# Встреча лаборатории автоматизации программирования ИПС им. А.К. Айламазяна с экспертами Санкт-Петербургского Филиала Российского Исследовательского Института

Докладчик: Непейвода А.Н.

12.03.2024

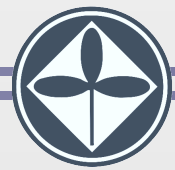
Переславль–Залесский, Санкт–Петербург

# Пример отображения

Здесь prefix(x,y) — проверка, что $y$ начинается с $x$.

```
z = ξ                          z → ξ
x = ε                          x → ε
y = ε                          y → ε
while (cond1)                  while (cond1)
   x+=z                           x → ξx = xξ
while (cond2)                  while (cond2)
   y+=z                           y → ξy = yξ
while (true)                   while (true)
   if (prefix(x,y))               if (|x| ≤ |y|)
      y-=x                           y → ξy = yξ
   elif (prefix(y,x))             elif (|y| ≤ |x|)
      x-=y                           x → ξx = xξ
   else break                     else break
```

Из анализа доменов становится возможно заменить условия prefix(x,y) и prefix(y,x) на неравенства длин, одно из которых всегда выполнено. Дальнейший статический анализ способен указать на недостижимость ветви, содержащей break.



## Соответствие Галуа

Введём отношение $\propto$ между множествами конкретных (строковых) значений $S$ и абстрактных значений $\Delta$.
А именно, скажем, что $s \propto \rho$, если строка $s$ удовлетворяет рестрикции (условию) $\rho$.

Антимонотонное соответствие Галуа, задаваемое отношением $\propto$, будет определять операции абстракции и конкретизации относительно домена абстрактных значений $\Delta$.

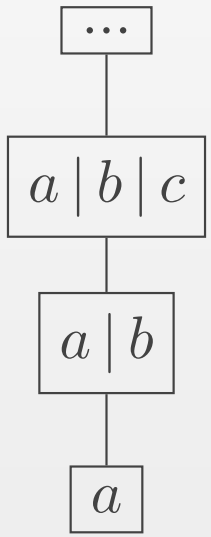
- $\phi : 2^S \to 2^\Delta; \qquad \phi(A) = \{\rho \mid \forall a \in A(a \propto \rho)\}$
- $\psi : 2^\Delta \to 2^S; \qquad \psi(\Gamma) = \{a \mid \forall \rho \in \Gamma(a \propto \rho)\}$
- $A \subseteq A' \Rightarrow \phi(A') \subseteq \phi(A); \qquad \Gamma \subseteq \Gamma' \Rightarrow \psi(\Gamma') \subseteq \psi(\Gamma)$

# Требуемые свойства домена $\Delta$

- У каждых двух рестрикций должен существовать точный супремум (для исключения «эффекта бабочки») и инфинум (для замыкания в решётку). А именно, $\forall \phi_1, \phi_2 \exists \psi_1, \psi_2 (\phi_1 \vee \phi_2 = \psi_1 \;\&\; \phi_1 \wedge \phi_2 = \psi_2)$.
- Высота строго возрастающих последовательностей должна быть конечной, желательно не больше 5.
- Свойство верхней полурешётки влечёт завершаемость (существование конечно вычислимых неподвижных точек).

# Общие замечания

- Абстрактный домен регулярных выражений удовлетворяет свойствам решётки (алгебраически замкнут относительно объединений и пересечений). Замкнут относительно гомоморфизмов и обратных гомоморфизмов. Однако высота этой решётки над бесконечным алфавитом бесконечна.



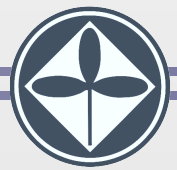

# Возможные сужения домена

Обозначение $\tau^n$ — сокращение для $\underbrace{\tau\tau\ldots\tau}_{n}$

Слово $\tau$ простое, если из $\tau = \xi^n$ следует, что $n = 1$ и $\xi = \tau$.
Т.е. $aab$ — простое, $abab$ — нет.

- Базовые уравнения (предикаты вида $\mathrm{ContAll}(s, \tau)$) и неравенства (вида $\mathrm{NCont}(s, \tau)$).
  - $\mathrm{ContAll}(s, \tau) \equiv \tau\, s = s\, \tau$, при этом $\tau$ — простое слово;
  - $\mathrm{NCont}(s, \tau) \equiv \forall z_1, z_2 (s \neq z_1 \tau z_2)$.

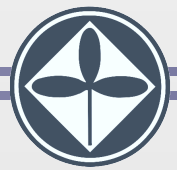
Предикаты, специализированные строкой $\tau$, обозначаем $\mathrm{ContAll}_\tau$ и $\mathrm{NCont}_\tau$. Абстрактными объектами решетки являются именно эти предикаты (вместе с предикатами «совпадать со строкой $\tau$», далее обозначаемыми просто $\tau$).



# Вычисление супремума рестрикций

Скажем, что $s_1 \ll s_2$, если $\neg\,\mathrm{NCont}(s_2, s_1)$.
Предикат $\mathrm{ContAll}_\varepsilon$ предполагаем некорректно заданным, поэтому всюду в случае определения предиката $\mathrm{ContAll}_\tau$ предполагаем, что $\tau$ простое и не равно $\varepsilon$.

- $\mathrm{ContAll}_\tau \vee \mathrm{ContAll}_\xi = \begin{cases} \mathrm{ContAll}_\tau, \text{ если } \xi = \tau \\ \top, \text{ иначе} \end{cases}$
- $\mathrm{NCont}_\tau \vee \mathrm{NCont}_\xi = \begin{cases} \mathrm{NCont}_\tau, \text{ если } \xi \ll \tau \\ \mathrm{NCont}_\xi, \text{ если } \tau \ll \xi \\ \top, \text{ иначе} \end{cases}$
- $\mathrm{ContAll}_\tau \vee\ \xi = \begin{cases} \mathrm{ContAll}_\tau, \text{ если } \exists n(\xi = \tau^n \ \&\ n \geq 0) \\ \top, \text{ иначе} \end{cases}$
- $\mathrm{NCont}_\tau \vee\ \xi = \begin{cases} \mathrm{NCont}_\tau, \text{ если } \neg(\tau \ll \xi) \\ \top, \text{ иначе} \end{cases}$



# Супремум равенства и неравенства

- $\mathrm{NCont}_\tau \vee \mathrm{ContAll}_\xi = \begin{cases} \mathrm{NCont}_\tau, \text{ если } \forall n(\neg(\tau \ll \xi^n)) \\ \top, \text{ иначе} \end{cases}$

Если строка не содержит подстроки $\tau$ или состоит из степеней строк, которые не могут содержать $\tau$, тогда эта строка не содержит подстроки $\tau$.

# Вычисление супремума строк

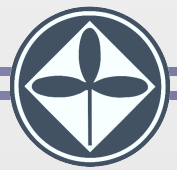
Зададим армейский (military) порядок на строках $\preceq^*_{lex}$ относительно упорядочения букв $\preceq$. Определим функцию «не-делители» $\mathrm{NFact}(s) = \{s' \mid \neg(s' \ll s)\}$.

Тогда $\tau_1 \vee \tau_2 =$

$$\begin{cases} \varepsilon, \text{ если } \tau_1 = \tau_2 = \varepsilon \\ \mathrm{ContAll}_{\tau'}, \text{ если } \exists \tau', n, m(\tau_1 = \tau'^n \ \& \ \tau_2 = \tau'^m \ \& \ n + m > 0) \\ \inf_{\preceq^*_{lex}}(\mathrm{NFact}(\tau_1) \cap \mathrm{NFact}(\tau_2)), \text{ иначе} \end{cases} .$$

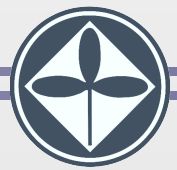
Пусть $a \preceq b$. Пример упорядочения на $\{a, b, aa, ab\}$.

| Армейский | Лексикографический |
|---|---|
| $a$ | $a$ |
| $b$ | $aa$ |
| $aa$ | $ab$ |
| $ab$ | $b$ |

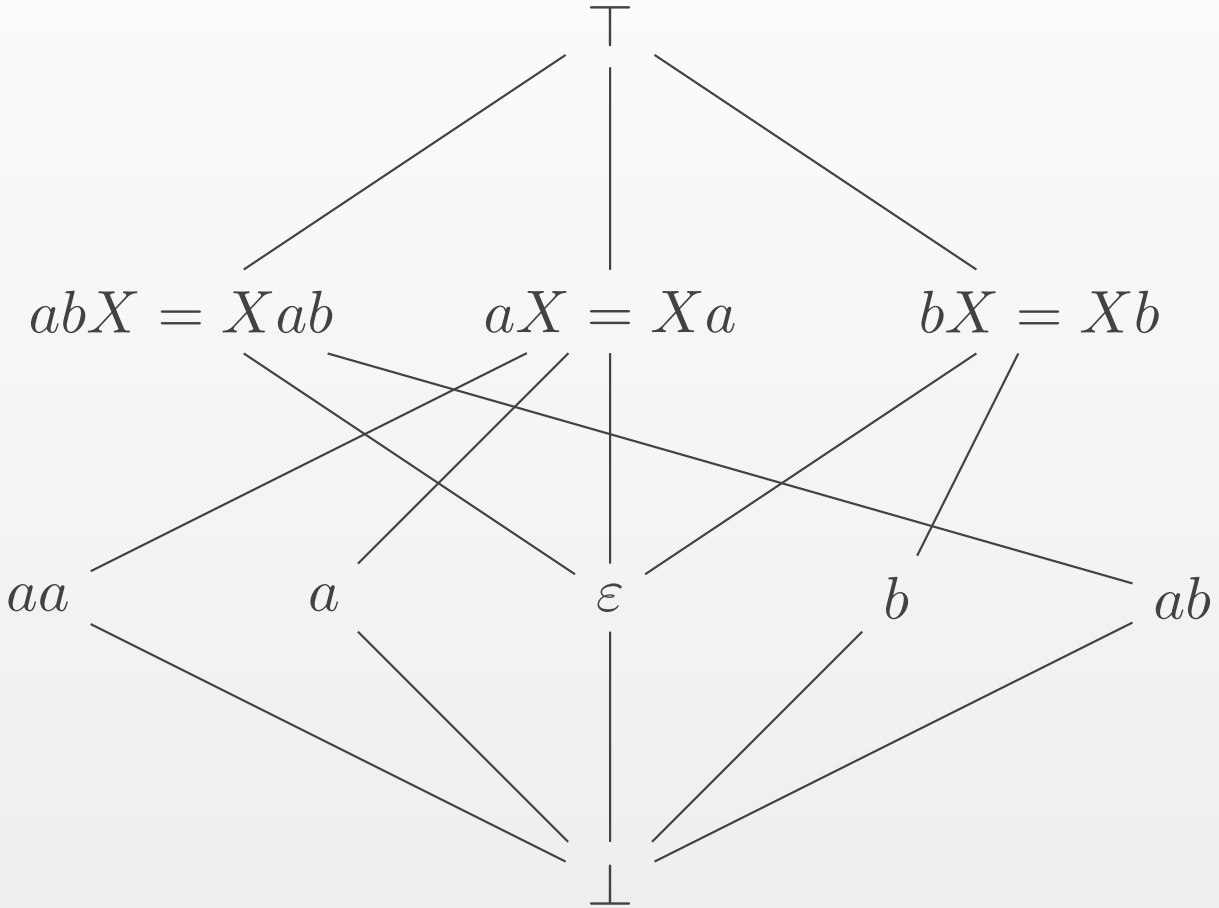


# Проблема инфинума

- Полученная модель — верхняя полурешётка (join-semilattice) конечной высоты. Теоретически, этого достаточно для вычисления неподвижных точек итерацией Клини за конечное время. Практически — зависит от деталей уже реализованного алгоритма.
- Полная решётка получается ограничением условий только на $\mathrm{ContAll}_\tau$. Тогда
  - $\mathrm{ContAll}_\tau \wedge \mathrm{ContAll}_\xi = \begin{cases} \mathrm{ContAll}_\tau, \text{ если } \xi = \tau \\ \varepsilon, \text{ иначе} \end{cases}$
  - $\mathrm{ContAll}_\tau \wedge \xi = \begin{cases} \xi, \text{ если } \exists n(\xi = \tau^n \ \& \ n \geq 0) \\ \bot, \text{ иначе} \end{cases}$
  - $\tau_1 \vee \tau_2 =$
    $$\begin{cases} \varepsilon, \text{ если } \tau_1 = \tau_2 = \varepsilon \\ \mathrm{ContAll}_{\tau'}, \text{ если } \exists \tau', n, m(\tau_1 = \tau'^n \ \& \ \tau_2 = \tau'^m \ \& \ n + m > 0) \\ \top, \text{ иначе} \end{cases}$$



# Пример решётки

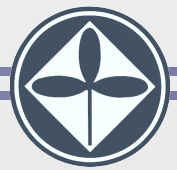


# Набросок алгоритма преобразования

Стандартно полагаем, что $x \leq y \equiv x \vee y = y$

Конкатенация, удаление префикса и взятие подстроки с произвольной позиции:

- $x + y \approx x \vee y$
- $x - y \approx$
$$\begin{cases} x - y, \text{ если } \exists\tau(x = y\tau) \\ \text{ошибка, если } x \text{ и } y \text{ константы, но } \forall\tau(x \neq y\tau) \\ \text{ошибка, если } x = \mathrm{ContAll}_\tau \ \& \ \forall\tau', \tau'', n(y \neq \tau^n\tau' \text{ или } \tau \neq \tau'\tau'') \\ x, \text{ если } y = \mathrm{ContAll}_\tau \text{ и либо } x \text{ константа и } \forall\xi'(x \neq \tau\xi'), \\ \qquad\qquad \text{либо } x = \mathrm{ContAll}_\xi \text{ и } \forall\xi'(\xi \neq \tau\xi') \\ x \vee y, \text{ иначе} \end{cases}$$
- $x = infix(y) \approx \begin{cases} \mathrm{ContAll}_a, \text{ если } y = a \text{ или } y = \mathrm{ContAll}_a \\ \qquad\qquad (a \text{ — буква}) \\ \varepsilon, \text{ если } y = \varepsilon \\ \top, \text{ иначе} \end{cases}$



# Замена подстроки в строке

- $z = replace(y, z_1, z_2) \approx$
$$\begin{cases} replace(y, z_1, z_2), \text{ если } y, z_1, z_2 \text{ константы.} \\ \text{ошибка, если } z_1 = \varepsilon \\ \mathrm{ContAll}_\tau, \text{ если хотя бы одно из } y, z_1, z_2 \text{ равно } \mathrm{ContAll}_\tau \\ \qquad \text{и при этом } y, z_1, z_2 \leq \mathrm{ContAll}_\tau \\ \mathrm{ContAll}_\tau, \text{ если } y = \mathrm{ContAll}_\tau, \& \ (z_1 = \mathrm{ContAll}_\xi \text{ или } z_1 = \xi), \\ \qquad \text{причём } \forall n(\mathrm{NCont}_\xi(\tau^n)) \\ \top, \text{ иначе} \end{cases}$$

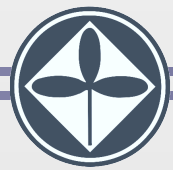
Четвертый случай соответствует невозможности найти хотя бы одну требуемую подстроку в первом аргументе. Если $z_1 = \mathrm{ContAll}_\xi$, тогда последнее условие в четвертом случае вырождается в $\mathrm{NCont}_\xi(\tau)$.

# Предикаты (на примере метода startsWith)

- $prefix(x, y)(x$ является префиксом $y) \approx$

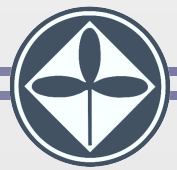
$$\begin{cases} (x = \tau \text{ OR } x = \varepsilon), \text{ если } y = \tau,\ x = \mathrm{ContAll}_{\tau} \\ |x| \le |y|, \text{ если } x = y = \mathrm{ContAll}_{\tau} \\ x = \varepsilon, \text{ если } y = \varepsilon \\ true, \text{ если } \exists \xi (y = x\xi) \\ false, \text{ если } x, y \text{ константы и } \forall \xi (y \neq x\xi) \\ (x = \tau_1 \text{ OR } x = \varepsilon), \text{ если } x = \mathrm{ContAll}_{\tau_1},\ y = \mathrm{ContAll}_{\tau_2}, \\ \quad \text{и } \exists \tau_3 (\tau_3 \neq \varepsilon \ \&\ \tau_2 = \tau_1 \tau_3) \\ x = \varepsilon, \text{ если } x = \mathrm{ContAll}_{\tau_1} \ \&\ (y = \mathrm{ContAll}_{\tau_2} \text{ или } y = \tau_2) \\ \quad \&\ \forall \tau_3 (\tau_2 \neq \tau_1 \tau_3) \\ prefix(x, y), \text{ иначе} \end{cases}$$

Предикаты преобразуются в более широком языке, включающем преобразования между типами и булевские операции. Чтобы не путать операцию супремума и логическое «или», для второго здесь используем код OR.



# Разбор примера

| | |
|---|---|
| z = $\xi$ | $z \to \xi$ |
| x = $\varepsilon$ | $x \to \varepsilon$ |
| y = $\varepsilon$ | $y \to \varepsilon$ |
| while (cond1) | while (cond1) |
| x+=z | $x \to \xi x = x\xi$ $(\varepsilon \vee \xi = \mathrm{ContAll}_\xi = \mathrm{ContAll}_\xi \vee \xi)$ |
| n while (cond2) | while (cond2) |
| y+=z | $y \to \xi y = y\xi$ $(\varepsilon \vee \xi = \mathrm{ContAll}_\xi = \mathrm{ContAll}_\xi \vee \xi)$ |
| while (true) | while (true) |
| if (prefix(x,y)) | if $(\lvert x\rvert \leq \lvert y\rvert)$ $(x = y = \mathrm{ContAll}_\xi)$ |
| y-=x | $y \to \xi y = y\xi$ $(\mathrm{ContAll}_\xi \vee \mathrm{ContAll}_\xi = \mathrm{ContAll}_\xi)$ |
| elif (prefix(y,x)) | elif $(\lvert y\rvert \leq \lvert x\rvert)$ $(x = y = \mathrm{ContAll}_{xi})$ |
| x-=y | $x \to \xi x = x\xi$ $(\mathrm{ContAll}_\xi \vee \mathrm{ContAll}_\xi = \mathrm{ContAll}_\xi)$ |
| else break | else break |



# Специализация по гомоморфизму

- Операции $\vee$, $\wedge$ могут быть перенесены на гомоморфные образы строк и условий по гомоморфизму $h$. Например, это позволяет отделить абстрактное значение «строка, представляющая число», от прочих строк.
- Аналогичные свойства выполняются для обратного гомоморфизма. В частности, это позволяет отделить абстрактное значение «строка, представляющая ключевое слово», от прочих строк.
- Комбинация двух подходов позволит обрабатывать сложные случаи: число, которое является ключевым словом (и не состоит из цифр) (в JavaScript: Infinitity, -Infinity).