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

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

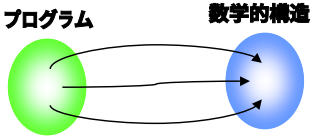
厳密で矛盾なく意味を定義するには
形式的意味論(formal semantics)
- 計算機とは独立して厳密に規定できること
- 定義に曖昧さがないこと
- プログラムの推論・解析に理論的基礎を提供すること

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

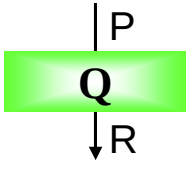
- Hoareモデル
- 時相論理による意味定義他

Hoare論理

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



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

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

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

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

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

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)

操作的意味論

数式の操作的意味

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

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

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

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

操作的意味論

簡易言語の意味

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

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

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

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

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