Semântica Natural

Semântica Natural (de Avaliação)

- A relação de transição $\longrightarrow$ especifica, para cada comando, a relação entre o estado inicial e o estado final.

$$\langle S,\, s\rangle \to s'$$

- $\longrightarrow$ é definida indutivamente por um conjunto de *regras*

premissas

$$\frac{\langle S_1,\, s_1\rangle \to s'_1,\, \cdots,\, \langle S_\mathrm{n},\, s_\mathrm{n}\rangle \to s'_\mathrm{n}}{\langle S,\, s\rangle \to s'} \text{ if } \cdots$$

condições

conclusão

As regras sem premissas chamam-se *axiomas*.

Regras de Avaliação

$[\mathrm{ass_{ns}}]$	$\langle x := a,\, s\rangle \to s[x \mapsto \mathcal{A}[\![a]\!]s]$
$[\mathrm{skip_{ns}}]$	$\langle \mathtt{skip},\, s\rangle \to s$
$[\mathrm{comp_{ns}}]$	$\dfrac{\langle S_1,\, s\rangle \to s',\, \langle S_2,\, s'\rangle \to s''}{\langle S_1 ; S_2,\, s\rangle \to s''}$

Regras de Avaliação

$[\mathrm{if_{ns}^{tt}}]$	$\dfrac{\langle S_1,\, s\rangle \to s'}{\langle \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2,\, s\rangle \to s'}$ if $\mathcal{B}[\![b]\!]s = \mathbf{tt}$
$[\mathrm{if_{ns}^{ff}}]$	$\dfrac{\langle S_2,\, s\rangle \to s'}{\langle \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2,\, s\rangle \to s'}$ if $\mathcal{B}[\![b]\!]s = \mathbf{ff}$

Regras de Avaliação

$$[\text{while}^{\text{tt}}_{\text{ns}}] \qquad \frac{\langle S,\, s\rangle \to s',\ \langle \texttt{while}\ b\ \texttt{do}\ S,\, s'\rangle \to s''}{\langle \texttt{while}\ b\ \texttt{do}\ S,\, s\rangle \to s''} \text{ if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$[\text{while}^{\text{ff}}_{\text{ns}}] \qquad \langle \texttt{while}\ b\ \texttt{do}\ S,\, s\rangle \to s \text{ if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

Árvores de Derivação

- As transições são ***derivadas*** construindo *árvores de derivação*.
- A ***raiz*** da derivação é a transição que queremos derivar e as ***folhas*** são instâncias dos axiomas.
- Os ***nodos internos*** são conclusões de instâncias de regras.
- As condições instanciadas das regras têm que ser satisfeitas.

Exercício

- Defina na linguagem **While** os seguintes programas:

  SWAP — Troca de valores entre as variáveis x e y.

  MIN — Cálculo em m do mínimo de x, y e z.

  EXP — Cálculo em r do valor de x elevado a y.

- Seja s o estado que mapeia todas as variáveis em 0, excepto x e y. Nestes casos: s x = 3 e s y = 2.

  Construa árvores de derivação para as transições correspondentes à execução de cada um dos programas nos estado s.

Terminação

- A execução de um programa S num estado s
  - *termina* se e só se existir um estado s' tal que $\langle S,\, s\rangle \to s'$
  - *diverge* (ou *entra em ciclo*) se e só se não existir nenhum estado s' tal que $\langle S,\, s\rangle \to s'$

Exercício: Como é que se comportam os seguintes programas?

- $\texttt{while}\ \neg(\texttt{x}=1)\ \texttt{do}\ (\texttt{y}:=\texttt{y}\star\texttt{x};\ \texttt{x}:=\texttt{x}-1)$
- $\texttt{while}\ 1\leq\texttt{x}\ \texttt{do}\ (\texttt{y}:=\texttt{y}\star\texttt{x};\ \texttt{x}:=\texttt{x}-1)$
- `while true do skip`

Equivalência Semântica

O sistema de transição permite-nos argumentar acerca dos programas e das suas propriedades.

Definição: Dois programas S_1 e S_2 dizem-se *semanticamente equivalentes* se, para todos os estados s e s',

$$\langle S_1,\, s\rangle \to s' \qquad \text{sse} \qquad \langle S_2,\, s\rangle \to s'$$

Exercício: Prove que os programas $\mathtt{while}\ b\ \mathtt{do}\ S$ e $\mathtt{if}\ b\ \mathtt{then}\ (S;\, \mathtt{while}\ b\ \mathtt{do}\ S)\ \mathtt{else\ skip}$. são semanticamente equivalentes.

Indução na estrutura da derivação

Provar uma propriedade para todas as árvores de derivação.

Casos de base (os axiomas)

- Provar que a propriedade se verifica para todos os axiomas.

Casos indutivos (as regras)

- Para cada regra assumir que a propriedade se verifica para as premissas da regra (são as *hipóteses de indução*) e provar que a propriedade também se verifica para a conclusão, desde que as condições da regra sejam satisfeitas.

Determinismo

A semântica natural aqui apresentada é *determinista*.

Teorema: Para quaisquer estados $s,\ s', s''$ e programa S,

se $\langle S,\, s\rangle \to s'$ e $\langle S,\, s\rangle \to s''$ então $s' = s''$.

Prova: Por indução na estrutura da derivação de $\langle S,\, s\rangle \to s'$

A função semântica $\mathcal{S}_{\text{ns}}$

O significado de um programa pode ser visto como uma *função parcial* de $\mathbf{State}$ para $\mathbf{State}$.

Definição: $\mathcal{S}_{\text{ns}}\colon \mathbf{Stm} \to (\mathbf{State} \hookrightarrow \mathbf{State})$

$$\mathcal{S}_{\text{ns}}[\![S]\!]s = \begin{cases} s' & \text{if } \langle S,\, s\rangle \to s' \\ \underline{\text{undef}} & \text{otherwise} \end{cases}$$

Para qualquer programa S, $\mathcal{S}_{\text{ns}}[\![S]\!] \in \mathbf{State} \hookrightarrow \mathbf{State}$ é uma função parcial.

Exercício: Qual o valor de $\mathcal{S}_{\text{ns}}[\![\mathtt{while\ true\ do\ skip}]\!]\ s$?

Semântica natural para expressões

- A semântica das expressões aritméticas foi dada pela função semântica $\mathcal{A}$.
- Podemos ter uma abordagem operacional e definir uma semântica natural para as expressões aritméticas.
- Teremos neste caso dois tipos de configurações:

$\langle a,\, s\rangle$ denota que a é avaliada no estado s

z denota o valor final (um elemento de $\mathbf{Z}$)

Semântica natural para expressões

A relação de transição tem a forma $\langle a,\, s\rangle \rightarrow_{\text{Aexp}} z$

Exemplo de algumas regras:

$$\langle n,\, s\rangle \rightarrow_{\text{Aexp}} \mathcal{N}[\![n]\!]$$

$$\langle x,\, s\rangle \rightarrow_{\text{Aexp}} s\ x$$

$$\frac{\langle a_1,\, s\rangle \rightarrow_{\text{Aexp}} z_1,\ \langle a_2,\, s\rangle \rightarrow_{\text{Aexp}} z_2}{\langle a_1 + a_2,\, s\rangle \rightarrow_{\text{Aexp}} z} \quad \text{where } z = z_1 + z_2$$

Exercício: Complete a especificação do sistema de transição e prove que o significado dado por esta definição é o mesmo do que por $\mathcal{A}$.

Exercício: Defina uma semântica natural para expressões booleanas.

Semântica natural para expressões

Exercício: Imagine que pretendemos acrescentar operadores com efeitos laterais (como o ++x e o x++ do C) à linguagem de expressões aritméticas.

- Proponha uma semântica natural que permita poderia capturar o efeito da avaliação destas novas expressões.
- Que consequências isto acarreta na semântica dos programas?
- Proponha um conjunto de regras de avaliação que captem corretamente o comportamento destes novos programas.