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

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

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

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

$\boxed{\longrightarrow \subseteq \textsc{Pgm} \times \textsc{Config}}$ **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)

Proposition 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* $err \in \textsc{Error}$.

Proposition 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)**
$err \in \text{ERROR}, \quad c \in \text{CONFIG} \quad o \in \text{OBS}$

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

$\rightsquigarrow \subseteq \text{REDEX} \times \text{PGM}$ **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**

$\models t : \mathsf{Bool}$ 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**

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

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

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

Proposition 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. $t \longrightarrow$ *underflow*.

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

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

2 MBA: Mixed Boolean and Arithmetic Language

Missing!

3 mMBA: Minimal Mixed Boolean and Arithmetic Language

Missing!