# Ficha 1

Semântica das Linguagens de Programação

## Semântica Natural

1. Defina na linguagem **While** os seguintes programas

   - SWAP - que troca de valores entre as variáveis `x` e `y`.
   - MIN - que calcula em `m` do mínimo de `x`, `y` e `z`.
   - EXP - que calcula em `r` do valor de `x` elevado a `y`.
   - FACT - que calcula em `f` do valor do factorial de `n`.

   Construa árvores de derivação para as transições correspondentes à execução de cada um dos programas nos estado $s$, sendo $s$ o estado que mapeia todas as variáveis em 0 excepto os sequintes casos: $s\,x = 3$, $s\,y = 2$ e $s\,n = 3$.

2. Prove que de acordo com a semântica natual da linguagem **While** apresentada nas aulas teóricas os seguintes comandos são semanticamente equivalentes.

   (a) $\{S_1 \texttt{ ; } S_1\} \texttt{ ; } S_3$ e $S_1 \texttt{ ; } \{S_1 \texttt{ ; } S_3\}$

   (b) `while` $b$ `do` $C$ e `if` $b$ `then` $\{C$ `;` `while` $b$ `do` $C\}$ `else skip`

3. Considere o seguinte programa da linguagem **While**:

```
if (x > 0) then y := x
else if (x < 0) then y := -x
     else z := 1
```

   (a) Recorrendo à semântica de avaliação (*big-step*) simule a execução do programa a partir do estado inicial $s$ em que $s\,x = -1$.

   (b) Diga, se o programa acima é *equivalente* a algum dos dois seguintes. Apresente em cada caso uma prova (se forem iguais) ou um contra-exemplo (se não forem).

```
y := x;
if (y < 0) then y := -y
else z := 1;
```

```
if (x == 0) then z := 1
else if (x < 0) then y := -x
     else y := x;
```

4. Pretende-se estender a linguagem **While** acrescentando-lhe uma nova forma de ciclo de acordo com a seguinte sintaxe abstracta:

$$\textbf{Stm} \ni C ::= \ \ldots \ \mid \texttt{repeat}\ C_1\ \texttt{until}\ b$$

   A descrição informal da semântica deste comando é a seguinte:

   > *O comando $C_1$ é executado repetidamente enquanto o valor da expressão $b$ for falso, sendo o teste feito depois da execução do comando.*

(a) Especifique formalmente o comportamento deste novo ciclo, escrevendo regras apropriadas para a *semântica natural* da linguagem. As regras não devem fazer referência a outros ciclos.

(b) Tendo em conta as regras que propôs, prove a *equivalência* entre dois comandos seguintes:

$$\texttt{repeat}\ C\ \texttt{until}\ b \qquad \text{e} \qquad C\ ;\ \texttt{while}\ \neg b\ \texttt{do}\ C$$

5. Pretende-se definir uma semântica natural para a avaliação de expressões.

(a) Especifique o sistema de transição para a avaliação de expressões aritméticas.

(b) Prove que o significado de uma expressão aritmérica dado por esta nova definição é o mesmo do que o dado por $\mathcal{A}$.

(c) Especifique o sistema de transição para a avaliação de expressões booleanas.

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

$$\mathbf{Aexp} \ni a ::= n \mid x \mid {+}{+}x \mid x{+}{+} \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2$$

(a) Proponha uma semântica natural que permita capturar o efeito da avaliação destas novas expressões ariteméticas.

(b) Defina uma semântica natural expressões booleanas.

(c) Com base na semântica que definiu, indique o valor das expressões $({+}{+}x) * x$ e $(x{+}{+}) * x = 0$ no estado $s$ em que $s\, x = 0$.

(d) Que consequências isto acarreta na semântica dos programas?

(e) Proponha um conjunto de regras de avaliação que captem corretamente o comportamento destes novos programas.

7. Pretende-se estender a linguagem **While**, acrescentando-lhe uma nova forma de ciclo, inspirado no comando com a mesma sintaxe na linguagem C, e com um comportamento igual.

$$\mathbf{Stm} \ni C ::= \ \ldots \ \mid \ \texttt{for}\ (C_1, b, C_3)\ \texttt{do}\ C_2$$

(a) Especifique formalmente o comportamento deste novo ciclo, escrevendo regras apropriadas para a *semântica natural* da linguagem. As regras não devem fazer referência a outros ciclos.

(b) Recorrendo à semântica de natural e tendo em conta as regras que propôs, prove que o comando $\texttt{while}\ b\ \texttt{do}\ \{C_2\,;\,C_3\}$ e o comando $\texttt{for}\,(\texttt{skip}, b, C_3)\ \texttt{do}\ C_2$ são *equivalentes*. Isto é, mostre que para qualquer $s, s_f \in \mathbf{State}$,

$$\langle\, \texttt{while}\ b\ \texttt{do}\ \{C_2\,;\,C_3\}, s\,\rangle \to s_f \qquad sse \qquad \langle\, \texttt{for}\,(\texttt{skip}, b, C_3)\ \texttt{do}\ C_2, s\,\rangle \to s_f$$

A prova das duas implicações são por indução na derivação do antecedente.

(c) Prove agora a *equivalência* dos dois comandos seguintes:

$$\texttt{for}\,(C_1, b, C_3)\ \texttt{do}\ C_2 \qquad \text{e} \qquad C_1\ ;\ \texttt{while}\ b\ \texttt{do}\,\{C_2\,;\,C_3\}$$

Terá que demonstrar que para qualquer $s, s_f \in \mathbf{State}$,

i. $\langle\, \texttt{for}\,(C_1, b, C_3)\ \texttt{do}\ C_2, s\,\rangle \to s_f \ \Rightarrow\ \langle\, C_1\,;\,\texttt{while}\ b\ \texttt{do}\ \{C_2\,;\,C_3\}, s\,\rangle \to s_f$

ii. $\langle\, C_1\,;\, \texttt{while}\ b\ \texttt{do}\ \{C_2\,;\, C_3\}\,, s\,\rangle \to s_f \;\Rightarrow\; \langle\, \texttt{for}\,(C_1, b\,, C_3)\ \texttt{do}\ C_2\,, s\,\rangle \to s_f$

A prova de (i) faz-se por indução na derivação de $\langle\, \texttt{for}\,(C_1, b\,, C_3)\ \texttt{do}\ C_2\,, s\,\rangle \to s_f$.
A prova de (ii) não é feita por indução. Usa o lema demonstrado na alínea anterior.

8. Pretende-se construir em Haskell um programa que simule a avaliação de programas.

   (a) Comece por simular a avaliação de expressões. Para isso:

      i. Defina tipos para as categorias sintácticas **Aexp** e **Bexp**.

      ii. Defina as operações de substituição para expressões **Aexp** e **Bexp**.

      iii. Defina as funções semânticas $\mathcal{A}$ e $\mathcal{B}$.

   (b) Implemente agora um programa que simule a avaliação de programas, num dado estado, de acordo com a semântica natural (*big-step*).

      i. Defina o tipo para a categoria sintáctica dos comandos, **Stm**.

      ii. Defina a função auxiliar de "update" do estado.

      iii. Defina a função `evalNS` que simula a relação de avaliação dos comandos.

      iv. Defina alguns exemplos de programas e estados e teste a relação de avaliação que definiu.