4190.310 Programming Language
The K-- Language

1 Syntax

$$
\begin{array}{lcll}
\textit{Expression}\ e & \rightarrow & \texttt{unit} & \text{unit} \\
 & \mid & x\ \texttt{:=}\ e & \text{assignment} \\
 & \mid & e\ \texttt{;}\ e & \text{sequence} \\
 & \mid & \texttt{if}\ e\ \texttt{then}\ e\ \texttt{else}\ e & \text{branch} \\
 & \mid & \texttt{while}\ e\ \texttt{do}\ e & \text{while loop} \\
 & \mid & \texttt{read}\ x & \text{input} \\
 & \mid & \texttt{write}\ e & \text{output} \\
 & \mid & \texttt{let}\ x\ \texttt{:=}\ e\ \texttt{in}\ e & \text{variable binding} \\
 & \mid & n & \text{integer} \\
 & \mid & \texttt{true} \mid \texttt{false} & \text{boolean} \\
 & \mid & x & \text{identifier} \\
 & \mid & e\ \texttt{+}\ e \mid e\ \texttt{-}\ e \mid e\ \texttt{*}\ e \mid e\ \texttt{/}\ e & \text{arithmetic operation} \\
 & \mid & e\ \texttt{<}\ e \mid e\ \texttt{=}\ e \mid \texttt{not}\ e & \text{conditional operation}
\end{array}
$$

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 while do 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

$$\begin{array}{llrcl}
n & \in & \mathbb{Z} & & \text{integer} \\
b & \in & \mathbb{B} & & \text{boolean} \\
v & \in & \mathit{Val} & = & \mathbb{Z} + \mathbb{B} + \{\cdot\} \\
\sigma & \in & \mathit{Env} & = & \mathit{Id} \xrightarrow{\text{fin}} \mathit{Addr} \\
M & \in & \mathit{Mem} & = & \mathit{Addr} \xrightarrow{\text{fin}} \mathit{Val} \\
x, y & \in & \mathit{Id} & & \text{identifier} \\
l & \in & \mathit{Addr} & & \text{address}
\end{array}$$

3 Semantics

[True]

$$\frac{}{\sigma, M \vdash \texttt{true} \Rightarrow \textit{true}, M}$$

[False]

$$\frac{}{\sigma, M \vdash \texttt{false} \Rightarrow \textit{false}, M}$$

[Num]

$$\frac{}{\sigma, M \vdash n \Rightarrow n, M}$$

[Unit]

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

[Var]

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

[Add]

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

[Sub]

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

[Mul]

$$\frac{\sigma, M \vdash e_1 \Rightarrow n_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow n_2, M''}{\sigma, M \vdash e_1) \ \times \ e_2 \Rightarrow n_1 \times n_2, M''}$$

[Div]

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

[EqualTrue]

$$\frac{\sigma, M \vdash e_1 \Rightarrow v_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v_2, M'' \qquad v_1 \ = \ v_2}{\sigma, M \vdash e_1 \ \texttt{=} \ e_2 \Rightarrow \textit{true}, M''}$$

[EqualFalse]

$$\frac{\sigma, M \vdash e_1 \Rightarrow v_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v_2, M'' \qquad v_1 \ \neq \ v_2}{\sigma, M \vdash e_1 \ \texttt{=} \ e_2 \Rightarrow \textit{false}, M''}$$

[Less]

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

[Not]

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

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

[Seq]
$$\frac{\sigma, M \vdash e_1 \Rightarrow v_1, M' \qquad \sigma, M' \vdash e_2 \Rightarrow v_2, M''}{\sigma, M \vdash e_1\ \ \texttt{;}\ \ e_2 \Rightarrow v_2, M''}$$

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

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

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

[WhileFalse]
$$\frac{\sigma, M \vdash e_1 \Rightarrow \mathit{false}, M'}{\sigma, M \vdash \texttt{while}\ \ e_1\ \ \texttt{do}\ \ e_2 \Rightarrow \cdot, M'}$$

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

[Read]
$$\frac{}{\sigma, M \vdash \texttt{read}\ \ x \Rightarrow n, M\{\sigma(x) \mapsto n\}}$$

[Write]
$$\frac{\sigma, M \vdash e \Rightarrow n, M'}{\sigma, M \vdash \texttt{write}\ \ e \Rightarrow n, M'}$$