14 Ноября 2013

Инсерционное моделирование 1

Лекция 12

Метод Флойда

Слайд 19

Метод Хоара: **преобразование структурной программы + спецификации + инвариантов цикла в логическую формулу и доказательство этой формулы**

Метод Флойда:
Символьное моделирование аннотированной программы

Модель программы:

Конечная размеченная настроенная транзиционная система

Действия программы:

пары < условие > → <оператор >

частные случаи:

1 → <оператор > : <оператор >

< условие > → empty : **check** < условие >

предположения: assumption < формула >

утверждения: assertion < формула >

Язык аннотаций может быть шире , чем язык условий

Пример программы

Программа с goto:

```
      y:=x;
      z:=1;
L1:   y:=y-1;
      if y>=0
      then z:=z*(y+1)
      else  go to L3;
      go to L1
L3:   stop
```

В структурном виде:

```
      y:=x;
      z:=1;
      y:=y-1;
      while y>=0 do
        z:=z*(y+1);
        y:=y-1
      end;
      stop
```

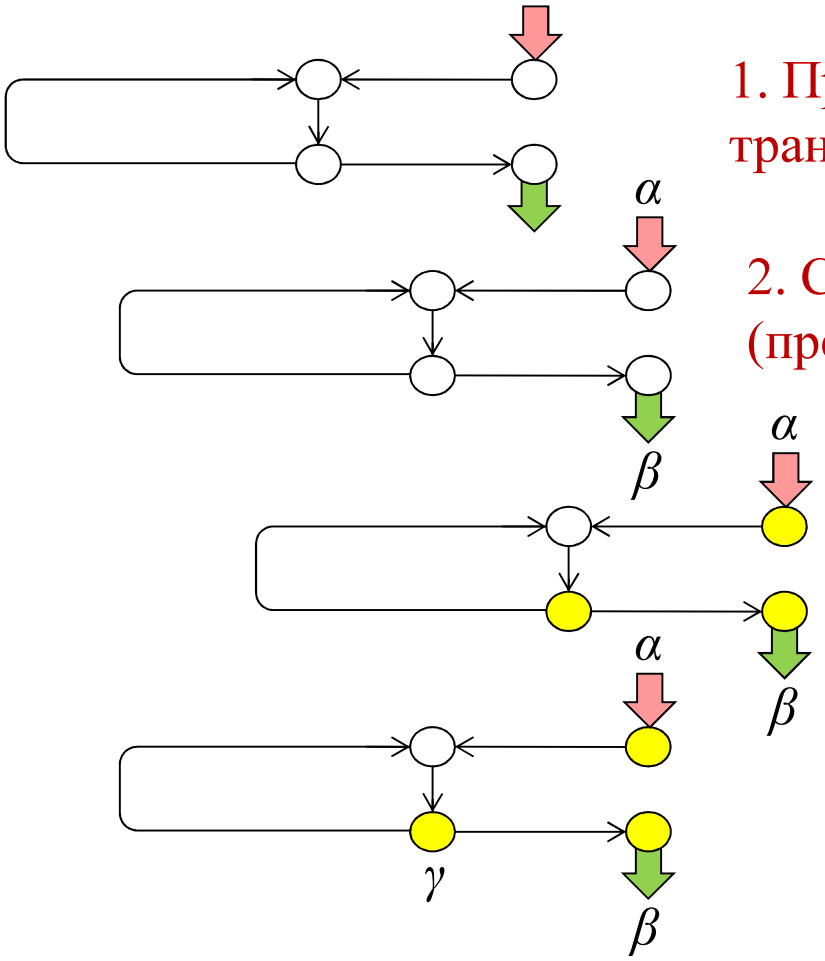
Недетерминированный выбор

Система уравнений в алгебре поведений:

L0 = (y:=x,z:=1).L1,
L1 = (y:=y-1).L2,
L2 = (y>=0 → z:=z*(y+1)).L1 + (y<0)

L0 = (y:=x,z:=1).L1,
L1 = (y:=y-1).((y>=0 → z:=z*(y+1)).L1 + (y<0))

Основные этапы метода Флойда



1. Представление программы в виде транзиционной системы

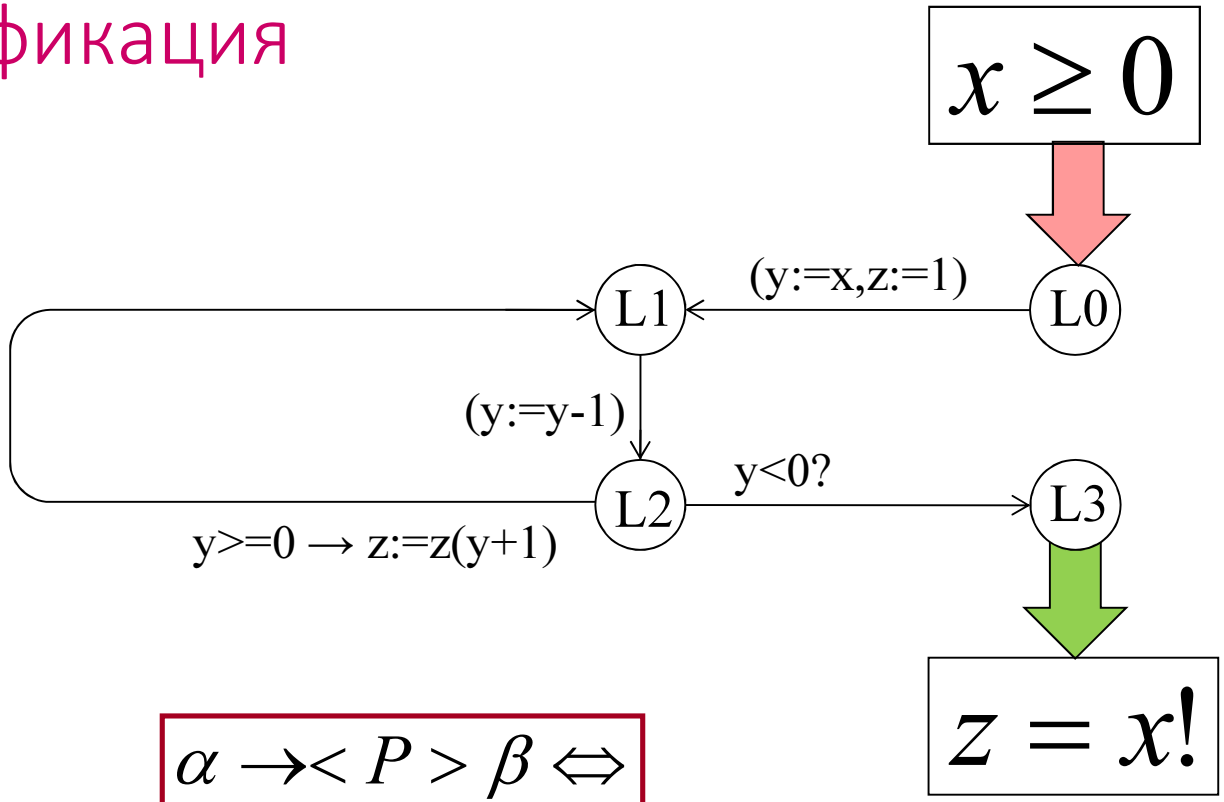
2. Спецификация программы (пред- и постусловия)

3. Выделение контрольных точек, которые разрезают все циклы

4. Аннотирование программы (утверждения и предположения в контрольных точках)

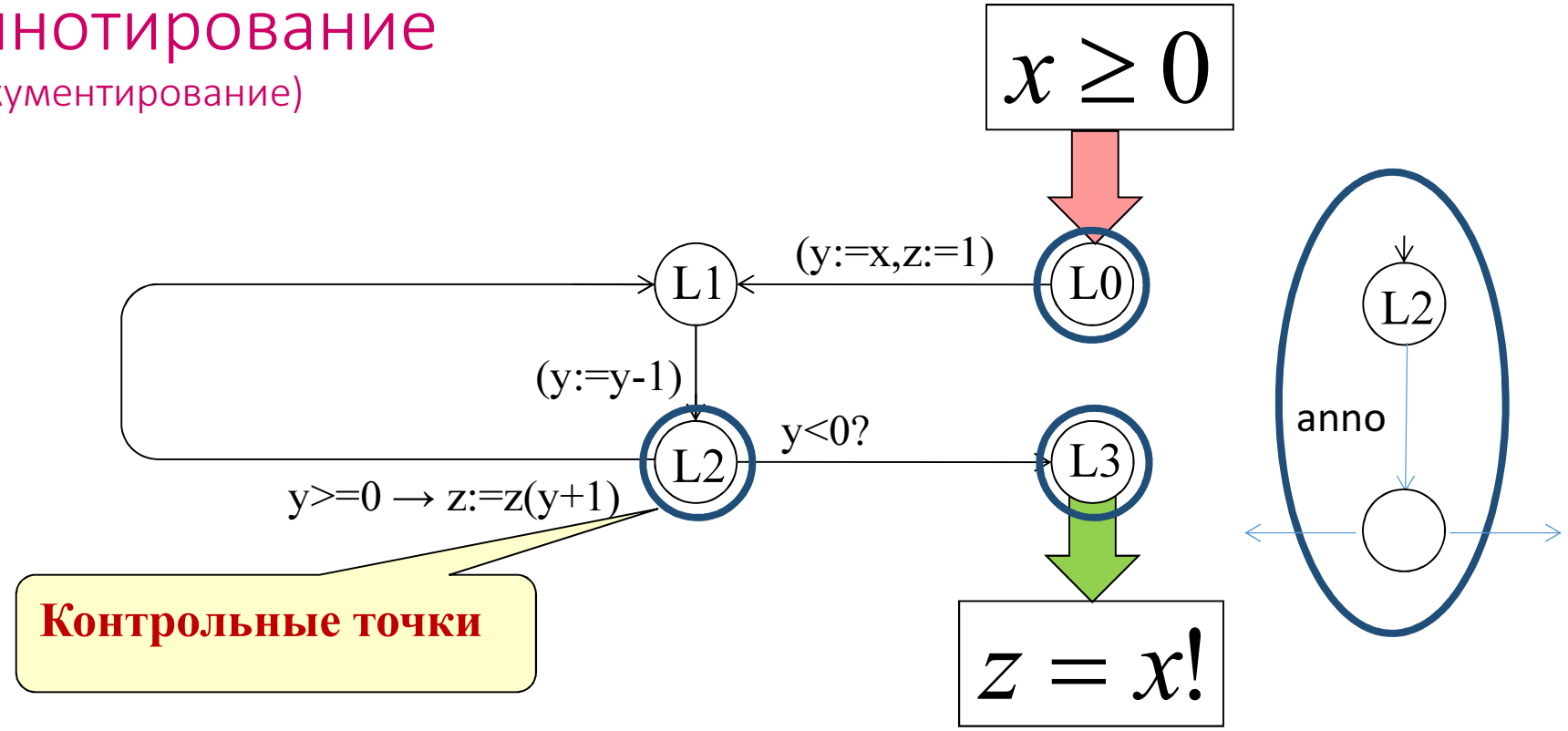
5. Доказательство верифицирующих условий

Спецификация



$$\alpha \rightarrow < P > \beta \Leftrightarrow$$
$$\alpha \rightarrow < P > 1 \wedge$$
$$\alpha \rightarrow [P]\beta$$

Аннотирование
(документирование)



аннотации

$$L0 : x \geq 0$$

$$L2 : -1 \leq y < x \wedge x! = z(y+1)!$$

$$L3 : z = x!$$

Главная идея программы или догадка?

```
L0:assumption:x>=0;
      y:=x;
      z:=1;
L1: y:=y-1;
L2: assertion: -1<=y<x/\x!=z*(y+1)!;
      if y>=0
      then z:=z*(y+1)
      else  go to L3;
      go to L1
L3: assertion: z=x!
      stop
```

Аннотации удобно также вставлять в текст программы, выделяя их специальным символом, например, **assertion**: x<y

```
L0: assumption: x>=0;
      y:=x;
      z:=1;
L1: y:=y-1;
L2: assertion: -1<=y<x/\x!=z*(y+1)!;
      while y>=0 do
         assertion: -1<=y<x/\x!=z*(y+1)!;
         z:=z*(y+1);
         L1: y:=y-1
      end;
L3: assertion: z=x!;
       stop
```

Верифицирующие условия

Для каждой пары контрольных точек *a* и *b* аннотированных условиями φ и ψ, соответственно, построим все пути из *a* в *b* (контрольные пути)

$$\pi : a = a_0 \xrightarrow{q_1} a_1 \xrightarrow{q_2} a_2 \ldots a_{m-1} \xrightarrow{q_m} a_m = b$$

$$q_i = u_i \rightarrow p_i, i = 1,\ldots, m > 0$$

Условие корректности (верифицирующее условие) для пути π:

$$C_\pi = \varphi \rightarrow [q_1 q_2 \ldots q_m]\psi = \varphi \rightarrow [Q_\pi]\psi$$

$$a / \varphi, b / \psi$$

$$Q_\pi = q_1 q_2 \ldots q_m$$

a/u: контрольная точка *a* отмечена условием *u*

Теорема (Флойд)

Все верифицирующие условия верны=>
программа частично корректна

Начальное состояние

Заключительное состояние

$$s[a] = s_0[a_0] \xrightarrow{q_1} s_1[a_1] \xrightarrow{q_2} s_2[a_2] \ldots s_m[a_m] \xrightarrow{q_m} s_{m+1}[a_{m+1}] = s_{m+1}[b] \Leftrightarrow$$

$$\pi : a = a_0 \xrightarrow{q_1} a_1 \xrightarrow{q_2} a_2 \ldots a_m \xrightarrow{q_m} a_{m+1} = b \Leftrightarrow$$

$$a \xrightarrow{\pi} b \Leftrightarrow a = c_1 / \varphi_1 \xrightarrow{\pi_1} c_2 / \varphi_2 \xrightarrow{\pi_2} c_3 / \varphi_3 \ldots \xrightarrow{\pi_{n-1}} c_n / \varphi_n = b$$

Контрольные точки

$$\alpha = \varphi_1, \beta = \varphi_n, \alpha \to [P]\beta$$

$$\varphi_1 \to [Q_{\pi_1}]\varphi_2, \varphi_2 \to [Q_{\pi_2}]\varphi_3, \ldots, \varphi_{n-1} \to [Q_{\pi_{n-1}}]\varphi_n, \alpha \to [Q_\pi]\beta$$

Проверка верифицирующих условий
(для простых атрибутов)

$$\pi : a = a_0 \xrightarrow{q_1} a_1 \xrightarrow{q_2} a_2 ... a_{m-1} \xrightarrow{q_m} a_m = b$$
$$q_i = u_i \to p_i, i = 1,...,m > 0$$

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

$$[(x_1 := t_1, x_2 := t_2, ...)]\beta(x_1, x_2, ...) <=> \beta(t_1, t_2, ...)$$

$$[p;(u \to q)]\beta \Leftrightarrow [p]u \wedge [pq]\beta$$
$$[pq]\beta \Leftrightarrow [p]([q]\beta)$$

$$C_\pi = \varphi \to [q_1 q_2 ... q_m]\psi \Leftrightarrow$$
$$\Leftrightarrow \varphi \to [u_1 \to p_1][q_2 ... q_m]\psi \Leftrightarrow$$
$$\Leftrightarrow \varphi \wedge u_1 \to [p_1]u_2 \wedge [p_2 ... q_m]\psi \Leftrightarrow$$
$$\Leftrightarrow \varphi \wedge u_1 \wedge [p_1]u_2 \to [p_1 p_2 q_3 ... q_m]\psi \Leftrightarrow$$
$$\Leftrightarrow \varphi \wedge u_1 \wedge [p_1]u_2 \wedge [p_1 p_2]u_3 \wedge ... \wedge [p_1 p_2 ... p_{m-1}]u_m \to [p_1 p_2 ... p_m]\psi$$

Композиция присваиваний

$$P = (x_1 := t_1(x_1, x_2, \ldots), x_2 := t_2(x_1, x_2, \ldots), \ldots)$$

$$P' = (x_1 := t'_1(x_1, x_2, \ldots), x_2 := t'_2(x_1, x_2, \ldots), \ldots)$$

$$(P; Q) = (x_1 := t'_1(t_1, t_2, \ldots), x_2 := t'_2(t_1, t_2, \ldots), \ldots)$$

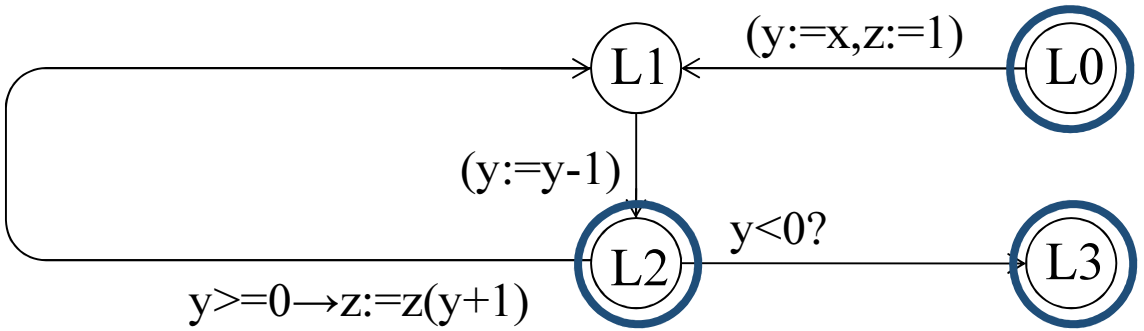
$$(P; P')(s) = P'(P(s)) \Rightarrow ((P; P')(s))(x_i) = t'_i(t_1, t_2, \ldots)$$

Верификация факториала
(частичная корректность)

$L0 : x \geq 0$

$L2 : -1 \leq y < x \wedge x! = z(y+1)!$

$L3 : z = x!$



$C_{02} : x \geq 0 \rightarrow [y := x-1, z := 1](-1 \leq y < x \wedge x! = z(y+1)!) \Leftrightarrow$
$x \geq 0 \rightarrow (-1 \leq x-1 < x \wedge x! = (x-1+1)!) \Leftrightarrow$ **истина**

$C_{22} : (-1 \leq y < x \wedge x! = z(y+1)!) \wedge y \geq 0 \rightarrow$
$[y := y-1, z := z*(y+1)](-1 \leq y < x \wedge x! = z(y+1)!) \Leftrightarrow$
$(-1 \leq y < x \wedge x! = z(y+1)!) \wedge y \geq 0 \rightarrow (-1 \leq y-1 < x \wedge x! = z(y+1)y!)$

доказательство

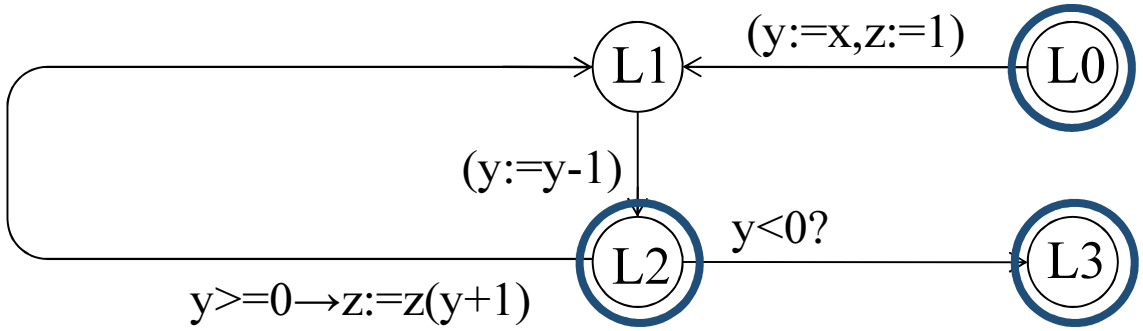
пусть $(-1 \leq y < x \wedge x! = z(y+1)!) \wedge y \geq 0$

$x! = z(y+1)! = z(y!(y+1)) = z(y+1)y!$

по определению факториала : $0! = 1, (y+1)! = y!(y+1)$

$C_{23} : (-1 \leq y < x \wedge x! = z(y+1)!) \wedge y < 0 \rightarrow z = x!$

Завершимость факториала



$$P = Q_{02} Q_{22}^{*} Q_{23}$$

При каждом прохождении цикла y убывает => станет < 0

Конкретное и символьное моделирование аннотированных императивных программ

Конкретная модель: транзиционная система s[P],
s – состояние памяти, P – фрагмент программы
Действия программы и среды: условия и присваивания
Функция погружения определена раньше.
Нужно только добавить переходы для аннотаций

Символьная модель: транзиционная система s[P],
s – логическая формула, определенная на состояниях памяти,
P – фрагмент аннотированной программы.
Действия программы и среды: условия, присваивания,
предположения и утверждения.

Переходы конкретной среды

$$s \xrightarrow{assumption(\alpha)} s, s \models \alpha$$

$$s \xrightarrow{assertion(\alpha)} s, s \models \alpha$$

$$s \models \alpha(x_1, x_2, \ldots) \Leftrightarrow \alpha(s(x_1), s(x_2), \ldots) = 1$$

Вычисления вместо доказательств

Переходы символьной среды

для простых атрибутов

$$s \xrightarrow{check(\alpha)} s',$$
$$s' = s \wedge \alpha, s' \neq 0$$

Если $s' = 0$ выбери другую альтернативу

$$s(x, y) \xrightarrow{x:=F(x,y)} s',$$
$$s' = \exists z(s(z, y), x = F(z, y) \neq \perp)$$

$$s \xrightarrow{assumption(\alpha)} s',$$
$$s' = s \wedge \alpha, s' \neq 0,$$

Если $s' = 0$ сигнализируй ошибку

$$s \xrightarrow{assertion(\alpha)} \alpha,$$
$$\models s \rightarrow \alpha,$$

$$s(x, y) \xrightarrow{\alpha(x,y) \rightarrow x:=F(x,y)} s'$$
$$s' = \exists z(s(z, y), \alpha(z, y), x = F(z, y) \neq \perp)$$

Деление на 0, выход за пределы массива и т.д.

Переходы символьной среды
для функциональных атрибутов

$$f(i) > 1 \xrightarrow{f(j):=2} ?$$

$$(i = j \wedge f(i) = 2 \vee i \neq j \wedge f(i) > 1 \wedge f(j) = 2)$$

$$assign(f, j, 2)(i) > 0$$

$$s(x, y) \xrightarrow{x:=F(x,y)} s',$$

$$s' = pt(s(x, y), x := F(x, y))$$

Функция погружения:

$$\frac{E \xrightarrow{\alpha \to a} E', u \xrightarrow{\alpha \to a} u'}{E[u] \to E'[u']} \; E \wedge \alpha \neq 0$$

$$\frac{L : E \xrightarrow{assumption(\alpha)} E', u \xrightarrow{assumption(\alpha)} u'}{E[u] \xrightarrow{cons(L,\alpha)} E'[u']} \; E \wedge \alpha \neq 0$$

$$\frac{L : E \xrightarrow{assumption(\alpha)} E', u \xrightarrow{assumption(\alpha)} u'}{E[u] \xrightarrow{incons(L,\alpha)} 0} \; E \wedge \alpha = 0$$

$$\frac{Q \xrightarrow{assertion(\alpha)} Q'}{E[Q] \xrightarrow{cons(\alpha)} \alpha[Q']} \; (E \to \alpha) = 1$$

$$\frac{Q \xrightarrow{assertion(\alpha)} Q'}{E[Q] \xrightarrow{incons(\alpha)} 0} \; \neg(E \to \alpha) \neq 0$$

Теоремы

Программа, аннотированная по методу Флойда относительно спецификации $\alpha \to [P]\beta$, **частично корректна** относительно этой спецификации, если ее символьная модель не имеет тупиков.

По правилам переходов тупики возникают лишь если программа некорректна.
Символьная модель проверяет больше!

Символьная модель программы *P*, аннотированной по методу Флойда, имеет конечное число состояний достижимых из начального состояния 1[*P*].

Модель имеет конечное число состояний вида $assumption(\alpha).E[Q]$.
Все остальные лежат на путях между ними. Но таких путей конечное число.

Символьная среда для метода Флойда

(ленивые вычисления)

Состояния среды представляются в виде:

$\varphi[p,Q]$, где

φ – формула базового языка,

p – параллельное присваивание

Q – аннотированная программа

Начальное состояние: $1[empty,Q]$

Заключительное состояние: $\varphi\ [empty,\Delta]$

Функция погружения

$$\frac{Q \xrightarrow{check(\alpha)} Q'}{s[p,Q] \to (s \wedge [p]\alpha)[p,Q']}\, s \wedge [p]\alpha \neq 0$$

$$\frac{Q \xrightarrow{x:=t} Q'}{s[p,Q] \to s[p*(x:=t),Q']}$$

$$\frac{Q \xrightarrow{assumption(\alpha)} Q'}{E[p,Q] \xrightarrow{cons(\alpha)} (E \wedge [p]\alpha)[p,Q']}\, E \wedge [p]\alpha \neq 0$$

$$\frac{Q \xrightarrow{assumption(\alpha)} Q'}{E[p,Q] \xrightarrow{incons(\alpha)} 0}\, E \wedge [p]\alpha = 0$$

$$\frac{Q \xrightarrow{assertion(\alpha)} Q'}{E[p,Q] \xrightarrow{cons(\alpha)} \alpha[empty,Q']}\, (E \to [p]\alpha) = 1$$

$$\frac{Q \xrightarrow{assertion(\alpha)} Q'}{E[p,Q] \xrightarrow{incons(\alpha)} 0}\, \neg(E \to [p]\alpha) \neq 0$$

При вычислениях нет кванторов