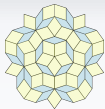


## Структуры над словами: образцы и уравнения

---

Летняя практика, Переславль-Залесский  
*3–6 июля, 2022 г.*



## Проектирование структур с образцами

- Вопрос достижимости образца:

$f (A : x) = \text{Expr1}$

$f [] = \text{Expr2}$

$f [A] = \text{Expr3}$

- Вопрос накрытия образцами:

$f ((x : y) : z) = \text{Expr1}$

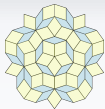
$f [] = \text{Expr2}$

- Вопрос перестановочности образцов:

$f (x : (A : y)) = \text{Expr1}$

$f (A : (y : z)) = \text{Expr2}$

...а с отказом от свободы и единственности вхождений переменных эти вопросы становятся намного сложнее.



## Проектирование структур с образцами

- Вопрос достижимости образца:

$f \{x \ t \ t \ y = \text{Expr1} \}$

$f \{x \ 'A' \ t \ z1 \ 'A' \ y = \text{Expr2} \}$

$f \{x \ 'A' \ y \ 'A' \ z = \text{Expr3} \}$

- Вопрос накрытия образцами:

$f \{x1 \ (z) \ x2 \ t \ x3 \ = \text{Expr1}\}$

$f \{x \ (z) \ = \text{Expr2}\}$

$f \{t \ x \ = \text{Expr3}\}$

- Вопрос перестановочности образцов:

$f \{x, \ x \ 'A' : 'A' \ x = \text{Expr1}\}$

$f \ (x, \ x \ 'AB' : 'BA' \ x = \text{Expr2})$

Необходимо определить выразительную силу образцов — языки, которые они описывают, и свойства этих языков.



## Базовые определения

$V_{\mathcal{T}}$  — множество переменных типа  $\mathcal{T}$ ,  $V = \bigcup^{\mathcal{T}} V_{\mathcal{T}}$ .

Рассматриваем е-переменные (типа строка/выражение) и т-переменные (типа терм). Т.е.  $V = V_e \cup V_t$ .

Кратность терма  $T$  в образце  $P$  обозначаем  $|P|_T$ .

$\Sigma$  — по умолчанию неограниченный алфавит констант.  $\mathcal{B}[S]$  — множество скобочных структур над строками из  $S$ .

Плоский образец  $P$  — строка в алфавите  $V_e \cup \mathcal{B}[\Sigma \cup V_t]$ .  
Образец  $P$  линейен, если  $\forall x \in V_e (|P|_x = 1)$ . Подстановка в образец — гомоморфизм, сохраняющий константы (т.е. для всех  $\mathbf{A} \in \Sigma \mathbf{A}\sigma = \mathbf{A}$ ).

Образец допускает плоское разбиение, если он плоский, либо имеет вид  $(P_1) (P_2) \dots (P_n) P_{n+1}$ , где все  $P_i$  допускают плоское разбиение. Максимальные плоские подобразцы такого образца называем фрагментами плоского разбиения (ФПР).

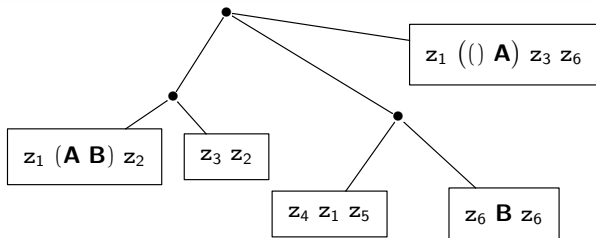


# Плоские разбиения и деревья

Рассмотрим следующий образец:

$$\left( (z_1 \ (\mathbf{A} \ \mathbf{B}) \ z_2) \ z_3 \ z_2 \right) \left( (z_4 \ z_1 \ z_5) \ z_6 \ \mathbf{B} \ z_6 \right) z_1 \ (\ () \ \mathbf{A}) \ z_3 \ z_6$$

Структура его ФПР приведена ниже.



Поскольку скобочные структуры могут возникнуть только сразу справа от открывающей скобки, то ФПР образуют древесные структуры, аналогичные АД.

Пример образца, не разбиваемого на ФПР:

$$\begin{pmatrix} \mathbf{x}_1 & (\mathbf{A} \mathbf{x}_2) & \mathbf{x}_1 & \mathbf{x}_2 \end{pmatrix} \mathbf{x}_1$$



## Языки, распознаваемые образцами

### Определение

*Языком  $\mathcal{L}(P)$ , распознаваемым образцом  $P$ , назовем множество элементов  $\Phi \in \mathcal{B}[\Sigma]^*$ , для которых существует подстановка  $\sigma: P\sigma = \Phi$ . Образец  $P_1$  сводится к образцу  $P_2$ , если  $\mathcal{L}(P_1) \subseteq \mathcal{L}(P_2)$ .*

Подстановка  $x\sigma = \varepsilon$  допустима! В терминологии pattern languages — рассматриваются E-pattern languages (EPL, сокращение от Erasing Pattern Languages, языки стирающих образцов).

- Язык, распознаваемый образцом-строкой  $P \in \Sigma^*$ , есть  $\{P\}$ .
- Язык, распознаваемый образцом  $P = x_1 x_2 x_1$ , есть всё множество  $\mathcal{B}[\Sigma]^*$ .



## Языки, распознаваемые образцами

### Определение

*Языком  $\mathcal{L}(P)$ , распознаваемым образцом  $P$ , назовем множество элементов  $\Phi \in \mathcal{B}[\Sigma]^*$ , для которых существует подстановка  $\sigma: P\sigma = \Phi$ . Образец  $P_1$  сводится к образцу  $P_2$ , если  $\mathcal{L}(P_1) \subseteq \mathcal{L}(P_2)$ .*

- Язык, распознаваемый образцом-строкой  $P \in \Sigma^*$ , есть  $\{P\}$ .
- Язык, распознаваемый образцом  $P = x_1 x_2 x_1$ , есть всё множество  $\mathcal{B}[\Sigma]^*$ .

С точки зрения семантики сопоставления, образец  $x_1 x_2 x_1$  также неудачный:  $x_1$  всегда успешно сопоставляется с  $\varepsilon$ .

Бывает и иначе: хотя  $\mathcal{L}(z_1 z_2 z_2) = \mathcal{L}(x_1 x_2 x_1) = \mathcal{B}[\Sigma]^*$  из-за существования тривиальной подстановки  $z_2 := \varepsilon$ , но ленивое сопоставление строки **ABV** с  $z_1 z_2 z_2$  построит подстановку  $z_2 := \mathbf{B}$ , а вовсе не  $z_2 := \varepsilon$ .



## Сводимость и эквивалентность

Если  $P_1, P_2$  оба из  $(V_e \cup \mathcal{B}[\Sigma])^*$ , то:

- $P_1$  сводится к  $P_2 \Leftrightarrow$  существует подстановка  $\sigma$  такая, что  $P_2\sigma = P_1$ ;
- если  $P_2$  линейен, тогда вычислительная сложность проверки сводимости образца  $P_1$  к образцу  $P_2$  линейна от суммы длин  $P_1$  и  $P_2$ .

Из-за того, что образцы стирающие (определяют EPL), двухсторонняя сводимость не эквивалентна наличию переименовки: вспомним те же  $x_1 x_2 x_1$  и  $z_1 z_2 z_2$ .





## Сводимость и эквивалентность

Если  $P_1, P_2$  оба из  $(V_e \cup \mathcal{B}[\Sigma])^*$ , то:

- $P_1$  сводится к  $P_2 \Leftrightarrow$  существует подстановка  $\sigma$  такая, что  $P_2\sigma = P_1$ ;
- если  $P_2$  линейен, тогда вычислительная сложность проверки сводимости образца  $P_1$  к образцу  $P_2$  линейна от суммы длин  $P_1$  и  $P_2$ .

Если рассматривать только образцы без идущих подряд переменных из  $V_e$ , тогда уже для образцов без переменных из  $V_t$  выполняется утверждение

$$\mathcal{L}(P_1) = \mathcal{L}(P_2) \Leftrightarrow \exists \sigma (P_1\sigma = P_2 \ \& \ \forall x \in V_e (x\sigma \in V_e))$$



## Краткие и избыточные образцы

### Определение

*Образец  $P_1$  называется кратким, если любой образец  $P_2$  такой, что  $\mathcal{L}(P_1) = \mathcal{L}(P_2)$ , имеет длину, не меньшую, чем  $P_1$ .*

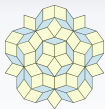
*Иначе  $P_1$  называется избыточным.*

### Пример

Образец  $x_1 \ x_2 \ A \ x_3 \ B \ x_1 \ x_2$  избыточен.

Образец  $x_1 \ x_2 \ x_2 \ x_1$  является кратким.

Алгебраисты также говорят, что избыточные образцы определяются нетривиальными неподвижными точками морфизмов над образцами (т.е. существует нетривиальная подстановка, переводящая избыточный образец в себя).



## Критерий избыточности образца (Reidenbach, 2004)

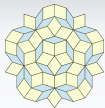
Образец  $P$  избыточен, если существует представление  $P = Q_0 R_1 Q_1 \dots R_n Q_n$ ,  $Q_i \in \{\mathcal{B}[\Sigma] \cup V_e\}^*$ ,  $R_i \in V_e^+ V_e^+$ , такое, что:

- множества переменных образцов  $Q_i$  и  $R_j$  не пересекаются;
- в каждом слове  $R_i$  найдется имеющая единственное вхождение в  $R_i$  переменная  $x_i$  (выделенная) такая, что

$$\forall j (|R_j|_{x_i} > 0 \Rightarrow R_j = R_i).$$

Этот критерий является необходимым и достаточным условием при рассмотрении плоских образцов в  $\{\mathcal{B}[\Sigma] \cup V_e\}^*$ <sup>a</sup>.

<sup>a</sup>У Рейденбаха он доказан для образцов в  $V_e^*$ . Для образцов над  $\mathcal{B}[\Sigma]$  доказательство где-то в моих старых тетрадях — здесь существенно, что скобки порождают бесконечный «алфавит констант».



## Критерий Рейденбаха под лупой

Пусть искомое разбиение образца  $P$  существует. Тогда по каждому блоку  $R_i$  построим подстановку  $\sigma_{\text{fix}}$  так:  $x_i \sigma_{\text{fix}} = R_i$ , а образы прочих переменных из  $R_i$  равны  $\varepsilon$  (они могут встречаться и в сочетании с другой выделенной переменной  $x_k$  в прочих  $R$ -блоках). Очевидно,  $P \sigma_{\text{fix}} = P$ .

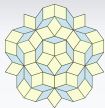
Образец  $P \in V_e^*$  допускает неоднозначную нестирающую подстановку  $\Leftrightarrow$  образец  $P$  избыточен по Рейденбаху.



## Критерий Рейденбаха под лупой

Образец  $P \in V_e^*$  допускает неоднозначную нестирающую подстановку  $\Leftrightarrow$  образец  $P$  избыточен по Рейденбаху.

- Для образцов, содержащих константные фрагменты, это не верно:  $x_1 \mathbf{A} x_2$  допускает много подстановок в  $\mathbf{A}^n$ , хотя является кратким. Однако это верно для фрагментов таких образцов, не содержащих констант.
- Стирающих подстановок может быть и несколько: например,  $x_1 x_2 x_2 x_1$  допускает две подстановки в  $\mathbf{A} \mathbf{B} \mathbf{A} \mathbf{B}$ . Однако поиск возможных подстановок в такой формулировке может быть сделан экспоненциально быстрее, чем без знания о критерии Рейденбаха.



## Добавление переменных типа терм

За увеличение выразительной силы образцов приходится платить усложнением теоретических конструкций.

- $\mathcal{L}(P_1) \subseteq \mathcal{L}(P_2)$  уже не определяется подстановкой.

$P_1 = \mathbf{A} \ x_1, P_2 = x_2 \ t$ . Язык  $P_1$  вкладывается в язык  $P_2$ , а подстановки нет.

- Нет (пока ещё) исследованного понятия избыточного и краткого образца. Более того, образцы для одного и того же языка не образуют нижнюю полурешётку.

Образы  $x \ t$  и  $t \ x$  оба краткие.



## Плавающие $t$ -переменные

### Определение

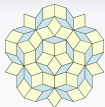
Назовем переменную  $t_i$  в плоском линейном образце  $P$  *якорной*, если

- $t_i$  имеет кратность, не меньшую 2;
- или в  $P$  существует подслово  $\alpha$ , не содержащее переменных из  $V_e$ , такое, что  $\alpha = \alpha_1 t_i \alpha_2$ , причем  $\alpha_1$  и  $\alpha_2$  оба содержат хотя бы один символ или  $t$ -переменную, имеющую кратность не меньше 2.

В противном случае назовем  $t_i$  *плавающей*.

### Пример

Рассмотрим образец  $t_1$   $t_2$   $x_1$   $t_3$   $t_4$   $t_2$   $x_2$   $t_5$ . Якорными переменными являются  $t_2$  и  $t_1$ .



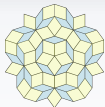
## Плавающие переменные и языки образцов

Плавающая переменная в образце — указатель на то, что в соответствующий фрагмент образца нельзя подставить пустое слово. Аналог «нестираемых» фрагментов.

**Плавающий сегмент** линейного образца  $P$  — максимальное подслово  $P$ , содержащее только плавающие  $t$ -переменные и переменные из  $V_e$ .

Образец, в котором все  $e$ -переменные входят в плавающие сегменты — аналог нестирающего (non-erasing) образца. Хуже всего, если есть и стирающие, и нестирающие фрагменты.



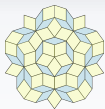


## Плавающие переменные и языки образцов

**Плавающий сегмент** линейного образца  $P$  — максимальное подслово  $P$ , содержащее только плавающие  $t$ -переменные и переменные из  $V_e$ .

Образец, в котором все  $e$ -переменные входят в плавающие сегменты — аналог нестирающего (non-erasing) образца. Хуже всего, если есть и стирающие, и нестирающие фрагменты.

Язык образца  $P_1 = \mathbf{BBA} \, x \, \mathbf{ABCD} \mathbf{A}$  вкладывается в язык образца  $P_2 = z_0 \, t \, t \, z_1 \, t_1 \, t_2 \, t_3 \, t \, z_2$ . Чтобы это доказать, приходится перебирать два случая: пустоты и непустоты подставляемого в  $x$  значения.



# Multi-pattern languages (Kari, Salomaa)

## Определение

*Языком  $\mathcal{L}(P)$ , распознаваемым множеством образцов  $P_i$  (англ. — multi-pattern language, сокращенно MPL), назовем множество элементов  $\Phi \in \mathcal{B}[\text{Sigma}]^*$ , для которых существует  $i \in \mathbb{N}$  и подстановка  $\sigma$ :  $\sigma(P_i) = \Phi$ .*

Множество MPL-объединений стирающих образцов совпадает с множеством MPL-объединений нестирающих образцов. Образец с плавающими  $t$ -переменными тоже определяет мультиобразец, и здесь уже смешивание стирающих и нестирающих фрагментов может быть разрешено.

Однако переход от стирающих образцов к нестирающим порождает экспоненциальное разрастание описания MPL.



## Пример «плавающего» MPL

Пусть  $P_1 = x_1 \mathbf{A A C} x_2 \mathbf{C A B} x_3 \mathbf{B B C}$ ,

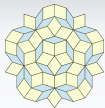
$P_2 = z_1 \mathbf{t_n t_n} z_2 \mathbf{t_{m1} z_3 t_{m2} z_4 t_{m3} z_5 t_n} z_6$ .

Множество нестирающих образцов, порождающих  $P_2$ :

$$\begin{array}{l|l} P_2^1 & t_n t_n z_1 z_2 z_3 t_n \\ P_2^2 & t_n t_n z_1 z_2 z_3 t_n z_4 \\ P_2^3 & z_0 t_n t_n z_1 z_2 z_3 t_n \\ P_2^4 & z_0 t_n t_n z_1 z_2 z_3 t_n z_4 \end{array}$$

Множество нестирающих образцов, порождающих  $P_1$ , и обобщающие их подобразцы из  $P_2$ :

$$\begin{array}{l|l} \mathbf{A A C C A B B B C} & P_2^3 \sigma_1, t_n \sigma_1 = \mathbf{C} \\ \mathbf{A A C C A B} x_3 \mathbf{B B C} & P_2^3 \sigma_2, t_n \sigma_2 = \mathbf{C} \\ \mathbf{A A C} x_2 \mathbf{C A B B B C} & P_2^2 \sigma_3, t_n \sigma_3 = \mathbf{A} \\ \mathbf{A A C} x_2 \mathbf{C A B} x_3 \mathbf{B B C} & P_2^2 \sigma_4, t_n \sigma_4 = \mathbf{A} \\ x_1 \mathbf{A A C C A B B B C} & P_2^3 \sigma_5, t_n \sigma_5 = \mathbf{C} \\ x_1 \mathbf{A A C C A B} x_3 \mathbf{B B C} & P_2^3 \sigma_6, t_n \sigma_6 = \mathbf{C} \\ x_1 \mathbf{A A C} x_2 \mathbf{C A B B B C} & P_2^4 \sigma_7, t_n \sigma_7 = \mathbf{A} \\ x_1 \mathbf{A A C} x_2 \mathbf{C A B} x_3 \mathbf{B B C} & P_2^4 \sigma_8, t_n \sigma_8 = \mathbf{A} \end{array}$$



## Размер алфавита

Все хорошие свойства образцов, позволяющие работать с ними обычными методами (поиск подстановки, разбиение Рейденбаха) — следствие того, что мы подразумеваем  $|\Sigma| = O(|\Sigma_{\text{prog}}|^2)$ , где  $\Sigma$  — алфавит входных данных,  $\Sigma_{\text{prog}}$  — множество символов, явно входящих в образцы. Допущение реалистичное, учитывая, что «буквами» выступают и константные деревья.

Языки образцов  $x \mathbf{A} \mathbf{B} y \mathbf{A} z$  и  $x \mathbf{A} y \mathbf{B} \mathbf{A} z$  в алфавите  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$  очевидно не сравнимы: первый распознаёт слово **ABCA**, второй распознаёт **ACBA**. А в алфавите  $\{\mathbf{A}, \mathbf{B}\}$  эти образцы описывают один и тот же язык<sup>а</sup>.

---

<sup>а</sup>И поэтому, если алфавит входных данных явно присутствует в образцах, нужны другие способы проверки подстановок на однозначность.