BATBAMBAMMBA: Boolean and Arithmetic Languages
Oregon Programming Languages Summer School

Ronald Garcia

4 July 2017

1 BA: Boolean Arithmetic Language

Program Static Syntax

$$t \in \textsc{Term}, \quad n \in \mathbb{N}, \quad b \in \mathbb{B}, \quad p \in \textsc{Pgm} = \textsc{Term}$$
$$\begin{array}{rcl} t & ::= & \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ t\ \mathsf{then}\ t\ \mathsf{else}\ t \\ & \mid & n \mid \mathsf{succ}(t) \mid \mathsf{pred}(t) \mid \mathsf{zero?}(t) \\ b & ::= & \mathsf{true} \mid \mathsf{false} \end{array}$$

Program Runtime Syntax

$$E \in \textsc{ECtxt}, \quad v \in \textsc{Value}, \quad r \in \textsc{Redex} \subseteq \textsc{Pgm}, \quad f \in \textsc{Faulty} \subseteq \textsc{Pgm}, \quad err \in \textsc{Error}, \quad c \in \textsc{Config} \quad o \in \textsc{Obs}$$
$$\begin{array}{rcl} v & ::= & b \mid n \\ E & ::= & \Box \mid \mathsf{if}\ E\ \mathsf{then}\ p\ \mathsf{else}\ p \mid \mathsf{succ}(E) \mid \mathsf{pred}(E) \mid \mathsf{zero?}(E) \\ r & ::= & \mathsf{if}\ v\ \mathsf{then}\ p\ \mathsf{else}\ p \mid \mathsf{succ}(v) \mid \mathsf{pred}(v) \mid \mathsf{zero?}(v) \\ f & ::= & \mathsf{if}\ n\ \mathsf{then}\ p\ \mathsf{else}\ p \mid \mathsf{succ}(b) \mid \mathsf{pred}(b) \mid \mathsf{zero?}(b) \\ err & ::= & \mathsf{mismatch} \mid \mathsf{underflow} \\ c & ::= & p \mid err \\ o & ::= & v \mid err \end{array}$$

$\boxed{\rightsquigarrow \subseteq \textsc{Redex} \times \textsc{Pgm}}$ **Notions of Reduction**

$$\begin{array}{rcll} \mathsf{if\ true\ then}\ p_2\ \mathsf{else}\ p_3 & \rightsquigarrow & p_2 & \\ \mathsf{if\ false\ then}\ p_2\ \mathsf{else}\ p_3 & \rightsquigarrow & p_3 & \\ \mathsf{pred}(n) & \rightsquigarrow & n-1 & \text{if } n > 0 \\ \mathsf{succ}(n) & \rightsquigarrow & n+1 & \\ \mathsf{zero?}(0) & \rightsquigarrow & \mathsf{true} & \\ \mathsf{zero?}(n) & \rightsquigarrow & \mathsf{false} & \text{if } n > 0 \end{array}$$

$\boxed{\longrightarrow \subseteq \textsc{Pgm} \times \textsc{Config}}$ **Single-step Reduction**

$$\frac{r \rightsquigarrow p}{E[r] \longrightarrow E[p]} \qquad \frac{}{E[f] \longrightarrow \mathsf{mismatch}} \qquad \frac{}{E[\mathsf{pred}(0)] \longrightarrow \mathsf{underflow}}$$

$\boxed{\longrightarrow^* \subseteq \textsc{Config} \times \textsc{Config}}$ **Multi-step Reduction**

$$(\text{incl})\frac{c_1 \longrightarrow c_2}{c_1 \longrightarrow^* c_2} \qquad (\text{refl})\frac{}{c \longrightarrow^* c} \qquad (\text{trans})\frac{c_1 \longrightarrow^* c_2 \quad c_2 \longrightarrow^* c_3}{c_1 \longrightarrow^* c_3}$$

$\boxed{eval_{BA} : \textsc{Pgm} \to \textsc{Obs}}$

$$\begin{array}{l} eval_{BA}(p) = b \text{ if } p \longrightarrow^* b \\ eval_{BA}(p) = n \text{ if } p \longrightarrow^* n \\ eval_{BA}(p) = \mathsf{mismatch} \text{ if } p \longrightarrow^* \mathsf{mismatch} \\ eval_{BA}(p) = \mathsf{underflow} \text{ if } p \longrightarrow^* \mathsf{underflow} \end{array}$$

Safety (AKA Coherence AKA Definedness)

Conjecture 1 (Progress). *For all $p \in \textsc{Pgm}$ one of the following is true:*

1. $p \in \textsc{Value}$;
2. $p \longrightarrow p'$ *for some* $p' \in \textsc{Pgm}$;
3. $p \longrightarrow err$ *for some* $err \in \textsc{Error}$.

Conjecture 2 (Preservation (Vacuous)). *If $p_1 \longrightarrow p_2$ then $p_2 \in \textsc{Pgm}$. (uhh...?!?)*

TBA: Typed Boolean Arithmetic Language

Program Static Syntax

$t \in \text{TERM}, \quad n \in \mathbb{N}, \quad b \in \mathbb{B}, \quad$ **Same as BA**
$T \in \text{TYPE}, \quad p \in \text{PGM} = \{\, t \in \text{TERM} \mid \exists T \in \text{TYPE}.\ \vdash t : T \,\}$
$T ::= \mathsf{Nat} \mid \mathsf{Bool}$

Program Runtime Syntax

$E \in \text{ECTXT}, \quad v \in \text{VALUE}, \quad r \in \text{REDEX} \subseteq \text{PGM}, \quad$ **Same *grammar* as BA (over updated p)**
$err \in \text{ERROR}, \quad c \in \text{CONFIG} \quad o \in \text{OBS}$

$$\begin{array}{rcl} v & ::= & b \mid n \\ r & ::= & \mathsf{if}\ v\ \mathsf{then}\ p\ \mathsf{else}\ p \mid \mathsf{succ}(v) \mid \mathsf{pred}(v) \mid \mathsf{zero?}(v) \\ err & ::= & \mathsf{underflow} \\ c & ::= & p \mid err \\ o & ::= & v \mid err \end{array}$$

$\boxed{\rightsquigarrow \subseteq \text{REDEX} \times \text{PGM}}$ **Same formal schema as BA (over updated p)**
$\boxed{\longrightarrow \subseteq \text{PGM} \times \text{CONFIG}}$ **Single-step Reduction**

$$\frac{r \rightsquigarrow p}{E[r] \longrightarrow E[p]} \qquad \frac{}{E[\mathsf{pred}(0)] \longrightarrow \mathsf{underflow}}$$

$\models \cdot : \cdot \subseteq \text{TERM} \times \text{TYPE}$ **Semantic Typing** (Q: Should t be statically typed? Interesting implications!)

$\models t : \mathsf{Bool}$ if and only if $t \longrightarrow^* b$ or $t \longrightarrow^* \mathsf{underflow}$
$\models t : \mathsf{Nat}$ if and only if $t \longrightarrow^* n$ or $t \longrightarrow^* \mathsf{underflow}$

$\vdash \cdot : \cdot \subseteq \text{TERM} \times \text{TYPE}$ **Syntactic Typing**

$$\frac{}{\vdash \mathsf{true} : \mathsf{Bool}} \qquad \frac{}{\vdash \mathsf{false} : \mathsf{Bool}} \qquad \frac{\vdash t_1 : \mathsf{Bool} \quad \vdash t_2 : T \quad \vdash t_3 : T}{\vdash \mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3 : T} \qquad \frac{}{\vdash n : \mathsf{Nat}} \qquad \frac{\vdash t : \mathsf{Nat}}{\vdash \mathsf{succ}(t) : \mathsf{Nat}}$$

$$\frac{\vdash t : \mathsf{Nat}}{\vdash \mathsf{pred}(t) : \mathsf{Nat}} \qquad \frac{\vdash t : \mathsf{Nat}}{\vdash \mathsf{zero?}(t) : \mathsf{Bool}}$$

Evaluator $\boxed{eval_{TBA} : \text{PGM} \to \text{OBS}}$

$eval_{TBA}(p) = b$ if $p \longrightarrow^* b$
$eval_{TBA}(p) = n$ if $p \longrightarrow^* n$
$eval_{TBA}(p) = \mathsf{underflow}$ if $p \longrightarrow^* \mathsf{underflow}$

Safety

Conjecture 3 (Progress). *For all $p \in \text{PGM}$ one of the following is true:*

1. $p \in \text{VALUE}$;
2. $p \longrightarrow p'$ *for some* $p' \in \text{PGM}$;
3. $p \longrightarrow$ *underflow*.

Conjecture 4 (Preservation). *If* $\vdash p_1 : T$ *and* $p_1 \longrightarrow p_2$ *then* $\vdash p_2 : T$.

Conjecture 5 (Semantic Type Soundness). *If* $\vdash t : T$ *then* $\models t : T$.

2 MBA: Mixed Boolean and Arithmetic Language

Program Static Syntax

$s \in \text{SBA}$, $d \in \text{DBA}$, $n, ns, nd \in \mathbb{N}^\star$, $b, bs, bd \in \mathbb{B}^\star$, $\star$: A bit sloppy perhaps
$ps \in \text{SPGM} = \{ s \in \text{SBA} \mid \exists T \in \text{TYPE}.\ \vdash s : T \}$, $pd \in \text{DPGM} = \{ d \in \text{DBA} \mid \vdash d \checkmark \}$, $p \in \text{PGM} = ps$

$$
\begin{array}{rcl}
s & ::= & \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ s\ \mathsf{then}\ s\ \mathsf{else}\ s \\
 & \mid & ns \mid \mathsf{succ}(s) \mid \mathsf{pred}(s) \mid \mathsf{zero?}(s) \mid \lceil d \rceil \\
d & ::= & \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ d\ \mathsf{then}\ d\ \mathsf{else}\ d \\
 & \mid & nd \mid \mathsf{succ}(d) \mid \mathsf{pred}(d) \mid \mathsf{zero?}(d) \mid \lfloor s \rfloor \\
bs & ::= & \mathsf{true} \mid \mathsf{false} \\
bd & ::= & \mathsf{true} \mid \mathsf{false} \\
b & ::= & bs \mid bd
\end{array}
$$

$\vdash \cdot : \cdot \subseteq \text{SBA} \times \text{TYPE}$
$\vdash \cdot \checkmark \subseteq \text{DBA}$

Syntactic Typing

$$\frac{}{\vdash \mathsf{true} : \mathsf{Bool}} \qquad \frac{}{\vdash \mathsf{false} : \mathsf{Bool}} \qquad \frac{\vdash s_1 : \mathsf{Bool} \quad \vdash s_2 : T \quad \vdash s_3 : T}{\vdash \mathsf{if}\ s_1\ \mathsf{then}\ s_2\ \mathsf{else}\ s_3 : T} \qquad \frac{}{\vdash ns : \mathsf{Nat}} \qquad \frac{\vdash s : \mathsf{Nat}}{\vdash \mathsf{succ}(s) : \mathsf{Nat}}$$

$$\frac{\vdash s : \mathsf{Nat}}{\vdash \mathsf{pred}(s) : \mathsf{Nat}} \qquad \frac{\vdash s : \mathsf{Nat}}{\vdash \mathsf{zero?}(s) : \mathsf{Bool}} \qquad \frac{\vdash d \checkmark}{\vdash \lceil d \rceil : T}$$

$$\frac{}{\vdash \mathsf{true} \checkmark} \qquad \frac{}{\vdash \mathsf{false} \checkmark} \qquad \frac{\vdash d_1 \checkmark \quad \vdash d_2 \checkmark \quad \vdash d_3 \checkmark}{\vdash \mathsf{if}\ d_1\ \mathsf{then}\ d_2\ \mathsf{else}\ d_3 \checkmark} \qquad \frac{}{\vdash nd \checkmark} \qquad \frac{\vdash d \checkmark}{\vdash \mathsf{succ}(d) \checkmark} \qquad \frac{\vdash d \checkmark}{\vdash \mathsf{pred}(d) \checkmark}$$

$$\frac{\vdash d \checkmark}{\vdash \mathsf{zero?}(d) \checkmark} \qquad \frac{\vdash s : T}{\vdash \lfloor s \rfloor \checkmark}$$

Program Runtime Syntax

$Es \in \text{SECTXT}$, $Ed \in \text{DECTXT}$, $vs \in \text{SVALUE}$, $vd \in \text{DVALUE}$,
$rs \in \text{SREDEX}$, $rd \in \text{DREDEX}$, $fs \in \text{SFAULTY}$, $fd \in \text{DFAULTY}$, $cs \in \text{SCONFIG}$, ~~$cd \in \text{DCONFIG}$~~,
$v \in \text{VALUE}$, $err \in \text{ERROR}$, $o \in \text{OBS}$

$$
\begin{array}{rcl}
Es & ::= & \square \mid Es[\mathsf{if}\ \square\ \mathsf{then}\ ps\ \mathsf{else}\ ps] \mid Es[\mathsf{succ}(\square) \mid Es[\mathsf{pred}(\square)] \mid Es[\mathsf{zero?}(\square)] \mid Ed[\lfloor \square \rfloor] \\
Ed & ::= & Ed[\mathsf{if}\ \square\ \mathsf{then}\ pd\ \mathsf{else}\ pd] \mid Ed[\mathsf{succ}(\square) \mid Ed[\mathsf{pred}(\square)] \mid Ed[\mathsf{zero?}(\square)] \mid Es[\lceil \square \rceil] \\
vs & ::= & \mathsf{bs} \mid ns \mid \lceil bd \rceil \mid \lceil nd \rceil \\
vd & ::= & \mathsf{bd} \mid nd \mid \lfloor bs \rfloor \mid \lfloor ns \rfloor \\
rs & ::= & \mathsf{if}\ vs\ \mathsf{then}\ ps\ \mathsf{else}\ ps \mid \mathsf{succ}(vs) \mid \mathsf{pred}(vs) \mid \mathsf{zero?}(vs) \\
rd & ::= & \mathsf{if}\ vd\ \mathsf{then}\ pd\ \mathsf{else}\ pd \mid \mathsf{succ}(vd) \mid \mathsf{pred}(vd) \mid \mathsf{zero?}(vd) \\
fs & ::= & \mathsf{if}\ \lceil nd \rceil\ \mathsf{then}\ ps\ \mathsf{else}\ ps \mid \mathsf{succ}(\lceil bd \rceil) \mid \mathsf{pred}(\lceil bd \rceil) \mid \mathsf{zero?}(\lceil bd \rceil) \\
fd & ::= & \mathsf{if}\ nd\ \mathsf{then}\ pd\ \mathsf{else}\ pd \mid \mathsf{succ}(bd) \mid \mathsf{pred}(bd) \mid \mathsf{zero?}(bd) \\
 & \mid & \mathsf{if}\ \lfloor ns \rfloor\ \mathsf{then}\ pd\ \mathsf{else}\ pd \mid \mathsf{succ}(bd) \mid \mathsf{pred}(\lfloor bs \rfloor) \mid \mathsf{zero?}(\lfloor bs \rfloor) \\
err & ::= & \mathsf{mismatch} \mid \mathsf{underflow} \\
v & ::= & b \mid n \\
cs & ::= & ps \mid err \\
o & ::= & v \mid err
\end{array}
$$

$\boxed{\rightsquigarrow_s \subseteq \text{SREDEX} \times \text{SPGM})}$ **Static Notions Red.**

$$\begin{array}{rll}
\mathsf{if\ true\ then\ ps_2\ else\ ps_3} & \rightsquigarrow_s \mathsf{ps_2} & \\
\mathsf{if\ false\ then\ ps_2\ else\ ps_3} & \rightsquigarrow_s \mathsf{ps_3} & \\
\mathsf{pred(ns)} & \rightsquigarrow_s n-1 & \text{if } n>0 \\
\mathsf{succ(ns)} & \rightsquigarrow_s n+1 & \\
\mathsf{zero?(0)} & \rightsquigarrow_s \mathsf{true} & \\
\mathsf{zero?(ns)} & \rightsquigarrow_s \mathsf{false} & \text{if } n>0 \\
\mathsf{if}\ \lceil \mathsf{true} \rceil\ \mathsf{then\ ps_2\ else\ ps_3} & \rightsquigarrow_s \mathsf{ps_2} & \\
\mathsf{if}\ \lceil \mathsf{false} \rceil\ \mathsf{then\ ps_2\ else\ ps_3} & \rightsquigarrow_s \mathsf{ps_3} & \\
\mathsf{pred}(\lceil \mathsf{nd} \rceil) & \rightsquigarrow_s n-1 & \text{if } n>0 \\
\mathsf{succ}(\lceil \mathsf{nd} \rceil) & \rightsquigarrow_s n+1 & \\
\mathsf{zero?}(\lceil \mathsf{0} \rceil) & \rightsquigarrow_s \mathsf{true} & \\
\mathsf{zero?}(\lceil \mathsf{nd} \rceil) & \rightsquigarrow_s \mathsf{false} & \text{if } n>0 \\
[RG :???] & &
\end{array}$$

$\boxed{\rightsquigarrow_d \subseteq \text{DREDEX} \times \text{DPGM})}$ **Dynamic Notions of Red.**

$$\begin{array}{rll}
\mathsf{if\ true\ then\ pd_2\ else\ pd_3} & \rightsquigarrow_d \mathsf{pd_2} & \\
\mathsf{if\ false\ then\ pd_2\ else\ pd_3} & \rightsquigarrow_d \mathsf{pd_3} & \\
\mathsf{pred(nd)} & \rightsquigarrow_d n-1 & \text{if } n>0 \\
\mathsf{succ(nd)} & \rightsquigarrow_d n+1 & \\
\mathsf{zero?(0)} & \rightsquigarrow_d \mathsf{true} & \\
\mathsf{zero?(nd)} & \rightsquigarrow_d \mathsf{false} & \text{if } n>0 \\
\mathsf{if}\ \lceil \mathsf{true} \rceil\ \mathsf{then\ pd_2\ else\ pd_3} & \rightsquigarrow_d \mathsf{pd_2} & \\
\mathsf{if}\ \lceil \mathsf{false} \rceil\ \mathsf{then\ pd_2\ else\ pd_3} & \rightsquigarrow_d \mathsf{pd_3} & \\
\mathsf{pred}(\lfloor \mathsf{ns} \rfloor) & \rightsquigarrow_d n-1 & \text{if } n>0 \\
\mathsf{succ}(\lfloor \mathsf{ns} \rfloor) & \rightsquigarrow_d n+1 & \\
\mathsf{zero?}(\lfloor \mathsf{0} \rfloor) & \rightsquigarrow_d \mathsf{true} & \\
\mathsf{zero?}(\lfloor \mathsf{ns} \rfloor) & \rightsquigarrow_d \mathsf{false} & \text{if } n>0 \\
[RG :???] & &
\end{array}$$

$\boxed{\longrightarrow \subseteq \text{PGM} \times \text{CONFIG}}$ **Single-step Reduction**

$$\frac{\mathsf{rs} \rightsquigarrow \mathsf{ps}}{\mathsf{Es[rs]} \longrightarrow \mathsf{Es[ps]}} \qquad \frac{\mathsf{rd} \rightsquigarrow \mathsf{pd}}{\mathsf{Ed[rd]} \longrightarrow \mathsf{Ed[pd]}} \qquad \frac{}{\mathsf{Es[fs]} \longrightarrow \mathsf{mismatch}} \qquad \frac{}{\mathsf{Ed[fd]} \longrightarrow \mathsf{mismatch}}$$

$$\frac{}{\mathsf{Es[pred(0)]} \longrightarrow \mathsf{underflow}} \qquad \frac{}{\mathsf{Es[pred}(\lceil \mathsf{0} \rceil)\mathsf{]} \longrightarrow \mathsf{underflow}} \qquad \frac{}{\mathsf{Ed[pred(0)]} \longrightarrow \mathsf{underflow}}$$

$$\frac{}{\mathsf{Ed[pred}(\lfloor \mathsf{0} \rfloor)\mathsf{]} \longrightarrow \mathsf{underflow}}$$

$\models \cdot : \cdot \subseteq \text{TERM} \times \text{TYPE}$ **Semantic Typing**

$\models \mathsf{s} : \mathsf{Bool}$ if and only if $\vdash \mathsf{s} : \mathsf{Bool}$ and $\mathsf{s} \longrightarrow^* \mathsf{vs}$ implies $observe_{MBA}(\mathsf{vs}) \in B$

$\models \mathsf{s} : \mathsf{Nat}$ if and only if $\mathsf{s} \longrightarrow^* \mathsf{vs}$ implies $observe_{MBA}(\mathsf{vs}) \in \mathbb{N}$

$\boxed{observe_{MBA} : \text{SVALUE} \to \text{OBS}}$

$$\begin{aligned}
observe_{MBA}(\mathsf{bs}) &= b \\
observe_{MBA}(\lceil \mathsf{bd} \rceil) &= b \\
observe_{MBA}(\mathsf{ns}) &= n \\
observe_{MBA}(\lceil \mathsf{nd} \rceil) &= n
\end{aligned}$$

$\boxed{eval_{MBA} : \text{PGM} \to \text{OBS}}$

$$\begin{array}{lll}
eval_{MBA}(p) & = observe_{MBA}(\mathsf{vs}) & \text{if } p \longrightarrow^* \mathsf{vs} \\
eval_{MBA}(p) & = \mathsf{mismatch} & \text{if } p \longrightarrow^* \mathsf{mismatch} \\
eval_{MBA}(p) & = \mathsf{underflow} & \text{if } p \longrightarrow^* \mathsf{underflow}
\end{array}$$

Safety

Conjecture 6 (Progress)**.** *For all* $ps \in$ SPGM *one of the following is true:*

1. $ps \in$ SVALUE*;*
2. $ps \longrightarrow ps'$ *for some* $ps' \in$ SPGM*;*
3. $ps \longrightarrow err$ *for some* $err \in$ ERROR.

Conjecture 7 (Preservation)**.** *If* $\vdash p_1 : T$ *and* $p_1 \longrightarrow p_2$ *then* $\vdash p_2 : T$.

Conjecture 8 (Semantic Type Soundness)**.** *If* $\vdash t : T$ *then* $\models t : T$.

3 mMBA: Minimal Mixed Boolean and Arithmetic Language

Program Static Syntax

$\mathsf{s} \in \mathrm{SBA}$, $\mathsf{d} \in \mathrm{DBA}$, $n, \mathsf{ns} \in \mathbb{N}^\star$, $b, \mathsf{bs} \in \mathbb{B}^\star$, $\star$: A bit sloppy perhaps

$\mathsf{ps} \in \mathrm{SPGM} = \{\, \mathsf{s} \in \mathrm{SBA} \mid \exists T \in \mathrm{TYPE}.\ \vdash \mathsf{s} : T \,\}$, $\mathsf{pd} \in \mathrm{DPGM} = \{\, \mathsf{d} \in \mathrm{DBA} \mid \vdash \mathsf{d}\ \checkmark \,\}$, $p \in \mathrm{PGM} = \mathsf{ps}$

$$
\begin{array}{rcl}
\mathsf{s} & ::= & \mathsf{true} \mid \mathsf{false} \mid \mathsf{if\ s\ then\ s\ else\ s} \\
& \mid & \mathsf{ns} \mid \mathsf{succ(s)} \mid \mathsf{pred(s)} \mid \mathsf{zero?(s)} \mid \lceil \mathsf{d} \rceil \\
\mathsf{d} & ::= & \lfloor \mathsf{s} \rfloor \\
\mathsf{bs} & ::= & \mathsf{true} \mid \mathsf{false} \\
b & ::= & \mathsf{bs} \mid \mathsf{bd}
\end{array}
$$

$\boxed{\begin{array}{l} \vdash \cdot : \cdot \subseteq \mathrm{SBA} \times \mathrm{TYPE} \\ \vdash \cdot\ \checkmark \subseteq \mathrm{DBA} \end{array}}$ **Syntactic Typing**

$$\frac{}{\vdash \mathsf{true} : \mathsf{Bool}} \qquad \frac{}{\vdash \mathsf{false} : \mathsf{Bool}} \qquad \frac{\vdash \mathsf{s}_1 : \mathsf{Bool} \quad \vdash \mathsf{s}_2 : T \quad \vdash \mathsf{s}_3 : T}{\vdash \mathsf{if\ s}_1\ \mathsf{then\ s}_2\ \mathsf{else\ s}_3 : T} \qquad \frac{}{\vdash \mathsf{ns} : \mathsf{Nat}} \qquad \frac{\vdash \mathsf{s} : \mathsf{Nat}}{\vdash \mathsf{succ(s)} : \mathsf{Nat}}$$

$$\frac{\vdash \mathsf{s} : \mathsf{Nat}}{\vdash \mathsf{pred(s)} : \mathsf{Nat}} \qquad \frac{\vdash \mathsf{s} : \mathsf{Nat}}{\vdash \mathsf{zero?(s)} : \mathsf{Bool}} \qquad \frac{\vdash \mathsf{d}\ \checkmark}{\vdash \lceil \mathsf{d} \rceil : T}$$

$$\frac{\vdash \mathsf{s} : T}{\vdash \lfloor \mathsf{s} \rfloor\ \checkmark}$$

Program Runtime Syntax

$\mathsf{Es} \in \mathrm{SECTXT}$, $\mathsf{Ed} \in \mathrm{DECTXT}$, $\mathsf{vs} \in \mathrm{SVALUE}$, $\mathsf{vd} \in \mathrm{DVALUE}$,
$\mathsf{rs} \in \mathrm{SREDEX}$, $\mathsf{rd} \in \mathrm{DREDEX}$, $\mathsf{fs} \in \mathrm{SFAULTY}$, $\mathsf{fd} \in \mathrm{DFAULTY}$, $\mathsf{cs} \in \mathrm{SCONFIG}$, ~~$\mathsf{cd} \in \mathrm{DCONFIG}$~~,
$v \in \mathrm{VALUE}$, $err \in \mathrm{ERROR}$, $o \in \mathrm{OBS}$

$$
\begin{array}{rcl}
\mathsf{Es} & ::= & \Box \mid \mathsf{Es[if}\ \Box\ \mathsf{then\ ps\ else\ ps]} \mid \mathsf{Es[succ(}\Box\mathsf{)} \mid \mathsf{Es[pred(}\Box\mathsf{)]} \mid \mathsf{Es[zero?(}\Box\mathsf{)]} \mid \mathsf{Ed}[\lfloor \Box \rfloor] \\
\mathsf{Ed} & ::= & \mathsf{Es}[\lceil \Box \rceil] \\
\mathsf{vs} & ::= & \mathsf{bs} \mid \mathsf{ns} \\
\mathsf{vd} & ::= & \lfloor \mathsf{bs} \rfloor \mid \lfloor \mathsf{ns} \rfloor \\
\mathsf{rs} & ::= & \mathsf{if\ vs\ then\ ps\ else\ ps} \mid \mathsf{succ(vs)} \mid \mathsf{pred(vs)} \mid \mathsf{zero?(vs)} \\
\mathsf{rd} & ::= & \mathsf{if\ vd\ then\ pd\ else\ pd} \mid \mathsf{succ(vd)} \mid \mathsf{pred(vd)} \mid \mathsf{zero?(vd)} \\
\mathsf{fs} & ::= & \mathsf{if}\ \lceil \mathsf{nd} \rceil\ \mathsf{then\ ps\ else\ ps} \mid \mathsf{succ}(\lceil \mathsf{bd} \rceil) \mid \mathsf{pred}(\lceil \mathsf{bd} \rceil) \mid \mathsf{zero?}(\lceil \mathsf{bd} \rceil) \\
\mathsf{fd} & ::= & \mathsf{if\ nd\ then\ pd\ else\ pd} \mid \mathsf{succ(bd)} \mid \mathsf{pred(bd)} \mid \mathsf{zero?(bd)} \\
& \mid & \mathsf{if}\ \lfloor \mathsf{ns} \rfloor\ \mathsf{then\ pd\ else\ pd} \mid \mathsf{succ(bd)} \mid \mathsf{pred}(\lfloor \mathsf{bs} \rfloor) \mid \mathsf{zero?}(\lfloor \mathsf{bs} \rfloor) \\
err & ::= & \mathsf{mismatch} \mid \mathsf{underflow} \\
v & ::= & b \mid n \\
\mathsf{cs} & ::= & \mathsf{ps} \mid err \\
o & ::= & v \mid err
\end{array}
$$

$\boxed{\rightsquigarrow_s \subseteq \text{SREDEX} \times \text{SPGM})}$ **Static Notions Red.**

$$\begin{array}{rl}
\mathsf{if\ true\ then\ ps_2\ else\ ps_3} \rightsquigarrow_s \mathsf{ps_2} & \\
\mathsf{if\ false\ then\ ps_2\ else\ ps_3} \rightsquigarrow_s \mathsf{ps_3} & \\
\mathsf{pred(ns)} \rightsquigarrow_s n-1 & \text{if } n > 0 \\
\mathsf{succ(ns)} \rightsquigarrow_s n+1 & \\
\mathsf{zero?(0)} \rightsquigarrow_s \mathsf{true} & \\
\mathsf{zero?(ns)} \rightsquigarrow_s \mathsf{false} & \text{if } n > 0 \\
\mathsf{if\ \lceil true \rceil\ then\ ps_2\ else\ ps_3} \rightsquigarrow_s \mathsf{ps_2} & \\
\mathsf{if\ \lceil false \rceil\ then\ ps_2\ else\ ps_3} \rightsquigarrow_s \mathsf{ps_3} & \\
\mathsf{pred(\lceil nd \rceil)} \rightsquigarrow_s n-1 & \text{if } n > 0 \\
\mathsf{succ(\lceil nd \rceil)} \rightsquigarrow_s n+1 & \\
\mathsf{zero?(\lceil 0 \rceil)} \rightsquigarrow_s \mathsf{true} & \\
\mathsf{zero?(\lceil ns \rceil)} \rightsquigarrow_s \mathsf{false} & \text{if } n > 0 \\
[RG :???] &
\end{array}$$

$\boxed{\rightsquigarrow_d \subseteq \text{DREDEX} \times \text{DPGM})}$ **Dynamic Notions of Red.**

$$\begin{array}{rl}
\mathsf{if\ true\ then\ pd_2\ else\ pd_3} \rightsquigarrow_s \mathsf{pd_2} & \\
\mathsf{if\ false\ then\ pd_2\ else\ pd_3} \rightsquigarrow_s \mathsf{pd_3} & \\
\mathsf{pred(nd)} \rightsquigarrow_s n-1 & \text{if } n > 0 \\
\mathsf{succ(nd)} \rightsquigarrow_s n+1 & \\
\mathsf{zero?(0)} \rightsquigarrow_s \mathsf{true} & \\
\mathsf{zero?(nd)} \rightsquigarrow_s \mathsf{false} & \text{if } n > 0 \\
\mathsf{if\ \lceil true \rceil\ then\ ps_2\ else\ ps_3} \rightsquigarrow_s \mathsf{ps_2} & \\
\mathsf{if\ \lceil false \rceil\ then\ ps_2\ else\ ps_3} \rightsquigarrow_s \mathsf{ps_3} & \\
\mathsf{pred(\lfloor ns \rfloor)} \rightsquigarrow_s n-1 & \text{if } n > 0 \\
\mathsf{succ(\lfloor nd \rfloor)} \rightsquigarrow_s n+1 & \\
\mathsf{zero?(\lfloor 0 \rfloor)} \rightsquigarrow_s \mathsf{true} & \\
\mathsf{zero?(\lfloor ns \rfloor)} \rightsquigarrow_s \mathsf{false} & \text{if } n > 0 \\
[RG :???] &
\end{array}$$

$\boxed{\longrightarrow \subseteq \text{PGM} \times \text{CONFIG}}$ **Single-step Reduction**

$$\frac{\mathsf{rs} \rightsquigarrow \mathsf{ps}}{\mathsf{Es[rs]} \longrightarrow \mathsf{Es[ps]}} \qquad \frac{\mathsf{rd} \rightsquigarrow \mathsf{pd}}{\mathsf{Ed[rd]} \longrightarrow \mathsf{Ed[pd]}} \qquad \frac{}{\mathsf{Es[fs]} \longrightarrow \mathsf{mismatch}} \qquad \frac{}{\mathsf{Ed[fd]} \longrightarrow \mathsf{mismatch}}$$

$$\frac{}{\mathsf{Es[pred(0)]} \longrightarrow \mathsf{underflow}} \qquad \frac{}{\mathsf{Es[pred(\lceil 0 \rceil)]} \longrightarrow \mathsf{underflow}} \qquad \frac{}{\mathsf{Ed[pred(0)]} \longrightarrow \mathsf{underflow}}$$

$$\frac{}{\mathsf{Ed[pred(\lfloor 0 \rfloor)]} \longrightarrow \mathsf{underflow}}$$

$\models \cdot : \cdot \subseteq \text{TERM} \times \text{TYPE}$ **Semantic Typing** [RG: Exercise! Here's an old one:]

$\models t : \mathsf{Bool}$ if and only if $t \longrightarrow^* b$ or $t \longrightarrow^* \mathsf{underflow}$

$\models t : \mathsf{Nat}$ if and only if $t \longrightarrow^* n$ or $t \longrightarrow^* \mathsf{underflow}$

$\boxed{eval_{MBA} : \text{PGM} \to \text{OBS}}$

$$\begin{array}{l}
eval_{MBA}(p) = b \text{ if } p \longrightarrow^* \mathsf{bs} \\
eval_{MBA}(p) = b \text{ if } p \longrightarrow^* \lceil \mathsf{bd} \rceil \\
eval_{MBA}(p) = n \text{ if } p \longrightarrow^* \mathsf{ns} \\
eval_{MBA}(p) = n \text{ if } p \longrightarrow^* \lceil \mathsf{nd} \rceil \\
eval_{MBA}(p) = \mathsf{mismatch} \text{ if } p \longrightarrow^* \mathsf{mismatch} \\
eval_{MBA}(p) = \mathsf{underflow} \text{ if } p \longrightarrow^* \mathsf{underflow}
\end{array}$$

Safety

Conjecture 9 (Progress). *For all* $\mathsf{ps} \in$ SPGM *one of the following is true:*

1. $\mathsf{ps} \in$ SVALUE*;*
2. $\mathsf{ps} \longrightarrow \mathsf{ps}'$ *for some* $\mathsf{ps}' \in$ SPGM*;*
3. $\mathsf{ps} \longrightarrow err$ *for some* $err \in$ ERROR.

Conjecture 10 (Preservation). *If* $\vdash p_1 : T$ *and* $p_1 \longrightarrow p_2$ *then* $\vdash p_2 : T$.

Conjecture 11 (Semantic Type Soundness). *If* $\vdash t : T$ *then* $\models t : T$.