The K- Language[*]

Daejun Park Heejong Lee Kwangkeun Yi

Programming Research Laboratory
Seoul National University

October 2, 2006

1 Syntax

The syntax of K- is:

$$
\begin{array}{rcl}
E & \rightarrow & num \mid \texttt{true} \mid \texttt{false} \mid \texttt{unit} \\
 & & \mid x \\
 & & \mid \texttt{call}\ f\texttt{(}E\texttt{)} \mid \texttt{call}\ f\texttt{<}x\texttt{>} \\
 & & \mid E\ \texttt{+}\ E \mid E\ \texttt{-}\ E \mid E\ \texttt{*}\ E \mid E\ \texttt{/}\ E \\
 & & \mid E\ \texttt{=}\ E \mid E\ \texttt{<}\ E \mid \texttt{not}\ E \\
 & & \mid x\ \texttt{:=}\ E \\
 & & \mid E\ \texttt{;}\ E \\
 & & \mid \texttt{if}\ E\ \texttt{then}\ E\ \texttt{else}\ E\ \texttt{end} \mid \texttt{if}\ E\ \texttt{then}\ E\ \texttt{end} \\
 & & \mid \texttt{while}\ E\ \texttt{do}\ E\ \texttt{end} \mid \texttt{for}\ x\ \texttt{:=}\ E\ \texttt{to}\ E\ \texttt{do}\ E\ \texttt{end} \\
 & & \mid \texttt{read}\ x \mid \texttt{write}\ E \\
 & & \mid \texttt{let}\ x\ \texttt{:=}\ E\ \texttt{in}\ E\ \texttt{end} \mid \texttt{let procedure}\ f\texttt{(}x\texttt{)}\ \texttt{=}\ E\ \texttt{in}\ E\ \texttt{end} \\
 & & \mid \{x_1\ \texttt{:=}\ E_1\texttt{,}\ \ x_2\ \texttt{:=}\ E_2\}^1 \mid E\texttt{.}x \mid E\texttt{.}x\ \texttt{:=}\ E \\
 & & \mid \texttt{malloc(}E\texttt{)} \mid \texttt{\&}x \mid \texttt{*}E \mid E\ \texttt{:=}\ E
\end{array}
$$

1.1 Numbers

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

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: `true false call not if then else end while do for to read write let in procedure malloc unit`

[*]SNU 4190.310 Programming Languages (Fall 2006)

[1]For the sake of concision, we assume that a record has only two fields.

1.3 Procedence / 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.

$$
\begin{array}{l}
\{\texttt{.}\}_L, \\
\{\texttt{not}, \texttt{\&}, \texttt{*}\}_R, \\
\{\texttt{*}, \texttt{/}\}_L, \\
\{\texttt{+}, \texttt{-}\}_L, \\
\{\texttt{<}\}_L, \\
\{\texttt{=}\}_L, \\
\{\texttt{:=}\}_R, \\
\{\texttt{write}\}, \\
\{\texttt{;}\}_L
\end{array}
$$

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.

1.4 Comments

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

2 Dynamic Semantics

$$
\begin{array}{rcllcl}
\sigma & \in & \mathit{Env} & = & \mathit{Id} \xrightarrow{\text{fin}} (\mathit{Loc} + \mathit{Procedure}) \\
x, y, f & \in & \mathit{Id} & & \\
\langle b, o \rangle, l & \in & \mathit{Loc} & = & \mathit{Base} \times \mathit{Offset} \\
 & & \mathit{Offset} & = & \mathbb{Z} \\
\langle x, E, \sigma \rangle & \in & \mathit{Procedure} & = & \mathit{Id} \times E \times \mathit{Env} \\
M & \in & \mathit{Mem} & = & \mathit{Loc} \xrightarrow{\text{fin}} \mathit{Val} \\
v & \in & \mathit{Val} & = & \mathbb{Z} + \mathbb{B} + \mathit{Record} + \mathit{Loc} + \{\cdot\} + \{\bot\} \\
n & \in & \mathbb{Z} & & \\
\mathit{tf} & \in & \mathbb{B} & = & \{T, F\} \\
\{x_1 \mapsto l_1, x_2 \mapsto l_2\}, r & \in & \mathit{Record} & = & \mathit{Id} \xrightarrow{\text{fin}} \mathit{Loc}
\end{array}
$$

Notation:

- We write $\{x_1 \mapsto v_1, \cdots, x_n \mapsto v_n\}$ for a finite function f. The domain $\mathit{dom}(f)$ is $\{x_1, \cdots, x_n\}$.
- We write $f(x)$ for v if $x \mapsto v \in f$. If $x \mapsto v \notin f$ then $f(x)$ is not defined.
- We write $f[v/x]$ for

$$
\begin{array}{rl}
f \cup \{x \mapsto v\} & \text{if } x \notin \mathit{dom}(f) \\
(f \setminus \{x \mapsto f(x)\}) \cup \{x \mapsto v\} & \text{if } x \in \mathit{dom}(f).
\end{array}
$$

The semantics rules precisely defines how relations of the form

$$\sigma, M \vdash E \Downarrow v, M'$$

to be inferred. The relation is read "expression E computes value v and memory M' under environment σ and memory M."

Definition 1 (Program's Semantics)**.** *A program E's semantics is defined to be the inference tree of relation $\emptyset, \emptyset \vdash E \Downarrow v, M$ for some v and M. If there is no such v and M, then the expression has no meaning.*

$$\frac{}{\sigma, M \vdash \texttt{n} \Downarrow n, M} \qquad \frac{}{\sigma, M \vdash \texttt{unit} \Downarrow \cdot, M}$$

$$\frac{}{\sigma, M \vdash \texttt{true} \Downarrow T, M} \qquad \frac{}{\sigma, M \vdash \texttt{false} \Downarrow F, M}$$

$$\frac{M(\sigma(x)) = v}{\sigma, M \vdash x \Downarrow v, M}$$

$$\frac{\begin{array}{c}\sigma(f) = \langle x, E_1, \sigma_1 \rangle \qquad \sigma, M \vdash E \Downarrow v_1, M_1 \\ l \notin M_1 \qquad \sigma_1[l/x][\langle x, E_1, \sigma_1 \rangle / f], M_1[v_1/l] \vdash E_1 \Downarrow v_2, M_2\end{array}}{\sigma, M \vdash \texttt{call } f\texttt{(}E\texttt{)} \Downarrow v_2, M_2}$$

$$\frac{\sigma(f) = \langle y, E_1, \sigma_1 \rangle \qquad \sigma_1[\sigma(x)/y][\langle y, E_1, \sigma_1 \rangle / f], M \vdash E_1 \Downarrow v_1, M_1}{\sigma, M \vdash \texttt{call } f\texttt{<}x\texttt{>} \Downarrow v_1, M_1}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v_2, M_2 \qquad plus(v_1, v_2) = v}{\sigma, M \vdash E_1 \texttt{ + } E_2 \Downarrow v, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v_2, M_2 \qquad minus(v_1, v_2) = v}{\sigma, M \vdash E_1 \texttt{ - } E_2 \Downarrow v, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow n_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow n_2, M_2}{\sigma, M \vdash E_1 \texttt{ * } E_2 \Downarrow n_1 \times n_2, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow n_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow n_2, M_2}{\sigma, M \vdash E_1 \texttt{ / } E_2 \Downarrow n_1 \ div \ n_2, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v_2, M_2 \qquad equal(v_1, v_2) = tf}{\sigma, M \vdash E_1 \texttt{=} E_2 \Downarrow tf, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v_2, M_2 \qquad less(v_1, v_2) = tf}{\sigma, M \vdash E_1 \texttt{<} E_2 \Downarrow tf, M_2}$$

$$\frac{\sigma, M \vdash E \Downarrow tf, M_1}{\sigma, M \vdash \texttt{not } E \Downarrow not \ tf, M_1}$$

$$\frac{\sigma(x) = l \qquad \sigma, M \vdash E \Downarrow v, M_1}{\sigma, M \vdash x \texttt{ := } E \Downarrow \cdot, M_1[v/l]}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v_2, M_2}{\sigma, M \vdash E_1 \texttt{ ; } E_2 \Downarrow v_2, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow T, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v, M_2}{\sigma, M \vdash \texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3 \texttt{ end} \Downarrow \cdot, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow F, M_1 \qquad \sigma, M_1 \vdash E_3 \Downarrow v, M_2}{\sigma, M \vdash \texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3 \texttt{ end} \Downarrow \cdot, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow T, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v, M_2}{\sigma, M \vdash \texttt{if } E_1 \texttt{ then } E_2 \texttt{ end} \Downarrow \cdot, M_2}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow F, M_1}{\sigma, M \vdash \texttt{if } E_1 \texttt{ then } E_2 \texttt{ end} \Downarrow \cdot, M_1}$$

$$\frac{\begin{array}{c}\sigma, M \vdash E_1 \Downarrow T, M_1 \\ \sigma, M_1 \vdash E_2 \Downarrow v, M_2 \qquad \sigma, M_2 \vdash \texttt{while } E_1 \texttt{ do } E_2 \texttt{ end} \Downarrow \cdot, M_3\end{array}}{\sigma, M \vdash \texttt{while } E_1 \texttt{ do } E_2 \texttt{ end} \Downarrow \cdot, M_3}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow F, M_1}{\sigma, M \vdash \texttt{while } E_1 \texttt{ do } E_2 \texttt{ end} \Downarrow \cdot, M_1}$$

$$\frac{\begin{array}{c}\sigma(x) = l \qquad \sigma, M \vdash E_1 \Downarrow n_1, M_1 \\ \sigma, M_1 \vdash E_2 \Downarrow n_2, M_2 \qquad n_1 \leq n_2 \qquad \sigma, M_2[n_1/l] \vdash E_3 \Downarrow v, M_3 \\ \sigma, M_3 \vdash \texttt{for } x \texttt{ := } n_1 + 1 \texttt{ to } E_2 \texttt{ do } E_3 \texttt{ end} \Downarrow \cdot, M_4\end{array}}{\sigma, M \vdash \texttt{for } x \texttt{ := } E_1 \texttt{ to } E_2 \texttt{ do } E_3 \texttt{ end} \Downarrow \cdot, M_4}$$

$$\frac{\sigma(x) = l \qquad \sigma, M \vdash E_1 \Downarrow n_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow n_2, M_2 \qquad n_1 > n_2}{\sigma, M \vdash \texttt{for } x \texttt{ := } E_1 \texttt{ to } E_2 \texttt{ do } E_3 \texttt{ end} \Downarrow \cdot, M_2[n_1/l]}$$

$$\frac{\sigma(x) = l \qquad \mathit{read}\ n}{\sigma, M \vdash \texttt{read } x \Downarrow \cdot, M[n/l]} \qquad\qquad \frac{}{\mathit{read}\ n}$$

$$\frac{\sigma, M \vdash E \Downarrow n, M_1 \qquad \mathit{write}\ n}{\sigma, M \vdash \texttt{write } E \Downarrow \cdot, M_1} \qquad\qquad \frac{}{\mathit{write}\ n}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad l \notin \mathit{dom}(M_1) \qquad \sigma[l/x], M_1[v_1/l] \vdash E_2 \Downarrow v_2, M_2}{\sigma, M \vdash \texttt{let } x \texttt{ := } E_1 \texttt{ in } E_2 \texttt{ end} \Downarrow v_2, M_2}$$

$$\frac{\sigma[\langle x, E_1, \sigma\rangle/f], M \vdash E_2 \Downarrow v, M_1}{\sigma, M \vdash \texttt{let procedure } f(x) \texttt{ = } E_1 \texttt{ in } E_2 \texttt{ end} \Downarrow v, M_1}$$

$$\frac{\begin{array}{c}x_1 \neq x_2 \\ \sigma, M \vdash E_1 \Downarrow v_1, M_1 \qquad \sigma, M_1 \vdash E_2 \Downarrow v_2, M_2 \qquad l_1, l_2 \notin \mathit{dom}(M_2) \qquad l_1 \neq l_2\end{array}}{\sigma, M \vdash \{x_1 \texttt{:=} E_1 \texttt{, } x_2 \texttt{ := } E_2\} \Downarrow \{x_1 \mapsto l_1, x_2 \mapsto l_2\}, M_2[v_1/l_1][v_2/l_2]}$$

$$\frac{\sigma, M \vdash E \Downarrow r, M_1}{\sigma, M \vdash E.x \Downarrow M_1(r(x)), M_1}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow r, M_1 \qquad r(x) = l \qquad \sigma, M_1 \vdash E_2 \Downarrow v, M_2}{\sigma, M \vdash E_1.x \texttt{ := } E_2 \Downarrow \cdot, M_2[v/l]}$$

$$\frac{\sigma, M \vdash E \Downarrow n, M_1 \qquad \langle b,0\rangle, \cdots, \langle b, n-1\rangle \notin dom(M_1)}{\sigma, M \vdash \texttt{malloc(}E\texttt{)} \Downarrow \langle b,0\rangle, M_1[\bot/\langle b,0\rangle, \cdots, \bot/\langle b,n-1\rangle]}$$

$$\frac{}{\sigma, M \vdash \texttt{\&}x \Downarrow \sigma(x), M} \qquad \frac{\sigma, M \vdash E \Downarrow l, M_1}{\sigma, M \vdash \texttt{*}E \Downarrow M_1(l), M_1}$$

$$\frac{\sigma, M \vdash E_1 \Downarrow l, M_1 \qquad l \in dom(M_1) \qquad \sigma, M_1 \vdash E_2 \Downarrow v, M_2}{\sigma, M \vdash E_1 \texttt{ := } E_2 \Downarrow v, M_2[v/l]} \begin{array}{l} E_1 \neq x \\ E_1 \neq E\texttt{.}x \end{array}$$

$$\begin{array}{lclclcl} plus(n_1, n_2) & \doteq^{2} & n_1 + n_2 & \quad & minus(n_1, n_2) & \doteq & n_1 - n_2 \\ plus(\langle b, o\rangle, n) & \doteq & \langle b, o+n\rangle & & minus(\langle b, o\rangle, n) & \doteq & \langle b, o-n\rangle \\ plus(n, \langle b, o\rangle) & \doteq & \langle b, o+n\rangle & & minus(\langle b, o_1\rangle, \langle b, o_2\rangle) & \doteq & o_1 - o_2 \end{array}$$

$$\begin{array}{lcl} less(n_1, n_2) & \doteq & n_1 < n_2 \\ less(\langle b, o_1\rangle, \langle b, o_2\rangle) & \doteq & o_1 < o_2 \\ less(\cdot, \cdot) & \doteq & F \end{array}$$

$$\begin{array}{lcl} equal(n_1, n_2) & \doteq & n_1 = n_2 \\ equal(tf_1, tf_2) & \doteq & tf_1 = tf_2 \\ equal(\langle b_1, o_1\rangle, \langle b_2, o_2\rangle) & \doteq & (b_1 = b_2) \wedge (o_1 = o_2) \\ equal(r_1, r_2) & \doteq & dom(r_1) = dom(r_2) \;\wedge\; \forall x \in dom(r_1).equal(r_1(x), r_2(x)) = T \\ equal(\cdot, \cdot) & \doteq & T \end{array}$$

[2] means “defined by”