

Sintaxe abstrata:

e	\in	Expr
e	$::=$	$n \mid b \mid e_1 \text{ op } e_2 \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3$ $\mid x \mid \text{let } x:T = e_1 \text{ in } e_2$ $\mid e_1 := e_2 \mid !e \mid \text{new } e \mid () \mid \text{while } e_1 \text{ do } e_2 \mid e_1; e_2 \mid \boxed{1}$ $\mid \text{read } () \mid \text{print } e$
v	\in	Values
v	$::=$	$n \mid b \mid () \mid \boxed{1}$
T	\in	Types
T	$::=$	$\text{int} \mid \text{bool} \mid \text{ref } T \mid \text{unit}$
in, out	\in	\mathbb{Z}^* (listas finitas de inteiros)
in	$::=$	$\boxed{} \mid n :: \text{in}$
out	$::=$	$\boxed{} \mid n :: \text{out}$
$n.l$		significa prefixar a lista l com o elemento n
$l.n$		significa posfixar a lista l com o elemento n
n	\in	\mathbb{Z}
b	\in	$\{\text{true}, \text{false}\}$
op	\in	$\{+, <, \dots\}$ (operações binárias, no mínimo adição e menor-que)
$\boxed{1}$	\in	Locations (localizações/endereços de memória)

 Semântica operacional *small-step*:

$\frac{\llbracket n \rrbracket = \llbracket n_1 \rrbracket + \llbracket n_2 \rrbracket}{n_1 + n_2, \sigma, \text{in}, \text{out} \longrightarrow n, \sigma, \text{in}, \text{out}}$	(OP+)
$\frac{\llbracket n_1 \rrbracket < \llbracket n_2 \rrbracket}{n_1 < n_2, \sigma, \text{in}, \text{out} \longrightarrow \text{true}, \sigma, \text{in}, \text{out}}$	(OP<TRUE)
$\frac{\llbracket n_1 \rrbracket \geq \llbracket n_2 \rrbracket}{n_1 < n_2, \sigma, \text{in}, \text{out} \longrightarrow \text{false}, \sigma, \text{in}, \text{out}}$	(OP<FALSE)
$\frac{e_1, \sigma, \text{in}, \text{out} \longrightarrow e'_1, \sigma', \text{in}', \text{out}'}{e_1 \text{ op } e_2, \sigma, \text{in}, \text{out} \longrightarrow e'_1 \text{ op } e_2, \sigma', \text{in}', \text{out}'}$	(OP1)
$\frac{e_2, \sigma, \text{in}, \text{out} \longrightarrow e'_2, \sigma', \text{in}', \text{out}'}{v \text{ op } e_2, \sigma, \text{in}, \text{out} \longrightarrow v \text{ op } e'_2, \sigma', \text{in}', \text{out}'}$	(OP2)
$\text{if true then } e_2 \text{ else } e_3, \sigma, \text{in}, \text{out} \longrightarrow e_2, \sigma, \text{in}, \text{out}$	(IF1)
$\text{if false then } e_2 \text{ else } e_3, \sigma, \text{in}, \text{out} \longrightarrow e_3, \sigma, \text{in}, \text{out}$	(IF2)
$\frac{e_1, \sigma, \text{in}, \text{out} \longrightarrow e'_1, \sigma', \text{in}', \text{out}'}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3, \sigma, \text{in}, \text{out} \longrightarrow \text{if } e'_1 \text{ then } e_2 \text{ else } e_3, \sigma', \text{in}', \text{out}'}$	(IF3)
$\frac{e_1, \sigma, \text{in}, \text{out} \longrightarrow e'_1, \sigma', \text{in}', \text{out}'}{\text{let } x:T = e_1 \text{ in } e_2, \sigma, \text{in}, \text{out} \longrightarrow \text{let } x:T = e'_1 \text{ in } e_2, \sigma', \text{in}', \text{out}'}$	(E-LET1)
$\frac{}{\text{let } x:T = v \text{ in } e_2, \sigma, \text{in}, \text{out} \longrightarrow \{v/x\} e_2, \sigma, \text{in}, \text{out}}$	(E-LET2)
$\frac{l \in \text{Dom}(\sigma)}{l := v, \sigma, \text{in}, \text{out} \longrightarrow (), \sigma[l \mapsto v], \text{in}, \text{out}}$	(ATR1)

$$\frac{e, \sigma, \text{in}, \text{out} \rightarrow e', \sigma', \text{in}', \text{out}'}{l := e, \sigma, \text{in}, \text{out} \rightarrow l := e', \sigma', \text{in}', \text{out}'} \quad (\text{ATR2})$$

$$\frac{e_1, \sigma, \text{in}, \text{out} \rightarrow e'_1, \sigma', \text{in}', \text{out}'}{e_1 := e_2, \sigma, \text{in}, \text{out} \rightarrow e'_1 := e_2, \sigma', \text{in}', \text{out}'} \quad (\text{ATR})$$

$$\frac{l \in \text{Dom}(\sigma) \quad \sigma(l) = v}{! l, \sigma, \text{in}, \text{out} \rightarrow v, \sigma, \text{in}, \text{out}} \quad (\text{DEREF1})$$

$$\frac{e, \sigma, \text{in}, \text{out} \rightarrow e', \sigma', \text{in}', \text{out}'}{! e, \sigma, \text{in}, \text{out} \rightarrow ! e', \sigma', \text{in}', \text{out}'} \quad (\text{DEREF})$$

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

$$\frac{e, \sigma, \text{in}, \text{out} \rightarrow e', \sigma', \text{in}', \text{out}'}{\text{new } e, \sigma, \text{in}, \text{out} \rightarrow \text{new } e', \sigma', \text{in}', \text{out}'} \quad (\text{NEW})$$

$$\frac{}{() ; e_2, \sigma, \text{in}, \text{out} \rightarrow e_2, \sigma, \text{in}, \text{out}} \quad (\text{SEQ1})$$

$$\frac{e_1, \sigma, \text{in}, \text{out} \rightarrow e'_1, \sigma', \text{in}', \text{out}'}{e_1 ; e_2, \sigma, \text{in}, \text{out} \rightarrow e'_1 ; e_2, \sigma', \text{in}', \text{out}'} \quad (\text{SEQ})$$

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

$$\text{print } n, \sigma, \text{in}, \text{out} \rightarrow (), \sigma, \text{in}, \text{out}.n \quad (\text{PRINT-N})$$

$$\frac{e, \sigma, \text{in}, \text{out} \rightarrow e', \sigma', \text{in}', \text{out}'}{\text{print } e, \sigma, \text{in}, \text{out} \rightarrow \text{print } e', \sigma', \text{in}', \text{out}'} \quad (\text{PRINT})$$

$$\text{read } (), \sigma, n, \text{in}, \text{out} \rightarrow n, \sigma, \text{in}, \text{out} \quad (\text{READ})$$

Sistema de Tipos:

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

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

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

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

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

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

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

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

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

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

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

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

$$\frac{}{\Gamma \vdash \text{read } () : \text{int}} \quad (\text{T-READ})$$

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

$$\frac{\Gamma \vdash e : \text{int}}{\Gamma \vdash \text{print } e : \text{unit}} \quad (\text{T-PRINT})$$

Trabalho

O trabalho consiste em implementar em OCaml um interpretador para a linguagem L2 da especificação acima e com variações definidas abaixo que serão deixadas propositalmente subespecificadas.

O trabalho será avaliado da seguinte forma:

- nota máxima 9,0 para os trabalhos que implementarem somente L2 conforme a especificação dada acima
- nota máxima 10,0 para os trabalhos que implementarem também uma dentre as seguintes opções:
 - arrays
 - mecanismo de exceções
 - expressão *for* para repetições

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 grupo (2 a 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.