Linguagem com construções imperativas (L2)

Sintaxe abstrata:

$$
\begin{array}{lcll}
e & \in & \mathsf{Expr} & \\
e & ::= & n \mid b \mid e_1\ op\ e_2 \mid \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 & \\
 & \mid & x \mid \mathsf{let}\ x : T = e_1\ \mathsf{in}\ e_2 & \\
 & \mid & e_1 := e_2 \mid\ !\ e \mid \mathsf{new}\ e \mid () \mid \mathsf{while}\ e_1\ \mathsf{do}\ e_2 \mid e_1; e_2 \mid \boxed{\mathrm{l}} &
\end{array}
$$

$$
\begin{array}{lcll}
v & \in & \mathsf{Values} & \\
v & ::= & n \mid b \mid () \mid \boxed{\mathrm{l}} &
\end{array}
$$

$$
\begin{array}{lcll}
n & \in & \mathbb{Z} & \\
b & \in & \{\mathsf{true}, \mathsf{false}\} & \\
op & \in & \{+, <, \ldots\} & \text{(operações binárias, no mínimo adição e menor-que)} \\
\boxed{\mathrm{l}} & \in & \mathsf{Locations} & \text{(localizações/endereços de memória)}
\end{array}
$$

Semântica operacional *small-step*:

$$\frac{[\![n]\!] = [\![n_1]\!] + [\![n_2]\!]}{n_1 + n_2, \sigma \longrightarrow n, \sigma} \quad (\textsc{op}+)$$

$$\frac{[\![n_1]\!] < [\![n_2]\!]}{n_1 < n_2, \sigma \longrightarrow \mathsf{true}, \sigma} \quad (\textsc{op}<\textsc{true})$$

$$\frac{[\![n_1]\!] \geq [\![n_2]\!]}{n_1 < n_2, \sigma \longrightarrow \mathsf{false}, \sigma} \quad (\textsc{op}<\textsc{false})$$

$$\frac{e_1, \sigma \longrightarrow e_1', \sigma'}{e_1\ op\ e_2, \sigma \longrightarrow e_1'\ op\ e_2, \sigma'} \quad (\textsc{op}1)$$

$$\frac{e_2, \sigma \longrightarrow e_2', \sigma'}{v\ op\ e_2, \sigma \longrightarrow v\ op\ e_2', \sigma'} \quad (\textsc{op}2)$$

$$\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3, \sigma \longrightarrow e_2, \sigma \quad (\textsc{if}1)$$

$$\mathsf{if}\ \mathsf{false}\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3, \sigma \longrightarrow e_3, \sigma \quad (\textsc{if}2)$$

$$\frac{e_1, \sigma \longrightarrow e_1', \sigma'}{\mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3, \sigma \longrightarrow \mathsf{if}\ e_1'\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3, \sigma'} \quad (\textsc{if}3)$$

$$\frac{e_1, \sigma \longrightarrow e_1', \sigma'}{\mathsf{let}\ x : T = e_1\ \mathsf{in}\ e_2, \sigma \longrightarrow \mathsf{let}\ x : T = e_1'\ \mathsf{in}\ e_2, \sigma'} \quad (\textsc{e-let}1)$$

$$\frac{}{\mathsf{let}\ x : T = v\ \mathsf{in}\ e_2, \sigma \longrightarrow \{v/x\}\ e_2, \sigma} \quad (\textsc{e-let}2)$$

$$\frac{l \in Dom(\sigma)}{l := v, \sigma \longrightarrow (), \sigma[l \mapsto v]} \quad (\textsc{atr}1)$$

$$\frac{e, \sigma \longrightarrow e', \sigma'}{l := e, \sigma \longrightarrow l := e', \sigma'} \quad (\textsc{atr}2)$$

$$\frac{e_1, \sigma \longrightarrow e_1', \sigma'}{e_1 := e_2, \sigma \longrightarrow e_1' := e_2, \sigma'} \quad (\textsc{atr})$$

$$\frac{l \in Dom(\sigma) \quad \sigma(l) = v}{!\ l, \sigma \longrightarrow v, \sigma} \quad (\textsc{deref}1)$$

$$\frac{e, \sigma \longrightarrow e', \sigma'}{!\ e, \sigma \longrightarrow\ !\ e', \sigma'} \quad \text{(DEREF)}$$

$$\frac{l \notin Dom(\sigma)}{\mathsf{new}\ v, \sigma \longrightarrow l, \sigma[l \mapsto v]} \quad \text{(NEW1)}$$

$$\frac{e, \sigma \longrightarrow e', \sigma'}{\mathsf{new}\ e, \sigma \longrightarrow \mathsf{new}\ e', \sigma'} \quad \text{(NEW)}$$

$$\frac{}{(); e_2, \sigma \longrightarrow e_2, \sigma} \quad \text{(SEQ1)}$$

$$\frac{e_1, \sigma \longrightarrow e_1', \sigma'}{e_1; e_2, \sigma \longrightarrow e_1'; e_2, \sigma'} \quad \text{(SEQ)}$$

$$\mathsf{while}\ e_1\ \mathsf{do}\ e_2, \sigma \longrightarrow \mathsf{if}\ e_1\ \mathsf{then}\ (e_2; \mathsf{while}\ e_1\ \mathsf{do}\ e_2)\ \mathsf{else}\ (), \sigma \quad \text{(E-WHILE)}$$

Sistema de Tipos:

$$\frac{}{\Gamma \vdash n : \mathsf{int}} \quad \text{(T-INT)}$$

$$\frac{}{\Gamma \vdash b : \mathsf{bool}} \quad \text{(T-BOOL)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{int} \qquad \Gamma \vdash e_2 : \mathsf{int}}{\Gamma \vdash e_1 + e_2 : \mathsf{int}} \quad \text{(T-OP+)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{int} \qquad \Gamma \vdash e_2 : \mathsf{int}}{\Gamma \vdash e_1 < e_2 : \mathsf{bool}} \quad \text{(T-OP<)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{bool} \qquad \Gamma \vdash e_2 : T \qquad \Gamma \vdash e_3 : T}{\Gamma \vdash \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3 : T} \quad \text{(T-IF)}$$

$$\frac{\Gamma(x) = T}{\Gamma \vdash x : T} \quad \text{(T-VAR)}$$

$$\frac{\Gamma \vdash e_1 : T \qquad \Gamma, x \mapsto T \vdash e_2 : T'}{\Gamma \vdash \mathsf{let}\ x : T = e_1\ \mathsf{in}\ e_2 : T'} \quad \text{(T-LET)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{ref}\ T \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1 := e_2 : \mathsf{unit}} \quad \text{(T-ATR)}$$

$$\frac{\Gamma \vdash e : \mathsf{ref}\ T}{\Gamma \vdash !\ e : T} \quad \text{(T-DEREF)}$$

$$\frac{\Gamma \vdash e : T}{\Gamma \vdash \mathsf{new}\ e : \mathsf{ref}\ T} \quad \text{(T-NEW)}$$

$$\frac{}{\Gamma \vdash () : \mathsf{unit}} \quad \text{(T-UNIT)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{bool} \qquad \Gamma \vdash e_2 : \mathsf{unit}}{\Gamma \vdash \mathsf{while}\ e_1\ \mathsf{do}\ e_2 : \mathsf{unit}} \quad \text{(T-WHILE)}$$

$$\frac{\Gamma \vdash e_1 : \mathsf{unit} \qquad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1;\ e_2 : T} \quad \text{(T-SEQ)}$$

Trabalho

O trabalho consiste em implementar em OCaml a inferência de tipo e o avaliador *small step* para a linguagem L2 da especificação acima

O trabalho será avaliado da seguinte forma:

- nota máxima 8,0 para os trabalhos que implementarem somente a inferência de tipos OU somente o avaliador *small step* para L2 conforme a especificação dada acima
- nota máxima 10,0 para os trabalhos que implementarem a inferência de tipos E o avaliador *small step*

Arquivo com as definições dos datatypes necessários e com alguns casos de teste referentes a L2 da especificação dada será disponibilizado no Moodle da disciplina.

O trabalho deve ser realizado em grupos de 3 componentes, e ser entregue via Moodle no prazo especificado. Após a entrega, o trabalho será apresentado em laboratório pelos componentes do grupo, conforme cronograma de apresentações disponível no Moodle.