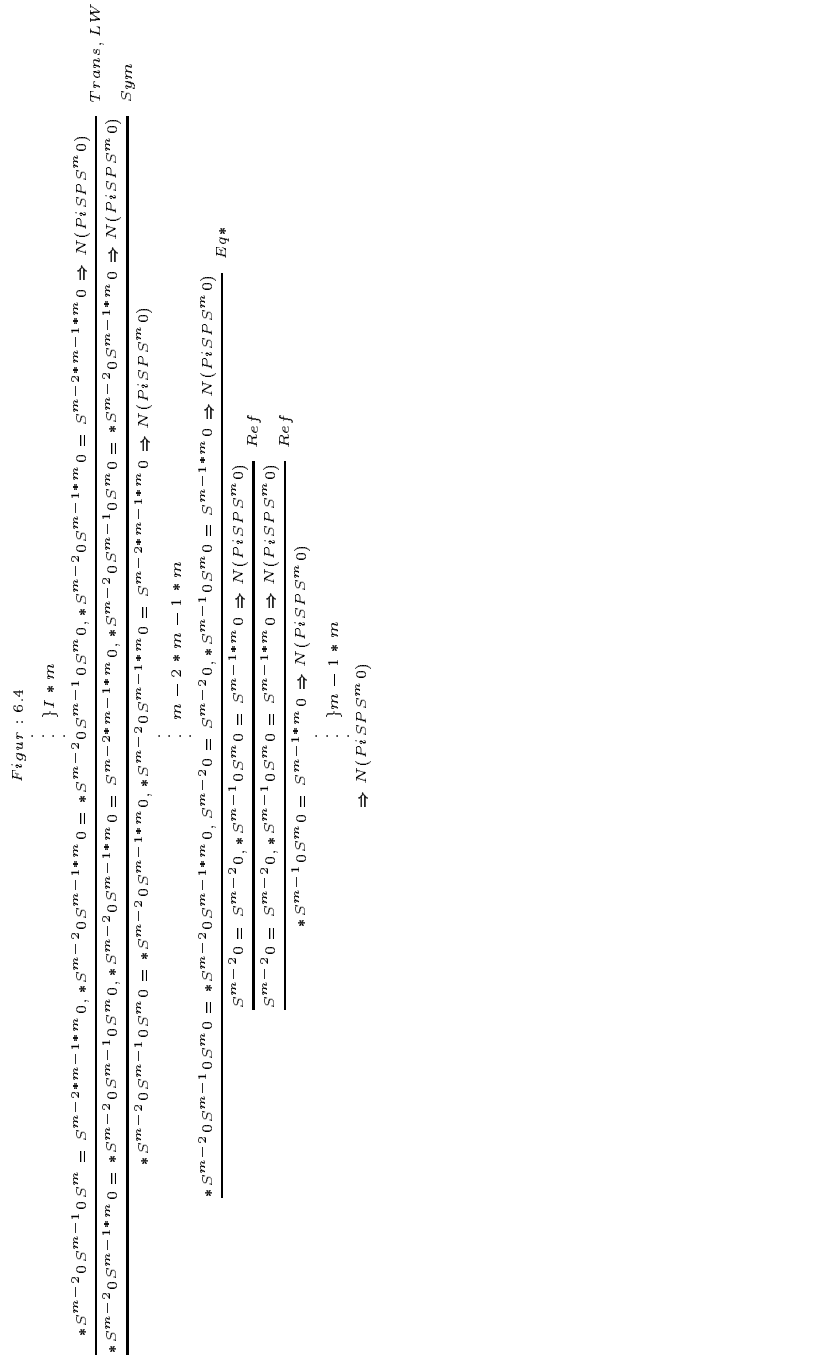
Korrolar 6.3 *Hvis f er en n-ær strengt voksende elementær funksjon finnes det en utledning $\mathcal{D}$ av $N(f\bar{m}_1, \ldots, \bar{m}_n)$ for alle m. $\mathcal{D}$ har høyde som begrenses oppad av en funksjon lineær i m.*

Bevis. Fra teorem 5.13 følger det at det kan utledes induktive predikater for alle de strengt voksende elementære funksjonene i PRA. Resten følger fra teorem 6.2 □

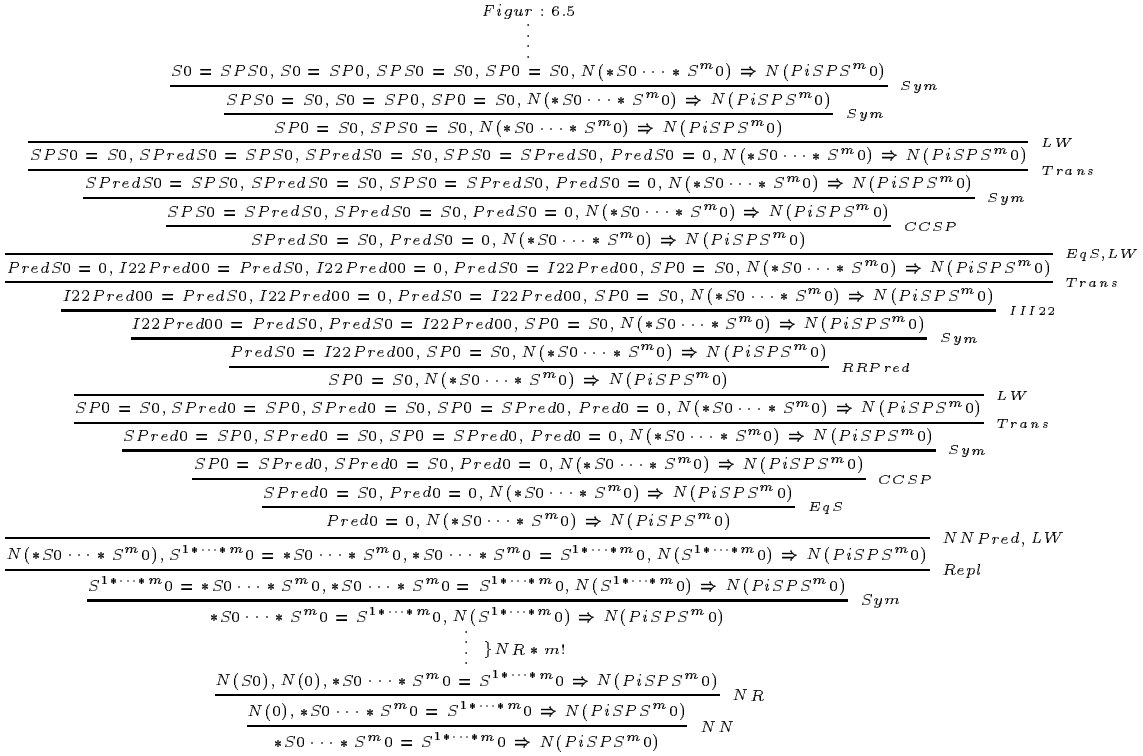
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{ax}{N(+S^m0S^n0), S^{m+n}0 = +S^m0S^n0, N(S^{m+n}0), +S^m0S^n0 = S^{m+n}0 \Rightarrow N(+S^m0S^n0)}\,Repl}{S^{m+n}0 = +S^m0S^n0, N(S^{m+n}0), +S^m0S^n0 = S^{m+n}0 \Rightarrow N(+S^m0S^n0)}\,Sym}{+S^m0S^n0 = S^{m+n}0, N(S^{m+n}0) \Rightarrow N(+S^m0S^n0)}}{\vdots\ \}NR * (n+m)}}{N(S0), N(0), +S^m0S^n0 = S^{m+n}0 \Rightarrow N(+S^m0S^n0)}\,NR}{N(0), +S^m0S^n0 = S^{m+n}0 \Rightarrow N(+S^m0S^n0)}\,NN}{\vdots\ \}EqS, RRAdd, Sym, Trans * m}}{+S0S^n0 = SS^n0, S + 0S^n0 = +S0S^n0, S + 0S^n0 = SS^n0, +S0S^n0 = S + 0S^n0, +0S^n0 = S^n0 \Rightarrow N(+S^m0S^n0)}\,Trans}{S + 0S^n0 = +S0S^n0, S + 0S^n0 = SS^n0, +S0S^n0 = S + 0S^n0, +0S^n0 = S^n0 \Rightarrow N(+S^m0S^n0)}\,Sym}{+S0S^n0 = S + 0S^n0, S + 0S^n0 = SS^n0, +0S^n0 = S^n0 \Rightarrow N(+S^m0S^n0)}\,RRAdd}{S + 0S^n0 = SS^n0, +0S^n0 = S^n0 \Rightarrow N(+S^m0S^n0)}\,EqS}{+0S^n0 = S^n0 \Rightarrow N(+S^m0S^n0)}\,NNAdd}{\Rightarrow N(+S^m0S^n0)}$$

Figur 6.1: Addisjon

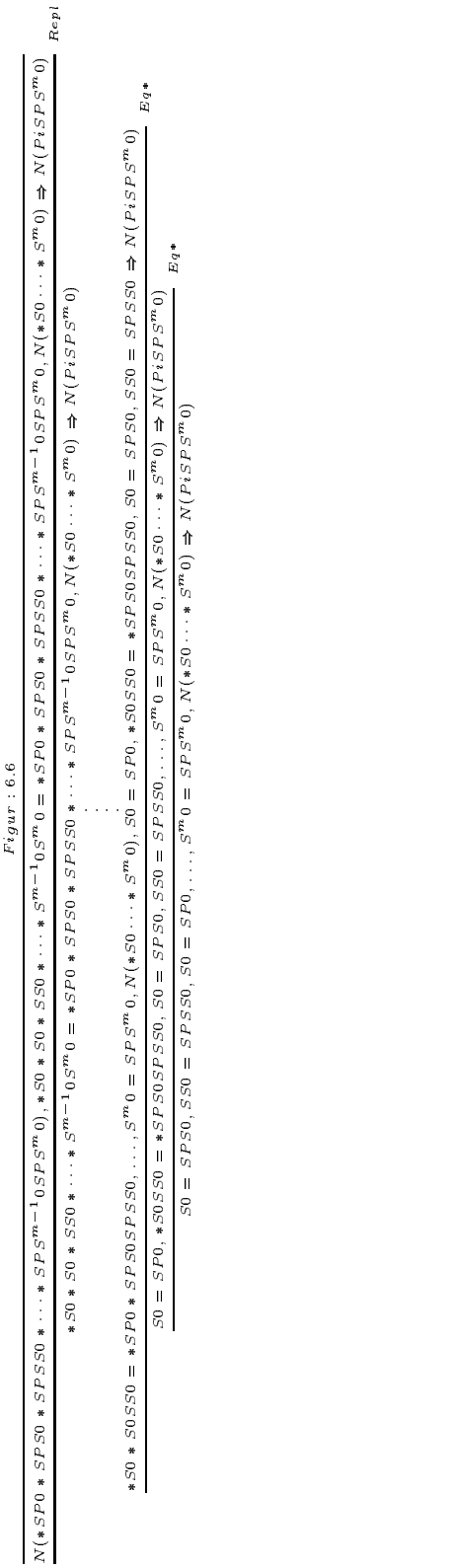
Figur 6.2: Multiplikasjon



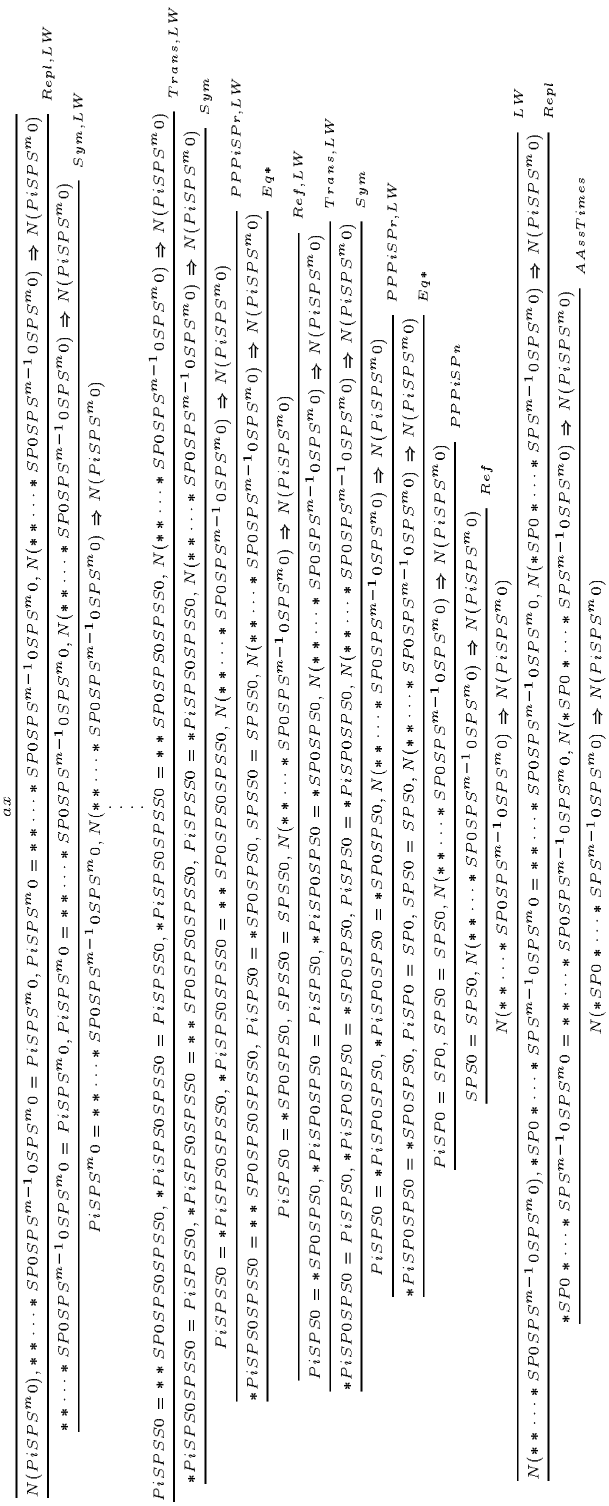
Figur 6.3: Fakultet: 1



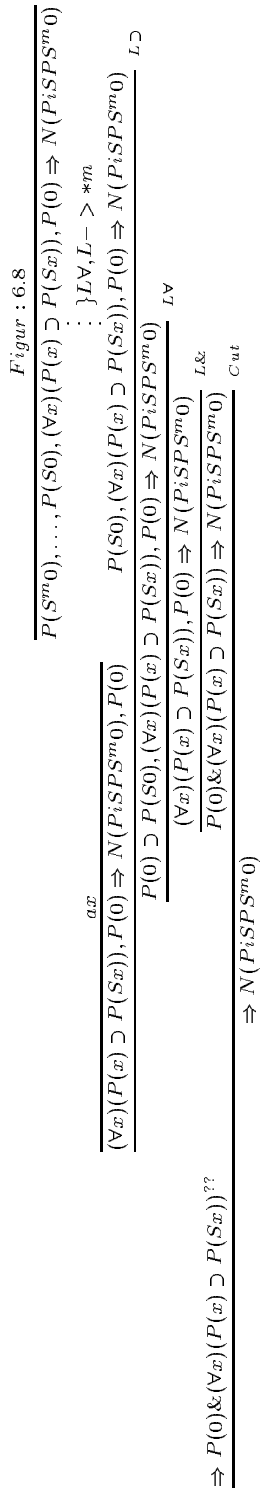
Figur 6.4: Fakultet: 2



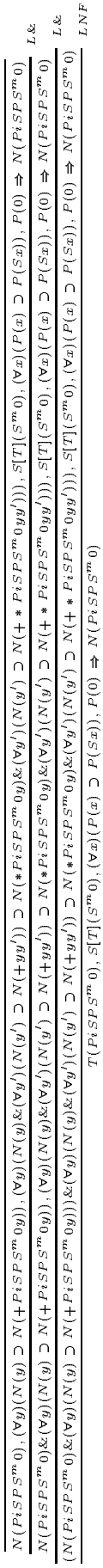
Figur 6.5: Fakultet: 3



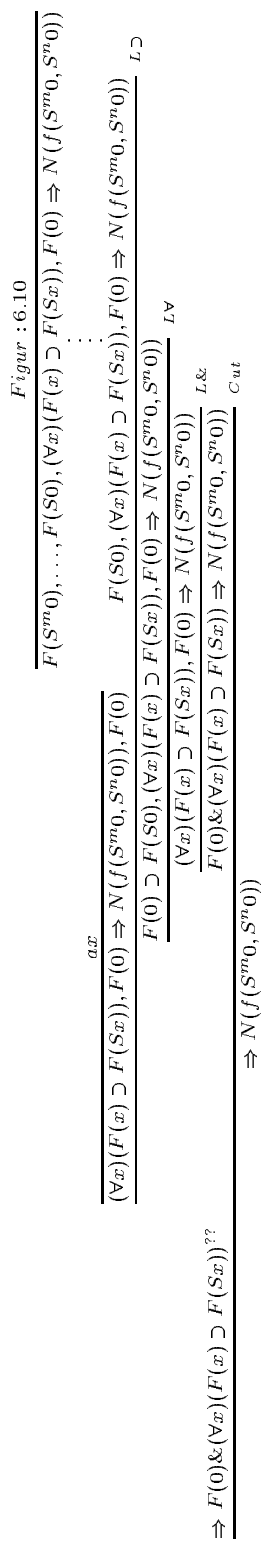
Figur 6.6: Fakultet 4



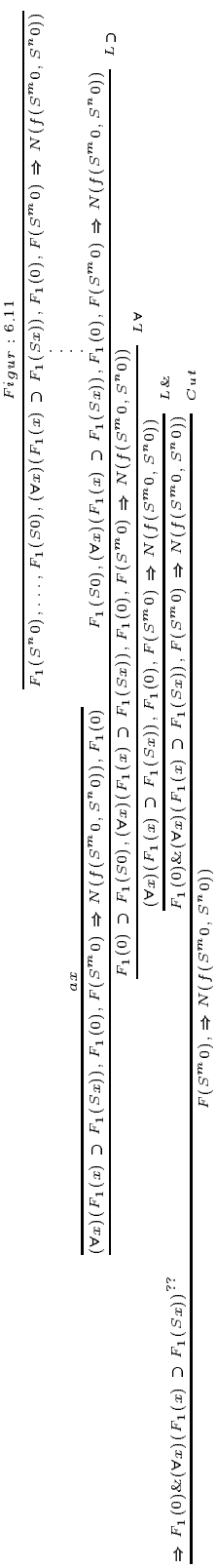
Figur 6.7: Fakultet med snitt 1



Figur 6.8: Fakultet med snitt 2



Figur 6.9: Bevis med snitt 1



Figur 6.10: Bevis med snitt 2

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0)), F_1(S^n0)}\,ax}{F_1(S^n0) \subset F_1(fS^m0S^n0), (\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,\forall_1}{(\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,L\&}{F_1(fS^m0S^n0), (\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,\subset_1$$

$$\dfrac{\dfrac{\dfrac{}{N(fS^m0S^n0), \ldots, (\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,ax}{N(fS^m0S^n0)\& \ldots, (\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,L\&$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F_1(S^n0), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,ax}{F(S^n0), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,L_{Exp}}{F_1(S^n0)\&(\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,L\&}{(\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,\forall_1}{F_1(S^n0) \subset F_1(fS^m0S^n0), (\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,\forall_1}{(\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,L\&}{F_1(fS^m0S^n0), (\forall y)(F_1(y) \subset F_1(fS^m0y)), F_1(S^n0) \Rightarrow N(f(S^m0, S^n0))}\,\subset_1$$

Figur 6.11: Bevis med snitt 3

Kapittel 7

Koding av bevistrær i PRA

En metode for å aritmetisere formelle uttrykk ble først gitt av Gödel [7]. Aritmetisering innebærer simpelthen å oversette til aritmetikk, eller tall. Gödel viste at det er mulig å tildele et unikt tall til hvert symbol, utsagn og bevis, eller sekvens av utsagn og brukte dette til å vise ufullstendighetsteoremet.

7.1 Aritmetisering

Det finnes mange måter å aritmetisere formelle uttrykk. Odifreddi [23] gir en metode for å tilegne tall til beregningstrær for primitivt rekursive funksjoner. Her utvider jeg denne metoden, slik at jeg kan tilegne tall til bevistrær i PRA. Jeg gjør bruk av de numeriske verktøyene for aritmetisering som ble introdusert i avsnitt 2.1.2. Når hver regel i PRA er gitt en unik koding kan man definere et predikat som sjekker hvorvidt et sekvenstall koder en utledning av et uttrykk $N(f(\bar{m}_1, \ldots, \bar{m}_n))$. Når man i tillegg har et bånd på kodetallet for en utledning kan predikatet brukes i kombinasjon med begrenset søk til å finne kodetallet.

Teorem 7.1 *For enhver n-ær primitivt rekursiv funksjon f finnes det et elementært predikat $\mathcal{T}_f$, slik at for alle $m_1, \ldots, m_n \in N$ finnes det et tall y slik at følgende holder: $\mathcal{T}_f(y, m_1, \ldots, m_n)$*

Bevis. La uten tap av generalitet $n = 2$. Idéen er å assosiere tall til symboler, formler, regler og bevistrær i PRA, slik at $\mathcal{T}_f(y, l, m)$ uttrykker at y koder et bevistre for $N(f(\bar{l}, \bar{m}))$.

1. Først må det assosieres tall til symboler, uttrykk og regler i PRA. Hvert tall må være unikt slik at to symboler og så videre ikke kodes av samme tall. Symboler, konnektiver, predikater og variable kodes med primtall større en 2. Dette er trygt siden en sekvens per definisjon har lengde

større eller lik 1 og alle sekvenstall derfor er delelig på 2. Tallet 2 selv går ut siden $\langle 0 \rangle = 2$.

Sekventpil

$\ulcorner \Rightarrow \urcorner \stackrel{def}{=} 3$

Logiske konnektiver

$\ulcorner \wedge \urcorner \stackrel{def}{=} 5$

$\ulcorner \vee \urcorner \stackrel{def}{=} 7$

$\ulcorner \bot \urcorner \stackrel{def}{=} 11$

$\ulcorner \supset \urcorner \stackrel{def}{=} 13$

$\ulcorner \forall \urcorner \stackrel{def}{=} 17$

$\ulcorner \exists \urcorner \stackrel{def}{=} 19$

Substitusjon

$\ulcorner sub \urcorner \stackrel{def}{=} 23$

Predikater/relasjoner

$\ulcorner = \urcorner \stackrel{def}{=} 29$

$\ulcorner N \urcorner \stackrel{def}{=} 31$[1]

Koding av termer defineres induktivt ved:

(a) Navn: PRA inneholder ingen navn. Tallet 0 er simpelthen funksjonen $\mathcal{O}$ uten argumenter.

(b) Variable tildeles primtall større enn 31.

$\ulcorner x_0 \urcorner, \ulcorner x_1 \urcorner, \ldots, \ulcorner x_i \urcorner, \ldots \stackrel{def}{=} 37, 41, \ldots, \langle p_{1+i} \rangle, \ldots$

(c) $\ulcorner f(t_1, \ldots, t_n) \urcorner \stackrel{def}{=} \langle \ulcorner f \urcorner, \ulcorner t_1 \urcorner, \ldots, \ulcorner t_n \urcorner \rangle$ når $t_1, \ldots, t_n$ er termer.

Koding av formler defineres induktivt ved:

(a) $\ulcorner R(t_1, \ldots, t_n) \urcorner \stackrel{def}{=} \langle \ulcorner R \urcorner, \langle \ulcorner t_1 \urcorner, \ldots, \ulcorner t_n \urcorner \rangle \rangle$

(b) $\ulcorner A \wedge B \urcorner \stackrel{def}{=} \langle \ulcorner A \urcorner, \ulcorner \wedge \urcorner, \ulcorner B \urcorner \rangle$

$\ulcorner A \vee B \urcorner \stackrel{def}{=} \langle \ulcorner A \urcorner, \ulcorner \vee \urcorner, \ulcorner B \urcorner \rangle$

$\ulcorner A \supset B \urcorner \stackrel{def}{=} \langle \ulcorner A \urcorner, \ulcorner \supset \urcorner, \ulcorner B \urcorner \rangle$

$\ulcorner \forall x A \urcorner \stackrel{def}{=} \langle \ulcorner \forall \urcorner, \ulcorner x \urcorner, \ulcorner A \urcorner \rangle$

$\ulcorner \exists x A \urcorner \stackrel{def}{=} \langle \ulcorner \exists \urcorner, \ulcorner x \urcorner, \ulcorner A \urcorner \rangle$

$\ulcorner sub(t, x, A) \urcorner \stackrel{def}{=} \langle \ulcorner sub \urcorner, \ulcorner x \urcorner, \ulcorner A \urcorner \rangle$[2] når A, B er formler.

[1] I forrige kapittel ble det definert nye predikater A, M osv. I det formelle bevissystemet PRA finnes imidlertid kun to predikater; $=$ og N. Predikater som A og M er forkortelser for mer komplekse uttrykk.

[2] $sub(t, x, A) \stackrel{def}{=} A[t/x]$, det vil si t substitueres for x i A.

Funksjoner $\lceil f \rceil$ defineres induktivt over den rekursive definisjonen av f.
$\lceil f \rceil \stackrel{def}{=} \langle 0 \rangle$ hvis$f = \mathcal{O}$
$\lceil f \rceil \stackrel{def}{=} \langle 1 \rangle$ hvis$f = \mathcal{S}$
$\lceil f \rceil \stackrel{def}{=} \langle 2, n, i \rangle$ hvis$f = \mathcal{I}_i^n$
$\lceil f \rceil \stackrel{def}{=} \langle 3, \lceil h \rceil, \lceil g_1 \rceil, \ldots, \lceil g_m \rceil \rangle$ når f er komposisjon av $g_1, \ldots, g_m$og$h$.
$\lceil f \rceil \stackrel{def}{=} \langle 4, \lceil g \rceil, \lceil h \rceil \rangle$ når f er primitiv rekursjon over g og h.

Sekventer: For å lette definisjonen av predikatene i neste avsnitt tilegnes det også tall til Γ og Δ. La $\Gamma = C_1, \ldots, C_n$ og $\Delta = D_1, \ldots, D_m, n, m \geq 0$. Da er
$\lceil \Gamma \rceil \stackrel{def}{=} \langle 5, \lceil C_1 \rceil, \ldots, \lceil C_n \rceil \rangle$ og
$\lceil \Delta \rceil \stackrel{def}{=} \langle 6, \lceil D_1 \rceil, \ldots, \lceil D_m \rceil \rangle$

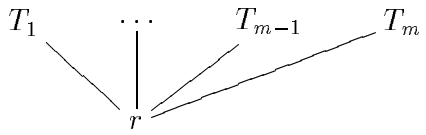
Slutningsregler:

Aksiom:	$\stackrel{def}{=} \langle 7 \rangle$
Logiske regler:	$L\wedge \stackrel{def}{=} \langle 8 \rangle, \ldots, L\bot \stackrel{def}{=} \langle 18 \rangle$
Strukturelle regler:	$Cut \stackrel{def}{=} \langle 19 \rangle$
Regler for likhet.:	$Ref \stackrel{def}{=} \langle 20 \rangle, \ldots, Sym \stackrel{def}{=} \langle 23 \rangle$
Induksjonsregler:	$NN \stackrel{def}{=} \langle 24 \rangle$ og $RN \stackrel{def}{=} \langle 25 \rangle$
Regler for assosiativitet:	$AssAdd \stackrel{def}{=} \langle 26 \rangle$ og $AssTimes \stackrel{def}{=} \langle 27 \rangle$
Projeksjonsfunksjonen:	$\mathcal{I}_i^n \stackrel{def}{=} \langle 28 \rangle$
Regelskjema:	$Eqf \stackrel{def}{=} \langle 29, \lceil f \rceil \rangle, Eqf_c \stackrel{def}{=} \langle 30, \lceil f \rceil \rangle$
	$f0 \stackrel{def}{=} \langle 31, \lceil f \rceil \rangle, frec \stackrel{def}{=} \langle 32, \lceil f \rceil \rangle,$
	$fcomp \stackrel{def}{=} \langle 33, \lceil f \rceil \rangle$

2. Så må det assosieres tall til bevistrær. Dette gjøres ved induksjon over konstruksjonen av bevistreet. Først må det assosieres tall til noder. La $\Gamma = C_1, \ldots, C_n$ og $\Delta = D_1, \ldots, D_m$. Siden hver node er på formen Regel $\Gamma \Rightarrow \Delta$, får de tall

$$\langle \lceil Regel \rceil, \lceil \Gamma \rceil, \lceil \Rightarrow \rceil, \lceil \Delta \rceil \rangle$$

Med unntak av *Axiom* og $L\bot$ har alle regler i PRA premisser og en konklusjon. Siden bevissøk i PRA foregår rot-først kodes navnet på en regel i konklusjons-noden og ikke i premissene. Med kodesystemet fra punkt 1 på plass kan det nå tildeles tall til trær: hvert tre T består av



Figur 7.1: Et tre T med rot r og deltrær T_i

en rot r, med assosiert tall r, og et endelig antall ordnede forgjengere (muligens ingen), som hver er et deltre T_i[3]. Ved induksjon tildeles treet tallet

$$\hat{T} = \langle r, \hat{T}_1, \ldots, \hat{T}_n \rangle,$$

hvor $\hat{T}_i$ er tallet assosiert med deltre T_i.

3. Definer et predikat $T(y)$ som sier at y koder et bevistre i PRA

   Dette gjøres ved å definere predikater for alle slutningsregler i PRA: $ax(y) \ldots fcomp(y)$ og sette dem sammen til et større predikat. Sekvenstallet y koder et bevistre hvis y begynner med en slutningsregel og alle delnoder av y er bevistrær. For å lette lesbarheten brukes komma istedenfor nøstede parenteser, i tråd med Odifreddis [23] notasjon. Eksempelvis skrives $(((a_i)_j)_k$ som $(a)_{i,j,k}$.

   For det første må alle trær oppfylle egenskapen:

$$S(y) \Leftrightarrow Seq(y) \wedge Seq((y)_1) \wedge ln((y)_1) = 4 \wedge Seq((y_{1,1}))$$

   I tillegg defineres en del hjelpepredikater. Predikatet *Subseq* sjekker om sekventtallet y er en delmengde av z. Den tar som argument to sekvenstall, startposisjon og sluttposisjon for delsekventene som skal sammenlignes. Predikatet *Eq* sjekker mengdelikhet av to sekventer ved

[3] I beregningsteori er det vanlig å tegne syntakstrær med rota over premissene. I bevisteori tegnes bevistrærne med rota nederst. Jeg holder meg her til tradisjonen i bevisteori og tegner trærne nedenfra og opp.

hjelp av $Subseq$.

$$Subseq(y, ys, ye, z, zs, ze) \Leftrightarrow (\forall i)_{ys \leq i \leq ye}(\exists j)_{zs \leq j \leq ze}[(y)_i = (z)_j]$$
$$Eq(y, ys, ye, z, zs, ze) \Leftrightarrow Subseq(y, ys, ye, z, zs, ze) \wedge Subseq(z, zs, ze, y, ys, ye)$$
$$Fun(y) \Leftrightarrow Seq(y) \wedge$$
$$\{[ln(y) = 1 \wedge ((y)_1 = 0 \vee (y)_1 = 1)] \vee$$
$$[ln(y) = 3 \wedge ((y)_1 = 2 \vee ((y)_1 = 4 \wedge Fun(y)_2 \wedge Fun(y)_3))] \vee$$
$$[(y)_1 = 3 \wedge (\forall i)_{2 \leq i \leq ln(y)} Fun(y)_i]\}$$
$$primesquare(x) \Leftrightarrow (\exists i)_{i \leq x}[i \times i = x \wedge prime(i)]$$
$$Var(y) \Leftrightarrow y \geq 37 \wedge primesquare(y)$$
$$Term(y) \Leftrightarrow Var(y) \vee (Seq(y) \wedge Fun(y)_1 \wedge (\forall i)_{2 \leq i \leq ln(y)} Term(y)_i)$$
$$Rel(y) \Leftrightarrow y \geq 29 \wedge prime(y)$$
$$Atomic(y) \Leftrightarrow Seq(y) \wedge ln(y) = 2 \wedge Rel(y)_1 \wedge (\forall i)_{1 \leq i \leq ln(y)_2} Term(y)_{2,i}$$

Det finnes til sammen 29 typer regler i PRA. Predikatene som tester om et tall koder en regel i PRA er omstendelige å lese. Jeg har tatt med et eksempel på en logisk og en ikkelogisk regel her som man kan lese for å forstå hvordan predikatene fungerer. I tillegg B viser jeg noen flere typiske tilfeller. De resterende predikatene kan defineres på tilsvarende måte. Jeg gir eksempler på predikater for reglene $L\wedge$ og Ref:

$$L\& \quad \frac{\langle \ulcorner R2 \urcorner, \langle 5, \ulcorner A \urcorner, \ulcorner B \urcorner, \ulcorner C_2 \urcorner, \ldots, \ulcorner C_n \urcorner \rangle, 3, \ulcorner \Delta \urcorner \rangle}{\langle \langle 8 \rangle, \langle 5, \langle \ulcorner A \urcorner, 5, \ulcorner B \urcorner \rangle, \ulcorner C_2 \urcorner, \ldots, \ulcorner C_n \urcorner \rangle, 3, \ulcorner \Delta \urcorner \rangle}$$

$$LA(y) \Leftrightarrow ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge$$
$$(y)_{1,1} = \langle 8 \rangle \wedge (y)_{1,2,2,2} = 5 \wedge (y)_{2,1,2,2} = (y)_{1,2,2,1}$$
$$\wedge (y)_{2,1,2,3} = (y)_{1,2,2,3} \wedge (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2$$
$$\wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge$$
$$Eq(y, (y)_{1,2,3}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,4}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge$$
$$ln(y)_{2,1,4} = ln(y)_{1,4}$$
$$\wedge Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})$$

$$Ref \quad \frac{\langle \ulcorner R2 \urcorner, \langle 5, \langle 29, \langle \ulcorner a \urcorner, \ulcorner a \urcorner \rangle \rangle, \ulcorner C_2 \urcorner, \ldots, \ulcorner C_n \urcorner \rangle, 3, \ulcorner \Delta \urcorner \rangle}{\langle \langle 20 \rangle, \ulcorner \Gamma \urcorner, 3, \ulcorner \Delta \urcorner \rangle}$$

$$\begin{aligned}Ref(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\ & (y)_{1,1} = \langle 20 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 29 \wedge \\ & Term(y)_{2,1,2,2,2,1} \wedge (y)_{2,1,2,2,2,1} = (y)_{2,1,2,2,2,2} \wedge \\ & (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\ & ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\ & Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & ln(y)_{2,1,4} = ln(y)_{1,4} \\ & \wedge Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

Nå kan egenskapen "y koder et bevistre i PRA" defineres induktivt:

$$\begin{aligned}\mathcal{T}(y) \Leftrightarrow & S(y) \wedge [Ax(y) \vee \cdots \vee LA(y) \vee \cdots \vee Ref(y) \vee \cdots \vee fcomp(y)] \\ & \wedge [ln(y) > 1 \supset (\forall i)_{2 \leq i \leq ln(y)} \mathcal{T}((y)_i)].\end{aligned}$$

Funksjonen $(y)_i$ gir eksponenten til det i'te primtallet i primtallsfaktoriseringen av y. Så $(y)_i \leq y$ og man ser at den karakteristiske funksjonen til $\mathcal{T}$ bruker verdiene fra sine tidligere argumenter. Videre er $\mathcal{T}(y) \leq c_1^1(y)$. Dermed kan den karakteristiske funksjonen til $\mathcal{T}$ defineres ved skjemaet for bundet verdiforløprekursjon. Siden alle hjelpefunksjonene til $\mathcal{T}$ og $c_1^1(y)$ er elementære, så er $\mathcal{T}$ elementær.

4. Definer $\mathcal{T}_f$. Hvis sekvenstallet y koder et bevistre for $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ må $(y)_1$ være:

$$\langle \lceil R \rceil, \langle 5 \rangle, 3, \langle 6, \langle 31, \langle \langle \lceil f \rceil, \langle \lceil \bar{m}_1 \rceil, \ldots, \lceil \bar{m}_n \rceil \rangle \rangle \rangle \rangle \rangle$$

La

$$\begin{aligned}R(f, y, m_1, \ldots, m_n) \Leftrightarrow & ln(y) = 4 \wedge ln(y)_2 = 1 \wedge (y)_{2,1} = 1 \wedge (y)_3 = 3 \wedge \\ & ln((y)_4) = 2 \wedge (y)_{4,1} = 2 \wedge ln((y)_{4,2}) = 2 \wedge \\ & (y)_{4,2,1} = 31 \wedge ln((y)_{4,2,2}) = 1 \wedge \\ & ln((y)_{4,2,2,1}) = 2 \wedge (y)_{4,2,2,1,1} = \lceil f \rceil \wedge \\ & (\forall i)_{1 \leq i \leq ln((y)_{4,2,2,1,2})}((y)_{4,2,2,1,2,i} = \lceil m_i \rceil)\end{aligned}$$

og

$$\mathcal{T}_f(y, \vec{m}) \Leftrightarrow R(f, (y)_1, \vec{m}) \wedge \mathcal{T}(y)$$ □

$\mathcal{T}_f$ er elementær.

Kapittel 8

Induktive funksjoner er elementære

I dette kapitlet vises at hvis en funksjon er induktiv, så er den elementær. I kapittel 5 viste jeg at de strengt voksende elementære funksjonene er induktive i PRA+$AddAss, TimesAss$. Som nevnt i avsnitt 5.4.1 får jeg dermed ingen reell avgrensning av de elementære funksjonene. Dette skyldes problemene med modifisert subtraksjon, beskrevet i avsnitt 5.4.

For en vilkårlig induktiv funksjon f konstrueres en elementær algoritme for å beregne $f(m_1, \ldots, m_n)$ for alle argumenter $m_1, \ldots, m_n$. Algoritmen konstrueres ved hjelp av en teknikk med primtallskoding, definert i kapittel 7. Hovedtrekkene i beviset går som følger: Når f er induktiv finnes en utledning, $\mathcal{D}$, av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ med høyde lineær i $m_1, \ldots, m_n$. Ved snitteliminasjonsteoremet kan $\mathcal{D}$ overføres til en snittfri utledning, $\mathcal{D}'$. En snittfri utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ kan overføres til en utledning, $\mathcal{D}''$, på normalform som inneholder en beregning av $f(\bar{m}_1, \ldots, \bar{m}_n)$. Transformasjonene gjøres elementært. Det finnes et elementært bånd på kodetallet for $\mathcal{D}''$, når høyden av utledningen er elementær i $m_1, \ldots, m_n$. Båndet brukes sammen med predikatet som sjekker om et tall koder en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, definert i forrige kapittel, til å finne kodetallet for $\mathcal{D}''$. Når $\mathcal{D}''$ er kodet som et tall kan beregningen av $f(\bar{m}_1, \ldots, \bar{m}_n)$ plukkes ut fra utledningen ved hjelp av en elementær algoritme.

8.1 Normalform

I avsnitt 3.3 ble det vist hvordan man kan sette sammen utledninger av $f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$ og $N(\bar{k})$ til en direkte utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$. En slik utledning innebærer en beregning av uttrykket $f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$

og man kan bruke den til å plukke ut talltermen $\bar{k}$. Slik reglene er i PRA er det ikke alltid tilfelle at en kan plukke ut en tallterm $\bar{k}$, fra en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, når $f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$.

Dette skyldes regelen RN:

$$\frac{N(St), N(t), \Gamma \Rightarrow \Delta}{N(t), \Gamma \Rightarrow \Delta}\ RN$$

I funksjoner som rekurserer over etterfølgerfunksjonen, kan regelen RN erstatte Eqf. Dette kommer fram av eksemplet under.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{N(+S00), S(0+0) = S0+0, N(S(0+0)), S0+0 = S(0+0), 0 = 0+0, N(0) \Rightarrow N(+S00)}\ ax}{S(0+0) = S0+0, N(S(0+0)), S0+0 = S(0+0), 0 = 0+0, N(0) \Rightarrow N(+S00)}\ Repl}{S0+0 = S(0+0), N(S(0+0)), 0 = 0+0, N(0) \Rightarrow N(+S00)}\ Sym}{N(S(0+0)), 0 = 0+0, N(0) \Rightarrow N(+S00)}\ RAdd}{N(0+0), 0 = 0+0, N(0) \Rightarrow N(+S00)}\ RN}{0 = 0+0, N(0), 0+0 = 0 \Rightarrow N(+S00)}\ Repl, LW}{0+0 = 0, N(0) \Rightarrow N(+S00)}\ Sym}{N(0) \Rightarrow N(+S00)}\ AddN}{\Rightarrow N(+S00)}\ NN$$

Figur 8.1: Utledning som ikke er på normalform

For å kunne plukke ut den korresponderende talltermen til $f(m_1, \ldots, m_n)$ fra en utledning, må den være på normalform.

Definisjon 8.1 (Kanonisk form)

1. *0 er på kanonisk form,*
2. *hvis t er på kanonisk form, så er $\mathcal{S}t$ på kanonisk form.*

Definisjon 8.2 (Normalform) *La f være en n-ær funksjon. En snittfri utledning $\mathcal{D}$ av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ er på normalform hvis $f(\bar{m}_1, \ldots, \bar{m}_n)$ er på kanonisk form eller $f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{k}$ forekommer i antecedenten hvor $\bar{k}$ er på kanonisk form, når $f(m_1, \ldots, m_n) = k$.*

Jeg viser at enhver utledning av et uttrykk av typen $N(f(\bar{m}_1, \ldots, \bar{m}_n))$ kan overføres til normalform. Et deltre kalles for en utledning når det er opplagt at den kan gjøres om til en utledning som beskrevet i observasjon 5.20.

Teorem 8.3 *La $\mathcal{D}$ være en snittfri utledning hvor $N(f(\bar{t}_1, \ldots, \bar{t}_n))$ er hovedformel på begge sider av sekventpila i toppnoden. Da er $f(\bar{t}_1, \ldots, \bar{t}_n)$ på kanonisk form, eller så kan $\mathcal{D}$ overføres til en utledning $\mathcal{D}'$, hvor $f(\bar{t}_1, \ldots, \bar{t}_n) = \bar{k}$ forekommer i Γ i toppsekventen, når $f(t_1, \ldots, t_n) = k$. D' har høyde høyst $2|\mathcal{D}|$.*

Bevis. Teorem 8.3 vises ved induksjon over høyden av $\mathcal{D}$.

Basistilfelle: $|\mathcal{D}| = 1$. Da må f være $\mathcal{O}$ og $\mathcal{D}$ er:

$$\dfrac{\dfrac{}{N(\mathcal{O}) \Rightarrow N(\mathcal{O})}\, ax}{\Rightarrow N(\mathcal{O})}\, NN$$

Her er $f(\bar{t}_1, \ldots, \bar{t}_n)$ på kanonisk form.

Induksjonshypotese: Påstanden holder for utledninger $\mathcal{D}_i$ med høyde n.

Induksjonssteg: Må vise at påstanden holder for $\mathcal{D}$ med høyde $n+1$. Se på den øverste regelen $\mathcal{R}$ i utledningen:

$$\dfrac{\dfrac{}{N(f(\bar{t}_1, \ldots, \bar{t}_n)), \Gamma \Rightarrow \Delta}\, ax}{\Gamma \Rightarrow \Delta}\, \mathcal{R}$$
$$\mathcal{D}_0$$

De eneste reglene hvor $N(f(\bar{t}_1, \ldots, \bar{t}_n))$ kan være hovedformel på venstre side i antakelsen er RN og $Repl$. Hvis $\mathcal{R}$ er RN må $f = \mathcal{S}$ med aritet 1:

$$\dfrac{\dfrac{}{N(\mathcal{S}(\bar{t})), N(\bar{t}), \Gamma \Rightarrow \Delta}\, ax}{N(\bar{t}), \Gamma \Rightarrow \Delta}\, RN$$
$$\mathcal{D}_0$$

Ved induksjonshypotesen er enten $\bar{t}$ på kanonisk form eller så kan $\mathcal{D}_0$ overføres til en utledning $\mathcal{D}_0'$, hvor $\bar{t} = \bar{k}$ forekommer i Γ, for et tall k. Hvis $\bar{t}$ er på kanonisk form er også $\mathcal{S}(\bar{t})$ på kanonisk form ved definisjon 8.1. I det andre tilfellet gir en anvendelse av regelen EqS utsagnet $\mathcal{S}(\bar{t}) = \mathcal{S}(\bar{k})$ i antecedenten. Ved definisjon 8.1 er også $\mathcal{S}(\bar{k})$ på kanonisk form. $\mathcal{D}_0'$ har høyde høyst $2|\mathcal{D}_0|$:

$$\dfrac{\dfrac{\dfrac{}{N(\mathcal{S}(\bar{t})), N(\bar{t}), \mathcal{S}(\bar{t}) = \mathcal{S}(\bar{k}), \bar{t} = \bar{k}, \Gamma' \Rightarrow \Delta}\, ax}{N(\bar{t}), \mathcal{S}(\bar{t}) = \mathcal{S}(\bar{k}), \bar{t} = \bar{k}, \Gamma' \Rightarrow \Delta}\, RN}{N(\bar{t}), \bar{t} = \bar{k}, \Gamma' \Rightarrow \Delta}\, EqS$$
$$\mathcal{D}_0'$$

La $\mathcal{R}$ være $Repl$ med $N(f(\bar{t}_1, \ldots, \bar{t}_n))$ som hovedformel. La uten tap av generalitet $n = 2$:

$$\dfrac{\dfrac{}{N(f(\bar{t}_1, \bar{t}_2)), \bar{t}_3 = f(\bar{t}_1, \bar{t}_2), N(\bar{t}_3), \Gamma \Rightarrow \Delta}\, ax}{\bar{t}_3 = f(\bar{t}_1, \bar{t}_2), N(\bar{t}_3), \Gamma \Rightarrow \Delta}\, Repl$$
$$\mathcal{D}_0$$

Hvis $\bar{t}_3$ er på kanonisk form er alt greit. I det andre tilfellet kan $\mathcal{D}_0$ overføres til en utledning $\mathcal{D}_0'$, hvor $\bar{t}_3 = \bar{q}$ forekommer i Γ i toppsekventen hvor $\bar{q}$ er på kanonisk form, når $t_3 = q$, ved induksjonshypotesen. $\mathcal{D}_0'$ har høyde høyst $2|\mathcal{D}_0|$. *Trans* og *Repl* gir $\mathcal{D}'$:

$$\dfrac{\dfrac{\dfrac{}{N(f(\bar{t}_1,\bar{t}_2)),\bar{t}_3 = f(\bar{t}_1,\bar{t}_2), N(\bar{t}_3), f(\bar{t}_1,\bar{t}_2) = \bar{q}, \bar{t}_3 = \bar{q}, \Gamma \Rightarrow \Delta}\, ax}{\bar{t}_3 = f(\bar{t}_1,\bar{t}_2), N(\bar{t}_3), f(\bar{t}_1,\bar{t}_2) = \bar{q}, \bar{t}_3 = \bar{q}, \Gamma \Rightarrow \Delta}\, Repl}{\bar{t}_3 = f(\bar{t}_1,\bar{t}_2), \bar{t}_3 = \bar{q}, N(\bar{t}_3), \Gamma \Rightarrow \Delta}\, Trans$$

$\mathcal{D}_0'$

Når $\mathcal{R}$ er RN eller *Repl* gis høyden på den nye utledningen av likningen:

$$|\mathcal{D}'| = |\mathcal{D}_0'| + 2 \overset{IH}{\leq} 2|\mathcal{D}_0| + 2 \leq 2(|\mathcal{D}_0| + 1) \overset{def}{=} 2|\mathcal{D}|.$$

Korrolar 8.4 (Normalisering) *La $\mathcal{D}$ være en snittfri utledning av sekventen $\Rightarrow N(f(\bar{m}_1,\ldots,\bar{m}_n))$. Da kan $\mathcal{D}$ overføres til en utledning $\mathcal{D}'$ på normalform, med høyde høyst $2|\mathcal{D}|$.*

Bevis. Anta at $\mathcal{D}$ er en snittfri utledning av $\Rightarrow N(f(\bar{m}_1,\ldots,\bar{m}_n))$. Uten tap av generalitet la $n = 2$. Ved teorem 8.3 er $f(\bar{l},\bar{m})$ på kanonisk form, eller så kan $\mathcal{D}$ overføres til en utledning $\mathcal{D}'$, hvor $f(\bar{l},\bar{m}) = \bar{k}$ forekommer i Γ i toppsekventen, når $f(l,m) = k$. Hvis $f(\bar{l},\bar{m})$ er på kanonisk form er $\mathcal{D}$ på normalform allerede. I det andre tilfellet overføres $\mathcal{D}$ til en utledning som per definisjon er på normalform. Ved teorem 8.3 har $\mathcal{D}'$ høyde høyst $2|\mathcal{D}|$. □

I kapittel 7 ble predikatet $\mathcal{T}_f$ definert, som er slik at $\mathcal{T}_f(y,m_1,\ldots,m_n)$ gir 1 hvis y koder en snittfri utledning i PRA av $N(f(\bar{m}_1,\ldots,\bar{m}_n))$, 0 ellers.

Ved å kreve at i regelen RN:

$$\frac{N(Sa), N(a), \Gamma \Rightarrow \Delta}{N(a), \Gamma \Rightarrow \Delta}\, RN$$

må termen a være på kanonisk form kan man definere et predikat som sjekker hvorvidt en utledning i PRA er på normalform.

Jeg trenger følgende hjelpepredikat

$$\begin{aligned} Canonic(y) \Leftrightarrow & Seq(y) \wedge \\ & \{[ln(y) = 1 \wedge (y)_1 = \langle 0\rangle] \vee \\ & [ln(y) > 1 \wedge (y)_1 = \langle 1\rangle \wedge (\forall i)_{i \leq 2 \leq ln(y)} Canonic(y)]\} \end{aligned}$$

Predikatet for RN definert i avsnitt 7.1 byttes ut med RN_{nf} som gir 1 hvis y koder et tre hvor siste regel er RN anvendt på en term på kanonisk form, 0 ellers.

$$\begin{aligned}RN_{nf}(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\ & (y)_{1,1} = 25 \wedge ln(y)_{2,1,2,2} = ln(y)_{1,2,2} = 2 \\ & \wedge (y)_{2,1,2,2,1} = (y)_{1,2,2,1} = 31 \wedge Term(y)_{1,2,2,2,1} \wedge Canonic(y)_{1,2,2,2,1} \\ & \wedge ln(y)_{2,1,2,2,2} = 2 \wedge (y)_{2,1,2,2,2,1} = \langle 1 \rangle \wedge (y)_{2,1,2,2,2,2,1} = (y)_{1,2,2,2,1} \\ & \wedge (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\ & ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\ & Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & ln(y)_{2,1,4} = ln(y)_{1,4} \\ & \wedge Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

La $\mathcal{T}_{nf}$ være $\mathcal{T}$, definert i avsnitt 7.1 med predikatet for snittregelen utelatt i definisjonen og $RN(y)$ byttet ut med RN_{nf}.

Korrolar 8.5 *[til teorem 7.1] Predikatet* $\mathcal{T}_{nff}$

$$\mathcal{T}_{nff}(y, \vec{m}) \Leftrightarrow R(f, (y)_1, \vec{m}) \wedge \mathcal{T}_{nf}(y)$$

som er slik at $\mathcal{T}_{nff}(y, \vec{m}) = 1$ *hvis* y *koder en utledning på normalform av* $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$ *i* PRA *og 0 ellers er elementært.*

Bevis. Predikatet $\mathcal{T}_{nff}$ defineres ved hjelp av $\mathcal{T}_{nf}$ og R, definert i avsnitt 7.1. Da følger det at $\mathcal{T}_{nff}$ er elementær fra beviset av teorem 7.1. □

8.2 Bånd på bevissøk

Her viser jeg at det finnes et elementært bånd på kodetallet for en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, når høyden av utledningen er elementær i $m_1, \ldots, m_n$. Dette båndet kan brukes sammen med predikatet som sjekker om et tall koder en utledning av $N(f(\bar{m}_1, \ldots, \bar{m}_n))$, definert i forrige kapittel, til å finne kodetallet for utledningen. For å gi et bånd på en slik utledning må jeg først vise at det finnes et bånd på maksimalt antall formler i bevistreet, maks antall symboler i hver formel og et bånd på kodetallet til hvert symbol. En utledning av $\Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{n}$ innebærer en beregning av uttrykket $f(m_1, \ldots, m_n)$. Derfor vil kun funksjonstermer som inngår i definisjonen av f forekomme i en slik utledning. Hvis f er en voksende funksjon og $f(m_1, \ldots, m_n) = n$ virker det rimelig at n er det største tallet som produseres

i beregningen av $f(m_1, \ldots, m_n)$ og at beregningen krever minst n skritt. Hvis f ikke er voksende og beregningen av $f(m_1, \ldots, m_n)$ krever k skritt kan det vises at det største tallet som forekommer i beregningen er mindre eller lik $\max(m_1, \ldots, m_n, k)$. Intuisjonen bak dette er at definisjonsskjemaene for primitiv rekursjon krever at funksjonene beregnes ved såkalt streng evaluering. Det vil si at alle argumenter i et funksjonskall regnes ut før selve funksjonen beregnes. Eventuelle hjelpefunksjoner som produserer store tall må dermed regnes ut. Unntaket er projeksjonsfunksjonen, $\mathcal{I}_i^n(\bar{m}_1, \ldots, \bar{m}_n) = \bar{m}_i$. Ved å si at det største tallet er mindre eller lik $\max(m_1, \ldots, m_n, k)$ har man imidlertid tatt høyde for dette.

Lemma 8.6 *La y kode en primitivt rekursiv funksjon f. Alle funksjoner som inngår i definisjonen av f har kodetall mindre eller lik y.*

Lemma 8.7 *La $\mathcal{D}$ være en utledning av $\Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{n}$.*

1. *Formlene i $\mathcal{D}$ inneholder kun funksjonstermer som inngår i definisjonen av f.*

2. *Det største tallet som beskrives i $\mathcal{D}$ er mindre eller lik det maksimale av argumentene til f og høyden av utledningen,* $\max(m_1, \ldots, m_n, |\mathcal{D}|)$.

Bevis. (Bevis av lemma 8.6 og 8.7) Ved induksjon over oppbyggingen av f med subinduksjon på rekursjonsargumentet til f. □

Korrolar 8.8 *La $\mathcal{D}$ være en utledning på normalform av $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n)))$.*

1. *Formlene i $\mathcal{D}$ inneholder kun funksjonstermer som inngår i definisjonen av f.*

2. *Det største tallet som beskrives i $\mathcal{D}$ er mindre eller lik det maksimale av argumentene til f og høyden av utledningen,* $\max(m_1, \ldots, m_n, |\mathcal{D}|)$.

Bevis. En utledning $\mathcal{D}$ på normalform av $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n)))$ konstrueres ved å sette sammen en utledning av $\Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{n}$ og $f(m_1, \ldots, m_n)$ anvendelser av regelen RN. Dermed vil egenskapene for utledningen av $\Rightarrow f(\bar{m}_1, \ldots, \bar{m}_n) = \bar{n}$ gitt ved lemma 8.6 og 8.7 også gjelde for $\mathcal{D}$. □

Fra lemmaene og korrolaret over har jeg bånd på kodetallene for funksjonssymboler og talltermer som inngår i en utledning av $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n)))$. Dette bruker jeg i beviset av følgende teorem:

Teorem 8.9 *La $\mathcal{D}$ være en utledning på normalform*

$$\mathcal{D} \vdash^{\mathrm{PRA}}_{\sigma(m_1, \ldots, m_n)} \Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n)),$$

med $\sigma \in \mathcal{E}^r, r \geq 3$. Da finnes en funksjon $\sigma' \in \mathcal{E}^r$ slik at $\sigma'(m_1, \ldots, m_n)$ legger et øvre bånd på kodetallet til $\mathcal{D}$.

Bevis. Anta at $\mathcal{D}$ er en utledning på normalform som gitt i teorem 8.9. La uten tap av generalitet $n = 2$. Jeg viser at man kan finne et bånd på maksimalt antall formler i bevistreet, maks antall symboler i hver formel og et bånd på kodetallet til hvert symbol. Jeg bruker notasjonen introdusert i avsnitt 2.2.4 hvor utledning av premisset angis med $\mathcal{D}_0$ og dybden av $\mathcal{D}_n$ angis med d_n.

Ved delformelegenskapen av snittfrie utledninger i PRA inneholder $\mathcal{D}$ kun ikkelogiske slutningsregler når $\mathcal{D}$ er snittfri og Γ er tom i konklusjonen. Siden ingen av de ikkelogiske slutningsreglene i PRA fører til forgreninger i bevistreet gir båndet på høyden av treet også et bånd på størrelsen.

Predikatene i formlene er enten N eller $=$, det vil si de er enten unære eller binære. Ved lemma 8.7 inneholder formlene i treet kun funksjonstermer som inngår i definisjonen av f. Det inngår kun et endelig antall funksjoner i definisjonen av f. Gitt definisjonen av disse funksjonene kan man finne hvilken som har størst aritet, heretter kalt k. Det er også nødvendig å finne et bånd på dybden til funksjonstermene i bevistreet. Den eneste regelen som øker dybden på et funksjonsuttrykk er *Eqf*. Denne kan maksimalt være brukt d ganger. Hvis for eksempel siste regel i treet er *fcomp*:

$$\frac{\begin{array}{c}\mathcal{D}_0 \\ f(\bar{l},\bar{m}) = h(g_1(\bar{l},\bar{m}),\ldots,g_m(\bar{l},\bar{m})) \Rightarrow N(f(\bar{l},\bar{m}))\end{array}}{\Rightarrow N(f(\bar{l},\bar{m}))}\, fcomp$$

og regelen *Eqf* er brukt i resten av treet vil hovedformelen i toppnoden inneholde en funksjonsterm med dybde d,

$$\underbrace{j(i(\ldots(h(g_1}_{d}(\bar{l},\bar{m}),\ldots,g_m(\bar{l},\bar{m})),\ldots))).$$

En term med dybde d hvor hver funksjonsterm har aritet k inneholder $k^{d-1}+1$ funksjonssymboler og k^d konstanter. Hvis man gir hver formel følgende tak på antall symboler,

$$\begin{aligned}&1 \text{ predikat } + \\ &2 \text{ funksjonstermer } \times (k^d \text{ konstanter } + (k^{d-1}+1) \text{ funksjonssymboler}) \\ &\leq 1 + 2(k^{\sigma(l,m)} + (k^{\sigma(l,m)}+1)),\end{aligned}$$

får man et elementært bånd på antall symboler i hver formel i treet med god margin.

Hver ikkelogisk slutningsregel legger maksimalt til en atomær formel i Γ, så antall formler i Γ er mindre eller lik antall skritt i treet. Dermed får man følgende bånd på antall symboler i bevistreet:

$$\begin{aligned}&\sigma(l,m) \times (1 + 2(k^{\sigma(l,m)} + (k^{\sigma(l,m)} + 1)))\\ &\leq \sigma(l,m)(4k^{\sigma(l,m)} + 3).\end{aligned}$$

Jeg trenger også et bånd på kodetallet for hver formel. Ved lemma 8.7 er det største tallet som beskrives i $\mathcal{D}$ mindre eller lik det maksimale av argumentene til f og høyden av utledningen, $\max(l, m, |\mathcal{D}|)$. La p kode termen som korresponderer til det største tallet. Det høyeste kodetallet en formel kan ha blir da:

$$\begin{aligned}&\ulcorner = \urcorner + 2(pk^d + \ulcorner f \urcorner k^{d-1} + \ulcorner f \urcorner)\\ &\leq 29 + 2(pk^{\sigma(l,m)} + \ulcorner f \urcorner k^{\sigma(l,m)} + \ulcorner f \urcorner).\end{aligned}$$

Kodetallet for $\mathcal{D}$ er mindre eller lik antall symboler ganget med maksimalt kodetall på en formel, det vil si:

$$\begin{aligned}\sigma'(l,m) =&(\sigma(l,m)(4k^{\sigma(l,m)} + 3)) \times\\ &(29 + 2(pk^{\sigma(l,m)} + \ulcorner f \urcorner k^{\sigma(l,m)} + \ulcorner f \urcorner))\end{aligned}$$

Funksjonen σ' er komponert over elementære funksjoner og σ som er i $\mathcal{E}^r$, så σ' er i $\mathcal{E}^r$. □

8.3 Algoritme for å beregne induktive funksjoner

Fra lemmaene over og resultater beskrevet tidligere i oppgaven følger hovedresultatet i dette kapitlet:

Teorem 8.10 *Hvis f er induktiv i* PRA*, så er f elementær.*

Bevis. Anta f er induktiv. Da finnes en utledning $\mathcal{D}$ i PRA med fast snittkompleksitet k og en funksjon σ slik at:

$$\mathcal{D} \vdash^{\mathrm{PRA}}_{\sigma(m_1,\ldots,m_n)} \Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$$

hvor σ er lineær i $m_1, \ldots, m_n$.

Ved snitteliminasjonsteoremet 2.42 finnes en snittfri utledning $\mathcal{D}'$, slik at $|\mathcal{D}'| \leq \beth^k(|\mathcal{D}|) \leq \beth^k(\sigma(m_1, \ldots, m_n))$. Funksjonen $\beth^k(n)$ (definisjon 2.24) gir

$$
\begin{aligned}
\alpha(\vec{m}) =& \mathit{if}\ \mathit{Canonic}((\mathcal{U}_f(\vec{m}))_{1,4,2,2,1}) \\
& \mathit{then}\ \ulcorner((\mathcal{U}_f(\vec{m}))_{1,4,2,2,1})\urcorner \\
& \mathit{else}\ \beta(\gamma(\vec{m}), \mathcal{U}_f(\vec{m})) \\
\beta(x,y) =& \sum_{i=2}^{ln((x)_2)} \mathit{if}\ (x)_{2,i,1} = 29 \wedge (x)_{2,i,2} = (y)_{1,4,2,2,1} \wedge \mathit{Canonic}((x)_{2,i,3}) \\
& \mathit{then}\ \ulcorner(x)_{2,i,3}\urcorner \\
& \mathit{else}\ 0 \\
\gamma(\vec{m}) =& \exp(\theta(\delta(\vec{m}), \mathcal{U}_f(\vec{m})), p_1) \\
\delta(\vec{m}) =& \sum_{i=0}^{\sigma'(\vec{m})} \mathcal{T}_{nff}(\theta(i, \mathcal{U}_f(\vec{m})) \\
\theta(0,x) =& x \\
\theta(Sy,x) =& \exp(\theta(y,x), p_2)
\end{aligned}
$$

Figur 8.2: Definisjon av funksjonen α

et tårn av toere, hvor eksponensieringen gjøres den "vrange veien". Ved lemma 2.28 er $\beth^k(\sigma(m_1,\ldots,m_n)) \in \mathcal{E}^3$. Ved korrolar 8.4 finnes en utledning $\mathcal{D}''$ på normalform, slik at $|\mathcal{D}''| \leq 2|\mathcal{D}'| \leq 2 \times \beth^k(\sigma(m_1,\ldots,m_n))$. Siden multiplikasjonsfunksjonen også er elementær får jeg at $|\mathcal{D}''|$ er begrenset av en elementær funksjon. Ved lemma 8.9 finnes en funksjon σ' slik at $\sigma'(m_1,\ldots,m_n)$ legger et øvre bånd på kodetallet for $\mathcal{D}''$. De elementære funksjonene er lukket under bundet minimalisering. Videre er predikatet $\mathcal{T}_{nff}$ elementært ved korrolar 8.5. Dermed følger det at følgende funksjon, som returnerer kodetallet for en snittfri utledning på normalform i PRA, er i $\mathcal{E}^3$:

$$\mathcal{U}_f(\vec{m}) = \mu_{B \leq \sigma'(\vec{m})}[\mathcal{T}_{nff}(y,\vec{m}) = 1]$$

La $k = \mathcal{U}_f(\vec{m})$. Siden k koder en utledning i PRA på normalform, er enten $f(\bar{m}_1,\ldots,\bar{m}_n)$ en term på kanonisk form, eller så inneholder antecedenten i toppsekventen termen $f(\bar{m}_1,\ldots,\bar{m}_n) = \bar{n}$, for en tallterm $\bar{n}$. Dermed kan $f(m_1,\ldots,m_n)$ plukkes ut fra k av den elementære funksjonen α, se figur 8.2. Den sjekker om $f(\bar{m}_1,\ldots,\bar{m}_n)$ er på kanonisk form i en utledning av $N(f(\bar{m}_1,\ldots,\bar{m}_n))$. I så fall returnerer den $f(\bar{m}_1,\ldots,\bar{m}_n)$. Ellers leter den etter en term på formen $f(\bar{m}_1,\ldots,\bar{m}_n) = \bar{n}$ i Γ i toppnoden og returnerer $\bar{n}$. Hjelpefunksjonen β leter etter en term på formen $f(\bar{m}_1,\ldots,\bar{m}_n) = \bar{n}$, i en liste av termer kodet som et sekvenstall. $\gamma(\vec{m})$ finner toppnoden i $\mathcal{U}_f(\vec{m})$. Funksjonen $\delta(\vec{m})$ gir høyden til $\mathcal{U}_f(\vec{m})$, når den er begrenset av $\sigma'(\vec{m})$. Funksjonen $\theta(k,x) = (x)_{2,\ldots,2}$. Det vil si den plukker ut elementet med koordinat

$2, 2, 2, \ldots, k$ ganger, fra sekvenstallet x. Funksjonen p_n definert i avsnitt 2.1.2 gir det n'te primtallet.

Funksjonen α er definert ved komposisjon over elementære funksjoner, predikater og bundet produkt, så α er elementær.

Alle operasjonene beskrevet over er elementære. Dermed har jeg en elementær algoritme for å beregne $f(m_1, \ldots, m_n)$ når f er induktiv. □

Korrolar 8.11 (Korrolar til påstand 5.10 og teorem 8.10) *La f være definert ved primitiv rekursjon over g og h:*

$$f(y, 0) = g(y)$$
$$f(y, Sx) = h(y, f(y, x))$$

La g være en av initialfunksjonene $\mathcal{O}, \mathcal{S}$ eller $\mathcal{I}_i^n$. La h være assosiativ og la bevissystemet inneholde følgende regel:

$$\frac{h(h(a,b),c) = h(a,h(b,c)), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ Ass_h$$

La H være et induktivt predikat for h

$$H(x) = G(x) \wedge \forall y(G(y) \supset G(h(x,y))$$

Da er F elementær.

Bevis. La funksjonen f være gitt som i korrolaret over. Ved påstand 5.10 er f induktiv i PRA $+Ass_h$. Ved teorem 8.10 er f elementær. □

Kapittel 9

Snittsøk integrert i en automatisk teorembeviser

I kapitler 5 og 6 gis en strategi for å søke etter snittformler i utledninger av utsagn om elementære funksjoner. Jeg har implementert strategien i det funksjonelle programmeringsspråket Haskell. Programmet er integrert i en utvidet versjon av den automatiske teorembeviseren PESCA, kalt PESCA*. PESCA—*A Proof Editor for Sequent Calculus* er en interaktiv teorembeviser med mulighet for automatisk bevissøk. Den er laget av Aarne Ranta som et supplement til boka *Structural Proof Theory* [22]. Boka inneholder en grundigere innføring i bruk av teorembeviseren.

I dette kapitlet gir jeg en oversikt over hvordan PESCA* er endret i forhold til den opprinnelige teorembeviseren for at den skal kunne gi støtte for snittsøk. Jeg presenterer også programmet `findpred` som søker etter snittformler for elementære strengt voksende funksjoner. Videre gir jeg eksempel på en sesjon hvor denne muligheten benyttes i PESCA*.

9.1 Valg av programmeringsspråk

Teorembeviseren PESCA er skrevet i Haskell og jeg har holdt meg til det samme språket for alle nye underfunksjoner. Da kunne jeg også benytte meg av det samme rammeverket for representasjon og parsing av formler.

Haskell er et rent, statisk typet og ikke-strengt (eng. non-strict) funksjonelt programmeringsspråk [27]. Det er oppkalt etter matematikeren Haskell Brooks Curry (1900-1982) som sammen med Alonzo Church la mye av det teoretiske grunnlaget for funksjonelle språk [11]. Funksjonelle programmeringsspråk kjennetegnes ved at de har høyere ordens funksjoner. Siden Haskell er et rent språk så består det i praksis kun av λ-kalkyle pluss syntaktisk

sukker.

Haskell kalles et *rent* funksjonelt språk fordi det ikke har noen observerbare sideeffekter [35], som for eksempel tilordninger. Haskell inneholder en slutningsmekansime for typer som innebærer av typen til variable og uttrykk er kjent før kjøring. Dette kalles *statisk typing. Ikke-streng* evaluering er en mellomting mellom streng og lat evaluering [36]. Lat evaluering innebærer at argumenter til en funksjon først evalueres når det er bruk for dem, mens streng evaluering innebærer at alle argumenter i et funksjonskall beregnes før funksjonen anvendes på dem.

Haskell har mye til felles med det funksjonelle programmeringsspråket ML (Meta Language) som ble utformet med det formål å implementere teorembevisere. En av de mer kjente såkalte taktiske teorembeviserene, Isabelle, er skrevet i ML.

9.2 PESCA*– en interaktiv teorembeviser

PESCA fungerer både som en bevis-editor og en automatisk teorembeviser. Jeg har blant annet utvidet PESCA med strukturelle regler. I den utvidede versjonen PESCA* kan bruker velge å la teorembeviseren søke etter snittformler i en utledning. Her gir jeg en oversikt hvilke endringer som er gjort for at PESCA* skal gi støtte for snittsøk. PESCA utgjør 1427 linjer kildekode fordelt på ni moduler. PESCA* er på 2253 linjer kildekode. I tillegg til de nye linjene med kildekode er omtrent 100 linjer endret i PESCA* i forhold til PESCA[1]. PESCA* har to nye moduler. Modulen `CutAlgorithm.hs` inneholder programmet `findpred` som søker etter snittformler. Modulen `GenLproof.hs` inneholder et program som kan generere utledninger i PESCA* direkte og skrive dem til en LaTeX-fil, uten å gå veien om en sesjon i PESCA*.

9.2.1 Tillegg av strukturelle regler

Brukeren av PESCA kan velge mellom ulike intuisjonistiske og klassiske kalkyler som kan utvides med ikkelogiske regler. Felles for kalkylene som behandles i PESCA er at reglene er utformet på en måte som gjør at premisset er bestemt når konklusjonen og regelen er gitt. Dette kalles **topp-ned determinisme**. Kalkylene har delt kontekst og de inneholder ikke strukturelle regler, slik som svekking og snitt. Jeg bruker en kalkyle (**G3c**+Cut) som inneholder snitt. Poenget i oppgaven er nettopp å vise at man kan oppnå korte

[1] Dette tallet er kommet fram ved å kjøre diff på de to katalogene, telle antall ! i diff og dele på 2. Kommandoen diff sammenligner to og to filer og viser endringer ved å sette ! foran en linje som er endret.

utledninger av visse utsagn, ved å introdusere snitt. Jeg har derfor lagt til *Cut*-regelen.

En logisk regel i PESCA er en funksjon som tar en konklusjon-sekvent som sitt argument og returnerer enten en liste av premisser eller ingenting. Strukturelle regler som *Cut*-regelen er av en annen type i Haskell-forstand enn logiske regler, fordi de i tillegg tar en formel som argument. Dette kan håndteres på flere måter. Jeg valgte å legge til to varianter av *Cut*-regelen. En av samme type som de logiske reglene. Dette for å få med *Cut*-regelen i lista over mulige regler som bruker kan velge mellom. I tillegg er det lagt til en *Cut*-regel som tar en formel som argument. Det er denne som til syvende og sist brukes når *Cut*-regelen skal implementeres i en utledning. Med denne løsningen blir *Cut*-regelen behandlet som et spesialtilfelle alle steder hvor regler anvendes.

Som nevnt innledningsvis kan utledninger i sekventkalkyle raskt ta stor plass og egner seg dårlig for gjengiving i A4-format. Når man skriver en utledning for hånd kan dette problemet løses ved å la Γ representere passive formler i antecedenten. For å kunne generere utledninger i PESCA* som får plass på A4-ark har jeg lagt til regelen for venstre-svekking, *LW*. Den er med andre ord tatt med kun av plasshensyn. Dette er uproblematisk siden teorem 2.35 gir at *LW*-regler kan elimineres fra en utledning uten at dette får konsekvenser for høyden.

9.2.2 Utvidelse med ikkelogiske regler

Som beskrevet i kapittel 3 har Negri og von Plato [22] utviklet en metode for å utvide kalkyler med ikkelogiske regler. De viser at en klassisk teori som aksiomatiseres av universal-aksiomer (avsnitt 3.1) kan overføres til et regel-system med full snitt-eliminasjon.

PESCA* gjør aksiomer om til ikkelogiske slutningsregler i tråd med metoden til Negri og von Plato. Reglene skrives til fil på LaTeX-format. Tillegg C inneholder LaTeX-varianten av aksiomfila, figur 9.1, som brukes i kjøreeksemplet presentert her.

Reglene for likhet, `Ref` og `Repl`, definert i avsnitt 3.2 er med i PESCA, men fungerte ikke på utgivelsestidspunktet. Begge disse er med i PRA, så jeg måtte implementere dem i PESCA*.

I PESCA behandles sekventer som par av multimengder av formler. Det vil si lister med multiplisitet men hvor rekkefølgen ikke spiller noen rolle. Siden reglene er definert slik at formlene lengst til venstre i sekventen er de aktive må derfor bruker ideelt sett kunne velge mellom alle mulige permutasjoner av sekventene.

I PESCA vises kun tilgjengelige regler hvor første formel i en sekvent

byttes ut. I regler hvor to eller flere, si n, formler er aktive i Γ må imidlertid bruker kunne velge mellom flere permutasjoner. Problemet er at hvis Γ er stor kan det fort bli veldig mange, antall permutasjoner av $n = n!$. I praksis vil det inngå maksimum to formler i en regel. Jeg har derfor endret PESCA* slik at den viser regler med permutasjoner på de to første plassene i Γ.

For å få PESCA* til å lese aksiomer med prefiks-notasjon for funksjoner og predikater måtte jeg endre parseren for aksiomfilene og lesing av input fra bruker. Parseren følger metoden for kombinator-parsing i Haskell som ble introdusert av Hutton og Meijer [12]. Med disse endringene kan jeg legge til aksiomer av typen vist i figur 9.1.

9.2.3 Integrering av snittsøk – funksjonen `findpred`

Algoritmen beskrevet i avsnitt 4 er integrert som en opsjon i PESCA*. Strengt tatt kan ikke PESCA* foreta et automatisk søk etter en utledning med snitt. Bruker må oppgi hvilke regler som skal introduseres når, men PESCA* finner snittformlene.

Den automatiske bevissøk-metoden i PESCA går ut på å erstatte målsekventen av den første utledningen som finnes ved et rekursivt forsøk på å applisere alle regler maksimalt et gitt antall ganger. Dette er en svært enkel søkemetode som sjelden fører fram i predikatlogikk med instansieringer.

For at PESCA* selv skal søke etter utledninger i sekventkalkyle med snitt, må en mer avansert søkestrategi legges til. Isabelle er en såkalt taktisk teorembeviser hvor man i større grad har mulighet for å spesialisere søkestrategier. Taktiske teorembevisere gjør en form for halvautomatisert bevissøk, hvor teorembeviseren kan søke ut i fra en spesialsydd taktikk.

Figur 6.9 gir en algoritme for å søke etter utledninger med snitt av utsagn på formen $\Rightarrow N(f(\bar{m}_1, \ldots, \bar{m}_n))$, når f navngir en strengt voksende elementær funksjon. Med litt jobb kan denne algoritmen trolig skrives som en taktikk i en mer avansert teorembeviser enn PESCA*.

Programmet `findpred` tar et funksjonsnavn, en assosiasjonsliste av funksjoner og en aksiomfil. Programmet har en semantisk begrensning: Hvis funksjonen f er definert ved primitiv rekursjon over en funksjon g må systemet inneholde aksiomer for assosiativitet av h. For øvrig genereres snittformlene ut i fra syntaktiske kriterier og har lik syntaktisk form. Snittformlene er bygd opp av induktive predikater, se definisjon 4.1. Det vil si hvis P er et induktivt predikat, så er snittformelen $P(0) \wedge \forall x(P(x) \supset P(Sx))$.

Assosiasjonslista er indeksert på navnene til funksjonene, "0", "S", "+" og så videre. Til hvert navn hører en liste over navnene på aksiomer om funksjonene som finnes i teorien representert ved en aksiomfil.

```
IINull      I0(a) = 0
III22       I22 a b = b
NNAdd       + 0 b =  b
RRAdd       + (S a) b = S(+ a b)
AAssAdd     + a (+ b c) = + (+ a b) c
NNTimes     * 0 b = 0
RRTimes     * (S a) b = +(* a b) b
AAssTimes   * a (* b c) = * (* a b) c
```

Figur 9.1: Utdrag fra aksiomfila

Aksiomfila består av aksiomene gitt i avsnitt 3.2, definerende likninger for en del elementære funksjoner og aksiomer om assosiativitet av addisjon og multiplikasjon. Det finnes uendelig mange elementære funksjoner, så en liste av aksiomer kan bare bestå av en endelig delmengde av dem. Det naturlige er å ha med de funksjonene som inngår i definisjonen av den funksjonen man vil finne predikatet for. Er det funksjonen $\times$ må man for eksempel ha med definerenede likninger for $+$ og et aksiom om assosiativitet av addisjon. Aksiomene skrives som vist i figur 9.1.

Første bokstav i navnet forteller hvilken type aksiom det dreier seg om[2]. Det finnes syv typer; initialfunksjon, funksjon definert ved komposisjon, ved skjema for primitiv rekursjon (basistilfelle og rekursjonstilfelle), ved skjema for bundet sum og produkt og aksiomer for assosiativitet.

Slik funksjonen `findpred` er definert gir den kun en delvis implementasjon av algoritmen for å søke etter snitt. For det ene tar den kun funksjoner med aritet mindre eller lik 2. For det andre er den definert for funksjoner komponert ved bundet produkt, men ikke bundet sum. Den kan generaliseres til å ta alle typer elementære funksjoner. Dette er først og fremst snakk om en del programmeringsarbeid og jeg har valgt ikke å bruke tid på det. Poenget er å illustrere at algoritmen beskrevet i oppgaven kan defineres ved hjelp av et funksjonelt program.

Funksjonen `findpred` sjekker først om en funksjon f identifisert ved en streng er en primitivt rekursiv initialfunksjon. I så fall tildeles dets respektive predikat gitt i avsnitt 5.1. Hvis f ikke er en initialfunksjon sjekker `findpred` om den er definert ved henholdsvis skjemaet for komposisjon, primitiv rekursjon eller bundet produkt. For hvert av tilfellene kalles underfunksjoner.

Underfunksjonene er direkte realiseringer av framgangsmåtene beskrevet i henholdsvis avsnitt 5.3 (rekursjon), avsnitt 5.2 (komposisjon) og 5.3.1

[2]Egenskaper ved parseren gjør at typeidentifikatoren gjentas to ganger.

(bundet produkt) for å tildele predikater til funksjoner definert ved de ulike skjemaene. Anta for eksempel at f er en binær funksjon definert ved skjema for primitiv rekursjon. Da kalles Haskell-funksjonen `rec`. Ved påstand 5.10 er f induktiv hvis h er assosiativ og induktiv. Funksjonen `rec` sjekker om h er assosiativ. I så fall gjøres et rekursivt kall på `findpred` for å tildele et predikat H til h. Dersom det lykkes å tildele et predikat til h får f et flatt predikat, se definisjon 5.9, på formen:

$$F(x) = H(x) \wedge \forall y(H(y) \supset H(f(y,x))),$$

ellers returneres ingenting og bruker får beskjed om at forsøket mislykkes. Her er eksempler på resultatet av å kalle `findpred` med elementære funksjoner definert ved henholdsvis komposisjon, primitiv rekursjon og bundet produkt. Kursiv viser input fra bruker. Eksemplet vises kalt med en pritty-print-funksjon for lesbarhetens skyld. Argument fra bruker skrives i kursiv.

```
CutAlgorithm>  prPred(findpred "SP" funalist pra_axioms)

"S = N(x)&N(SP x)"

CutAlgorithm> prPred(findpred "*" funalist pra_axioms)
"T = A(x)&(/Ay)(A(y)->A(*xy))"

CutAlgorithm> prPred(findpred "PiSP" funalist pra_axioms)
"P = S[T](x)&T(PiSP x)"
```

For å la bruker be PESCA* søke etter snitt trengtes noe I/O-håndtering som er lagt inn i hovedfunksjonen `editProofs`.

9.2.4 Eksempel på en sesjon i PESCA*

PESCA* har et grensesnitt for kommunikasjon med bruker. Når man starter PESCA* ser man promptet `|-`. En sesjon i PESCA* foregår i en omgivelse som endres som en funksjon av kommandoene fra bruker. Omgivelsen består av den gjeldende kalkylen og en gjeldende utledning. I starten er den intuisjonistiske predikatkalkylen **G3i** gjeldende kalkyle og målet er den tomme sekventen $\Rightarrow$.

```
|- c G3c + Geq + GStruct + GExp
```

endrer gjeldende kalkyle til den logiske delen av PRA. Siden en kalkyle kun er en mengde av regler kan de legges sammen ved operatoren $+$.

```
|- x numbers2
```

leser aksiomfila `numbers2`, parser den til regler, legger reglene til gjeldende kalkyle og skriver reglene til en fil, se appendiks C.

Man legger til et nytt mål med kommandoen `n` etterfulgt av sekventen skrevet i `ASCII`.

```
|- n => N(+ S0 SS0)
```

PESCA* svarer med et nytt bevistre. Foreløpig er konklusjonen det eneste delmålet i treet. Kommandoen `s` viser alle åpne delmål. Kommandoen `a` viser tilgjengelige regler for et gitt delmål. Foreløpig er Γ tom så det vises bare én versjon av hver regel. Merk at siden bevistreet kun inneholder en atomær formel er kun ikkelogiske regler og *Cut* tilgjengelig. Aksiomfila inneholder en god del aksiomer, så jeg viser bare et utdrag av dem her:

```
|- a 1
a 1

r 1 A0 S1 Ref --  => N(+ S0 SS0)
r 1 A0 S1 Cut --  => N(+ S0 SS0)
r 1 A0 S1 RExp --  => N(+ S0 SS0)
r 1 A0 S1 RNF --  => N(+ S0 SS0)
r 1 A0 S1 NN --  => N(+ S0 SS0)
```

Kommandoen `r` står for refine. Ved å velge en av reglene brytes gjeldende delmål ned i ett eller flere nye delmål. Når jeg velger snittregelen får jeg valget å oppgi snittformel eller å la PESCA* søke for meg. Hvis det aktuelle målet for bevissøket er på formen $\Gamma \Rightarrow N(f(m))$ vil funksjonen `findpred` kalles. Ellers gir PESCA* beskjed om at målsekventen ikke er av riktig type. Hvis søket lykkes skrives et nytt bevistre ut på skjermen hvor *Cut*-regelen er applisert på den aktuelle snittformelen. Ved teorem 5.13 er en funksjon f induktiv hvis den er elementær og strengt voksende. Algoritmen lykkes alltid når f navngir en strengt voksende elementær funksjon, så framt de nødvendige aksiomene er inkludert i aksiomfila og med forbehold om de begrensninger ved algoritmen som beskrives i avsnitt 4.

Jeg velger regelen *Cut* og ber PESCA* om å finne en snittformel:

```
|- r 1 A0 S1 Cut
r 1 A0 S1 Cut
Provide a cut formula or type s
to let pesca* search for you: s
s

 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
------------------------------------------------------------------------
 => N(+ S0 SS0)
```

PESCA* finner predikatet for addisjon og konstruerer en snittformel som settes inn i bevistreet. *Cut*-regelen fører til forgrening i bevistreet, som nå

har to nye delmål. Venstre grein vil tilsvare utledningen av at predikatet A er induktivt i figurer A.1, A.1 og A.3: Jeg velger å gå videre med høyre grein:

```
|- r 12 A1 S1 L&
                                  A(0), (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
                                  ----------------------------------------
 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
-------------------------------------------------------------------------
 => N(+ S0 SS0)

|- r 121 A2 S1 L/A
r 121 A2 S1 L/A

                  A(t) -> A(St), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                  ---------------------------------------------------------
                                (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                                ----------------------------------------
 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
-------------------------------------------------------------------------
 => N(+ S0 SS0)

Parameters t
```

I siste regel introduseres et parameter `t` som jeg instansierer med 0 ved følgende kommando:

```
|- i t 0
i t 0
                  A(0) -> A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                  ---------------------------------------------------------
                                (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                                ----------------------------------------
 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
-------------------------------------------------------------------------
 => N(+ S0 SS0)

|- r 1211 A1 S1 L->
r 1211 A1 S1 L->

(/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0), A(0)
                          A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
-------------------------------------------------------------------------
                  A(0) -> A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                  ---------------------------------------------------------
                                (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                                ----------------------------------------
 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
-------------------------------------------------------------------------
 => N(+ S0 SS0)
```

For å få plass må greinene etterhvert skrives oppå hverandre. Merk at den venstre greina nå har samme formel på begge sider av sekventpila. Nå kan regelen `ax` brukes til å lukke denne greina:

```
|- r 12111 A1 S1 ax
r 12111 A1 S1 ax

(/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0), A(0)
                          A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
-------------------------------------------------------------------------
                  A(0) -> A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                  ---------------------------------------------------------
                                (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                                 ---------------------------------------
 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
-------------------------------------------------------------------------
 => N(+ S0 SS0)
```

Høyre grein inneholder formelen $A(S0)$. Alle predikater P har en normalform P_n der definisjonen ikke inneholder noen andre predikater enn N, se avsnitt 5.3. PRA inneholder ingen regler for likhet mellom predikater, så i utledningene må alle predikatene være på normalform. Problemet er at trærne etterhvert ikke vil få plass på skjermen. $A(0)$ er metanotasjon for $N(0) \wedge \forall y(N(y) \supset N(+0y)$.

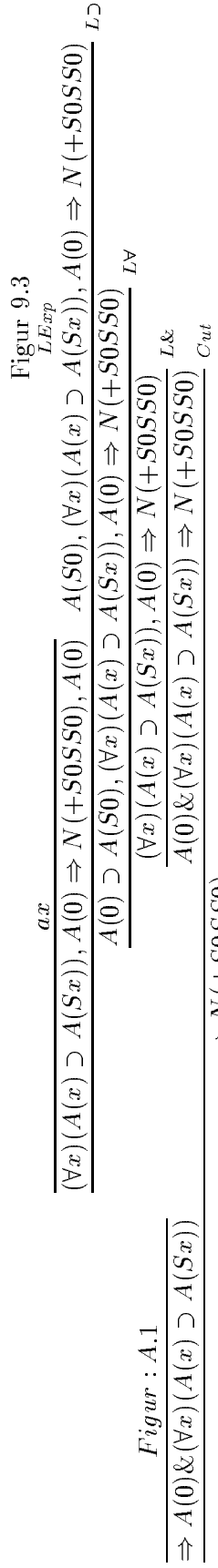
På dette punktet i utledningen er det nødvendig å skrive predikatet helt ut. Jeg har lagt til metareglene *LExp* og *RExp* som ekspanderer det aktive predikatet til normalform. Disse reglene er ikke en del av kalkylen. I en reell utledning vil predikatene være på normalform fra start.

```
|- r 12112 A1 S1 LExp
r 12112 A1 S1 LExp

  N(S0)&(/Ay)(N(y) -> N(+S0 y)), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
  ------------------------------------------------------------------------
(/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0), A(0)
                          A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
-------------------------------------------------------------------------
                  A(0) -> A(S0), (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                  ---------------------------------------------------------
                                (/Ay)(A(y) -> A(Sy)), A(0) => N(+ S0 SS0)
                                 ---------------------------------------
 => A(0) & (/Ay)(A(y) -> A(Sy)) A(0) & (/Ay)(A(y) -> A(Sy)) => N(+ S0 SS0)
-------------------------------------------------------------------------
 => N(+ S0 SS0)
```

Jeg viser ikke resten av sesjonen. En LaTeX-utskrift av den endelige utledningen av høyregreina vises i figur 9.2.

Kildekoden for PESCA* ligger på `http://folk.uio.no/gyrdb/`.



Figur 9.2: Addisjon med snitt

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\forall y)(N(y) \supset N(+S0y)), N(SS0), N(S0) \Rightarrow N(+S0SS0), N(SS0)}\,ax \quad \dfrac{}{N(+S0SS0), (\forall y)(N(y) \supset N(+S0y)), N(SS0), N(S0) \Rightarrow N(+S0SS0)}\,ax}{N(SS0) \supset N(+S0SS0), (\forall y)(N(y) \supset N(+S0y)), N(SS0), N(S0) \Rightarrow N(+S0SS0)}\,\supset_T}{(\forall y)(N(y) \supset N(+S0y)), N(SS0), N(S0) \Rightarrow N(+S0SS0)}\,\forall_T}{N(S0), (\forall y)(N(y) \supset N(+S0y)) \Rightarrow N(+S0SS0)}\,RN}{N(S0)\&(\forall y)(N(y) \supset N(+S0y)) \Rightarrow N(+S0SS0)}\,L\&}{N(S0)\&(\forall y)(N(y) \supset N(+S0y)), (\forall x)(A(x) \supset A(Sx)), A(0) \Rightarrow N(+S0SS0)}\,LW}{A(S0), (\forall x)(A(x) \supset A(Sx)), A(0) \Rightarrow N(+S0SS0)}\,LExp$$

Figur 9.3: Addisjon med snitt: Høyre side

Kapittel 10

Diskusjon

I dette kapitlet tar jeg opp et spørsmål som har kommet opp under presentasjoner av arbeidet med induktive predikater på logikkseminaret til UiO og i et foredrag på SINTEF. I avsnitt 10.2 forsøker jeg å knytte nyere relatert arbeid til teorien rundt induktive predikater. Til slutt peker jeg på en mulig retning for videre arbeid.

10.1 Kan predikatene brukes på ikke-trivielle spørsmål?

Et spørsmål som har blitt stilt ved ulike anledninger under presentasjoner av arbeidet med induktive predikater, er om de kan brukes til å korte ned bevis for ikke-trivielle påstander. Her er et eksempel sendt meg av professor Stål Aandera: Definer et utsagn $A(t, x, y)$ der t er en term som er et navn på et stort tall, x og y er tall. $A(t, x, y)$ er sant dersom $t = mod(x, y)$. Med andre ord: x er resten når t deles med y. I de tilfelle at dette utsagnet er sant, kan da snitt vesentlig forkorte beviset for at det er sant?

Svaret på dette eksemplet er så vidt jeg kan se negativt. Det gjelder også i det generelle tilfellet for problemer av typen skissert over. $A(t, x, y)$ kan oversettes til $t = mod(x, y)$. Denne påstanden kan beskrives av en karakteristisk funksjon med verdiområde $\{0, 1\}$. Dersom påstanden er gyldig gir kompletthetsteoremet (teorem 2.45) at $(t = mod(x, y)) = 1$ er utledbart, men det vil ikke finnes en kort utledning av uttrykket. I beste fall kan man finne en kort utledning av utsagnet $N((t = mod(m, n)))$, for alle t, m, n. Siden utsagnet $t = mod(m, n)$ har verdi 0 eller 1 vil påstanden alltid være sann, men det er jo ikke så veldig interessant. Poenget er nettopp at man ikke trenger å vite om uttrykket har verdi 0 eller 1, bare at det er et tall.

10.2 Relatert arbeid

Leivant [18] og Bellantoni og Cook [1] har kommet fram til rekursjonsteoretiske beskrivelser av de lavere kompleksitetsklassene etter rent syntaktiske kriterier. Dermed kan de kontrollere vekstraten til en funksjon uten å introdusere eksplisitte bånd. Et problem med Grzegorczyks originale hierarki er at hver klasse blant annet er kjennetegnet ved at funksjonene i den majoriseres av en bestemt funksjon. Videre er klassene lukket under begrenset rekursjon. Dette gir en semantisk begrensing på hver klasse.

Bellantoni og Cook innfører det de kaller *safe recursion* hvor det skilles mellom *normale* og *sikre* (safe) argumenter. Normale argumenter betraktes som kjent, mens sikre verdier oppnås ved rekursjon. Sikre argumentposisjoner beskrives som posisjoner hvor man kan substituere inn store verdier uten å øke størrelsen på resultatet av funksjonen dramatisk. Bellantoni og Cook definerer en klasse av funksjoner B hvor det ikke er tillatt å substituere rekursive termer inn i en posisjon som tidligere har vært brukt til en rekursiv definisjon. De viser at funksjonene beregnbare i polynomiell tid er nøyaktig de funksjonene i B som kun har normale argumenter.

Leivant innfører en alternativ måte å definere trygg rekursjon som han kaller lagdelt rekursjon. Han gir et rekursjonsskjema med to rekursjonsnivåer som skal sikre at rekursjonsargumentet til en funksjon er på et høyere nivå enn dens resultat. Leivant viser at en funksjon kan defineres ved lagdelt rekursjon hvis og bare hvis den er beregnbar i PTIME. Videre oppnår han en klassifisering av de primitivt rekursive funksjonene som skiller mellom polynomiell og eksponensiell vekst av funksjoner og er identisk med Grzegorczyks hierarki fra de elementære funksjonene og oppover. Leivant har også overført ideene til en teori for aritmetikk [19]. Ved hjelp av disse idéene gir han bevisteoretiske karakteristikker av PTIME og de lavere Grzegorczyk-klassene, $\mathcal{E}^2$ og $\mathcal{E}^3$.

Ostrin og Wainer [25] har nylig gitt en bevisteoretisk karakteristikk av de elementære funksjonene. Arbeidet deres er inspirert av skillet mellom sikre og normale variable, men de bruker andre metoder enn Leivant. Ostrin og Wainer bruker snitteliminasjon for å komme fram til sitt resultat. Deres metode likner mer på arbeidet til Jervell og Zhang [15], som denne oppgaven bygger videre på. En viktig forskjell er at både Leivant og Ostrin og Wainer bruker teorier som inneholder induksjonsaksiomet noe Jervell og Zhang ikke gjør.

Jervell og Zhang [15] har forsøkt å gi en bevisteoretisk karakteristikk av de elementære funksjonene ved hjelp av snitt. Målet er å vise at man kan finne korte bevis for at en funksjon f terminerer ved hjelp av snitt hvis og bare hvis f er elementær. Deres arbeid bygger videre på eksempler fra O-

revkov [24] og Zhang [34] på bevis med snitt for at eksponensialfunksjonen er veldefinert. Šereš [31] gir et liknende eksempel hvor snittformlene minner om induksjonshypoteser. Holden [10] generaliserte idéene til å gi en bevisteoretisk karakteristikk av alle de elementære funksjonene, men systemet hun bruker er ikke vanlig predikatlogikk. For å simulere induksjon introduserer hun formler med uendelige konnektiver. Jervell og Zhang har villet gjøre noe liknende i en klassisk logikk uten induksjonsaksiomer. De gir blant annet strategier for å konstruere snittformler for elementære funksjoner definert ved komposisjon og skjemaene for bundet sum og produkt.

Jervell og Zhang sammelikner sine induksjonsvariable med de normale variablene til Bellantoni og Cook [1]. Slik jeg ser det er det ingen direkte sammenheng mellom induktive predikater og Bellantoni og Cooks system for trygg rekursjon. De to metodene beskriver ulike kompleksitetsklasser. Jeg har vist at induktive predikater beskriver strengt voksende elementære funksjoner i tallteorien PRA$+AddAss, TimesAss$, mens Bellantoni og Cooks klasse B beskriver funksjonene i $PTIME$. Eksponensialfunksjonen er med andre ord ikke mulig å definere ved hjelp av skjemaet for trygg rekursjon. I systemet mitt følger jeg de vanlige definisjonsskjemaene for primitiv rekursjon hvor rekursjonsargumentet og det rekursive kallet begge alltid er på argumentplassen lengst til høyre. For å vise at multiplikasjon- og eksponensialfunksjonen er induktive må jeg ha med aksiomer om assosiativitet av addisjon og multiplikasjon, se avsnitt 5.3. Det er ikke tillatt å substituere inn rekursive termer på plasser som har vært brukt til en tidligere definisjon ved rekursjon i klassen B. Bellantoni og Cooks system rekurserer over binær notasjon, det vil si bit-strenger, istedenfor tall.

De induktive predikatene kan kanskje sammenliknes med Leivants nivåer [18], som minner om, men ikke er det samme som, trygg rekursjon. Ved å definere funksjonen $\times$ slik at det rekursive kallet ikke er på plassen til rekursjonsargumentet i pluss, kan man vise at $\times$ er på nivå N, men da klarer man ikke å vise at eksponensialfunksjonen er induktiv. Legger man til assosiativitet eller kommutativitet av multiplikasjon kan man vise at $\times$ også er i nivået over, A, og da kan man også vise at eksponensialfunksjonen er på nivå A.

10.3 Videre arbeid

Et opplagt spørsmål å jobbe videre med er hvorvidt man kan finne "lure" formuleringer av egenskaper ved funksjonen for modifisert subtraksjon som gjør det mulig å utlede et induktivt predikat for den. Som beskrevet i avsnitt 1.5 fører Jervell og Zhangs tilleggsaksiomer for modifisert subtraksjon til at for mange funksjoner kan bevises totale med korte bevis. Uten disse aksiomene

klarer man ikke å definere et induktivt predikat for modifisert subtraksjon og får dermed heller ikke en bevisteoretisk karakteristikk av de elementære funksjonene. Kanskje arbeidene til Leivant [19], Ostrin og Wainer [25] og Holden [10] kan gi en pekepinn til hvordan man kan gå fram for å utlede et induktivt predikat for modifisert subtraksjon. Et interessant spørsmål er om dette i det hele tatt er mulig i et såpass svakt system som PRA$+AddAss$,-$TimesAss$ uten noen form for induksjon, se neste avsnitt (10.3.1).

En eventuell sammenheng mellom induktive predikater og trygg rekursjon kan også være interessant og undersøke nærmere i et videre arbeid med disse spørsmålene.

Et annet uløst spørsmål i denne oppgaven er hva som kan være en passende betegnelse på den delmengden av de Kalmárelementære funksjonene jeg klarer å beskrive ved hjelp av induktive predikater. Som nevnt i kapittel 5 dekker betegnelsen strengt voksende for få, siden blant annet projeksjonsfunksjonen er induktiv. Betegnelsen monotont stigende favner om for mange, da det fins elementære monotone $\{0,1\}$-funksjoner som jeg ikke har induktive predikater for.

10.3.1 Alternativ håndtering av forgjengerfunksjonen

Jeg har gjort meg noen tanker om muligheten for å utlede et induktivt predikat for modifisert subtraksjon i en utvidelse av PRA$+AddAss,TimesAss$, som det kan passe å ta opp her. La f være definert ved et forenklet skjema for primitiv rekursjon:

$$f(x,0) = g(x)$$
$$f(x,Sy) = h(x,f(x,y))$$

Spørsmålet om hvorvidt

$$\forall x((F_1(x) \wedge \forall y(F_2(y) \Rightarrow F_3(f(y,x)))) \supset (F_1(Sx) \wedge \forall z(F_2(z) \supset F_3(f(z,Sx))))$$

er utledbart kan brytes ned til spørsmålet om

$$F_3(f(x,y)) \Rightarrow F_3(h(x,f(x,y)))$$

er utledbart. Som beskrevet i kapittel 4 kan induktive predikater betraktes som en måte å simulere induksjon på. Hvis man skal vise $N(\dot{-}(x,y)) \Rightarrow N(P(\dot{-}(x,y)))$ må man allerede ha vist $\forall x(N(x) \supset N(P(x))$, noe som ikke kan utledes i PRA$+AddAss,TimesAss$. At dette ikke er utledbart kan vises ved et vanlig ikkestandard-argument. Tilsvarende trenger man $\forall x(N(x) \supset N(\mathcal{S}(x))$ for å vise $N(+(x,y)) \Rightarrow N(S(+(x,y)))$. Men dette har man ved aksiom 5.1 og 5.2. Skal man vise $N(\times(x,y)) \Rightarrow N(+(x,times(x,y)))$ trenger

man noe sånt som $N(x) \wedge \forall y(N(y) \supset N(+(x,y)))$. Denne typen setninger klarer man å uttrykke ved induktive predikater. Det kan se ut som man trenger at funksjonen h som f rekurserer over har aritet større eller lik 2 og at den er kommutativ og kanskje assosiativ. Grunnen til at man klarer det for addisjon er at aksiomet $N(x) \supset N(Sx)$ er gitt av systemet.

I systemet for trygg rekursjon ville det å definere modifisert subtraksjon uten at forgjenger var en av initialfunksjonene trolig ikke la seg gjøre, siden rekursjonskallet må være på plassen til rekursjonsvariabelen i forgjenger. Bellantoni og Cook har løst dette problemet ved å gi forgjengerfunksjonen som en av initialfunksjonene hvor argumentet er på den trygge plassen.

Ut i fra disse betraktningene virker det usannsynlig at man klarer å utlede et induktivt predikat for modifisert subtraksjon uten å enten legge til induksjon eller legge til en regel som uttrykker

$$N(x) \supset N(P(x)). \tag{10.1}$$

Grunnen til at det ikke går å legge til den typen aksiomer som Jervell og Zhang gjør er som nevnt at man da får et system som er for uttrykkskraftig, fordi man har en alternativ måte å vise $N(\dot{-}(m,n))$ uten å regne ut uttrykket.

Jervell og Zhang har i sitt arbeid vært opptatt av at aksiomer som legges til systemet ikke må inkludere predikatet N. En innvending mot dette standpunktet kan være at man ved å legge til (10.1) får til en bevisteoretisk karakteristikk av de elementære funksjonene, uten å få et for uttrykkskraftig system.

Tillegg A

Beviser for de induktive predikatene

Figurer A.1 til A.22 viser utledninger av induktive predikater for funksjonene addisjon, multiplikasjon, fakultet og eksponensiering. Haskell-programmet som genererer snittformler for strengt voksende elementære funksjoner tar definisjonene vist i tillegg C som argument. Her er addisjon og multiplikasjon definert ved skjemaet for primitiv rekursjon, mens fakultet og eksponensialfunksjonen, kalt henholdsvis $PiSP$ og $PiI22$, er gitt noen litt omstendelige definisjoner ved hjelp av bundet produkt.

Utledninger i sekventkalkyle kan fort ta stor plass. Derfor har jeg nøyd meg med å gjengi deler av bevisene som er generert i PESCA*, mens repetisjoner av delbevis og overganger som jeg har vist er mulige, er angitt med metanotasjon.

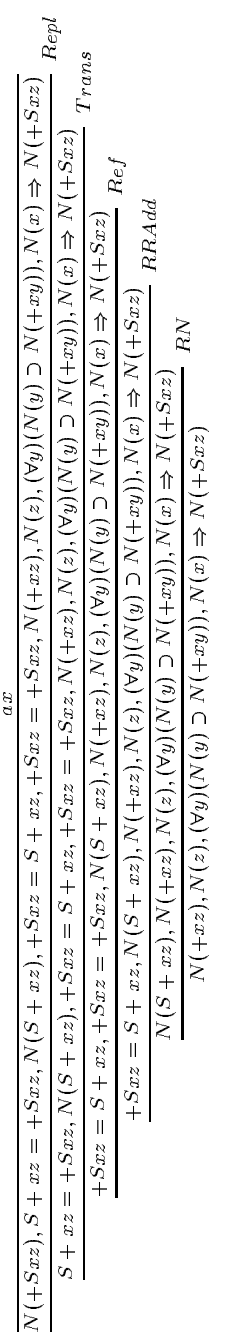
I kapitler 4, 5 og 6 har jeg fulgt det vanlige definisjonsskjemaet for primitiv rekursjon hvor rekursjonsargumentet og det rekursive kallet begge alltid er på argumentplassen lengst til høyre. I PESCA* brukes et definisjonsskjema hvor rekursjonsargumentet og det rekursive kallet er på argumentplassen lengst til venstre. Dette er fordi jeg i utgangspunktet brukte notasjon som lå tettere opp til Jervell og Zhang [15]. Så lenge dette gjøres konsekvent utgjør dette kun en syntaktisk forskjell og har ingenting å si for utledningene. Jeg har derfor valgt å ikke endre på dette i PESCA* eller i beviser generert i PESCA*.

$$\dfrac{\dfrac{\dfrac{}{N(0)\Rightarrow N(0)}\,ax}{\Rightarrow N(0)}\,NN \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{N(+0y),y=+0y,N(y),+0y=y\Rightarrow N(+0y)}\,ax}{y=+0y,N(y),+0y=y\Rightarrow N(+0y)}\,Repl}{+0y=y,N(y)\Rightarrow N(+0y)}\,Sym}{N(y)\Rightarrow N(+0y)}\,NNAdd}{\Rightarrow N(y)\supset N(+0y)}\,R\supset}{\Rightarrow (\forall y)(N(y)\supset N(+0y))}\,R\forall$$
$$\dfrac{\Rightarrow N(0)\&(\forall y)(N(y)\supset N(+0y))}{\Rightarrow A(0)}\,R\&\qquad\qquad \dfrac{}{\Rightarrow A(0)}\,REXP$$
$$\dfrac{}{\Rightarrow A(0)\&(\forall x)(A(x)\supset A(Sx))}\,Figur:A.2\;R\&$$

Figur A.1: Addisjon: rot

$$\dfrac{\dfrac{\dfrac{\dfrac{}{N(Sx),N(x),(\forall y)(N(y)\supset N(+xy))\Rightarrow N(Sx)}\,ax}{N(x),(\forall y)(N(y)\supset N(+xy))\Rightarrow N(Sx)}\,RN \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{N(z),(\forall y)(N(y)\supset N(+xy)),N(x)\Rightarrow N(+Sxz),N(z)}\,ax}{N(z)\supset N(+xz),N(z),(\forall y)(N(y)\supset N(+xy)),N(x)\Rightarrow N(+Sxz)}\,Figur:A.3\;L\supset}{N(z)\supset N(+xz),(\forall y)(N(y)\supset N(+xy)),N(x)\Rightarrow N(z)\supset N(+Sxz)}\,R\supset}{(\forall y)(N(y)\supset N(+xy)),N(x)\Rightarrow N(z)\supset N(+Sxz)}\,L\forall}{N(x),(\forall y)(N(y)\supset N(+xy))\Rightarrow (\forall z)(N(z)\supset N(+Sxz))}\,R\forall$$
$$\dfrac{N(x),(\forall y)(N(y)\supset N(+xy))\Rightarrow N(Sx)\&(\forall z)(N(z)\supset N(+Sxz))}{N(x)\&(\forall y)(N(y)\supset N(+xy))\Rightarrow N(Sx)\&(\forall z)(N(z)\supset N(+Sxz))}\,L\&$$
$$\dfrac{N(x)\&(\forall y)(N(y)\supset N(+xy))\Rightarrow N(Sx)\&(\forall z)(N(z)\supset N(+Sxz))}{\Rightarrow (\forall x)(N(x)\&(\forall y)(N(y)\supset N(+xy))\supset N(Sx)\&(\forall z)(N(z)\supset N(+Sxz)))}\,R\forall$$

Figur A.2: Addisjon (HS): rot

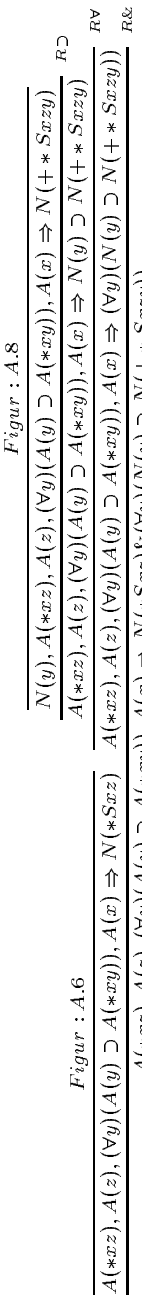


$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{N(+Sxz), S + xz = +Sxz, N(S + xz), +Sxz = S + xz, +Sxz = +Sxz, N(+xz), N(z), (\forall y)(N(y) \supset N(+xy)), N(x) \Rightarrow N(+Sxz)}\ Repl}{S + xz = +Sxz, N(S + xz), +Sxz = S + xz, +Sxz = +Sxz, N(+xz), N(z), (\forall y)(N(y) \supset N(+xy)), N(x) \Rightarrow N(+Sxz)}\ Trans}{+Sxz = S + xz, +Sxz = +Sxz, N(S + xz), N(+xz), N(z), (\forall y)(N(y) \supset N(+xy)), N(x) \Rightarrow N(+Sxz)}\ Ref}{+Sxz = S + xz, N(S + xz), N(+xz), N(z), (\forall y)(N(y) \supset N(+xy)), N(x) \Rightarrow N(+Sxz)}\ RRAdd}{N(S + xz), N(+xz), N(z), (\forall y)(N(y) \supset N(+xy)), N(x) \Rightarrow N(+Sxz)}\ RN}{N(+xz), N(z), (\forall y)(N(y) \supset N(+xy)), N(x) \Rightarrow N(+Sxz)}\ ax$$

Figur A.3: Addisjon (HS): 2

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A(z),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow A(*Sxz),A(z)}\,ax}{A(z)\supset A(*xz),A(z),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow A(*Sxz)}\,Figur:A.5\;L\supset}{A(z)\supset A(*xz),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow A(z)\supset A(*Sxz)}\;R\supset}{(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow A(z)\supset A(*Sxz)}\;L\forall}{A(x),(\forall y)(A(y)\supset A(*xy))\Rightarrow(\forall z)(A(z)\supset A(*Sxz))}\;R\forall \qquad \begin{array}{c}X\\ \vdots\\ A(x),(\forall y)(A(y)\supset A(*xy))\Rightarrow A(Sx)\end{array}}{A(x),(\forall y)(A(y)\supset A(*xy))\Rightarrow A(Sx)\&(\forall z)(A(z)\supset A(*Sxz))}\;R\&$$
$$\dfrac{A(x),(\forall y)(A(y)\supset A(*xy))\Rightarrow A(Sx)\&(\forall z)(A(z)\supset A(*Sxz))}{A(x)\&(\forall y)(A(y)\supset A(*xy))\Rightarrow A(Sx)\&(\forall z)(A(z)\supset A(*Sxz))}\;L\&$$
$$\dfrac{A(x)\&(\forall y)(A(y)\supset A(*xy))\Rightarrow A(Sx)\&(\forall z)(A(z)\supset A(*Sxz))}{\Rightarrow A(x)\&(\forall y)(A(y)\supset A(*xy))\supset A(Sx)\&(\forall z)(A(z)\supset A(*Sxz))}\;R\supset$$
$$\dfrac{\Rightarrow A(x)\&(\forall y)(A(y)\supset A(*xy))\supset A(Sx)\&(\forall z)(A(z)\supset A(*Sxz))}{\Rightarrow(\forall x)(A(x)\&(\forall y)(A(y)\supset A(*xy))\supset A(Sx)\&(\forall z)(A(z)\supset A(*Sxz)))}\;R\forall$$

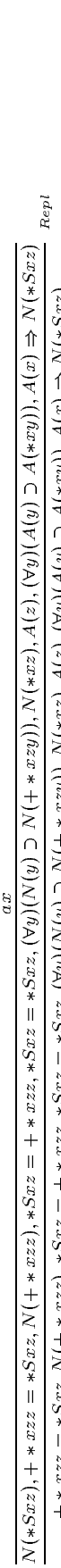
Figur A.4: Multiplikasjon



Figur A.5: Multiplikasjon: 2

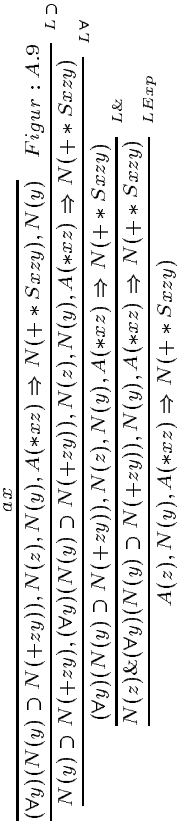
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{N(z),(\forall y)(N(y)\supset N(+zy)),(\forall y)(N(y)\supset N(+*xzy)),N(*xz),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow N(*Sxz),N(z)}\,{\scriptstyle L\&}}{N(z)\&(\forall y)(N(y)\supset N(+zy)),(\forall y)(N(y)\supset N(+*xzy)),N(*xz),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow N(*Sxz),N(z)}\,{\scriptstyle L\,Exp}}{A(z),(\forall y)(N(y)\supset N(+*xzy)),N(*xz),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow N(*Sxz),N(z)}\,Figur:A.7}{N(z)\supset N(+*xzz),(\forall y)(N(y)\supset N(+*xzy)),N(*xz),A(z),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow N(*Sxz)}\,{\scriptstyle L\supset}}{(\forall y)(N(y)\supset N(+*xzy)),N(*xz),A(z),(\forall y)(A(y)\supset A(*xy)),A(x)\Rightarrow N(*Sxz)}\,{\scriptstyle A7}$$

Figur A.6: Multiplikasjon: 2b



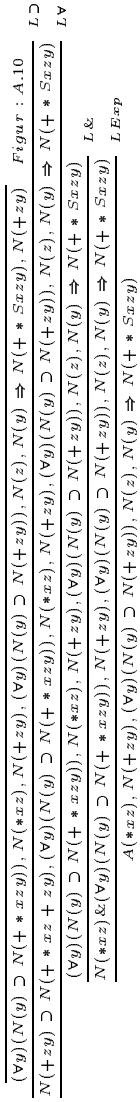
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{ax}{N(*Sxz), + * xzz = *Sxz, N(+ * xzz), *Sxz = + * xzz, *Sxz = *Sxz, (\forall y)(N(y) \supset N(+ * xzy)), N(*xz), A(z), (\forall y)(A(y) \supset A(*xy)), A(x) \Rightarrow N(*Sxz)}\,Repl}{+ * xzz = *Sxz, N(+ * xzz), *Sxz = + * xzz, *Sxz = *Sxz, (\forall y)(N(y) \supset N(+ * xzy)), N(*xz), A(z), (\forall y)(A(y) \supset A(*xy)), A(x) \Rightarrow N(*Sxz)}\,Trans}{*Sxz = + * xzz, *Sxz = *Sxz, N(+ * xzz), (\forall y)(N(y) \supset N(+ * xzy)), N(*xz), A(z), (\forall y)(A(y) \supset A(*xy)), A(x) \Rightarrow N(*Sxz)}\,Ref}{*Sxz = + * xzz, N(+ * xzz), (\forall y)(N(y) \supset N(+ * xzy)), N(*xz), A(z), (\forall y)(A(y) \supset A(*xy)), A(x) \Rightarrow N(*Sxz)}\,RRTimes}{N(+ * xzz), (\forall y)(N(y) \supset N(+ * xzy)), N(*xz), A(z), (\forall y)(A(y) \supset A(*xy)), A(x) \Rightarrow N(*Sxz)}$$

Figur A.7: Multiplikasjon (2b): 2



$$\dfrac{\dfrac{\dfrac{\dfrac{ax}{(\forall y)(N(y) \supset N(+zy)), N(z), N(y), A(*xz) \Rightarrow N(+*Sxzy), N(y)} \quad Figur : A.9}{N(y) \supset N(+zy), (\forall y)(N(y) \supset N(+zy)), N(z), N(y), A(*xz) \Rightarrow N(+*Sxzy)}\,{\scriptstyle \supset L}}{(\forall t)(N(y) \supset N(+zy)), N(z), N(y), A(*xz) \Rightarrow N(+*Sxzy)}\,{\scriptstyle \forall L}}{N(z) \& (\forall y)(N(y) \supset N(+zy)), N(y), A(*xz) \Rightarrow N(+*Sxzy)}\,{\scriptstyle \& L}}{A(z), N(y), A(*xz) \Rightarrow N(+*Sxzy)}\,{\scriptstyle LExp}$$

Figur A.8: Multiplikasjon: 3



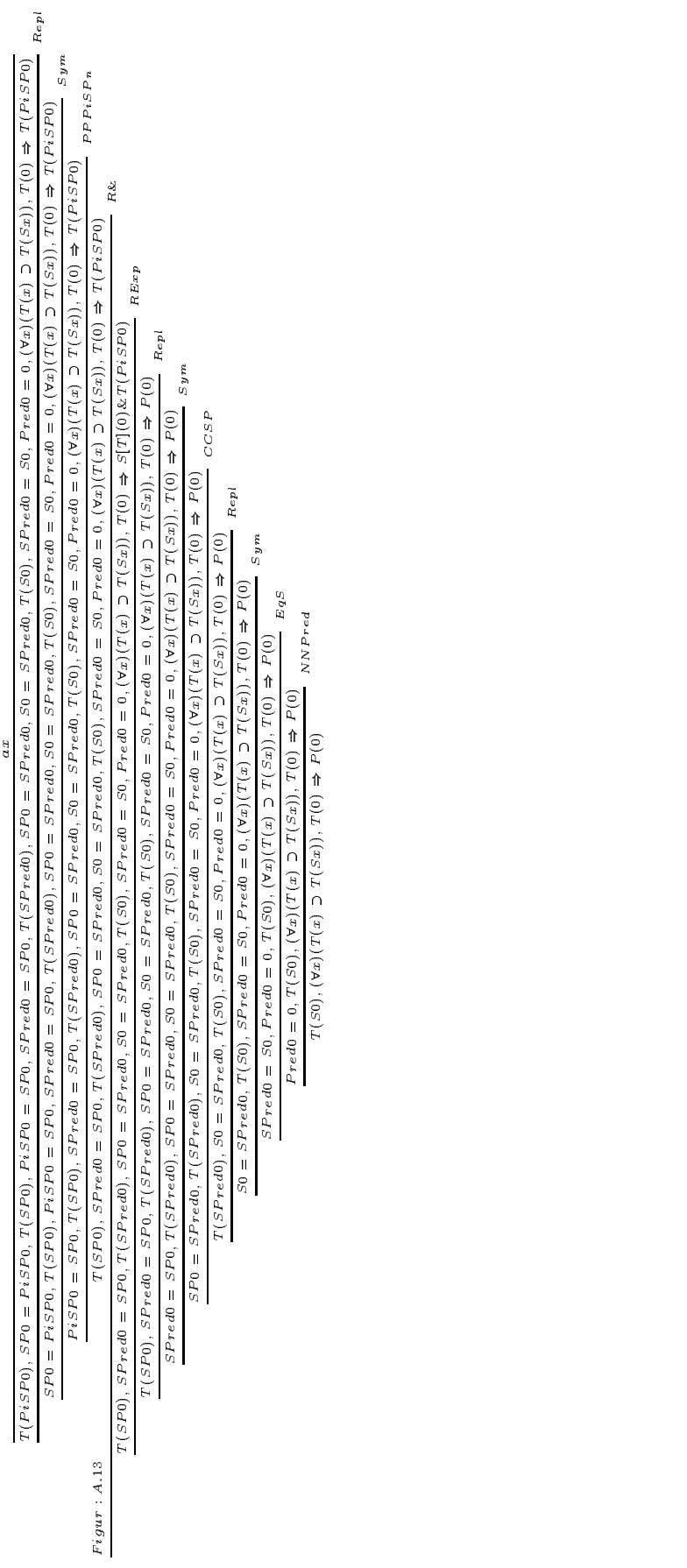
Figur A.9: Multiplikasjon: 4

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{ax}{N(++*Sxzy),+**xz+zy=++*Sxzy,N(+**xz+zy),+++*xzzy=+**xz+zy,++*xzzy=++*Sxzy,+**Sxzy=++*xzzy,*Sxz=+**xzz,y=y\Rightarrow N(+**Sxzy)}\,Repl}{++*xzzy=+**xz+zy,++*xzzy=++*Sxzy,+**Sxzy=++*xzzy,*Sxz=+**xzz,y=y\Rightarrow N(+**Sxzy)}\,Trans}{++*xzzy=+**xz+zy,++*xzzy=+**Sxzy,+**Sxzy=++*xzzy,*Sxz=+**xzz,y=y,N(+**xz+zy)\Rightarrow N(+**Sxzy)}\,Sym}{++*Sxzy=+++*xzzy,+++*xzzy=+**xz+zy,*Sxz=+**xzz,y=y,N(+**xz+zy)\Rightarrow N(+**Sxzy)}\,AAssAdd}{*Sxz=++*xzzy,*Sxz=+**xzz,y=y,N(+**xz+zy)\Rightarrow N(+*Sxzy)}\,Eq+}{*Sxz=+**xzz,y=y,N(+**xz+zy)\Rightarrow N(+*Sxzy)}\,Ref}{*Sxz=+**xzz,N(+**xz+zy)\Rightarrow N(+*Sxzy)}\,RRTimes}{N(+**xz+zy)\Rightarrow N(+*Sxzy)}$$

Figur A.10: Multiplikasjon: 5

$$\dfrac{\begin{array}{c}X\\ \vdots\\ \Rightarrow T(0)\&(\forall x)(T(x)\supset T(Sx))\end{array} \qquad \dfrac{\dfrac{\dfrac{\dfrac{ax}{(\forall x)(T(x)\supset T(Sx)),T(0)\Rightarrow P(0),T(0)}\quad Figur:A.12}{T(0)\supset T(S0),(\forall x)(T(x)\supset T(Sx)),T(0)\Rightarrow P(0)}\,L\supset}{(\forall x)(T(x)\supset T(Sx)),T(0)\Rightarrow P(0)}\,L\forall}{T(0)\&(\forall x)(T(x)\supset T(Sx))\Rightarrow P(0)}\,L\&}{\Rightarrow P(0)}\,Cut \qquad Figur:A.14}{\Rightarrow P(0)\&(\forall x)(P(x)\supset P(Sx))}\,R\&$$

Figur A.11: Fakultet



Figur : A.13

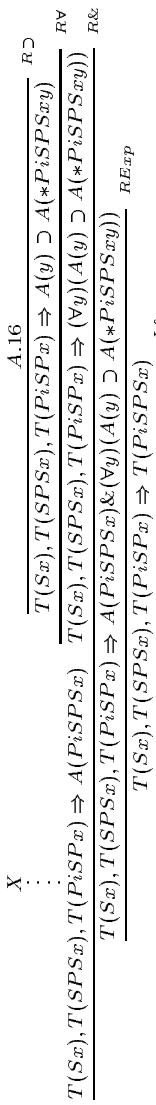
Figur A.12: Fakultet (VS): 2

$$\dfrac{\dfrac{\dfrac{\dfrac{ax}{T(SP0), T(0) \Rightarrow T(0)} \quad \dfrac{ax}{T(SP0), T(0) \Rightarrow T(SP0)}\, R\&}{T(SP0), T(0) \Rightarrow T(0)\&T(SP0)}\, RExp}{T(SP0), T(0) \Rightarrow S[T](0)}}{T(SP0), SPred0 = SP0, T(SPred0), SP0 = SPred0, S0 = SPred0, T(S0), SPred0 = S0, Pred0 = 0, (\forall x)(T(x) \supset T(Sx)), T(0) \Rightarrow S[T](0)}\, RW$$

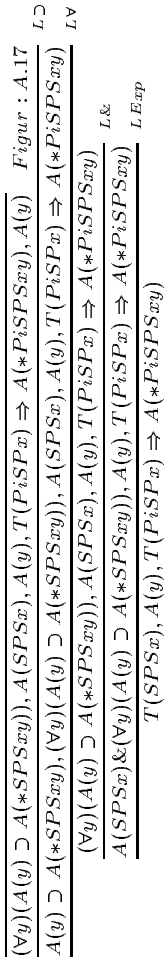
Figur A.13: Fakultet (VS): 3

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X}{\vdots}}{S[T](x), T(PiSPx) \Rightarrow S[T](Sx)} \quad \dfrac{A.15}{S[T](x), T(PiSPx) \Rightarrow T(PiSPSx)}\, R\&}{S[T](x), T(PiSPx) \Rightarrow S[T](Sx)\&T(PiSPSx)}}{S[T](x)\&T(PiSPx) \Rightarrow S[T](Sx)\&T(PiSPSx)}\, L\&}{\Rightarrow S[T](x)\&T(PiSPx) \supset S[T](Sx)\&T(PiSPSx)}\, R\supset}{\Rightarrow (\forall x)(S[T](x)\&T(PiSPx) \supset S[T](Sx)\&T(PiSPSx))}\, R\forall}{\Rightarrow (\forall x)(P(x) \supset P(Sx))}\, RExp$$

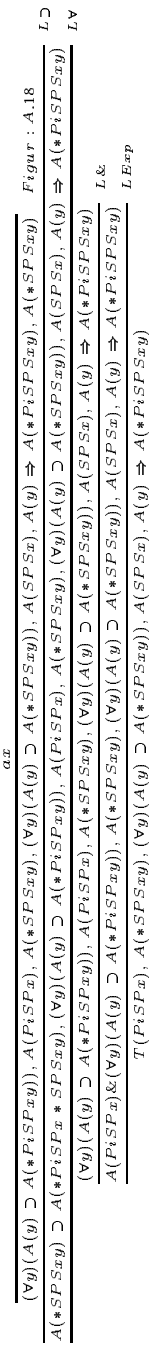
Figur A.14: Fakultet (HS): rot



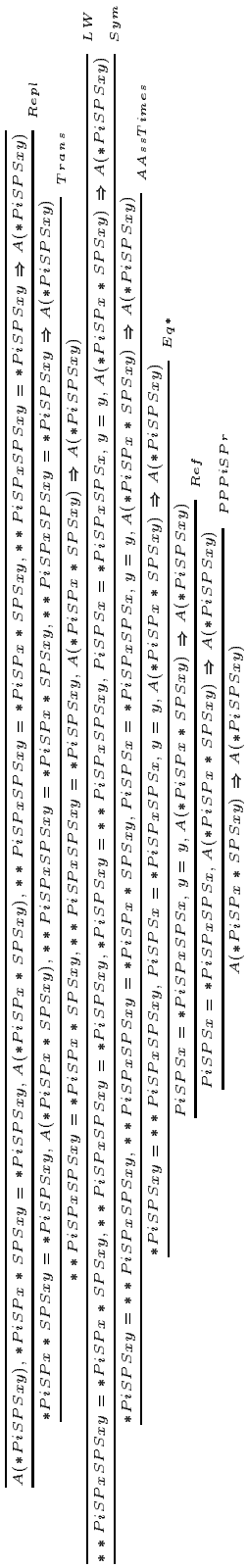
Figur A.15: Fakultet (HS): 2



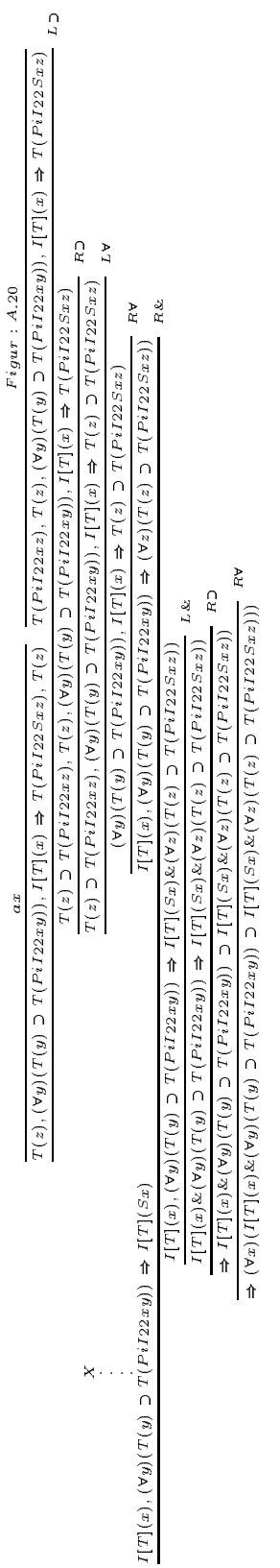
Figur A.16: Fakultet (HS): 3



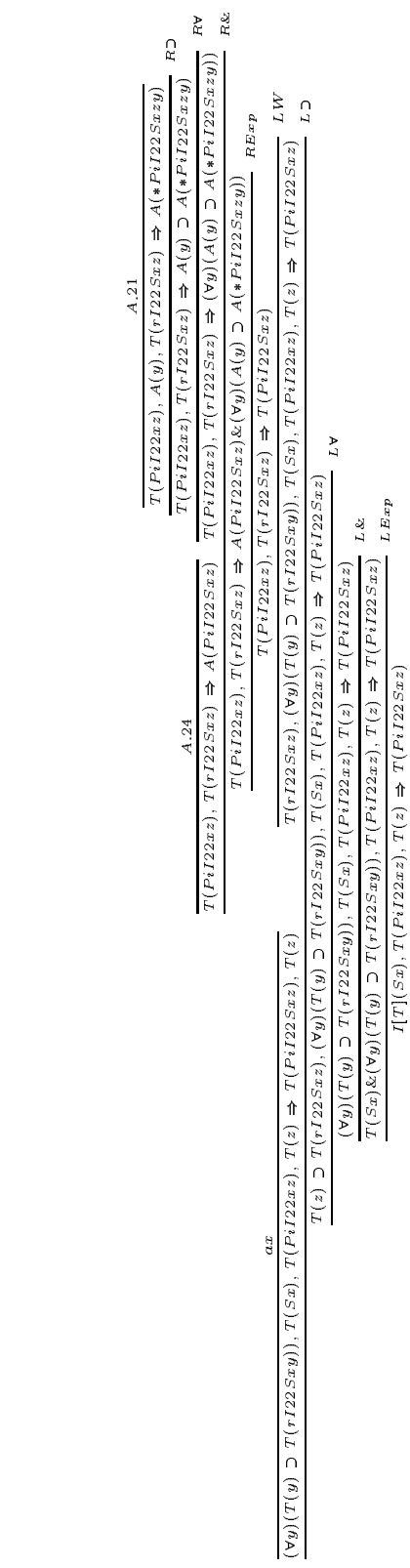
Figur A.17: Fakultet (HS): 4



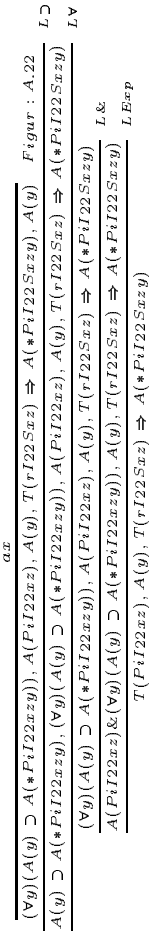
Figur A.18: Fakultet (HS): 5



Figur A.19: Eksponensialfunksjonen: Rot

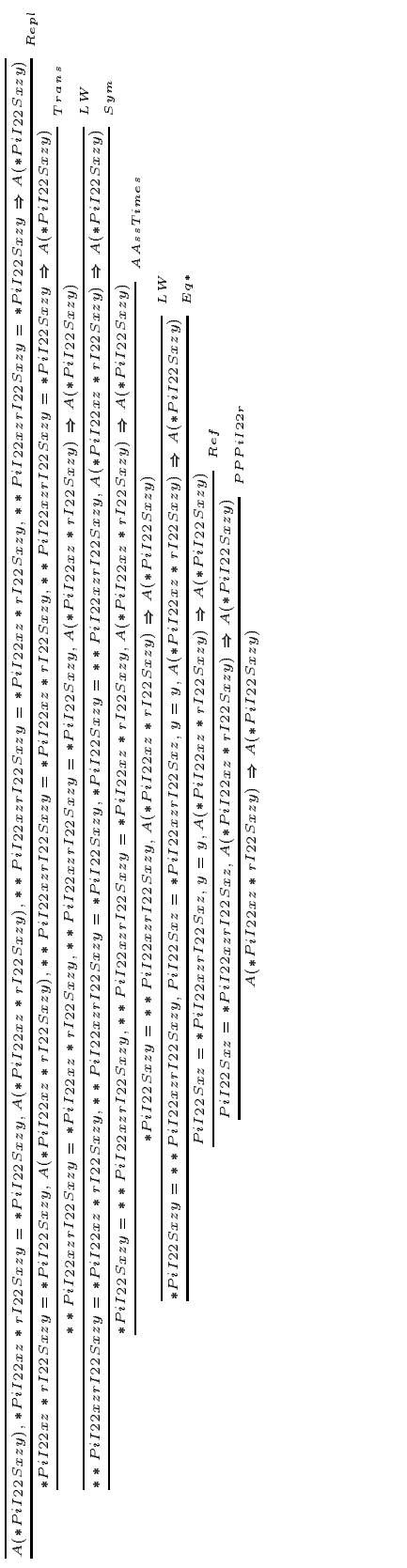


Figur A.20: Eksponensialfunksjonen: 2

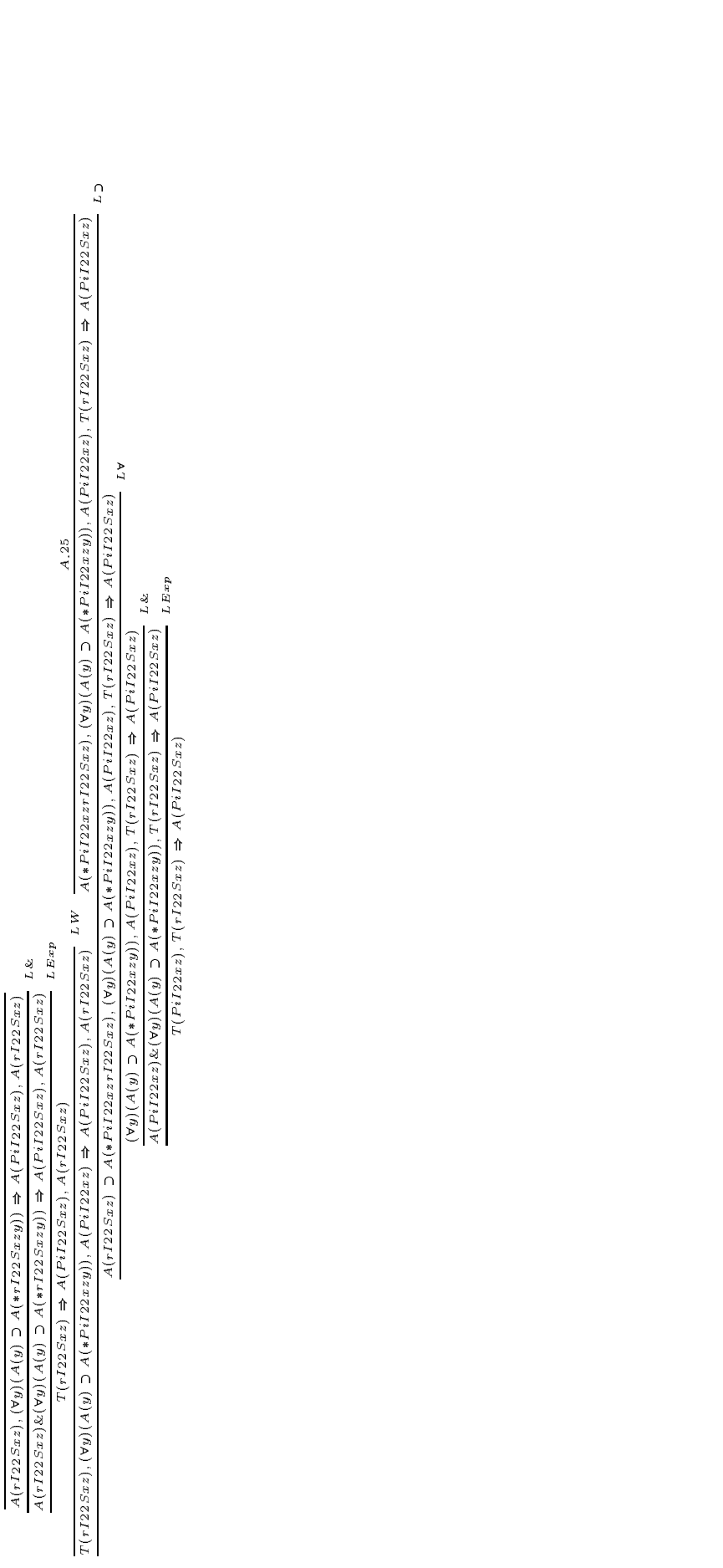


Figur A.21: Eksponensialfunksjonen: 3 (2 HS)

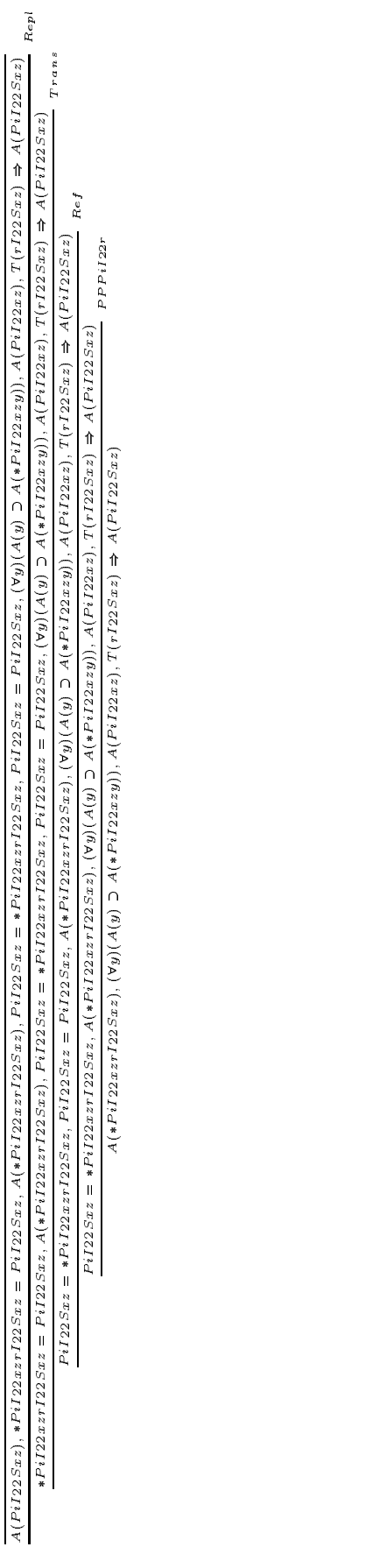
Figur A.22: Eksponensialfunksjonen: 4



Figur A.23: Eksponensialfunksjonen: 5



Figur A.24: Eksponensialfunksjonen: 2b (2 VS)



Figur A.25: Eksponensialfunksjonen: 2c (2b HS)

Tillegg B

Noen predikater til koding

Her defineres noen flere predikater for regler i PRA til kodesystemet i kapittel 7. Det finnes til sammen 29 typer regler i PRA. For de logiske slutningsreglene nøyer jeg meg med å angi predikater for noen typiske tilfeller. De resterende predikatene kan defineres på tilsvarende måte.

- *Axiom* og $L\bot$ er de eneste nodene uten forgjengere. Da er $r = (y)_1$ på formen:

  $$\langle\langle 7\rangle, \langle 1, \lceil P\rceil, \lceil C_1\rceil, \ldots, \lceil C_n\rceil\rangle, 3, \langle 2, \lceil D_1\rceil, \ldots, \lceil D_{m-1}\rceil, \lceil P\rceil\rangle\rangle$$

  hvor $n, m \geq 0$ og

  $$\langle\langle 18\rangle, \langle 1, 11, \lceil C_1\rceil, \ldots, \lceil C_n\rceil\rangle, 3, \lceil\Delta\rceil\rangle$$

  for henholdsvis *Axiom* og $L\bot$.

  La

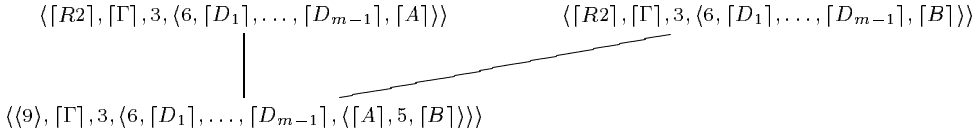
  $$\begin{aligned} Ax(y) \Leftrightarrow & ln(y) = 1 \wedge ln(y)_{1,1} = 4 \wedge (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{1,3} = 3 \wedge \\ & (y)_{1,1} = \langle 7\rangle \wedge Atomic(y)_{1,2,2} \wedge (y)_{1,2,2} = (y)_{4,ln(y)_4} \\ B(y) \Leftrightarrow & ln(y) = 1 \wedge ln(y)_{1,1} = 4 \wedge (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{1,3} = 3 \wedge \\ & (y)_{1,1} = \langle 18\rangle \wedge (y)_{1,2,2} = 11 \end{aligned}$$

- $L\&$

  $$\begin{array}{c} \langle\lceil R2\rceil, \langle 5, \lceil A\rceil, \lceil B\rceil, \lceil C_2\rceil, \ldots, \lceil C_n\rceil\rangle, 3, \lceil\Delta\rceil\rangle \\ \Big| \\ \langle\langle 8\rangle, \langle 5, \langle\lceil A\rceil, 5, \lceil B\rceil\rangle, \lceil C_2\rceil, \ldots, \lceil C_n\rceil\rangle, 3, \lceil\Delta\rceil\rangle \end{array}$$

$$\begin{aligned}LA(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\ & (y)_{1,1} = \langle 8 \rangle \wedge (y)_{1,2,2,2} = 5 \wedge (y)_{2,1,2,2} = (y)_{1,2,2,1} \\ & \wedge (y)_{2,1,2,3} = (y)_{1,2,2,3} \wedge (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \\ & \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\ & Eq(y, (y)_{1,2,3}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,4}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & ln(y)_{2,1,4} = ln(y)_{1,4} \\ & \wedge Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

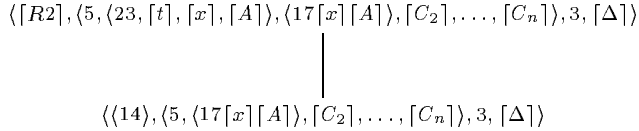
- $R\&$



$$\begin{aligned}RA(y) \Leftrightarrow & ln(y) = 3 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge ln(y)_{3,1} = 4 \\ & (y)_{1,1} = \langle 9 \rangle \wedge (y)_{1,4,ln(y)_{1,4},2} = 5 \wedge \\ & (y)_{2,1,4,ln(y)_{2,1,4}} = (y)_{1,4,ln(y)_{1,4},1} \wedge (y)_{3,1,4,ln(y)_{3,1,4}} = (y)_{1,4,ln(y)_{1,4},3} \wedge \\ & (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{3,1,3} = (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\ & ln(y)_{2,1,4} = ln(y)_{3,1,4} = ln(y)_{1,4} \wedge \\ & Eq(y, (y)_{1,4,1}, (y)_{1,4,(ln(y)_{1,4}-1)}, y, (y)_{2,1,4,1}, (y)_{2,1,4,(ln(y)_{2,1,4}-1)}) \wedge \\ & Eq(y, (y)_{1,4,1}, (y)_{1,4,(ln(y)_{1,4}-1)}, y, (y)_{3,1,4,1}, (y)_{3,1,4,(ln(y)_{3,1,4}-1)}) \wedge \\ & ln(y)_{2,1,2} = ln(y)_{3,1,2} = ln(y)_{1,2} \wedge \\ & Eq(y, (y)_{1,2,1}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,1}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & Eq(y, (y)_{1,2,1}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{3,1,2,1}, (y)_{3,1,2,ln(y)_{3,1,2}})\end{aligned}$$

Predikater for $L\vee, R\vee, L\supset, R\supset$ kan defineres på tilsvarende måte. For allkvantorreglene defineres følgende predikater:

- $L\forall$



$$
\begin{aligned}
LF(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 14 \rangle \wedge ln(y)_{1,2,2} = 3 \wedge (y)_{1,2,3,1} = 17 \wedge Term(y)_{1,2,2,2} \wedge \\
& ln(y)_{2,1,2,2} = 4 \wedge (y)_{2,1,2,2,1} = 23 \wedge \\
& Term(y)_{2,1,2,2,2} \wedge (y)_{2,1,2,2,3} = (y)_{1,2,2,2} \wedge (y)_{2,1,2,2,4} = (y)_{1,2,2,3} \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

- $R\forall$

$$
\begin{array}{c}
\langle \lceil R2 \rceil, \lceil \Gamma \rceil, 3, \langle 6, \lceil D_1 \rceil, \ldots, \lceil D_{m-1} \rceil, \langle 23, \lceil y \rceil, \lceil x \rceil, \lceil A \rceil \rangle \rangle \rangle \\
\Big| \\
\langle \langle 15 \rangle, \lceil \Gamma \rceil, 3, \langle 6, \lceil D_1 \rceil, \ldots, \lceil D_{m-1} \rceil, \langle 17 \lceil x \rceil \lceil A \rceil \rangle \rangle \rangle
\end{array}
$$

$$
\begin{aligned}
RF(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 15 \rangle \wedge ln(y)_{1,4,ln(y)_{1,4}} = 3 \wedge (y)_{1,4,ln(y)_{1,4},1} = 17 \wedge \\
& Term(y)_{1,4,ln(y)_{1,4},2} \wedge ln(y)_{2,1,4,ln(y)_{2,1,4}} = 4 \wedge (y)_{2,1,4,ln(y)_{2,1,4},1} = 23 \wedge \\
& Term(y)_{2,1,4,ln(y)_{2,1,4},2} \wedge (y)_{2,1,4,ln(y)_{2,1,4},3} = (y)_{1,4,ln(y)_{1,4},2} \wedge \\
& (y)_{2,1,4,ln(y)_{2,1,4},4} = (y)_{1,4,ln(y)_{1,4},3} \wedge \\
& (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} \wedge \\
& Eq(y, (y)_{1,2,1}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,1}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,(ln(y)_{1,4}-1)}, y, (y)_{2,1,4,1}, (y)_{2,1,4,(ln(y)_{2,1,4}-1)})
\end{aligned}
$$

Predikater for $L\exists$ og $R\exists$ defineres på tilsvarende måte. *Cut*-regelen må oppfylle predikatet:

- *Cut*

$$
\begin{array}{ccc}
\langle \lceil R2 \rceil, \lceil \Gamma \rceil, 3, \langle 6, \lceil D_1 \rceil, \ldots, \lceil D_m \rceil, \lceil A \rceil \rangle \rangle & & \langle \lceil R2 \rceil, \langle 5, \lceil A \rceil, \lceil C_1 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle \\
\Big| & \diagdown & \\
\langle \langle 19 \rangle, \lceil \Gamma \rceil, 3, \lceil \Delta \rceil \rangle & &
\end{array}
$$

$$
\begin{aligned}
C(y) \Leftrightarrow & ln(y) = 3 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge ln(y)_{3,1} = 4 \\
& (y)_{1,1} = \langle 19 \rangle \wedge (y)_{2,1,4,ln(y)_{2,1,4}} = (y)_{3,1,2,2} \wedge \\
& (y)_{1,2,1} = (y)_{3,1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{3,1,3} = (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{1,2} = ln(y)_{2,1,2} \wedge \\
& Eq(y, (y)_{1,2,1}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,1}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{1,4} = ln(y)_{3,1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{3,1,4,1}, (y)_{3,1,4,ln(y)_{3,1,4}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} + 1 \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,(ln(y)_{2,1,4}-1)}) \wedge \\
& ln(y)_{3,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,1}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{3,1,2,3}, (y)_{3,1,2,ln(y)_{3,1,2}})
\end{aligned}
$$

Jeg gir også predikater for noen av de ikkelogiske slutningsreglene i PRA:

- *Ref*

$$\frac{\langle \lceil R2 \rceil, \langle 5, \langle 29, \langle \lceil a \rceil, \lceil a \rceil \rangle \rangle, \lceil C_2 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}{\langle \langle 20 \rangle, \lceil \Gamma \rceil, 3, \lceil \Delta \rceil \rangle}$$

$$
\begin{aligned}
Ref(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 20 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 29 \wedge \\
& Term(y)_{2,1,2,2,2,1} \wedge (y)_{2,1,2,2,2,1} = (y)_{2,1,2,2,2,2} \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

- *Repl*

$$\frac{\langle \lceil R2 \rceil, \langle 5, \langle \lceil P \rceil, \langle \lceil b \rceil \rangle \rangle, \langle 29, \langle \lceil a \rceil, \lceil b \rceil \rangle \rangle, \langle \lceil P \rceil, \langle \lceil a \rceil \rangle \rangle, \lceil C_3 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}{\langle \langle 21 \rangle, \langle 5, \langle \lceil a \rceil, 29, \lceil b \rceil \rangle, \langle \lceil P \rceil, \langle \lceil a \rceil \rangle \rangle, \lceil C_3 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}$$

$$
\begin{aligned}
Repl(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 21 \rangle \wedge ln(y)_{1,2,2} = 2 \wedge (y)_{1,2,2,1} = 29 \wedge \\
& Term(y)_{1,2,2,2,1} \wedge Term(y)_{1,2,2,2,2} \wedge \\
& ln(y)_{2,1,2,2} = ln(y)_{1,2,3} = 2 \wedge Atomic(y)_{1,2,3} \wedge (y)_{1,2,2,2} = y_{1,2,3,2,1} \wedge \\
& (y)_{2,1,2,2,1} = (y)_{2,1,2,4,1} \wedge (y)_{2,1,2,2,2,1} = (y)_{2,1,2,3,2,2} \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

- *Trans*

$$
\frac{\langle \lceil R2 \rceil, \langle 5, \langle 29, \langle \lceil b \rceil, \lceil c \rceil \rangle \rangle, \langle 29, \langle \lceil a \rceil, \lceil b \rceil \rangle \rangle, \langle 29, \langle \lceil a \rceil, \lceil c \rceil \rangle \rangle, \lceil C_3 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}{\langle \langle 22 \rangle, \langle 5, \langle 29, \langle \lceil a \rceil, \lceil b \rceil \rangle \rangle, \langle \langle 29, \lceil a \rceil, \lceil c \rceil \rangle \rangle, \lceil C_3 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}
$$

$$
\begin{aligned}
Trans(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 22 \rangle \wedge ln(y)_{2,1,2,2} = ln(y)_{1,2,2} = ln(y)_{1,2,3} = 2 \wedge \\
& (y)_{2,1,2,2,1} = (y)_{1,2,2,1} = (y)_{1,2,3,1} = 29 \wedge \\
& Term(y)_{1,2,2,2,1} \wedge Term(y)_{1,2,2,2,2} \wedge Term(y)_{1,2,3,2,2} \wedge \\
& (y)_{1,2,2,2,1} = (y)_{1,2,3,2,1} \wedge (y)_{2,1,2,2,2,1} = (y)_{2,1,2,3,2,2} \wedge \\
& (y)_{2,1,2,2,2,2} = (y)_{2,1,2,4,2,2} \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

- *Sym*

$$
\frac{\langle \lceil R2 \rceil, \langle 5, \langle 29, \langle \lceil b \rceil, \lceil a \rceil \rangle \rangle, \langle 29, \langle \lceil a \rceil, \lceil b \rceil \rangle \rangle, \lceil C_2 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}{\langle \langle 23 \rangle, \langle 5, \langle 29, \langle \lceil a \rceil, \lceil b \rceil \rangle \rangle, \lceil C_2 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}
$$

$$\begin{aligned}Sym(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\ & (y)_{1,1} = \langle 23 \rangle \wedge ln(y)_{2,1,2,2} = ln(y)_{1,2,2} = 2 \wedge \\ & (y)_{2,1,2,2,1} = (y)_{1,2,2,1} = 29 \wedge \\ & Term(y)_{1,2,2,2,1} \wedge Term(y)_{1,2,2,2,2} \wedge \\ & (y)_{2,1,2,2,2,1} = (y)_{2,1,2,3,2,2} \wedge (y)_{2,1,2,2,2,2} = (y)_{2,1,2,3,2,1} \wedge \\ & (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\ & ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\ & Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\ & Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

- *NN*

$$\begin{array}{c}\langle \lceil R2 \rceil, \langle 5, \langle 31, \langle \langle 0 \rangle \rangle \rangle, \lceil C_2 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle \\ | \\ \langle \langle 24 \rangle, \lceil \Gamma \rceil 3, \lceil \Delta \rceil \rangle\end{array}$$

$$\begin{aligned}NN(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\ & (y)_{1,1} = \langle 24 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 31 \wedge \\ & ln(y)_{2,1,2,2,2} = 1 \wedge (y)_{2,1,2,2,2,1,1} = 0 \wedge \\ & (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\ & ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\ & Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\ & Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

- *RN*

$$\begin{array}{c}\langle \lceil R2 \rceil, \langle 5, \langle 31, \langle \langle 1 \rangle, \lceil a \rceil \rangle \rangle, \langle 31, \langle \lceil a \rceil \rangle \rangle, 3, \lceil \Delta \rceil \rangle \\ | \\ \langle \langle 25 \rangle, \langle 5, \langle 31, \langle \lceil a \rceil \rangle \rangle, \lceil C_2 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle\end{array}$$

$$\begin{aligned}RN(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \\ & \wedge (y)_{1,1} = \langle 25 \rangle \wedge ln(y)_{2,1,2,2} = ln(y)_{1,2,2} = 2 \\ & \wedge (y)_{2,1,2,2,1} = (y)_{1,2,2,1} = 31 \wedge Term(y)_{1,2,2,2,1} \\ & \wedge ln(y)_{2,1,2,2,2} = 2 \wedge (y)_{2,1,2,2,2,1} = \langle 1 \rangle \wedge (y)_{2,1,2,2,2,2} = (y)_{1,2,2,2,1} \\ & \wedge (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \\ & \wedge ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \\ & \wedge Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \\ & \wedge ln(y)_{2,1,4} = ln(y)_{1,4} \\ & \wedge Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

- $\mathcal{I}_i^n$

$$\frac{\langle \lceil R2 \rceil, \langle 5, \langle 29, \langle \langle \langle 2, n, i \rangle, \lceil a_1 \rceil, \ldots, \lceil a_n \rceil \rangle, \lceil a_i \rceil \rangle \rangle, \lceil C_2 \rceil, \ldots, \lceil C_n \rceil \rangle, 3, \lceil \Delta \rceil \rangle}{\langle \langle 28 \rangle, \lceil \Gamma \rceil, 3, \lceil \Delta \rceil \rangle}$$

$$\begin{aligned}\mathcal{I}_i^n(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\ & (y)_{1,1} = \langle 28 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 29 \wedge \\ & ln(y)_{2,1,2,2,2} = 2 \wedge ln(y)_{2,1,2,2,2,1,1} = 3 \wedge (y)_{2,1,2,2,2,1,1,1} = 2 \wedge \\ & (y)_{2,1,2,2,2,1,1,2} = ln(y)_{2,1,2,2,2,1} \dot{-} 1 \wedge 1 \leq (y)_{2,1,2,2,2,1,1,3} \\ & \leq (y)_{2,1,2,2,2,1,1,2} \wedge (y)_{2,1,2,2,2,2} = (y)_{2,1,2,2,2,1,(y)_{2,1,2,2,2,1,1,3}} + 1 \wedge \\ & (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\ & ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\ & Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\ & ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\ & Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})\end{aligned}$$

Predikater for $AssAdd$, $AssTimes$, regelskjemaet for likhet av funksjoner Eqf og den sammentrekte (eng. contracted) versjonen av denne, $Eqfc$, defineres på tilsvarende måte. Her er predikater for funksjoner definert ved skjemaene for primitiv rekursjon og komposisjon.

- *f0*

$$\frac{\langle \lceil R2 \rceil, \langle 5, \langle 29, \langle \langle \langle 4, \lceil g \rceil, \lceil h \rceil \rangle, \lceil a_1 \rceil, \ldots, \lceil a_n \rceil, 0 \rangle, \langle \lceil g \rceil, \lceil a_1 \rceil, \ldots, \lceil a_n \rceil \rangle \rangle \rangle, \ldots \rangle, 3, \lceil \Delta \rceil \rangle}{\langle \langle 31 \rangle, \lceil \Gamma \rceil, 3, \lceil \Delta \rceil \rangle}$$

$$
\begin{aligned}
f0(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 31 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 29 \wedge \\
& ln(y)_{2,1,2,2,2} = 2 \wedge ln(y)_{2,1,2,2,2,1,1} = 3 \wedge (y)_{2,1,2,2,2,1,1,1} = 4 \wedge \\
& (y)_{2,1,2,2,2,1,ln(y)_{2,1,2,2,2,1}} = 0 \wedge (y)_{2,1,2,2,2,2,1} = (y)_{2,1,2,2,2,1,1,2} \wedge \\
& (\forall j)_{2 \leq j \leq ln(y)_{2,1,2,2,2,2}} ((\forall i)_{2 \leq i \leq ln(y)_{2,1,2,2,1}} (y)_{2,1,2,2,2,1,i} = (y)_{2,1,2,2,2,2,j}) \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

- *frec*

$\langle \ldots, \langle 29, \langle\langle 4, \lceil g \rceil, \lceil h \rceil\rangle, \lceil a_1 \rceil, \ldots, \lceil a_n \rceil, \langle\langle 1\rangle, \lceil b \rceil\rangle\rangle, \langle \lceil h \rceil, \lceil a_1 \rceil, \ldots, \lceil a_n \rceil, \lceil b \rceil, \langle\langle 4, \lceil g \rceil, \lceil h \rceil\rangle, \lceil a_1 \rceil, \ldots, \lceil a_n \rceil, \lceil b \rceil\rangle\rangle\rangle\rangle, \ldots \rangle$

$$
\Big|
$$

$\langle\langle 32\rangle, \lceil \Gamma \rceil, 3, \lceil \Delta \rceil\rangle$

$$
\begin{aligned}
frec(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 32 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 29 \wedge \\
& ln(y)_{2,1,2,2,2} = 2 \wedge ln(y)_{2,1,2,2,2,1,1} = 3 \wedge (y)_{2,1,2,2,2,1,1,1} = 4 \wedge \\
& (y)_{2,1,2,2,2,1,ln(y)_{2,1,2,2,2,1}} \geq 0 \wedge (y)_{2,1,2,2,2,1,1} = (y)_{2,1,2,2,2,2,ln(y)_{2,1,2,2,2,2},1} \\
& \wedge ln(y)_{2,1,2,2,2,1} = ln(y)_{2,1,2,2,2,2,ln(y)_{2,1,2,2,2,2}} = ln(y)_{2,1,2,2,2,2} + 1 \\
& \wedge (\forall i)_{2 \leq i < ln(y)_{2,1,2,2,2,1}} ((y)_{2,1,2,2,2,2,i} = (y)_{2,1,2,2,2,2,ln(y)_{2,1,2,2,2,2},i} \\
& = (y)_{2,1,2,2,2,1,i}) \wedge (y)_{2,1,2,2,2,1,ln(y)_{2,1,2,2,2,1},1} = \langle 1 \rangle \wedge \\
& (y)_{2,1,2,2,2,1,ln(y)_{2,1,2,2,2,1},2} = (y)_{2,1,2,2,2,2,ln(y)_{2,1,2,2,2,2},ln(y)_{2,1,2,2,2,2,ln(y)_{2,1,2,2,2,2}}} \\
& = (y)_{2,1,2,2,2,2,(ln(y)_{2,1,2,2,2,2}-1)} \wedge (y)_{2,1,2,2,2,2,1} = (y)_{2,1,2,2,2,1,3} \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

- *fcomp*

$\langle \dots, \langle 29, \langle \langle \langle 3, \lceil h \rceil, \lceil g_1 \rceil, \dots, \lceil g_m \rceil \rangle, \lceil a_1 \rceil \dots, \lceil a_n \rceil \rangle, \langle \lceil h \rceil, \langle \lceil g_1 \rceil, \lceil a_1 \rceil \dots, \lceil a_n \rceil \rangle, \dots, \langle \lceil g_m \rceil, \lceil a_1 \rceil \dots, \lceil a_n \rceil \rangle \rangle \rangle \rangle, \dots \rangle$

$\langle \langle 33 \rangle, \lceil \Gamma \rceil, 3, \lceil \Delta \rceil \rangle$

$$
\begin{aligned}
fcomp(y) \Leftrightarrow & ln(y) = 2 \wedge ln(y)_1 = 4 \wedge ln(y)_{2,1} = 4 \wedge \\
& (y)_{1,1} = \langle 33 \rangle \wedge ln(y)_{2,1,2,2} = 2 \wedge (y)_{2,1,2,2,1} = 29 \wedge \\
& ln(y)_{2,1,2,2,2} = 2 \wedge ln(y)_{2,1,2,2,2,1,1} \geq 3 \wedge (y)_{2,1,2,2,2,1,1,1} = 3 \wedge \\
& (y)_{2,1,2,2,2,1,1,2} = (y)_{2,1,2,2,2,2,1} \wedge \\
& (\forall i)_{3 \leq i \leq ln(y)_2, 1,2,2,2,1,1}((y)_{2,1,2,2,2,1,1,i} = (y)_{2,1,2,2,2,2,(i \dot{-} 1),1} \wedge \\
& (\forall j)_{2 \leq j \leq ln(y)_{2,1,2,2,2,1}}((y)_{2,1,2,2,2,1,j} = (y)_{2,1,2,2,2,2,i,j})) \wedge \\
& (y)_{2,1,2,1} = (y)_{1,2,1} = 1 \wedge (y)_{1,4,1} = 2 \wedge (y)_{2,1,3} = (y)_{1,3} = 3 \wedge \\
& ln(y)_{2,1,2} = ln(y)_{1,2} + 1 \wedge \\
& Eq(y, (y)_{1,2,2}, (y)_{1,2,ln(y)_{1,2}}, y, (y)_{2,1,2,3}, (y)_{2,1,2,ln(y)_{2,1,2}}) \wedge \\
& ln(y)_{2,1,4} = ln(y)_{1,4} \wedge \\
& Eq(y, (y)_{1,4,1}, (y)_{1,4,ln(y)_{1,4}}, y, (y)_{2,1,4,1}, (y)_{2,1,4,ln(y)_{2,1,4}})
\end{aligned}
$$

Tillegg C

Aksiomfil

1. $\dfrac{N(0), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ NN$
2. $\dfrac{N(Sa), N(a), \Gamma \Rightarrow \Delta}{N(a), \Gamma \Rightarrow \Delta}\ RN$
3. $\dfrac{Sa = Sb, a = b, \Gamma \Rightarrow \Delta}{a = b, \Gamma \Rightarrow \Delta}\ EqS$
4. $\dfrac{+ac = +bd, a = b, c = d, \Gamma \Rightarrow \Delta}{a = b, c = d, \Gamma \Rightarrow \Delta}\ Eq+$
5. $\dfrac{*ac = *bd, a = b, c = d, \Gamma \Rightarrow \Delta}{a = b, c = d, \Gamma \Rightarrow \Delta}\ Eq*$
6. $\dfrac{b = c, a = b, a = c, \Gamma \Rightarrow \Delta}{a = b, a = c, \Gamma \Rightarrow \Delta}\ Trans$
7. $\dfrac{b = a, a = b, \Gamma \Rightarrow \Delta}{a = b, \Gamma \Rightarrow \Delta}\ Sym$
8. $\dfrac{Oa = 0, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ IINull$
9. $\dfrac{I22ab = b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ III22$
10. $\dfrac{Pred0 = 0, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ NNPred$
11. $\dfrac{PredSa = I22Predaa, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ RRPred$
12. $\dfrac{rI220b = S0, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ NNI22$

13. $\dfrac{rI22Sab = I22rI22abb, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ RRI22$
14. $\dfrac{+0b = b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ NNAdd$
15. $\dfrac{+Sab = S + ab, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ RRAdd$
16. $\dfrac{++abc = +a + bc, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ AAssAdd$
17. $\dfrac{*0b = 0, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ NNTimes$
18. $\dfrac{*Sab = + * abb, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ RRTimes$
19. $\dfrac{** abc = *a * bc, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ AAssTimes$
20. $\dfrac{PiI220b = rI220b, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ PPPiI22n$
21. $\dfrac{PiI22Sab = *PiI22abrI22Sab, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ PPPiI22r$
22. $\dfrac{SPa = SPreda, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ CCSP$
23. $\dfrac{PiSP0 = SP0, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ PPPiSPn$
24. $\dfrac{PiSPSa = *PiSPaSPSa, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ PPPiSPr$

Referanser

[1] S. Bellantoni og S. Cook. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity*, 2:97–110, 1992.

[2] G. Boolos. Don't eliminate cut. I R. Jeffrey, redaktør, *Logic, Logic and Logic*, kapittel 23. Harvard University Press, 1999.

[3] P. Clote. Computation models and function algebras. I E. R. Griffor, redaktør, *Handbook of Computability Theory*, bind 140, side 589–681. Elsevier, 1999.

[4] A. Cobham. The intrinsic computational difficulty of functions. I Y. Bar-Hillel, redaktør, *Logic, methodology and philosophy of science*, side 24–30, 1964.

[5] D. van Dalen. *Logic and Structure.* Springer, 1997.

[6] J. E. Fenstad og D. Normann. Innføring i matematisk logikk. Universitetet i Oslo, matematisk institutt, 1990.

[7] K. Gödel. Über formal unentscheidbare Sätze der Principa mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 1931. English translation in [9], 596-616.

[8] P. Hájek og P. Pudlák. *Metamathematics of First-Order Arithmetic.* Springer, 1998.

[9] J. van Heijenoort, redaktør. *From Frege to Gödel. A source book in mathematical logic 1879 - 1931.* Harvard University Press, 1967.

[10] M. Holden. A proof theoretic characterization of the Kalmar elementary functions. University of Oslo, Department of informatics, March 1996. Research report 211.

[11] P. Hudak. *The Haskell School of Expression.* Cambridge University Press, 2000.

[12] G. Hutton og E. Meijer. Functional pearls: Monadic parsing in Haskell. *Journal of Functional Programming*, 8(4):437–444, 1998.

[13] R. C. Jeffrey. *Formal logic: Its scope and limits.* Tata McGraw-Hill Publishing Company Ltd., 1980.

[14] H. R. Jervell. The importance of indirect arguments. Forelesning gitt ved Universitetet i Helsinki, oktober 2002.

[15] H. R. Jervell og W. Zhang. Cut formulas for Kalmar elementary functions. Report No. 33, 2000/2001, Institut Mittag-Leffler, The Royal Swedish Academy of Sciences, 2000.

[16] G. Kreisel. Mathematical logic. I T. L. Saaty, redaktør, *Lectures on Modern Mathematics*, bind III, side 95–195. Wiley and Sons, 1965.

[17] L. Kristiansen. Litt subrekursjonsteori og enda mindre rekursjonsteori. HiO-notat 2000 nr. 24, Høgskolen i Oslo, 2000.

[18] D. Leivant. Stratified functional programs and computational complexity. I *Proceedings of Principles of Programming Languages*, side 325–333, 1993.

[19] D. Leivant. Intrinsic theories and computational complexity. I *Logic and Computational Complexity*, bind 960 av *Lecture Notes in Computer Science*, side 177–194. Springer, 1995.

[20] S. Negri. Sequent calculus proof theory of inuitionistic apartness relations. *Archive for Mathematical Logic*, 38(8):521–547, 1999.

[21] S. Negri og J. von Plato. Cut elimination in the presence of axioms. *The Bulletin of Symbolic Logic*, 4(4):418–436, 1998.

[22] S. Negri og J. von Plato. *Structural Proof Theory.* Cambridge University Press, 2001.

[23] P. Odifreddi. *Classical Recursion Theory*, bind 125. North Holland, 2nd utgave, 1999.

[24] V. P. Orevkov. Lower bounds for lengthening proofs. *Journal of Soviet mathematics*, 20:2337–2350, 1982. A Translation of Selected Russian-Language Serial Publications in Mathematics.

[25] G. E. Ostrin og S. S. Wainer. Proof theoretic complexity. I H. Schwichtenberg og R. Steinbrüggen, redaktører, *Proof and System-Reliability*, bind 62 av *NATO Science Series: II: Mathematics, Physics and Chemistry*. Kluwer, 2002.

[26] R. Péter. *Recursive Functions*. Academic Press, 1967.

[27] S. Peyton Jones og J. Hughes (redaktører). Report on the programming language Haskell 98. a non-strict, purely functional language. Rapport, `http://www.haskell.org/`, 1999.

[28] R. W. Ritchie. Classes of predictably computable functions. *Transactions of the American Mathematical Society*, 106:139–173, 1963.

[29] H. Rogers Jr. *Theory of Recursive Functions and Effective Computability*. Mc.Graw-Hill, 1967.

[30] H. E. Rose. *Subrecursion, Functions and hierarchies*. Clarendon Press, 1984.

[31] S. Šereš. Snitt induksjon ved automatisk generering av bevis. Hovedfagsoppgave, Universitetet i Oslo, Institutt for informatikk, 1996.

[32] T. A. Sudkamp. *Languages and Machines. An Introduction to the Theory of Computer Science*. Addison-Wesley Longman, 1997.

[33] A. Troelstra og H. Schwichtenberg. *Basic Proof Theory*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2nd utgave, 2000.

[34] W. Zhang. Cut elimination and automatic proof procedures. *Theoretical Computer Science*, 91:265–284, 1991.

[35] B. M. Østvold. *Synthesis of Recursive Functional Programs from Examples*. Doktorgradsoppgave, NTNU Trondheim, 1999.

[36] B. M. Østvold. Funksjonell programmering og Haskell. Forelesning gitt på hovedfagsseminar i logikk, Universitetet i Oslo, 2001.