操作的意味論

国立情報学研究所

佐藤一郎
E-mail: ichiro@nii.ac.jp

操作的意味論

抽象的な計算機を定義し、プログラミング言語の意味を抽象的計算機の動作（状態遷移）として記述する

抽象機械の例

- 有限状態機械（オートマトン）
- チューリングマシン
- ランダムアクセス機械
- ラムダ計算

プログラミング言語の意味定義には不適当

- 構造操作的意味論 (Plotkin)

操作的意味の定義

計算はコードeと実行状況σの組　$\langle e, \boldsymbol{\sigma} \rangle$

遷移関係を計算のステップとする　$c \longrightarrow c'$

遷移関係を推論規則により定義

$$\frac{c \longrightarrow c'}{f(c) \longrightarrow f(c')}$$

操作的意味

数式の操作的意味　$e ::= m \mid e_0 + e_1$　(mは数字)

遷移関係　$e \longrightarrow e'$

推論規則

(Rule 1)
$$\frac{e_0 \longrightarrow e_0'}{e_0 + e_1 \longrightarrow e_0' + e_1}$$

(Rule 2)
$$\frac{e_1 \longrightarrow e_1'}{m_0 + e_1 \longrightarrow m_0 + e_1'}$$

(Rule 3) $m_0 + m_1 \longrightarrow m_2$
if m_2 is the sum of m_0 and m_1

操作的意味の例

例: $(1+(2+3))+(4+5) \longrightarrow (1+5)+(4+5)$

$$\begin{array}{lll} 2+3 & \longrightarrow\ 5 & \text{(By rule 3)} \\ 1+(2+3) & \longrightarrow\ 1+5 & \text{(By rule 2)} \\ (1+(2+3))+(4+5) & \longrightarrow\ (1+5)+(4+5) & \text{(By rule 1)} \end{array}$$

rule 1 $\dfrac{e_0 \longrightarrow e_0'}{e_0 + e_1 \longrightarrow e_0' + e_1}$

rule 2 $\dfrac{e_1 \longrightarrow e_1'}{m_0 + e_1 \longrightarrow m_0 + e_1'}$

rule 3 $m_0 + m_1 \longrightarrow m_2$

if m_2 is the sum of m_0 and m_1

Ichiro Satoh

遷移システム

- 計算状況 (Configuration) をΓとする:
  Γ={e, σ}
- 遷移システム (Transition System)
  →⊆Γ×Γとするとき、⟨Γ、→⟩を遷移システムといい
  (γ, γ') ∈ → のとき、γ→γ' と書く
- 遷移システム⟨Γ、→⟩においてγ→γ' となるγ'が存在しないとき、
  Γを終了状況と呼ぶ
- 遷移システム⟨Γ、→⟩は次の条件を満足するときにチャーチ・ロッサー性あるいは合流性を持つという。
  任意のγ ∈Γに対してγ→* γ_1 及びγ→* γ_2 であれば、あるγ'が存在し、γ_1 →* γ'及びγ_2 →* γ'となる

Ichiro Satoh

操作的意味論 (数式)

数式の構文

$$e ::= m \mid v \mid (e + e') \mid (e - e') \mid (e \times e')$$

Ichiro Satoh

操作的意味論 (数式)

計算状況 $\Gamma = \{\langle e, \sigma \rangle\}$ 　遷移関係 $\langle e, \sigma \rangle \longrightarrow \langle e', \sigma \rangle$

足し算の操作的意味の定義 (他の演算も同様)

$$\frac{\langle e_0, \sigma \rangle \longrightarrow \langle e_0', \sigma \rangle}{\langle e_0 + e_1, \sigma \rangle \longrightarrow \langle e_0' + e_1, \sigma \rangle}$$

$$\frac{\langle e_1, \sigma \rangle \longrightarrow \langle e_1', \sigma \rangle}{\langle m + e_1, \sigma \rangle \longrightarrow \langle m + e_1', \sigma \rangle}$$

$$\langle m + m', \sigma \rangle \longrightarrow \langle n, \sigma \rangle \qquad (\text{where } n = m + m')$$

$$\langle v, \sigma \rangle \longrightarrow \langle \sigma(v), \sigma \rangle$$

Ichiro Satoh

操作的意味論 (論理式)

構文: $b ::= t \mid b \text{ or } b' \mid e = e' \mid {\sim}b$

操作的意味 (or-式)

1. $\dfrac{\langle b_0, \sigma\rangle \longrightarrow \langle b_0', \sigma\rangle}{\langle b_0 \text{ or } b_1, \sigma\rangle \longrightarrow \langle b_0' \text{ or } b_1, \sigma\rangle}$
2. $\dfrac{\langle b_1, \sigma\rangle \longrightarrow \langle b_1', \sigma\rangle}{\langle b_0 \text{ or } b_1, \sigma\rangle \longrightarrow \langle b_0 \text{ or } b_1', \sigma\rangle}$
3. $\langle \mathrm{tt} \text{ or } b_1, \sigma\rangle \longrightarrow \langle \mathrm{tt}, \sigma\rangle$
4. $\langle b_0 \text{ or } \mathrm{tt}, \sigma\rangle \longrightarrow \langle \mathrm{tt}, \sigma\rangle$
5. $\langle \mathrm{ff} \text{ or } b_1, \sigma\rangle \longrightarrow \langle b_1, \sigma\rangle$
6. $\langle b_0 \text{ or } \mathrm{ff}, \sigma\rangle \longrightarrow \langle b_0, \sigma\rangle$

操作的意味論 (論理式)

構文: $b ::= t \mid b \text{ or } b' \mid e = e' \mid {\sim}b$

操作的意味 (等号)

1. $\dfrac{e_0 \longrightarrow e_0'}{e_0 = e_1 \longrightarrow e_0' = e_1}$
2. $\dfrac{e_1 \longrightarrow e_1'}{m = e_1 \longrightarrow m = e_1'}$
3. $m = n \longrightarrow t$
   (where t is tt if $m = n$ and ff otherwise)

操作的意味論 (論理式)

構文: $b ::= t \mid b \text{ or } b' \mid e = e' \mid {\sim}b$

操作的意味 (否定)

1. $\dfrac{b \longrightarrow b'}{{\sim}b \longrightarrow {\sim}b'}$
2. ${\sim}t \longrightarrow t'$ $\quad (\mathit{where}\ t' = \neg t)$

操作的意味論 (コマンド)

構文: $c ::= \mathbf{nil} \mid v := e \mid c;\ c'$

操作的意味 $\Gamma = \{\langle c, \sigma\rangle\} \cup \{\sigma\}$

Nil: $\langle \mathbf{nil}, \sigma\rangle \longrightarrow \sigma$

Assignment:

1. $\dfrac{\langle e, \sigma\rangle \longrightarrow^{*} \langle m, \sigma\rangle}{\langle v := e, \sigma\rangle \longrightarrow \sigma[m/v]}$

Composition]:

1. $\dfrac{\langle c_0, \sigma\rangle \longrightarrow \langle c_0', \sigma'\rangle}{\langle c_0;\ c_1, \sigma\rangle \longrightarrow \langle c_0';\ c_1, \sigma'\rangle}$ 2. $\dfrac{\langle c_0, \sigma\rangle \longrightarrow \sigma'}{\langle c_0;\ c_1, \sigma\rangle \longrightarrow \langle c_1, \sigma'\rangle}$

簡易言語

構文:

$$c ::= \mathbf{nil} \mid v := e \mid c; c' \mid \mathbf{if}\ b\ \mathbf{then}\ c\ \mathbf{else}\ c' \mid \mathbf{while}\ b\ \mathbf{c}$$

Ichiro Satoh

操作的意味論 (簡易言語)

簡易言語の意味

$$\langle \mathbf{nil}, \sigma \rangle \longrightarrow \sigma$$

$$\frac{\langle c_0, \sigma \rangle \longrightarrow \langle c_0', \sigma' \rangle}{\langle c_0;\ c_1, \sigma \rangle \longrightarrow \langle c_0';\ c_1, \sigma' \rangle} \qquad \frac{\langle e, \sigma \rangle \longrightarrow^* \langle m, \sigma \rangle}{\langle v := e, \sigma \rangle \longrightarrow \sigma[m/v]}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow^* \langle \mathrm{tt}, \sigma \rangle}{\langle \mathbf{if}\ b\ \mathbf{then}\ c\ \mathbf{else}\ c', \sigma \rangle \longrightarrow \langle c, \sigma \rangle}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow^* \langle \mathrm{tt}, \sigma \rangle}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma \rangle \longrightarrow \langle c;\ \mathbf{while}\ b\ \mathbf{do}\ c, \sigma \rangle}$$

$$\frac{\langle b, \sigma \rangle \longrightarrow^* \langle \mathrm{ff}, \sigma \rangle}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma \rangle \longrightarrow \sigma}$$

Ichiro Satoh

静的意味と動的意味

静的意味

- 実行前に定められる意味
  - 型チェック、変数領域

動的意味

- 実行中に定まる意味
  - 制御フロー、変数代入

Ichiro Satoh

型付き簡易言語

構文

Expressions: $e \in \mathrm{Exp}$ where:

$$e ::= m \mid t \mid v \mid e_0\ bop\ e_1 \mid \sim e$$

Commands: $c \in Com$ where:

$$c ::= \mathbf{nil} \mid v := e \mid c_0;\ c_1 \mid \mathbf{if}\ e\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1 \mid \mathbf{while}\ e\ \mathbf{do}\ c$$

$$bop \in \mathrm{BOp} = \{+, -, *, =, \mathrm{or}\}$$

Ichiro Satoh

型付き簡易言語

型 Types: $\tau \in \text{Types} = \{\text{int}, \text{bool}\}$

$e : \tau \quad \equiv \quad e$ has type τ

型推論 (静的意味論)

Truthvalues: $t : \text{bool}$

Numbers: $m : \text{int}$

Variables: $v : \text{int}$

BinaryOperations: $\dfrac{e_0 : \tau_0 \quad e_1 : \tau_1}{e_0 \text{ bop } e_1 : \tau_2}$

Negation: $\dfrac{e : \text{bool}}{\sim e : \text{bool}}$

+, −, *	bool	int
bool	?	?
int	?	int

=	bool	int
bool	?	?
int	?	bool

or	bool	int
bool	bool	?
int	?	?

型付き簡易言語

型推論 $\text{Wfc}(c) \equiv c$ is a well-formed command.

Nil: $\text{Wfc}(\textbf{nil})$

Assignment: $\dfrac{e : \text{int}}{\text{Wfc}(v := e)}$

Sequencing: $\dfrac{\text{Wfc}(c_0) \quad \text{Wfc}(c_1)}{\text{Wfc}(c_0; c_1)}$

Conditional: $\dfrac{e : \text{bool} \quad \text{Wfc}(c_0) \quad \text{Wfc}(c_1)}{\text{Wfc}\ (\textbf{if } e \textbf{ then } c_0 \textbf{ else } c_1)}$

While: $\dfrac{e : \text{bool} \quad \text{Wfc}(c)}{\text{Wfc}\ (\textbf{while } e \textbf{ do } c)}$

自然意味論

環境Eのもとで式Mが結果vを計算する：

$$E \triangleright M \Downarrow v$$

ここでEは変数名の集合から値への集合への関数、vは値の集合

自然意味論

Natural Semantics

$$\frac{(S_1, m_1) \Downarrow m' \quad (S_2, m_2) \Downarrow m''}{(S_1; S_2, m) \Downarrow m''} \qquad (v := e, m) \Downarrow m\{v \to \langle e \rangle m\}$$

$$\frac{\langle B \rangle m = \text{true} \quad (C_1, m) \Downarrow m'}{(\textbf{if } B \textbf{ then } C_1 \textbf{ else } C_2, m) \Downarrow m'} \qquad \frac{\langle B \rangle m = \text{false} \quad (C_2, m) \Downarrow m'}{(\textbf{if } B \textbf{ then } C_1 \textbf{ else } C_2, m) \Downarrow m''}$$

$$\frac{\langle B \rangle m = \text{false}}{(\textbf{while } B \textbf{ do } C \textbf{ od}, m) \Downarrow m''}$$

$$\frac{\langle B \rangle m = \text{false} \quad (C, m) \Downarrow m' \quad (\textbf{while } B \textbf{ do } C \textbf{ od}, m') \Downarrow m''}{(\textbf{while } B \textbf{ do } C \textbf{ od}, m) \Downarrow m''}$$

λ計算の変数束縛

β変換を行うことなく意味を定義するには
→ 自由変数の束縛過程を示す

$$\langle x\rangle\eta = \eta(x)$$

$$\langle(\lambda x.M)N\rangle\mu = \langle M\rangle\eta\{x:\langle N\rangle\eta\}$$

適用順評価

- λ式Mの計算とは自由変数の値を与える環境のもとでMの値を求めること
- 関数適用(λx.M)Nは引数Nの値を求めて、関数Mの値を求めること
- λx.Mの評価はMの評価はせずに現在の環境とλx.Mの組を保存すること

Ichiro Satoh

λ計算の自然意味論

自然意味論 (Natural Sementics)
環境Eのもとでラムダ式Mが結果vを計算する: $E \rhd M \Downarrow v$

$$E \rhd c \Downarrow c \qquad\qquad E \rhd x \Downarrow v \quad (E(x)=v)$$

$$E \rhd \lambda x.M \Downarrow cls(\lambda x.M, E)$$

$$\frac{E \rhd M_2 \Downarrow v' \quad E \rhd M_1 \Downarrow f}{E \rhd M_1 M_2 \Downarrow v} \; (f(v') = v)$$

$$\frac{E \rhd M_1 \Downarrow cls(\lambda x.M'_1, E_1) \quad E \rhd M_2 \Downarrow v_2 \quad E_1\{x{:}v_2\} \rhd M'_1 \Downarrow v}{E \rhd M_1 M_2 \Downarrow v}$$

cls(λx.M,E)は関数閉包:
引数xを受け取り、自由変数の値を与える環境EのもとでMの値を計算する

Ichiro Satoh

引数の渡し方

関数/手続き呼び出し/遠隔手続き呼び出しの引数の渡し方

call-by-value:
　値だけ渡す

call-by-name:
　変数名だけ渡す、実行時にバインド

call-by-reference:
　参照 (ポインタ) だけ渡す

call-by-copy:
　値のコピーを生成し、そのコピーを渡す

Ichiro Satoh

参考文献 (カテゴリ)

R. L. Crole, "Categories for Types", Cambridge University Press, 1993.

B. Jacobs, "Categorical Logic and Type Theory", Elsevier, 1999.

F. Borceux, "Handbook of Categorical Algebra", Cambridge Univ. Press, 1994.

M. Barr and C. Wells, "Category Theory for Computing Science", Prentice-Hall

Ichiro Satoh