4190.310 Programming Language

The K- Language

2011 Spring

1 Syntax

Expression e	$\rightarrow$	`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
	$\mid$	`read` x	input
	$\mid$	`write` e	output
	$\mid$	`let` x `:=` e `in` e	variable binding
	$\mid$	`let proc` f`(`x_1`,`x_2`,`$\cdots$`,`x_n`)` `=` e `in` e	procedure binding
	$\mid$	f`(`e_1`,`e_2`,`$\cdots$`,`e_n`)`	call by value
	$\mid$	f`<`x_1`,`x_2`,`$\cdots$`,`x_n`>`	call by reference
	$\mid$	n	integer
	$\mid$	`true` $\mid$ `false`	boolean
	$\mid$	`{}` $\mid$ `{`x_1`:=`e_1`,`x_2`:=`e_2`,`$\cdots$`,`x_n`:=`e_n`}`	record
	$\mid$	e`.`x	record lookup
	$\mid$	e`.`x `:=` e	record assignment
	$\mid$	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
	$\mid$	`malloc` e $\mid$ `&`x $\mid$ `&`e`.`x	
	$\mid$	`*`e $\mid$ `*`e `:=` e	locations as values

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 proc while do read write malloc.

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 (* *). Comment blocks can be nested.

1.4 Precedence/Associativity

In parsing K- program text, the precedence of 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.

$\{.\}_L$,
$\{$not, malloc, &, *$\}_R$,
$\{*, /\}_L$,
$\{+, -\}_L$,
$\{=, <\}_L$,
$\{$write$\}_R$,
$\{:=\}_R$,
$\{$else$\}$,
$\{$then$\}$,
$\{$do$\}$,
$\{;\}_L$,
$\{$in$\}$

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: If your test programs are hard to read (hence can be parsed not as you expected) then put parentheses around.

2 Domains

n	$\in$	$\mathbb{Z}$		integer
b	$\in$	$\mathbb{B}$		boolean
r	$\in$	$Record$	$=$	$Id \xrightarrow{fin} Addr$
v	$\in$	Val	$=$	$\mathbb{Z} + \mathbb{B} + \{\cdot\} + Record + Addr$
σ	$\in$	Env	$=$	$Id \xrightarrow{fin} Addr + Procedure$
M	$\in$	Mem	$=$	$Addr \xrightarrow{fin} Val$
x, y	$\in$	Id		identifier
l	$\in$	$Addr$		address
		$Procedure$	$=$	$(Id \times Id \times \cdots) \times Expression \times Env$

3 Semantics

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

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

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

$$\text{RECF}\ \frac{}{\sigma, M \vdash \texttt{\{\}} \Rightarrow \cdot, M}$$

$$\text{RECT}\ \frac{\begin{array}{c}\sigma, M \vdash e_1 \Rightarrow v_1, M_1 \\ \sigma, M_1 \vdash e_2 \Rightarrow v_2, M_2 \\ \vdots \\ \sigma, M_{n-1} \vdash e_n \Rightarrow v_n, M_n\end{array}}{\begin{array}{c}\sigma, M \vdash \{x_1\ \texttt{:=}\ e_1\texttt{,}\cdots\texttt{,}x_n\ \texttt{:=}\ e_n\} \Rightarrow \\ \{x_1 \mapsto l_1, \cdots, x_n \mapsto l_n\}, M_n\{l_1 \mapsto v_1, \cdots, l_n \mapsto v_n\}\end{array}}\ l_i \notin Dom\ M_n$$

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

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

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

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

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

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

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

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

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

$$\text{RECASSIGN}\ \frac{\sigma, M \vdash e_1 \Rightarrow r, M_1 \qquad \sigma, M_1 \vdash e_2 \Rightarrow v, M_2}{\sigma, M \vdash e_1.x\ \texttt{:=}\ e_2 \Rightarrow v, M_2\{r(x) \mapsto v\}}$$

$$\text{RECLOOKUP}\ \frac{\sigma, M \vdash e \Rightarrow r, M'}{\sigma, M \vdash e.x \Rightarrow M'(r(x)), M'}$$

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

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

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

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

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

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

$$\text{LETF}\ \frac{\sigma\{f \mapsto \langle (x_1, \cdots, x_n), e_1, \sigma\rangle\}, M \vdash e_2 \Rightarrow v, M'}{\sigma, M \vdash \texttt{let proc}\ f\texttt{(}x_1\texttt{,}\cdots\texttt{,}x_n\texttt{)}\ \texttt{=}\ e_1\ \texttt{in}\ e_2 \Rightarrow v, M'}$$

$$\text{CALLV}\ \frac{\begin{array}{c}\sigma, M \vdash e_1 \Rightarrow v_1, M_1 \\ \sigma, M_1 \vdash e_2 \Rightarrow v_2, M_2 \\ \vdots \\ \sigma, M_{n-1} \vdash e_n \Rightarrow v_n, M_n \\ \sigma'\{x_1 \mapsto l_1\}\cdots\{x_n \mapsto l_n\}\{f \mapsto \langle (x_1, \cdots, x_n), e', \sigma'\rangle\}, \\ M_n\{l_1 \mapsto v_1\}\cdots\{l_n \mapsto v_n\} \vdash e' \Rightarrow v', M'\end{array}}{\sigma, M \vdash f\texttt{(}e_1\texttt{,}\cdots\texttt{,}e_n\texttt{)} \Rightarrow v', M'}\ \begin{array}{l}\sigma(f) = \langle (x_1, \cdots, x_n), e', \sigma'\rangle \\ l_i \notin Dom\, M'\end{array}$$

$$\text{CALLR}\ \frac{\begin{array}{c}\sigma'\{x_1 \mapsto \sigma(y_1)\}\cdots\{x_n \mapsto \sigma(y_n)\}\{f \mapsto \langle (x_1, \cdots, x_n), e, \sigma'\rangle\}, \\ M \vdash e \Rightarrow v, M'\end{array}}{\sigma, M \vdash f\texttt{<}y_1\texttt{,}\cdots\texttt{,}y_n\texttt{>} \Rightarrow v, M'}\ \sigma(f) = \langle (x_1, \cdots, x_n), e, \sigma'\rangle$$

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

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

$$\text{MALLOC}\ \frac{\sigma, M \vdash e \Rightarrow n, M'}{\sigma, M \vdash \texttt{malloc}\ \ e \Rightarrow l, M'}\ n > 0, \{l, l+1, ..., l+n-1\} \nsubseteq dom(M')$$

$$\text{AMPER}\ \frac{}{\sigma, M \vdash \texttt{\&}x \Rightarrow \sigma(x), M}$$

$$\text{AMPERFIELD}\ \frac{\sigma, M \vdash e \Rightarrow r, M'}{\sigma, M \vdash \texttt{\&}e.x \Rightarrow r(x), M'}$$

$$\text{STAR}\ \frac{\sigma, M \vdash e \Rightarrow l, M'}{\sigma, M \vdash *e \Rightarrow M'(l), M'}$$

$$\text{ASSIGNSTAR}\ \frac{\sigma, M \vdash e_1 \Rightarrow l, M_1 \qquad \sigma, M_1 \vdash e_2 \Rightarrow v, M_2}{\sigma, M \vdash *e_1 := e_2 \Rightarrow v, M_2\{l \mapsto v\}}$$