

Syntaks og semantik

Lektion 12

15 april 2008

Blokke

- 1 Abstrakt syntaks for **Bip**
- 2 Environment-store-modellen
- 3 Aritmetiske og boolske udtryk
- 4 Variabel-erklæringer
- 5 Kommandoer minus procedurekald
- 6 Scoperegler
- 7 Statisk binding

Bip = Bims + blokke og parameterløse procedurer:

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

$\mid \text{while } b \text{ do } S$
 $\mid \text{begin } D_V D_P S \text{ end} \mid \text{call } p$

ErkV: $D_V ::= \text{var } x := a; D_V \mid \varepsilon$

ErkP: $D_P ::= \text{proc } p \text{ is } S; D_P \mid \varepsilon$

- lokale erklæringer af variable (**ErkV**) og procedurer (**ErkP**) i en **blok**
- variable *initialiseres* ved erklæring
- semantikken af *procedurekald* afhænger af **scope-regler**
- *bogen* beskæftiger sig både med **dynamisk** og **statisk scope**
- vi lægger mest vægt på *statisk scope* her

3 / 18

- brug for ny tilstandsmodel for at kunne erklære variable
 - før: **Tilstande** = $\text{Var} \rightarrow \mathbb{Z}$
 - nu: **Var** \rightarrow **Loc** $\rightarrow \mathbb{Z}$
 - **Loc:** lokationer; lager-adresser
- \Rightarrow en tilstand (env_V, sto) beskrives ved:

env_V variabel-environment

- hvilken adresse er en given variabel bundet til?
- $Env_V = \text{Var} \cup \{\text{next}\} \rightarrow \text{Loc}$
- next peger til næste *frie* lokation
- for os: **Loc** = \mathbb{Z}
- opdatering:

$$env_V[x \mapsto \ell](x') = \begin{cases} env_V(x') & \text{hvis } x' \neq x \\ \ell & \text{hvis } x' = x \end{cases}$$

sto store

- hvilken værdi indeholder en given adresse?
- **Store** = $\text{Loc} \rightarrow \mathbb{Z}$
- opdatering: $sto[\ell \mapsto v](\ell') = \begin{cases} sto(\ell') & \text{hvis } \ell' \neq \ell \\ v & \text{hvis } \ell' = \ell \end{cases}$

2 / 18

$[plus_{bss}]$	$\frac{env_V, sto \vdash a_1 \rightarrow_a V_1 \quad env_V, sto \vdash a_2 \rightarrow_a V_2}{env_V, sto \vdash a_1 + a_2 \rightarrow_a V} \quad \text{hvor } V = V_1 + V_2$
$[minus_{bss}]$	$\frac{env_V, sto \vdash a_1 \rightarrow_a V_1 \quad env_V, sto \vdash a_2 \rightarrow_a V_2}{env_V, sto \vdash a_1 - a_2 \rightarrow_a V} \quad \text{hvor } V = V_1 - V_2$
$[mult_{bss}]$	$\frac{env_V, sto \vdash a_1 \rightarrow_a V_1 \quad env_V, sto \vdash a_2 \rightarrow_a V_2}{env_V, sto \vdash a_1 * a_2 \rightarrow_a V} \quad \text{hvor } V = V_1 \cdot V_2$
$[parent_{bss}]$	$\frac{env_V, sto \vdash a_1 \rightarrow_a V_1}{env_V, sto \vdash (a_1) \rightarrow_a V_1}$
$[num_{bss}]$	$env_V, sto \vdash n \rightarrow_a V \quad \text{hvis } \mathcal{N}[[n]] = V$
$[val_{bss}]$	$env_V, sto \vdash x \rightarrow_a V \quad \text{hvis } sto(env_V(x)) = V$

5 / 18

ErkV: $D_V ::= \text{var } X := a; D_V \mid \varepsilon$

- erklæringer **modificerer** env_V (pga. nye variable) og sto (pga. nye værdier til nye variable)
- \Rightarrow transitionssystem:

- konfigurationer
- $\Gamma_{D_V} = \text{ErkV} \times \text{Env}_V \times \text{Store} \cup \text{Env}_V \times \text{Store}$
- slutkonfigurationer $T_{D_V} = \text{Env}_V \times \text{Store}$
- dvs. konfigurationer (D_V, env_V, sto) og (env_V, sto)

$[var\text{-}erk_{bss}]$	$\frac{\langle D_V, env_V[x \mapsto \ell][next \mapsto new(\ell)], sto[\ell \mapsto v] \rangle \rightarrow_{D_V} \langle env'_V, sto' \rangle}{\langle \text{var } X := a; D_V, env_V, sto \rangle \rightarrow_{D_V} \langle env'_V, sto' \rangle} \quad \text{hvor } env_V, sto \vdash a \rightarrow_a v \text{ og } \ell = env_V(next)$
$[from\text{-}var\text{-}erk_{bss}]$	$\langle \varepsilon, env_V, sto \rangle \rightarrow_{D_V} \langle env_V, sto \rangle$

- big-step:** variabelerklæringer sker i ét hug
- new : Loc \rightarrow Loc** giver næste lokation; $new(\ell) = \ell + 1$

6 / 18

- også procedure-environment $env_P \in \text{Env}_P$, til at holde styr på *procedurer*
 - med tilhørende big-step-semantic for procedure-erklæringer ($\Gamma_{DP}, \rightarrow_{DP}, T_{DP}$)
 - men det snakker vi om senere**
 - dvs. **procedure-environment** env_P , **variabel-environment** env_V og **store** sto
 - men kommandoer **kan ikke ændre** env_V og env_P !
- \Rightarrow transitioner på formen $env_V, env_P \vdash \langle S, sto \rangle \rightarrow sto'$
- dvs. konfigurationer **Kom \times Store \cup Store**
 - og slutkonfigurationer **Store**

7 / 18

$[ass_{bss}]$	$env_V, env_P \vdash \langle x := a, sto \rangle \rightarrow sto[\ell \mapsto v] \quad \text{hvor } env_V, sto \vdash a \rightarrow_a v \text{ og } env_V(x) = \ell$
$[skip_{bss}]$	$env_V, env_P \vdash \langle skip, sto \rangle \rightarrow sto$
$[comp_{bss}]$	$\frac{env_V, env_P \vdash \langle S_1, sto \rangle \rightarrow sto' \quad env_V, env_P \vdash \langle S_2, sto' \rangle \rightarrow sto'}{env_V, env_P \vdash \langle S_1; S_2, sto \rangle \rightarrow sto'}$
$[if\text{-}sand_{bss}]$	$\frac{env_V, env_P \vdash \langle S_1, sto \rangle \rightarrow sto'}{env_V, env_P \vdash \langle \text{if } b \text{ then } S_1 \text{ else } S_2, sto \rangle \rightarrow sto'} \quad \text{hvis } env_V, sto \vdash b \rightarrow_b tt$
$[if\text{-}falst_{bss}]$	$\frac{env_V, env_P \vdash \langle S_2, sto \rangle \rightarrow sto'}{env_V, env_P \vdash \langle \text{if } b \text{ then } S_1 \text{ else } S_2, sto \rangle \rightarrow sto'} \quad \text{hvis } env_V, sto \vdash b \rightarrow_b ff$

8 / 18

$$\begin{array}{c}
 \text{[while-sand}_{bss}\text{]} \\
 \frac{env_V, env_P \vdash \langle S, sto \rangle \rightarrow sto' \quad env_V, env_P \vdash \langle \text{while } b \text{ do } S, sto' \rangle \rightarrow sto'}{env_V, env_P \vdash \langle \text{while } b \text{ do } S, sto \rangle \rightarrow sto'} \\
 \text{hvis } env_V, sto \vdash b \rightarrow_b tt \\
 \\
 \text{[while-falsk}_{bss}\text{]} \\
 env_V, env_P \vdash \langle \text{while } b \text{ do } S, sto \rangle \rightarrow sto \\
 \text{hvis } env_V, sto \vdash b \rightarrow_b ff \\
 \\
 \text{[blok}_{bss}\text{]} \\
 \frac{\langle D_V, env_V, sto \rangle \rightarrow_{DV} \langle env'_V, sto' \rangle \quad env'_V \vdash \langle D_P, env_P \rangle \rightarrow_{DP} env'_P \quad env'_V, env'_P \vdash \langle S, sto' \rangle \rightarrow sto'}{env_V, env_P \vdash \langle \text{begin } D_V D_P S \text{ end}, sto \rangle \rightarrow sto'}
 \end{array}$$

9 / 18

- **dynamisk binding af variable og procedurer:** $y = 10$
- **statisk binding af variable og procedurer:** $y = 9$ (hint: det er et andet x !)
- også muligt: *statisk binding af variable og dynamisk binding af procedurer*, og omvendt

```

begin var x := 0;
var y := 42
proc p is x := x+3;
proc q is call p;
begin var x := 9;
    proc p is x := x+1;
    call q;
    y := x
end
end
    
```

10 / 18

- **statisk binding af variable og procedurer:** ved *procedurekald* skal anvendes det variabel- og procedure-environment der fandtes ved *erklæringen*
- ⇒ procedurer skal **huske** env_V og env_P
- ⇒ $Env_P = Pnavne \rightarrow Kom \times Env_V \times Env_P$
- (**Pnavne** : procedurenavne)
- dvs. **Env_P** består af partielle afbildninger $p \mapsto \langle S, env_V, env_P \rangle$
 - S : procedure"kroppen"
 - env_V, env_P : variabel- og procedure-environment da p blev erklæret
- en **rekursiv definition**!
- big-step-semantik:
 - tilstande $ErkP \times Env_P \cup Env_P$
 - sluttilstande Env_P
 - transitioner \rightarrow_{DP}

11 / 18

$$\begin{array}{c}
 \text{[proc}_{bss}\text{]} \\
 \frac{env_V \vdash \langle D_P, env_P[p \mapsto (S, env_V, env_P)] \rangle \rightarrow_{DP} env'_P}{env_V \vdash \langle \text{proc } p \text{ is } S; D_P, env_P \rangle \rightarrow_{DP} env'_P} \\
 \\
 \text{[proc-tom}_{bss}\text{]} \\
 env_V \vdash \langle \epsilon, env_P \rangle \rightarrow_{DP} env_P \\
 \\
 \text{[call}_{bss}\text{]} \\
 \frac{env'_V[next \mapsto \ell], env'_P \vdash \langle S, sto \rangle \rightarrow sto' \quad env_V, env_P \vdash \langle \text{call } p, sto \rangle \rightarrow sto'}{env_V, env_P \vdash \langle p, sto \rangle \rightarrow sto'} \\
 \text{hvor } env_P(p) = (S, env'_V, env'_P) \\
 \text{og } \ell = env_V(next)
 \end{array}$$

12 / 18

Procedurer med parametre

- 8 Referenceparametre
- 9 Rekursion
- 10 Værdiparametre

13 / 18

At udvide **Bip** med procedurer med én **referenceparameter**:

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

$\mid \text{while } b \text{ do } S$

$\mid \text{begin } D_V D_P S \text{ end} \mid \text{call } p(y)$

EkV: $D_V ::= \text{var } x := a; D_V \mid \epsilon$

EkKP: $D_P ::= \text{proc } p(\text{var } x) \text{ is } S; D_P \mid \epsilon$

- **referenceparametre**: den **formelle** parameter x er en *reference* til adressen på den **aktuelle** parameter y
- (klares ved pointers i C)
- Eksempel:

```
begin
  var y:=3;
  proc p(var x) is x:= x+1;
  call p(y)
end
```

$\Rightarrow y = 4$

14 / 18

- procedure-environment:

$\text{Env}_P = \text{Pnavne} \rightarrow \text{Kom} \times \text{Var} \times \text{Env}_V \times \text{Env}_P$

- skal huske navnet på den formelle parameter
- at **erklære** procedurer:

$$[\text{proc}_{\text{bss}}] \quad \frac{\text{env}_V \vdash \langle D_P, \text{env}_P[p \mapsto (S, x, \text{env}_V, \text{env}_P)] \rangle \rightarrow_{DP} \text{env}_P}{\text{env}_V \vdash \langle \text{proc } p(\text{var } x) \text{ is } S; D_P, \text{env}_P \rangle \rightarrow_{DP} \text{env}_P}$$

$[\text{proc-tom}_{\text{bss}}] \quad \text{env}_V \vdash \langle \epsilon, \text{env}_P \rangle \rightarrow_{DP} \text{env}_P$

- at **kalde** procedurer:

$$[\text{call-ref}_{\text{bss}}] \quad \frac{\text{env}_V[x \mapsto \ell][\text{next} \mapsto \ell'], \text{env}_P \vdash \langle S, \text{sto} \rangle \rightarrow \text{sto}'}{\text{env}_V, \text{env}_P \vdash \langle \text{call } p(y), \text{sto} \rangle \rightarrow \text{sto}'}$$

hvor $\text{env}_P(p) = (S, x, \text{env}_V, \text{env}_P)$,
 $\ell = \text{env}_V(y)$ og $\ell' = \text{env}_V(\text{next})$

15 / 18

$$[\text{call-ref}_{\text{bss}}] \quad \frac{\text{env}_V[x \mapsto \ell][\text{next} \mapsto \ell'], \text{env}_P \vdash \langle S, \text{sto} \rangle \rightarrow \text{sto}'}{\text{env}_V, \text{env}_P \vdash \langle \text{call } p(y), \text{sto} \rangle \rightarrow \text{sto}'}$$

hvor $\text{env}_P(p) = (S, x, \text{env}_V, \text{env}_P)$,
 $\ell = \text{env}_V(y)$ og $\ell' = \text{env}_V(\text{next})$

Problem: den regel **tillader ikke rekursive procedurekald**

- fordi env_P er procedure-environmentet fra **før** p blev erklæret

Løsning: ny regel:

$$\text{env}_V[x \mapsto \ell][\text{next} \mapsto \ell'], \text{env}_P[p \mapsto (S, x, \text{env}_V, \text{env}_P)]$$

$$[\text{call-ref-rec}_{\text{bss}}] \quad \frac{\text{env}_V, \text{env}_P \vdash \langle \text{call } p(y), \text{sto} \rangle \rightarrow \text{sto}'}{\vdash \langle S, \text{sto} \rangle \rightarrow \text{sto}'}$$

hvor $\text{env}_P(p) = (S, x, \text{env}_V, \text{env}_P)$,
 $\ell = \text{env}_V(y)$ og $\ell' = \text{env}_V(\text{next})$

(kan også klares ved at modificere $[\text{proc}_{\text{bss}}]$ i stedet (**hvordan?**))

16 / 18

At udvide **Bip** med procedurer med én **værdiparametre**:

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

$\mid \text{while } b \text{ do } S$

$\mid \text{begin } D_V \ D_P \ S \ \text{end} \mid \text{call } p(a)$

ErkV: $D_V ::= \text{var } x := a; D_V \mid \varepsilon$

ErkP: $D_P ::= \text{proc } p(\text{var } x) \text{ is } S; D_P \mid \varepsilon$

- **værdiparametre**: den **formelle** parameter x bliver til en lokal variabel i proceduren, med **startværdi** = værdien af den **aktuelle** parameter

- Eksempel:

```
begin
```

$\text{var } y := 3;$

$\text{proc } p(\text{var } x) \text{ is } x := x+1;$

$\text{call } p(y)$

end

$\Rightarrow y = 3$

17 / 18

- **procedure-erklæringer** (uændret):

$$\frac{[proc_{bss}] \quad env_V \vdash \langle D_P, env_P[p \mapsto (S, x, env_V, env_P)] \rangle \rightarrow_{DP} env'_P}{env_V \vdash \langle \text{proc } p(\text{var } x) \text{ is } S; D_P, env_P \rangle \rightarrow_{DP} env'_P}$$

$[proc_tom_{bss}] \quad env_V \vdash \langle \varepsilon, env_P \rangle \rightarrow_{DP} env_P$

- **procedurekald**:

$env'_V[x \mapsto \ell][next \mapsto \text{new}(\ell)], env'_P$

$\vdash \langle S, sto[\ell \mapsto v] \rangle \rightarrow_{sto'} std'$

$[call_val_{bss}]$

$env_V, env_P \vdash \langle \text{call } p(a), sto \rangle \rightarrow_{sto'}$

$\text{hvor } env_P(p) = (S, x, env'_V, env'_P),$

$env_V, sto \vdash a \rightarrow_a v \text{ og } \ell = env_V(next)$

$env'_V[x \mapsto \ell][next \mapsto \text{new}(\ell)],$

$env'_P[p \mapsto (S, x, env'_V, env'_P)]$

$\vdash \langle S, sto[\ell \mapsto v] \rangle \rightarrow_{sto'}$

$[call_val_rec_{bss}]$

$env_V, env_P \vdash \langle \text{call } p(a), sto \rangle \rightarrow_{sto'}$

$\text{hvor } env_P(p) = (S, x, env'_V, env'_P),$

$env_V, sto \vdash a \rightarrow_a v \text{ og } \ell = env_V(next)$