Programmeren & Correctheid

Docent: Prof. dr. F.S. de Boer, email: frb@cwi.nl

Literatuur
Verification of Sequential and Concurrent Programs.
Krzysztof R. Apt, Frank S. de Boer, Ernst-Rüdiger Olderog.
Series: Texts in Computer Science. Springer.
3rd ed. 2nd Printing. ISBN: 978-1-84882-744-8.

Te behandelen stof
Course Towards Object-Oriented Program Verification (zie *Preface*: *Outlines of One-Semester Courses* en slides). Uit bovenstaand boek behandelen we de hoofdstukken 2, 3, 4, en 5 onderverdeeld in de volgende blokken B1-3:

	Onderwerp	Secties uit het boek
B1	Partiële Correctheid While Programma's	2.1, 2.2, 2.4, 2.5, 2.7, 3.1, 3.3, 3.4 3.10, 3.11.
B2	Totale Correctheid While Programma's	3.3, 3.4 en 3.8..
B3	Partiële Correctheid Recursieve Programma's	4.1, 4.3, 5.1, 5.2, 5.3.

Evaluatie

- De behandelde stof (zie de paragraaf "Te behandelen stof" hierboven) wordt schriftelijk geëxamineerd .
- Daarnaast wordt er voor elke blok een *deeltentamen* gegeven. In totaal zijn er dus drie deeltentamens (t.z.t. worden de data bekend gemaakt).
    1. **Het eerste deeltentamen vindt plaats op vrijdag 22 Maart, 13.30 - 15.00**. De stof betreft blok B1.
    2. **Het tweede deeltentamen vindt plaats op vrijdag 26 April, 13.30 - 15.00**. De stof betreft blok B2.
    3. **Het derde deeltentamen vindt plaats op vrijdag 17 Mei, 13.45 - 15.30.**

Voor het eindcijfer telt het schriftelijk tentamen voor 75%. Het gemiddelde van de deeltentamens telt voor 25%.

Agenda We maken in de opgaven (en de uitwerkingen) gebruik van de volgende afkortingen:

$$\begin{array}{rll}
\bigwedge_{i=1}^{n} p_i & \text{staat voor} & p_1 \wedge ... \wedge p_n \\
s \leq t & \text{staat voor} & s < t \vee s = t \\
s \leq t < u & \text{staat voor} & s \leq t \wedge t < u \\
s \neq t & \text{staat voor} & \neg(s = t) \\
\exists x, y : p & \text{staat voor} & \exists x : \exists y : p \\
\forall x, y : p & \text{staat voor} & \forall x : \exists y : p \\
\exists x \leq t : p & \text{staat voor} & \exists x : (x \leq t \wedge p) \\
\forall x \leq t : p & \text{staat voor} & \forall x : (x \leq t \rightarrow p) \\
\forall x \in [s : t] : p & \text{staat voor} & \forall x : (s \leq x \leq t \rightarrow p)
\end{array}$$

In de slides maken we ook gebruik van andere voor zich sprekende afkortingen (als $\forall n : s \leq n \leq t : p$ voor $\forall n : (s \leq n \wedge n \leq t) \rightarrow p)$).

15-2 Introduktie (zie slides 1 t/m 21).
Opgaven: Zie slide 18.

22-2 Hoofdstuk 2, secties 2.1, 2.2, , 2.4, 2.5 en 2.7, en sectie 3.1 van hoofdstuk 3. Zie ook slides 22–30.
Opgaven:

1. Bereken de volgende substituties:
   (a) $(x = Y \wedge y = X)[y := z][x := y][z := x]$
   (merk op dat $z := x; x := y; y := z$ de waarden van de integer variabelen x en y verwisselt ("swapt").

   (b) $(a[i] = Y \wedge a[j] = X)[a[j] := z][a[i] := a[j]][z := a[i]]$
2. Stel dat we geen *gebonden* variabelen in p herbenoemen als we een substitutie $[u := t]$ toepassen (zie Sectie 2.7 van het boek). Laat zien dat er dan *ongeldige* instanties zijn van het assignment axioma, d.w.z., vind een postconditie p en een assignment $u := t$ zodat $\{p[u := t]\}\ u := t\ \{p\}$ *niet* waar is.
3. Opgave 3.4 (Boek) en voor welke statements gelden de volgende correctheidsbeweringen.
4. Onderzoek of de assertie

   $$a[a[n]] \leq \textbf{if } a[a[n]] = a[n] \textbf{ then } a[a[n]] \textbf{ else } a[a[a[n]]] \textbf{ fi}$$

   waar is voor alle arrays a (van type **integer** $\rightarrow$ **integer**). Zo niet, geef een tegenvoorbeeld.

5. Welke programma's S waarin de variabele z niet voorkomt voldoen aan

$$\{x = z * z\}S\{x = z\}$$

voor partiële correctheid?

6. Specificeer een programma dat de sectie $a[1 : n]$ van array a van type **integer** $\rightarrow$ **integer** sorteert (let op: hier is ook voor nodig dat het resultaat een permutatie is van de de initiele array a).

7. Opgave 2.2(i) uit het boek en bereken
   (a) $(\exists z : z = x - 1)[x := z]$
   (b) $(\exists y : y = x - 1)[x := z]$

1-3 Hoofdstuk 3, sectie 3.3 (Partial Correctness). Zie ook slides 31 – 39.
Opgaven:

1. (a) Bewijs de correctheidbewering

$$\{\textbf{true}\}\ \textbf{while true do}\ \text{skip}\ \textbf{od}\ \{\textbf{false}\}$$

   (b) Bewijs vervolgens dat

$$\{p\}\ \textbf{while true do}\ \text{skip}\ \textbf{od}\ \{q\}$$

   voor willekeurige preconditie p en postconditie q.

2. Bewijs de correctheidsbewering (array copy)

$$\{i = 0\}\ \textbf{while}\ i < k\ \textbf{do}\ a[i] := b[i]; i := i + 1\ \textbf{od}\ \{\forall 0 \leq n < k : a[n] = b[n]\}$$

8-3 Hoofdstuk 3, sectie 3.4 (Proof Outlines en Partial Correctness) en sectie 3.11 (Case Study: Minimum Section Problem). Zie ook slides 40 – 50.
Opgaven 3.9(ii) (in PW, partial correctness) en 3.10(i) uit het boek, en opgave

1. Bewijs de correctheidsbewering (array copy)

$$\{i = 1\}\ \textbf{while}\ i < k\ \textbf{do}\ a[i] := b[i]; i := i + 1\ \textbf{od}\ \{\forall n : 1 \leq n < k : a[n] = b[n]\}$$

22-3 Hoofdstuk 3, sectie 3.3 (Total Correctness). Zie ook slides 56 – 60.
Werkcollege: Eerste deeltentamen.

29-3 Example: Total Correctness Zero Search (zie slides 61 – 64).
Tijdens het werkcollege is het eerste deeltentamen besproken.

5-4 Hoofdstuk 3, secie 3.3 (Decomposition) en sectie 3.4 (Proof Outlines/Total Correctnes). Zie ook slides 65 – 67.
Opgaven:

1. Gegeven is het volgende programma S voor het berekenen van de "grootste gemene deler".

   while $x \neq y$ **do if** $x > y$
   then $x := x - y$
   else $y := y - x$
   fi
   od

   Geef een preconditie voor welke S termineert en bewijs terminatie.

2. Gegeven het volgende programma S

   $k := 0$;
   while $a[k] < x$ **do**
   $k := k + 1$
   od

   bewijs de totale correctheidsbewering

   $$\{\forall i \geq 0 : a[i] < a[i+1]\}\ S\ \{\mathbf{true}\}$$

12-4 Voor de behandelde stof zie de slides onderaan van Hans-Dieter.

26-4 Hoofdstuk 4 (Recursive Programs), Secties 4.1 (Syntax) en 4.3 (Verification en Partial Correctness). Zie ook slides 68 – 77.
Opgaven:

1. Gegeven de procedure declaratie

   $$P :: \mathbf{if}\ x \neq 0\ \mathbf{then}\ x := x - 1; P\ \mathbf{fi}$$

   bewijs

   (a) $\{x \geq 0\}P\{x = 0\}$
   (b) $\{x < 0\}P\{\mathbf{false}\}$

2. Gegeven de procedure declaratie

   P ::**if** $x \neq y$ **then**
   if $x > y$ **then** $x := x - y; P$
   else $y := y - x; P$
   fi
   fi

   Bewijs $\{x = a \wedge y = b \wedge a > 0 \wedge b > 0\}P\{x = gcd(a, b)\}$.

3-5 Hoofdstuk 4, Sectie 4.4 (Case Study: Binary Search, Partial Correctness). Zie ook slides 78 – 85.
Tijdens het werkcollege is het tweede deeltentamen behandeld.

10-5 Hoofdstuk 5, Secties 5.1 en 5.3 (Partial Correctness: Non-recursive Procedures en Partial Correctness: Recursive Procedures). Zie ook slides 86 – 99.

17-5 Zie slides 100 – 107.

24-5 Zie slides 108 – 112.

Hoorcollege
Programmeren & Correctheid

Hans-Dieter A. Hiep

Agenda

Programma-afkortingen en hulpregels

Afleidbaarheid van hulpregels:

- Invariantie
- Conjunctie

Bewijssysteem partiële correctheid

$$\frac{}{\{\phi\}\textbf{skip}\{\phi\}}\ \textit{skip} \qquad \frac{}{\{\phi[v := E]\}v := E\{\phi\}}\ \textit{assign}$$

$$\frac{\{\phi\}S_1\{\psi\} \quad \{\psi\}S_2\{\chi\}}{\{\phi\}S_1; S_2\{\chi\}}\ \textit{comp}$$

$$\frac{\{\phi \wedge B\}S_1\{\psi\} \quad \{\phi \wedge \neg B\}S_2\{\psi\}}{\{\phi\}\textbf{if } B \textbf{ then } S_1 \textbf{ else } S_2 \textbf{ fi}\{\psi\}}\ \textit{cond}$$

$$\frac{\{\phi \wedge B\}S\{\phi\}}{\{\phi\}\textbf{while } B \textbf{ do } S \textbf{ od}\{\phi \wedge \neg B\}}\ \textit{loop}$$

$$\frac{\phi \to \phi' \quad \{\phi'\}S\{\psi'\} \quad \psi' \to \psi}{\{\phi\}S\{\psi\}}\ \textit{cons}$$

Bewijssysteem totale correctheid

$$\frac{}{[\phi]\textbf{skip}[\phi]}\ \textit{skip} \qquad \frac{}{[\phi[v := E]]v := E[\phi]}\ \textit{assign}$$

$$\frac{[\phi]S_1[\psi] \quad [\psi]S_2[\chi]}{[\phi]S_1; S_2[\chi]}\ \textit{comp}$$

$$\frac{[\phi \wedge B]S_1[\psi] \quad [\phi \wedge \neg B]S_2[\psi]}{[\phi]\textbf{if } B \textbf{ then } S_1 \textbf{ else } S_2 \textbf{ fi}[\psi]}\ \textit{cond}$$

$$\frac{[\phi \wedge B \wedge 0 \leq E = \mathbf{z}]S[\phi \wedge 0 \leq E < \mathbf{z}]}{[\phi]\textbf{while } B \textbf{ do } S \textbf{ od}[\phi \wedge \neg B]}\ \textit{loop2}$$

$$\frac{\phi \to \phi' \quad [\phi']S[\psi'] \quad \psi' \to \psi}{[\phi]S[\psi]}\ \textit{cons}$$

Bewijssysteem totale correctheid

Nieuwe regel:

$$\frac{[\phi \wedge B \wedge 0 \leq E = \mathbf{z}]S[\phi \wedge 0 \leq E < \mathbf{z}]}{[\phi]\textbf{while } B \textbf{ do } S \textbf{ od}[\phi \wedge \neg B]} \; \textit{loop2}$$

waarbij **z** een 'freeze' variabele is, en
de 'bound' E een arithmetische expressie.

Feit
Als $[\phi]S[\psi]$ *bewijsbaar is, dan is* $\{\phi\}S\{\psi\}$ *ook bewijsbaar.*

Omgekeerde geldt niet:
tegenvoorbeeld $S =$ **while true do skip od**
waar $\{\textbf{true}\}S\{\textbf{true}\}$ bewijsbaar is, maar [**true**]S[**true**] niet

Hulpregels

$$\frac{}{\{\phi\}S\{\phi\}}\ \mathit{inv} \qquad \text{waarbij } \mathrm{free}(\phi)\cap\mathrm{change}(S)=\emptyset$$

$$\frac{[\phi_1]S[\psi] \quad [\phi_2]S[\psi]}{[\phi_1\vee\phi_2]S[\psi]}\ \mathit{disj} \qquad \frac{[\phi_1]S[\psi_1] \quad [\phi_2]S[\psi_2]}{[\phi_1\wedge\phi_2]S[\psi_1\wedge\psi_2]}\ \mathit{conj}$$

$$\frac{[\phi]S[\psi]}{[\exists\mathbf{x}.\phi]S[\psi]}\ \mathit{exi} \qquad \text{waarbij } \mathbf{x}\notin \mathrm{occur}(S)\cup\mathrm{free}(\psi)$$

Feit

Hulpregels zijn niet noodzakelijk voor (relatieve) volledigheid.

Hulpregels

Waarom geen axioma voor $[\phi]S[\phi]$?

Neem $S =$ **while true do skip od**

Hulpregels

Voorbeelden

Gegeven het programma S:

$\mathbf{z} := \mathbf{x}; \quad \mathbf{x} := \mathbf{y}; \quad \mathbf{y} := \mathbf{z};$

- *Bewijs* $\{\mathbf{w} = n\}S\{\mathbf{w} = n\}$

$\text{change}(S) = \{\mathbf{z}, \mathbf{x}, \mathbf{y}\}$ *en* $\text{free}(\mathbf{w} = n) = \{\mathbf{w}\}$

$$\frac{}{\{\mathbf{w} = n\}S\{\mathbf{w} = n\}}\ \mathit{inv}$$

Hulpregels

Voorbeelden

Gegeven het programma S:

$\mathbf{z} := \mathbf{x}; \quad \mathbf{x} := \mathbf{y}; \quad \mathbf{y} := \mathbf{z};$

- *Bewijs* $[\mathbf{x} = n \wedge \mathbf{y} = m]S[\mathbf{x} = m \wedge \mathbf{y} = n]$

$$\frac{\begin{array}{cc}\vdots & \vdots \\ [\mathbf{x} = n]S[\mathbf{y} = n] & [\mathbf{y} = m]S[\mathbf{x} = m]\end{array}}{[\mathbf{x} = n \wedge \mathbf{y} = m]S[\mathbf{x} = m \wedge \mathbf{y} = n]}\ \mathit{conj}$$

Hulpregels

Voorbeelden

Gegeven het programma S:

$\mathbf{z} := \mathbf{x}; \quad \mathbf{x} := \mathbf{y}; \quad \mathbf{y} := \mathbf{z};$

- *Bewijs* $[\mathbf{x} = n \vee \mathbf{y} = n]S[\mathbf{x} = n \vee \mathbf{y} = n]$

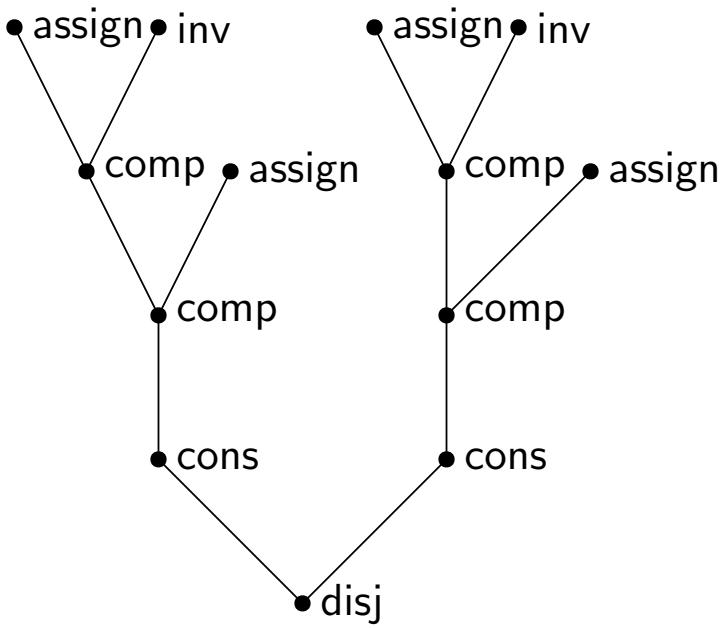
$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ [\mathbf{x} = n]S[\mathbf{y} = n]\end{array}}{[\mathbf{x} = n]S[\mathbf{x} = n \vee \mathbf{y} = n]}\ \textit{cons} \quad \dfrac{\begin{array}{c}\vdots\\ [\mathbf{y} = n]S[\mathbf{x} = n]\end{array}}{[\mathbf{y} = n]S[\mathbf{x} = n \vee \mathbf{y} = n]}\ \textit{cons}}{[\mathbf{x} = n \vee \mathbf{y} = n]S[\mathbf{x} = n \vee \mathbf{y} = n]}\ \textit{disj}$$

Bewijs

Een bewijs is een boomstructuur, waarbij de knooppunten toepassingen zijn van regels en bladeren van axioma's.

Voorbeeld

Bewijs voor $[\mathbf{x} = n \vee \mathbf{y} = n]S[\mathbf{x} = n \vee \mathbf{y} = n]$



Afleidbaarheid

Een standaard bewijs gebruikt **alleen**
axioma's *skip*, *assign* en
regels *comp*, *cond*, *loop2* en *cons*

Een regel is afleidbaar als

- elk bewijs mét die regel

kan worden vertaald naar

- een bewijs zónder die regel (= standaard)

waarbij vertaalde bewijs zelfde conclusie heeft

Bewijsmethoden:

- Knippen & plakken
- (Structurele) inductie

Programma-afkortingen

Afkorting:

$$\textbf{if } B \textbf{ then } S \textbf{ fi} \quad \equiv \quad \textbf{if } B \textbf{ then } S \textbf{ else skip fi}$$

Hulpregel:

$$\frac{[\phi \wedge B]S[\psi] \quad (\phi \wedge \neg B \rightarrow \psi)}{[\phi]\textbf{if } B \textbf{ then } S \textbf{ fi}[\psi]} \; \mathit{cond}'$$

Programma-afkortingen

Aangenomen:
standaard bewijs voor $[\phi \wedge B]S[\psi]$ en $\phi \wedge \neg B \rightarrow \psi$
Doel:
standaard bewijs

$$\dfrac{\dfrac{\vdots}{[\phi \wedge B]S[\psi]} \qquad \dfrac{\dfrac{\vdots}{\phi \wedge \neg B \rightarrow \psi} \quad \dfrac{}{[\psi]\textbf{skip}[\psi]}\ \textit{skip} \qquad \psi \rightarrow \psi}{[\phi \wedge \neg B]\textbf{skip}[\psi]}\ \textit{cons}}{[\phi]\textbf{if } B \textbf{ then } S \textbf{ else skip fi}[\psi]}\ \textit{cond}$$

Programma-afkortingen

Afkorting:

$$\textbf{repeat } S \textbf{ until } B \quad \equiv \quad S; \textbf{while } \neg B \textbf{ do } S \textbf{ od}$$

Hulpregel:

$$\frac{\{\phi\}S\{\psi\} \quad (\psi \wedge \neg B \rightarrow \phi)}{\{\phi\}\textbf{repeat } S \textbf{ until } B\{\psi \wedge B\}} \; \textit{repeat}$$

Programma-afkortingen

Aangenomen:
standaard bewijs voor $\{\phi\}S\{\psi\}$ en $\psi \wedge \neg B \rightarrow \phi$
Doel:
standaard bewijs

$$\dfrac{\{\phi\}\vdots S\{\psi\} \qquad \dfrac{\dfrac{\dfrac{\psi \wedge \neg B \rightarrow \phi \quad \{\phi\}\vdots S\{\psi\}}{\{\psi \wedge \neg B\}S\{\psi\}}\ cons}{\{\psi\}\textbf{while } \neg B \textbf{ do } S \textbf{ od}\{\psi \wedge \neg\neg B\}}\ while \qquad \neg\neg B \equiv B}{\{\psi\}\textbf{while } \neg B \textbf{ do } S \textbf{ od}\{\psi \wedge B\}}\ cons}{\{\phi\}S; \textbf{while } \neg B \textbf{ do } S \textbf{ od}\{\psi \wedge B\}}\ comp$$

Afleidbaarheid

$$\frac{}{\{\phi\}S\{\phi\}}\ \mathit{inv} \qquad \text{waarbij } \mathrm{free}(\phi) \cap \mathrm{change}(S) = \emptyset$$

Lemma

inv is afleidbaar

Bewijs door structurele inductie op S**.**

- Geval **skip**: met axioma *skip*.
- Geval $v := E$: met axioma *assign*,
  $\phi[v := E] = \phi$ vanwege conditie.
- Geval $S = S_1; S_2$:
  $\mathrm{change}(S_1) \subseteq \mathrm{change}(S)$ en $\mathrm{change}(S_2) \subseteq \mathrm{change}(S)$
  pas *inv* toe om $\{\phi\}S_1\{\phi\}$ en $\{\phi\}S_2\{\phi\}$ te bewijzen
  (IH) er bestaan standaard bewijzen,
  pas daarop *comp* toe om $\{\phi\}S\{\phi\}$ te bewijzen

Afleidbaarheid

Bewijs (vervolg).

- Geval $S =$ **if** B **then** S_1 **else** S_2 **fi**:
  change$(S_1) \subseteq$ change(S) en change$(S_2) \subseteq$ change(S)
  pas *inv* toe om $\{\phi\}S_1\{\phi\}$ en $\{\phi\}S_2\{\phi\}$ te bewijzen
  (IH) er bestaan standaard bewijzen,
  pas *cons* toe: bewijs voor $\{\phi \wedge B\}S_1\{\phi\}$ en $\{\phi \wedge \neg B\}S_2\{\phi\}$
  pas ten slotte *cond* toe om $\{\phi\}S\{\phi\}$ te bewijzen

- Geval $S =$ **while** B **do** S' **od**:
  change$(S') \subseteq$ change(S)
  pas *inv* toe om $\{\phi\}S'\{\phi\}$ te bewijzen
  (IH) er bestaat standaard bewijs,
  pas daarop *cons* toe: $\{\phi \wedge B\}S'\{\phi\}$
  pas *loop* toe om $\{\phi\}S\{\phi \wedge \neg B\}$ te bewijzen
  pas *cons* toe: $\{\phi\}S\{\phi\}$

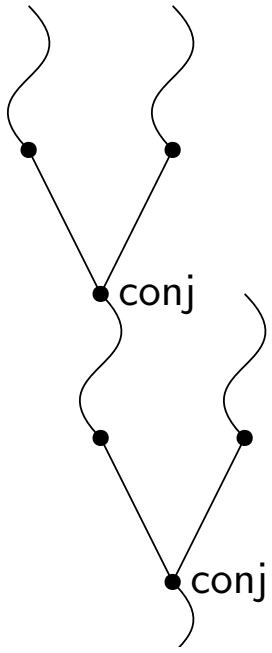
Afleidbaarheid

$$\frac{[\phi_1]S[\psi_1] \quad [\phi_2]S[\psi_2]}{[\phi_1 \wedge \phi_2]S[\psi_1 \wedge \psi_2]}\ \mathit{conj}$$

Lemma

conj is afleidbaar

Bewijs. Idee: bewijsboom bewerken van boven naar onder



Afleidbaarheid

$$\frac{[\phi_1]S[\psi_1] \quad [\phi_2]S[\psi_2]}{[\phi_1 \wedge \phi_2]S[\psi_1 \wedge \psi_2]}\ \mathit{conj}$$

Bewijs (vervolg).
(I) Inductie op hoogte voorkomen *conj* in bewijsboom
Aanname: er is standaard bewijs voor $[\phi_1]S[\psi_1]$ en $[\phi_2]S[\psi_2]$
Doel: geef standaard bewijs voor $[\phi_1 \wedge \phi_2]S[\psi_1 \wedge \psi_2]$
(II) Structurele inductie op programma S

- Geval **skip**:
  uit aanname volgt $\psi_1 = \phi_1$ en $\psi_2 = \phi_2$
  bewijs $[\phi_1 \wedge \phi_2]\mathbf{skip}[\phi_1 \wedge \phi_2]$ met axioma *skip*
- Geval $v := E$:
  uit aanname volgt $\phi_1 = \psi_1[v := E]$ en $\phi_2 = \psi_2[v := E]$
  dus $\phi_1 \wedge \phi_2 = \psi_1[v := E] \wedge \psi_2[v := E] = (\psi_1 \wedge \psi_2)[v := E]$
  bewijs $[(\psi_1 \wedge \psi_2)[v := E]]v := E[\psi_1 \wedge \psi_2]$ met axioma *assign*

Afleidbaarheid

$$\frac{[\phi_1]S[\psi_1] \quad [\phi_2]S[\psi_2]}{[\phi_1 \wedge \phi_2]S[\psi_1 \wedge \psi_2]}\ \mathit{conj}$$

Bewijs (vervolg).

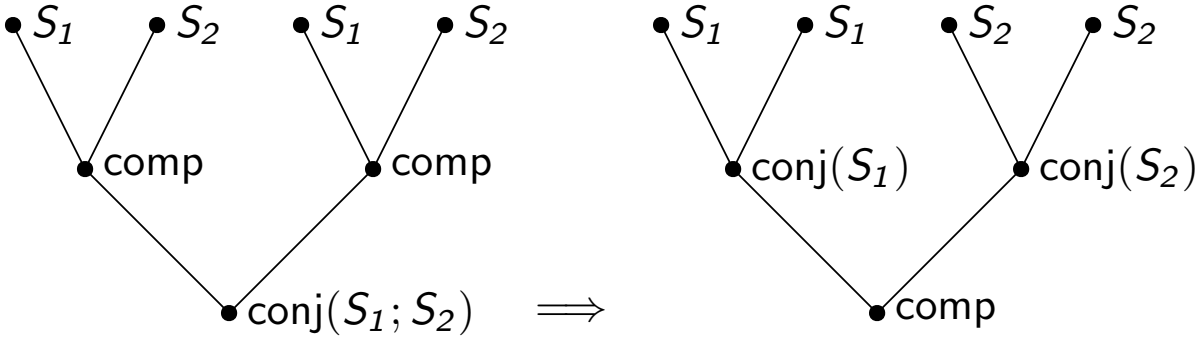
- Geval $S = S_1 ; S_2$:
  uit aanname standaard bewijs voor $[\phi_1]S_1[\chi_1]$ en $[\chi_1]S_2[\psi_1]$
  en standaard bewijs voor $[\phi_2]S_1[\chi_2]$ en $[\chi_2]S_2[\psi_2]$
  construeer bewijs $[\phi_1 \wedge \phi_2]S_1[\chi_1 \wedge \chi_2]$ met *conj*
  construeer bewijs $[\chi_1 \wedge \chi_2]S_2[\psi_1 \wedge \psi_2]$ met *conj*
  (IH) er bestaan standaard bewijzen
  pas daarop *comp* toe om $[\phi_1 \wedge \phi_2]S[\psi_1 \wedge \psi_2]$ te bewijzen

Afleidbaarheid

$$\frac{[\phi_1]S[\psi_1] \quad [\phi_2]S[\psi_2]}{[\phi_1 \wedge \phi_2]S[\psi_1 \wedge \psi_2]}\ \mathit{conj}$$

Bewijs (vervolg).

- Geval $S = S_1; S_2$:



- Andere gevallen vergelijkbaar