

7 Ноября 2013

Инсерционное моделирование 1

Лекция 11

Семантика и верификация структурных императивных программ

Структурные программы

(программные фрагменты без описаний и ввода/вывода)

Базовые операторы

$x := y$

$(x_1 := y_1, \dots, x_n := y_n)$

- Именуемые выражения (константные атрибутивные выражения)
- Алгебраические выражения (арифметические, булевские, ...)
- Вызовы функций в алгебраических выражениях
- Типы данных, многосортные алгебраические системы

Основные композиции (синтаксические конструкции)

$(P; Q), if(u, P, Q), while(u, P)$

синтаксис

семантика

Денотационная семантика (что)

Операционная семантика (как)

Денотационная семантика

R – множество имен (константных атрибутивных термов)

D – область значений (всех типов)

$S = D^R = R \rightarrow D$ состояния памяти

$[[P]]: S \rightarrow S = (R \rightarrow D) \rightarrow (R \rightarrow D)$ смысл программы

частичные функции \perp

$$[[P]](s) = P(s)$$

аналогично для выражений

семантика
выражений

$[[y]]_D : S \rightarrow D$

алгебраические выражения

$[[x]]_R : S \rightarrow R$

именующие выражения

$[[u]]_C : S \rightarrow \{0,1\}$ условия

$[[y]]_D(s) = y_D(s)$ значение алгебраического выражения

$[[x]]_R(s) = x_R(s)$ значение именующего выражения

$[[u]]_C(s) = u_C(s)$ значение условия

Рекурсивное определение $[[P]]$ (присваивание)

$$[[P]](s) = P(s) = P_s$$

$$P = (f_1(x_1) := t_1, f_2(x_2) := t_2, \dots) \Rightarrow P_s = s',$$

$$s'(f_1((x_1)_D s)) = (t_1)_D s,$$

$$s'(f_2((x_2)_D s)) = (t_2)_D s,$$

.....

$$r \notin \{f_1((x_1)_D s), f_2((x_2)_D s), \dots\} \Rightarrow s'(r) = s(r)$$

Рекурсивное определение $[[P]]$ (композиции и цикл)

$$(P; Q)s = Q(Ps)$$

$$\text{if } (u, P, Q)s = \mathbf{if} (u_c(s)) \mathbf{then} (Ps) \mathbf{else} (Qs)$$

$$\text{while}(u, P) = \text{if } (u, (P; \text{while}(u, P)), \varepsilon)$$

$$\varepsilon s = s$$

$$\text{while}(u, P)s = P^n s, \perp$$

n – наименьшее целое ≥ 0 такое, что

$$\forall k ((k < n) \rightarrow (P^k s \neq \perp, \alpha(P^k s)), \\ (\neg \alpha(P^n s), P^n s \neq \perp))$$

Действия

- проверка условий, формулы базового языка
- присваивания

Переходы

$$s \xrightarrow{check(\alpha)} s, \alpha(s) = 1$$

$$s \xrightarrow{y} y(s)$$

$$\alpha_C(s) = 1$$

$$s[if(\alpha)then(P)else(Q)] \rightarrow s[P]$$

$$\alpha_C(s) = 1$$

$$s[while(u, P)] \rightarrow s[P; while(u, P)]$$

$$\alpha_C(s) = 0$$

$$s[if(\alpha)then(P)else(Q)] \rightarrow s[Q]$$

$$\alpha_C(s) = 0$$

$$s[while(u, P)] \rightarrow s[\Delta]$$

Операционная семантика

структурных императивных программ

Конкретная атрибутивная среда

Агенты – структурные программы

Последовательная

Композиция: лекция 7

Теоремы

1. Система $s[P]$ ординарная детерминированная (из каждого состояния возможен только один переход)

2. $P(s) = s' \Leftrightarrow s[P] \xrightarrow{*} s'[\Delta]$
 \Rightarrow Индукция по длине программы и числу повторений `while` – циклов
 \Leftarrow Индукция по длине истории

Лемма 1. $s[P] \xrightarrow{*} s'[\Delta] \Rightarrow s[P; Q] \xrightarrow{*} s'[Q]$

Лемма 2. $while(u, Q)(s)$ определено $\Leftrightarrow \exists m (while(u, Q)(s) = Q^m(s))$

3. $P(s)$ определено $\Leftrightarrow s[P] \xrightarrow{*} (P(s))[\Delta]$

Упражнение

- Доказать теоремы

Логика Хора-Флойда

Формулы:

$$\alpha \rightarrow [P]\beta$$

$$\alpha \rightarrow \langle P \rangle \beta$$

предусловие

постусловие

программа

Частичная корректность программ

(если предусловие и программа останавливается, то постусловие)

$$\{\alpha\}P\{\beta\}$$

Полная корректность программ

(если предусловие, то программа останавливается и постусловие)

$$\alpha \rightarrow \langle P \rangle \beta \Leftrightarrow$$

$$\alpha \rightarrow \langle P \rangle 1 \wedge$$

$$\alpha \rightarrow [P]\beta$$

*Все это формулы
темпоральной динамической
логики*

Hoare 1969 структурные программы,
Floyd 1967 программы с goto

Денотационная семантика Хоаровских троек

$$[[\alpha \rightarrow [P]\beta]]: D^R \rightarrow \{0,1\}$$

$$[[\alpha \rightarrow [P]\beta]](s) =$$

$$= \alpha(s) \wedge (P(s) \neq \perp) \rightarrow \beta(P(s))$$

$$P(s) \models \beta$$

$$[[\alpha \rightarrow \langle P \rangle \beta]](s) =$$

$$= \alpha(s) \rightarrow (P(s) \neq \perp) \wedge \beta(P(s))$$

Исчисление Хоара

$$\frac{\alpha \rightarrow [P]\gamma, \gamma \rightarrow [Q]\beta}{\alpha \rightarrow [PQ]\beta}$$

$$\frac{\alpha \wedge \gamma \rightarrow [P]\beta, \alpha \wedge \bar{\gamma} \rightarrow [Q]\beta}{\alpha \rightarrow \text{invariant}(\gamma, P, Q)]\beta}$$

$$\frac{\alpha \rightarrow \delta, \gamma \wedge \delta \rightarrow [P]\delta, \delta \wedge \bar{\gamma} \rightarrow \beta}{\alpha \rightarrow \text{while}(\gamma, P)]\beta}$$
$$\alpha \rightarrow \beta(t_1, t_2, \dots)$$

$$\alpha \rightarrow [(x_1 := t_1, x_2 := t_2, \dots)]\beta(x_1, x_2, \dots)$$

Интерпретируется на семантической модели императивного структурного программирования

Для доказательства частичной корректности нужно найти инварианты циклов

инвариант цикла

```
s:=0;
for j:=1 .. n do(
  s:=s+a(j)
);
```

Пример

```
s:=0;
j:=1;
while (j<=n,
  s:=s+a(j);
  j:=j+1
);
```

$$\frac{\alpha \rightarrow [P]\gamma, \gamma \rightarrow [Q]\beta}{\alpha \rightarrow [PQ]\beta}$$

$$\frac{\alpha \wedge \gamma \rightarrow [P]\beta, \alpha \wedge \bar{\gamma} \rightarrow [Q]\beta}{\alpha \rightarrow [\text{if}(\gamma, P, Q)]\beta}$$

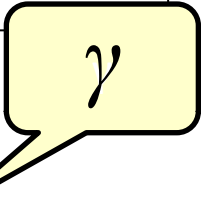
$$\frac{\alpha \rightarrow \delta, \delta \wedge \bar{\gamma} \rightarrow [P]\delta, \delta \wedge \gamma \rightarrow \beta}{\alpha \rightarrow [\text{while}(\gamma, P)]\beta}$$

$$\frac{\alpha \rightarrow \beta(t_1, t_2, \dots)}{\alpha \rightarrow [(x_1 := t_1, x_2 := t_2, \dots)]\beta(x_1, x_2, \dots)}$$

Спецификация:

$$1 \rightarrow \langle P \rangle \beta$$

$$\beta = (s = \sum_{k=1}^n a[k])$$



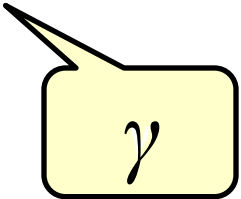
$$1 \rightarrow [P]\beta \Leftrightarrow$$

$$[s := 0; P']\beta$$

$$[s := 0](s = 0), (s = 0) \rightarrow [j := 1; P'']\beta$$

$$s = 0 \rightarrow [j := 1](s = 0 \wedge j = 1),$$

$$(s = 0 \wedge j = 1) \rightarrow \text{while}(j \leq n, Q)\beta$$



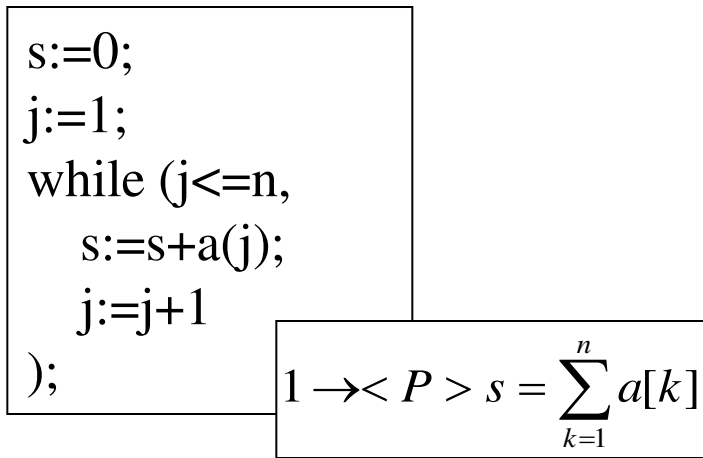
$$\alpha = (s = 0 \wedge j = 1)$$

Инвариант цикла

$$\delta = (s = \sum_{k=1}^{j-1} a[k])$$

$$\alpha \rightarrow \delta, \delta \wedge j \leq n \rightarrow [Q]\delta, \delta \wedge j > n \rightarrow \beta$$

Не получается



Настоящий инвариант

$$\delta = (s = \sum_{k=1}^{j-1} a[k]) \wedge (j \leq n + 1)$$

$$\alpha \rightarrow \delta, \delta \wedge j \leq n \rightarrow [Q]\delta, \delta \wedge j > n \rightarrow \beta$$

```

s:=0;
j:=1;
while (j<=n,
  s:=s+a(j);
  j:=j+1
);

```

$$1 \rightarrow \langle P \rangle s = \sum_{k=1}^n a[k]$$

$$\delta = (s = \sum_{k=1}^{j-1} a[k]) \wedge (j \leq n+1)$$

$$\alpha \rightarrow \delta$$

$$s = 0 \wedge j = 1 \rightarrow (s = \sum_{k=1}^{j-1} a[k]) \wedge (j \leq n+1)$$

$$\delta \wedge j \leq n \rightarrow [Q]\delta$$

$$(s = \sum_{k=1}^{j-1} a[k]) \wedge (j \leq n+1) \wedge (j \leq n) \rightarrow [$$

$$s := s + a[j];$$

$$j := j + 1$$

$$](s = \sum_{k=1}^{j-1} a[k]) \wedge (j \leq n+1)$$

$$\delta \wedge j > n \rightarrow \beta$$

$$(s = \sum_{k=1}^{j-1} a[k]) \wedge (j \leq n+1) \wedge j > n \rightarrow (s = \sum_{k=1}^n a[k])$$

Обоснования и применения

Правила логики Хоара \Leftrightarrow высказывания динамической логики

$$\frac{\alpha \rightarrow [P]\gamma, \gamma \rightarrow [Q]\beta}{\alpha \rightarrow [PQ]\beta} \quad (\alpha \rightarrow [P]\gamma) \wedge (\gamma \rightarrow [Q]\beta) \rightarrow (\alpha \rightarrow [PQ]\beta)$$

Пропозициональная динамическая логика

**Применяя правила в обратном порядке,
Хоаровскую тройку можно редуцировать к
формуле логики первого порядка!**

**Полнота доказана для целочисленной арифметики
(с использованием Геделевской нумерации)**

Задача

Построить программу поиска минимального и максимального
элементов
одномерного массива
Специфицировать
Доказать правильность методом Хоара