# Diderot Formal Semantics

John Reppy
University of Chicago
jhr@cs.uchicago.edu

Lamont Samuels
University of Chicago
lamonts@cs.uchicago.edu

March 22, 2011

## 1 Abstract Syntax

- Definitions: variable := string | id := integer
- Values: v := Num(n) | Bool(b) | String(s) | tensor...?? | vecs...??
- Arithmetic expression (Aexp):

| | | | |
|---|---|---|---|
| e | ::= | $n$ | for n $\in \mathbb{R}$ |
| | \| | x | for x = variable |
| | \| | $e_1$ + $e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1$ - $e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1$ * $e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1$ / $e_2$ | for $e_1$, $e_2$ $\in$ Aexp |

- Relational expression (Rexp):

| | | | |
|---|---|---|---|
| r | ::= | true | |
| | \| | false | |
| | \| | $e_1 == e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1 < e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1 > e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1 \leq e_2$ | for $e_1$, $e_2$ $\in$ Aexp |
| | \| | $e_1 \geq e_2$ | for $e_1$, $e_2$ $\in$ Aexp |

- Commands (Comms):

| | | | |
|---|---|---|---|
| c | ::= | x := e | for x = variable, e $\in$ Aexp |
| | \| | $c_1$; $c_2$ | for $c_1$, $c_2$ $\in$ Comms |
| | \| | $if\ r\ then\ c_1\ else\ c_2$ | for $c_1$, $c_2$ $\in$ Aexp and b $\in$ Rexp |
| | \| | stabalize | |

## 2 Strand Environment

strand_status := Active * env | Stabilized * env | Dead * env
strand_env := strand_status * id * comms
env := variable $\rightarrow$ value

## 3 Global Environment

The $\cup$ character denotes a parallel construct that states that every stand environment runs in parallel with each other.

$$\bigcup_{\substack{id \in strand\_env \\ \Gamma = id \to strand\_env}} update(\Gamma(id), \Gamma)$$

dom($\Gamma$) := all active & stabilized strand environments
newId := $\Gamma$ * id $\to$ id

## 4 Inference Rules

• Update function: strand_env * $\Gamma \to \Gamma'$

$$\frac{}{update((Stabalized, env), id, c), \Gamma) \Downarrow \Gamma}$$

$$\frac{comm\_eval((Active, env), c, id, \Gamma) \Downarrow (new\_status, env', \Gamma')}{update((Active, env), id, c), \Gamma) \Downarrow \Gamma'[(new\_status, env')/id]}$$

• comm_eval: strand_status * commands * id * $\Gamma \to$ strand_status * env * $\Gamma'$

$$\frac{}{comm\_eval((s, env), stabalize, id, \Gamma) \Downarrow (Stablize, env, \Gamma)}$$

$$\frac{}{comm\_eval((s, env), die, id, \Gamma) \Downarrow (Dead, env, \Gamma)}$$

$$\frac{expr\_eval(e, env) \Downarrow v}{comm\_eval((s, env), x := e, id, \Gamma) \Downarrow (s, env[x := v], \Gamma)}$$

$$\frac{comm\_eval((s, env), c_1, \Gamma) \Downarrow (s', env', \Gamma') \quad comm\_eval((s', env), c_2, \Gamma') \Downarrow (s'', env'', \Gamma'')}{comm\_eval((s, env), c_1; c_2, id, \Gamma) \Downarrow (s', env'', \Gamma'')}$$

$$\frac{rel\_op(b, env) \Downarrow Bool(true) \quad comm\_eval((s, env), c_1, \Gamma) \Downarrow (s', env', \Gamma')}{comm\_eval((s, env), if\ r\ then\ c_1\ else\ c_2, id, \Gamma) \Downarrow (s', env', \Gamma')}$$

$$\frac{rel\_op(b, env) \Downarrow Bool(false) \quad comm\_eval((s, env), c_2, \Gamma) \Downarrow (s', env', \Gamma')}{comm\_eval((s, env), if\ r\ then\ c_1\ else\ c_2, \Gamma) \Downarrow (s', env', \Gamma')}$$

$$\frac{newId(\Gamma, id) \Downarrow id'}{comm\_eval((s, env), spawn, id, \Gamma) \Downarrow (s, env, \Gamma[id'])}$$

- expr_eval: Aexp * $\Gamma \rightarrow$ v
- rel_op: Rexp * $\Gamma \rightarrow$ Bool(b)