

# Syntax and Semantics of $V^-$

Roger Burtonpatel

October 22, 2023

## 1 Syntax

We present a grammar of  $V^-$ :

Programs	$P$	$::=$	$\{d\}$	definition
Definitions	$d$	$::=$	$\text{val } x = e_\alpha$	bind name to expression
Expressions	$e_\alpha$	$::=$	$x, y, z$	names
			$\text{if } [g_\alpha \{[]g_\alpha\}] \text{ fi}$	if-fi
			$K\{e_\alpha\}$	value constructor application
			$e_{\alpha_1} e_{\alpha_2}$	function application
Guarded Expressions	$g_\alpha$	$::=$	$\rightarrow \alpha$	terminating $\alpha$
			$e_\alpha; g_\alpha$	intermediate expression
			$\exists\{x\}.g_\alpha$	existential
			$e_{\alpha_1} = e_{\alpha_2}; g_\alpha$	equation
Value Constructors	$K$	$::=$	$::$	cons
			$[]$	empty list
			$\#x$	name beginning with #
			$\text{A-Z}x$	name beginning 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.

## 2 Refinement ordering on environments

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

## 3 Forms of Judgement for $V^-$ :

<i>Metavariables</i>	
$\vartheta$	a value produced from evaluating $\alpha$ .
$eq$	equation
<b>reject</b>	equation rejection
$r$	$\vartheta \mid \text{reject}$ : a result of $\vartheta$ or rejection
$\rho$	environment: $\text{name} \rightarrow \mathcal{V}_\perp$
$\rho\{x \mapsto y\}$	environment extended with name $x$ mapping to $y$
$\mathcal{T}$	Context of all temporarily stuck equations (a sequence)
$e_\alpha$	An expression
$g$	A guarded expression
$\Leftarrow$	Inability to compile to a decision tree; a compile time error
<i>Sequences</i>	
$\varepsilon$	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  $\alpha$ . If  $\alpha$  is a Verse-like expression,  $\vartheta$  will 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.

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

$$\rho; \mathcal{T} \vdash \alpha \Downarrow \vartheta \quad (\text{EVAL-EXPR})$$

$$\rho; \mathcal{T} \vdash g \Downarrow r \quad (\text{EVAL-GUARDED-EXPR})$$

If a guarded expression cannot be evaluated without producing logical variables at runtime, it cannot be expressed as a decision tree. This notation indicates this failure (think of  $\Leftarrow$  as a fallen tree), which results in a compile-time error.

$$\rho; \mathcal{T} \vdash g \rightsquigarrow \Leftarrow \quad (\text{NOTREE})$$

## 4 Sequences

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

$$\begin{aligned} \varepsilon \cdot ys &\equiv ys \\ ys \cdot \varepsilon &\equiv ys \\ (xs \cdot ys) \cdot zs &\equiv xs \cdot (ys \cdot zs) \end{aligned}$$

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

### Evaluating Guarded Expressions

#### Evaluating simple parts of guarded expressions

$$(\text{EVAL-ARROWEXPR}) \quad \frac{\rho; \varepsilon \vdash e \Downarrow \vartheta}{\rho; \varepsilon \vdash \rightarrow e \Downarrow \vartheta}$$

$$(\text{EVAL-EXISTS}) \quad \frac{\rho\{x \mapsto \perp\}; \mathcal{T} \vdash g \Downarrow r}{\rho; \mathcal{T} \vdash \exists x. g \Downarrow r}$$

$$(\text{EVAL-EXPSEQ}) \quad \frac{\rho; \mathcal{T} \vdash e_\alpha \Downarrow \vartheta \quad \rho; \mathcal{T} \vdash g \Downarrow r}{\rho; \mathcal{T} \vdash e_\alpha; g \Downarrow r}$$

#### Shifting an equation to the context

$$(\text{G-MOVE-TO-CTX}) \quad \frac{\rho; eq \cdot \mathcal{T} \vdash g \Downarrow r}{\rho; \mathcal{T} \vdash eq; g \Downarrow r}$$

#### Evaluating with different types of equations

$$(\text{G-EQEXPS}) \quad \frac{\begin{array}{c} x, y \text{ are distinct and fresh} \\ \rho\{x \mapsto \perp, y \mapsto \perp\}; x = e_{\alpha_1} \cdot y = e_{\alpha_2} \cdot x = y \cdot \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r \end{array}}{\rho; \mathcal{T} \cdot e_{\alpha_1} = e_{\alpha_2} \cdot \mathcal{T}' \vdash g \Downarrow r}$$

$$(\text{G-EQNAMEEXP}) \quad \frac{\rho; \mathcal{T} \vdash e_\alpha \Downarrow \vartheta \quad \rho\{x \mapsto \vartheta\}; \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r'}{\rho; \mathcal{T} \cdot x = e_\alpha \cdot \mathcal{T}' \vdash g \Downarrow r'}$$

$$(G\text{-EQNAMES-VALS-SUCC}) \frac{x, y \in \text{dom } \rho \quad \rho(x) = \vartheta, \rho(y) = \vartheta \quad \rho; \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r}{\rho; \mathcal{T} \cdot x = y \cdot \mathcal{T}' \vdash g \Downarrow r}$$

$$(G\text{-EQNAMES-VALS-FAIL}) \frac{x, y \in \text{dom } \rho \quad \rho(x) = \vartheta, \rho(y) = \vartheta' \quad \vartheta \neq \vartheta'}{\rho; \mathcal{T} \cdot x = y \cdot \mathcal{T}' \vdash g \Downarrow \mathbf{reject}}$$

$$(G\text{-EQNAMES-BOTS-FAIL}) \frac{x, y \in \text{dom } \rho \quad \rho(x) = \perp, \rho(y) = \perp \quad x, y \text{ do not appear in } \mathcal{T}, \mathcal{T}'}{\rho; \mathcal{T} \cdot x = y \cdot \mathcal{T}' \vdash g \rightsquigarrow \notin}$$

$$(G\text{-EQNAMES-BOTVAL-SUCC}) \frac{x, y \in \text{dom } \rho \quad \rho(x) = \perp, \rho(y) = \vartheta \quad \rho\{x \mapsto \vartheta\}; \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r'}{\rho; \mathcal{T} \cdot x = y \cdot \mathcal{T}' \vdash g \Downarrow r'}$$

$$(G\text{-VCON-SINGLE-FAIL}) \frac{K \neq K'}{\rho; \mathcal{T} \cdot K = K' \cdot \mathcal{T}' \vdash g \Downarrow \mathbf{reject}}$$

$$(G\text{-VCON-SINGLE-SUCC}) \frac{\rho; \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r}{\rho; \mathcal{T} \cdot K = K \cdot \mathcal{T}' \vdash g \Downarrow r}$$

$$(G\text{-VCON-MULTI-FAIL}) \frac{K \neq K'}{\rho; \mathcal{T} \cdot K(e_{\alpha_1}, \dots e_{\alpha_n}) = K'(e'_{\alpha_1}, \dots e'_{\alpha_n}) \cdot \mathcal{T}' \vdash g \Downarrow \mathbf{reject}}$$

$$(G\text{-VCON-MULTI-ARITY-FAIL}) \frac{n \neq m}{\rho; \mathcal{T} \cdot K(e_{\alpha_1}, \dots e_{\alpha_n}) = K(e'_{\alpha_1}, \dots e'_{\alpha_m}) \cdot \mathcal{T}' \vdash g \Downarrow \mathbf{reject}}$$

$$(G\text{-VCON-MULTI-SUCC}) \frac{\rho; [e_{\alpha_i} = e'_{\alpha_i} \mid 1 \leq i \leq n] \cdot \mathcal{T} \cdot \mathcal{T}' \vdash g \Downarrow r}{\rho; \mathcal{T} \cdot K(e_{\alpha_1}, \dots e_{\alpha_n}) = K(e'_{\alpha_1}, \dots e'_{\alpha_n}) \cdot \mathcal{T}' \vdash g \Downarrow r}$$

## Evaluating General Expressions

$$(IF\text{-FI-SUCCESS}) \frac{\rho; \mathcal{T} \vdash g \Downarrow \vartheta}{\rho; \mathcal{T} \vdash IF [g \square \dots] FI \Downarrow \vartheta}$$

$$(IF\text{-FI-REJECT}) \frac{\rho; \mathcal{T} \vdash g \Downarrow \mathbf{reject} \quad \rho; \mathcal{T} \vdash IF [\dots] FI \Downarrow \vartheta}{\rho; \mathcal{T} \vdash IF [g \square \dots] FI \Downarrow \vartheta}$$

$$(VCON\text{-EMPTY}) \frac{}{\rho; \mathcal{T} \vdash K \Downarrow K}$$

$$(VCON\text{-MULTI}) \frac{\rho; \mathcal{T} \vdash e_{\alpha_i} \Downarrow \vartheta_i \quad 1 \leq i \leq n}{\rho; \mathcal{T} \vdash K(e_{\alpha_1}, \dots e_{\alpha_n}) \Downarrow K(\vartheta_1, \dots \vartheta_i)}$$

## 6 The very suspect rule from question 5

$$(EQ\text{NAMES-BOTS-SUCC}) \frac{x, y \in \text{dom } \rho \quad \rho(x) = \perp, \rho(y) = \perp \quad \text{Either } x \text{ or } y \text{ appears in } \mathcal{T}, \mathcal{T}' \quad \rho; \mathcal{T} \cdot x = y \cdot \mathcal{T}' \vdash g \Downarrow r}{\rho; \mathcal{T} \cdot x = y \cdot \mathcal{T}' \vdash g \Downarrow r}$$