《软件分析与验证》
第四次习题课：形式语义回顾

朱俸民

清华大学

2020 年 4 月

Contents

分类

Operational semantics: The meaning of a construct is specified by the computation it induces when it is executed on a machine. In particular, it is of interest **how** the effect of a computation is produced.

Denotational semantics: Meanings are modelled by mathematical objects that represent the effect of executing the constructs. Thus only the **effect** is of interest, not how it is obtained.

Axiomatic semantics: Specific properties of the effect of executing the constructs are expressed as **assertions**. Thus there may be aspects of the executions that are ignored.

– Semantics with Applications: An Appetizer

Contents

语法

项/表达式

$$e \in E ::= \mathsf{Const}(n) \mid \mathsf{Plus}(e_1, e_2)$$

其中 n 为自然数。

元变元 (meta-variable)
归纳定义 (inductive definition)
抽象语法 (abstract syntax)

大步操作语义

$$[\![\mathsf{Const}(n)]\!] = n$$
$$[\![\mathsf{Plus}(e_1, e_2)]\!] = [\![e_1]\!] + [\![e_2]\!]$$

语义是一个函数/关系

语法制导 (syntax-directed)

语义对象 v.s. **语法对象**

另一种写法

$$\text{(Const)}\ \frac{}{[\![\text{Const}(n)]\!] = n}$$

$$\text{(Plus)}\ \frac{[\![e_1]\!] = n_1 \qquad [\![e_2]\!] = n_2}{[\![\text{Plus}(e_1, e_2)]\!] = n_1 + n_2}$$

规则模式 (rule scheme)

横线：**蕴含**

横线上方的空格：**合取** (conjunction)

上方为空：**公理** (axiom)

证明

如何证明 $[\![\mathsf{Plus}(\mathsf{Const}(1), \mathsf{Plus}(\mathsf{Const}(2), \mathsf{Const}(3)))]\!] = 6$?

“线性”的证明

“结构化”的证明：证明树 (proof tree)

还有一种写法

$$\text{(R-Const)}\ \frac{}{R(\mathsf{Const}(n), n)}$$

$$\text{(R-Plus)}\ \frac{R(e_1, n_1) \qquad R(e_2, n_2)}{R(\mathsf{Plus}(e_1, e_2), n_1 + n_2)}$$

其中 $R(e, n)$ 为 E 与 $\mathbb{N}$ 上的二元关系。

小步操作语义：“左型”

$$\text{(ST-L-PlusCC)}\ \frac{}{\mathsf{Plus}(\mathsf{Const}(n_1), \mathsf{Const}(n_2)) \to_L \mathsf{Const}(n_1 + n_2)}$$

$$\text{(ST-L-Plus1)}\ \frac{e_1 \to_L e_1'}{\mathsf{Plus}(e_1, e_2) \to_L \mathsf{Plus}(e_1', e_2)}$$

$$\text{(ST-L-Plus2)}\ \frac{e_2 \to_L e_2'}{\mathsf{Plus}(\mathsf{Const}(n_1), e_2) \to_L \mathsf{Plus}(\mathsf{Const}(n_1), e_2')}$$

其中 $\to_L$ 是 E 上的二元关系。

语法制导 (syntax-directed)

有些项没有可应用的规则

确定性 (deterministic)：若 $e \to_L e_1$ 且 $e \to_L e_2$，则 $e_1 = e_2$

多步归约

记二元关系 $\to_L^*$ 为 $\to_L$ 的自反传递闭包，即：

$$(\text{Refl})\ \frac{}{e \to_L^* e}$$

$$(\text{Trans})\ \frac{e_1 \to_L^* e_2 \qquad e_2 \to_L^* e_3}{e_1 \to_L^* e_3}$$

思考：证明
$\text{Plus}(\text{Const}(1), \text{Plus}(\text{Const}(2), \text{Const}(3))) \to_L^* \text{Const}(6)$

范式/常态

一般定义：给定一个集合 S 上的二元关系 R，设 R^* 为其自反传递闭包。若不存在 $s' \in S$ 使得 $R(s, s')$，则称 s 为范式/常态 (normal form)。

例如：在 $\to_L$ 系统中，$\mathsf{Const}(6)$ 是一个常态。

来一点奇怪的操作

去掉 (ST-L-Plus2)：

$$\text{(ST-L-PlusCC)}\ \frac{}{\mathsf{Plus}(\mathsf{Const}(n_1), \mathsf{Const}(n_2)) \to_L \mathsf{Const}(n_1 + n_2)}$$

$$\text{(ST-L-Plus1)}\ \frac{e_1 \to_L e_1'}{\mathsf{Plus}(e_1, e_2) \to_L \mathsf{Plus}(e_1', e_2)}$$

发现 $\mathsf{Plus}(\mathsf{Const}(1), \mathsf{Plus}(\mathsf{Const}(2), \mathsf{Const}(3)))$ 是常态，这符合直观吗？

小步操作语义：“右型”

$$\text{(ST-R-PlusCC)}\ \frac{}{\mathsf{Plus}(\mathsf{Const}(n_1), \mathsf{Const}(n_2)) \to_R \mathsf{Const}(n_1 + n_2)}$$

$$\text{(ST-R-Plus1)}\ \frac{e_2 \to_R e_2'}{\mathsf{Plus}(e_1, e_2) \to_R \mathsf{Plus}(e_1, e_2')}$$

$$\text{(ST-R-Plus2)}\ \frac{e_1 \to_R e_1'}{\mathsf{Plus}(e_1, \mathsf{Const}(n_2)) \to_R \mathsf{Plus}(e_1', \mathsf{Const}(n_2))}$$

其中 $\to_R$ 是 E 上的二元关系。

小步操作语义：“非确定型”

$$\text{(ST-N-PlusCC)}\ \frac{}{\mathsf{Plus}(\mathsf{Const}(n_1), \mathsf{Const}(n_2)) \to_N \mathsf{Const}(n_1 + n_2)}$$

$$\text{(ST-N-Plus1)}\ \frac{e_1 \to_N e_1'}{\mathsf{Plus}(e_1, e_2) \to_N \mathsf{Plus}(e_1', e_2)}$$

$$\text{(ST-N-Plus2)}\ \frac{e_2 \to_N e_2'}{\mathsf{Plus}(e_1, e_2) \to_N \mathsf{Plus}(e_2, e_2')}$$

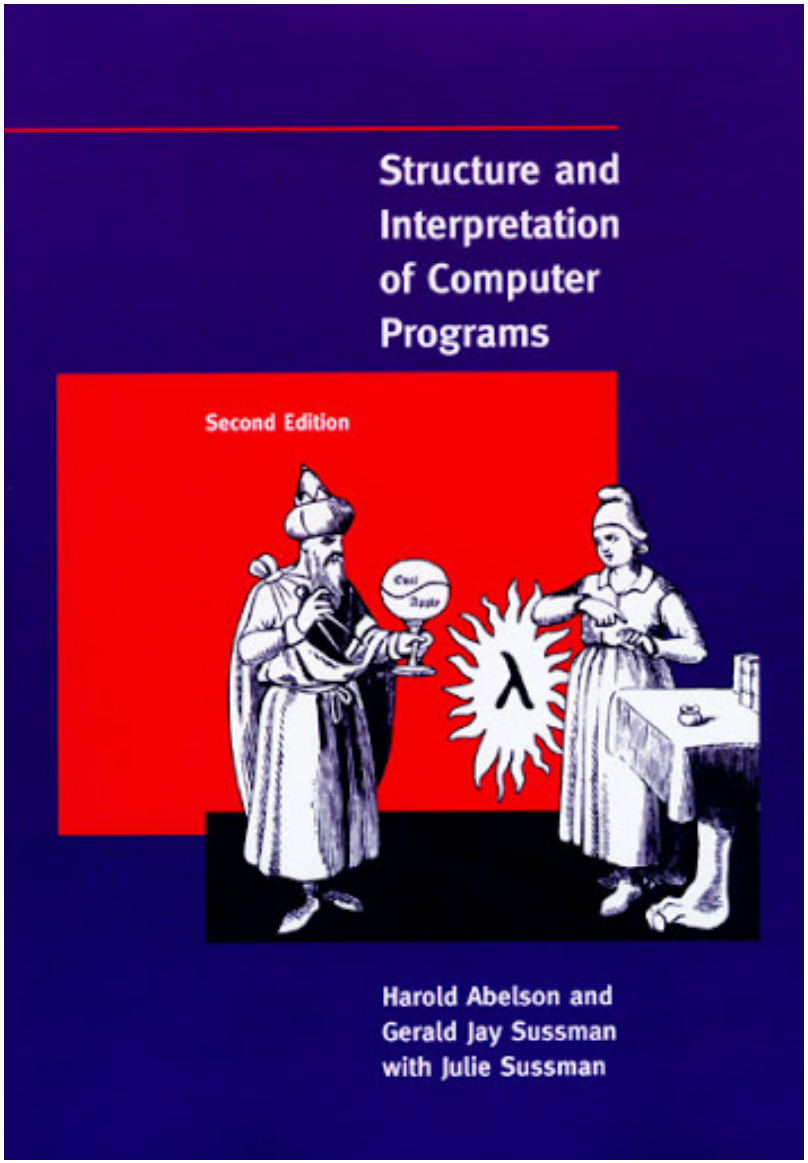
其中 $\to_N$ 是 E 上的二元关系。

等价性

不难发现，$\to_L$，$\to_R$ 和 $\to_N$ 是三种不同但“等价”的设计：若 $e \to_L^* e_1$，$e \to_R^* e_2$，$e \to_N^* e_3$ 且 e_1，e_2，e_3 均为常态，则 $e_1 = e_2 = e_3$。

Contents

SICP


Structure and
Interpretation
of Computer
Programs
Second Edition
λ
Harold Abelson and
Gerald Jay Sussman
with Julie Sussman

语法

$$
\begin{aligned}
\text{项 } t ::= {} & x && \text{(variable)} \\
\mid {} & (t_1\, t_2) && (\lambda\text{-application}) \\
\mid {} & (\lambda x.t) && (\lambda\text{-abstraction})
\end{aligned}
$$

其中，$x \in V$ 是一个变元，λ 相当于量词（如同 $\forall$）。

惯例：λ-application 具有最高优先级、左结合；λ-abstraction 的点具有最低优先级。

非确定性小步语义

$$(\beta)\frac{}{(\lambda x.t_1)\,t_2 \to t_1[x := t_2]}$$

$$(\text{App1})\frac{t_1 \to t_1'}{t_1\,t_2 \to t_1'\,t_2} \qquad (\text{App2})\frac{t_2 \to t_2'}{t_1\,t_2 \to t_1\,t_2'}$$

其中，$\to$ 是项上的二元关系；记号 $t[x := t']$ 表示将 t 中所有自由出现的 x 替换为 t'。同样有 $\to^*$ 是 $\to$ 的自反传递闭包。

思考：(1) 哪些项是常态？(2) 为何非确定？

非确定性不好吗

定理

(Church-Rosser) 对任意项 t, t_1, t_2，若 $t \to^* t_1$ 且 $t \to^* t_2$，则存在一个项 s 使得 $t_1 \to^* s$ 且 $t_2 \to^* s$。

推论

如果一个项能求值到常态，那么常态是唯一的。

换言之，无论按照何种顺序归约，最终结果都一样！

确定性小步语义：Call-by-value

定义谓词 $val(v)$ 表示项 v 是一个 λ-abstraction 或变元。

$$\text{(CBV-App1)}\frac{t_1 \to_{CBV} t_1'}{t_1\, t_2 \to_{CBV} t_1'\, t_2}$$

$$\text{(CBV-App2)}\frac{t_2 \to_{CBV} t_2'}{(\lambda x.t)\, t_2 \to_{CBV} (\lambda x.t)\, t_2'}$$

$$\text{(CBV-}\beta\text{)}\frac{val(v)}{(\lambda x.t)\, v \to_{CBV} t[x := v]}$$

直观含义：函数调用时，先求值参数到常态。

确定性小步语义：Call-by-name

$$\text{(CBN-App)}\frac{t_1 \to_{CBN} t_1'}{t_1\, t_2 \to_{CBN} t_1'\, t_2}$$

$$\text{(CBN-}\beta\text{)}\frac{}{(\lambda x.t)\, t' \to_{CBN} t[x := t']}$$

直观含义：函数调用时，直接用参数替换函数体。

每个项都可以求值到常态吗

经典反例：$\omega\omega$，**其中** $\omega = \lambda x.xx$

小结

纯 (pure) 语言的小步语义比不纯 (impure) 的在形式上更“简单”

小结

纯 (pure) 语言的小步语义比不纯 (impure) 的在形式上更“简单”

研究形式系统

- “理解”规则，包括一些非形式化的直观
- 内定理证明（本课程要求）
- 元性质证明（本课程不要求）

小结

纯 (pure) 语言的小步语义比不纯 (impure) 的在形式上更“简单”

研究形式系统

- “理解”规则，包括一些非形式化的直观
- 内定理证明（本课程要求）
- 元性质证明（本课程不要求）

形式语义有什么用

- 更准确地理解已学过的语言（或片段）的语义
- 更本质地了解新语言（或片段）的语义
- 一种研究程序语言 (PL) 的手段