DENOTATIONAL DEFINITION OF MINI-LANGUAGE CORE SUBSET †

Concrete Syntax of Core Subset

<statement sequence> ::= <statement> **;** {<statement> **;**}

<statement> ::= <assignment statement> | <if statement> | <loop statement>
| <input statement> | <output statement>

<assignment statement> ::= <identifier> **:=** <expression>

<if statement> ::= **if** <comparison> **then** <statement sequence> [**else** <statement>] **end if**

<while statement> ::= **while** <comparison> **loop** <statement sequence> **end loop**

<input statement> ::= **input** <identifier>

<output statement> ::= **output** <identifier>

<comparison> ::= **(** <factor> <comparison operator> <factor> **)**

<expression> ::= <term> {<add operator> <term>}

<term> ::= <factor> {***** <factor>}

<factor> ::= **(** <expression> **)** | <identifier> | <integer>

<comparison operator> ::= < | ≤ | = | ≠ | > | ≥

<add operator> ::= + | –

Syntax Domains

P : Prog	programs	**S** : Stmt	statements
E : Exp	expressions	**C** : Comp	comparisons
R : Rel	relations	**O** : Opr	operations
I : Num	integer constants	**V** : Var	variables

Abstract Syntax Rules

P ::= **S**

S ::= $\mathbf{S}_1$; $\mathbf{S}_2$ | **V := E** | **while C loop S end loop** | **if C then S end if** | **if C then** $\mathbf{S}_1$ **else** $\mathbf{S}_2$ **end if**
| **input V** | **output V**

C ::= $\mathbf{E}_1$ **R** $\mathbf{E}_2$

E ::= $\mathbf{E}_1$ **O** $\mathbf{E}_2$ | **V** | **I**

R ::= < | ≤ | = | ≠ | > | ≥

O ::= + | – | *

Semantic Domains

conf	: Cf = S × Fi × Fi	*file*	: Fi = $\mathbf{N}^*$
store	: S = Var → N	*t*	: T = {*true*, *false*}$^\circ$
n	: N = {..., –2, –1, 0, 1, 2, ...}$^\circ$		

---

† M. Marcotty and H. F. Ledgard, *Programming Language Landscape: Syntax, Semantics, and Implementation*, 2nd ed., SRA, 1986.

Semantic Functions

M : Prog $\rightarrow$ Fi $\rightarrow$ Fi
C : Comp $\rightarrow$ S $\rightarrow$ T
N : Num $\rightarrow$ N
S : Stmt $\rightarrow$ Cf $\rightarrow$ Cf
E : Exp $\rightarrow$ S $\rightarrow$ N

Semantic Equations

$M[[\mathbf{P}]]file_1 = (S[[\mathbf{P}]](<\lambda \mathbf{V}.\perp, file_1, nil>)) \downarrow 3$

$S[[\mathbf{S}_1 ; \mathbf{S}_2]]conf = S[[\mathbf{S}_2]](S[[\mathbf{S}_1]]conf)$

$S[[\mathbf{V} := \mathbf{E}]]conf = <store[E[[\mathbf{E}]]store/\mathbf{V}], file_1, file_2>$
where $store = conf \downarrow 1$, $file_1 = conf \downarrow 2$, $file_2 = conf \downarrow 3$

$S[[$**while C loop S end loop**$]]conf =$
if $C[[\mathbf{C}]](conf \downarrow 1)$ then
$S[[$**while C loop S end loop**$]](S[[\mathbf{S}]]conf)$
else
$conf$

$S[[$**if C then S end if**$]]conf =$ if $C[[\mathbf{C}]](conf \downarrow 1)$ then $S[[\mathbf{S}]]conf$ else $conf$

$S[[$**if C then S**$_1$ **else S**$_2$ **end if**$]]conf =$ if $C[[\mathbf{C}]](conf \downarrow 1)$ then $S[[\mathbf{S}_1]]conf$ else $S[[\mathbf{S}_2]]conf$

$S[[$**input V**$]]conf =$ if $file_1 = nil$ then $\top$ else $<store[hd(file_1)/\mathbf{V}], tl(file_1), file_2>$
where $store = conf \downarrow 1$, $file_1 = conf \downarrow 2$, $file_2 = conf \downarrow 3$

$S[[$**output V**$]]conf =$ if $store[[\mathbf{V}]] = \perp$ then $\top$ else $<store, file_1, append(file_2, list(store[[\mathbf{V}]]))>$
where $store = conf \downarrow 1$, $file_1 = conf \downarrow 2$, $file_2 = conf \downarrow 3$

$C[[\mathbf{E}_1 < \mathbf{E}_2]]store =$ if $E[[\mathbf{E}_1]]store < E[[\mathbf{E}_2]]store$ then *true* else *false*

$C[[\mathbf{E}_1 \leq \mathbf{E}_2]]store =$ if $E[[\mathbf{E}_1]]store \leq E[[\mathbf{E}_2]]store$ then *true* else *false*

$C[[\mathbf{E}_1 = \mathbf{E}_2]]store =$ if $E[[\mathbf{E}_1]]store = E[[\mathbf{E}_2]]store$ then *true* else *false*

$C[[\mathbf{E}_1 \neq \mathbf{E}_2]]store =$ if $E[[\mathbf{E}_1]]store \neq E[[\mathbf{E}_2]]store$ then *true* else *false*

$C[[\mathbf{E}_1 > \mathbf{E}_2]]store =$ if $E[[\mathbf{E}_1]]store > E[[\mathbf{E}_2]]store$ then *true* else *false*

$C[[\mathbf{E}_1 \geq \mathbf{E}_2]]store =$ if $E[[\mathbf{E}_1]]store \geq E[[\mathbf{E}_2]]store$ then *true* else *false*

$E[[\mathbf{E}_1 + \mathbf{E}_2]]store = E[[\mathbf{E}_1]]store + E[[\mathbf{E}_2]]store$

$E[[\mathbf{E}_1 - \mathbf{E}_2]]store = E[[\mathbf{E}_1]]store - E[[\mathbf{E}_2]]store$

$E[[\mathbf{E}_1 * \mathbf{E}_2]]store = E[[\mathbf{E}_1]]store \times E[[\mathbf{E}_2]]store$

$E[[\mathbf{I}]]store = N[[\mathbf{I}]]$

$E[[\mathbf{V}]]store =$ if $store[[\mathbf{V}]] = \perp$ then $\top$ else $store[[\mathbf{V}]]$