**Semânticas de Linguagens de Programação - Exercícios 4**
**Semântica operacional *small-step* para a linguagem While. Conceitos de Linguagens Imperativas.**

## Semântica operacional estrutural (*small-step*) da linguagem While

1. Considera a semântica operacional estrutural dada no curso.
   (a) Sendo $s_0 = [x = 3, y = 5]$, determina o estado a após a a execução do comando

   $$\texttt{if } x > y \texttt{ then } z := x \texttt{ else } z := y$$

   (b) Sendo $s_0 = [x = 3]$, determinar o estado após a execução de:

   $$y := 1; \texttt{while } \neg(x = 1) \texttt{ do } (y := y \times x; x := x - 1)$$

   (c) Sendo $s_0 = [x = 10, y = 5]$, determinar o estado após a execução de:

   $$z := 0; \texttt{while } y \leq x \texttt{ do } (z := z + 1; x := x - y)$$

2. Define uma semântica operacional estrutural para o comando `repeat` $c$ `until` $b$ (sem usar a do `while`).

3. Define uma semântica operacional estrutural para o comando `for` $x := a_1$ `to` $a_2$ `do` $c$ (sem usar a do `while`).

4. Mostra que para todo o $k \geq 1$, se $\langle c_1, s\rangle \Rightarrow^k s'$ então $\langle c_1; c_2, s\rangle \Rightarrow^k \langle c_2, s'\rangle$ .

5. Mostra que a semântica operacional estrutural dada para a linguagem **While** é determinística.

6. Mostra a equivalência semântica dos seguintes comandos:
   (a) `skip`; $c$ e $c$
   (b) `while` $b$ `do` $c$ e `if` $b$ `then` ($c$ ; `while` $b$ `do` $c$) `else skip`

7. Implementa em Haskell a semântica operacional estrutural para a linguagem **While**, usando as estruturas de dados definidas no exercício dado na prática 2.

## Conceitos de linguagens imperativas: âmbito de variáveis

8. Considera o seguinte programa em Pascal.

   **program** main;
   **var** $a$ : *integer*;
   **procedure** $P1$
   **begin**
   **writeln**($'a =', a$);
   **end**;
   **procedure** $P2$
   **var** $a$ : *integer*;

**begin**
$a := 10$;
$P1$
**end**;
**begin**
$a := 4$;
$P2$
**end**

Supondo âmbito estático para a variáveis o que imprime o programa. E se o âmbito fosse dinâmico.

9. Escolhe uma linguagem de programação (com constructores imperativos) que te seja familiar. Quais as regras de âmbito de variáveis que usa? Exemplifica com um programa simples.

## Semântica operacionais para blocos Block

10. Usando a semântica natural para blocos determina os valores finais da variável $x$ e da variável $y$:

```
begin var y := 1;
      x := 1;
      begin var x := 2;
      y := x + 1
      end;
      x := y + x
end
```

11. Considera o seguinte programa **c** em **Block**:

```
begin var z := x;
      x := y;
      y := z
end
```

Usando a semântica natural para blocos **Block** e $s_0$ tal que $s_0(x) = 3$ e $s_0(y) = 5$, determina a semântica de do comando **c**.

12. Considera no `The Programming Languages Zoo: https://plzoo.andrej.com` a linguagem `comm`. Implementa uma versão dos programas dados nos exercícios 10 e 11 e compara os resultados.

# Semântica operacionais para While

## Semântica operacional natural (*big-step*)

$$\texttt{att}_{sn} \quad \langle x := a, s\rangle \longrightarrow s[x \mapsto \mathcal{A}[\![a]\!]\, s]$$

$$\texttt{skip}_{sn} \quad \langle \texttt{skip}, s\rangle \longrightarrow s$$

$$\texttt{comp}_{sn} \quad \frac{\langle c_1, s\rangle \longrightarrow s' \ , \ \langle c_2, s'\rangle \longrightarrow s''}{\langle c_1; c_2, s\rangle \longrightarrow s''}$$

$$\texttt{if}^{v}_{sn} \quad \frac{\langle c_1, s\rangle \longrightarrow s'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, s\rangle \longrightarrow s'} \text{ se } \mathcal{B}[\![b]\!]\, s = \mathsf{true}$$

$$\texttt{if}^{f}_{sn} \quad \frac{\langle c_2, s\rangle \longrightarrow s'}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, s\rangle \longrightarrow s'} \text{ se } \mathcal{B}[\![b]\!]\, s = \mathsf{false}$$

$$\texttt{while}^{v}_{sn} \quad \frac{\langle c, s\rangle \longrightarrow s',\ \langle \texttt{while } b \texttt{ do } c, s'\rangle \longrightarrow s''}{\langle \texttt{while } b \texttt{ do } c, s\rangle \longrightarrow s''} \text{ se } \mathcal{B}[\![b]\!]\, s = \mathsf{true}$$

$$\texttt{while}^{f}_{sn} \quad \langle \texttt{while } b \texttt{ do } c, s\rangle \longrightarrow s \text{ se } \mathcal{B}[\![b]\!]\, s = \mathsf{false}$$

## Semântica operacional estrutural (*small-step*)

$$\texttt{att}_{sos} \quad \langle x := a, s\rangle \Rightarrow s[x \mapsto \mathcal{A}[\![a]\!]\, s]$$

$$\texttt{skip}_{sos} \quad \langle \texttt{skip}, s\rangle \Rightarrow s$$

$$\texttt{comp}^{1}_{sos} \quad \frac{\langle c_1, s\rangle \Rightarrow \langle c_1', s'\rangle}{\langle c_1; c_2, s\rangle \Rightarrow \langle c_1'; c_2, s'\rangle}$$

$$\texttt{comp}^{2}_{sos} \quad \frac{\langle c_1, s\rangle \Rightarrow s'}{\langle c_1; c_2, s\rangle \Rightarrow \langle c_2, s'\rangle}$$

$$\texttt{if}^{v}_{sos} \quad \langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, s\rangle \Rightarrow \langle c_1, s\rangle \text{ se } \mathcal{B}[\![b]\!]\, s = \mathsf{true}$$

$$\texttt{if}^{f}_{sos} \quad \langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, s\rangle \Rightarrow \langle c_2, s\rangle \text{ se } \mathcal{B}[\![b]\!]\, s = \mathsf{false}$$

$$\texttt{while}_{sos} \quad \langle \texttt{while } b \texttt{ do } c, s\rangle \Rightarrow \langle \texttt{if } b \texttt{ then}(c; \texttt{while } b \texttt{ do } c) \texttt{ else skip}, s\rangle$$

## Semântica operacional natural para comandos do Block

$$\texttt{block}_{sn} \quad \frac{\langle D_V, s\rangle \longrightarrow_D s' \qquad \langle c, s'\rangle \longrightarrow s''}{\langle \texttt{begin } D_V \ c \texttt{ end}, s\rangle \longrightarrow s''[DV(D_V) \longmapsto s]}$$

onde $DV$ determina as variáveis declaradas em $D_V$:

$$\begin{aligned} DV(\texttt{var } x := a;\ D_V) &= \{x\} \cup DV(D_V) \\ DV(\varepsilon) &= \emptyset \end{aligned}$$

e $s'[X \longmapsto s] : Var \to \mathbb{Z}$ é definido por

$$s'[X \longmapsto s](x) = \begin{cases} s(x) & \text{se } x \in X \\ s'(x) & \text{se } x \notin X \end{cases}$$

e ainda

$$\texttt{var}_{sn} \quad \frac{\langle D_V, s[x \mapsto \mathcal{A}[\![a]\!]\, s]\rangle \;\longrightarrow_{D}\; s'}{\langle \texttt{var}\ x := a;\ D_V, s\rangle \;\longrightarrow_{D}\; s'}$$

$$\texttt{empty}_{sn} \quad \langle \varepsilon, s\rangle \;\longrightarrow_{D}\; s$$

As restantes regras coincidem com a semântica natural para **While**.