Statics and Dynamics: E ($L^{num,str}$)

Yu Zhang

http://staff.ustc.edu.cn/~yuzhang/pldpa/

yuzhang@ustc.edu.cn

References

- [PFPL]
  - Chapters: E ($L^{\text{num,str}}$)
    - 4 Statics, 5 Dynamics
    - 6 Type Safety
    - 7 Evaluation Dynamics
- [TAPL]
- [The algebra (and calculus!) of algebraic data types]

Outline

- Syntax
- Statics
- Dynamics
- Type Safety
- Evaluation Dynamics

Syntax

- Abstract syntax vs. concrete syntax

Typ	τ	$::=$	$\mathtt{num}$	$\mathtt{num}$	numbers
			$\mathtt{str}$	$\mathtt{str}$	strings
Exp	e	$::=$	x	x	variable
			$\mathtt{num}[n]$	n	numeral
			$\mathtt{str}[s]$	$"s"$	literal
			$\mathtt{plus}(e_1;e_2)$	$e_1 + e_2$	addition
			$\mathtt{times}(e_1;e_2)$	$e_1 * e_2$	multiplication
			$\mathtt{cat}(e_1;e_2)$	e_1 ^ e_2	concatenation
			$\mathtt{len}(e)$	$\|e\|$	length
			$\mathtt{let}(e_1;x.e_2)$	$\mathtt{let}\ x\ \mathtt{be}\ e_1\ \mathtt{in}\ e_2$	definition

Semantics

Typ	τ	::=	num
			str
Exp	e	::=	x
			$\mathtt{num}[n]$
			$\mathtt{str}[s]$
			$\mathtt{plus}(e_1;e_2)$
			$\mathtt{times}(e_1;e_2)$
			$\mathtt{cat}(e_1;e_2)$
			$\mathtt{len}(e)$
			$\mathtt{let}(e_1;x.e_2)$

- Semantics: judgements, rules
- **Statics** (Type System)
  - Generic hypothetical judgments $\vec{x} \mid \Gamma \vdash e : \tau$,

variables

typing context: $x : \tau$

  - Rules

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}$$

$$\frac{}{\Gamma \vdash \mathtt{num}[n] : \mathtt{num}}$$

Introduction form of num type (引入num类型的值)

$$\frac{}{\Gamma \vdash \mathtt{str}[s] : \mathtt{str}}$$

$$\frac{\Gamma \vdash e_1 : \mathtt{num} \quad \Gamma \vdash e_2 : \mathtt{num}}{\Gamma \vdash \mathtt{plus}(e_1;e_2) : \mathtt{num}}$$

$$\frac{\Gamma \vdash e_1 : \mathtt{num} \quad \Gamma \vdash e_2 : \mathtt{num}}{\Gamma \vdash \mathtt{times}(e_1;e_2) : \mathtt{num}}$$

Elimination form of num type (消去) (操作num类型的值)

$$\frac{\Gamma \vdash e_1 : \mathtt{str} \quad \Gamma \vdash e_2 : \mathtt{str}}{\Gamma \vdash \mathtt{cat}(e_1;e_2) : \mathtt{str}}$$

$$\frac{\Gamma \vdash e : \mathtt{str}}{\Gamma \vdash \mathtt{len}(e) : \mathtt{num}}$$

Semantics

- Type system

variable x is not declared in Γ

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \mathtt{let}(e_1; x.e_2) : \tau_2}$$

  - Unicity(唯一性): $\Gamma \vdash \mathrm{e}: \tau$, at most one type for each exp
  - Inversion for typing(定型反转)
    - If $\Gamma \vdash \mathrm{e}: \tau$, $\mathrm{e} = \mathrm{plus}(e_1; e_2)$, then $\tau = \mathrm{num}$, $\Gamma \vdash e_1 : \mathrm{num}$, and $\Gamma \vdash$

$$\frac{\Gamma \vdash e_1 : \mathtt{num} \quad \Gamma \vdash e_2 : \mathtt{num}}{\Gamma \vdash \mathtt{plus}(e_1; e_2) : \mathtt{num}}$$

  - Weakening(弱化)
    - If $\Gamma \vdash \mathrm{e}' : \tau'$, then $\Gamma, x: \tau \vdash \mathrm{e}' : \tau'$ for any $\mathrm{x} \notin dom(\Gamma)$ and any type τ
  - Substitution(代换)
    - If $\Gamma, x: \tau \vdash \mathrm{e}' : \tau'$ and $\Gamma \vdash \mathrm{e}: \tau$, then $\Gamma \vdash [e/x]\mathrm{e}' : \tau'$
  - Decomposition(分解)
    - If $\Gamma \vdash [e/x]\mathrm{e}' : \tau'$, then for each type τ such that $\Gamma \vdash \mathrm{e}: \tau$, we have $\Gamma, x: \tau \vdash \mathrm{e}' : \tau'$

Dynamics: Transition Systems

- Structural dynamics (step-by-step, small step)
  - Judgements
    - s state，断言s是转换系统的一个状态。
    - s final，对s state，断言s是一个终结状态。
    - s initial，对s state，断言s是一个初始状态。
    - $s \longmapsto s'$，对s state，s' state，断言状态s可以转换到状态 s'。
    - $s \longmapsto^{*} s'$

$$\frac{}{s \longmapsto^{*} s} \quad (5.1a)$$

$$\frac{s \longmapsto s' \quad s' \longmapsto^{*} s''}{s \longmapsto^{*} s''} \quad (5.1b)$$

Dynamics: Transition Systems

- Values: e val

$$\frac{}{\texttt{num}[n]\ \mathsf{val}}$$

$$\frac{}{\texttt{str}[s]\ \mathsf{val}}$$

- Transition judgment $e \longmapsto e'$

$$\frac{n_1 + n_2 = n}{\texttt{plus}(\texttt{num}[n_1];\texttt{num}[n_2]) \longmapsto \texttt{num}[n]}$$

$$\frac{s_1 \mathbin{\hat{}} s_2 = s\ \mathsf{str}}{\texttt{cat}(\texttt{str}[s_1];\texttt{str}[s_2]) \longmapsto \texttt{str}[s]}$$

$$\frac{e_1 \longmapsto e_1'}{\texttt{plus}(e_1;e_2) \longmapsto \texttt{plus}(e_1';e_2)}$$

$$\frac{e_1 \longmapsto e_1'}{\texttt{cat}(e_1;e_2) \longmapsto \texttt{cat}(e_1';e_2)}$$

$$\frac{e_1\ \mathsf{val} \quad e_2 \longmapsto e_2'}{\texttt{plus}(e_1;e_2) \longmapsto \texttt{plus}(e_1;e_2')}$$

$$\frac{e_1\ \mathsf{val} \quad e_2 \longmapsto e_2'}{\texttt{cat}(e_1;e_2) \longmapsto \texttt{cat}(e_1;e_2')}$$

$$\left[\frac{e_1 \longmapsto e_1'}{\texttt{let}(e_1;x.e_2) \longmapsto \texttt{let}(e_1';x.e_2)}\right]$$

$$\frac{[e_1\ \mathsf{val}]}{\texttt{let}(e_1;x.e_2) \longmapsto [e_1/x]e_2}$$

Included for call-by-value, Omitted for call-by-name

Search transitions
决定指令执行次序

Instruction transitions
基础的求值步

Dynamics: Transition Systems

- Examples

$$
\begin{aligned}
&\texttt{let}(\texttt{plus}(\texttt{num}[1];\texttt{num}[2]);x.\texttt{plus}(\texttt{plus}(x;\texttt{num}[3]);\texttt{num}[4]))\\
&\quad\longmapsto\quad \texttt{let}(\texttt{num}[3];x.\texttt{plus}(\texttt{plus}(x;\texttt{num}[3]);\texttt{num}[4]))\\
&\quad\longmapsto\quad \texttt{plus}(\texttt{plus}(\texttt{num}[3];\texttt{num}[3]);\texttt{num}[4])\\
&\quad\longmapsto\quad \texttt{plus}(\texttt{num}[6];\texttt{num}[4])\\
&\quad\longmapsto\quad \texttt{num}[10]
\end{aligned}
$$

- Finality of values
  - For no e, we have both e val and $e \longmapsto e'$
- Determinacy
  - If $e \longmapsto e'$ and $e \longmapsto e''$, then e' and e'' are α-equivalent

Contextual Dynamics

- Contextual Dynamics
  - Judgement $\mathcal{E}$ ectxt
    - The position of the next instruction step is specified by a "*hole*"
  - Judgement $e' = \mathcal{E}\{e\}$
    - filling the hole in the evaluation context $\mathcal{E}$ with the exp. e

$$\frac{}{\circ \text{ ectxt}}$$

$$\frac{\mathcal{E}_1 \text{ ectxt}}{\texttt{plus}(\mathcal{E}_1; e_2) \text{ ectxt}}$$

$$\frac{e_1 \text{ val} \quad \mathcal{E}_2 \text{ ectxt}}{\texttt{plus}(e_1; \mathcal{E}_2) \text{ ectxt}}$$

$$\frac{}{e = \circ\{e\}}$$

$$\frac{e_1 = \mathcal{E}_1\{e\}}{\texttt{plus}(e_1; e_2) = \texttt{plus}(\mathcal{E}_1; e_2)\{e\}}$$

$$\frac{e_1 \text{ val} \quad e_2 = \mathcal{E}_2\{e\}}{\texttt{plus}(e_1; e_2) = \texttt{plus}(e_1; \mathcal{E}_2)\{e\}}$$

Contextual Dynamics

$$\frac{e = \mathcal{E}\{e_0\} \quad e_0 \to e_0' \quad e' = \mathcal{E}\{e_0'\}}{e \longmapsto e'}$$

Type Safety

- A language is *safe*/*sound* if satisfying two theorems
  - Preservation
    - If $e : \tau$ and $e \longmapsto e'$, then $e' : \tau$
  - Progress
    - If $e : \tau$, then either e val, or there exists e' such that $e \longmapsto e'$

    Prove the theorem by *induction* on the *typing rules*

interpreter.ml

Run-time Errors

- Zero divisor
  - Well-typed, yet stuck!
  - How to correct
    - Enhance the type system ☹
    - Add dynamic checks ☺

$$\frac{e_1 : \texttt{num} \quad e_2 : \texttt{num}}{\texttt{div}(e_1; e_2) : \texttt{num}} \;.$$

Statics $\quad \overline{\Gamma \vdash \texttt{error} : \tau}$

$$\frac{e_1 \ \mathsf{val}}{\texttt{div}(e_1; \texttt{num}[0]) \ \mathsf{err}}$$

$$\frac{e_1 \ \mathsf{err}}{\texttt{div}(e_1; e_2) \ \mathsf{err}}$$

Dynamics $\quad \overline{\texttt{error} \ \mathsf{err}}$

$$\frac{e_1 \ \mathsf{val} \quad e_2 \ \mathsf{err}}{\texttt{div}(e_1; e_2) \ \mathsf{err}}$$

Progress with error

If e: τ, then either e val, or e err, or there exists e' such that e $\longmapsto e'$

Evaluation Dynamics

- Evaluation Dynamics (big step)
  - Evaluation judgment: $e \Downarrow v$
  - Rules

$$\frac{}{\mathrm{num}[n] \Downarrow \mathrm{num}[n]} \qquad \frac{}{\mathrm{str}[s] \Downarrow \mathrm{str}[s]}$$

$$\frac{e_1 \Downarrow \mathrm{num}[n_1] \quad e_2 \Downarrow \mathrm{num}[n_2] \quad n_1 + n_2\ is\ n\ \mathrm{nat}}{\mathrm{plus}(e_1; e_2) \Downarrow \mathrm{num}[n]} \qquad \frac{e_1 \Downarrow \mathrm{str}[s_1] \quad e_2 \Downarrow \mathrm{str}[s_2] \quad s_1 \wedge s_2 = s\ \mathrm{str}}{\mathrm{cat}(e_1; e_2) \Downarrow \mathrm{str}[s]}$$

$$\frac{e \Downarrow \mathrm{str}[s] \quad |s| = n\ \mathrm{nat}}{\mathrm{len}(e) \Downarrow \mathrm{num}[n]}$$

$$\frac{[e_1/x]e_2 \Downarrow v_2}{\mathrm{let}(e_1; x.e_2) \Downarrow v_2}$$

by-name interpretation

by-value interpretation

$$\frac{e_1 \Downarrow v_1 \quad [v_1/x]e_2 \Downarrow v_2}{\mathrm{let}(e_1; x.e_2) \Downarrow v_2}$$

  - If $e \Downarrow v$, then v val

Structural vs. Evaluation Dynamics

- For all closed expressions (闭式) e and values v, $e \longmapsto^* v$ iff $e \Downarrow v$.
  - If $e \Downarrow v$, then e $\longmapsto^* v$
  - If e $\longmapsto e'$ and $e' \Downarrow v$, then $e \Downarrow v$
- Type safety
  - Cannot be proved if using evaluation dynamics
    - Have an analog of the preservation property, but no clear analog of the progress property
    - **Progress**: If $e : \tau$, then $e \Downarrow v$ for some v requires that every expression evaluate to a value! If E were extended to admit non-terminating expressions or operations that may cause an error, the progress would fail.

Cost Dynamics

- *Time complexity* for programs
  - Structural dynamics: the number of steps
  - Evaluation dynamics: does not provide
- Cost dynamics
  - Evaluation judgment $e \Downarrow^{k} v$
  - Rules

$$\frac{}{\mathrm{num}[n] \Downarrow^{0} \mathrm{num}[n]} \qquad \frac{e_1 \Downarrow^{k_1} \mathrm{num}[n_1] \quad e_2 \Downarrow^{k_2} \mathrm{num}[n_2]}{\mathrm{plus}(e_1; e_2) \Downarrow^{k_1+k_2+1} \mathrm{num}[n_1 + n_2]}$$

$$\frac{}{\mathrm{str}[s] \Downarrow^{0} \mathrm{str}[s]} \qquad \frac{e_1 \Downarrow^{k_1} s_1 \quad e_2 \Downarrow^{k_2} s_1}{\mathrm{cat}(e_1; e_2) \Downarrow^{k_1+k_2+1} \mathrm{str}[s_1 \wedge s_2]}$$

by-name interpretation

by-value interpretation

$$\frac{[e_1/x]e_2 \Downarrow^{k_2} v_2}{\mathrm{let}(e_1; x.e_2) \Downarrow^{k_2+1} v_2} \qquad \frac{e_1 \Downarrow^{k_1} v_1 \quad [v_1/x]e_2 \Downarrow^{k_2} v_2}{\mathrm{let}(e_1; x.e_2) \Downarrow^{k_1+k_2+1} v_2}$$

  - For all closed expressions e and values v, $e \Downarrow^{k} v$ iff $e \longmapsto^{k} v$.