
INF05516 - Semântica formal N
Ciência da Computação - UFRGS
2006-2

Marcus Ritt
mrpritt@inf.ufrgs.br

05/11/2006

Introdução	2
Agenda	3
Extensões simples	4
Escopos	5
Escopos simples	6
Escopos avançados: Gramática	7
A abordagem operacional	8
A abordagem denotational	9
Declarações	10
Comandos	11
Comandos novos	12
Exemplo: IMP	13
Exemplo: Semântica	14
Desvantagens	15
Escopos estáticos	16
Escopos estáticos	17
Separação	18
Nova abordagem: Variáveis.	19
Nova abordagem: Procedimentos	20
Declarações	21
Comandos	22
Comandos novos	23
Exemplo: IMP	24
Exemplo: Semântica	25

Agenda

Última aula:

- Ordenações parciais e domínios.

Hoje:

- Semântica denotacional de extensões de IMP.

v2021

Semântica formal N, aula 23 – 3 / 25

Extensões simples

- Considere `repeat c until b`.
- Usando

$$\text{repeat } c \text{ until } b \equiv c; \text{if } b \text{ then skip else repeat } c \text{ until } b.$$

chegamos à equação

$$C[\text{repeat } c \text{ until } b] = \text{fix } F$$

com

$$F(f) = \text{cond}(B[b], \text{id}, f) \circ C[c].$$

- O que falta, para ter a certeza que essa equação tem uma solução única?

v2021

Semântica formal N, aula 23 – 4 / 25

Escopos simples

- Como definir escopos na semântica denotational?
- Vamos começar com *escopos simples* do “tipo ML” (veja também exercícios 6):

$$c ::= \dots | \text{let } l = a \text{ in } c | \dots$$

- O que seria uma equação semântica adequada?

$$C[\text{let } l = a \text{ in } c] = \lambda \sigma \in \Sigma_{\perp}. (C[c]\sigma[l \mapsto A[a]\sigma])[l \mapsto \sigma(l)]$$

- Os escopos são estáticos ou dinâmicos?

v2021

Semântica formal N, aula 23 – 6 / 25

Escopos avançados: Gramática

- Declarações: Uma categoria sintática Decl com meta-variável $d \in \text{Decl}$.

$$d ::= \text{var } l := a; d | \epsilon$$

- Procedimentos: Categorias sintáticas Pname (nomes de procedimentos) e Pdef (definições de procedimentos) com meta-variáveis $p \in \text{Pname}$, $P \in \text{Pdef}$, respetivamente.

$$P ::= \text{proc } p \text{ is } c; P | \epsilon$$

- Comandos: Permitem escopos e chamada de funções

$$c ::= \dots | \text{begin } d \ P \ c \ \text{end} | \text{call } p | \dots$$

v2021

Semântica formal N, aula 23 – 7 / 25

A abordagem operacional

- Na semântica operacional a idéia para escopos dinâmicos foi
 - ◆ Declarações: Entrando em um escopo modifique o estado atual e saindo restaure o estado inicial das variáveis declarados.
 - ◆ Procedimentos: A mesma técnica, usando um ambiente de *definições* de procedimentos.

v2021

Semântica formal N, aula 23 – 8 / 25

A abordagem denotational

- O que seria uma denotação adequada de declarações? Uma tentativa é

$$D[\cdot] : \text{Decl} \rightarrow (\Sigma_{\perp} \rightarrow \Sigma_{\perp})$$

(uma declaração modifique o estado).

- O que seria uma denotação adequada de procedimentos?
- Novamente precisamos um ambiente de procedimentos, desta vez das *denotações*

$$\text{Amb}_P = [\text{Pname} \rightarrow (\Sigma_{\perp} \rightarrow \Sigma_{\perp})].$$

e o significado de uma declaração de procedimentos seria uma modificação desse ambiente

$$P[\cdot] : \text{Pdef} \rightarrow (\text{Amb}_P \rightarrow \text{Amb}_P).$$

- Finalmente, os comandos usam esse ambiente

$$C[\cdot] : \text{Com} \rightarrow \text{Amb}_P \rightarrow \Sigma_{\perp} \rightarrow \Sigma_{\perp}$$

v2021

Semântica formal N, aula 23 – 9 / 25

Equações: Declarações

$$\begin{aligned}D[\epsilon] &= \text{id} \\D[\text{var } l := a] &= \lambda\sigma \in \Sigma_{\perp}. \sigma[l \mapsto A[a]\sigma] \\D[d_1; d_2] &= D[d_2] \circ D[d_1] \\P[\epsilon] &= \text{id} \\P[\text{proc } p \text{ is } c] &= \lambda a_P \in \text{Amb}_P. a_P[p \mapsto C[c] a_P] \\P[p_1; p_2] &= P[p_2] \circ P[p_1]\end{aligned}$$

v2021

Semântica formal N, aula 23 – 10 / 25

Equações: Comandos

Equações conhecidas (extendidas para ambientes de procedimentos):

$$\begin{aligned}C[\text{skip}] &= \lambda a \in \text{Amb}_P. \text{id} \\C[l := a] &= \lambda a \in \text{Amb}_P. \lambda\sigma \in \Sigma_{\perp}. \sigma[l \mapsto A[a]\sigma] \\C[c_1; c_2] &= \lambda a \in \text{Amb}_P. (C[c_2] a) \circ (C[c_1] a) \\C[\text{if } b \text{ then } c \text{ else } c'] &= \lambda a \in \text{Amb}_P. \text{cond}(B[b], C[c] a, C[c'] a) \\C[\text{while } b \text{ do } c] &= \lambda a \in \text{Amb}_P. \text{fix } F\end{aligned}$$

com

$$F(f) = \text{cond}(B[b], f \circ (C[c] a), \text{id})$$

v2021

Semântica formal N, aula 23 – 11 / 25

Equações: Comandos novos

Equações novas:

$$\begin{aligned} C[\text{begin } d \text{ P } c \text{ end}] &= \\ &\lambda a_P \in \text{Amb}_P. \lambda \sigma \in \Sigma_{\perp}. (C[c](P[\text{P}] a_P)(D[d] \sigma))[DV(d) \mapsto \sigma] \\ C[\text{call } p] &= \lambda a_P \in \text{Amb}_P. a_P p \end{aligned}$$

v2021

Semântica formal N, aula 23 – 12 / 25

Exemplo: IMP

Considere

```
begin proc p is x:=x 1
  begin
    var x:=5;
    call p
  end
end
```

num estado σ tal que $\sigma(x) = 0$ e um ambiente de procedimentos a_P vazio (totalmente indefinido).

v2021

Semântica formal N, aula 23 – 13 / 25

Exemplo: Semântica

$$\begin{aligned} & C[\text{begin proc...end}]_{a_P} \sigma \\ &= (C[\text{begin var } x:=5; \text{call p end}] (P[\text{proc p is } x:=x-1]_{a_P}) (D[\epsilon] \sigma)) [\emptyset \mapsto \sigma] \\ &= (C[\text{begin var } x:=5; \text{call p end}] (P[\text{proc p is } x:=x-1]_{a_P}) \sigma) \\ &= (C[\text{begin var } x:=5; \text{call p end}]_{a_P} [p \mapsto C[x:=x-1]_{a_P}] \sigma) \\ &= (C[\text{begin var } x:=5; \text{call p end}]_{a_P} [p \mapsto \lambda \sigma. \sigma[x \mapsto \sigma(x) - 1]] \sigma) \\ &= (C[\text{begin var } x:=5; \text{call p end}]_{a_P} [p \mapsto \lambda \sigma. \sigma[x \mapsto \sigma(x) - 1]] \sigma) \\ &= (C[\text{call p}] (P[\epsilon]_{a_P} [p \mapsto \lambda \sigma. \sigma[x \mapsto \sigma(x) - 1]]) (D[\text{var } x:=5] \sigma)) [x \mapsto \sigma(x)] \\ &= (C[\text{call p}]_{a_P} [p \mapsto \lambda \sigma. \sigma[x \mapsto \sigma(x) - 1]] (D[\text{var } x:=5] \sigma)) [x \mapsto \sigma(x)] \\ &= ((\lambda \sigma. \sigma[x \mapsto \sigma(x) - 1]) (D[\text{var } x:=5] \sigma)) [x \mapsto \sigma(x)] \\ &= ((\lambda \sigma. \sigma[x \mapsto \sigma(x) - 1]) \sigma[x \mapsto 5]) [x \mapsto \sigma(x)] \\ &= (\sigma[x \mapsto 4]) [x \mapsto \sigma(x)] \\ &= \sigma \end{aligned}$$

v2021

Semântica formal N, aula 23 – 14 / 25

Desvantagens

Essa abordagem tem algumas desvantagens:

- A definição resulta em escopos dinâmicos. Como definir escopos estáticos?
- A definição não permite procedimentos recursivos: O escopo de um procedimento não tem acesso à definição do procedimento:

$$P[\text{proc p is c}] = \lambda a_P \in \text{Amb}_P. a_P [p \mapsto C[\text{c}]_{a_P}]$$

v2021

Semântica formal N, aula 23 – 15 / 25

Escopos estáticos

- Suponhe uma variável x global e local.
- Com um estado $\Sigma = [\text{Ident} \rightarrow \mathbb{Z}]$ o procedimento do exemplo se refere ao x global.
- Mas no tempo da aplicação do procedimento o estado refere ao x local.

$$\text{Ident} \longrightarrow \mathbb{Z}$$

- Para obter escopos estáticos podemos separar o *ambiente de variáveis* da *memória*, usando *endereços*.

$$\text{Ident} \longrightarrow \text{End} \longrightarrow \mathbb{Z}$$

- Assim, um procedimento pode usar os endereços das variáveis correspondente ao ponto da definição.

v2021

Semântica formal N, aula 23 – 17 / 25

Separação

Ao invés de um estado $\sigma \in \Sigma = [\text{Ident} \rightarrow \mathbb{Z}]$ vamos usar

- um *ambiente de identificadores* $a \in \text{Amb}_I = [\text{Ident} \rightarrow \text{End}]$ que mapa identificadores para endereços;
- um conjunto de endereços End (p.ex. \mathbb{Z});
- um memória $m \in \text{Mem} = [\text{End} \cup \{\text{prox}\} \rightarrow \mathbb{Z}]$ que mapa endereços para valores;
- a memória contém um endereço especial prox que contém o próximo endereço livre;
- uma função $\text{novo} : \text{End} \rightarrow \text{End}$ que, dado um endereço, retorna um endereço novo;
- uma função

$$\text{lookup} : \text{Amb}_V \rightarrow \text{Mem} \rightarrow \Sigma = \lambda a_V \in \text{Amb}_V. \lambda m \in \text{Mem}. m \circ a$$

que ajuda de achar o valor de uma variável diretamente.

v2021

Semântica formal N, aula 23 – 18 / 25

Nova abordagem: Variáveis

- Usando essa separação podemos modificar nossa abordagem como segue.
- A denotação de uma declaração de variável em um ambiente de locais e um memória resulta em
 - ◆ um novo ambiente (com uma nova endereço para a variável declarada)
 - ◆ uma nova memória (com o valor da declaração no endereço correspondente)

$$D[\cdot] : \text{Decl} \rightarrow (\text{Amb}_I \times \text{Mem}) \rightarrow (\text{Amb}_I \times \text{Mem})$$

v2021

Semântica formal N, aula 23 – 19 / 25

Nova abordagem: Procedimentos

- O ambiente de procedimentos agora contém funções que modificam a *memória*:

$$\text{Amb}_P = [\text{Pname} \rightarrow (\text{Mem}_\perp \rightarrow \text{Mem}_\perp)]$$

- A denotação de uma declaração usa o ambiente de locais também (para achar os endereços da variáveis)

$$P[\cdot] : \text{Pdef} \rightarrow \text{Amb}_I \rightarrow \text{Amb}_P \rightarrow \text{Amb}_P$$

- Finalmente, a denotação de um comando agora depende do ambiente de variáveis e procedimentos

$$C[\cdot] : \text{Com} \rightarrow \text{Amb}_I \rightarrow \text{Amb}_P \rightarrow \text{Mem}_\perp \rightarrow \text{Mem}_\perp$$

v2021

Semântica formal N, aula 23 – 20 / 25

Equações: Declarações

$$\begin{aligned}D[\epsilon] &= \text{id} \\D[\text{var } l := a] &= \lambda(a, m) \in \text{Amb}_I \times \text{Mem}. (a[l \mapsto e], m[e \mapsto v][\text{prox} \mapsto \text{novos } e]) \\ \text{com } e = m \text{ prox e } v = A[a](\text{lookup } a \ m) \\D[d_1 ; d_2] &= D[d_2] \circ D[d_1] \\P[\epsilon] &= \lambda a_I \in \text{Amb}_I. \text{id} \\P[\text{proc } p \text{ is } c] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. a_P[p \mapsto C[c] a_I a_P] \\P[p_1 ; p_2] &= \lambda a_I \in \text{Amb}_I. (P[p_2] a_I) \circ (P[p_1] a_I)\end{aligned}$$

v2021

Semântica formal N, aula 23 – 21 / 25

Equações: Comandos

Equações conhecidas (extendidas para ambientes de variáveis e procedimentos):

$$\begin{aligned}C[\text{skip}] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. \text{id} \\C[l := a] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. \lambda m \in \text{Mem}_\perp. m[e \mapsto A[a](\text{lookup } a_I \ m)] \\ \text{com } e = \text{Amb}_I \ l \\C[c_1 ; c_2] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. (C[c_2] a_I a_P) \circ (C[c_1] a_I a_P) \\C[\text{if } b \text{ then } c \text{ else } c'] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. \text{cond}(B[b] \circ (\text{lookup } a_I), C[c] a_I a_P, C[c'] a_I a_P) \\ \text{com } \text{cond} : (\text{Mem} \rightarrow \text{Bool}) \times (\text{Mem}_\perp \rightarrow \text{Mem}_\perp) \times (\text{Mem}_\perp \rightarrow \text{Mem}_\perp) \rightarrow (\text{Mem}_\perp \rightarrow \text{Mem}_\perp) \\C[\text{while } b \text{ do } c] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. \text{fix } F\end{aligned}$$

com

$$F(f) = \text{cond}(B[b] \circ (\text{lookup } a_I), f \circ (C[c] a_I a_P), \text{id})$$

v2021

Semântica formal N, aula 23 – 22 / 25

Equações: Comandos novos

Equações novas:

$$\begin{aligned} C[\text{begin } d \text{ P } c \text{ end}] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. \lambda m \in \text{Mem}_\perp. C[\text{c}] a'_I (P[\text{P}] a'_I a_P) m' \\ &\text{com } (a'_I, m') = D[\text{d}] (a_I, m) \\ C[\text{call } p] &= \lambda a_I \in \text{Amb}_I. \lambda a_P \in \text{Amb}_P. a_P p \end{aligned}$$

v2021

Semântica formal N, aula 23 – 23 / 25

Exemplo: IMP

Considere o exemplo

```
begin
  proc p is x:=x-1
  begin
    var x:=5;
    call p
  end
end
```

novamente, desta vez num ambiente de variáveis a_I , um memória m , tal que $a_I x = 0$, $m 0 = 0$, $m \text{ prox} = 1$ e um ambiente de procedimentos a_P vazio (totalmente indefinido).

v2021

Semântica formal N, aula 23 – 24 / 25

Exemplo: Semântica

$$\begin{aligned} & C[\text{begin proc...end}] a_I a_P m \\ & \text{com } (a_I, m) = D[\epsilon](a_I, m) \\ & = C[\text{begin var } x:=5; \text{call } p \text{ end}] a_I (P[\text{proc } p \text{ is } x:=x-1] a_I a_P m) \\ & = C[\text{begin var } x:=5; \text{call } p \text{ end}] a_I (a_P[p \mapsto C[x:=x-1] a_I a_P] m) \\ & = C[\text{begin var } x:=5; \text{call } p \text{ end}] a_I (a_P[p \mapsto \lambda m \in \text{Mem}_\perp. m[0 \mapsto (m\ 0) - 1]]) m \\ & = C[\text{call } p] a'_I (P[\epsilon] a'_I (a_P[p \mapsto \lambda m \in \text{Mem}_\perp. m[0 \mapsto (m\ 0) - 1]])) m' \\ & \text{com } (a'_I, m') = D[\text{var } x:=5](a_I, m) \\ & \text{e } a'_I = a_I[x \mapsto 1], m' = m[1 \mapsto 5][\text{prox} \mapsto 2] \\ & = C[\text{call } p] a'_I (a_P[p \mapsto \lambda m \in \text{Mem}_\perp. m[0 \mapsto (m\ 0) - 1]]) m' \\ & = ((a_P[p \mapsto \lambda m \in \text{Mem}_\perp. m[0 \mapsto (m\ 0) - 1]]) p) m' \\ & = (\lambda m \in \text{Mem}_\perp. m[0 \mapsto (m\ 0) - 1]) m' \\ & = m'[0 \mapsto (m'\ 0) - 1] \\ & = m[1 \mapsto 5][\text{prox} \mapsto 2][0 \mapsto -1] \end{aligned}$$