4190.310 Programming Language
The K-- Language

TA {08 최원태 신희제, 09 허기홍 김희정}

1 Syntax

Expression e	$\rightarrow$	$\texttt{unit}$	unit
	$\mid$	x := e	assignment
	$\mid$	e ; e	sequence
	$\mid$	if e then e else e	branch
	$\mid$	while e do e	while loop
	$\mid$	for x := e to e do e	for loop
	$\mid$	read x	input
	$\mid$	write e	output
	$\mid$	let x := e in e	variable binding
	$\mid$	let proc f(x)= e in e	procedure binding
	$\mid$	f(e)	call by value
	$\mid$	$f\texttt{<}x\texttt{>}$	call by reference
	$\mid$	n	integer
	$\mid$	true $\mid$ false	boolean
	$\mid$	x	identifier
	$\mid$	e + e $\mid$ e - e $\mid$ e * e $\mid$ e / e	arithmetic operation
	$\mid$	e $<$ e $\mid$ e = e $\mid$ not e	conditional operation

1.1 Program

A program is an expression.

1.2 Identifiers

Alpha-numeric identifiers are [a-zA-Z][a-zA-Z0-9_]*. Identifiers are case sensitive: `z` and `Z` are different. The reserved words cannot be used as identifiers: `unit true false not if then else let in end proc while do for to read write`

1.3 Numbers/Comments

Numbers are integers, optionally prefixed with `-`(for negative integer): -?[0-9]+.

A comment is any character sequence within the comment block `(* *)`. The comment block can be nested.

1.4 Precedence/Associativity

In parsing K-- program text, the precedence of the K-- constructs in decreasing order is as follows. Symbols in the same set have identical precedence. Symbols with subscript L (respectively R) are left (respectively right) associative. Symbols without subscript are nonassociative.

$$
\begin{array}{l}
\{\texttt{not}\}_R,\\
\{\texttt{*},\texttt{/}\}_L,\\
\{\texttt{+},\texttt{-}\}_L,\\
\{\texttt{=},\texttt{<}\}_L,\\
\{\texttt{write}\}_R,\\
\{\texttt{:=}\}_R,\\
\{\texttt{else}\},\\
\{\texttt{then}\},\\
\{\texttt{do}\},\\
\{\texttt{;}\}_L,\\
\{\texttt{in}\}
\end{array}
$$

For example, K-- program

```
x := e1; e2                 ⇒  (x := e1) ; e2
while e do e1; e2           ⇒  (while e do e1); e2
if e1 then e2 else e3; e4   ⇒  (if e1 then e2 else e3); e4
```

Rule of thumb: for your test programs, if your programs are hard to read (hence can be parsed not as you expected) then put parentheses around.

2 Domains

$$
\begin{array}{lclcl}
n & \in & \mathbb{Z} & & \text{integer} \\
b & \in & \mathbb{B} & & \text{boolean} \\
v & \in & Val & = & \mathbb{Z} + \mathbb{B} + \{\cdot\} \\
\sigma & \in & Env & = & Id \xrightarrow{fin} Addr + Procedure \\
M & \in & Mem & = & Addr \xrightarrow{fin} Val \\
x, y & \in & Id & & \text{identifier} \\
l & \in & Addr & & \text{an index set} \\
 & & Procedure & = & Id \times Expression \times Env
\end{array}
$$

3 Semantics

$$\text{TRUE}\ \frac{}{\sigma, M \vdash \texttt{true} \Rightarrow true, M} \qquad \text{FALSE}\ \frac{}{\sigma, M \vdash \texttt{false} \Rightarrow false, M}$$

$$\text{NUM}\ \frac{}{\sigma, M \vdash \texttt{n} \Rightarrow n, M} \qquad \text{UNIT}\ \frac{}{\sigma, M \vdash \texttt{unit} \Rightarrow \cdot, M}$$

$$\text{VAR}\ \frac{}{\sigma, M \vdash x \Rightarrow M(\sigma(x)), M}$$

$$\text{ADD}\ \frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash e_1 \ \texttt{+}\ e_2 \Rightarrow n_1 + n_2, M''}$$

$$\text{SUB}\ \frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash e_1 \ \texttt{-}\ e_2 \Rightarrow n_1 - n_2, M''}$$

$$\text{MUL}\ \frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash e_1 \ \texttt{*}\ e_2 \Rightarrow n_1 * n_2, M''}$$

$$\text{DIV}\ \frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash e_1 \ \texttt{/}\ e_2 \Rightarrow n_1 / n_2, M''}$$

$$\text{EQUALT}\ \frac{\sigma, M \vdash e_1 \Rightarrow v_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v_2, M''}{\sigma, M \vdash e_1 \texttt{ = } e_2 \Rightarrow \texttt{true}, M''}\ \begin{array}{l} v_1 = v_2 = n \\ \vee\ v_1 = v_2 = b \\ \vee\ v_1 = v_2 = \cdot \end{array}$$

$$\text{EQUALF}\ \frac{\sigma, M \vdash e_1 \Rightarrow v_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v_2, M''}{\sigma, M \vdash e_1 \texttt{ = } e_2 \Rightarrow \texttt{false}, M''}\ \texttt{otherwise}$$

$$\text{LESS}\ \frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash e_1 \texttt{ < } e_2 \Rightarrow n_1 < n_2, M''}$$

$$\text{NOT}\ \frac{\sigma, M \vdash e \Rightarrow b, M'}{\sigma, M \vdash \texttt{not}\ e \Rightarrow \mathit{not}\ b, M'}$$

$$\text{ASSIGN}\ \frac{\sigma, M \vdash e \Rightarrow v, M'}{\sigma, M \vdash x \texttt{ := } e \Rightarrow v, M'\{\sigma(x) \mapsto v\}}$$

$$\text{SEQ}\ \frac{\sigma, M \vdash e_1 \Rightarrow v_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v_2, M''}{\sigma, M \vdash e_1 \texttt{ ; } e_2 \Rightarrow v_2, M''}$$

$$\text{IFT}\ \frac{\sigma, M \vdash e \Rightarrow \mathit{true}, M' \qquad \sigma, M' \vdash e_1 \Rightarrow v, M''}{\sigma, M \vdash \texttt{if}\ e\ \texttt{then}\ e_1\ \texttt{else}\ e_2 \Rightarrow v, M''}$$

$$\text{IFF}\ \frac{\sigma, M \vdash e \Rightarrow \mathit{false}, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v, M''}{\sigma, M \vdash \texttt{if}\ e\ \texttt{then}\ e_1\ \texttt{else}\ e_2 \Rightarrow v, M''}$$

$$\text{WHILEF}\ \frac{\sigma, M \vdash e_1 \Rightarrow \mathit{false}, M'}{\sigma, M \vdash \texttt{while}\ e_1\ \texttt{do}\ e_2 \Rightarrow \cdot, M'}$$

$$\text{WHILET}\ \frac{\begin{array}{c}\sigma, M \vdash e_1 \Rightarrow \mathit{true}, M' \\ \sigma, M' \vdash e_2 \Rightarrow v_1, M_1 \qquad \sigma, M_1 \vdash \texttt{while}\ e_1\ \texttt{do}\ e_2 \Rightarrow v_2, M_2\end{array}}{\sigma, M \vdash \texttt{while}\ e_1\ \texttt{do}\ e_2 \Rightarrow v_2, M_2}$$

$$\text{FORT}\ \frac{\begin{array}{c}\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M'' \\ \sigma, M''\{\sigma(x) \mapsto n_1 + 0\} \vdash e_3 \Rightarrow v_0, M_0 \\ \vdots \\ \sigma, M_{n_2-n_1-1}\{\sigma(x) \mapsto n_1 + (n_2 - n_1)\} \vdash e_3 \Rightarrow v_{n_2-n_1}, M_{n_2-n_1}\end{array}}{\sigma, M \vdash \texttt{for}\ x\ \texttt{:=}\ e_1\ \texttt{to}\ e_2\ \texttt{do}\ e_3 \Rightarrow \cdot, M_{n_2-n_1}}\ n_2 \geq n_1$$

$$\text{FORF}\ \frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash \texttt{for}\ x\ \texttt{:=}\ e_1\ \texttt{to}\ e_2\ \texttt{do}\ e_3 \Rightarrow \cdot, M''}\ n_2 < n_1$$

$$\text{LETV}\ \frac{\begin{array}{c}\sigma, M \vdash e_1 \Rightarrow v, M' \\ \sigma\{x \mapsto l\}, M'\{l \mapsto v\} \vdash e_2 \Rightarrow v', M''\end{array}}{\sigma, M \vdash \texttt{let}\ x\ \texttt{:=}\ e_1\ \texttt{in}\ e_2 \Rightarrow v', M''}\ l \notin \mathit{Dom}\, M'$$

$$\text{LETF}\ \frac{\sigma\{f \mapsto \langle x, e_1, \sigma\rangle\}, M \vdash e_2 \Rightarrow v, M'}{\sigma, M \vdash \texttt{let proc}\ f\texttt{(}x\texttt{)}\ \texttt{=}\ e_1\ \texttt{in}\ e_2 \Rightarrow v, M'}$$

$$\text{CALLV}\ \frac{\begin{array}{c}\sigma, M \vdash e \Rightarrow v, M' \\ \sigma'\{x \mapsto l\}\{f \mapsto \langle x, e', \sigma'\rangle\}, M'\{l \mapsto v\} \vdash e' \Rightarrow v', M''\end{array}}{\sigma, M \vdash f\texttt{(}e\texttt{)} \Rightarrow v', M''}\ \begin{array}{l}\sigma(f) = \langle x, e', \sigma'\rangle \\ l \notin \mathit{Dom}\, M'\end{array}$$

$$\text{CALLR}\ \frac{\sigma'\{x \mapsto \sigma(y)\}\{f \mapsto \langle x, e, \sigma'\rangle\}, M \vdash e \Rightarrow v, M'}{\sigma, M \vdash f\texttt{<}y\texttt{>} \Rightarrow v, M'}\ \sigma(f) = \langle x, e, \sigma'\rangle$$

$$\text{READ}\ \frac{}{\sigma, M \vdash \texttt{read}\ x \Rightarrow n, M\{\sigma(x) \mapsto n\}}$$

$$\text{WRITE}\ \frac{\sigma, M \vdash e \Rightarrow v, M'}{\sigma, M \vdash \texttt{write}\ e \Rightarrow v, M'}$$