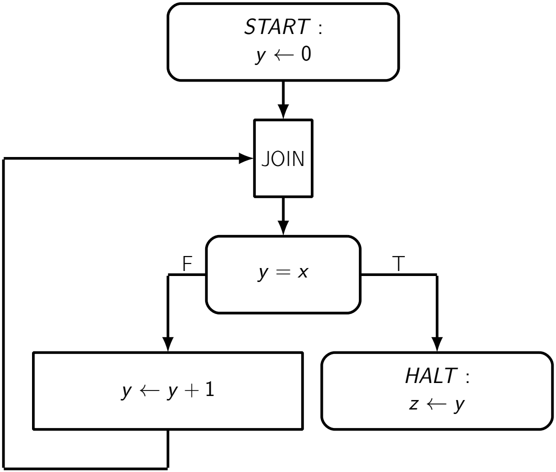
Лекция 2. Метод индуктивных утверждений Флойда

Цель лекции

Определить метод доказательства частичной корректности.
NB: это не поиск «ошибки» в блок-схеме! То есть здесь мы не решаем «задачу достижимости ошибочной конфигурации». Это задача доказательства отсутствия ошибки.

Содержание

Пример для доказательства



$$D_x = \mathbb{Z}$$
$$D_y = \mathbb{Z}$$
$$D_z = \mathbb{Z}$$
$$\varphi(x) \equiv T$$
$$\psi(x, z) \equiv 2 * z \geq x$$

Доказать, что

$\{\varphi\}\ P\ \{\psi\}$.

Поиск доказательства

Нельзя составить $M[P]$ в виде формулы, не прибегая к «знанию принципа работы блок-схемы» (иначе формулу для $M[P]$ можно было бы подставить в определение частичной корректности и свести задачу доказательства частичной корректности к задаче доказательства истинности формулы частичной корректности). Задача составления $M[P]$ алгоритмически неразрешима.

Поиск доказательства

Надо доказать, что во всех конфигурациях на псевдосвязке после HALT, достижимых из START, выполнено $2 * y \geq x$. То есть, что то же выполнено на всех конфигурациях на связке между TEST и HALT, достижимых из START. Туда можно попасть только из связки между JOIN и TEST.

Если у нас было бы множество всех конфигураций для связки JOIN и TEST, достижимых из START (обозначим его C), то доказать частичную корректность значит доказать, что $\forall x \in D_x, y \in D_y \cdot (x,\ y) \in C \wedge y = x \Rightarrow 2 * y \geq x$. Но множество C не всегда можно выразить.

Но может получиться выразить надмножество множества C (обозначим его C'), которого будет достаточно для доказательства частичной корректности: см. следующий слайд.

Поиск доказательства

Пусть B — множество конфигураций, гарантирующих выполнение постусловия (тех, у которых $2 * y \geq x$). Множество C уже было введено на предыдущем слайде. Множество T – все конфигурации, при которых условие в операторе TEST истинно. Частичная корректность — то же, что и $C \cap T \subseteq B$. Это можно доказать так:

1. предложить такое множество конфигураций C', что $C \subseteq C'$ (1) и $C' \cap T \subseteq B$ (2);
2. доказать (1) и (2).

Тогда из $C \subseteq C'$ будет следовать $C \cap T \subseteq C' \cap T$, добавляем $C' \cap T \subseteq B$ и получаем $C \cap T \subseteq B$, т.е. частичную корректность.

Вся хитрость в том, что C может быть невыразимо в виде формул, а C' можно выразить, причем еще и можно доказать (1) и (2).

Поиск доказательства

$C' = \{(x,\ y)\ |x \in D_x,\ y \in D_y\ \cdot\ p(x,\ y)\}$, где для предиката p выполнены такие соотношения:

$$\begin{cases} \forall x \in D_x \cdot p(x,\ 0) \\ \forall x \in D_x,\ y \in D_y \cdot p(x,\ y) \wedge \neg(y = x) \Rightarrow p(x,\ y + 1) \end{cases}$$

Тогда методом математической индукции можно доказать, что во всех конфигурациях на связке между JOIN и TEST выполнено $p(x,\ y)$, то есть, что $C\ \subseteq\ C'$.

И не забываем, что должно быть выполнено $\forall x \in D_x,\ y \in D_y \cdot p(x,\ y) \wedge (y = x) \Rightarrow 2 * y \geq x$. Это докажет $C'\ \cap\ T\ \subseteq\ B$.

Доказательство по индукции

Лемма Пусть $p: \ D_x \times D_y \to \{T, \ F\}$ таков, что выполнены (1) и (2). Тогда на всех конфигурациях на связке между JOIN и TEST, достижимых из START, выполнен предикат p.

$$\begin{cases} \forall x \in D_x \cdot p(x, \ 0) & (1) \\ \forall x \in D_x, \ y \in D_y \cdot p(x, \ y) \wedge \neg(y = x) \Rightarrow p(x, \ y+1) & (2) \end{cases}$$

Доказательство по индукции. Рассмотрим произвольное вычисление. Отметим в нем подпоследов-ть связок между JOIN и TEST. Индукция будет вестись по этой подпослед-ти.

База индукции. Самое первое вхождение такой связки возможно лишь единственным способом – из оператора START. Из (1) следует утверждение.

Переход. Предположим, что утверждение доказано для некоторого вхождения A_n этой связки со значениями $(x, \ y)$. Тогда на вхождении A_{n+1} переменные будут равны $(x, \ y+1)$ и из-за (2) утверждение верно на A_{n+1}.

Доказательство частичной корректности

Предположим, что существует такой предикат
$p: \ D_x \times D_y \to \{T, \ F\}$, для которого выполнено:

$$\begin{cases} \forall x \in D_x \cdot p(x, \ 0) & (1) \\ \forall x \in D_x, \ y \in D_y \cdot p(x, \ y) \wedge \neg(y = x) \Rightarrow p(x, \ y+1) & (2) \\ \forall x \in D_x, \ y \in D_y \cdot p(x, \ y) \wedge (y = x) \Rightarrow 2 * y \geq x & (3) \end{cases}$$

Тогда по лемме этот предикат выполнен во всех конфигурациях на связке между JOIN и TEST, достижимых из START. Но тогда по (3) следует, что на всех конфигурациях между TEST и HALT, достижимых из START, выполнено постусловие, т.е. что блок-схема частично корректна относительно спецификации.
Такой предикат действительно существует: $p(x, \ y) \equiv y \geq 0$.

Содержание

Пути в блок-схемах

Дополним блок-схему «псевдосвязками»: перед оператором START и после каждого оператора HALT.

Путь в блок-схеме — это последовательность связок или псевдосвязок, начинающаяся и заканчивающая на связке или псевдосвязке, являющаяся путем в графе блок-схемы. Обозначение: $e_1 - [n_1] - > e_2 - [n_2] - > \ldots - [n_k] - > e_{k+1}$.

Циклический путь — это путь, в котором некоторая связка используется более 1 раза.

Предварительные определения

$R_\alpha : D_{\bar{x}} \times D_{\bar{y}} \to \{T, F\}$ – предикат пути α в блок-схеме (множество значений переменных в начале пути, при которых вычисление «пойдет» по пути α).

$r_\alpha : D_{\bar{x}} \times D_{\bar{y}} \to D_{\bar{y}}$ – функция пути α в блок-схеме (значения промежуточных переменных в конце пути α).

Определение функций R_α и r_α (по индукции)

$R_\alpha(\bar{x},\ \bar{y}) \equiv R^1_\alpha(\bar{x},\ \bar{y})$. $r_\alpha(\bar{x},\ \bar{y}) \equiv r^1_\alpha(\bar{x},\ \bar{y})$.

- $R^{k+1}_\alpha(\bar{x},\ \bar{y}) \equiv T$, $r^{k+1}_\alpha(\bar{x},\ \bar{y}) \equiv \bar{y}$
- если n_m – START с функцией f, то $R^m_\alpha(\bar{x},\ \bar{y}) \equiv R^{m+1}_\alpha(\bar{x},\ \bar{y})$, $r^m_\alpha(\bar{x},\ \bar{y}) \equiv r^{m+1}_\alpha(\bar{x},\ f(\bar{x}))$
- если n_m – ASSIGN с функцией g, то $R^m_\alpha(\bar{x},\ \bar{y}) \equiv R^{m+1}_\alpha(\bar{x},\ \bar{y})$, $r^m_\alpha(\bar{x},\ \bar{y}) \equiv r^{m+1}_\alpha(\bar{x},\ g(\bar{x},\ \bar{y}))$
- если n_m – TEST с функцией t и связка e_{m+1} помечена значением b, то $R^m_\alpha(\bar{x},\ \bar{y}) \equiv t(\bar{x},\ \bar{y}) = b\ \wedge\ R^{m+1}_\alpha(\bar{x},\ \bar{y})$, $r^m_\alpha(\bar{x},\ \bar{y}) \equiv r^{m+1}_\alpha(\bar{x},\ \bar{y})$
- если n_m – JOIN, то $R^m_\alpha(\bar{x},\ \bar{y}) \equiv R^{m+1}_\alpha(\bar{x},\ \bar{y})$, $r^m_\alpha(\bar{x},\ \bar{y}) \equiv r^{m+1}_\alpha(\bar{x},\ \bar{y})$

Содержание

Определения

Множество точек сечения — это подмножество множества связок блок-схемы такое, что каждый циклический путь блок-схемы содержит хотя бы одну связку из этого множества.

Базовый путь — это путь без самопересечений, который начинается в точке сечения или псевдосвязке и заканчивается в точке сечения или псевдосвязке и внутри которого нет точек сечения.

Метод индуктивных утверждений (1)

Шаг 1

Выбрать множество точек сечения.

Шаг 2

Каждой точке сечения сопоставить *индуктивное утверждение*, т.е. предикат $p : D_{\bar{x}} \times D_{\bar{y}} \to \{T,\ F\}$. Псевдосвязке перед START сопоставить $p(\bar{x},\ \bar{y}) \equiv \varphi(\bar{x})$. Каждой псевдосвязке после HALT с функцией h сопоставить $p(\bar{x},\ \bar{y}) \equiv \psi(\bar{x},\ h(\bar{x},\ \bar{y}))$.

Шаг 3

Выписать *условие верификации* для каждого *базового пути* α (началу пути сопоставлено p_1, концу пути — p_2):
$\forall \bar{x} \in D_{\bar{x}}, \bar{y} \in D_{\bar{y}}\ \varphi(\bar{x}) \wedge p_1(\bar{x}, \bar{y}) \wedge R_\alpha(\bar{x}, \bar{y}) \Rightarrow p_2(\bar{x}, r_\alpha(\bar{x}, \bar{y}))$

Корректность метода индуктивных утверждений

Теорема

Дана произвольная блок-схема P и спецификация для нее $(\varphi,\ \psi)$. Пусть сделаны все шаги метода индуктивных утверждений. Тогда если все выписанные условия верификации истинны, то $\{\varphi\}\ P\ \{\psi\}$.

Замечание: иногда индуктивные утверждения называют *инвариантами циклов* (т.к. они должны быть выполнены всегда, когда вычисление программы находится в точке, куда они приписаны).

Как искать инварианты циклов?

Идеи к автоматическому поиску инвариантов циклов:

- конструирование инварианта при известной структуре цикла (`for(int i = 0; i < 1024; ++i) Body` $\Rightarrow 0 \le i \le 1024$)
- итеративное уточнение инварианта (пытаемся угадать инвариант $\rightarrow$ проверяем инвариант $\rightarrow$ подстраиваем инвариант по полученному контрпримеру).

Очень много статей на эту тему.