

МЕТОДЫ ДОКАЗАТЕЛЬСТВА СВОЙСТВ ПРОГРАММ

Рассмотрим три основных метода, базирующиеся на понятии фиксированных точек:

- 1) Метод доказательства, основанный на свойстве фиксированных точек.
- 2) Метод численной индукции.
- 3) Метод индукции фиксированной точки.

МЕТОД ИНДУКЦИИ ФИКСИРОВАННОЙ ТОЧКИ

Являясь столь же мощным, как и метод численной индукции, этот метод часто упрощает процесс доказательства.

Суть метода: Если $q(x)$ – утверждение, такое что:

- 1) $q(\perp)$ справедливо;
- 2) из справедливости $q(x)$ следует справедливость $q(H(x))$, то $q(\text{fix } H)$ также справедливо.

Утверждение q часто может быть равенством, либо отношением $\sqsubseteq$, а $q(\text{fix } H)$ – требующая доказательства теорема.

Индукция фиксированной точки часто используется при доказательстве методом, основанным на свойствах фиксированных точек.

ПРИМЕР ДОКАЗАТЕЛЬСТВА

Рассмотрим в качестве примера использования индукции фиксированной точки другое доказательство **теоремы 5**. Если **f** – строгая функция и **$f \cdot h = g \cdot f$** , то **$\text{fix } g = f(\text{fix } h)$** .

Для доказательства **теоремы 5** покажем, что **$\text{fix } g \sqsubseteq f(\text{fix } h)$** и **$f(\text{fix } h) \sqsubseteq \text{fix } g$** , откуда будет следовать искомый результат.

В одну сторону.

$$g(f(\text{fix } h)) = (g \cdot f)(\text{fix } h) = (f \cdot h)(\text{fix } h) = f(h(\text{fix } h)) = f(\text{fix } h).$$

Воспользуемся допущением теоремы о том, что **$g \cdot f = f \cdot h$** , а также тем, что **$\text{fix } h$** – фиксированная точка **h** , а следовательно **$h(\text{fix } h) = \text{fix } h$** .

Таким образом, получаем, что **$f(\text{fix } h)$** – фиксированная точка **g** , откуда, по свойству минимальной фиксированной точки, следует, что **$\text{fix } g \sqsubseteq f(\text{fix } h)$** .

ПРИМЕР ДОКАЗАТЕЛЬСТВА

В другую сторону. Искомым утверждением является утверждение $f(\text{fix } h) \sqsubseteq \text{fix } g$. В качестве утверждения $q(x)$ возьмем: $f(x) \sqsubseteq \text{fix } g$.

Поскольку f – строгая функция, то $f(\perp) = \perp$, а следовательно $f(\perp) \sqsubseteq \text{fix } g$, откуда следует справедливость $q(\perp)$.

Предположим, что $q(x)$ справедливо для некоторого x , т.е. $f(x) \sqsubseteq \text{fix } g$ и докажем, что $q(h(x))$ также справедливо.

Из допущения теоремы имеем: $f(h(x)) = g(f(x))$.

ПРИМЕР ДОКАЗАТЕЛЬСТВА

Согласно индуктивной гипотезе и в силу монотонности g , справедливо утверждение $g(f(x)) \sqsubseteq g(\text{fix } g) = \text{fix } g$, откуда следует, что $f(h(x)) \sqsubseteq \text{fix } g$.

Таким образом, $q(h(x))$ – справедливо.

По индукции фиксированной точки следует справедливость утверждения $q(\text{fix } h)$, т.е. $f(\text{fix } h) \sqsubseteq \text{fix } g$.

Учитывая ранее доказанное утверждение $\text{fix } g \sqsubseteq f(\text{fix } h)$, получаем: $f(\text{fix } h) = \text{fix } g$.

ПРИМЕР ДОКАЗАТЕЛЬСТВА

Доказательство **теоремы 5** можно еще упростить.

Поскольку пара $\langle \text{fix } h, \text{fix } g \rangle$ является минимальной фиксированной точкой функции $\lambda(x.y). (h(x), g(y))$, то можно использовать индукцию фиксированной точки одновременно по двум фиксированным точкам.

Будем искать утверждение $q(x,y)$, такое что:

- 1) $q(\perp, \perp)$ – справедливо;
- 2) из справедливости $q(x,y)$ следует справедливость $q(h(x), g(y))$.

Тогда можно вывести, что $q(\text{fix } h, \text{fix } g)$ также справедливо.

ПРИМЕР ДОКАЗАТЕЛЬСТВА

В качестве утверждения $q(x,y)$ возьмем: $f(x) = y$.

Справедливость $q(\perp, \perp)$ следует из того, что f – строгая функция.

Предположим, что $q(x,y)$ справедливо для некоторых x и y .

По допущению теоремы имеем: $f(h(x)) = g(f(x))$.

Далее, по индуктивной гипотезе, получаем: $g(f(x)) = g(y)$, откуда следует, что $f(h(x)) = g(y)$, а следовательно $q(h(x), g(y))$ справедливо.

По индукции фиксированной точки получаем: $f(\text{fix } h) = \text{fix } g$, что и требовалось доказать.

ТЕХНИКА ИНДУКЦИИ ФИКСИРОВАННОЙ ТОЧКИ

Заметим, что не все утверждения вида $q(x)$ допустимы при доказательствах методом индукции фиксированной точки.

Критерий допустимости утверждений может быть получен с помощью метода численной индукции. Если утверждение $q(\perp)$ справедливо и из справедливости $q(x)$ можем доказать справедливость $q(h(x))$, то, используя метод численной индукции, получим, что $q(h^n(\perp))$ справедливо для всех $n \geq 0$. Желательным утверждением является $q(\sqcup_{n=0}^{\infty} h^n(\perp))$. Поскольку $\{h^n(\perp), n \geq 0\}$ есть цепь, то можно показать, что это утверждение справедливо, если q является замкнутым утверждением.

Под замкнутостью утверждения $q(x)$ понимается сохранение справедливости утверждения для н.в.г. всех цепей X из области значений, т.е. $\forall X \{q(x) \mid x \in X\} \rightarrow q(\sqcup X)$.

Техника индукции фиксированной точки широко применяется в семантических доказательствах.

СВОЙСТВА УСЛОВНОГО ОПЕРАТОРА

Введем определения **коммутативности** и **идемпотентности**.

Определение коммутативности. Две команды γ_0 и γ_1 , коммутативны, если $\gamma_0 \cdot \gamma_1 \equiv \gamma_1 \cdot \gamma_0$.

Команда γ и выражение ω коммутативны, если $\omega \equiv \gamma \cdot \omega$, т.е. выражение имеет то же самое значение до или после выполнения команды.

Два выражения всегда коммутируют, т.к. вычисление одного не может влиять на другое (положение меняется, когда допускается побочный эффект).

Определение идемпотентности. Команда γ идемпотентна, если $\gamma \cdot \gamma \equiv \gamma$.

Таким образом, идемпотентность команды означает, что выполнение ее более, чем один раз, ничего не меняет.

СВОЙСТВА УСЛОВНОГО ОПЕРАТОРА

Пусть **b** обозначает истинностное значение, а греческие буквы обозначают элементы подходящей области (например, $(S \rightarrow S)$ – для условных команд или $(S \rightarrow T)$ – для условных выражений).

Полагаем, что значения всех команд и выражений – строгие функции.

(1) Левая факторизация:

$$\gamma_0 \cdot (b \rightarrow \gamma_1, \gamma_2) = b \rightarrow \gamma_0 \cdot \gamma_1, \gamma_0 \cdot \gamma_2$$

(2) Если γ коммутирует с ω , то

$$(\lambda \sigma. \omega \sigma \rightarrow p \sigma, q \sigma) \cdot \gamma = (\lambda \sigma. \omega \sigma \rightarrow (p \cdot \gamma) \sigma, (q \cdot \gamma) \sigma)$$

$$\begin{aligned} (3) \quad b \rightarrow (b \rightarrow \gamma_0, \gamma_1), (b \rightarrow \gamma_2, \gamma_3) &= b \rightarrow \gamma_0, (b \rightarrow \gamma_2, \gamma_3) = \\ &= b \rightarrow (b \rightarrow \gamma_0, \gamma_1), \gamma_3 = b \rightarrow \gamma_0, \gamma_3 \end{aligned}$$

СВОЙСТВА УСЛОВНОГО ОПЕРАТОРА

(4) Если ω коммутирует с γ_2 и γ_3 , то

$$(\lambda\sigma.\omega\sigma \rightarrow \gamma_0\sigma, \gamma_1\sigma) \cdot (\lambda\sigma.\omega\sigma \rightarrow \gamma_2\sigma, \gamma_3\sigma) = (\lambda\sigma.\omega\sigma \rightarrow (\gamma_0 \cdot \gamma_2)\sigma, (\gamma_1 \cdot \gamma_3)\sigma)$$

(5) Если ω_0 и ω_1 , имеют допустимые подходящие значения на σ_0 , то $\omega_1\sigma_0 \rightarrow (\lambda\sigma.\omega_0\sigma \rightarrow \gamma_0\sigma, \gamma_1\sigma), (\lambda\sigma.\omega_0\sigma \rightarrow \gamma_2\sigma, \gamma_3\sigma) = \omega_0\sigma_0 \rightarrow (\lambda\sigma.\omega_1\sigma \rightarrow \gamma_0\sigma, \gamma_2\sigma), (\lambda\sigma.\omega_1\sigma \rightarrow \gamma_1\sigma, \gamma_3\sigma)$

Все эти свойства могут быть легко доказаны простой подстановкой в левую и правую части всех возможных проверяемых значений.

Правило (3) дает упрощение для вложенных условий; (4) позволяет объединить два условных оператора в один; (5) говорит о том, что вложенные условные операторы могут быть вынесены наружу.

ПОБОЧНЫЙ ЭФФЕКТ

Теперь снимем ранее принятые условия, что вычисление выражения не меняет состояние машины.

Ранее мы сопоставляли выражениям в качестве значений элементы ω семантической области $(S \rightarrow T)$.

Теперь же мы будем полагать, что вычисление выражения в некотором состоянии производит не только результат, но также и состояние машины.

Таким образом, значение ω , сопоставляемое выражению, является элементом области $(S \rightarrow (T \times S))$.

Команды, как и ранее, задают преобразование состояний, т.е. их значениями являются элементы из области $(S \rightarrow S)$.

ПОБОЧНЫЙ ЭФФЕКТ

Семантические функции для выражений языка L имеют функциональное определение $E: \text{Exp} \rightarrow (S \rightarrow (T \times S))$ и задаются следующим образом:

$E[p] = \text{strict } \omega$, где ω – функция из области $(S \rightarrow (T \times S))$, ассоциированная с « p »;

$E[\text{true}] = \text{strict}(\lambda\sigma. \langle \text{true}, \sigma \rangle)$

$E[\text{false}] = \text{strict } \{\lambda\sigma. \langle \text{false}, \sigma \rangle\}$

Заметим, что вычисление констант не дает побочного эффекта.

Для условного выражения и условной команды вначале вычисляется условие в исходном состоянии, а затем вычисляется одна из альтернативных ветвей в состоянии, полученном после вычисления условия. Для описания семантических функций удобно ввести новый комбинатор, обозначаемый звездочкой (*).

ПОБОЧНЫЙ ЭФФЕКТ

Определение. Пусть функции f и g имеют вид $f: \mathcal{E} \rightarrow (S \rightarrow D)$ и $g: S \rightarrow (\mathcal{E} \times S)$. Тогда функция $f * g$ есть элемент области $(S \rightarrow D)$, и для любого $\sigma \in S$ $(f * g)\sigma = (f\beta)\sigma'$, где $\langle \beta, \sigma' \rangle = g\sigma$.

Таким образом, комбинатор $*$ может рассматриваться как взятие результатов применения функции g к σ и подстановку их по одному в качестве аргументов функции f .

Введем следующие соглашения:

- 1) Комбинатор $*$ имеет меньший приоритет, чем композиция функций, т.е. $a(b)*c(d)$ означает то же, что и $(a(b))*c(d)$;
- 2) Комбинатор ассоциируется справа налево, т.е. $(a*b*c)$ означает то же, что и $(a*(b*c))$.

ПОБОЧНЫЙ ЭФФЕКТ

Тогда семантическая функция для условного выражения имеет вид:
 $E[\text{if } E_0 \text{ then } E_1 \text{ else } E_2 \text{ fi}] = (\lambda\beta.\beta \rightarrow E[E_1], E[E_2]) * E[E_0]$.

Если правую часть этого равенства применить к состоянию σ , то мы получим: $(\lambda\beta.\beta \rightarrow E[E_1], E[E_2]) * E[E_0] \sigma$.

Вначале вычисляется $E[E_0] \sigma$. Пусть в результате этого вычисления получается значение $\langle \beta', \sigma' \rangle$.

Далее необходимо вычислять $((\lambda\beta.\beta \rightarrow E[E_1], E[E_2]) \beta') \sigma'$.

После подстановки β' вместо аргументе β получаем $(\beta' \rightarrow E[E_1], E[E_2]) \sigma'$. Таким образом, если β' – подходящее значение, то оно используется для выбора альтернативной ветви, которая вычисляется в состоянии σ' .

Условная команда определяется точно также. Аналогичным образом определяется семантика команды **while E do C od** (и других циклических команд). Остальные пункты определения семантики языка **L** в точности совпадают с ранее описанными.