Syntax and Semantics of V^-

Roger Burtonpatel

October 22, 2023

1 Syntax

We present a grammar of V^-:

Programs	P	::=	$\{d\}$	definition
Definitions	d	::=	`val` x `=` e	bind name to expression
Expressions	e	::=	x	name
		|	`if` g_α $\{[]g_\alpha\}$ `fi`	if-fi
		|	$K\{e\}$	value constructor application
		|	e_1 e_2	function application
Guarded Expressions	g_α	::=	$\rightarrow \alpha$	terminating α
		|	$e;\ g_\alpha$	intermediate expression
		|	$\mathtt{E}\{x\}\,.\,g_\alpha$	existential
		|	$e_1 = e_2;\ g_\alpha$	equation
Value Constructors	K	::=	`::`	cons
		|	`[]`	empty list
		|	`#`x	name beginning with `#`
		|	`A-Z`x	name beggining with capital letter
		|	[`-`|`+`](`0` $-$ `9`)+	signed integer literal

A *name* is any token that is not an integer literal, does not contain whitespace, a bracket, or parenthesis, and is not a value constructor name or a reserved word.

rab: Would like help cleaning up the format on this, specifically with regards to the regex. The one downside of this nicer package is that descriptions will not wrap, so describing an integer literal in english isn't an option as far as I can tell.

2 Refinement ordering on environments

$$\rho \subseteq \rho' \text{ when } \operatorname{dom}\rho \subseteq \operatorname{dom}\rho' \text{ and } \forall x \in \operatorname{dom}\rho : \rho(x) \subseteq \rho'(x)$$

3 Forms of Judgement for V^-:

Metavariables	
ϑ	a value produced from evaluating α.
eq	equation
reject	equation rejection
r	ϑ | **reject**: a result of ϑor rejection
ρ	environment: $name \to \mathcal{V}_\perp$
$\rho\{x \mapsto y\}$	environment extended with name x mapping to y
$\mathcal{T}$	Context of all temporarily stuck equations (a sequence)
e	An expression
g	A guarded expression

Sequences	
ε	the empty sequence
$S_1 \cdot S_2$	Concatenate sequence S_1 and sequence S_2
$x \cdot S_2$	Cons x onto sequence S_2

Expressions

An expression in core Verse evaluates to produce possibly-empty sequence of values. In V^-, values depend on α. If α is a Verse-like expression, ϑwill be a value sequence. If it is an ML-like expression, it will be a single value.
A guarded expression evaluates to produce a **result**. A result is either a possibly-empty sequence of values or reject.

$$r ::= \vartheta \mid \textbf{reject}$$

$$\rho;\ \mathcal{T} \vdash \alpha \Downarrow \vartheta \quad \text{(Eval-Expr)}$$
$$\rho;\ \mathcal{T} \vdash g \Downarrow r \quad \text{(Eval-Guarded-Expr)}$$

$$\rho;\ \mathcal{T} \vdash g \rightsquigarrow \ \lightning \quad \text{(NoTree)}$$

4 Sequences

The trivial sequence is ε. Sequences can be concatenated with infix $\cdot$. In an appropriate context, a value like x stands for the singleton sequence containing x.

$$\varepsilon \cdot ys \equiv ys$$
$$ys \cdot \varepsilon \equiv ys$$
$$(xs \cdot ys) \cdot zs \equiv xs \cdot (ys \cdot zs)$$

5 Rules (Big-step Operational Semantics) for V^-:

Evaluating Guarded Expressions

$$\text{(Eval-ArrowExpr)}\ \frac{\rho;\ \varepsilon \vdash e \Downarrow \vartheta}{\rho;\ \varepsilon \vdash \rightarrow e \Downarrow \vartheta}$$

$$\text{(Eval-Exists)}\ \frac{\rho\{x \mapsto \bot\};\ \mathcal{T} \vdash g \Downarrow r}{\rho;\ \mathcal{T} \vdash \exists x.\ g \Downarrow r}$$

$$\text{(G-Move-To-Ctx)}\ \frac{\rho;\ eq \cdot \mathcal{T} \vdash g \Downarrow r}{\rho;\ \mathcal{T} \vdash eq;\ g \Downarrow r}$$

$$\text{(Eval-Move-Eqn)}\ \frac{\rho;\ \mathcal{T} \vdash e \Downarrow \vartheta \qquad \rho\{x \mapsto \vartheta\};\ \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r'}{\rho;\ \mathcal{T} \cdot x = e \cdot \mathcal{T}' \vdash g \Downarrow r'}$$

$$\text{(G-Vcon-Single-Fail)}\ \frac{K \neq K'}{\rho;\ \mathcal{T} \cdot K = K' \cdot \mathcal{T}' \vdash g \Downarrow \textbf{reject}}$$

$$\text{(G-Vcon-Single-Succ)}\ \frac{\rho;\ \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r}{\rho;\ \mathcal{T} \cdot K = K \cdot \mathcal{T}' \vdash g \Downarrow r}$$

$$\text{(G-Vcon-Multi-Fail)}\ \frac{K \neq K'}{\rho;\ \mathcal{T} \cdot K(e_1, \ldots e_n) = K'(e'_1, \ldots e'_n) \cdot \mathcal{T}' \vdash g \Downarrow \textbf{reject}}$$

$$\text{(G-Vcon-Multi-Succ)}\ \frac{\rho;\ [e_i = e'_i \mid 1 \leq i \leq n] \cdot \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r}{\rho;\ \mathcal{T} \cdot K(e_1, \ldots e_n) = K(e'_1, \ldots e'_n) \cdot \mathcal{T}' \vdash g \Downarrow r}$$

Evaluating General Expressions

$$\text{(If-Fi-Success)}\quad \frac{\rho;\ \mathcal{T} \vdash g \Downarrow \vartheta}{\rho;\ \mathcal{T} \vdash \textsc{if}\ g \ \square\ \ldots\ \textsc{fi} \Downarrow \vartheta}$$

$$\text{(If-Fi-Reject)}\quad \frac{\rho;\ \mathcal{T} \vdash g \Downarrow \textbf{reject} \qquad \rho;\ \mathcal{T} \vdash \textsc{if}\ \ldots\ \textsc{fi} \Downarrow \vartheta}{\rho;\ \mathcal{T} \vdash \textsc{if}\ g \ \square\ \ldots\ \textsc{fi} \Downarrow \vartheta}$$

$$\text{(Vcon-Empty)}\quad \frac{}{\rho;\ \mathcal{T} \vdash K \Downarrow K}$$

$$\text{(Vcon-Multi)}\quad \frac{\rho;\ \mathcal{T} \vdash e_i \Downarrow \vartheta_i \qquad 1 \leq i \leq n}{\rho;\ \mathcal{T} \vdash K(e_1, \ldots e_n) \Downarrow K(\vartheta_1, \ldots \vartheta_i)}$$