Анализ зависимостей по данным

DDG (data dependency graph)

Пример

```
for (i=0;i<N;i++)
{
        A[i] = ...;      // S
        ... = A[i-1];  // T
}
```

?

```
for (i=0;i<N/2;i++)
{
        A[i] = ...;      // S
        ... = A[i-1];  // T
}
```

```
for (i=N/2+1;i<N;i++)
{
        A[i] = ...;      // S
        ... = A[i-1];  // T
}
```

Как формализовать эти ограничения ?

Thread 1

Thread 2

Зависимость по данным

Def: statement T зависит от statement S ($S\delta T$), если ∃ срабатывание T` statement-a T и

срабатывание S` statement-a S и

ячейка памяти M |

- S` и T` обращаются в ячейку M и по крайней мере одно обращение является записью

- S` выполняется перед T`

- M не записывается между S` и T`

Типы зависимостей

тип	смысл	обозначение
Flow-dependent	S` - write T` - read	Sδ^fT read after write
Anti-dependent	S` - read T` - write	Sδ^aT write after read
Output-dependent	S` - write T` - write	Sδ^oT write after write
Input-dependent (не подпадает под определение с предыдущего слайда, т.к. не даёт ограничений, но бывает полезной)	S` - read T` - read	read after read

read after read

S1 … = A[i-1];

…

S2 … = A[i-1];

зависимость

read after read

scalar replacement

t = A[i-1];

…

S1 … = t;

…

S2 … = t;

Отношение порядка на векторах

Def: Лексикографический порядок (lex order)

Пусть $\mathbf{x}, \mathbf{y} \in R^n$

$$\mathbf{x} = (x_1, x_2, \dots x_n)$$

$$\mathbf{y} = (y_1, y_2, \dots y_n)$$

Строгий порядок $\mathbf{x} <_v \mathbf{y} \quad (1 \le v \le n)$ <=>

$$x_1 = y_1, x_2 = y_2, \dots x_{v-1} = y_{v-1}, x_v < y_v$$

Отношение порядка на векторах

Строгий порядок $\mathbf{x} <_v \mathbf{y}$ $(1 \leq v \leq n)$ <=>

$$x_1 = y_1, x_2 = y_2, \ldots x_{v-1} = y_{v-1}, x_v < y_v$$

Строгий порядок $\mathbf{x} >_v \mathbf{y}$ $(1 \leq v \leq n)$ <=>

$$x_1 = y_1, x_2 = y_2, \ldots x_{v-1} = y_{v-1}, x_v > y_v$$

Нестрогий порядок $\mathbf{x} \leq_v \mathbf{y}$ $(1 \leq v \leq n)$ <=>

$$x_1 = y_1, x_2 = y_2, \ldots x_{v-1} = y_{v-1}, x_v \leq y_v$$

Нестрогий порядок $\mathbf{x} \geq_v \mathbf{y}$ $(1 \leq v \leq n)$ <=>

$$x_1 = y_1, x_2 = y_2, \ldots x_{v-1} = y_{v-1}, x_v \geq y_v$$

Отношение порядка на векторах

Строгий порядок $\mathbf{x} < \mathbf{y}$ -

объединение отношений

$\mathbf{x} <_v \mathbf{y} \;\; (1 \leq v \leq n)$

Остальные отношения определяются аналогично

Отношение порядка на векторах

Свойства:

* $\mathbf{x} <_v \mathbf{y}$ $(1 \le v \le n)$ - нерефлексивно, транзитивно
* $\mathbf{x} \le_v \mathbf{y}$ $(1 \le v \le n)$ – частичный порядок
* если $\mathbf{x} \ne \mathbf{y}$ $\exists!\, v$ $(1 \le v \le n)$ | справедливо $\mathbf{x} <_v \mathbf{y}$ или $\mathbf{x} >_v \mathbf{y}$
* если $\mathbf{x} <_u \mathbf{y}$ и $\mathbf{y} <_v \mathbf{z}$ => $\mathbf{x} <_w \mathbf{z}$ *w=min(u,v)*
* $\mathbf{x} < \mathbf{y}$ – нерефлексивно, транзитивно
* $\mathbf{x} \le \mathbf{y}$ – полный порядок

Отношение порядка на векторах

Def: расширим определение

$$\mathbf{x} <_v \mathbf{y} \ , \ v = n+1$$

<=>

$$\mathbf{x} = \mathbf{y}$$

Отношение порядка на векторах

Def: Первый ненулевой элемент вектора $\mathbf{x}$ – лидирующий элемент (leading element)

вектор $\mathbf{x} < \mathbf{0}$, если лидирующий элемент отрицательный

вектор $\mathbf{x} > \mathbf{0}$, если лидирующий элемент положительный

Отношение порядка на векторах

Def: Функция signum

sig: R -> Z

$$\mathrm{sig}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Def:

Пусть $\mathbf{x}, \mathbf{y} \in R^n$

$$\mathbf{x} = (x_1, x_2, \dots x_n)$$
$$\mathbf{y} = (y_1, y_2, \dots y_n)$$

distance vector для упорядоченной пары $(\mathbf{x}, \mathbf{y})$

$$\mathbf{y} - \mathbf{x} = (y_1 - x_1, y_2 - x_2, \dots, y_n - x_n)$$

direction vector для упорядоченной пары $(\mathbf{x}, \mathbf{y})$

$$(\ \mathrm{sig}(y_1 - x_1),\ \mathrm{sig}(y_2 - x_2), \dots \mathrm{sig}(y_n - x_n)\ )$$

Отношение порядка на векторах

Лемма #1

упорядоченная пара $(\mathbf{x}, \mathbf{y})$

$\boldsymbol{s}$ direction vector

$\mathbf{x} < \mathbf{y}$ <=> $\boldsymbol{s} > \mathbf{0}$ <=> $\boldsymbol{s}$ имеет одну из следующих форм

(1,*, … *)

(0,1, … *)

(0,0,1, … *)

…

(0,0,0, … ,0,1)

Модель программы

```
for(i1=p1..q1)
{
   …
   for(i2=p2..q2)
   {
      …
      for(i3=p3..q3)
      {
         …
            for(ik=pk..qk)
            {
               … ←
            }
         …
      }
      …
   }
   …
}
```

Циклы - L1, L2, … ,Lk

I=(i1, i2, … ik)

P=(p1, p2, … pk) Шаг индекса +1

Q=(q1, q2, … qk)

Iteration space:

(i1, i2, … ik) $\in Z^k$ |

p1 <= i1 <= q1

p2 <= i2 <= q2

…

pk <= ik <= qk

Модель программы

```
for(…)
{
   …
   for(…)
   {
      …
      S
      …
   }
   …
   for(…)
   {
      …
      T
      …
   }
   …
}
```

Вложенность циклов

L = (L1, L2, … , Le) - включают в себя S и T

Ls = (L[e+1], … Lm) - включают в себя только S

Lt = (L[m+1], … Ln) - включают в себя только T

nest(S) = (L, Ls) - гнездо циклов для S

nest(T) = (L, Lt) - гнездо циклов для T

Бегущие индексы

I = (i1, i2, … ie) - индексы для циклов L

Is = (i[e+1], … im) - индексы для циклов Ls

It = (i[m+1], … in) - индексы для циклов Lt

Модель программы

Def: отношение порядка для инструкций S и T

(в лексическом смысле)

S < T

<=>

S предшествует инструкции T в коде

пример

```
for(…)
{
    for(…)
    {
        S
    }
    for(…)
    {
        T
    }
}
```

S < T

Модель программы

Лемма #2

Пусть даны две инструкции S и T

nest(S) = (L, Ls)

nest(T) = (L, Lt)

Срабатывание инструкции S(I, Is) происходит перед срабатыванием T(J,Jt)

<=>

если S < T, I <= J

если T <= S, I < J

$\lhd$ Следует рассмотреть случаи $S < T, T < S, S = T$ $\rhd$

Зависимость по данным

Def: statement T зависит от statement S ($S\delta T$), если ∃ переменная x в S и переменная y в T, ∃ итерации (I, Is) в nest(S) и (J, Jt) в nest(T) |

- по крайней мере в одну переменную x или y делается запись
- x в итерации (I, Is) и y в итерации (J, Js) представляет собой общую ячейку памяти M
- S(I, Is) выполняется перед T(J, Jt) (лемма #2)
- M не перезаписывается между S(I, Is) и T(J, Jt)

Зависимость по данным

Def: statement T зависит от statement S на уровне u по паре переменной (x, y), если

- (x, y) является причиной зависимости T от S
- (I, Is) и (J, Js) $\mathrm{I} <_v \mathrm{J}$

Следств. из леммы #2

u=1, 2, …, e, e+1 для S < T

u=1, 2, …, e для T<=S

Зависимость по данным

Def: (x, y) является причиной зависимости T от S с distance vector S = (s1, s2, … , se), если

для срабатываний (I, Is) и (J, Js), где проявляется зависимость,

$s1 = j1 – i1$

$s2 = j2 – i2$

$s3 = j3 – i3$

…

$se = je - ie$

Зависимость по данным

Def: (x, y) является причиной зависимости T от S c direction vector S = (s1, s2, … , se), если

для срабатываний (I, Is) и (J, Js), где проявляется зависимость,

$s1 = sig(j1 – i1)$

$s2 = sig(j2 – i2)$

$s3 = sig(j3 – i3)$

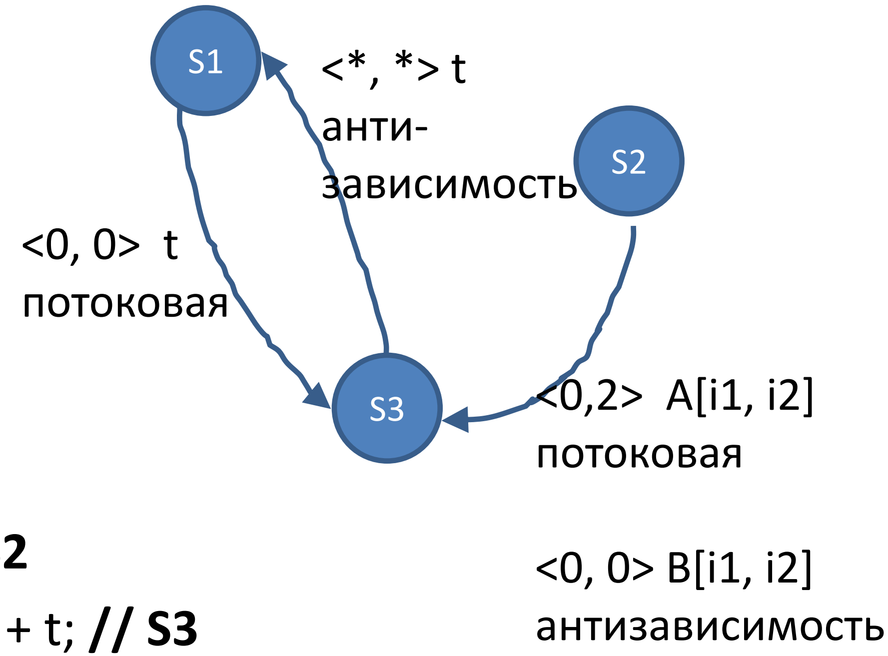
…

$se = sig(je – ie)$

Зависимость по данным

```
// A, B, C, D – разные массивы

for(int i1=1; i1 <= 3; i++)
{
    for(int i2=1; i2 <= 4; i++)
    {
        t = x + y;  // S1
        A[i1, i2] = B[i1, i2] + C[i1, i2]; // S2
        B[i1, i2] = A[i1, i2-2] * D[i1+1, i2] + t; // S3
    }
}
```



Зависимость по данным

Факт: x – output variable, y – input variable

(x, y) может являться причиной $S\delta^{f}T$
(y, x) может являться причиной $T\delta^{a}S$

```
for( … ) // x, y, t – одна и та же ячейка памяти
{
    t = …                  (S)
    … = … t …              (T)
}
```

Зависимость по данным

Факт: x – output variable, y – output variable

(x, y) может являться причиной Sδ^{o}T
(y, x) может являться причиной T$\delta^{o}S$

```
for( … ) // x, y, t – одна и та же ячейка памяти
{
    t = …                (S)
    t = …                (T)
}
```

Зависимость по данным

Def: T косвенно зависит от S, если ∃ цепочка зависимостей

S δ S1, S1 δ S2, S2 δ S3, … ,S[p-1] δ Sp, Sp δ T

δ – flow, anti, output зависимости

Лемма #3.

$f_r : \mathbb{Z}^m \to \mathbb{Z} \quad g_r : \mathbb{Z}^{n-m+e} \to \mathbb{Z}$, 1<= r <= DIM

Две переменной (DIM – размерность массива)

x=A[f_1(I,Is), f_2(I,Is), … f_{DIM}(I,Is)]

y=A[g_1(I,It), g_2(I,It), … g_{DIM}(I,It)]

(x, y) – причина S δ T c direction s=(s1, s2,…, se)

<=>

∃ целочисленное решение уравнения

$$f_r(i_1, i_2, \ldots i_e, i_{e+1}, \ldots i_m) = g_r(j_1, \ldots j_e, j_{m+1}, \ldots j_n),$$

где 1 <= r <= DIM,

которое удовлетворяет условиям

$p_k \le i_k \le q_k$ (1 <= k <=m)

$p_k \le j_k \le q_k$ (1 <= k <=e, m+1 <= k <= n)

sgn($i_k - j_k$) = s_k (1 <= k <= e)

Лемма #3’.

$f_r : \mathbb{Z}^m \to \mathbb{Z}$ $g_r : \mathbb{Z}^{n-m+e} \to \mathbb{Z}$, 1<= r <= DIM

Две переменной (DIM – размерность массива)

x=A[f_1(I,Is), f_2(I,Is), … f_{DIM}(I,Is)]

y=A[g_1(I,It), g_2(I,It), … g_{DIM}(I,It)]

(x, y) – причина S δ T с direction s=(s1, s2,…, se), на уровне u

<=>

∃ целочисленное решение уравнения

$f_r(i_1, i_2, \dots i_e, i_{e+1}, \dots i_m) = g_r(j_1, \dots j_e, j_{m+1}, \dots j_n)$,

где 1 <= r <= DIM,

которое удовлетворяет условиям

$p_k \le i_k \le q_k$ (1 <= k <=m)

$p_k \le j_k \le q_k$ (1 <= k <=e, m+1 <= k <= n)

$i_u < j_u$, если 1 <= u <= e

Зависимость по данным

Замечания по лемме #3

- нас будет интересовать случай линейных функций (линейные диофантовы уравнения)
- у нас нет необходимости искать решение, нам достаточно установить факт существования такого решения

Зависимость по данным

```
extern int f(int i);
extern int g(int i);
```

```
for (i=0;i<N;i++)
{
        A[f(i)] = …;  // S
        … = A[g(i)];  // T
}
```

?

```
for (i=0;i<N/2;i++)
{
        A[f(i)] = …;  // S
        … = A[g(i)];  // T
}
```

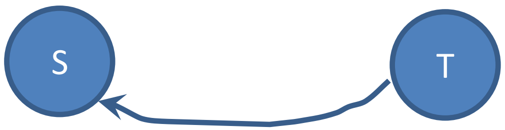
```
for (i=N/2+1;i<N;i++)
{
        A[f(i)] = …;   // S
        … = A[g(i)];   // T
}
```

Можно ли выполнить преобразование ?

Безопасно будет предположить наличие межитерационной потоковой зависимости

Thread 1

Thread 2



<*> потоковая зависимость

Зависимость по данным

Если мы не можем доказать отсутствие зависимости, которая препятствует преобразованию, то мы предполагаем её присутствие.

Преобразование должно сохранять эквивалентность.

Проведение тестов

Два подхода:

- Проверить некие формальные условия, которые говорят, ∃ общее решение, удовлетворяющее ограничениям (подход #1)
- Найти общее решение уравнений, которые удовлетворяет ограничениям (подход #2)

Проведение тестов (подход #1)

Из курса матанализа

(обобщение теоремы Больцано-Коши)

Пусть дана функция $f\colon \mathbb{R}^n \to \mathbb{R}$ непрерывная на $\mathfrak{R}$. Известно, что f принимает два значения b и B на $\mathfrak{R} \subset \mathbb{R}^n$ (связная область) и $b \le c \le B$

=>

$\exists$ решение уравнения $f(\vec{x}) = c$

причём $\vec{x} \in \mathfrak{R}$

Проведение тестов (подход #1) прямоугольные границы

Def: положительная часть числа

$$a^+ = max(a, 0)$$

Def: отрицательная часть числа

$$a^- = max(-a, 0)$$

Проведение тестов (подход #1) прямоугольные границы

Св-ва:

- $a^+ \geq 0, a^- \geq 0$
- $a = a^+ - a^-, |a| = a^+ + a^-$
- $(-a)^+ = a^-, (-a)^- = a^+$
- $(a^+)^+ = a^+, (a^-)^+ = a^-$
- $(a^+)^- = 0, (a^-)^- = 0$
- $-a^- \leq a \leq a^+$

Проведение тестов (подход #1) прямоугольные границы

Лемма #5

Если $p \leq x \leq q$ =>

$$a^+p - a^-q \leq ax \leq a^+q - a^-p$$

причём

$a^+p - a^-q$ – минимум *f(x) = ax* на [p, q]

$a^+q - a^-p$ – максимум *f(x) = ax* на [p, q]

Проведение тестов (подход #1) прямоугольные границы

Док-во леммы #5

$\lhd \quad p \le x \le q \; =>$

$a^+p \le a^+x \le a^+q, \;\; -a^-q \le -a^-x \le -a^-p =>$

$a^+p - a^-q \le (a^+ - a^-)x \le a^+q - a^-p =>$

$a^+p - a^-q \le ax \le a^+q - a^-p$

не трудно видеть, что $a^+p - a^-q$ и $a^+q - a^-p$ достигаются функцией *f(x) = ax* на концах отрезка *[p, q]*

$\rhd$

Проведение тестов (подход #1) прямоугольные границы

Лемма #6

Если $p_k \le x_k \le q_k, \quad 1 \le k \le n$ =>

$$\sum_{k=1}^{n}(a_k{}^{+}p_k - a_k{}^{-}q_k) \le \sum_{k=1}^{n} a_k x_k \le \sum_{k=1}^{n}(a_k{}^{+}q_k - a_k{}^{-}p_k)$$

$\sum_{k=1}^{n}(a_k{}^{+}p_k - a_k{}^{-}q_k)$ - минимум функции $\sum_{k=1}^{n} a_k x_k$ на указанном прямоугольнике

$\sum_{k=1}^{n}(a_k{}^{+}q_k - a_k{}^{-}p_k)$ - максимум функции $\sum_{k=1}^{n} a_k x_k$ на указанном прямоугольнике

Проведение тестов (подход #1) прямоугольные границы

$\lhd$ $p_k \le x_k \le q_k$, $1 \le k \le n$ =>

$$a_k{}^+ p_k - a_k{}^+ q_k \le a_k x_k \le a_k{}^+ q_k - a_k{}^- p_k$$

Суммируя по k получаем искомое нер-во

Не трудно видеть, что $\sum_{k=1}^{n}(a_k{}^+ p_k - a_k{}^- q_k)$ и $\sum_{k=1}^{n}(a_k{}^+ q_k - a_k{}^- p_k)$ достигаются на мно-ве точек $\{(x_1, \dots x_n) : x_1 = p_1 \text{ или } q_1, \dots x_n = p_n \text{ или } q_n\}$

$\rhd$

Проведение тестов (подход #1) прямоугольные границы

Пример

```
for(i1=1; i1<=10; i1++)
   for(i2=1; i2<=10; i2++)
   {
         A[i1-2*i2-11] = … // S
         A[-i1+i2+8] = …     // T
   }
```

Проверим факт ∃-ия зависимости по выходу на уровне 3

Приравниваем индексные выражения

$i1-2*i2-11 = -i1 + i2 + 8$

$\Rightarrow 2*i1 - 3*i2 = 19$

Применяем лемму #6

$1 <= i1 <= 10, 1 <= i2 <= 10$

$a1=2, a2=-3$

$\Rightarrow -28 <= 2*i1 - 3*i2 <= 17$

$19 \notin [-28, 17]$ =>

зависимости нет

Проведение тестов (подход #1) трапецеидальные границы

```
for(…i1…)
{
   …
   for(…i2…)
   {
      …
      for(…i3…)
      {
         …
          for(…in…)
          {
             … ←
          }
        …
      }
      …
   }
   …
}
```

$$p_{1.0} \le i_1 \le q_{1.0}$$

$$p_{2.0} + p_{2.1} i_1 \le i_2 \le q_{2.0} + q_{2.1} i_1$$

$$p_{3.0} + p_{3.1} i_1 + p_{3.2} i_2 \le i_3 \le q_{3.0} + q_{3.1} i_1 + q_{3.2} i_2$$

$$\dots$$

$$p_{n.0} + p_{n.1} i_1 + p_{n.2} i_2 + \dots + p_{n.n-1} i_{n-1} \le i_n \le q_{n.0} + q_{n.1} i_1 + q_{n.2} i_2 + \dots + q_{n.n-1} i_{n-1}$$

Расширение модели программы

Проведение тестов (подход #1) трапецеидальные границы

Лемма #7

Если $0 \le x \le q$ и $0 \le y \le x$

=>

$$-(a - b^{-})^{-}q \le ax + by \le (a + b^{+})^{+}q$$

причём

$-(a - b^{-})^{-}q$ - минимум функции $ax + by$ на указанной области

$(a + b^{+})^{+}q$ - максимум функции $ax + by$ на указанной области

Проведение тестов (подход #1) трапецеидальные границы

$\lhd \quad 0 \leq y \leq x$ => по лемме #5 $-b^{-}x \leq by \leq b^{+}x$ =>

$(a - b^{-})x \leq ax + by \leq (a + b^{+})x$

$0 \leq x \leq q$ => по лемме #5 $-(a - b^{-})^{-}q \leq (a - b^{-})x$

$0 \leq x \leq q$ => по лемме #5 $(a + b^{+})x \leq (a + b^{+})^{+}q$

$\Rightarrow$ Искомое двойное неравенство

Вторая часть утв. следует из леммы #5

$\rhd$

Проведение тестов (подход #1) трапецеидальные границы

Пусть даны трапецеидальные границы

$$p_{1.0} \le i_1 \le q_{1.0}$$
$$p_{2.0} + p_{2.1} i_1 \le i_2 \le q_{2.0} + q_{2.1} i_1$$
$$p_{3.0} + p_{3.1} i_1 + p_{3.2} i_2 \le i_3 \le q_{3.0} + q_{3.1} i_1 + q_{3.2} i_2$$
$$\dots$$
$$p_{n.0} + p_{n.1} i_1 + p_{n.2} i_2 + \cdots + p_{n.n-1} i_{n-1} \le i_n \le q_{n.0} + q_{n.1} i_1 + q_{n.2} i_2 + \cdots + q_{n.n-1} i_{n-1}$$

Вопрос: как будет ограничена функция

$$a_1 i_1 + a_2 i_2 + a_3 i_3 + \cdots + a_n i_n$$

на указанной области?

Проведение тестов (подход #1) трапецеидальные границы

$$p_{n.0} + p_{n.1}i_1 + p_{n.2}i_2 + \cdots + p_{n.n-1}i_{n-1} \le i_n \le q_{n.0} + q_{n.1}i_1 + q_{n.2}i_2 + \cdots + q_{n.n-1}i_{n-1}$$

по лемме #5 умножаем все части неравенства на a_n, а потом прибавляем $a_1 i_1$, $a_2 i_2$, …,$a_{n-1} i_{n-1}$ ко всем частям нер-ва

=>

$$(a_n{}^{+}p_{n.0} - a_n{}^{-}q_{n.0}) + (a_1 + a_n{}^{+}p_{n.1} - a_n{}^{-}q_{n.1})i_1 + \cdots + (a_{n-1} + a_n{}^{+}p_{n.n-1} - a_n{}^{-}q_{n.n-1})i_{n-1} \le a_1 i_1 + \cdots + a_n i_n \le (a_n{}^{+}q_{n.0} - a_n{}^{-}p_{n.0}) + (a_1 + a_n{}^{+}q_{n.1} - a_n{}^{-}p_{n.1})i_1 + \cdots + (a_{n-1} + a_n{}^{+}q_{n.n-1} - a_n{}^{-}p_{n.n-1})i_{n-1}$$

свободные члены

Проведение тестов (подход #1) трапецеидальные границы

Введём обозначения для скобок

$$\begin{aligned} b_{low} + d_1 i_1 + \cdots + d_{n-1} i_{n-1} \\ \le a_1 i_1 + \cdots + a_n i_n \\ \le b_{up} + e_1 i_1 + \cdots + e_{n-1} i_{n-1} \end{aligned}$$

Проведение тестов (подход #1) трапецеидальные границы

Возьмём следующее нер-во

$$p_{n-1.0} + p_{n-1.1} i_1 + \cdots + p_{n-1.n-2} i_{n-2} \le i_{n-1} \le q_{n-1.0} + q_{n-1.1} i_1 + \cdots + q_{n-1.n-2} i_{n-2}$$

по лемме #5 умножаем все части нер-ва на d_{n-1}, а потом прибавляем $d_1 i_1$, $d_2 i_2$, …,$d_{n-2} i_{n-2}$ к всем частям нер-ва

$$b_{low}{}' + d_1{}' i_1 + \cdots + d_{n-2}{}' i_{n-2} \le d_1 i_1 + \cdots + d_{n-1} i_{n-1}$$

Проведение тестов (подход #1) трапецеидальные границы

Возьмём то же нер-во

$$p_{n-1.0} + p_{n-1.1} i_1 + \cdots + p_{n-1.n-2} i_{n-2} \le i_{n-1} \\ \le q_{n-1.0} + q_{n-1.1} i_1 + \cdots + q_{n-1.n-2} i_{n-2}$$

по лемме #5 умножаем все части нер-ва на e_{n-1}, а потом прибавляем $e_1 i_1$, $e_2 i_2$, …,$e_{n-2} i_{n-2}$ к всем частям нер-ва

$$e_1 i_1 + \cdots + e_{n-1} i_{n-1} \\ \le {b_{up}}' + {e_1}' i_1 + \cdots + {e_{n-2}}' i_{n-2}$$

Проведение тестов (подход #1) трапецеидальные границы

Итого

$$b_{low} + \underline{d_1 i_1 + \dots + d_{n-1} i_{n-1}} \le a_1 i_1 + \dots + a_n i_n \le b_{up} + \underline{e_1 i_1 + \dots + e_{n-1} i_{n-1}}$$

$$b_{low}{}' + d_1{}' i_1 + \dots + d_{n-2}{}' i_{n-2} \le \underline{d_1 i_1 + \dots + d_{n-1} i_{n-1}}$$

$$\underline{e_1 i_1 + \dots + e_{n-1} i_{n-1}} \le b_{up}{}' + e_1{}' i_1 + \dots + e_{n-2}{}' i_{n-2}$$

=>

$$b_{low} + b_{low}{}' + d_1{}' i_1 + \dots + d_{n-2}{}' i_{n-2} \le a_1 i_1 + \dots + a_n i_n \le b_{up} + b_{up}{}' + e_1{}' i_1 + \dots + e_{n-2}{}' i_{n-2}$$

Мы видим, что кол-во переменных сокращается.

Если мы повторим процедуру n раз, то будут только константы

Проведение тестов (подход #1) трапецеидальные границы

Вход: трапецеидальные границы и линейная функция

$$p_{1.0} \le i_1 \le q_{1.0}$$
$$p_{2.0} + p_{2.1} i_1 \le i_2 \le q_{2.0} + q_{2.1} i_1$$
$$p_{3.0} + p_{3.1} i_1 + p_{3.2} i_2 \le i_3 \le q_{3.0} + q_{3.1} i_1 + q_{3.2} i_2$$
$$\dots$$
$$p_{n.0} + p_{n.1} i_1 + p_{n.2} i_2 + \dots + p_{n.n-1} i_{n-1} \le i_n \le q_{n.0} + q_{n.1} i_1 + q_{n.2} i_2 + \dots + q_{n.n-1} i_{n-1}$$

$$a_1 i_1 + a_2 i_2 + a_3 i_3 + \dots + a_n i_n$$

Выход: b_{low} и b_{up} |

$$b_{low} \le a_1 i_1 + a_2 i_2 + a_3 i_3 + \dots + a_n i_n \le b_{up}$$

Проведение тестов (подход #1) трапецеидальные границы

1. [Initialize.] Set $b_{low} \leftarrow 0,\ b_{up} \leftarrow 0;$

$$k \leftarrow n;$$

$$(d_1, d_2, \ldots, d_k) \leftarrow (a_1, a_2, \ldots, a_k),$$
$$(e_1, e_2, \ldots, e_k) \leftarrow (a_1, a_2, \ldots, a_k).$$

2. [Eliminate x_k.] Set $b_{low} \leftarrow b_{low} + d_k^+ p_{k0} - d_k^- q_{k0}$

$$b_{up} \leftarrow b_{up} + e_k^+ q_{k0} - e_k^- p_{k0}.$$

If $k > 1$, then set

$$(d_1, d_2, \ldots, d_{k-1}) \leftarrow (d_1 + d_k^+ p_{k1} - d_k^- q_{k1},\ d_2 + d_k^+ p_{k2} - d_k^- q_{k2}, \ldots,$$
$$d_{k-1} + d_k^+ p_{k,k-1} - d_k^- q_{k,k-1}),$$

$$(e_1, e_2, \ldots, e_{k-1}) \leftarrow (e_1 + e_k^+ q_{k1} - e_k^- p_{k1},\ e_2 + e_k^+ q_{k2} - e_k^- p_{k2}, \ldots,$$
$$e_{k-1} + e_k^+ q_{k,k-1} - e_k^- p_{k,k-1}),$$

$$k \leftarrow k - 1;$$

and go to **2**. Otherwise, halt.

Проведение тестов (подход #1) трапецеидальные границы

Пример:

```
for(int i1=1; i1<=100; i1++)
    for(int i2=i1; i2<=50; i2++)
    {
        A[i1-i2+11] = … ;
        … = … A[-2*i1 + i2 + 23] … ;
    }
```

$\underline{i1 - i2 + 11 = -2*i1' + i2' + 23}$

$1 \le i1 \le 100$

$i1 \le i2 \le 50$

$1 \le i1' \le 100$

$i1' \le i2' \le 50$

$i1 < i1'$

$\underline{i1 - i2 + 2*i1' - i2' = 12}$

$-95 \le i1 - i2 + 2*i1' - i2' \le 100$

зависимость возможно есть

Проверить факт ∃-ия flow-зависимости на уровне 1

Список литературы

- Dependence Analysis for Supercomputing. U. Banergee