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

$$x := y$$

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

синтаксис

–Именующие выражения (константные атрибутные выражения)
–Алгебраические выражения (арифметические, булевские, …)
–Вызовы функций в алгебраических выражениях
–Типы данных, многосортные алгебраические системы

Основные композиции (синтаксические конструкции)

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

семантика

Денотационная семантика (что)

Операционная семантика (как)

Денотационная семантика

R – **множество имен (константных атрибутных термов)**

D – **область значений (всех типов)**

$$S = D^R = R \to D$$ **состояния памяти**

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

частичные функции $\perp$

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

аналогично для выражений

семантика выражений

$$[[y]]_D : S \to D$$ **алгебраические выражения**

$$[[x]]_R : S \to R$$ **именующие выражения**

$$[[u]]_C : S \to \{0,1\}$$ **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)

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

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

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

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

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

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

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

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

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

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

$$\varepsilon s = s$$

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

n – наименьшее целое ≥ 0 такое, что

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

Операционная семантика

структурных императивных программ

Действия

– проверка условий, формулы базового языка

– присваивания

Переходы

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

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

Конкретная атрибутная среда

Агенты – структурные программы

$$\frac{\alpha_C(s) = 1}{s[if\,(\alpha)then(P)else(Q)] \to s[P]}$$

$$\frac{\alpha_C(s) = 1}{s[while(u, P)] \to s[P; while(u, P)]}$$

Последовательная Композиция: лекция 7

$$\frac{\alpha_C(s) = 0}{s[if\,(\alpha)then(P)else(Q)] \to s[Q]}$$

$$\frac{\alpha_C(s) = 0}{s[while(u, P)] \to s[\Delta]}$$

Теоремы

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

2. $P(s) = s' \Leftrightarrow s[P] \xrightarrow{*} s'[\Delta]$

$\Longrightarrow$ Индукция по длине программы и числу повторений while – циклов

$\Longleftarrow$ 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:

$$\alpha \to [P]\beta$$

$$\alpha \to < P > \beta$$

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)

$$\alpha \to < P > \beta \Leftrightarrow$$

$$\alpha \to < P > 1 \wedge$$

$$\alpha \to [P]\beta$$

Все это формулы темпоральной динамической логики

Hoare 1969 структурные программы,
Floyd 1967 программы с goto

Денотационная семантика Хоаровских троек

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

$$[[\alpha \to [P]\beta]](s) =$$
$$= \alpha(s) \wedge (P(s) \neq \perp) \to \beta(P(s))$$

$$\boxed{P(s) \models \beta}$$

$$[[\alpha \to < P > \beta]](s) =$$
$$= \alpha(s) \to (P(s) \neq \perp) \wedge \beta(P(s))$$

Исчисление Хоара

Интерпретируется на семантической модели императивного структурного программирования

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

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

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

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

Для доказательства частичной корректности нужно найти инварианты циклов

инвариант цикла

Пример

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

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

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

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

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

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

Спецификация:

$$1 \to < P > \beta$$

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

Инвариант цикла

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

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

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

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

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

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

$$\gamma$$

$$\gamma$$

$$\alpha \to \delta, \delta \wedge j <= n \to [Q]\delta, \delta \wedge j > n \to \beta$$

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

Не получается

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

$$1 \to < P > s = \sum_{k=1}^{n} a[k]$$

Настоящий инвариант

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

$$\alpha \to \delta, \delta \wedge j <= n \to [Q]\delta, \delta \wedge j > n \to \beta$$

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

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

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

$$\alpha \rightarrow \delta$$

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

$$\delta \wedge j <= n \rightarrow [Q]\delta$$

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

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

$$j := j + 1$$

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

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

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

Обоснования и применения

Правила логики Хоара <=> высказывания динамической логики

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

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