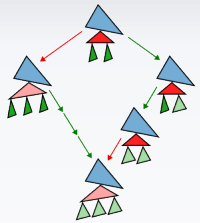
Кодирующие КС-языки

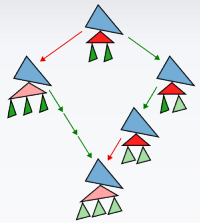
Теория формальных языков
2021 г.



Кодировка путей праволинейной грамматики

Рассмотрим путь вывода произвольного слова $a_1 \dots a_n$ в праволинейной грамматике. Он имеет вид
$S \to a_1 A_1; A_1 \to a_2 A_2; \dots A_n \to a_n$. Применим к нему обратный гомоморфизм $h(A_i; A_i \to) = \varepsilon$ и сотрём префикс $S \to$, получим искомое слово.

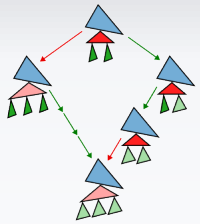
Алфавит: $\Sigma \cup N \cup \{;, \to\}$. Описание языка:
$\{S \to a_i (A_i; A_i \to a_j)^*\}$.



Кодировка путей праволинейной грамматики

Рассмотрим путь вывода произвольного слова $a_1 \dots a_n$ в праволинейной грамматике. Он имеет вид
$S \to a_1 A_1; A_1 \to a_2 A_2; \dots A_n \to a_n$. Применим к нему обратный гомоморфизм $h(A_i; A_i \to) = \varepsilon$ и сотрём префикс $S \to$, получим искомое слово.

Алфавит: $\Sigma \cup N \cup \{;, \to\}$. Описание языка:
$\{S \to a_i (A_i; A_i \to a_j)^*\}$.
Описание языка привязано к множеству нетерминалов в рассматриваемой RLG.



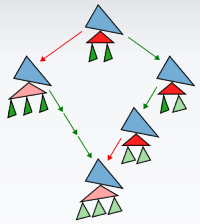
Теорема Хомского–Шутценбергера

Пусть PAREN_n — язык из $4 * n$ элементов $\{[_1,]_1, \ldots, [_n,]_n, (_1,)_1, \ldots, (_n,)_n\}$.

Теорема

Любой CF-язык получается гомоморфизмом из языка $L' = \mathsf{PAREN}_n \cap R$, где R — регулярный.

Пусть G — грамматика L в нормальной форме Хомского. Пронумеруем правила G и поставим им в соответствие следующие.



Теорема Хомского–Шутценбергера

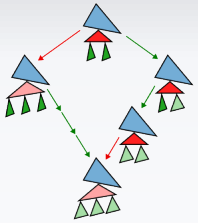
Пусть PAREN_n — язык из $4 * n$ элементов $\{[_1,]_1, \ldots, [_n,]_n, (_1,)_1, \ldots, (_n,)_n\}$.

Теорема

Любой CF-язык получается гомоморфизмом из языка $L' = \mathrm{PAREN}_n \cap R$, где R — регулярный.

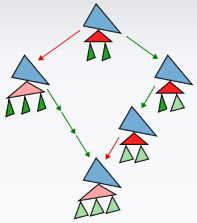
Пусть G — грамматика L в нормальной форме Хомского. Пронумеруем правила G и поставим им в соответствие следующие.

1. Если правило n имеет вид $A \to BC$, тогда порождаем правило $A \to [_n B]_n (_n C)_n$.
2. Если правило n имеет вид $A \to a$, тогда порождаем правило $A \to [_n]_n (_n)_n$.



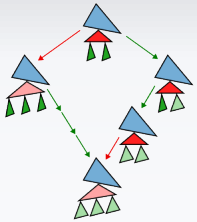
Свойства языка $L(G')$

- Все $]_n$ строго предшествуют $(_n$.



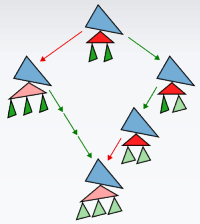
Свойства языка $L(G')$

- Все $]_n$ строго предшествуют $(_n$.
- Ни одна $)_n$ не предшествует непосредственно левой скобке.



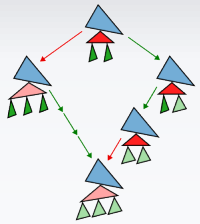
Свойства языка $L(G')$

- Все $]_n$ строго предшествуют $(_n$.
- Ни одна $)_n$ не предшествует непосредственно левой скобке.
- Если правило n — это $A \to BC$, тогда $[_n$ непосредственно предшествует некоторой $[_p$, так же как и $(_n$.



Язык R

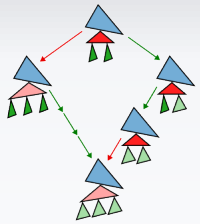
$R = \{x \in \{[_j,]_j, (_j,)_j\}^* \mid x$ начинается с $[_n$ для некоторого правила n : $A \to \cdots \&$ все $]_n$ предшествуют $(_n\}$.



Язык R

$R = \{x \in \{[_j,]_j, (_j,)_j\}^* \mid x$ начинается с $[_n$ для некоторого правила $n : A \to \cdots \&$ все $]_n$ предшествуют $(_n\}$.

Можно убедиться, что $L' = R \cap PAREN_n$.

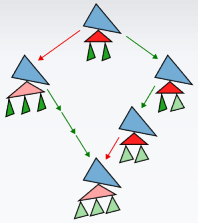


Язык R

$R = \{x \in \{[_j,]_j, (_j,)_j\}^* \mid x$ начинается с $[_n$ для некоторого правила $n : A \to \cdots \&$ все $]_n$ предшествуют $(_n\}$.

Можно убедиться, что $L' = R \cap PAREN_n$.

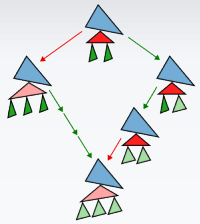
Осталось определить h. Если n — нефинальное правило, то $h([_n) = h(]_n) = h((_n) = h()_n) = \varepsilon$. Иначе $h([_n) = a$, для остальных скобок так же.



Значение теоремы Х.-Ш.

Возможно разделить парсинг любого КС-языка на две стадии: лексический анализ (проверка условия R) и разбор правильных скобочных структур.

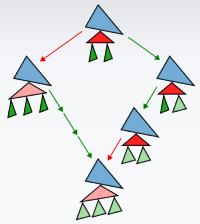
Замечание: поскольку гомоморфизм h не обязан быть инъективным, разбор ПСП не всегда можно определить однозначно. Пример: $\{a^n b^n\} \cup \{a^n b^{2n}\}$ (полностью неоднозначность устранить нельзя, т.к. этот язык не является детерминированным). Однако Т.Х.Ш. даёт подсказку, как строить КС-грамматики: надо найти в языке все скрытые «скобочные структуры».



Построение грамматики по Х.-Ш.

Построить КС-грамматику для языка $\{a^n b^m c^k \mid n = 2 * m - k\}$.

Ищем возможную скобочную структуру. Для этого сначала избавимся от вычитания: $n + k = 2 * m$. Значит, буквы a должны балансироваться буквами b справа (т.е. буквы b являются «закрывающими скобками» для a), а буквы c — буквами b слева (т.е. буквы b являются «открывающими» для c). Возможны два случая: n и k оба чётны либо оба нечётны. Построим соответствующие им разбиения: $\{a^{2*n'} b^{n'} b^{k'} c^{2*k'}\}$ и $\{a a^{2*n'} b^{n'} b b^{k'} c^{2*k'} c\}$. Дальнейшее построение грамматики уже очевидно. Заметим, что гомоморфизм подразумевает минимум четыре вида скобок: пара $(_{2a},)_b$, пара $(_b,)_{2c}$, внешняя пара $[_a,]_c$ (для нечётного варианта) и $[_b,]_\varepsilon$ для него же, чтобы породить внутреннюю букву b.



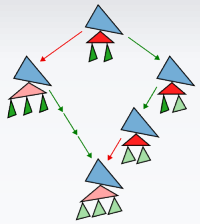
Построение грамматики по Х.-Ш.

Построить КС-грамматику для языка $\{a^n b^m c^k \mid n = 2 * m - k\}$.

Как итог, получаем язык, гомоморфно порождаемый языком Дика над $\{(_{2a},)_b, (_b,)_{2c}, [_b,]_\varepsilon, [_a,]_c\}$ со следующим лексером:

1. До $(_{2a}$ может идти лишь единственная $[_a$.
2. После $)_b$ распознаётся одна $[_b$, если распозналась $[_a$.
3. После $)_b$ или $]_\varepsilon$ не может идти ничего другого, кроме $(_b$ или $]_c$ (последняя — только после $]_\varepsilon$).
4. После $)_{2c}$ не может быть ничего, кроме $)_{2c}$ или $]_c$. Дополнительное условие на существование $[_a$ уже не требуется — оно следует из сбалансированности ПСП.

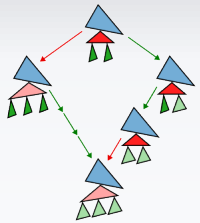
Конструкция выше отличается от используемой в доказательстве теоремы — в целях экономии, в ней почти нет скобок, гомоморфно отображаемых в пустое слово.



Язык Грейбах

Здесь ε-free вариант. D — язык сбалансированных скобочных структур над $\{(,),[,]\}$.

$$L_0 = \{x_1 c y_1 c z_1 d \dots d x_n c y_n c z_n d \mid y_1 \dots y_n \in eD \;\&\; z_i, x_i \text{ не содержат } e \;\&\; y_1 \in e\{(,),[,]\}^* \;\&\; y_{i+1} \in \{(,),[,]\}^*\}$$



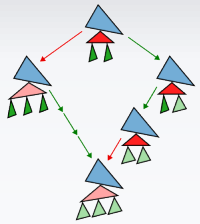
Язык Грейбах

Здесь ε-free вариант. D — язык сбалансированных скобочных структур над $\{(,),[,]\}$.

$L_0 = \{x_1 c y_1 c z_1 d \dots d x_n c y_n c z_n d \mid y_1 \dots y_n \in eD$ &
z_i, x_i не содержат e & $y_1 \in e\{(,),[,]\}^*$ & $y_{i+1} \in \{(,),[,]\}^*\}$

Утверждение

Если L — CFL, тогда существует $h \in \mathrm{Hom}$ такой, что $h^{-1}(L_0) = L$.



Гомоморфизм Грейбах

Пусть G — GNF грамматика для L. Пронумеруем нетерминалы G так, чтобы стартовый был первым. Построим вспомогательную функцию ξ:

- для правил $A_i \to a$ положим $\xi(i) =)]^i)$
- для правил $A_i \to aA_{j1} \dots A_{jn}$ положим $\xi(i) =)]^i)([^{jm}(\dots([^{j1}($
- если $i = 1$, тогда дополнительно припишем префикс $e([($.

Пусть терминалом a начинаются левые части правил $k_1, \dots, k_m$). Тогда $h(a) = c\xi(k_1)c \dots c\xi(k_m)d$.