計算モデル
プログラミング意味論

国立情報学研究所

佐藤一郎
E-mail: ichiro@nii.ac.jp

http://research.nii.ac.jp/~ichiro

プログラムと計算

プログラムは計算の手順
　プログラムの基本要素：変数代入（読み書き）、条件分岐、繰り返し

プログラムで計算機を定義・再現できる
- 万能チューリングマシン
- 仮想機械(Java、CLR)など

プログラムは計算機と同じ計算能力がある

プログラミング言語

構文(syntax)
- プログラミング言語の文法、表現

意味論(semantics)
- プログラムがどのような動作・結果を持つかを定式化

語用論(pragmatics)
- プログラムの書き方

プログラミング言語の意味

プログラミング言語の意味の定義方法
- 計算機のハードウェア動作として理解する
- 日本語や英語などの自然言語により意味を記述する
- 既知のプログラミング言語によって類似の文・式の意味を記述する

厳密で矛盾なく意味を定義するには
　形式的意味論(formal semantics)
- 計算機とは独立して厳密に規定できること
- 定義に曖昧さがないこと
- プログラムの推論・解析に理論的基礎を提供すること

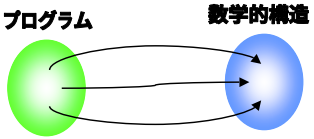
意味論

プログラミング言語意味論

- 操作意味論(Operational Semantics)
- 表示意味論(Denotational Semantics)
- 公理意味論(Axiomatic Semantics)

表示的意味論

集合や関数などの数学的概念を用いて構文上の対象の意味を記述



表示意味論

構文構成要素ごとに意味を定義

S ::= if E then S else S
| while E do S od
| S;S
| E:=E

⟨⟩ c: M →(M→M)　　ここでcは環境(変数の中身を定義)
⟨if E then S1 else S2⟩ c = ⟨S1⟩ c (⟨E⟩ c=trueのとき)
= ⟨S2⟩ c (⟨E⟩ c=falseのとき)
⟨while E do S od⟩ c = ⟨while E do S od⟩ c
(⟨S⟩ c) (⟨E⟩ c=trueのとき)
= ⟨⟩
⟨S1;S2⟩ c = ⟨S2⟩ c(⟨S1⟩ c)
⟨E1:=E2⟩ c = c[⟨E2⟩ c/ ⟨E1⟩ c]

公理的意味論

プログラミング言語の意味を公理と推論規則によって定めようとするもの

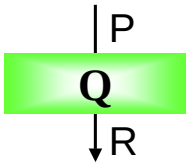
- Hoareモデル
- 時相論理による意味定義他

Hoare論理

C.A.R. Hoare
プログラムの事前条件と事後条件を論理式により定式化

P{Q}R

プログラムQを実行直前に論理式Pが成立して、
もしQの実行が終了すれば、その直後に論理式Rが成立する



Ichiro Satoh

Hoare論理

Hoare論理によるプログラミング意味論

$$c ::= \texttt{null} \mid v = e \mid c_1 \,;\, c_2 \mid \texttt{if}(e)\, c_1\, \texttt{else}\, c_2 \mid \texttt{while}(e)\, c$$

null $\dfrac{P \supset Q}{P\,\{\texttt{null}\}\,Q}$

assignment $\dfrac{P \supset Q\left[{}^{e}/_{x}\right]}{P\,\{x = e\}\,Q}$

sequence $\dfrac{P\,\{c_1\}\,R \qquad R\,\{c_2\}\,Q}{P\,\{c_1\,;\,c_2\}\,Q}$

if-statement $\dfrac{P \wedge e\,\{c_1\}\,Q \qquad P \wedge \neg e\,\{c_2\}\,Q}{P\,\{\texttt{if}(e)\, c_1\, \texttt{else}\, c_2\}\,Q}$

while-statement $\dfrac{P \supset L \qquad L \wedge e\,\{c\}\,L \qquad L \wedge \neg e \supset Q}{P\,\{\texttt{while}(e)\, c\}\,Q}$

Ichiro Satoh

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)

Ichiro Satoh

操作的意味の定義

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

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

Ichiro Satoh

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

$2 + 3$	$\longrightarrow 5$	(By rule 3)
$1 + (2 + 3)$	$\longrightarrow 1 + 5$	(By rule 2)
$(1 + (2 + 3)) + (4 + 5)$	$\longrightarrow (1 + 5) + (4 + 5)$	(By rule 1)

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

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

操作的意味論 (数式)

数式の構文

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

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. $\langle \text{tt or } b_1, \sigma \rangle \longrightarrow \langle \text{tt}, \sigma \rangle$
4. $\langle b_0 \text{ or tt}, \sigma \rangle \longrightarrow \langle \text{tt}, \sigma \rangle$
5. $\langle \text{ff or } b_1, \sigma \rangle \longrightarrow \langle b_1, \sigma \rangle$
6. $\langle b_0 \text{ or ff}, \sigma \rangle \longrightarrow \langle b_0, \sigma \rangle$

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 (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)

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 (否定)

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

Ichiro Satoh

操作的意味論 (コマンド)

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

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

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

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:

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

操作的意味論 (簡易言語)

簡易言語の意味

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

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

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

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

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

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

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:

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

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

Ichiro Satoh

型付き簡易言語

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

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

型推論 (静的意味論)

Truthvalues: $t : \mathrm{bool}$

Numbers: $m : \mathrm{int}$

Variables: $v : \mathrm{int}$

BinaryOperations:
$$\frac{e_0 : \tau_0 \quad e_1 : \tau_1}{e_0\ \mathrm{bop}\ e_1 : \tau_2}$$

Negation:
$$\frac{e : \mathrm{bool}}{\sim e : \mathrm{bool}}$$

+, −, *	bool	int
bool	?	?
int	?	int

=	bool	int
bool	?	?
int	?	bool

or	bool	int
bool	bool	?
int	?	?

Ichiro Satoh

型付き簡易言語

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

Nil: $\mathrm{Wfc}(\mathbf{nil})$

Assignment:
$$\frac{e : \mathrm{int}}{\mathrm{Wfc}(v := e)}$$

Sequencing:
$$\frac{\mathrm{Wfc}(c_0) \quad \mathrm{Wfc}(c_1)}{\mathrm{Wfc}(c_0; c_1)}$$

Conditional:
$$\frac{e : \mathrm{bool} \quad \mathrm{Wfc}(c_0) \quad \mathrm{Wfc}(c_1)}{\mathrm{Wfc}\ (\mathbf{if}\ e\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1)}$$

While:
$$\frac{e : \mathrm{bool} \quad \mathrm{Wfc}(c)}{\mathrm{Wfc}\ (\mathbf{while}\ e\ \mathbf{do}\ c)}$$

Ichiro Satoh

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

Ichiro Satoh

自然意味論

Natural Semantics

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

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

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

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

Ichiro Satoh

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：

$$E \triangleright M \Downarrow v$$

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

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

$$\frac{E\triangleright M_2\Downarrow v' \quad E\triangleright M_1\Downarrow f}{E\triangleright M_1M_2\Downarrow v}\ (f(v')=v)$$

$$\frac{E\triangleright M_1\Downarrow cls(\lambda x.M'_1,E_1) \quad E\triangleright M_2\Downarrow v_2 \quad E_1\{x{:}v_2\}\triangleright M'_1\Downarrow v}{E\triangleright M_1M_2\Downarrow v}$$

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