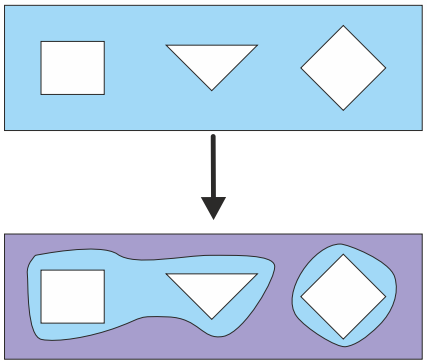
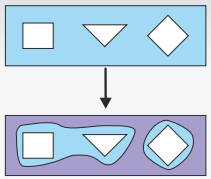


Анализ моделей в теории строк для преобразования и ускорения Рефал-программ

Совместное совещание по языку Рефал
ИПС им. А.К. Айламазяна РАН
и МГТУ им. Н.Э. Баумана
17 июня 2023 г.

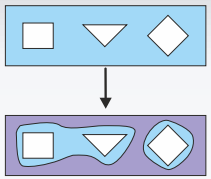


Бесквантворный SMT-фрагмент

- Можно объявлять параметры: `(declare-fun x () String)`.
- Далее параметры связываются допущениями (аксиомами): `(assert (str.contains x "a"))`.

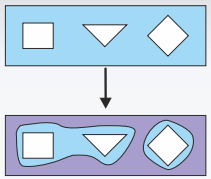
Описывает конечное число пропозициональных формул над операторами и предикатами.

- Существование модели — поиск контрпримеров, атак, ошибок (например, в конколической интерпретации или фаззинге).
- Несуществование модели — возможные оптимизации и верификация безопасности.



Теория строк: базовые структуры

- Основная операция — конкатенация, базовый предикат — равенство.
  - $X_1 ++ X_2 ++ \cdots ++ X_n$, X_i могут включать параметры.
  - Экзистенциальная теория — теория уравнений в словах.
- Дополнительный предикат — отношение подстроки $X_2 \preceq X_1$.
- $X_2 \preceq X_1$ — выразим через экзистенциальную теорию уравнений в словах.
- $\neg\, X_2 \preceq X_1$ — не выразим через равенство.
- Дополнительная операция — замена подстроки в строке $replace(X_1, X_2 \mapsto X_3)$.
- Естественное расширение — предикат вхождения слова в регулярный язык.
  - Не ломает разрешимость экзистенциальной теории уравнений в словах.
  - Позволяет выразить отрицания для отношений подстроки, но только при константном втором аргументе.

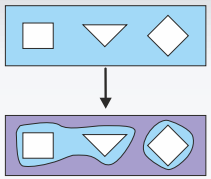


Противоречивые строковые модели

Эксперимент: порождение малых случайных противоречивых моделей в теории строк.

- 3–5 строковых параметра;
- 3–15 аксиом;
- все строки в операторе замены и в предикате проверки на подстроку константны и имеют длину не больше 5;
- отсеиваются тривиальные противоречия (такие, как несовпадения префиксов констант в уравнениях).

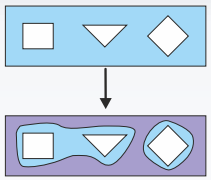
Противоречивость от 20% до 30% таких малых моделей не определяется солверами cvc5 и Z3!



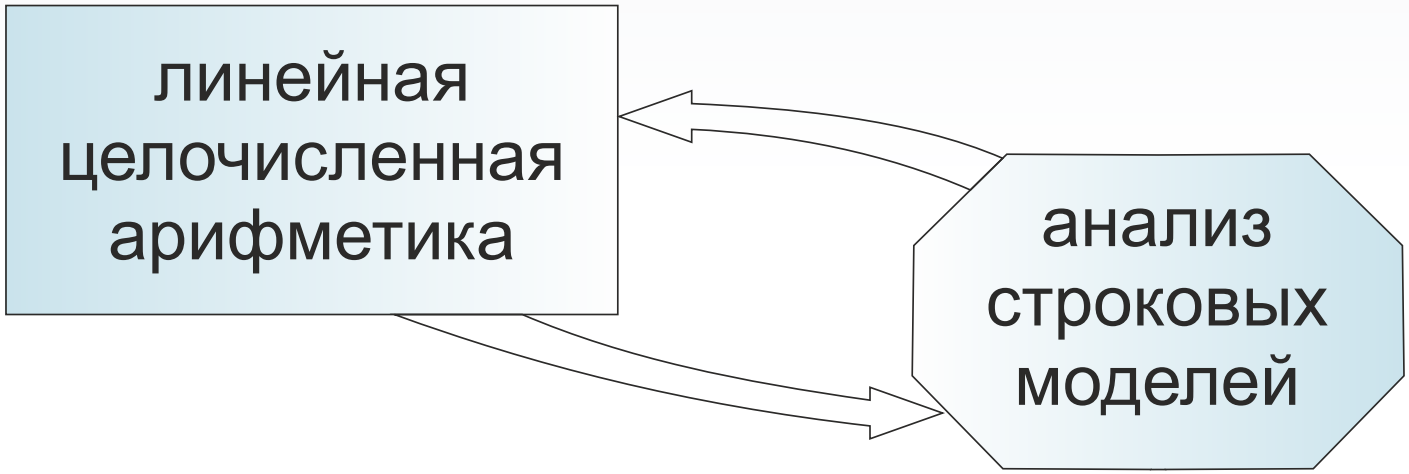
Теории и сложность анализа

Здесь $\mathcal{EST}$ — экзистенциальная теория строк (или теория уравнений в словах).

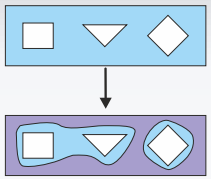
Теория	Replace Const	All ∀	Concat Linear	∀	счёт букв	счёт длин	REGEX	Сложность
$\mathcal{EST}$	✗	✗	✓	✓	✗	✗	✓	PSPACE
$\mathcal{EST}$+len	✗	✗	✓	✓	✗	✓	✗	???
$\mathcal{EST}$+count	✗	✗	✓	✓	✓	✗	✗	Неразр.
$\mathcal{EST}$+repl	✓	✗	✓	✓	✗	✗	✗	Неразр.
repl+SL	✓	✓	✓	✗	✗	✗	✗	EXPSPACE
repl+len	✓	✓	✓	✗	✗	✓	✗	Неразр.
repl+count	✓	✓	✓	✗	✓	✗	✗	Неразр.



Эксперименты с противоречивыми моделями



- Противоречивость условий на длины уравнений (12/50).
- Противоречивость условий на кратность букв (28/50).
- Рекурсивный анализ моделей как программ на языке Рефал (25/50).
- Совместно 2 и 3 (41/50).

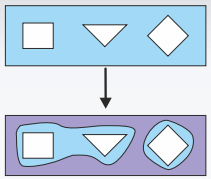


Понятия образца и уравнения

Плоский образец — строка в смешанном алфавите переменных и констант.
Уравнение в словах — равенство между строками в смешанном алфавите.

Скажем, что образец содержит (синтаксически) открытые переменные, если число различных переменных типа выражение, входящих в него, больше 1.

Самобытность Рефала — поддержка образцов с открытыми и кратными переменными типа выражение, а также условий, порождающих уравнения в словах.

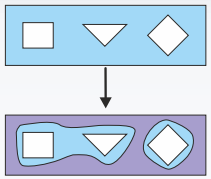


Представление моделей в Рефале

- Логические функции
- Равенство
- Предикат подстроки
- Оператор замены

- Классическое сопоставление слева направо (C)
- Рефал-сопоставление с образцом (O)
- Предварительная специализация (KMP-подобный) (P)

No	&	$X_1 = X_2$	$X_2 \preceq X_1$	$replace(X_1, X_2 \mapsto X_3)$	Успехи
1	C	C	C	C	20
2	C	C	O	O	25
3	O	O	O	O	11
4	O	C	P	P	9
5	O	C	C	C	11



Внутренний язык суперкомпилятора MSCP-A

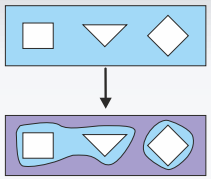
$$\mathcal{C}_i = \langle C_i, P_i \rangle$$

- C_i — параметризованные состояния стека.
- P_i — предикаты на параметры в КНФ, содержащие литералы двух типов.
  - Отрицательные условия в форме $(X_i \neq \Phi_i)$, где Φ_i может содержать как связанные значения (параметры), так и пробегающие все возможные значения (переменные).
  - Уравнения в словах $\Delta_1 = \Delta_2$, где Δ_i могут содержать только параметры, но не переменные.

Каждая конфигурация соответствует модели в теории строк $\Rightarrow$ противоречивые модели влекут отсечение пути вычисления.

D1


создание графа путей
прогонка
прогонка пассивных конфигураций
создание графов уравнений
извлечение теорем
обобщение
извлечение кода



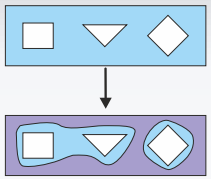
Нётеровость

- Множество путей, порождаемых обобщённым состоянием, включает в себя множества путей обобщаемых состояний. Т.е. определяется возрастание $\mathcal{C}_{k_i} \trianglelefteq \mathcal{C}_g$ по вложению.
- Нет гарантии, что возрастающая последовательность $\mathcal{C}_{g_1} \trianglelefteq \mathcal{C}_{g_2} \trianglelefteq \cdots \trianglelefteq \mathcal{C}_{g_i} \trianglelefteq \ldots$ когда-нибудь стабилизируется.

В алгебрах, содержащих ассоциативные операции, существуют бесконечные возрастающие последовательности, не обладающие нётеровостью, например:

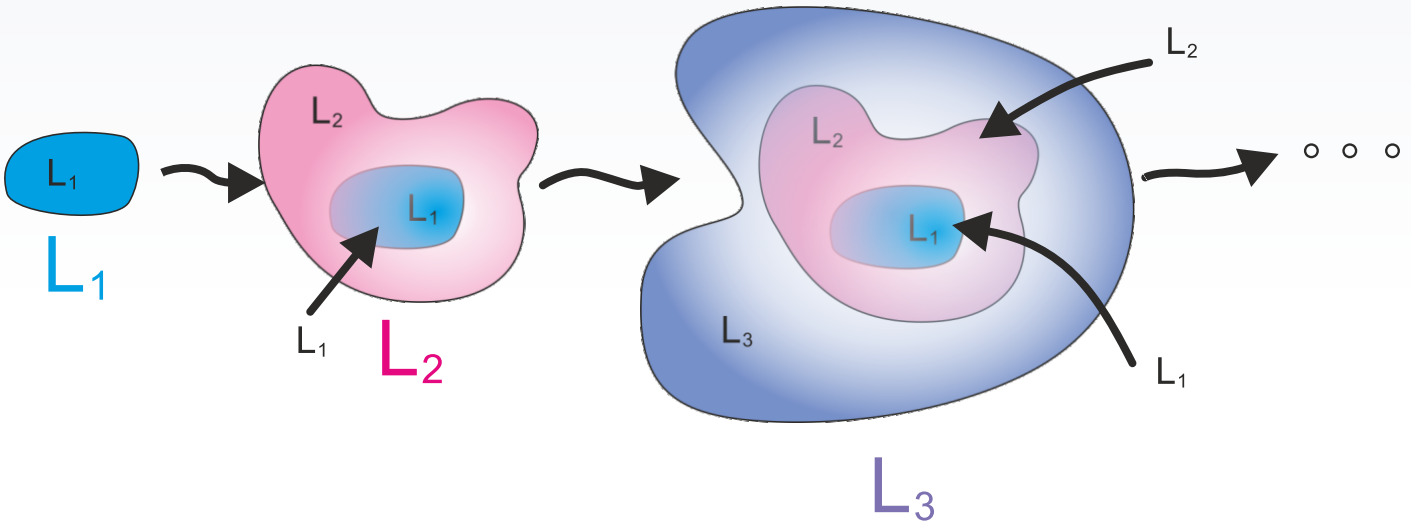
$x_0\,x_0,\ x_1\,x_0\,x_0\,x_1,\ \ldots,\ x_{k+1}\,x_k\,\ldots\,x_0\,x_0\,\ldots\,x_k\,x_{k+1},\ \ldots$

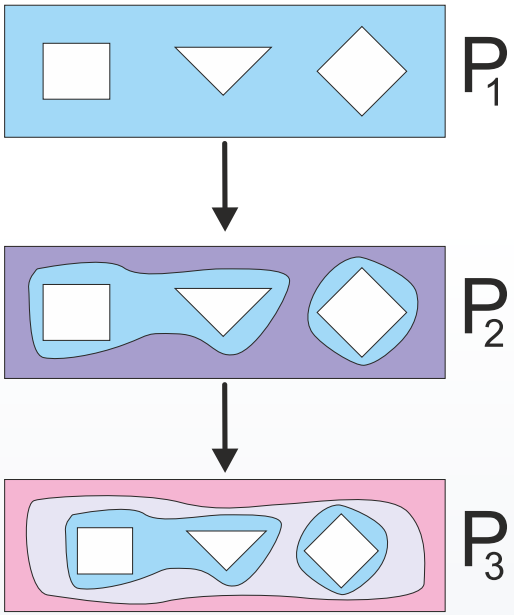
Над данными в свободной алгебре (древесными термами) таких последовательностей не существует, если возрастание $T_i \trianglelefteq T_j$ определяется наличием подстановки σ такой, что $T_j\sigma = T_i$.



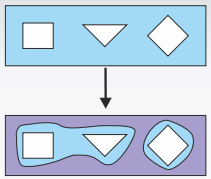
Нётеровость

D3

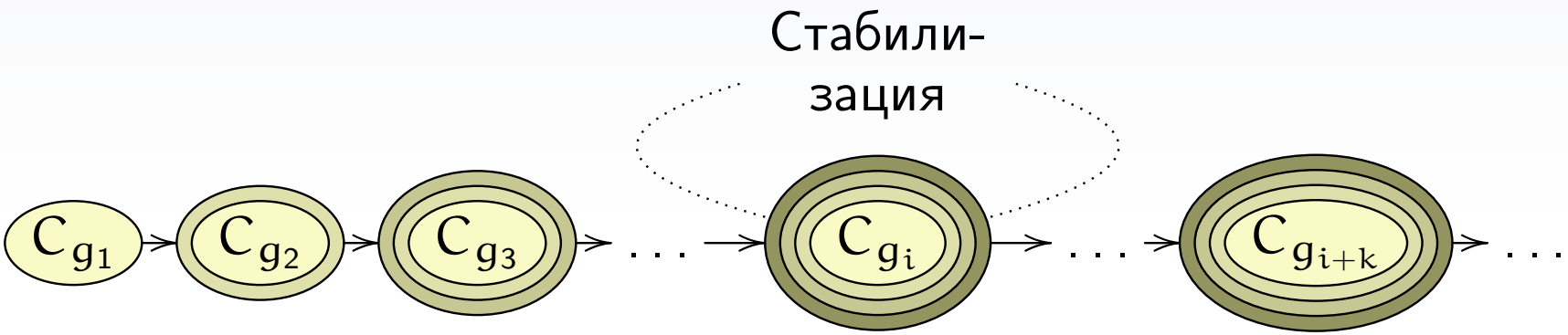




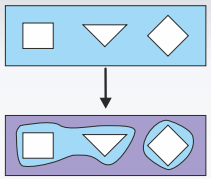
Эквивалентно артиновости (фундированности, Well-Founded-Ordering) обратного отношения ($\mathcal{C}_{k_i}$ к $\mathcal{C}_g$).



Нётеровость



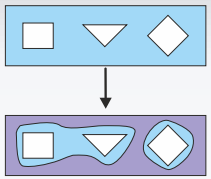
Алгоритм обобщения корректен, если порождаемые им состояния являются нётеровыми относительно отношения вложения путей. Свойство нётеровости гарантирует, что вместо обобщения рано или поздно выполнится вложение.



Пример прошлого года

При обобщении $(\varepsilon)(\mathtt{x})$ и $(\mathbf{A})(\mathbf{A}\,\mathtt{x}')$ получается заготовка $(\mathtt{x}_1)(\mathtt{x}_1\,\mathtt{x}_2)$. После чего параметры $\mathtt{x}_1$ и $\mathtt{x}_2$ сливаются как подряд идущие, даже если на $\mathtt{x}$ и $\mathtt{x}'$ есть рестрикция, запрещающая вхождения буквы $\mathbf{A}$, переносимая на $\mathtt{x}_2$.

Частичное решение: записать уравнение, устанавливающее связь между переменными после слияния — оказывается слабым, потому что негативные рестрикции всё равно теряются.

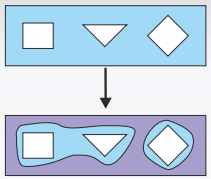


n-замкнутость

Пусть $\mathscr{P}$ — образец (выражение); $P_1,\ldots,P_n$ — свободные фрагменты элементов его плоского разбиения (далее кратко СФР).

- Если x_j — единственная переменная, входящая в некоторый P_k, тогда x_j — 0-замкнутая.
- Если $x_{j_1},\ldots,x_{j_k}$ — j_i-замкнутые переменные, входящие в P_k вместе с некоторой $x_{j_{k+1}}$, степень замкнутости которой неизвестна либо больше $max(j_i)+1$, тогда $x_{j_{k+1}}$ — $max(j_i)+1$-замкнутая.

Дан образец P. n-ку $(t_{n-1},\ldots,t_0)$ такую, что t_i — количество различных переменных замкнутости i, входящих в P, назовём мерой открытости $\mu(P)$.



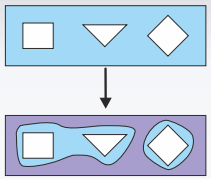
Решения проблемы однозначности

Решение прошлого года гарантировало нётеровость только очень ограниченных классов образцов.

Лемма

Отношение возрастания языков ($\mathscr{L}(\mathrm{P_i}) \subset \mathscr{L}(\mathrm{P_j})$) нётерово для образцов замкнутости $\leqslant \mathrm{k}$.

Доказательство: индукцией по степени замкнутости.



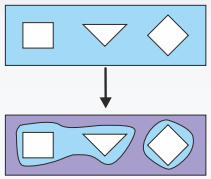
Решения проблемы однозначности

Но как выяснилось, выполняется более сильное утверждение.

Теорема

Отношение возрастания языков ($\mathscr{L}(\mathrm{P_i}) \subset \mathscr{L}(\mathrm{P_j})$) нётерово для образцов, матрица кратности которых имеет ненулевой определитель.

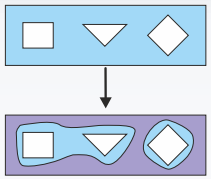
- Ограниченность множества различных переменных $\Rightarrow$ кратности переменных — элементы конечного кортежа;
- Лемма Диксона $\Rightarrow$ найдутся два кортежа, каждый элемент первого из которых не больше каждого из элементов второго. Построим различающую подстановку в «меньший» образец.
- Пустые подстановки $\Rightarrow$ линейная зависимость между уравнениями на длины переменных.
- Нет пустых подстановок $\Rightarrow$ возможна лишь переименовка.



Строковые модели и сопоставление в Рефале

В конфигурациях суперкомпилятора могут возникать почти произвольные условия (большое число повторных вхождений параметров, из-за особенностей прогонки).

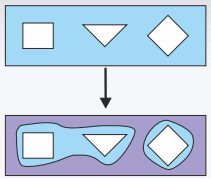
Однако в самих образцах редко появляется больше трёх повторных переменных.



Строковые модели и сопоставление в Рефале

Что встречается:

- пассивные условия (уравнения) вида $\mathrm{x}_1 = \mathrm{z}_1\mathrm{x}_1\mathrm{z}_2 \ldots \mathrm{x}_n\mathrm{z}_{n+1}$, где z_i — свежие переменные. Чаще $\mathrm{i} = 1$. Таких условий может быть несколько.
- образцы с повторными переменными в разном контексте, «плавающими» внутри открытых. $\mathrm{z}_1\Phi_1\mathrm{x}_1\Phi_2\mathrm{z}_2\Phi_3\mathrm{x}_1\Phi_4\mathrm{z}_3$ — чаще всего повторение однократное.
- образцы, состоящие из нескольких фрагментов, погружённых в открытые переменные. Пример: $(\mathrm{x}_1)(\mathrm{x}_3(\mathrm{x}_2)\mathrm{x}_4)\mathrm{z}_1\mathrm{x}_1\mathrm{x}_2\mathrm{z}_2$



Резюме

- Анализ программ на языке Рефал $\Rightarrow$ анализ строковых моделей; суперкомпилятор $\Rightarrow$ внутренний SMT-солвер в модели SLIA $\Rightarrow$ снова суперкомпилятор.
- Эффективные Рефал-образцы и Рефал-условия $\Rightarrow$ стремительная разработка достаточно эффективных программ на Рефале.
- Использование мощного сопоставления с образцом $\Rightarrow$ анализ простых строковых моделей.