

# Syntaks og semantik

Lektion 12

10 april 2007

# 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

- brug for ny tilstandsmodel for at kunne erklære variable
- før: **Tilstande** = **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**<sub>V</sub> = **Var**  $\cup \{\text{next}\} \rightarrow$  **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** = **Loc**  $\rightarrow \mathbb{Z}$
- opdatering:  $sto[\ell \mapsto v](\ell') = \begin{cases} sto(\ell') & \text{hvis } \ell' \neq \ell \\ v & \text{hvis } \ell' = \ell \end{cases}$

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

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

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

$$[\text{parent}_{\text{bss}}] \quad \frac{\text{env}_V, \text{sto} \vdash a_1 \rightarrow_a v_1}{\text{env}_V, \text{sto} \vdash (a_1) \rightarrow_a v_1}$$

$$[\text{num}_{\text{bss}}] \quad \text{env}_V, \text{sto} \vdash n \rightarrow_a v \quad \text{hvis } \mathcal{N}[\![n]\!] = v$$

$$[\text{var}_{\text{bss}}] \quad \text{env}_V, \text{sto} \vdash x \rightarrow_a v \quad \text{hvis } \text{sto}(\text{env}_V(x)) = v$$

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

- erklæringer **modificerer**  $\text{env}_V$  (pga. nye variable) og  $\text{sto}$  (pga. nye værdier til nye variable)

⇒ transitionssystem:

- konfigurationer

$$\Gamma_{DV} = \mathbf{ErkV} \times \mathbf{Env}_V \times \mathbf{Store} \cup \mathbf{Env}_V \times \mathbf{Store}$$

- slutkonfigurationer  $T_{DV} = \mathbf{Env}_V \times \mathbf{Store}$
- dvs. konfigurationer  $(D_V, \text{env}_V, \text{sto})$  og  $(\text{env}_V, \text{sto})$

[var-erkl<sub>bss</sub>]

$$\frac{\langle D_V, \text{env}_V[x \mapsto \ell][\text{next} \mapsto \text{new}(\ell)], \text{sto}[\ell \mapsto v] \rangle \rightarrow_{DV} \langle \text{env}'_V, \text{sto}' \rangle}{\langle \text{var } x := a; D_V, \text{env}_V, \text{sto} \rangle \rightarrow_{DV} \langle \text{env}'_V, \text{sto}' \rangle}$$

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

[tom-var-erkl<sub>bss</sub>]  $\langle \varepsilon, \text{env}_V, \text{sto} \rangle \rightarrow_{DV} \langle \text{env}_V, \text{sto} \rangle$

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

- også procedure-environment  $env_P \in \mathbf{Env}_P$ , til at holde styr på *procedurer*
  - med tilhørende big-step-semantik 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  $env_P$  !
- ⇒ transitioner på formen  $env_V, env_P \vdash \langle S, sto \rangle \rightarrow sto'$
- dvs. konfigurationer **Kom**  $\times$  **Store**  $\cup$  **Store**
  - og slutkonfigurationer **Store**

$$[\text{ass}_{\text{bss}}] \quad \text{env}_V, \text{env}_P \vdash \langle x := a, \text{sto} \rangle \rightarrow \text{sto}[\ell \mapsto v]$$

hvor  $\text{env}_V, \text{sto} \vdash a \rightarrow_a v$  og  $\text{env}_V(x) = \ell$

$$[\text{skip}_{\text{bss}}] \quad \text{env}_V, \text{env}_P \vdash \langle \text{skip}, \text{sto} \rangle \rightarrow \text{sto}$$

$$[\text{comp}_{\text{bss}}] \quad \frac{\begin{array}{l} \text{env}_V, \text{env}_P \vdash \langle S_1, \text{sto} \rangle \rightarrow \text{sto}' \\ \text{env}_V, \text{env}_P \vdash \langle S_2, \text{sto}' \rangle \rightarrow \text{sto}' \end{array}}{\text{env}_V, \text{env}_P \vdash \langle S_1; S_2, \text{sto} \rangle \rightarrow \text{sto}'}$$

$$[\text{if-sand}_{\text{bss}}] \quad \frac{\text{env}_V, \text{env}_P \vdash \langle S_1, \text{sto} \rangle \rightarrow \text{sto}'}{\text{env}_V, \text{env}_P \vdash \langle \text{if } b \text{ then } S_1 \text{ else } S_2, \text{sto} \rangle \rightarrow \text{sto}'}$$

hvis  $\text{env}_V, \text{sto} \vdash b \rightarrow_b \text{tt}$

$$[\text{if-falsk}_{\text{bss}}] \quad \frac{\text{env}_V, \text{env}_P \vdash \langle S_2, \text{sto} \rangle \rightarrow \text{sto}'}{\text{env}_V, \text{env}_P \vdash \langle \text{if } b \text{ then } S_1 \text{ else } S_2, \text{sto} \rangle \rightarrow \text{sto}'}$$

hvis  $\text{env}_V, \text{sto} \vdash b \rightarrow_b \text{ff}$



$$\begin{array}{c}
 \text{[while-sand}_{\text{bss}}\text{]} \quad \frac{\begin{array}{c} env_V, env_P \vdash \langle S, sto \rangle \rightarrow sto'' \\ env_V, env_P \vdash \langle \text{while } b \text{ do } S, sto'' \rangle \rightarrow sto' \end{array}}{env_V, env_P \vdash \langle \text{while } b \text{ do } S, sto \rangle \rightarrow sto'} \\
 \text{hvis } env_V, sto \vdash b \rightarrow_b tt
 \end{array}$$

$$\begin{array}{c}
 \text{[while-falsk}_{\text{bss}}\text{]} \quad env_V, env_P \vdash \langle \text{while } b \text{ do } S, sto \rangle \rightarrow sto \\
 \text{hvis } env_V, sto \vdash b \rightarrow_b ff
 \end{array}$$

$$\begin{array}{c}
 \text{[blok}_{\text{bss}}\text{]} \quad \frac{\begin{array}{c} \langle D_V, env_V, sto \rangle \rightarrow_{D_V} \langle env'_V, sto'' \rangle \\ env'_V \vdash \langle D_P, env_P \rangle \rightarrow_{D_P} env'_P \\ env'_V, env'_P \vdash \langle S, sto'' \rangle \rightarrow sto' \end{array}}{env_V, env_P \vdash \langle \text{begin } D_V \ D_P \ S \text{ end, } sto \rangle \rightarrow sto'}
 \end{array}$$

```
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
```

- dynamisk binding af variable og procedurer:  $y = 10$
- statisk binding af variable og procedurer:  $y = 9$
- også muligt: *statisk* binding af variable og *dynamisk* binding af procedurer, og omvendt

- **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  $env_P$

⇒  **$Env_P = P_{navne} \rightarrow Kom \times Env_V \times Env_P$**

- ( **$P_{navne}$**  : 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  $\rightarrow_{DP}$

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

$$[\text{proc-tom}_{\text{bss}}] \quad env_V \vdash \langle \varepsilon, env_P \rangle \rightarrow_{DP} env_P$$

$$[\text{call}_{\text{bss}}] \quad \frac{env'_V[\text{next} \mapsto \ell], env'_P \vdash \langle S, sto \rangle \rightarrow sto'}{env_V, env_P \vdash \langle \text{call } p, sto \rangle \rightarrow sto'}$$

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

# Procedurer med parametre

- 8 Referenceparametre
- 9 Rekursion
- 10 Værdiparametre

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

**Kom:**  $S ::= x := a \mid \text{skip} \mid S_1; S_2 \mid \text{if } b \text{ then } S_1 \text{ else } S_2$   
 $\mid \text{while } b \text{ do } S$   
 $\mid \text{begin } D_V \ D_P \ S \text{ end} \mid \text{call } p(y)$

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

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

- **referenceparametre**: den **formelle** parameter  $x$  er en *reference* til *adressen* på den **aktuelle** parameter  $y$

- **Eksempel**: 

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

$\Rightarrow y = 4$

- procedure-environment:

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

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

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

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

- 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  $\ell' = \text{env}_V(\text{next})$

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

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: dén regel **tillader ikke rekursive procedurekald**

- fordi  $\text{env}'_P$  er procedure-environmentet fra **før  $p$  blev erklæret**

Løsning: ny regel:

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

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



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

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

$\Rightarrow y = 3$

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

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

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

- procedure**kald**:

$$[\text{call-val}_{\text{bss}}] \quad \frac{\begin{array}{c} \text{env}'_V[x \mapsto \ell][\text{next} \mapsto \text{new}(\ell)], \text{env}'_P \\ \vdash \langle S, \text{sto}[\ell \mapsto v] \rangle \rightarrow \text{sto}' \end{array}}{\text{env}_V, \text{env}_P \vdash \langle \text{call } p(a), \text{sto} \rangle \rightarrow \text{sto}'}$$

hvor  $\text{env}_P(p) = (S, x, \text{env}'_V, \text{env}'_P)$ ,  
 $\text{env}_V, \text{sto} \vdash a \rightarrow_a v$  og  $\ell = \text{env}_V(\text{next})$

$$[\text{call-val-rec}_{\text{bss}}] \quad \frac{\begin{array}{c} \text{env}'_V[x \mapsto \ell][\text{next} \mapsto \text{new}(\ell)], \\ \text{env}'_P[p \mapsto (S, x, \text{env}'_V, \text{env}'_P)] \\ \vdash \langle S, \text{sto}[\ell \mapsto v] \rangle \rightarrow \text{sto}' \end{array}}{\text{env}_V, \text{env}_P \vdash \langle \text{call } p(a), \text{sto} \rangle \rightarrow \text{sto}'}$$

hvor  $\text{env}_P(p) = (S, x, \text{env}'_V, \text{env}'_P)$ ,  
 $\text{env}_V, \text{sto} \vdash a \rightarrow_a v$  og  $\ell = \text{env}_V(\text{next})$