Et bevis i Peano aritmetik

Frederik Eriksen (eriksen@diku.dk)

29. juni 2005

Indhold

1 Indledning **3**

2 Lidt om Logiweb **3**
2.1 Formelle konstruktioner . . . 3
2.2 Særlige definitioner . . . 4

3 Peano aritmetik **4**
3.1 Syntaks . . . 4
3.1.1 Termer . . . 4
3.1.2 Formler . . . 5
3.2 Aksiomatisk system . . . 5

4 Lemmaer, afledte lemmaer og makroer **7**
4.1 Lemmaer . . . 7
4.2 Afledte lemmaer . . . 9
4.2.1 Regellemmaer . . . 9
4.2.2 Betingede regellemmaer . . . 9
4.2.3 Navneafledte lemmaer . . . 10
4.3 Makroer . . . 10

5 Beviser for makroerne **13**
5.1 Bevis for Weakening . . . 13
5.2 Bevis for DoubleMP′ . . . 14
5.3 Bevis for ConditionedMP′ . . . 15
5.4 Bevis for DoubleConditionedMP′ . . . 15
5.5 Bevis for ImplyTransitivity . . . 16
5.6 Bevis for PermuteAntecedents . . . 17
5.7 Bevis for AddOne . . . 18
5.8 Bevis for SubstitutionMacro . . . 18
5.9 Bevis for InductionMacro . . . 19

6 Beviser for lemmaerne **20**
6.1 Bevis for EqualReflexivity . . . 22
6.2 Bevis for EqualSymmetry og heraf afledt lemma . . . 22
6.3 Bevis for EqualTransitivity og heraf afledte lemmaer . . . 23
6.4 Bevis for Mendelson3.2(d) og heraf afledte lemmaer . . . 25
6.5 Bevis for Mendelson3.2(f) . . . 26
6.6 Bevis for Mendelson3.2(g) og heraf afledt lemma . . . 28
6.7 Bevis for PlusCommutativity . . . 30

A Navnet på dette dokument **32**

B TEX definitioner **32**

C Litteratur **33**

Figurer

1 Den overordnede struktur af beviset for PlusCommutativity . . . 20

Tabeller

1 Makroer der beviser makroer . . . 13
2 Makroer der beviser lemmaer . . . 21

1 Indledning

Her er en matematisk sætning: Ligheden

$$n + m \quad = \quad m + n \tag{1}$$

gælder for alle naturlige tal n og m. (Ved et "naturligt tal" forstår jeg et ikke-negativt heltal).

Denne rapport indeholder et formelt bevis for sætning (1). Beviset gennemføres inden for rammerne af et formelt system kaldet "Peano aritmetik". Rapporten er udarbejdet ved hjælp af Logiweb, som er et system til verifikation og publikation af tekster, der indeholder formel matematik. Logiweb har vericeret, at rapportens bevis for sætning (1) er korrekt — så jeg kan tillade mig at påstå, at rapporten ikke indeholder nogen formelle fejl. Jeg har også brugt Logiweb til at publicere rapporten på internettet; adressen er:
`http://www.diku.dk/~grue/logiweb/20050502/home/eriksen/peano-commutativity`.

Rapportens bevis for (1) findes allerede i litteraturen — se [3], sætning 3.2(h). Det er altså mere rimeligt at betragte rapporten som en formel øvelse i at bruge Logiweb, end som et selvstændigt stykke matematisk arbejde; men formelle øvelser har jo også deres berettigelse.

Rapporten er struktueret som følger: Afsnit 2 handler om Logiweb. Det er ikke en detaljeret beskrivelse af systemet, men giver forhåbentlig oplysninger nok til, at man kan forstå rapportens formelle indhold uden yderligere viden om Logiweb. Afsnit 3 beskriver syntaksen for Peano aritmetik samt det aksiomsystem, der bliver brugt i rapporten. Afsnit 4 præsenterer de 28 hjælpesætninger, som jeg vil bruge til at bevise ligning (1). For overskuelighedens skyld har jeg inddelt hjælpesætningerne i 3 undergrupper: Lemmaer, afledte lemmaer og makroer. Makroerne bliver bevist i afsnit 5; og de resterende hjælpesætninger bliver bevist i afsnit 6.

2 Lidt om Logiweb

Som nævnt i indledningen er denne rapport skrevet ved hjælp af Logiweb. Dette betyder dels, at rapportens formelle indhold er defineret ud fra nogle andre Logiweb-dokumenter, og dels at rapporten har nogle stilistiske særtræk. Disse to facetter ved Logiweb vil jeg kort forklare i afsnit 2.1 og 2.2. For en detaljeret beskrivelse af hele Logiweb systemet vil jeg henvise til [1]; her kan man bla. læse om den bevischecker, der har verificeret rapportens beviser.

2.1 Formelle konstruktioner

Det formelle indhold af et Logiweb-dokument er sammensat af en række formelle konstruktioner. En formel konstruktion kan repræsentere alt, hvad der har med formel matematik at gøre: Variable, funktioner, lemmaer, beviser, osv. Der er to kilder til konstruktionerne i et Logiweb-dokument: Dels kan man indføre sine egne konstruktioner, og dels kan man importere konstruktioner fra andre

Logiiweb-dokumenter. I denne rapport er alle konstruktioner importerede fra de to Logiweb-dokumenter [1] og [2] — med undtagelse af de konstruktioner, der repræsenterer rapportens 28 hjælpesætninger; dem definerer jeg selv.

2.2 Særlige definitioner

Den formelle del af et Logiweb-dokument skrives i et formateringssprog ved navn "pyk". Hver formel konstruktion, man arbejder med i dokumentet, har tilknyttet en såkaldt "pyk definition". Dette er en angivelse af, hvad man skal skrive, hvis man i et andet Logiweb-dokument ønsker at benytte den pågældende konstruktion. Hvis man indfører en ny konstruktion i sit Logiweb-dokument, er det et krav, at man gør den tilsvarende pyk definition tilgængelig i dokumentet. Dette krav vil jeg imødekomme ved at nævne pyk definitionen i en fodnote, hver gang jeg præsenterer en ny hjælpesætning.

Logiweb genererer det færdige dokument ved hjælp af det kendte formateringssprog LaTeX. Derfor har hver formel konstruktion også tilknyttet en såkaldt "TeX definition", som angiver, hvordan konstruktionen skal skrives i LaTeX. Ligesom pyk definitioner skal også TeX definitioner være tilgængelige i dokumentet. TeX definitionerne for denne rapports lemmaer er vedlagt i bilag B.

Endelig skal det nævnes, at Logiweb opfatter selve navnet på et Logiweb-dokument som en formel konstruktion. Derfor skal ethvert Logiweb-dokument indeholde mindst én pyk definition — nemlig pyk definitionen af dokumentets eget navn. Denne definition har jeg vedlagt i bilag A.

3 Peano aritmetik

Peano aritmetik er et formelt sprog, der bruges til at repræsentere de naturlige tal. Jeg vil bruge nøjagtig den samme version af Peano aritmetik som dén, der præsenteres i [2]. Dette gælder både mht. syntaksen (som jeg gennemgår i afsnit 3.1), og mht. det anvendte aksiomsystem (som jeg gennemgår i afsnit 3.2).

3.1 Syntaks

Peano aritmetik er opbygget af termer og formler; jeg gennemgår hver af disse begreber i de følgende to underafsnit.

3.1.1 Termer

Syntaksen for en term $\mathcal{T}$ kan beskrives ved den følgende BNF-grammatik:

$$\mathcal{T} \quad ::= \quad \dot{0} \mid \mathcal{T}' \mid \mathcal{T} \dot{+} \mathcal{T} \mid \mathcal{T} \dot{\cdot} \mathcal{T} \mid \texttt{Pvar} \tag{2}$$

Efterfølgeren m til et naturligt tal n er det mindste naturlige tal, der er større end n. Efterfølgeroperatoren er den funktion, der fører n over i m. Ideen med den syntaktiske konstruktion $'$ er, at den skal repræsentere efterfølgeroperatoren.

På denne måde kan vi repræsentere alle naturlige tal med de to konstruktioner $\dot{0}$ og $'$ — 1 kan repræsenteres som $\dot{0}'$, 2 som $(\dot{0}')'$, [1] osv.

Ideen med konstruktionerne $\dot{+}$ og $\dot{\cdot}$ er, at de skal repræsentere regningsarterne addition og multiplikation. Da denne rapport kun omhandler addition, kommer vi ikke til at se mere til tegnet $\dot{\cdot}$.

Den sidste konstruktion i (2) er `Pvar`, som er en forkortelse for "Peano variabel" . En Peano variabel er en variabel, der varierer over konstante termer (hvor en konstant term er en term, der ikke indeholder Peano variable). Artiklen [2] bruger små bogstaver med en prik over til at benævne Peano variable — altså f.eks. $\dot{a}$ og $\dot{b}$. Jeg vil gøre det samme, idet jeg dog kun vil bruge bogstaverne $\dot{t}$, $\dot{r}$ og $\dot{s}$.

3.1.2 Formler

Syntaksen for en formel $\mathcal{F}$ kan beskrives ved den følgende BNF-grammatik:

$$\mathcal{F} \quad ::= \quad \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{T} \mid \dot{\neg}\mathcal{F} \mid \mathcal{F} \dot{\Rightarrow} \mathcal{F} \mid \dot{\forall}\texttt{Pvar} : \mathcal{F}$$

Med en formel kan vi altså påstå, at to termer er lig hinanden. Herudover kan vi negere formler, sætte formler sammen med implikationstegnet $\dot{\Rightarrow}$ samt sætte alkvantoren $\dot{\forall}$ foran formler. Bemærk at $\dot{\Rightarrow}$ er højreassociativ; et udtryk som f.eks. $\dot{r} \dot{\Rightarrow} \dot{s} \dot{\Rightarrow} \dot{t}$ skal læses som $\dot{r} \dot{\Rightarrow} (\dot{s} \dot{\Rightarrow} \dot{t})$.

3.2 Aksiomatisk system

Beviserne i denne rapport baserer sig på det aksiomatiske system S$'$ fra [2]. Dette system består af 14 aksiomskemaer (kaldet A1$'$— A5$'$ og S1$'$—S9$'$) og 2 slutningsregler (kaldet MP$'$ og Gen$'$). For overblikkets skyld genoptrykker jeg dem her:

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{A1}'\colon \forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A} \dot{\Rightarrow} \mathcal{B} \dot{\Rightarrow} \mathcal{A}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{A2}'\colon \forall\mathcal{A}\colon\forall\mathcal{B}\colon\forall\mathcal{C}\colon (\mathcal{A} \dot{\Rightarrow} \mathcal{B} \dot{\Rightarrow} \mathcal{C}) \dot{\Rightarrow} (\mathcal{A} \dot{\Rightarrow} \mathcal{B}) \dot{\Rightarrow} \mathcal{A} \dot{\Rightarrow} \mathcal{C}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{A3}'\colon (\dot{\neg}\mathcal{B} \dot{\Rightarrow} \dot{\neg}\mathcal{A}) \dot{\Rightarrow} (\dot{\neg}\mathcal{B} \dot{\Rightarrow} \mathcal{A}) \dot{\Rightarrow} \mathcal{B}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{A4}'\colon \forall\mathcal{C}\colon\forall\mathcal{A}\colon\forall\mathcal{X}\colon\forall\mathcal{B}\colon \lceil\mathcal{A}\rceil \equiv \langle\lceil\mathcal{B}\rceil \mid \lceil\mathcal{X}\rceil := \lceil\mathcal{C}\rceil\rangle \Vdash \dot{\forall}\mathcal{X}\colon \mathcal{B} \dot{\Rightarrow} \mathcal{A}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{A5}'\colon \forall\mathcal{X}\colon\forall\mathcal{A}\colon\forall\mathcal{B}\colon \dot{\mathrm{nonfree}}(\mathcal{X},\mathcal{A}) \Vdash \dot{\forall}\mathcal{X}\colon (\mathcal{A} \dot{\Rightarrow} \mathcal{B}) \dot{\Rightarrow} \mathcal{A} \dot{\Rightarrow} \dot{\forall}\mathcal{X}\colon \mathcal{B}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{MP}'\colon \forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A} \dot{\Rightarrow} \mathcal{B} \vdash \mathcal{A} \vdash \mathcal{B}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{Gen}'\colon \forall\mathcal{X}\colon\forall\mathcal{A}\colon \mathcal{A} \vdash \dot{\forall}\mathcal{X}\colon \mathcal{A}]$$

$$[\mathrm{S}' \ \textbf{rule}\ \mathrm{S1}'\colon \forall\mathcal{A}\colon\forall\mathcal{B}\colon\forall\mathcal{C}\colon \mathcal{A} \overset{\mathrm{p}}{=} \mathcal{B} \dot{\Rightarrow} \mathcal{A} \overset{\mathrm{p}}{=} \mathcal{C} \dot{\Rightarrow} \mathcal{B} \overset{\mathrm{p}}{=} \mathcal{C}]$$

[1] Egentlig bør $(\dot{0}')'$ skrives $\dot{0}''$. Imidlertid er der en lille fejl ved TEX definitionen af konstruktionen $'$ i [2]. Fejlen betyder, at LATEX ikke accepterer to anvendelser af $'$ i træk — derfor den ekstra parentes.

[S′ **rule** S2′: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A} \stackrel{\text{p}}{=} \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{A}' \stackrel{\text{p}}{=} \mathcal{B}'$]

[S′ **rule** S3′: $\forall\mathcal{A}\colon \dot{\neg}\dot{0} \stackrel{\text{p}}{=} \mathcal{A}'$]

[S′ **rule** S4′: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A}' \stackrel{\text{p}}{=} \mathcal{B}' \mathrel{\dot{\Rightarrow}} \mathcal{A} \stackrel{\text{p}}{=} \mathcal{B}$]

[S′ **rule** S5′: $\forall\mathcal{A}\colon \mathcal{A} \dot{+} \dot{0} \stackrel{\text{p}}{=} \mathcal{A}$]

[S′ **rule** S6′: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A} \dot{+} \mathcal{B}' \stackrel{\text{p}}{=} (\mathcal{A} \dot{+} \mathcal{B})'$]

[S′ **rule** S7′: $\forall\mathcal{A}\colon \mathcal{A} \dot{\cdot} \dot{0} \stackrel{\text{p}}{=} \dot{0}$]

[S′ **rule** S8′: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A} \dot{\cdot} (\mathcal{B}') \stackrel{\text{p}}{=} (\mathcal{A} \dot{\cdot} \mathcal{B}) \dot{+} \mathcal{A}$]

[S′ **rule** S9′: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon\forall\mathcal{C}\colon\forall\mathcal{X}\colon$
$\mathcal{B}\equiv\langle\mathcal{A}|\mathcal{X} := \dot{0}\rangle \Vdash \mathcal{C}\equiv\langle\mathcal{A}|\mathcal{X} := \mathcal{X}'\rangle \Vdash$
$\mathcal{B} \mathrel{\dot{\Rightarrow}} \dot{\forall}\mathcal{X}\colon (\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{C}) \mathrel{\dot{\Rightarrow}} \dot{\forall}\mathcal{X}\colon \mathcal{A}$]

For at lette forståelsen af S′ vil jeg slutte dette afsnit med en generel beskrivelse af de to komponenter, som S′ består af: Aksiomskemaer og slutningsregler.

Et aksiom er en formel, som vi kan tage for givet. Et aksiomskema repræsenterer en mængde af aksiomer. F.eks. følger det af aksiom A1′, at formler som $\dot{r} \mathrel{\dot{\Rightarrow}} \dot{t} \mathrel{\dot{\Rightarrow}} \dot{r}$ og $\dot{s} \mathrel{\dot{\Rightarrow}} \dot{0} \stackrel{\text{p}}{=} \dot{0}' \mathrel{\dot{\Rightarrow}} \dot{s}$ er aksiomer. Variablene $\mathcal{A}$ og $\mathcal{B}$ i A1′ kaldes for meta-variable. Meta-variable er ikke en del af Peano aritmetik; de bruges i stedet, når vi taler om Peano aritmetik — hvilket vi jo gør i vores aksiomsystem. (Der er en tilsvarende forskel mellem den alkvantor $\dot{\forall}$, som vi bruger i Peano aritmetik, og den alkvantor $\forall$, som vi bruger i S′). Meta-variable kan variere over Peano variable, vilkårlige termer eller formler. I S′ bruges "$\mathcal{X}$" om den første slags meta-variable, mens "$\mathcal{A}$", "$\mathcal{B}$" og "$\mathcal{C}$" bruges om de to andre slags meta-variable.

Ud over aksiomskemaerne indeholder S′ som nævnt to slutningsregler. En slutningsregel er en regel for, hvordan vi kan vise formler ud fra andre formler, som allerede er bevist. F.eks. siger MP′, at vi kan vise formlen $\mathcal{B}$, hvis vi i forvejen har vist de to formler $\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B}$ og $\mathcal{A}$. Slutningsreglerne definerer således en beviselighedsrelation $\vdash$ mellem formler.

Tilsammen definerer aksiomskemaerne og slutningsreglerne en mængde af beviselige formler: En beviselig formel er enten et aksiom, eller også kan den bevises ud fra aksiomerne ved hjælp af slutningsreglerne. Jeg vil bruge ordet "systemsætning" som en fællesbetegnelse for aksiomskemaer og slutningsregler.

En systemsætning kan indeholde en eller flere sidebetingelser; dette gælder for tre af aksiomskemaerne i system S′ (A4′, A5′ og S9′). En sidebetingelse i en systemsætning definerer et krav til den korrekte anvendelse af systemsætningen. I S9′ er det f.eks. klart, at implikationen [$\mathcal{B} \mathrel{\dot{\Rightarrow}} \dot{\forall}\mathcal{X}\colon (\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{C}) \mathrel{\dot{\Rightarrow}} \dot{\forall}\mathcal{X}\colon \mathcal{A}$] ikke har generel gyldighed: $\mathcal{B}$ skal være et basistilfælde, og formlen $\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{C}$ skal repræsentere det induktive skridt. Disse ekstra krav er formuleret i de to sidebetingelser i S9′. Rent syntaktisk bruger vi tegnet "$\models$" til at adskille en sidebetingelse fra den øvrige del af systemsætningen.

4 Lemmaer, afledte lemmaer og makroer

Som nævnt i indledningen har jeg opdelt beviset for (1) i 28 hjælpesætninger. For overskuelighedens skyld har jeg inddelt disse hjælpesætninger i tre grupper: Lemmaer, afledte lemmaer og makroer. Distinktionen mellem disse grupper er ikke knivskarp, men den gør det lettere at forklare bevisets struktur. I dette afsnit vil jeg gennemgå hver gruppe og præsentere de hjælpesætninger, som grupperne indeholder.

I min definition af de forskellige hjælpesætninger vil jeg bruge de følgende konventioner for variabelnavne:

- Meta-variable, der varierer over formler, betegnes med "$\mathcal{A}$", "$\mathcal{B}$", "$\mathcal{C}$", "$\mathcal{D}$", "$\mathcal{E}$" og "$\mathcal{F}$".
- Meta-variable, der varierer over vilkårlige termer, betegnes med "$\mathcal{T}$", "$\mathcal{R}$", "$\mathcal{S}$" og "$\mathcal{U}$".
- Meta-variable, der varierer over Peano variable, betegnes med "$\mathcal{Z}$".
- Peano variable betegnes med "$\dot{t}$", "$\dot{r}$" og "$\dot{s}$"(som allerede nævnt i afsnit 3.1.1).

4.1 Lemmaer

Under kategorien "lemmaer" vil jeg for det første rubricere hovedsætningen (1), formuleret i Peano aritmetik:

[S′ **lemma** PlusCommutativity: $\dot{t} \dot{+} \dot{r} \stackrel{\text{p}}{=} \dot{r} \dot{+} \dot{t}$] [2]

Som det kan ses, benytter PlusCommutativity sig ikke af meta-variable (som f.eks. $\mathcal{T}$ og $\mathcal{R}$), men derimod af Peano variablene $\dot{t}$ og $\dot{r}$. Der er en teknisk forklaring herpå: PlusCommutativity skal vises ved induktion, og derfor kommer aksiomskemaet S9′ uundgåeligt til at indgå i beviset. Som nævnt i afsnit 3.2 benytter S9′ sig af sidebetingelser; og Logiwebs bevischecker kan (endnu) ikke evaluere sidebetingelser, der indeholder meta-variable.

I afsnit 3.1 i lærebogen [3] beviser Mendelson PlusCommutativity under navnet "sætning 3.2(h)". For at nå frem til sætning 3.2(h) viser Mendelson først de følgende seks lemmaer: [3]

[S′ **lemma** EqualReflexivity: $\forall \mathcal{T} : \mathcal{T} \stackrel{\text{p}}{=} \mathcal{T}$] [4]

[S′ **lemma** EqualSymmetry: $\forall \mathcal{T} : \forall \mathcal{R} : \mathcal{T} \stackrel{\text{p}}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{R} \stackrel{\text{p}}{=} \mathcal{T}$] [5]

[2] [PlusCommutativity $\stackrel{\text{pyk}}{=}$ "plus commutativity"]. Dette er en "pyk definition", som omtalt i afsnit 2.2.

[3] Det skal lige nævnes, at Mendelson bruger meta-variable i sin formulering af Mendelson3.2(f) og Mendelson3.2(g). Fordi disse lemmaer skal vises ved induktion, bruger jeg Peano variable i stedet for.

[4] [EqualReflexivity $\stackrel{\text{pyk}}{=}$ "equal reflexivity"]

[5] [EqualSymmetry $\stackrel{\text{pyk}}{=}$ "equal symmetry"]

[S′ **lemma** EqualTransitivity: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \forall \mathcal{S}\colon \mathcal{T} \stackrel{p}{=} \mathcal{R} \mathrel{\dot\Rightarrow} \mathcal{R} \stackrel{p}{=} \mathcal{S} \mathrel{\dot\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$] [6]

[S′ **lemma** Mendelson3.2(d): $\forall \mathcal{T}\colon \forall \mathcal{U}\colon \forall \mathcal{S}\colon \mathcal{T} \stackrel{p}{=} \mathcal{U} \mathrel{\dot\Rightarrow} \mathcal{S} \stackrel{p}{=} \mathcal{U} \mathrel{\dot\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$] [7]

[S′ **lemma** Mendelson3.2(f): $\dot{t} \stackrel{p}{=} \dot{0} \mathbin{\dot+} \dot{t}$] [8]

[S′ **lemma** Mendelson3.2(g): $\dot{t}' \mathbin{\dot+} \dot{r} \stackrel{p}{=} (\dot{t} \mathbin{\dot+} \dot{r})'$] [9]

Lemmaerne EqualReflexivity, EqualSymmetry og EqualTransitivity udsiger tilsammen, at $\stackrel{p}{=}$ er en ækvivalensrelation — altså en relation, som er refleksiv, symmetrisk og transitiv. Indholdet af de tre lemmaer Mendelson3.2(d), Mendelson3.2(f) og Mendelson3.2(g) kan ikke sammenfattes lige så koncist, så jeg har givet dem de samme navne, som de har i [3].

Ligesom PlusCommutativity skal Mendelson3.2(f) og Mendelson3.2(g) bevises ved induktion. For at begrænse længden af de enkelte beviser har jeg valgt at definere basistilfældene og de induktive skridt som selvstændige lemmaer. På denne måde får vi yderligere seks lemmaer:

[S′ **lemma** Mendelson3.2(f)Base: $\dot{0} \stackrel{p}{=} \dot{0} \mathbin{\dot+} \dot{0}$] [10]

[S′ **lemma** Mendelson3.2(f)Indu: $\dot{t} \stackrel{p}{=} \dot{0} \mathbin{\dot+} \dot{t} \mathrel{\dot\Rightarrow} \dot{t}' \stackrel{p}{=} \dot{0} \mathbin{\dot+} \dot{t}'$] [11]

[S′ **lemma** Mendelson3.2(g)Base: $\dot{t}' \mathbin{\dot+} \dot{0} \stackrel{p}{=} (\dot{t} \mathbin{\dot+} \dot{0})'$] [12]

[S′ **lemma** Mendelson3.2(g)Indu: $\dot{t}' \mathbin{\dot+} \dot{r} \stackrel{p}{=} (\dot{t} \mathbin{\dot+} \dot{r})' \mathrel{\dot\Rightarrow} \dot{t}' \mathbin{\dot+} \dot{r}' \stackrel{p}{=} (\dot{t} \mathbin{\dot+} \dot{r}')'$] [13]

[S′ **lemma** PlusCommutativityBase: $\dot{t} \mathbin{\dot+} \dot{0} \stackrel{p}{=} \dot{0} \mathbin{\dot+} \dot{t}$] [14]

[S′ **lemma** PlusCommutativityIndu: $\dot{t} \mathbin{\dot+} \dot{r} \stackrel{p}{=} \dot{r} \mathbin{\dot+} \dot{t} \mathrel{\dot\Rightarrow} \dot{t} \mathbin{\dot+} \dot{r}' \stackrel{p}{=} \dot{r}' \mathbin{\dot+} \dot{t}$] [15]

Bemærk at induktionen i Mendelson3.2(g) og PlusCommutativity forløber efter variablen $\dot{r}$ (og ikke efter $\dot{t}$).

Flere lemmaer er der ikke i denne rapport; en optælling viser, at der i alt er 13 lemmaer.

[6] [EqualTransitivity $\stackrel{\text{pyk}}{=}$ "equal transitivity"]

[7] [Mendelson3.2(d) $\stackrel{\text{pyk}}{=}$ "mendelson three two d"]

[8] [Mendelson3.2(f) $\stackrel{\text{pyk}}{=}$ "mendelson three two f"]

[9] [Mendelson3.2(g) $\stackrel{\text{pyk}}{=}$ "mendelson three two g"]

[10] [Mendelson3.2(f)Base $\stackrel{\text{pyk}}{=}$ "mendelson three two f base"]

[11] [Mendelson3.2(f)Indu $\stackrel{\text{pyk}}{=}$ "mendelson three two f induction"]

[12] [Mendelson3.2(g)Base $\stackrel{\text{pyk}}{=}$ "mendelson three two g base"]

[13] [Mendelson3.2(g)Indu $\stackrel{\text{pyk}}{=}$ "mendelson three two g induction"]

[14] [PlusCommutativityBase $\stackrel{\text{pyk}}{=}$ "plus commutativity base"]

[15] [PlusCommutativityIndu $\stackrel{\text{pyk}}{=}$ "plus commutativity induction"]

4.2 Afledte lemmaer

Som navnet antyder, er de afledte lemmaer afledt fra de lemmaer, som jeg præsenterede i afsnit 4.1. [16] Jeg vil i alt definere 6 afledte lemmaer. De kan opdeles i tre undergrupper: Regellemmaer, betingede regellemmaer og navneafledte lemmaer.

4.2.1 Regellemmaer

Som vi senere vil se i afsnit 5.3, kan vi altid bevise
$\mathcal{A} \vdash \mathcal{B}$,
hvis vi i forvejen har et bevis for:
$\mathcal{A} \Rrightarrow \mathcal{B}$.
Ved et "regellemma" forstår jeg et lemma af formen $\mathcal{A} \vdash \mathcal{B}$, som er afledt fra et lemma af formen $\mathcal{A} \Rrightarrow \mathcal{B}$. Når et lemma skal anvendes i beviser, er det som regel lettere at arbejde med regellemmaet end med det tilsvarende moderlemma. Det gælder også for de tre lemmaer EqualSymmetry, EqualTransitivity og Mendelson3.2(d) fra afsnit 4.1. Derfor vil jeg bevise de tilsvarende tre regellemmaer:

[S′ **lemma** Mendelson3.2(d)Rule:
$\forall \mathcal{T}\colon \forall \mathcal{R}\colon \forall \mathcal{S}\colon \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{S} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{S}$] [17]

[S′ **lemma** EqualTransitivityRule:
$\forall \mathcal{T}\colon \forall \mathcal{R}\colon \forall \mathcal{S}\colon \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{R} \overset{\mathrm{p}}{=} \mathcal{S} \vdash \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{S}$] [18]

[S′ **lemma** EqualSymmetryRule: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{R} \overset{\mathrm{p}}{=} \mathcal{T}$] [19]

Jeg kunne også have bevist regellemmaer svarende til Mendelson3.2(f)Indu, Mendelson3.2(g)Indu og PlusCommutativityIndu. Imidlertid har jeg valgt kun at bevise de tre regellemmaer, som kan bruges i beviset for PlusCommutativity. Regellemmaer har ikke megen værdi i sig selv; de er et middel til at gøre beviser kortere og mere overskuelige.

4.2.2 Betingede regellemmaer

Et yderligere raffinement er at gå fra regellemmaets form:
$\mathcal{A} \vdash \mathcal{B}$
til den betingede form:
$\mathcal{C} \Rrightarrow \mathcal{A} \vdash \mathcal{C} \Rrightarrow \mathcal{B}$
Ved et "betinget regellemma" forstår jeg et afledt lemma, der har denne form. Jeg vil bevise de følgende to betingede regellemmaer:

[16] Jeg vil bruge ordet "moderlemma" om et lemma, ud fra hvilket man kan aflede andre lemmaer.

[17] [Mendelson3.2(d)Rule $\overset{\mathrm{pyk}}{=}$ "mendelson three two d rule"]

[18] [EqualTransitivityRule $\overset{\mathrm{pyk}}{=}$ "equal transitivity rule"]

[19] [EqualSymmetryRule $\overset{\mathrm{pyk}}{=}$ "equal symmetry rule"]

[S′ **lemma** EqualTransitivityCondRule:
$\forall\mathcal{A}\colon\forall\mathcal{T}\colon\forall\mathcal{R}\colon\forall\mathcal{S}\colon \mathcal{A} \dot{\Rightarrow} \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{R} \overset{\mathrm{p}}{=} \mathcal{S} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{S}$] [20]

[S′ **lemma** Mendelson3.2(d)CondRule:
$\forall\mathcal{A}\colon\forall\mathcal{T}\colon\forall\mathcal{R}\colon\forall\mathcal{S}\colon \mathcal{A} \dot{\Rightarrow} \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{S} \overset{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{T} \overset{\mathrm{p}}{=} \mathcal{S}$] [21]

Jeg beviser netop disse to betingede regellemmaer, fordi de kan bruges i beviset for PlusCommutativity. Ligesom i afsnit 4.2.1 gælder det, at der ikke er nogen selvstændig grund til at bevise betingede regellemmaer.

4.2.3 Navneafledte lemmaer

Den sidste kategori af afledte lemmaer indeholder kun et enkelt lemma. Det er defineret som følger:

[S′ **lemma** Mendelson3.2(g)Switch: $\dot{r}' \dot{+} \dot{t} \overset{\mathrm{p}}{=} (\dot{r} \dot{+} \dot{t})'$] [22]

Som man kan se, er dette lemma helt identisk med lemma Mendelson3.2(g) fra afsnit 4.1 — bortset fra at der er byttet om på de to Peano variable $\dot{r}$ og $\dot{t}$. Derfor kalder jeg Mendelson3.2(g)Switch for et “navneafledt lemma”. Det viser sig, at det er denne form af Mendelson3.2(g), som vi får brug for, når vi skal vise PlusCommutativity. Dette er et eksempel på ulempen ved at arbejde med Peano variable frem for meta-variable: Vi kan ikke abstrahere fra variablenes navne. Man kunne naturligvis løse problemet ved at bevise den brugbare form til at begynde med. Jeg har valgt løsningen med Mendelson3.2(g)Switch for at se, hvor meget ekstra arbejde det kræver at ændre variabelnavnene i et tilfælde som dette.

4.3 Makroer

Ved en makro forstår jeg en generel hjælpesætning, der kan bruges i beviset for mange forskellige slags sætninger. I denne rapport er der i alt 9 makroer. Jeg har alene defineret og bevist disse makroer for at bruge dem i beviset for PlusCommutativity; alligevel er de fleste af makroerne generelle nok til, at man også kan bruge dem i andre sammenhænge.

I det følgende vil jeg for hver enkelt makro præsentere den formelle definition og forklare makroens indhold.

[S′ **lemma** Weakening: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon \mathcal{A} \vdash \mathcal{B} \dot{\Rightarrow} \mathcal{A}$] [23]

Denne makro tillader os at gøre formler betingede. Hvis vi har vist, at formel $\mathcal{A}$ gælder ubetinget, så må vi også gerne antage, at $\mathcal{A}$ gælder under betingelsen $\mathcal{B}$.

[20] [EqualTransitivityCondRule $\overset{\text{pyk}}{=}$ “equal transitivity conditioned rule”]

[21] [Mendelson3.2(d)CondRule $\overset{\text{pyk}}{=}$ “mendelson three two d conditioned rule”]

[22] [Mendelson3.2(g)Switch $\overset{\text{pyk}}{=}$ “mendelson three two g rt switch”]

[23] [Weakening $\overset{\text{pyk}}{=}$ “weakening”]. Bemærk at Logiweb kalder alle hjælpesætninger for for “lemmaer” — også dem, som jeg kalder for “makroer”.

[S′ **lemma** DoubleMP′: $\forall \mathcal{C}\colon \forall \mathcal{D}\colon \forall \mathcal{E}\colon \mathcal{C} \Rightarrow \mathcal{D} \Rightarrow \mathcal{E} \vdash \mathcal{C} \vdash \mathcal{D} \vdash \mathcal{E}$] [24]

Makroen DoubleMP′ kan spare os for en enkelt bevislinie i de tilfælde, hvor man ellers ville bruge slutningsreglen MP′ to gange i træk.

[S′ **lemma** ConditionedMP′:
$\forall \mathcal{A}\colon \forall \mathcal{B}\colon \forall \mathcal{C}\colon \mathcal{A} \Rightarrow \mathcal{B} \Rightarrow \mathcal{C} \vdash \mathcal{A} \Rightarrow \mathcal{B} \vdash \mathcal{A} \Rightarrow \mathcal{C}$] [25]

Indholdet af slutningsreglen MP′ er som bekendt, at man kan slutte $\mathcal{C}$ ud fra $\mathcal{B} \Rightarrow \mathcal{C}$ og $\mathcal{B}$. Man kan tænke på makroen ConditionedMP′ som en betinget udgave af MP′: Den siger, at vi har lov til at bruge MP′ under en vilkårlig betingelse $\mathcal{A}$, hvis vi tilføjer $\mathcal{A}$ til begge præmisser og til konklusionen.

[S′ **lemma** DoubleConditionedMP′:
$\forall \mathcal{A}\colon \forall \mathcal{D}\colon \forall \mathcal{E}\colon \forall \mathcal{F}\colon \mathcal{A} \Rightarrow \mathcal{D} \Rightarrow \mathcal{E} \Rightarrow \mathcal{F} \vdash \mathcal{A} \Rightarrow \mathcal{D} \vdash \mathcal{A} \Rightarrow \mathcal{E} \vdash \mathcal{A} \Rightarrow \mathcal{F}$] [26]

Af og til kan man få brug for at benytte makroen ConditionedMP′ to gange i træk. Her kan DoubleConditionedMP′ spare en bevislinie. Denne makro forholder sig altså til ConditionedMP′ på nøjagtigt samme måde, som DoubleMP′ forholder sig til MP′.

[S′ **lemma** ImplyTransitivity: $\forall \mathcal{D}\colon \forall \mathcal{E}\colon \forall \mathcal{F}\colon \mathcal{D} \Rightarrow \mathcal{E} \vdash \mathcal{E} \Rightarrow \mathcal{F} \vdash \mathcal{D} \Rightarrow \mathcal{F}$] [27]

Som navnet antyder, udsiger ImplyTransitivity, at relationen $\Rightarrow$ er transitiv; ud fra $\mathcal{D} \Rightarrow \mathcal{E}$ og $\mathcal{E} \Rightarrow \mathcal{F}$ kan vi slutte $\mathcal{D} \Rightarrow \mathcal{F}$.

[S′ **lemma** PermuteAntecedents:
$\forall \mathcal{D}\colon \forall \mathcal{E}\colon \forall \mathcal{F}\colon \mathcal{D} \Rightarrow \mathcal{E} \Rightarrow \mathcal{F} \vdash \mathcal{E} \Rightarrow \mathcal{D} \Rightarrow \mathcal{F}$] [28]

I en term som $\mathcal{D} \Rightarrow \mathcal{E} \Rightarrow \mathcal{F}$ kaldes $\mathcal{D}$ og $\mathcal{E}$ for antecedenterne, mens $\mathcal{F}$ kaldes for konsekvensen. Indholdet af PermuteAntecedents er, at antecedenternes rækkefølge er ligegyldig; vi kan omskrive $\mathcal{D} \Rightarrow \mathcal{E} \Rightarrow \mathcal{F}$ til $\mathcal{E} \Rightarrow \mathcal{D} \Rightarrow \mathcal{F}$.

[S′ **lemma** AddOne: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{T}' \stackrel{\mathrm{p}}{=} \mathcal{R}'$] [29]

Denne makro er afledt fra aksiomskemaet S2′: Hvis S2′ havde været et lemma, ville jeg have rubriceret AddOne som et regellemma (med S2′ som moderlemma).

[24] [DoubleMP′ $\stackrel{\mathrm{pyk}}{=}$ "double mp prime"]

[25] [ConditionedMP′ $\stackrel{\mathrm{pyk}}{=}$ "conditioned mp prime"]

[26] [DoubleConditionedMP′ $\stackrel{\mathrm{pyk}}{=}$ "double conditioned mp prime"]

[27] [ImplyTransitivity $\stackrel{\mathrm{pyk}}{=}$ "imply transitivity"]

[28] [PermuteAntecedents $\stackrel{\mathrm{pyk}}{=}$ "permute antecedents"]

[29] [AddOne $\stackrel{\mathrm{pyk}}{=}$ "add one"]

[S′ **lemma** SubstitutionMacro:
$\forall \mathcal{Z}\colon \forall \mathcal{C}\colon \forall \mathcal{A}\colon \forall \mathcal{D}\colon \lceil \mathcal{A} \rceil \equiv \langle \lceil \mathcal{D} \rceil | \lceil \mathcal{Z} \rceil := \lceil \mathcal{C} \rceil \rangle \Vdash \mathcal{D} \vdash \mathcal{A}$] [30]

Denne makro siger, at vi kan slutte $\mathcal{A}$ ud fra $\mathcal{D}$, hvis vi kan frembringe $\mathcal{A}$ ud fra $\mathcal{D}$ ved hjælp af en substitution. F.eks. kan vi slutte $\dot{0} \overset{\mathrm{p}}{=} \dot{0}$ ud fra $\dot{t} \overset{\mathrm{p}}{=} \dot{t}$, fordi vi kan få $\dot{0} \overset{\mathrm{p}}{=} \dot{0}$ frem ved at erstatte $\dot{t}$ med $\dot{0}$ i $\dot{t} \overset{\mathrm{p}}{=} \dot{t}$.

[S′ **lemma** InductionMacro: $\forall \mathcal{A}\colon \forall \mathcal{B}\colon \forall \mathcal{C}\colon \forall \mathcal{Z}\colon$
$\lceil \mathcal{A} \rceil \equiv \langle \lceil \mathcal{A} \rceil | \lceil \mathcal{Z} \rceil := \lceil \mathcal{Z} \rceil \rangle \Vdash \mathcal{B} \equiv \langle \mathcal{A} | \mathcal{Z} := \dot{0} \rangle \Vdash \mathcal{C} \equiv \langle \mathcal{A} | \mathcal{Z} := \mathcal{Z}' \rangle \Vdash$
$\mathcal{B} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{C} \vdash \mathcal{A}$] [31]

I et induktionsbevis ligger det meste af arbejdet som regel i at vise basistilfældet og det induktive skridt. Når dette er gjort, mangler man bare at henvise til princippet om matematisk induktion. Dette arbejde er indkapslet i InductionMacro. Med denne makro kan man færdiggøre et induktionsbevis med en enkelt linie, når man først har vist basistilfældet $\mathcal{B}$ og det induktive skridt $\mathcal{A} \dot{\Rightarrow} \mathcal{C}$.

Bemærk iøvrigt den lidt pudsige sidebetingelse $\lceil \mathcal{A} \rceil \equiv \langle \lceil \mathcal{A} \rceil | \lceil \mathcal{Z} \rceil := \lceil \mathcal{Z} \rceil \rangle$, der siger, at $\mathcal{A}$ er lig med $\mathcal{A}$, hvis vi erstatter enhver fri forekomst af Peano variablen $\mathcal{Z}$ i $\mathcal{A}$ med sig selv. Det er klart, at dette er opfyldt for enhver formel $\mathcal{A}$ og enhver Peano variabel $\mathcal{Z}$, så ved første øjekast virker det som en overflødig sidebetingelse. Det er imidlertid nødvendigt eksplicit at fortælle Logiweb, at betingelsen er opfyldt, hvis beviset for InductionMacro skal gå igennem. Dette skyldes, at Logiweb ikke kan evaluere udtryk, der indeholder meta-variable (som allerede omtalt i afsnit 3.2). Ved at formulere meta-udsagnet $\lceil \mathcal{A} \rceil \equiv \langle \lceil \mathcal{A} \rceil | \lceil \mathcal{Z} \rceil := \lceil \mathcal{Z} \rceil \rangle$ som en sidebetingelse, kan vi bruge det direkte i beviset, uden at Logiweb evaluerer det. [32] Når vi skal til at anvende InductionMacro i andre beviser, vil betingelsen altid være opfyldt; så den ekstra sidebetingelse udgør kun et æstetisk problem.

[30] [SubstitutionMacro $\overset{\mathrm{pyk}}{=}$ “substitution macro”]

[31] [InductionMacro $\overset{\mathrm{pyk}}{=}$ “induction macro”]

[32] Det omtalte bevis står i afsnit 5.9.

5 Beviser for makroerne

I dette afsnit vil jeg bevise de 9 makroer, som blev præsenteret i afsnit 4.3. Jeg beviser makroerne i den samme rækkefølge, som blev brugt i dette afsnit.

Tre af makroerne er så generelle, at de kan bruges i beviserne for de andre makroer. I tabel 1 kan man se, hvilke makroer der bruges i beviserne for hvilke andre makroer.

	Weakening	DoubleMP′	ConditionedMP′	DoubleConditionedMP′	ImplyTransitivity	PermuteAntecedents	AddOne	SubstitutionMacro	InductionMacro
Weakening					•	•			
DoubleMP′			•						•
ConditionedMP′				•	•	•			

Tabel 1: Makroer der beviser makroer. En "•" i række X, søjle Y betyder, at makro X bliver brugt i beviset for makro Y.

5.1 Bevis for Weakening

Her er beviset for Weakening:

[S′ **lemma** Weakening: $\forall \mathcal{A} \colon \forall \mathcal{B} \colon \mathcal{A} \vdash \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{A}$]

S′ **proof of** Weakening:

L01:	Arbitrary $\gg$	$\mathcal{A}$	;
L02:	Arbitrary $\gg$	$\mathcal{B}$	;
L03:	Premise $\gg$	$\mathcal{A}$	;
L04:	A1′ $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{A}$	;
L05:	MP′ $\rhd$ L04 $\rhd$ L03 $\gg$	$\mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{A}$	$\square$

Da dette er rapportens første bevis, vil jeg bringe nogle ekstra kommentarer. Oven over beviset har jeg gentaget definitionen af det, der skal bevises; dette er kun for overblikkets skyld — det er ikke en formel nødvendighed. Selve beviset for Weakening består af fem linier, nummereret fra 1 til 5. En bevislinie kan have to former. Den første form er:

```
Argumentation  ≫  Konklusion
```

hvor `Konklusion` er det som linien beviser, mens teksten i `Argumentation` udgør en begrundelse for, at `Konklusion` gælder. F.eks. siger linie 4, at meta-formlen $\mathcal{A} \dot{\Rightarrow} \mathcal{B} \dot{\Rightarrow} \mathcal{A}$ gælder, fordi den kan udledes fra aksiomskemaet A1′ ved substitution. I linie 5 skal argumentationen læses på den måde, at konklusionerne fra linie 4 og 3 bliver brugt som præmisser til slutningsreglen MP′. Den generelle betydning af konstruktionen "$\mathsf{x} \rhd \mathsf{y}$" er, at konklusionen fra y bliver brugt som præmis i forhold til x.

Den anden form, en bevislinie kan have, er:

```
Nøgleord  ≫  Konklusion
```

hvor **`Nøgleord`** er et af de tre ord "Arbitrary", "Premise" eller "Side-condition". Betydningen af ordene "Premise" og "Side-condition" er åbenlys: De angiver, at liniens konklusion indgår som en præmis (hhv. sidebetingelse) i den hjælpesætning, der skal bevises. F.eks. siger bevisets linie 3, at Weakening bruger meta-formlen $\mathcal{A}$ som præmis. Når ordet "Arbitrary" bruges, består konklusionen af en meta-variabel (f.eks. $\mathcal{A}$ i linie 1 og $\mathcal{B}$ i line 2). Meningen med "Arbitrary" er af teknisk karakter: Ideen er at udtrykke, at vi ikke antager noget om den pågældende meta-variabel, og at vi derfor har ret til at binde meta-variablen med en alkvantor i den hjælpesætning, der skal bevises. I det forhåndenværende bevis fungerer de to linier med "Arbitrary" altså som berettigelse for, at Weakening er kvantificeret med "$\forall\mathcal{A}\colon\forall\mathcal{B}\colon$".

Alt dette har drejet sig om den formelle syntaks for et Logiweb bevis. Der er egentlig ikke så meget at sige om selve beviset for Weakening; det er en ganske enkel anvendelse af A1′ og MP′.

5.2 Bevis for DoubleMP′

$$[\mathrm{S}' \;\textbf{lemma}\; \mathrm{DoubleMP}'\colon \forall\mathcal{C}\colon\forall\mathcal{D}\colon\forall\mathcal{E}\colon \mathcal{C} \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{E} \vdash \mathcal{C} \vdash \mathcal{D} \vdash \mathcal{E}]$$

S′ **proof of** DoubleMP′:

L01:	Arbitrary ≫	$\mathcal{C}$	;
L02:	Arbitrary ≫	$\mathcal{D}$	;
L03:	Arbitrary ≫	$\mathcal{E}$	;
L04:	Premise ≫	$\mathcal{C} \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{E}$	;
L05:	Premise ≫	$\mathcal{C}$	;
L06:	Premise ≫	$\mathcal{D}$	;
L07:	MP′ $\rhd$ L04 $\rhd$ L05 ≫	$\mathcal{D} \dot{\Rightarrow} \mathcal{E}$	;
L08:	MP′ $\rhd$ L07 $\rhd$ L06 ≫	$\mathcal{E}$	□

Som nævnt i afsnit 4.3 er ideen med DoubleMP′ at indkapsle to anvendelser af MP′ i en enkelt bevislinie. Det er derfor ganske naturligt, at kernen i beviset for DoubleMP′ netop består af to anvendelser af MP′.

5.3 Bevis for ConditionedMP′

[S′ **lemma** ConditionedMP′: $\forall \mathcal{A} \colon \forall \mathcal{B} \colon \forall \mathcal{C} \colon \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{C} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{C}$]

S′ **proof of** ConditionedMP′:

L01:	Arbitrary $\gg$	$\mathcal{A}$	;
L02:	Arbitrary $\gg$	$\mathcal{B}$	;
L03:	Arbitrary $\gg$	$\mathcal{C}$	;
L04:	Premise $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{C}$	;
L05:	Premise $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B}$	;
L06:	A2′ $\gg$	$(\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{C}) \mathrel{\dot{\Rightarrow}} (\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B}) \mathrel{\dot{\Rightarrow}} \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{C}$	;
L07:	DoubleMP′ $\rhd$ L06 $\rhd$ L04 $\rhd$ L05 $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{C}$	$\Box$

I afsnit 4.3 nævnte jeg, at ConditionedMP′ kunne opfattes som en betinget udgave af MP′. Vi kan også opfatte ConditionedMP′ som et regellemma i forhold til aksiomskemaet A2′: A2′ har formen $\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{B} \mathrel{\dot{\Rightarrow}} \mathcal{C}$, mens ConditionedMP′ har formen $\mathcal{A} \vdash \mathcal{B} \vdash \mathcal{C}$. Dermed kan vi bruge en meget enkel teknik til at bevise ConditionedMP′: Først opskriver vi præmisserne og moderlemmaet, og så bruger vi MP′ én gang for hver præmis. (I dette tilfælde er "moderlemmaet" et aksiomskema, men det gør ingen forskel). Jeg vil også bruge denne teknik, når vi senere hen skal bevise de egentlige regellemmaer.

5.4 Bevis for DoubleConditionedMP′

[S′ **lemma** DoubleConditionedMP′:
$\forall \mathcal{A} \colon \forall \mathcal{D} \colon \forall \mathcal{E} \colon \forall \mathcal{F} \colon \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{D} \mathrel{\dot{\Rightarrow}} \mathcal{E} \mathrel{\dot{\Rightarrow}} \mathcal{F} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{D} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{E} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{F}$]

S′ **proof of** DoubleConditionedMP′:

L01:	Arbitrary $\gg$	$\mathcal{A}$	;
L02:	Arbitrary $\gg$	$\mathcal{D}$	;
L03:	Arbitrary $\gg$	$\mathcal{E}$	;
L04:	Arbitrary $\gg$	$\mathcal{F}$	;
L05:	Premise $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{D} \mathrel{\dot{\Rightarrow}} \mathcal{E} \mathrel{\dot{\Rightarrow}} \mathcal{F}$	;
L06:	Premise $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{D}$	;
L07:	Premise $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{E}$	;
L08:	ConditionedMP′ $\rhd$ L05 $\rhd$ L06 $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{E} \mathrel{\dot{\Rightarrow}} \mathcal{F}$	;
L09:	ConditionedMP′ $\rhd$ L08 $\rhd$ L07 $\gg$	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{F}$	$\Box$

Makroen DoubleConditionedMP′ indkapsler to anvendelser af ConditionedMP′, så beviset er enkelt: Vi anvender ConditionedMP′ to gange.

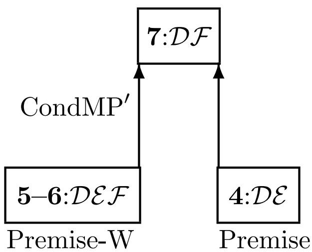
5.5 Bevis for ImplyTransitivity

[S′ **lemma** ImplyTransitivity: $\forall\mathcal{D}\colon\forall\mathcal{E}\colon\forall\mathcal{F}\colon\mathcal{D}\dot{\Rightarrow}\mathcal{E}\vdash\mathcal{E}\dot{\Rightarrow}\mathcal{F}\vdash\mathcal{D}\dot{\Rightarrow}\mathcal{F}$]

S′ **proof of** ImplyTransitivity:

L01:	Arbitrary $\gg$	$\mathcal{D}$	;
L02:	Arbitrary $\gg$	$\mathcal{E}$	;
L03:	Arbitrary $\gg$	$\mathcal{F}$	;
L04:	Premise $\gg$	$\mathcal{D}\dot{\Rightarrow}\mathcal{E}$	;
L05:	Premise $\gg$	$\mathcal{E}\dot{\Rightarrow}\mathcal{F}$	;
L06:	Weakening $\rhd$ L05 $\gg$	$\mathcal{D}\dot{\Rightarrow}\mathcal{E}\dot{\Rightarrow}\mathcal{F}$	;
L07:	ConditionedMP′ $\rhd$ L06 $\rhd$ L04 $\gg$	$\mathcal{D}\dot{\Rightarrow}\mathcal{F}$	$\square$

Hidtil har beviserne været så enkle, at det har været let at forklare deres struktur med et par sætninger. Denne fremgangsmåde er ikke holdbar for de mere komplicerede beviser. I disse tilfælde vil jeg i stedet illustrere bevisernes struktur med træer som dette:



I et "bevistræ" som dette repræsenterer hver knude en eller to af de centrale bevislinier. Bladene svarer til aksiomer eller præmisser, mens bevisets konklusion står i roden. Kanterne er orienteret fra barneknude til forældreknude; betydningen af en kant er, at barneknuden bruges til at bevise forældreknuden. Hver kant er mærket med den systemsætning eller hjælpesætning, der bruges i argumentationen for forældreknuden. [33] Bevislinier, der bruger Weakening i argumentationen, er ikke vist som selvstændige knuder; disse bevislinier er i stedet slået sammen med de linier, som Weakening virker på. Der er altså nogle knuder, som svarer til to bevislinier; disse knuder er mærket med endelsen "-W" (som står for "Weakening").

For at gøre den ovenstående tegning pæn har jeg forkortet navnet "ConditionedMP′" og udeladt tegnet "$\dot{\Rightarrow}$" fra de tre knuder. Den slags friheder vil jeg også tage mig ved de kommende bevistræer; og jeg vil kun gøre opmærksom på den ændrede notation, når jeg skønner det nødvendigt.

Så meget om bevistræer. Jeg vil ikke kommentere selve beviset for

[33]Hvis to kanter fører op til en knude, er kun den ene kant mærket.

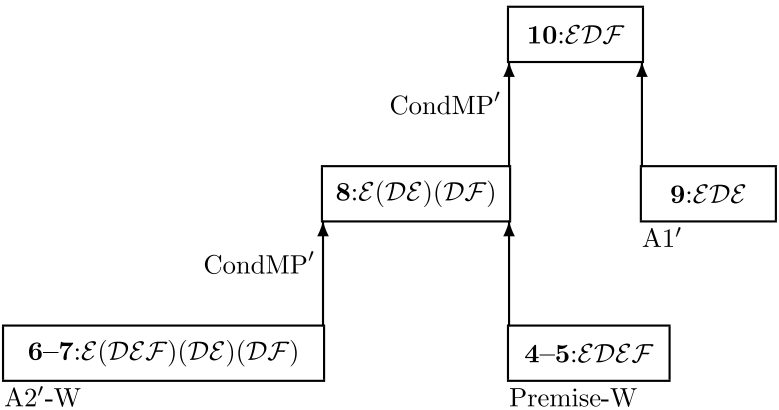
ImplyTransitivity, men lade bevistræet tale for sig selv. Jeg vil heller ikke kommentere nogen af de andre beviser, der er illustrerede med bevistræer.

5.6 Bevis for PermuteAntecedents

[S′ **lemma** PermuteAntecedents: $\forall \mathcal{D} \colon \forall \mathcal{E} \colon \forall \mathcal{F} \colon \mathcal{D} \dot{\Rightarrow} \mathcal{E} \dot{\Rightarrow} \mathcal{F} \vdash \mathcal{E} \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{F}$]

S′ **proof of** PermuteAntecedents:

L01:	Arbitrary $\gg$	$\mathcal{D}$	;
L02:	Arbitrary $\gg$	$\mathcal{E}$	;
L03:	Arbitrary $\gg$	$\mathcal{F}$	;
L04:	Premise $\gg$	$\mathcal{D} \dot{\Rightarrow} \mathcal{E} \dot{\Rightarrow} \mathcal{F}$	;
L05:	Weakening $\rhd$ L04 $\gg$	$\mathcal{E} \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{E} \dot{\Rightarrow} \mathcal{F}$	;
L06:	A2′ $\gg$	$(\mathcal{D} \dot{\Rightarrow} \mathcal{E} \dot{\Rightarrow} \mathcal{F}) \dot{\Rightarrow} (\mathcal{D} \dot{\Rightarrow} \mathcal{E}) \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{F}$	;
L07:	Weakening $\rhd$ L06 $\gg$	$\mathcal{E} \dot{\Rightarrow} (\mathcal{D} \dot{\Rightarrow} \mathcal{E} \dot{\Rightarrow} \mathcal{F}) \dot{\Rightarrow} (\mathcal{D} \dot{\Rightarrow} \mathcal{E}) \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{F}$	;
L08:	ConditionedMP′ $\rhd$ L07 $\rhd$ L05 $\gg$	$\mathcal{E} \dot{\Rightarrow} (\mathcal{D} \dot{\Rightarrow} \mathcal{E}) \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{F}$	;
L09:	A1′ $\gg$	$\mathcal{E} \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{E}$	;
L10:	ConditionedMP′ $\rhd$ L08 $\rhd$ L09 $\gg$	$\mathcal{E} \dot{\Rightarrow} \mathcal{D} \dot{\Rightarrow} \mathcal{F}$	$\square$



5.7 Bevis for AddOne

[S′ **lemma** AddOne: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{T}' \stackrel{\mathrm{p}}{=} \mathcal{R}'$]

S′ **proof of** AddOne:

L01:	Arbitrary $\gg$	$\mathcal{T}$	;
L02:	Arbitrary $\gg$	$\mathcal{R}$	;
L03:	Premise $\gg$	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R}$	;
L04:	S2′ $\gg$	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{T}' \stackrel{\mathrm{p}}{=} \mathcal{R}'$	;
L05:	MP′ $\rhd$ L04 $\rhd$ L03 $\gg$	$\mathcal{T}' \stackrel{\mathrm{p}}{=} \mathcal{R}'$	$\square$

Som nævnt i afsnit 4.3 kan vi opfatte AddOne som et regellemma i forhold til aksiomskemaet S2′. Derfor kan vi vise AddOne med den samme enkle teknik, som vi så første gang i beviset for ConditionedMP′ (afsnit 5.3): Vi skriver præmissen og moderlemmaet op, og vi anvender MP′ én gang for hver præmis (i dette tilfælde kun en enkelt gang).

5.8 Bevis for SubstitutionMacro

[S′ **lemma** SubstitutionMacro:
$\forall \mathcal{Z}\colon \forall \mathcal{C}\colon \forall \mathcal{A}\colon \forall \mathcal{D}\colon \lceil \mathcal{A} \rceil \equiv \langle \lceil \mathcal{D} \rceil | \lceil \mathcal{Z} \rceil := \lceil \mathcal{C} \rceil \rangle \Vdash \mathcal{D} \vdash \mathcal{A}$]

S′ **proof of** SubstitutionMacro:

L01:	Arbitrary $\gg$	$\mathcal{Z}$	;
L02:	Arbitrary $\gg$	$\mathcal{C}$	;
L03:	Arbitrary $\gg$	$\mathcal{A}$	;
L04:	Arbitrary $\gg$	$\mathcal{D}$	;
L05:	Side-condition $\gg$	$\lceil \mathcal{A} \rceil \equiv \langle \lceil \mathcal{D} \rceil | \lceil \mathcal{Z} \rceil := \lceil \mathcal{C} \rceil \rangle$	;
L06:	Premise $\gg$	$\mathcal{D}$	;
L07:	Gen′ $\rhd$ L06 $\gg$	$\dot{\forall} \mathcal{Z}\colon \mathcal{D}$	;
L08:	A4′ $\unrhd$ L05 $\gg$	$\dot{\forall} \mathcal{Z}\colon \mathcal{D} \dot{\Rightarrow} \mathcal{A}$	;
L09:	MP′ $\rhd$ L08 $\rhd$ L07 $\gg$	$\mathcal{A}$	$\square$

Kernen i dette bevis er linie 8. Konklusionen [$\dot{\forall} \mathcal{Z}\colon \mathcal{D} \dot{\Rightarrow} \mathcal{A}$] i denne linie udtrykker, at den generelle formel $\mathcal{D}$ medfører den mindre generelle formel $\mathcal{A}$. Da vi har antaget $\mathcal{D}$, er det let at konkludere $\mathcal{A}$ i linie 9.

I argumentationen i linie 8 benytter vi A4′, der som nævnt i afsnit 3.2 kræver, at en sidebetingelse skal være opfyldt. Derfor bruger vi konstruktionen “x $\unrhd$ y”, som markerer, at konklusionen fra y kan bruges som sidebetingelse for x .

5.9 Bevis for InductionMacro

[S′ **lemma** InductionMacro: $\forall\mathcal{A}\colon\forall\mathcal{B}\colon\forall\mathcal{C}\colon\forall\mathcal{Z}\colon$
$\lceil\mathcal{A}\rceil\equiv\langle\lceil\mathcal{A}\rceil|\lceil\mathcal{Z}\rceil := \lceil\mathcal{Z}\rceil\rangle \Vdash \mathcal{B}\equiv\langle\mathcal{A}|\mathcal{Z} := \dot{0}\rangle \Vdash \mathcal{C}\equiv\langle\mathcal{A}|\mathcal{Z} := \mathcal{Z}'\rangle \Vdash$
$\mathcal{B} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{C} \vdash \mathcal{A}$]

S′ **proof of** InductionMacro:

L01:	Arbitrary $\gg$	$\mathcal{A}$	;
L02:	Arbitrary $\gg$	$\mathcal{B}$	;
L03:	Arbitrary $\gg$	$\mathcal{C}$	;
L04:	Arbitrary $\gg$	$\mathcal{Z}$	;
L05:	Side-condition $\gg$	$\lceil\mathcal{A}\rceil\equiv\langle\lceil\mathcal{A}\rceil|\lceil\mathcal{Z}\rceil := \lceil\mathcal{Z}\rceil\rangle$	;
L06:	Side-condition $\gg$	$\mathcal{B}\equiv\langle\mathcal{A}|\mathcal{Z} := \dot{0}\rangle$	;
L07:	Side-condition $\gg$	$\mathcal{C}\equiv\langle\mathcal{A}|\mathcal{Z} := \mathcal{Z}'\rangle$	;
L08:	Premise $\gg$	$\mathcal{B}$	;
L09:	Premise $\gg$	$\mathcal{A} \dot{\Rightarrow} \mathcal{C}$	;
L10:	Gen′ $\rhd$ L09 $\gg$	$\dot{\forall}\mathcal{Z}\colon(\mathcal{A} \dot{\Rightarrow} \mathcal{C})$	;
L11:	S9′ $\unrhd$ L06 $\unrhd$ L07 $\gg$	$\mathcal{B} \dot{\Rightarrow} \dot{\forall}\mathcal{Z}\colon(\mathcal{A} \dot{\Rightarrow} \mathcal{C}) \dot{\Rightarrow} \dot{\forall}\mathcal{Z}\colon\mathcal{A}$	;
L12:	DoubleMP′$\rhd$L11$\rhd$L08$\rhd$L10 $\gg$	$\dot{\forall}\mathcal{Z}\colon\mathcal{A}$	;
L13:	A4′ $\unrhd$ L05 $\gg$	$\dot{\forall}\mathcal{Z}\colon\mathcal{A} \dot{\Rightarrow} \mathcal{A}$	;
L14:	MP′ $\rhd$ L13 $\rhd$ L12 $\gg$	$\mathcal{A}$	$\square$

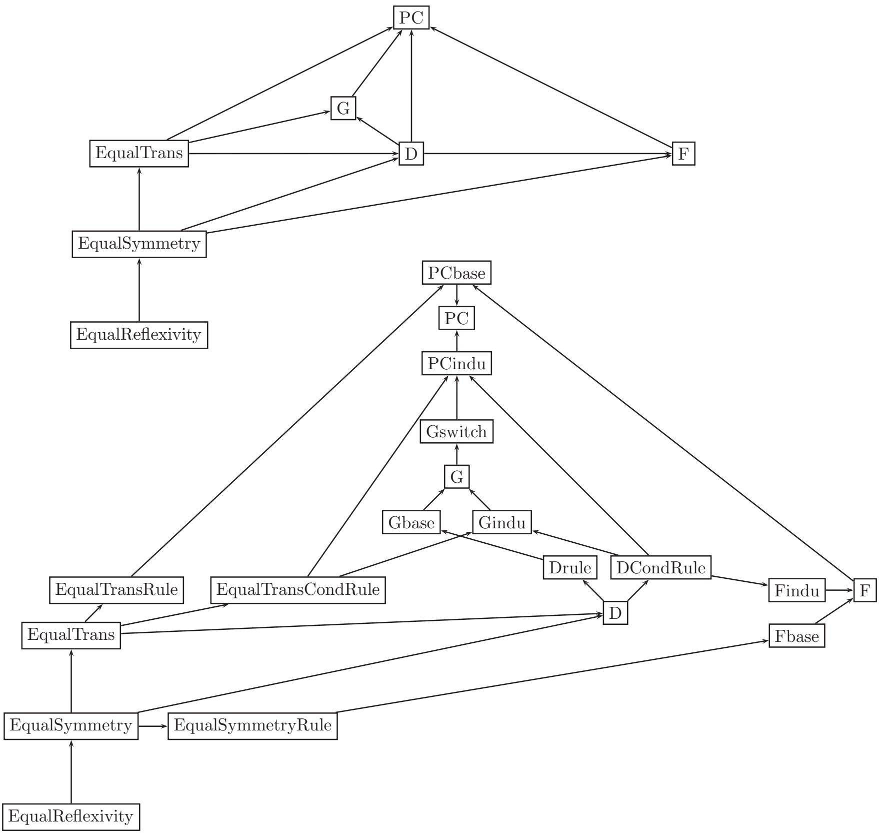
Beviset for InductionMacro er langt, men det er ikke så kompliceret, at jeg vil illustrere det med et bevistræ. Kernen i beviset er de følgende tre blokke:

- Linie 8, der fortæller os, at vi har vist basistilfældet.
- Linie 9–10, der fortæller os, at også det induktive skridt er vist.
- Linie 11, der formulerer princippet om matematisk induktion.

I linie 12 binder vi disse blokke sammen med en DoubleMP′, hvilket giver os konklusionen $\forall\mathcal{Z}\colon\mathcal{A}$. I de sidste to linier fjerner vi Peano alkvantoren herfra, hvorved vi når frem til målet: $\mathcal{A}$.

6 Beviser for lemmaerne

Jeg beviser lemmaerne i den samme rækkefølge, som bliver brugt i [3]. Figur 1 giver en oversigt over, hvordan lemmaerne bruges til at bevise hinanden.



Figur 1: Den overordnede struktur af beviset for PlusCommutativity. En pil fra lemma A til lemma B betyder, at A bliver brugt i beviset for B. Den øverste graf viser ikke de afledte lemmaer, og heller ikke basistilfældene og de induktive skridt i de 3 induktionsbeviser. Den nederste graf har disse detaljer med. For at gøre tegningen pæn er nogle af lemmaernes navne forkortede. “PC” står for “PlusCommutativity”; de øvrige forkortelser skulle være umiddelbart forståelige.

De nedenstående tabeller illustrerer, hvordan makroerne bruges i beviserne for lemmaerne.

	EqualReflexivity	EqualSymmetry	EqualSymmetryRule	EqualTransitivity	EqualTransitivityRule	EqualTransitivityCondRule	Mendelson3.2(d)	Mendelson3.2(d)Rule	Mendelson3.2(d)CondRule
Weakening				•		•			•
DoubleMP′	•				•			•	
ConditionedMP′				•					
DoubleConditionedMP′						•			•
ImplyTransitivity							•		
PermuteAntecedents		•					•		

	Mendelson3.2(f)Base	Mendelson3.2(f)Indu	Mendelson3.2(f)	Mendelson3.2(g)Base	Mendelson3.2(g)Indu	Mendelson3.2(g)	Mendelson3.2(g)Switch	PlusCommutativityBase	PlusCommutativityIndu	PlusCommutativity
Weakening		•			•				•	
AddOne				•	•					
SubstitutionMacro	•						•			
InductionMacro			•			•				•

Tabel 2: Makroer der beviser lemmaer. En "•" i række X, søjle Y betyder, at makro X bliver brugt i beviset for lemma Y. For at gøre tegningen pæn er informationerne fordelt på to tabeller.

6.1 Bevis for EqualReflexivity

[S′ **lemma** EqualReflexivity: $\forall \mathcal{T} \colon \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{T}$]

S′ **proof of** EqualReflexivity:

L01:	Arbitrary ≫	$\mathcal{T}$	;
L02:	S5′ ≫	$\mathcal{T} \dot{+} \dot{0} \stackrel{\mathrm{p}}{=} \mathcal{T}$	;
L03:	S1′ ≫	$\mathcal{T} \dot{+} \dot{0} \stackrel{\mathrm{p}}{=} \mathcal{T} \dot{\Rightarrow} \mathcal{T} \dot{+} \dot{0} \stackrel{\mathrm{p}}{=} \mathcal{T} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{T}$	;
L04:	DoubleMP′ ⊳ L03 ⊳ L02 ⊳ L02 ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{T}$	□

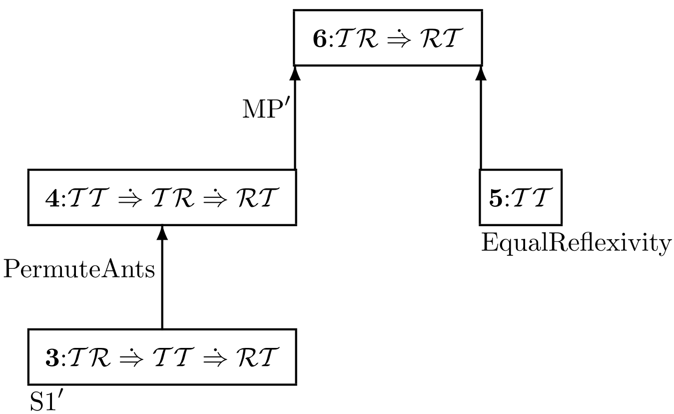
Beviset for EqualReflexivity er så simpelt, at jeg kun vil give en enkelt kommentar: Ideen med aksiomskemaet S5′ er egentlig, at det skal være med til at definere betydningen af addition i Peano aritmetik; men som det kan ses af beviset, kan S5′ også bruges til at bevise lemmaer om $\stackrel{\mathrm{p}}{=}$.

6.2 Bevis for EqualSymmetry og heraf afledt lemma

[S′ **lemma** EqualSymmetry: $\forall \mathcal{T} \colon \forall \mathcal{R} \colon \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{T}$]

S′ **proof of** EqualSymmetry:

L01:	Arbitrary ≫	$\mathcal{T}$	;
L02:	Arbitrary ≫	$\mathcal{R}$	;
L03:	S1′ ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{T} \dot{\Rightarrow} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{T}$	;
L04:	PermuteAntecedents ⊳ L03 ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{T} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{T}$	;
L05:	EqualReflexivity ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{T}$	;
L06:	MP′ ⊳ L04 ⊳ L05 ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{T}$	□



I det ovenstående bevistræ er alle forekomster af tegnet “$\dot{\Rightarrow}$” taget med; til gengæld er “$\stackrel{\mathrm{p}}{=}$” underforstået — f.eks. skal “$\mathcal{T}\mathcal{R}$” læses som “$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R}$”. Jeg vil bruge den samme notation i afsnit 6.3 og 6.4.

[S′ **lemma** EqualSymmetryRule: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \mathcal{T} \stackrel{p}{=} \mathcal{R} \vdash \mathcal{R} \stackrel{p}{=} \mathcal{T}$]

S′ **proof of** EqualSymmetryRule:

L01:	Arbitrary ≫	$\mathcal{T}$	;
L02:	Arbitrary ≫	$\mathcal{R}$	;
L03:	Premise ≫	$\mathcal{T} \stackrel{p}{=} \mathcal{R}$	;
L04:	EqualSymmetry ≫	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{p}{=} \mathcal{T}$	;
L05:	MP′ ⊳ L04 ⊳ L03 ≫	$\mathcal{R} \stackrel{p}{=} \mathcal{T}$	□

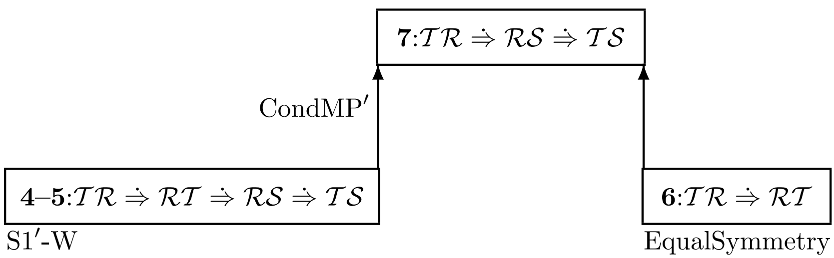
Beviset for dette regellemma bruger den samme teknik, som vi så første gang i afsnit 5.3.

6.3 Bevis for EqualTransitivity og heraf afledte lemmaer

[S′ **lemma** EqualTransitivity: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \forall \mathcal{S}\colon \mathcal{T} \stackrel{p}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{p}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{p}{=} \mathcal{S}$]

S′ **proof of** EqualTransitivity:

L01:	Arbitrary ≫	$\mathcal{T}$	;
L02:	Arbitrary ≫	$\mathcal{R}$	;
L03:	Arbitrary ≫	$\mathcal{S}$	;
L04:	S1′ ≫	$\mathcal{R} \stackrel{p}{=} \mathcal{T} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{p}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	;
L05:	Weakening ⊳ L04 ≫	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}}$ $\mathcal{R} \stackrel{p}{=} \mathcal{T} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{p}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	;
L06:	EqualSymmetry ≫	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{p}{=} \mathcal{T}$	;
L07:	ConditionedMP′ ⊳ L05 ⊳ L06 ≫	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{p}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	□



[S′ **lemma** EqualTransitivityRule: $\forall\mathcal{T}\colon\forall\mathcal{R}\colon\forall\mathcal{S}\colon \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S} \vdash \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$]

S′ **proof of** EqualTransitivityRule:

L01:	Arbitrary ≫	$\mathcal{T}$	;
L02:	Arbitrary ≫	$\mathcal{R}$	;
L03:	Arbitrary ≫	$\mathcal{S}$	;
L04:	Premise ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R}$	;
L05:	Premise ≫	$\mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L06:	EqualTransitivity ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L07:	DoubleMP′ ⊳ L06 ⊳ L04 ⊳ L05 ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	□

Beviset for dette regellemma bruger den samme teknik, som vi så første gang i afsnit 5.3.

[S′ **lemma** EqualTransitivityCondRule: $\forall\mathcal{A}\colon\forall\mathcal{T}\colon\forall\mathcal{R}\colon\forall\mathcal{S}\colon$
$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S} \vdash \mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$]

S′ **proof of** EqualTransitivityCondRule:

L01:	Arbitrary ≫	$\mathcal{A}$	;
L02:	Arbitrary ≫	$\mathcal{T}$	;
L03:	Arbitrary ≫	$\mathcal{R}$	;
L04:	Arbitrary ≫	$\mathcal{S}$	;
L05:	Premise ≫	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R}$	;
L06:	Premise ≫	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L07:	EqualTransitivity ≫	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L08:	Weakening ⊳ L07 ≫	$\mathcal{A} \mathrel{\dot{\Rightarrow}}$ $\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{R} \mathrel{\dot{\Rightarrow}} \mathcal{R} \stackrel{\mathrm{p}}{=} \mathcal{S} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L09:	DoubleConditionedMP′ ⊳ L08 ⊳ L05 ⊳ L06 ≫	$\mathcal{A} \mathrel{\dot{\Rightarrow}} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	□

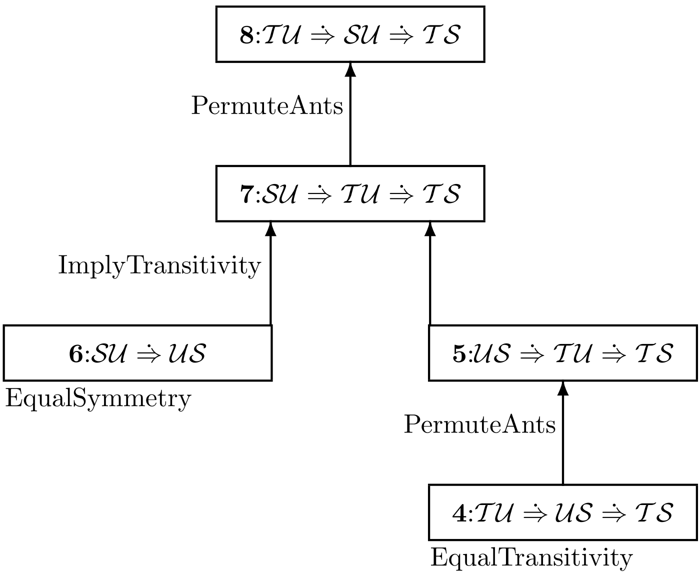
Vi viser et betinget regellemma som EqualTransitivityCondRule på næsten samme måde, som vi viser et ubetinget regellemma. Først skriver vi præmisserne og moderlemmaet op. Derefter gør vi moderlemmaet betinget ved at bruge Weakening. Til slut bruger vi ConditionedMP′ én gang for hver præmis; dette giver os den ønskede konklusion.

6.4 Bevis for Mendelson3.2(d) og heraf afledte lemmaer

[S′ **lemma** Mendelson3.2(f): $\dot{t} \stackrel{\mathrm{p}}{=} \dot{0} \dot{+} \dot{t}$]

S′ **proof of** Mendelson3.2(d):

L01:	Arbitrary $\gg$	$\mathcal{T}$	;
L02:	Arbitrary $\gg$	$\mathcal{U}$	;
L03:	Arbitrary $\gg$	$\mathcal{S}$	;
L04:	EqualTransitivity $\gg$	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{U} \stackrel{\mathrm{p}}{=} \mathcal{S} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L05:	PermuteAntecedents $\rhd$ L04 $\gg$	$\mathcal{U} \stackrel{\mathrm{p}}{=} \mathcal{S} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L06:	EqualSymmetry $\gg$	$\mathcal{S} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{U} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L07:	ImplyTransitivity$\rhd$L06$\rhd$L05 $\gg$		
		$\mathcal{S} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	;
L08:	PermuteAntecedents $\rhd$ L07 $\gg$	$\mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{S} \stackrel{\mathrm{p}}{=} \mathcal{U} \dot{\Rightarrow} \mathcal{T} \stackrel{\mathrm{p}}{=} \mathcal{S}$	$\square$



[S$'$ **lemma** Mendelson3.2(d)Rule: $\forall \mathcal{T}\colon \forall \mathcal{R}\colon \forall \mathcal{S}\colon \mathcal{T} \stackrel{p}{=} \mathcal{R} \vdash \mathcal{S} \stackrel{p}{=} \mathcal{R} \vdash \mathcal{T} \stackrel{p}{=} \mathcal{S}$]

S$'$ **proof of** Mendelson3.2(d)Rule:

L01:	Arbitrary $\gg$	$\mathcal{T}$	;
L02:	Arbitrary $\gg$	$\mathcal{R}$	;
L03:	Arbitrary $\gg$	$\mathcal{S}$	;
L04:	Premise $\gg$	$\mathcal{T} \stackrel{p}{=} \mathcal{R}$	;
L05:	Premise $\gg$	$\mathcal{S} \stackrel{p}{=} \mathcal{R}$	;
L06:	Mendelson3.2(d) $\gg$	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{S} \stackrel{p}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	;
L07:	DoubleMP$'\rhd$L06$\rhd$L04$\rhd$L05 $\gg$	$\mathcal{T} \stackrel{p}{=} \mathcal{S}$	$\square$

Beviset for dette regellemma bruger den samme teknik, som vi så første gang i afsnit 5.3.

[S$'$ **lemma** Mendelson3.2(d)CondRule: $\forall \mathcal{A}\colon \forall \mathcal{T}\colon \forall \mathcal{R}\colon \forall \mathcal{S}\colon$
$\mathcal{A} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{R} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{S} \stackrel{p}{=} \mathcal{R} \vdash \mathcal{A} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$]

S$'$ **proof of** Mendelson3.2(d)CondRule:

L01:	Arbitrary $\gg$	$\mathcal{A}$	;
L02:	Arbitrary $\gg$	$\mathcal{T}$	;
L03:	Arbitrary $\gg$	$\mathcal{R}$	;
L04:	Arbitrary $\gg$	$\mathcal{S}$	;
L05:	Premise $\gg$	$\mathcal{A} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{R}$	;
L06:	Premise $\gg$	$\mathcal{A} \dot{\Rightarrow} \mathcal{S} \stackrel{p}{=} \mathcal{R}$	;
L07:	Mendelson3.2(d) $\gg$	$\mathcal{T} \stackrel{p}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{S} \stackrel{p}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	
L08:	Weakening $\rhd$ L07 $\gg$	$\mathcal{A} \dot{\Rightarrow}$ $\mathcal{T} \stackrel{p}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{S} \stackrel{p}{=} \mathcal{R} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	;
L09:	DoubleConditionedMP$'$ $\rhd$ L08 $\rhd$ L05 $\rhd$ L06 $\gg$	$\mathcal{A} \dot{\Rightarrow} \mathcal{T} \stackrel{p}{=} \mathcal{S}$	$\square$

Beviset for dette betingede regellemma bruger den samme teknik, som vi brugte til at vise EqualTransitivityCondRule i afsnit 6.3.

6.5 Bevis for Mendelson3.2(f)

[S$'$ **lemma** Mendelson3.2(f)Base: $\dot{0} \stackrel{p}{=} \dot{0} \dot{+} \dot{0}$]

S$'$ **proof of** Mendelson3.2(f)Base:

L01:	S5$'$ $\gg$	$\dot{t} \dot{+} \dot{0} \stackrel{p}{=} \dot{t}$	;
L02:	EqualSymmetryRule $\rhd$ L01 $\gg$	$\dot{t} \stackrel{p}{=} \dot{t} \dot{+} \dot{0}$	;
L03:	SubstitutionMacro @ $\dot{t}$ @ $\dot{0}$ $\rhd$ L02 $\gg$	$\dot{0} \stackrel{p}{=} \dot{0} \dot{+} \dot{0}$	$\square$

Dette bevis illustrerer nytten af makroen SubstitutionMacro. Bemærk i øvrigt de to snabel-a'er i linie 3; deres formål er at fortælle Logiwebs bevischecker, at de to meta-variable $\mathcal{Z}$ og $\mathcal{C}$ i SubstitutionMacro skal instantieres til hhv. $\dot{t}$ og $\dot{0}$.

I skrivende stund kan Logiwebs bevischecker ikke selv nå frem til, hvordan $\mathcal{Z}$ og $\mathcal{C}$ skal instantieres (fordi disse meta-variable kun optræder i sidebetingelsen i SubstitutionMacro).

[S′ **lemma** Mendelson3.2(f)Indu: $\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t} \dot{\Rightarrow} \dot{t}' \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t}'$]

S′ **proof of** Mendelson3.2(f)Indu:

L01:	S2′ $\gg$	$\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t} \dot{\Rightarrow} \dot{t}' \stackrel{\text{p}}{=} (\dot{0} \dot{+} \dot{t})'$	;
L02:	S6′ $\gg$	$\dot{0} \dot{+} \dot{t}' \stackrel{\text{p}}{=} (\dot{0} \dot{+} \dot{t})'$	;
L03:	Weakening $\rhd$ L02 $\gg$	$\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t} \dot{\Rightarrow} \dot{0} \dot{+} \dot{t}' \stackrel{\text{p}}{=} (\dot{0} \dot{+} \dot{t})'$	;
L04:	Mendelson3.2(d)CondRule $\rhd$ L01 $\rhd$ L03 $\gg$	$\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t} \dot{\Rightarrow} \dot{t}' \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t}'$	$\square$

Jeg vil ikke illustrere dette bevis med et bevistræ; hele arbejdet gøres med en anvendelse af Mendelson3.2(d)CondRule på to aksiomer (hvoraf det ene er gjort betinget med Weakening).

[S′ **lemma** Mendelson3.2(f): $\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t}$]

S′ **proof of** Mendelson3.2(f):

L01:	Mendelson3.2(f)Base $\gg$	$\dot{0} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{0}$	;
L02:	Mendelson3.2(f)Indu $\gg$	$\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t} \dot{\Rightarrow} \dot{t}' \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t}'$	;
L03:	InductionMacro $\rhd$ L01 $\rhd$ L02 $\gg$	$\dot{t} \stackrel{\text{p}}{=} \dot{0} \dot{+} \dot{t}$	$\square$

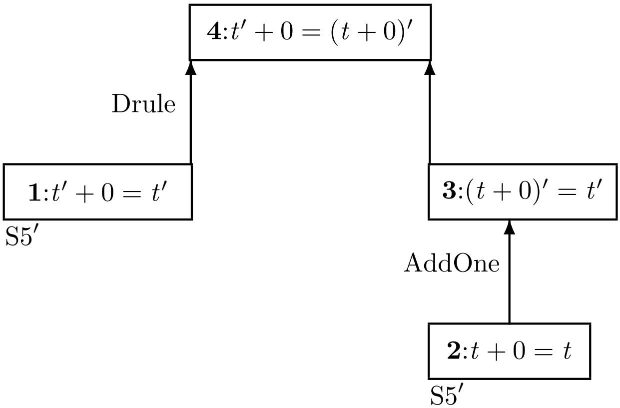
Som det kan ses, gør InductionMacro det meget enkelt at færdiggøre et induktionsbevis.

6.6 Bevis for Mendelson3.2(g) og heraf afledt lemma

[S′ **lemma** Mendelson3.2(g)Base: $\dot{t}' \dot{+} \dot{0} \stackrel{p}{=} (\dot{t} \dot{+} \dot{0})'$]

S′ **proof of** Mendelson3.2(g)Base:

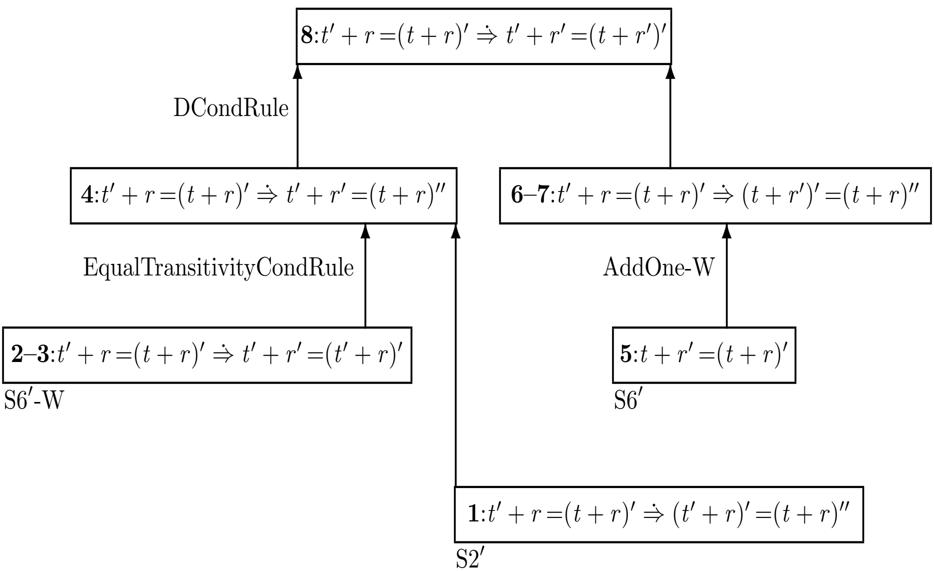
L01:	S5′ $\gg$	$\dot{t}' \dot{+} \dot{0} \stackrel{p}{=} \dot{t}'$	;
L02:	S5′ $\gg$	$\dot{t} \dot{+} \dot{0} \stackrel{p}{=} \dot{t}$	;
L03:	AddOne $\rhd$ L02 $\gg$	$(\dot{t} \dot{+} \dot{0})' \stackrel{p}{=} \dot{t}'$	;
L04:	Mendelson3.2(d)Rule $\rhd$ L01 $\rhd$ L03 $\gg$	$\dot{t}' \dot{+} \dot{0} \stackrel{p}{=} (\dot{t} \dot{+} \dot{0})'$	$\square$



[S$'$ **lemma** Mendelson3.2(g)Indu:
$\dot{t}' \dot{+} \dot{r} \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow} \dot{t}' \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r}')'$]

S$'$ **proof of** Mendelson3.2(g)Indu:

L01:	S2$'$ $\gg$	$\dot{t}' \dot{+} \dot{r} \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow}$ $(\dot{t}' \dot{+} \dot{r})' \overset{\mathrm{p}}{=} ((\dot{t} \dot{+} \dot{r})')'$	;
L02:	S6$'$ $\gg$	$\dot{t}' \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t}' \dot{+} \dot{r})'$	;
L03:	Weakening $\rhd$ L02 $\gg$	$\dot{t}' \dot{+} \dot{r} \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow}$ $\dot{t}' \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t}' \dot{+} \dot{r})'$	;
L04:	EqualTransitivityCondRule $\rhd$ L03 $\rhd$ L01 $\gg$	$\dot{t}' \dot{+} \dot{r} \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow}$ $\dot{t}' \dot{+} \dot{r}' \overset{\mathrm{p}}{=} ((\dot{t} \dot{+} \dot{r})')'$	;
L05:	S6$'$ $\gg$	$\dot{t} \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})'$	;
L06:	AddOne $\rhd$ L05 $\gg$	$(\dot{t} \dot{+} \dot{r}')' \overset{\mathrm{p}}{=} ((\dot{t} \dot{+} \dot{r})')'$	;
L07:	Weakening $\rhd$ L06 $\gg$	$\dot{t}' \dot{+} \dot{r} \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow}$ $(\dot{t} \dot{+} \dot{r}')' \overset{\mathrm{p}}{=} ((\dot{t} \dot{+} \dot{r})')'$	;
L08:	Mendelson3.2(d)CondRule $\rhd$ L04 $\rhd$ L07 $\gg$	$\dot{t}' \dot{+} \dot{r} \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow}$ $\dot{t}' \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r}')'$	$\square$



[S′ **lemma** Mendelson3.2(g): $\dot{t}' \dot{+} \dot{r} \stackrel{p}{=} (\dot{t} \dot{+} \dot{r})'$]

S′ **proof of** Mendelson3.2(g):

L01:	Mendelson3.2(g)Base ≫	$\dot{t}' \dot{+} \dot{0} \stackrel{p}{=} (\dot{t} \dot{+} \dot{0})'$	;
L02:	Mendelson3.2(g)Indu ≫	$\dot{t}' \dot{+} \dot{r} \stackrel{p}{=} (\dot{t} \dot{+} \dot{r})' \dot{\Rightarrow}$ $\dot{t}' \dot{+} \dot{r}' \stackrel{p}{=} (\dot{t} \dot{+} \dot{r}')'$	;
L03:	InductionMacro ▷ L01 ▷ L02 ≫	$\dot{t}' \dot{+} \dot{r} \stackrel{p}{=} (\dot{t} \dot{+} \dot{r})'$	□

Ligesom i afsnit 6.5 har vi her færdiggjort induktionsbeviset ved hjælp af InductionMacro.

[S′ **lemma** Mendelson3.2(g)Switch: $\dot{r}' \dot{+} \dot{t} \stackrel{p}{=} (\dot{r} \dot{+} \dot{t})'$]

S′ **proof of** Mendelson3.2(g)Switch:

L01:	Mendelson3.2(g) ≫	$\dot{t}' \dot{+} \dot{r} \stackrel{p}{=} (\dot{t} \dot{+} \dot{r})'$	;
L02:	SubstitutionMacro @ $\dot{t}$ @ $\dot{s}$ ▷ L01 ≫	$\dot{s}' \dot{+} \dot{r} \stackrel{p}{=} (\dot{s} \dot{+} \dot{r})'$	;
L03:	SubstitutionMacro @ $\dot{r}$ @ $\dot{t}$ ▷ L02 ≫	$\dot{s}' \dot{+} \dot{t} \stackrel{p}{=} (\dot{s} \dot{+} \dot{t})'$	;
L04:	SubstitutionMacro @ $\dot{s}$ @ $\dot{r}$ ▷ L03 ≫	$\dot{r}' \dot{+} \dot{t} \stackrel{p}{=} (\dot{r} \dot{+} \dot{t})'$	□

Som det kan ses, er det ganske enkelt at bytte om på variabelnavnene i et lemma som Mendelson3.2(g), når vi først har makroen SubstitutionMacro til rådighed. Det eneste raffinement er, at vi må introducere en hjælpevariabel $\dot{s}$: Hvis vi f.eks. havde konkluderet $\dot{r}' \dot{+} \dot{r} \stackrel{p}{=} (\dot{r} \dot{+} \dot{r})'$ i linie 2, ville vi ikke længere kunne udtrykke, at lemmaet gælder for to forskellige variable.

6.7 Bevis for PlusCommutativity

[S′ **lemma** PlusCommutativityBase: $\dot{t} \dot{+} \dot{0} \stackrel{p}{=} \dot{0} \dot{+} \dot{t}$]

S′ **proof of** PlusCommutativityBase:

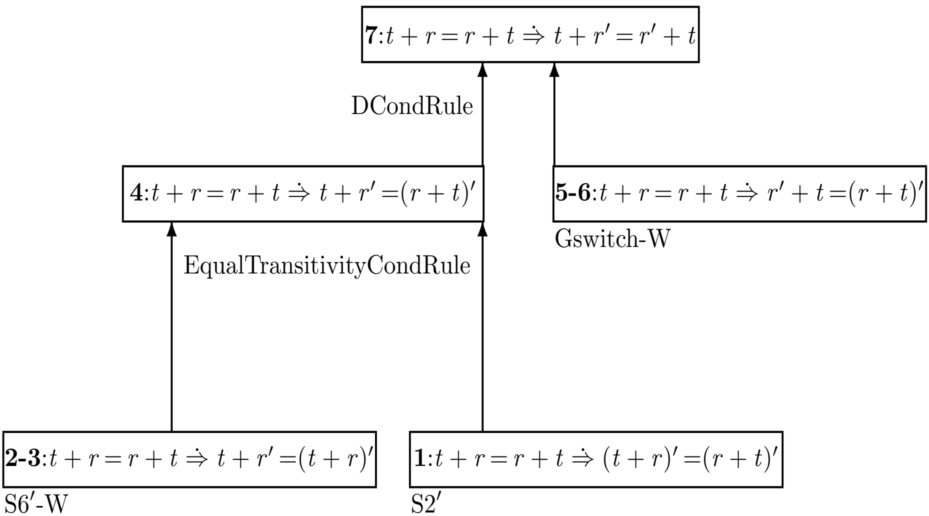
L01:	S5′ ≫	$\dot{t} \dot{+} \dot{0} \stackrel{p}{=} \dot{t}$	;
L02:	Mendelson3.2(f) ≫	$\dot{t} \stackrel{p}{=} \dot{0} \dot{+} \dot{t}$	;
L03:	EqualTransitivityRule ▷ L01 ▷ L02 ≫	$\dot{t} \dot{+} \dot{0} \stackrel{p}{=} \dot{0} \dot{+} \dot{t}$	□

Vi har udelukkende vist Mendelson3.2(f) for at kunne bruge det i dette bevis; men som det ses, har vi heller ikke brug for meget mere for at vise PlusCommutativityBase.

[S′ **lemma** PlusCommutativityIndu: $\dot{t} \dot{+} \dot{r} \overset{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \dot{\Rightarrow} \dot{t} \dot{+} \dot{r}' \overset{\mathrm{p}}{=} \dot{r}' \dot{+} \dot{t}$]

S′ **proof of** PlusCommutativityIndu:

L01:	S2′ $\gg$	$\dot{t} \dot{+} \dot{r} \overset{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \dot{\Rightarrow}$ $(\dot{t} \dot{+} \dot{r})' \overset{\mathrm{p}}{=} (\dot{r} \dot{+} \dot{t})'$	;
L02:	S6′ $\gg$	$\dot{t} \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})'$	;
L03:	Weakening $\rhd$ L02 $\gg$	$\dot{t} \dot{+} \dot{r} \overset{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \dot{\Rightarrow}$ $\dot{t} \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{t} \dot{+} \dot{r})'$	;
L04:	EqualTransitivityCondRule $\rhd$ L03 $\rhd$ L01 $\gg$	$\dot{t} \dot{+} \dot{r} \overset{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \dot{\Rightarrow}$ $\dot{t} \dot{+} \dot{r}' \overset{\mathrm{p}}{=} (\dot{r} \dot{+} \dot{t})'$	;
L05:	Mendelson3.2(g)Switch $\gg$	$\dot{r}' \dot{+} \dot{t} \overset{\mathrm{p}}{=} (\dot{r} \dot{+} \dot{t})'$	;
L06:	Weakening $\rhd$ L05 $\gg$	$\dot{t} \dot{+} \dot{r} \overset{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \dot{\Rightarrow}$ $\dot{r}' \dot{+} \dot{t} \overset{\mathrm{p}}{=} (\dot{r} \dot{+} \dot{t})'$	;
L07:	Mendelson3.2(d)CondRule $\rhd$ L04 $\rhd$ L06 $\gg$	$\dot{t} \dot{+} \dot{r} \overset{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \dot{\Rightarrow} \dot{t} \dot{+} \dot{r}' \overset{\mathrm{p}}{=} \dot{r}' \dot{+} \dot{t}$	□



[S′ **lemma** PlusCommutativity: $\dot{t} \dot{+} \dot{r} \stackrel{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t}$]

S′ **proof of** PlusCommutativity:

L01:	PlusCommutativityBase ≫	$\dot{t} \dot{+} \dot{0} \stackrel{\mathrm{p}}{=} \dot{0} \dot{+} \dot{t}$	;
L02:	PlusCommutativityIndu ≫	$\dot{t} \dot{+} \dot{r} \stackrel{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t} \mathrel{\dot{\Rightarrow}} \dot{t} \dot{+} \dot{r}' \stackrel{\mathrm{p}}{=} \dot{r}' \dot{+} \dot{t}$	;
L03:	InductionMacro ▷ L01 ▷ L02 ≫	$\dot{t} \dot{+} \dot{r} \stackrel{\mathrm{p}}{=} \dot{r} \dot{+} \dot{t}$	□

Den sidste af de 28 hjælpesætninger vises altså ved en simpel anvendelse af InductionMacro. Hermed er det lovede bevis for sætning (1) færdiggjort.

A Navnet på dette dokument

Den følgende definition fastlægger dette dokuments Logiweb-navn:

[peano commutativity $\stackrel{\mathrm{pyk}}{=}$ “peano commutativity”]

B TEX definitioner

[Weakening $\stackrel{\mathrm{tex}}{=}$ “Weakening”]

[DoubleMP′ $\stackrel{\mathrm{tex}}{=}$ “DoubleMP’”]

[ConditionedMP′ $\stackrel{\mathrm{tex}}{=}$ “ConditionedMP’”]

[DoubleConditionedMP′ $\stackrel{\mathrm{tex}}{=}$ “DoubleConditionedMP’”]

[ImplyTransitivity $\stackrel{\mathrm{tex}}{=}$ “ImplyTransitivity”]

[PermuteAntecedents $\stackrel{\mathrm{tex}}{=}$ “PermuteAntecedents”]

[AddOne $\stackrel{\mathrm{tex}}{=}$ “AddOne”]

[SubstitutionMacro $\stackrel{\mathrm{tex}}{=}$ “SubstitutionMacro”]

[InductionMacro $\stackrel{\mathrm{tex}}{=}$ “InductionMacro”]

[EqualReflexivity $\stackrel{\mathrm{tex}}{=}$ “EqualReflexivity”]

[EqualSymmetry $\stackrel{\mathrm{tex}}{=}$ “EqualSymmetry”]

[EqualSymmetryRule $\stackrel{\mathrm{tex}}{=}$ “EqualSymmetryRule”]

[EqualTransitivity $\stackrel{\mathrm{tex}}{=}$ “EqualTransitivity”]

[EqualTransitivityRule $\stackrel{\mathrm{tex}}{=}$ “EqualTransitivityRule”]

[EqualTransitivityCondRule $\stackrel{\text{tex}}{=}$ "EqualTransitivityCondRule"]

[Mendelson3.2(d) $\stackrel{\text{tex}}{=}$ "Mendelson3.2(d)"]

[Mendelson3.2(d)Rule $\stackrel{\text{tex}}{=}$ "Mendelson3.2(d)Rule"]

[Mendelson3.2(d)CondRule $\stackrel{\text{tex}}{=}$ "Mendelson3.2(d)CondRule"]

[Mendelson3.2(f)Base $\stackrel{\text{tex}}{=}$ "Mendelson3.2(f)Base"]

[Mendelson3.2(f)Indu $\stackrel{\text{tex}}{=}$ "Mendelson3.2(f)Indu"]

[Mendelson3.2(f) $\stackrel{\text{tex}}{=}$ "Mendelson3.2(f)"]

[Mendelson3.2(g)Base $\stackrel{\text{tex}}{=}$ "Mendelson3.2(g)Base"]

[Mendelson3.2(g)Indu $\stackrel{\text{tex}}{=}$ "Mendelson3.2(g)Indu"]

[Mendelson3.2(g) $\stackrel{\text{tex}}{=}$ "Mendelson3.2(g)"]

[Mendelson3.2(g)Switch $\stackrel{\text{tex}}{=}$ "Mendelson3.2(g)Switch"]

[PlusCommutativityBase $\stackrel{\text{tex}}{=}$ "PlusCommutativityBase"]

[PlusCommutativityIndu $\stackrel{\text{tex}}{=}$ "PlusCommutativityIndu"]

[PlusCommutativity $\stackrel{\text{tex}}{=}$ "PlusCommutativity"]

C Litteratur

[1] Klaus Grue. A logiweb base page. Logiweb-dokument tilgængelig på hjemmesiden: http://www.diku.dk/~grue/logiweb/20050502/home/grue/base/GRD-2005-06-03-UTC-15-49-33-495567.

[2] Klaus Grue. Peano arithmetic. Logiweb-dokument tilgængelig på hjemmesiden: http://www.diku.dk/~grue/logiweb/20050502/home/grue/peano-axioms/GRD-2005-06-03-UTC-15-58-02-061312.

[3] Elliott Mendelson. *Introduction to Mathematical Logic*. Chapman & Hall, fourth edition, 1997.