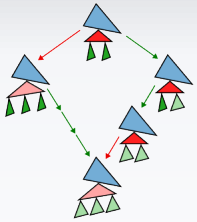
Контекстно-свободные грамматики. Деревья разбора. Нормальная форма Хомского. Алгоритм Кока–Янгера–Касами

Теория формальных языков

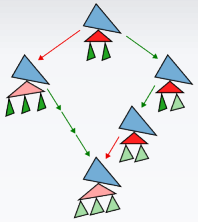
2021 г.



Ограничения регулярных грамматик

- (классы эквивалентности) Префиксы лишь конечно различимы
- (алфавитно-префиксные грамматики) Доступ лишь к началу (концу) слова

Что будет, если дать возможность доступа к переписыванию с середины?

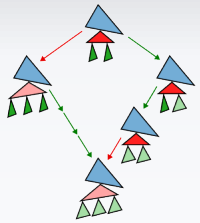


Ограничения регулярных грамматик

- (классы эквивалентности) Префиксы лишь конечно различимы
- (алфавитно-префиксные грамматики) Доступ лишь к началу (концу) слова

Что будет, если дать возможность доступа к переписыванию с середины?

Структура вывода — дерево, а не последовательность.

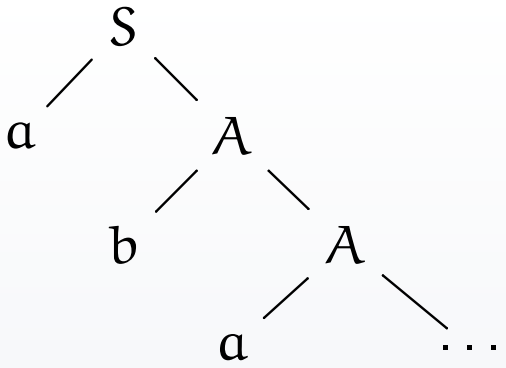


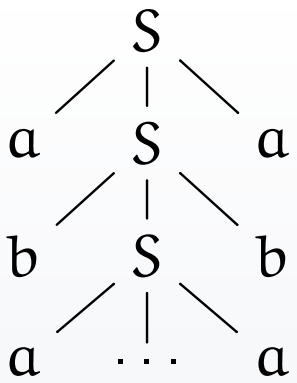
Ограничения регулярных грамматик

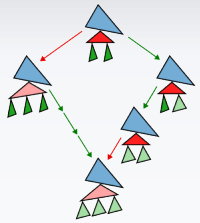
- (классы эквивалентности) Префиксы лишь конечно различимы
- (алфавитно-префиксные грамматики) Доступ лишь к началу (концу) слова

Что будет, если дать возможность доступа к переписыванию с середины?

Структура вывода — дерево, а не последовательность.





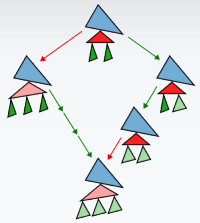


Контекстно-свободные грамматики

Определение

Контекстно-свободная грамматика (CFG) — это грамматика $\langle \Sigma, N, P, S \rangle$, где правила переписывания P имеют вид $A \to \alpha$, $A \in N$, $\alpha \in (\Sigma \cup N)^*$.

- Нетерминалы переписываются независимо друг от друга (можно понимать их как нульместные функции).
- Вывод в грамматике (разбор слова) не линеен.



Контекстно-свободные грамматики

Определение

Контекстно-свободная грамматика (CFG) — это грамматика $\langle \Sigma, N, P, S \rangle$, где правила переписывания P имеют вид $A \to \alpha$, $A \in N$, $\alpha \in (\Sigma \cup N)^*$.

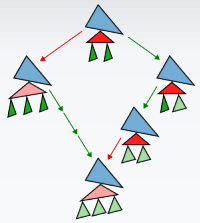
- Нетерминалы переписываются независимо друг от друга (можно понимать их как нульместные функции).
- Вывод в грамматике (разбор слова) не линеен.

Грамматика G_1

$$
\begin{array}{lcl}
S & \to & S\,S \\
S & \to & (S) \\
S & \to & \varepsilon
\end{array}
$$

Грамматика G_2

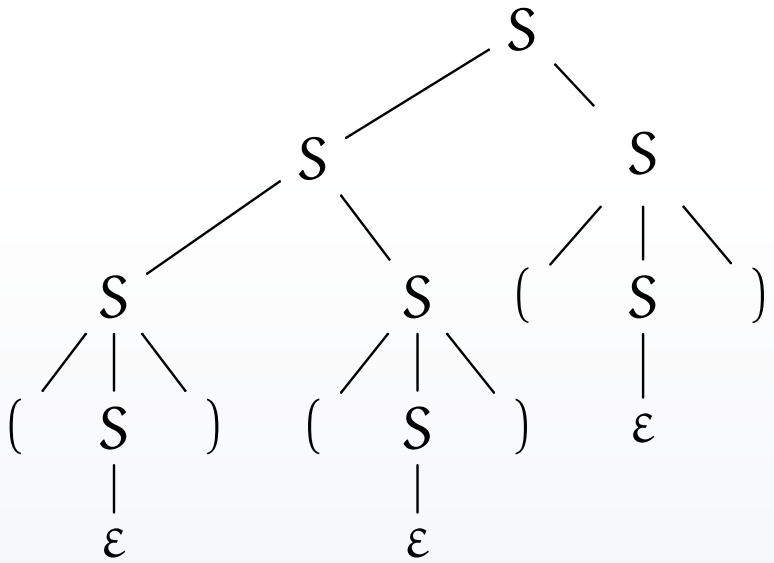
$$
\begin{array}{lcllcl}
S & \to & B & R & \to &) \\
B & \to & (RB & R & \to & (RR \\
B & \to & \varepsilon & & &
\end{array}
$$

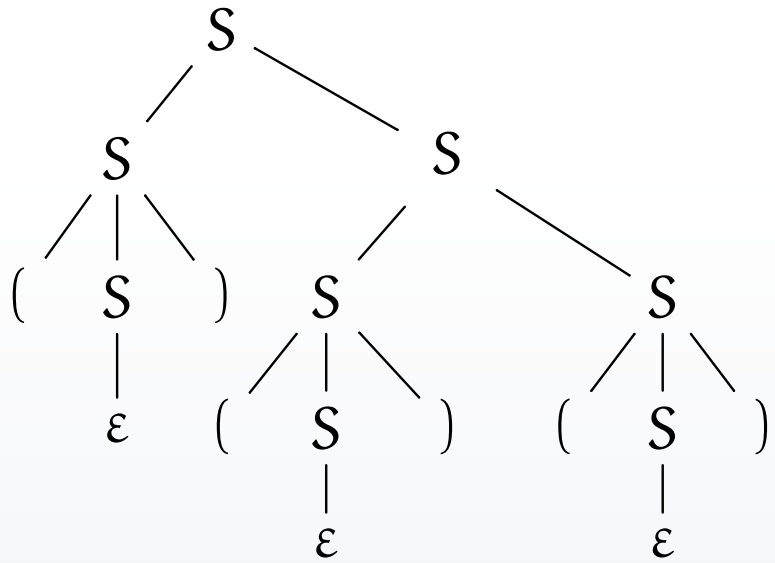


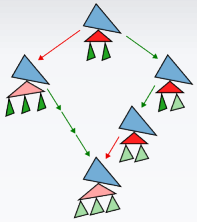
Неоднозначность разбора

Грамматика G_1 для языка Дика

$$
\begin{aligned}
S &\to SS \\
S &\to (S) \\
S &\to \varepsilon
\end{aligned}
$$

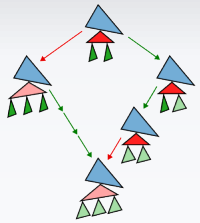






Левосторонний разбор

Шаг левостороннего разбора с.ф. $\alpha_1 A \alpha_2$, где $\alpha_1 \in \Sigma^*$, $A \in N$, — замена выделенного вхождения A на правую часть $A \to \beta$. Левосторонний разбор S — разбор, каждый шаг которого левосторонний.

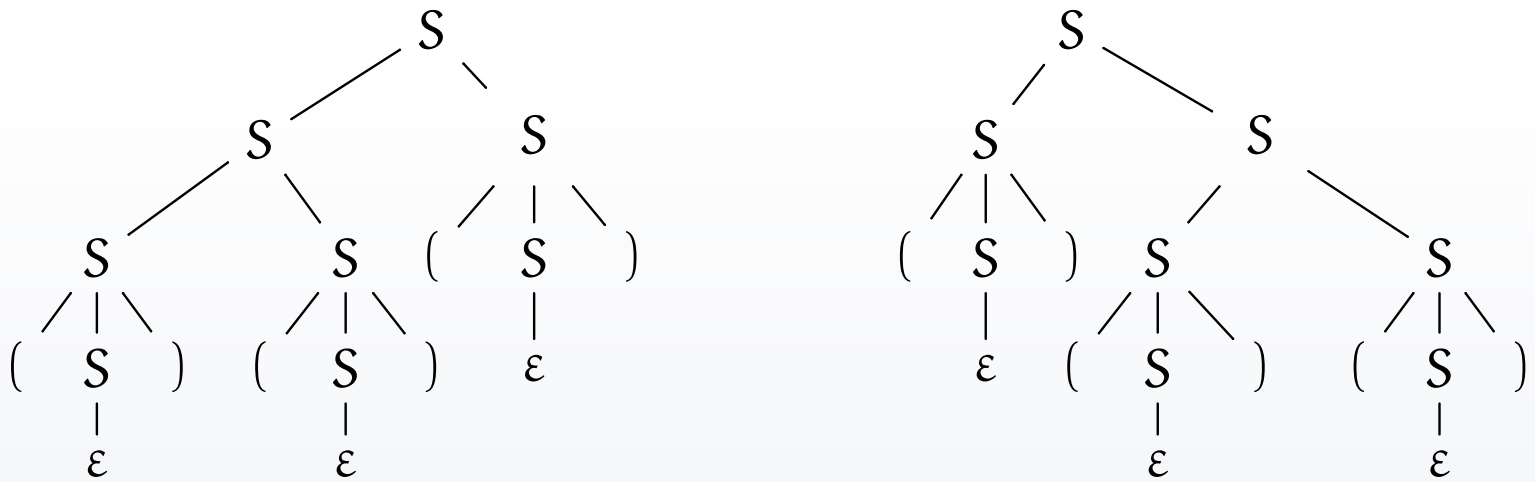


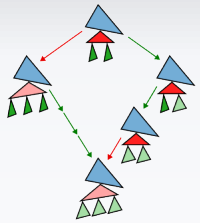
Левосторонний разбор

Шаг левостороннего разбора с.ф. $\alpha_1 A \alpha_2$, где $\alpha_1 \in \Sigma^*$, $A \in N$, — замена выделенного вхождения A на правую часть $A \to \beta$. Левосторонний разбор S — разбор, каждый шаг которого левосторонний.

Левосторонний разбор не обязательно единственный, см. ниже.

$$S \to SS \qquad S \to (S) \qquad S \to \varepsilon$$



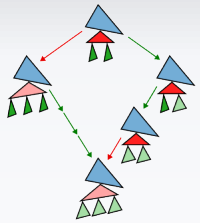


Левосторонний разбор

Шаг левостороннего разбора с.ф. $\alpha_1 A \alpha_2$, где $\alpha_1 \in \Sigma^*$, $A \in N$, — замена выделенного вхождения A на правую часть $A \to \beta$. Левосторонний разбор S — разбор, каждый шаг которого левосторонний.

Утверждение

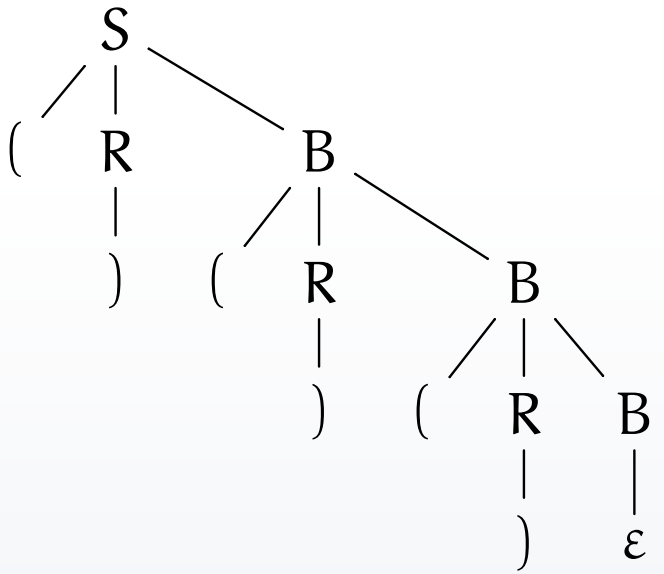
Между деревьями разбора слов $w \in L(G)$ и левосторонними разборами w есть взаимно-однозначное соответствие.



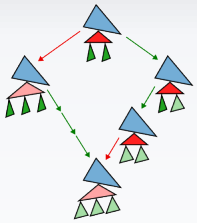
(Не)однозначность грамматик

Грамматика G_2 для языка Дика

S	$\to$	B	R	$\to$	)
B	$\to$	(RB	R	$\to$	(RR
B	$\to$	ε			

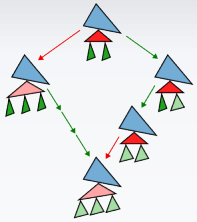


Грамматика G_2 однозначна — для всех $w \in L(G_2)$ существует единственный левосторонний разбор w. Достаточно заглянуть на 1 символ после разобранной позиции.



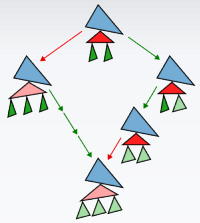
Другие проблемы контекстно-свободного разбора слов

- ε-правила (правила вида $A \to \varepsilon$);
- «ε-переходы», или цепные правила (правила вида $A \to B$).



Устранение ε-правил

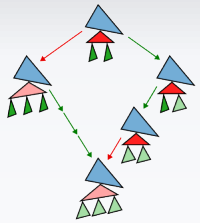
$A \in N$ коллапсирует, если $A \to \varepsilon \in P$ или $A \to \alpha \in P$ и все элементы α коллапсируют.



Устранение ε-правил

$A \in N$ коллапсирует, если $A \to \varepsilon \in P$ или $A \to \alpha \in P$ и все элементы α коллапсируют.

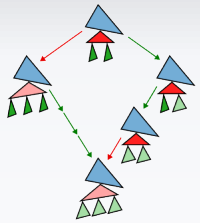
- Объявляем $\texttt{Nullable} = \emptyset$;
- $\forall A \in N$, если $A \to \varepsilon$, тогда $\texttt{Nullable} = \texttt{Nullable} \cup \{A\}$;
- Пока $\texttt{Nullable}$ меняется:
  - для всех $A \in N$, если $A \to B_1 \dots B_n$, $B_i \in \texttt{Nullable}$ $\Rightarrow \texttt{Nullable} = \texttt{Nullable} \cup \{A\}$.
- Итоговое множество $\texttt{Nullable}$ — множество всех коллапсирующих нетерминалов.



Устранение ε-правил

$A \in N$ коллапсирует, если $A \to \varepsilon \in P$ или $A \to \alpha \in P$ и все элементы α коллапсируют.

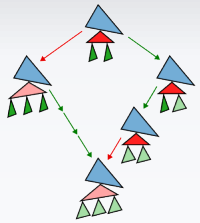
- Если $\varepsilon \in L(G)$, тогда добавляем новый стартовый символ S_0 и правила $S_0 \to \varepsilon$, $S_0 \to S$.
- Стираем все правила $B_i \to \varepsilon$, кроме $S_0 \to \varepsilon$.
- Для всех правил $A \to \alpha_1 B_i \alpha_2$, где $B_i \in \texttt{Nullable}$, добавляем правила $A \to \alpha_1 \alpha_2$.



Устранение ε-правил

$A \in N$ коллапсирует, если $A \to \varepsilon \in P$ или $A \to \alpha \in P$ и все элементы α коллапсируют.

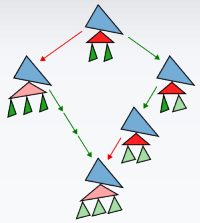
- Если $\varepsilon \in L(G)$, тогда добавляем новый стартовый символ S_0 и правила $S_0 \to \varepsilon$, $S_0 \to S$.
- Стираем все правила $B_i \to \varepsilon$, кроме $S_0 \to \varepsilon$.
- Для всех правил $A \to \alpha_1 B_i \alpha_2$, где $B_i \in \texttt{Nullable}$, добавляем правила $A \to \alpha_1 \alpha_2$. И получаем новые ε-правила! Порядок преобразований существенен.



Устранение ε-правил

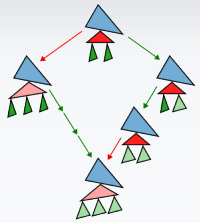
$A \in N$ коллапсирует, если $A \to \varepsilon \in P$ или $A \to \alpha \in P$ и все элементы α коллапсируют.

- Если $\varepsilon \in L(G)$, тогда добавляем новый стартовый символ S_0 и правила $S_0 \to \varepsilon$, $S_0 \to S$.
- Для всех правил $A \to \alpha_1 B_i \alpha_2$ ($|\alpha_1\alpha_2| \geqslant 1$), где $B_i \in \texttt{Nullable}$, добавляем правила $A \to \alpha_1\alpha_2$.
- Стираем все правила $B_i \to \varepsilon$, кроме $S_0 \to \varepsilon$.



Уничтожение цепных правил

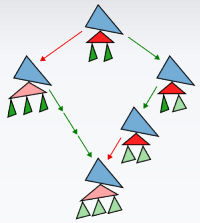
- Строим транзитивное замыкание $A \to_c^* B$ отношения $A \to_c B : A \to B \in P$.
- $\forall A, B : A \to_c B$, строим множество правил $A \to \phi_i$, для которых $\exists C, \phi_i (C \to \phi_i \in P \ \& \ (B \to_c^* C \vee C = B) \ \& \ (|\phi_i| > 1 \vee \phi_i = \varepsilon \vee (\phi_i = a \ \& \ a \in \Sigma)))$.
- Удаляем все правила $A \to B$.



Нормальная форма Хомского

Определение

Грамматика G находится в нормальной форме Хомского (CNF) $\Leftrightarrow$ все её правила имеют вид либо $A \to a$, либо $A \to BC$, либо $S \to \varepsilon$, причём S не входит в правую часть никакого правила из G.

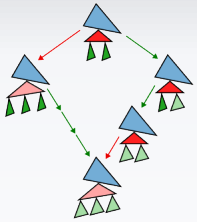


Нормальная форма Хомского

Определение

Грамматика G находится в нормальной форме Хомского (CNF) $\Leftrightarrow$ все её правила имеют вид либо $A \to a$, либо $A \to BC$, либо $S \to \varepsilon$, причём S не входит в правую часть никакого правила из G.

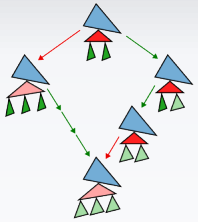
- Устраняем ε-правила.
- Устраняем цепные правила.
- $\forall a \in \Sigma$ таких, что a входит в правую часть правила, отличную от a, заводим нетерминал–охранник G_a, строим правило $G_a \to a$, и во всех правых частях, кроме совпадающих с a, заменяем a на G_a.
- $\forall A \to B_1 \dots B_n$, $n > 2$, вводим новый нетерминал B_{1f} и заменяем $A \to B_1 \dots B_n$ на два правила $A \to B_1 B_{1f}$, $B_{1f} \to B_2 \dots B_n$ (рекурсивно).



Смысл нормальной формы Хомского

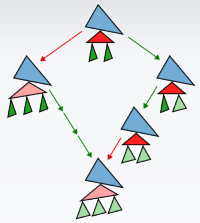
1. Неукорачивающие применения правил
2. Нет пустых переходов — правила либо финальные, либо удлиняющие
3. Контролируемый рост длины сентенциальной формы от количества шагов разбора

Перевод грамматики в CNF позволяет легче анализировать свойства её языка и проводить разбор слов.



Недостижимость и зацикливание

- Стартовый нетерминал $S \in N$ достижим.
- Нетерминал $A \in N$ достижим, если существует правило $B \to \alpha$ такое, что $|\alpha|_A \geqslant 1$ и B достижим.

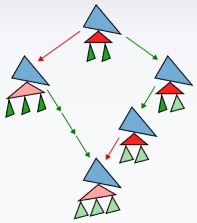


Недостижимость и зацикливание

- Стартовый нетерминал $S \in N$ достижим.
- Нетерминал $A \in N$ достижим, если существует правило $B \to \alpha$ такое, что $|\alpha|_A \geqslant 1$ и B достижим.

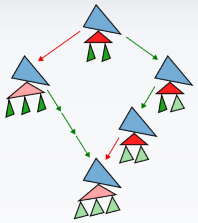
- Если существует правило $A \to w$, $w \in \Sigma^*$, A порождающий.
- Если $A \to \alpha$ и $\forall B_i(|\alpha|_{B_i} \geqslant 1 \Rightarrow B_i$ порождающий$)$, то A порождающий.

1. Удаляем из G все правила, в левых или правых частях которых стоят непорождающие нетерминалы.
2. Удаляем из G все правила, в левых или правых частях которых стоят недостижимые нетерминалы.



Проверка корректности рекурсивных алгоритмов

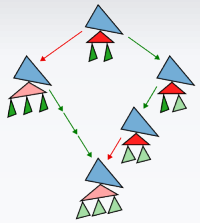
1. Завершаемость — фундированность — искомое множество M нетерминалов не может уменьшаться, и количество нетерминалов грамматики конечно.



Проверка корректности рекурсивных алгоритмов

1. Завершаемость — фундированность — искомое множество M нетерминалов не может уменьшаться, и количество нетерминалов грамматики конечно.

2. Корректность — способ доказательства «*minimal bad sequence*» — пусть существуют элементы $k_i \in M$, которые не находятся рекурсивным алгоритмом. Выберем тот из них, до которого минимальный путь из S (варианты — из которого минимальный путь до Σ^*; до ε). Покажем, что есть ещё какой-то с путём вывода ещё короче.

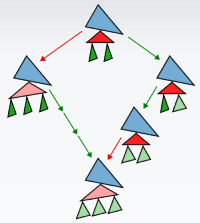


Алгоритм Кока–Янгера–Касами (CYK)

Задача

Дано слово $w_1 \dots w_n \in \Sigma^+$ и грамматика G в CNF.
Проверить, выполнено ли $w \in L(G)$.

Идея алгоритма: переход к более простым задачам порождения подстрок w.



Алгоритм Кока–Янгера–Касами (CYK)

Задача

Дано слово $w_1 \dots w_n \in \Sigma^+$ и грамматика G в CNF. Проверить, выполнено ли $w \in L(G)$.

Идея алгоритма: переход к более простым задачам порождения подстрок w.

Определим функцию $f(A, i, j)$ (где $i \leqslant j$), возвращающую ответ, можно ли вывести слово $w_i \dots w_j$ из $A \in N$.

- Если $i = j$, тогда $f(A, i, j) = T \Leftrightarrow A \to w_i \in P$, и $f(A, i, j) = F$ иначе.
- Если $i < j$, тогда

$$f(A, i, j) = \bigvee_{(A \to BC \in P)} \bigvee_{k=i+1}^{j} (f(B, i, k-1) \,\&\, f(C, k, j)).$$