

Syntaks og semantik

Lektion 10

18 marts 2008

Operationel semantik

Big vs. small step

At opskrive

Derivationstræer

Fra sidst

- 1 Operationel semantik
- 2 Big vs. small step
- 3 At opskrive en operationel semantik
- 4 Derivationstræer

- **Operationel semantik**: at oversætte et *program* til et *transitionssystem*:
 - konfigurationer: programtilstande
 - transitioner: programskridt
 - slutkonfigurationer: (succesfuld) terminering af programmet
- **Transitionssystemer**: (Γ, \rightarrow, T)
 - konfigurationer Γ , transitioner \rightarrow , slutkonfigurationer T
 - fra nu af: slutkonfigurationer er **terminale**:

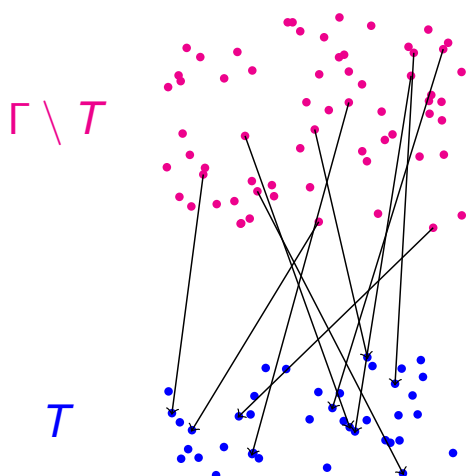
$$\forall \gamma \in T : \neg \exists \gamma' \in \Gamma : \gamma \rightarrow \gamma'$$

- men ikke alle terminale konfigurationer er nødvendigvis slutkonfigurationer! – **deadlock**

3/28

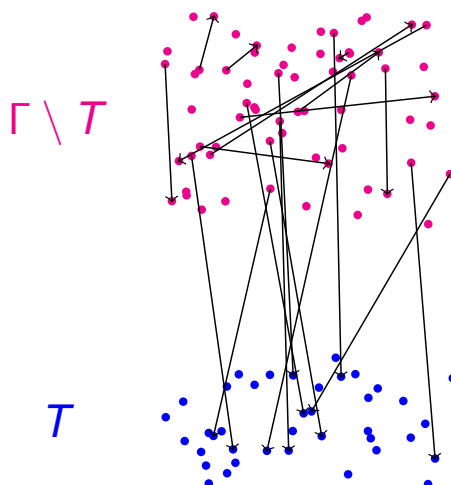
Big-step-semantik:

- at evaluere ting *i ét hug*
- transitioner fra konfigurationer til slutkonfigurationer



Small-step-semantik:

- at evaluere ting *ét skridt ad gangen*
- transitioner fra konfigurationer til konfigurationer og til slutkonfigurationer



4/28

At opskrive en operationel semantik for et programmeringssprog:

1 abstrakt syntaks

- syntaktiske kategorier

$n \in \mathbf{Num}$ – Numeraler

$x \in \mathbf{Var}$ – Variable

$a \in \mathbf{Aud}$ – Aritmetiske udtryk

$b \in \mathbf{Bud}$ – Boolske udtryk

$S \in \mathbf{Kom}$ – Kommandoer

- opbygningsregler

$S ::= x := a \mid \text{skip} \mid S_1; S_2 \mid \text{if } b \text{ then } S_1 \text{ else } S_2$
 $\mid \text{while } b \text{ do } S$

$b ::= a_1 = a_2 \mid a_1 < a_2 \mid \neg b_1 \mid b_1 \wedge b_2 \mid (b_1)$

$a ::= n \mid x \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1)$

5/28

At opskrive en operationel semantik for et programmeringssprog:

1 abstrakt syntaks

- syntaktiske kategorier
- opbygningsregler

2 semantiske mængder og hjælpefunktioner

- værdier af numeraler er elementer i \mathbb{Z}
- funktionen $\mathcal{N} : \mathbf{Num} \rightarrow \mathbb{Z}$ giver værdien af en numeral

6/28

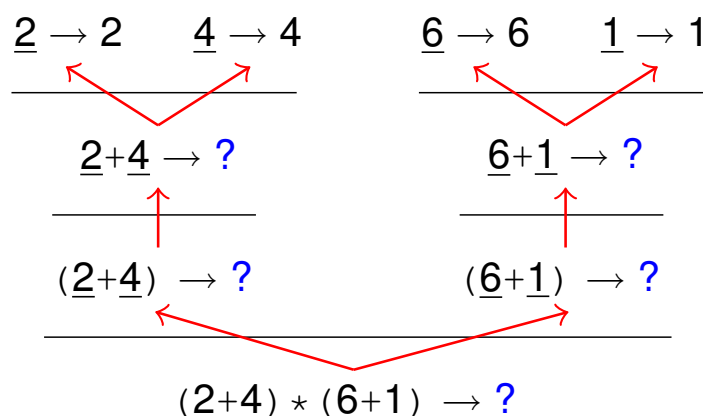
At opskrive en operationel semantik for et programmeringssprog:

- 1 abstrakt syntaks
 - syntaktiske kategorier
 - opbygningsregler
- 2 semantiske mængder og hjælpefunktioner
- 3 transitionssystem(er)
 - konfigurationer og slutkonfigurationer
 $\Gamma = \text{Aud} \cup \mathbb{Z}, T = \mathbb{Z}$
 - transitionsrelationen givet ved *transitionsregler*

f.eks.
$$\frac{a_1 \rightarrow v_1 \quad a_2 \rightarrow v_2}{a_1 + a_2 \rightarrow v} \quad \text{hvor } v = v_1 + v_2$$

7/28

For at vise at en bestemt transition findes i en operationel semantik, konstrueres et **derivationstræ**:



- aksiomer i bladene
- knude k har sønner p_1, p_2, \dots, p_n hvis og kun hvis der er en transitionsregel

$$\frac{p_1, p_2, \dots, p_n}{k}$$
- mekanisk proces \Rightarrow automatisering!

8/28

Operationelle semantikker for **Bims**

- 5 Programtilstande
- 6 Big-step-semantik for aritmetiske udtryk med variable
- 7 Big-step-semantik for boolske udtryk
- 8 Big-step-semantik for **Bims**
- 9 At konstruere et derivationstræ
- 10 Terminering (big-step)
- 11 Small-step-semantik for **Bims**
- 12 Terminering (small-step)
- 13 Ækvivalens af big-step- og small-step-semantikken for **Bims**

9 / 28

Mål: Transitionssystem hvori transitioner beskriver udførelser af **Bims**-kommandoer.

Hvad skal *konfigurationerne* være?

- konfiguration = *programtilstand*
 - programmers opførsel kan afhænge af værdier af variable
- ⇒ programtilstand = de kommandoer vi mangler at udføre
+ værdier af alle variable

Definition 4.1: En *tilstand* er en *partiel* funktion $\mathbf{Var} \rightarrow \mathbb{Z}$.

Definition 4.3: Mængden af alle tilstande kaldes **Tilstande**.

Dvs. **Tilstande** = $\mathbf{Var} \rightharpoonup \mathbb{Z}$. ← mængden af alle *partielle* funktioner fra **Var** til \mathbb{Z}

– konfigurationerne vil være *par af kommandoer og tilstande*:

$\Gamma = \mathbf{Kom} \times \mathbf{Tilstande}$

Aritmetiske udtryk med variable:

Aud: $a ::= n \mid x \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \mid (a_1)$

- big-step-semantik
- semantikken *afhænger* af tilstanden, men *ændrer den ikke*
- ⇒ konfigurationer $\Gamma = \mathbf{Aud} \cup \mathbb{Z}$ (som før!), **men**
transitionssystemet afhænger af tilstanden!
- transitioner skrives $s \vdash a \rightarrow_a v$: i tilstand s kan a evaluere til v
- slutkonfigurationer $T = \mathbb{Z}$ (også som før)

11 / 28

$$[\text{plus}_{\text{bss}}] \quad \frac{s \vdash a_1 \rightarrow_a v_1 \quad s \vdash a_2 \rightarrow_a v_2}{s \vdash a_1 + a_2 \rightarrow_a v} \quad \text{hvor } v = v_1 + v_2$$

$$[\text{minus}_{\text{bss}}] \quad \frac{s \vdash a_1 \rightarrow_a v_1 \quad s \vdash a_2 \rightarrow_a v_2}{s \vdash a_1 - a_2 \rightarrow_a v} \quad \text{hvor } v = v_1 - v_2$$

$$[\text{mult}_{\text{bss}}] \quad \frac{s \vdash a_1 \rightarrow_a v_1 \quad s \vdash a_2 \rightarrow_a v_2}{s \vdash a_1 * a_2 \rightarrow_a v} \quad \text{hvor } v = v_1 \cdot v_2$$

$$[\text{parent}_{\text{bss}}] \quad \frac{s \vdash a_1 \rightarrow_a v_1}{s \vdash (a_1) \rightarrow_a v_1}$$

$$[\text{num}_{\text{bss}}] \quad s \vdash n \rightarrow_a v \quad \text{hvis} \quad \mathcal{N}[[n]] = v$$

$$[\text{var}_{\text{bss}}] \quad s \vdash x \rightarrow_a v \quad \text{hvis} \quad s(x) = v$$

- **syntaksdirigerede**: ethvert sammensat element fra syntaksen optræder som konklusion i en transitionsregel, ethvert basiselement som aksiom
- **kompositionelle**: præmisserne i en regel udtaler sig om de *umiddelbare bestanddele* af elementet i konklusionen

12 / 28

Boolske udtryk:

Bud: $b ::= a_1 = a_2 \mid a_1 < a_2 \mid \neg b_1 \mid b_1 \wedge b_2 \mid (b_1)$

- samme big-step-semantik som før
- bortset fra at alle transitioner nu er af formen $s \vdash b \rightarrow_b tt$ eller $s \vdash b \rightarrow_b ff$
- det gider vi ikke vise igen ...

13/28

Kommandoer i Bims:

$S ::= x := a \mid \text{skip} \mid S_1; S_2 \mid \text{if } b \text{ then } S_1 \text{ else } S_2$
 $\mid \text{while } b \text{ do } S$

- kommandoer kan ændre tilstanden (f.x. kommandoen $x := 2$)
 \Rightarrow skal have tilstanden med i konfigurationerne
- dvs. konfigurationer $\Gamma = \text{Kom} \times \text{Tilstande} \cup \text{Tilstande}$ og slutkonfigurationer $T = \text{Tilstande}$
- skrives $\langle S, s \rangle$ (S kommando, s tilstand)
- (og transitionsrelationen \rightarrow defineres ved transitionsregler; coming up)
- at ændre en tilstand: **Definition 4.4:** Lad $s \in \text{Tilstande}$, $x \in \text{Var}$ og $v \in \mathbb{Z}$. Den opdaterede tilstand $s[x \mapsto v]$ er givet ved

$$s[x \mapsto v](y) = \begin{cases} s(y) & \text{hvis } y \neq x \\ v & \text{hvis } y = x \end{cases}$$

14/28

$[\text{ass}_{\text{bss}}]$	$\langle x := a, s \rangle \rightarrow s[x \mapsto v]$	hvor $s \vdash a \rightarrow_a v$
$[\text{skip}_{\text{bss}}]$	$\langle \text{skip}, s \rangle \rightarrow s$	
$[\text{comp}_{\text{bss}}]$	$\frac{\langle S_1, s \rangle \rightarrow s'' \quad \langle S_2, s'' \rangle \rightarrow s'}{\langle S_1; S_2, s \rangle \rightarrow s'}$	
$[\text{if-sand}_{\text{bss}}]$	$\frac{\langle S_1, s \rangle \rightarrow s'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'}$	hvis $s \vdash b \rightarrow_b tt$
$[\text{if-falsk}_{\text{bss}}]$	$\frac{\langle S_2, s \rangle \rightarrow s'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'}$	hvis $s \vdash b \rightarrow_b ff$
$[\text{while-sand}_{\text{bss}}]$	$\frac{\langle S, s \rangle \rightarrow s'' \quad \langle \text{while } b \text{ do } S, s'' \rangle \rightarrow s'}{\langle \text{while } b \text{ do } S, s \rangle \rightarrow s'}$	hvis $s \vdash b \rightarrow_b tt$
$[\text{while-falsk}_{\text{bss}}]$	$\langle \text{while } b \text{ do } S, s \rangle \rightarrow s$	hvis $s \vdash b \rightarrow_b ff$

15/28

$[\text{while-sand}_{\text{bss}}]$	$\frac{\langle S, s \rangle \rightarrow s'' \quad \langle \text{while } b \text{ do } S, s'' \rangle \rightarrow s'}{\langle \text{while } b \text{ do } S, s \rangle \rightarrow s'}$	hvis $s \vdash b \rightarrow_b tt$
------------------------------------	--	------------------------------------

Dén regel er **ikke kompositionel**: præmissen indeholder ikke kun umiddelbare bestanddele af konklusionens syntakselement

- fordi `while`-løkker er *rekursive*
- reglen skal anvendes *indtil b bliver falsk*
- ellers: *uendelig løkke* – ikke-terminering
- fikspunkt-teori!

Eksempel: Givet kommando

$S = i := 6; \text{ while } i \neq 0 \text{ do } (x := x + i; i := i - 2)$

og tilstanden s ved $s(x) = 5$, konstruer et derivationstræ for at finde en transition $\langle S, s \rangle \rightarrow s'$:

- 1
$$\frac{\langle i := 6, s \rangle \rightarrow s_2 \quad \langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_2 \rangle \rightarrow s'}{\langle i := 6; \text{ while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s \rangle \rightarrow s'}$$
- 2 $\langle i := 6, s \rangle \rightarrow s[i \mapsto 6]$, fordi $s \vdash 6 \rightarrow_a 6$. Så $s_2 = s[i \mapsto 6]$.
- 3
$$\frac{\langle x := x + i; i := i - 2, s_2 \rangle \rightarrow s_3 \quad \langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_3 \rangle \rightarrow s'}{\langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_2 \rangle \rightarrow s'}$$
 fordi $s_2 \vdash i \neq 0 \rightarrow_b tt$
- 4
$$\frac{\langle x := x + i, s_2 \rangle \rightarrow s_4 \quad \langle i := i - 2, s_4 \rangle \rightarrow s_3}{\langle x := x + i; i := i - 2, s_2 \rangle \rightarrow s_3}$$

17/28

- 5 $\langle x := x + i, s_2 \rangle \rightarrow s_2[x \mapsto 11]$, fordi $s_2 \vdash x + i \rightarrow_a 11$ (anvend $[\text{plus}_{\text{bss}}]!$)
 $\Rightarrow s_4 = s_2[x \mapsto 11] = s[i \mapsto 6, x \mapsto 11]$
- 6 $\langle i := i - 2, s_4 \rangle \rightarrow s_4[i \mapsto 4]$, fordi $s_4 \vdash i - 2 \rightarrow_a 4$ (anvend $[\text{plus}_{\text{bss}}]!$)
 $\Rightarrow s_3 = s_4[i \mapsto 4] = s[i \mapsto 4, x \mapsto 11]$
- 7
$$\frac{\langle x := x + i; i := i - 2, s_3 \rangle \rightarrow s_5 \quad \langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_5 \rangle \rightarrow s'}{\langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_3 \rangle \rightarrow s'}$$
 fordi $s_3 \vdash i \neq 0 \rightarrow_b tt$
- 8 ...
- 9 ...
- 10 ... $s_5 = s[i \mapsto 2, x \mapsto 15]$
- 11
$$\frac{\langle x := x + i; i := i - 2, s_5 \rangle \rightarrow s_7 \quad \langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_7 \rangle \rightarrow s'}{\langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_5 \rangle \rightarrow s'}$$
 fordi $s_5 \vdash i \neq 0 \rightarrow_b tt$

18/28

12 ...

13 ...

14 ... $s_7 = s[i \mapsto 0, x \mapsto 17]$

15 $\langle \text{while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s_7 \rangle \rightarrow s_7$, fordi
 $s_7 \vdash i \neq 0 \rightarrow_b \text{ff} !$

$\Rightarrow s' = s_7 = s[i \mapsto 0, x \mapsto 17]$, dvs.

$$\langle i := 6; \text{ while } i \neq 0 \text{ do } (x := x + i; i := i - 2), s \rangle \rightarrow s[i \mapsto 0, x \mapsto 17]$$

• at konstruere derivationstræer = *kedeligt*, *mekanisk*

\Rightarrow automatisering \Rightarrow *fortolker!*

19/28

Definition: Givet $S \in \mathbf{Kom}$ og $s \in \mathbf{Tilstande}$:

- S siges at **terminere fra s** hvis der findes $s' \in \mathbf{Tilstande}$ så $\langle S, s \rangle \rightarrow s'$.
- S siges at **gå i uendelig løkke på s** hvis S ikke terminerer fra s .
- S **terminerer altid** hvis S terminerer fra alle $s \in \mathbf{Tilstande}$.
- S **går altid i uendelig løkke** hvis S går i uendelig løkke på alle $s \in \mathbf{Tilstande}$.

Opgave 4.8: Vis at $S = \text{while } 0 = 0 \text{ do skip}$ altid går i uendelig løkke.

20/28

- (Husk: **Tilstande** = **Var** \rightarrow \mathbb{Z})
- konfigurationer $\Gamma = \text{Kom} \times \text{Tilstande} \cup \text{Tilstande}$,
slutkonfigurationer $T = \text{Tilstande}$
- transitionsregler for \Rightarrow coming up
- transition $\langle S, s \rangle \Rightarrow s'$: terminering i s' *efter ét skridt*
- transition $\langle S, s \rangle \Rightarrow \langle S', s' \rangle$: *efter ét skridt* kommer vi fra S i tilstand s til S' i tilstand s'

21 / 28

[ass _{sss}]	$\langle x := a, s \rangle \Rightarrow s[x \mapsto v]$	hvor $s \vdash a \rightarrow_a v$
[skip _{sss}]	$\langle \text{skip}, s \rangle \Rightarrow s$	
[comp-1 _{sss}]	$\frac{\langle S_1, s \rangle \Rightarrow \langle S'_1, s' \rangle}{\langle S_1; S_2, s \rangle \Rightarrow \langle S'_1; S_2, s' \rangle}$	
[comp-2 _{sss}]	$\frac{\langle S_1, s \rangle \Rightarrow s'}{\langle S_1; S_2, s \rangle \Rightarrow \langle S_2, s' \rangle}$	
[if-sand _{sss}]	$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \Rightarrow \langle S_1, s \rangle$	hvis $s \vdash b \rightarrow_b tt$
[if-falsk _{sss}]	$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \Rightarrow \langle S_2, s \rangle$	hvis $s \vdash b \rightarrow_b ff$
[while _{sss}]	$\langle \text{while } b \text{ do } S, s \rangle \Rightarrow$ $\langle \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s \rangle$	

– reglen for while-løkken indeholder igen **rekursion**

22 / 28

Ikke-terminering svarer nu til *uendelige transitionsfølger*:

$$\langle \text{while } 0=0 \text{ do skip}, s \rangle \xRightarrow{3} \langle \text{while } 0=0 \text{ do skip}, s \rangle \xRightarrow{3} \dots$$

(eller til *løkker i transitionssystemet!*)

Definition: Givet $S \in \mathbf{Kom}$ og $s \in \mathbf{Tilstande}$:

- S siges at **terminere fra s** hvis der findes $s' \in \mathbf{Tilstande}$ så $\langle S, s \rangle \xRightarrow{*} s'$.
- S siges at **gå i uendelig løkke på s** hvis der findes en uendelig transitionsfølge

$$\langle S, s \rangle \Rightarrow \langle S_1, s_1 \rangle \Rightarrow \langle S_2, s_2 \rangle \Rightarrow \dots$$

23 / 28

Sætning 4.11 / 4.13 : Lad $S \in \mathbf{Kom}$ og $s, s' \in \mathbf{Tilstande}$. Da har vi $\langle S, s \rangle \rightarrow s'$ hvis og kun hvis $\langle S, s \rangle \xRightarrow{*} s'$.

– dvs. kommandoen S terminerer fra tilstand s i tilstand s' i big-step-semantikken hvis og kun hvis den gør det i small-step-semantikken.

– dvs. big-step- og small-step-semantikken er **ækvivalent**.

Vi viser her sætning 4.11 med tilhørende lemma 4.12. Beviserne for sætning 4.13 og lemma 4.14 springes over.

24 / 28

Lemma 4.12: Lad $S_1, S_2 \in \mathbf{Kom}$ og $s, s' \in \mathbf{Tilstande}$. Hvis $\langle S_1, s \rangle \xRightarrow{*} s'$ så $\langle S_1; S_2, s \rangle \xRightarrow{*} \langle S_2, s' \rangle$.

Bevis ved *induktion* i transitionsfølgers længde:

(Bemærk forskellen fra bogens bevis!)

- ① Lad $\langle S_1, s \rangle \xRightarrow{*} s'$, dvs. $\langle S_1, s \rangle \xRightarrow{k} s'$ for et eller andet $k \in \mathbb{N}_0$.
- ② Vi må have $k \neq 0$, da $\langle S_1, s \rangle \neq s'$. ($\xRightarrow{0}$ er defineret som $=$!)
- ③ *Induktionsbasis:* Lad $k = 1$. Reglen [comp-2_{sss}] giver at $\langle S_1, s \rangle \Rightarrow s'$ medfører $\langle S_1; S_2, s \rangle \Rightarrow \langle S_2, s' \rangle$. ✓
- ④ *Induktionsskridt:* Lad $k \geq 1$ og antag at vi har vist påstanden for alle transitionsfølger af længde k .
- ⑤ Lad $\langle S_1, s \rangle \xRightarrow{k+1} s'$. Vi må have $S'_1 \in \mathbf{Kom}$ og $s'' \in \mathbf{Tilstande}$ med $\langle S_1, s \rangle \Rightarrow \langle S'_1, s'' \rangle \xRightarrow{k} s'$.
- ⑥ Pga. induktionsantagelsen kan vi konkludere $\langle S'_1; S_2, s'' \rangle \xRightarrow{k} \langle S_2, s' \rangle$. Og med [comp-1_{sss}] har vi $\langle S_1; S_2, s \rangle \Rightarrow \langle S'_1; S_2, s'' \rangle$. Dvs.

$$\langle S_1; S_2, s \rangle \Rightarrow \langle S'_1; S_2, s'' \rangle \xRightarrow{k} \langle S_2, s' \rangle \quad \checkmark$$

25 / 28

Sætning 4.11: Lad $S \in \mathbf{Kom}$ og $s, s' \in \mathbf{Tilstande}$. Hvis $\langle S, s \rangle \rightarrow s'$ så $\langle S, s \rangle \xRightarrow{*} s'$.

Bevis ved *transitionsinduktion*:

Vis at egenskaben gælder for alle aksiomer, og at den bevares ved opbygning af derivationstræer.

[ass_{bss}]: Hvis $\langle S, s \rangle \rightarrow s'$ kommer fra [ass_{bss}], må vi have $S = x := a$, $s \vdash a \rightarrow_a v$ og $s' = s[x \mapsto v]$ for nogle x , a og v . [ass_{sss}] medfører $\langle S, s \rangle \Rightarrow s'$ ✓

[skip_{bss}]: Hvis $\langle S, s \rangle \rightarrow s'$ kommer fra [skip_{bss}], må vi have $S = \text{skip}$ og $s' = s$. [skip_{sss}] medfører $\langle S, s \rangle \Rightarrow s'$ ✓

[comp_{bss}]: Hvis $\langle S_1; S_2, s \rangle \rightarrow s''$ kommer fra reglen

$$\frac{\langle S_1, s \rangle \rightarrow s' \quad \langle S_2, s' \rangle \rightarrow s''}{\langle S_1; S_2, s \rangle \rightarrow s''}$$

og vores påstand gælder for præmisserne, må vi have $\langle S_1, s \rangle \xRightarrow{*} s'$ og $\langle S_2, s' \rangle \xRightarrow{*} s''$.

Med lemma 4.12 bliver den første til

$\langle S_1; S_2, s \rangle \xRightarrow{*} \langle S_2, s' \rangle$, sammensæt \Rightarrow ✓

26 / 28

[if-falsk_{bss}]: Hvis $\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'$ kommer fra reglen

$$\frac{\langle S_2, s \rangle \rightarrow s'}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'} \quad s \vdash b \rightarrow_b \text{ff}$$

giver **[if-falsk_{sss}]** transitionen

$$\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \Rightarrow \langle S_2, s \rangle$$

Med induktionsantagelsen har vi $\langle S_2, s \rangle \xRightarrow{*} s'$,
sammensæt $\Rightarrow \checkmark$

[if-sand_{bss}]: tilsvarende

27/28

[while-sand_{bss}]: Hvis $\langle \text{while } b \text{ do } S, s \rangle \rightarrow s'$ kommer fra reglen

$$\frac{\langle S, s \rangle \rightarrow s'' \quad \langle \text{while } b \text{ do } S, s'' \rangle \rightarrow s'}{\langle \text{while } b \text{ do } S, s \rangle \rightarrow s'} \quad s \vdash b \rightarrow_b \text{tt}$$

kan vi per antagelse konkludere at

$$\langle S, s \rangle \xRightarrow{*} s'' \quad \text{og} \quad \langle \text{while } b \text{ do } S, s'' \rangle \xRightarrow{*} s'$$

dvs. med lemma 4.12: $\langle S; \text{while } b \text{ do } S, s \rangle \xRightarrow{*} s'$
Og med **[if-sand_{sss}]** og **[while_{sss}]** har vi så

$$\begin{aligned} &\langle \text{while } b \text{ do } S, s \rangle \\ &\Rightarrow \langle \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else skip}, s \rangle \\ &\Rightarrow \langle S; \text{while } b \text{ do } S, s \rangle \\ &\xRightarrow{*} s' \end{aligned}$$

[while-falsk_{bss}]: tilsvarende

Færdig!

28/28