Леммы о накачке для контекстно-свободных языков. Нормальная форма Грейбах

Теория формальных языков

2021 г.

Высота вывода слова в CFL

Вопрос

Дана КС-грамматика G в CNF (н.ф. Хомского). Дерево разбора какой высоты может соответствовать непустому слову длины $w \in \mathrm{L}(\mathrm{G})$?

Высота вывода слова в CFL

Вопрос

Дана КС-грамматика G в CNF (н.ф. Хомского). Дерево разбора какой высоты может соответствовать непустому слову длины $w \in \mathrm{L}(\mathrm{G})$?

- Максимум: $|w|$ (каждое нефинальное правило увеличивает длину слова хотя бы на 1);

Высота вывода слова в CFL

Вопрос

Дана КС-грамматика G в CNF (н.ф. Хомского). Дерево разбора какой высоты может соответствовать непустому слову длины $w \in L(G)$?

- Максимум: $|w|$ (каждое нефинальное правило увеличивает длину слова хотя бы на 1);
- Минимум: $[\log_2 w] + 1$ (если вывод высоты k порождает максимум слова длины s, тогда вывод высоты $k + 1$ породит максимум слово длины $2 * s$).

Лемма о накачке КС-языков

Лемма о накачке (разрастании)

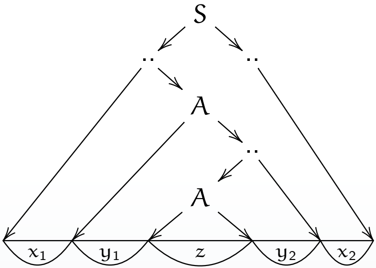
Пусть G — КС-грамматика в форме Хомского. Тогда существует $p \in \mathbb{N}$ такое, что любое слово $w \in L(G)$ длины не меньше p имеет представление вида $x_1 y_1 z y_2 x_2$, где $|y_1 y_2| \geqslant 1$, $|y_1 z y_2| \leqslant p$, и все слова вида $x_1 y_1^k z y_2^k x_2$ также принадлежат $L(G)$.

Лемма о накачке КС-языков

Пусть в н.ф. Хомского G n нетерминалов. Возьмём $p = 2^n$. Его вывод будет иметь минимум высоту $n+1 \Rightarrow$ в нём будет существовать путь, содержащий два одинаковых нетерминала A.

Лемма о накачке КС-языков

Пусть в н.ф. Хомского G n нетерминалов. Возьмём $p = 2^n$. Его вывод будет иметь минимум высоту $n + 1 \Rightarrow$ в нём будет существовать путь, содержащий два одинаковых нетерминала A.



Выберем самые нижние два одинаковых нетерминала $\Rightarrow$ высота поддерева от первого из них не больше $n + 1 \Rightarrow$ длина выводимого слова $y_1 z y_2 \leqslant 2^n$ (т.е. $\leqslant p$). □

Пример применения

Парсинг в Python

Проанализировать язык
$\{a^n z_1 a^n z_2 a^n | n \geqslant 1, |z_i|_a = 0, |z_i| \geqslant 1\}$.

Пример применения

Парсинг в Python

Проанализировать язык $\{a^n z_1 a^n z_2 a^n | n \geqslant 1, |z_i|_a = 0, |z_i| \geqslant 1\}$.

Пусть длина накачки есть p. Рассмотрим слово $a^p b a^p b a^p$. Заметим, что если $y_1 z y_2 = a^i b a^j$ (где i и j могут быть равны 0), тогда $|y_1 y_2|_b = 0$. Действительно, иначе нулевая накачка породит слово $a^m b a^p$, которое не принадлежит языку.

Значит, $y_1 = a^j$, $y_2 = a^i$. Однако слова $a^{p+i+j} b a^p b a^p$, $a^p b a^{p+i+j} b a^p$, $a^p b a^p b a^{p+i+j}$, $a^{p+j} b a^{p+i} b a^p$, $a^p b a^{p+j} b a^{p+i}$ ни одно не принадлежат требуемому языку $\Rightarrow$ он не контекстно-свободен.

Теоретико-игровая интерпретация

Достаточное условие непринадлежности языка L к КС по лемме о накачке: $\forall p \exists w \in L(|w| > p \ \& \ \forall x_i, y_i, z(w = x_1 y_1 z y_2 x_2 \ \& \ |y_1 z y_2| < p \Rightarrow \exists i(x_1 y_1^i z y_2^i x_2 \notin L)))$.

В пренексной форме этого условия кванторы образуют последовательность: $\forall\exists\forall\exists$. Эта последовательность задаёт правила игры, где каждый квантор $\exists$ — ход протагониста, квантор $\forall$ — ход антагониста. Ходы антагониста назначают неопределённые параметры. Ходы протагониста дают выбор известной вам структуры, зависящей от ходов антагониста. В случае леммы о накачке это выглядит так.

- Антагонист выбирает длину накачки p.
- Зная p, протагонист выбирает w.
- Антагонист выбирает разбиение w на пять подстрок.
- Возможно, в зависимости от этого разбиения, протагонист предъявляет i, для которого накачка не выполняется.

Теоретико-игровая интерпретация

- Антагонист выбирает длину накачки p.
- Зная p, протагонист выбирает w.
- Антагонист выбирает разбиение w на пять подстрок.
- Возможно, в зависимости от этого разбиения, протагонист предъявляет i, для которого накачка не выполняется.

Иногда такая система анализа свойств, записанных в виде формул с чередующимися кванторами, также называется игрой Элоизы и Абеляра (по буквам, образующим кванторы $\exists$ и $\forall$).

Техника применения

Сужение перебора

Если в язык L входят подслова произвольной формы из Σ^+, где $|\Sigma| > 1$, тогда, скорее всего, потребуется пересечь L с регулярным языком, чтобы облегчить поиск свидетельства о ненакачиваемости. Пример: язык $\{w_1 w_1 w_2 \mid |w_1|_a = |w_2|_a\}$. Пересечение этого языка с $ba^*bba^*bba^*$ гораздо легче поддаётся анализу, поскольку такие слова разбиваются на подходящие w_1 и w_2 однозначно.

- Начальная буква b вынуждает w_1 содержать ровно две буквы b. Действительно, если $|w_1|_b = 1$, тогда второе вхождение w_1 должно будет начинаться с b^2, что противоречит выбору w_1.
- Последняя буква b навязывает позицию начала w_2.

Техника применения

Работа с отрицанием

Если характеристическая функция L содержит предикат отрицания, связывающий две структуры неопределённого размера, в некоторых случаях это приводит к невозможности применения леммы о накачке. В других можно попробовать воспользоваться приёмом «всё включено». Поскольку мы знаем, что длина накачиваемого фрагмента $y_1 z y_2$ меньше p, то выберем w так, чтобы в нём нашлись всевозможные фрагменты такой длины, удовлетворяющие желательному свойству.

Техника применения

Покажем, что язык $L = \{w \mid w \neq a^{n^2} \;\&\; w \in \{a,b\}^*\}$ не является КС. Для начала заметим, что слова L содержат произвольные подслова в $\{a,b\}^*$, и пересечём L с a^*. Получим $L' = \{a^k \mid k \neq n^2\}$ — если он не КС, то исходный язык также не КС.

- Антагонист выбирает p.
- Наша задача — подобрать такое k, что $\forall p' \exists i, m (p' < p \Rightarrow k + p' * i = m^2)$. То есть включить возможность взятия любого такого p' в наше значение k как конструктивного элемента для построения квадрата числа.
- Возьмём $k = (p!)^2 + 1$. Тогда при любом значении p', меньшем p, можно взять $i = \frac{p!}{p'} * 2$, и получим $k + p' * i = (p! + 1)^2$.

Ещё пример применения

Покажем, что язык $L = \{ww^R a^n \mid |w|_a = n\}$ не является КС. Опять сначала избавимся от произвольных подслов в L и пересечём его с языком $ba^+b^2a^+ba^+$. Пересечение с таким языком вынуждает w иметь вид ba^ib, а весь язык — вид $L' = \{ba^nbba^nba^n\}$.

- Абеляр выбирает p. Элоиза строит слово $ba^pb^2a^pba^p$. Абеляру предоставляется возможность построить его разбиение на $x_1y_1zy_2x_2$.
- Если Абеляр выберет $|y_1|_a > 0$ & $|y_1|_b > 0$ (т.е. y_1 содержащим сразу буквы a и b), тогда ненулевая накачка сразу же выведет нас из языка. Аналогично с y_2.
- Если Абеляр решит накачивать только b (т.е. выберет y_1 либо y_2 равными b или b^2), тогда любая накачка также будет выводить из языка.

Ещё пример применения

Покажем, что язык $L = \{ww^R a^n \mid |w|_a = n\}$ не является КС. Опять сначала избавимся от произвольных подслов в L и пересечём его с языком $ba^+b^2a^+ba^+$. Пересечение с таким языком вынуждает w иметь вид ba^ib, а весь язык — вид $L' = \{ba^nbba^nba^n\}$.

- Абеляр выбирает p. Элоиза строит слово $ba^pb^2a^pba^p$. Абеляру предоставляется возможность построить его разбиение на $x_1y_1zy_2x_2$.
- Остаётся только возможность $y_1 = a^i$, $y_2 = a^j$, что позволяет следующие накачки y_1, y_2 на расстоянии не больше p:
    - $ba^{p+i*k}b^2a^{p+j*k}ba^p$ — можно сохранить свойство палиндрома, но нельзя сохранить корректный подсчёт букв a, последний индекс не меняется.
    - $ba^pb^2a^{p+i*k}ba^{p+j*k}$ — теряется свойство палиндрома.
    - $ba^{p+(i+j)*k}b^2a^pba^p$ — теряется свойство палиндрома, при накачке только второго подслова a^p аналогично.
    - $ba^pb^2a^pba^{p+(i+j)*k}$ — некорректный подсчёт букв a в w.

Языки, накачиваемые обманно

Некоторые не КС-языки тоже накачиваются, например, $\{a^m b^n c^n d^n \mid m > 0\} \cup \{b^i c^j d^k\}$.

Языки, накачиваемые обманно

Некоторые не КС-языки тоже накачиваются, например, $\{a^m b^n c^n d^n \mid m > 0\} \cup \{b^i c^j d^k\}$.

Действительно, если слово языка содержит буквы a, тогда мы можем взять $y_1 y_2 = a^i$. Иначе накачку можно выбрать произвольно.

Языки, накачиваемые обманно

Некоторые не КС-языки тоже накачиваются, например, $\{a^m b^n c^n d^n \mid m > 0\} \cup \{b^i c^j d^k\}$.

Действительно, если слово языка содержит буквы a, тогда мы можем взять $y_1 y_2 = a^i$. Иначе накачку можно выбрать произвольно.

То, что этот язык — не КС, можно понять по тому факту, что его пересечение с регулярным языком $ab^*c^*d^*$ не контекстно-свободно.

Языки, накачиваемые обманно

Некоторые не КС-языки тоже накачиваются, например, $\{a^m b^n c^n d^n \mid m > 0\} \cup \{b^i c^j d^k\}$.

Действительно, если слово языка содержит буквы a, тогда мы можем взять $y_1 y_2 = a^i$. Иначе накачку можно выбрать произвольно.

То, что этот язык — не КС, можно понять по тому факту, что его пересечение с регулярным языком $ab^*c^*d^*$ не контекстно-свободно.

Иногда пересечение с регулярным языком делает язык «излишне накачиваемым»: например, пересекая $L = \{ww^R a^n \mid |w|_a = n\}$ с $ba^+b^*a^+ba^+$, мы даём возможность Абеляру выбрать в качестве y_1 пару букв из центрального блока b^* (положив $y_2 = \varepsilon$). Заметим, что слова без этого блока будут иметь вид $ba^{2n}ba^n$ — а такие слова тоже можно накачивать, выбрав y_1 из a^{2n}, y_2 — из a^n.

Лемма Огдена

Пусть L — КС-язык. Тогда существует такое число n, что в любом слове w, $|w| \geqslant n$, можно отметить n или более букв так, что w представляется в виде $x_1 y_1 z y_2 x_2$, причём либо во всех трех из x_1, y_1, z есть отмеченные буквы, либо они есть во всех трех из z, y_2, x_2, в слове $y_1 z y_2$ отмечено не более n букв, и $\forall k(x_1 y_1^k z y_2^k x_2 \in \mathrm{L})$.

Исследуем «плохой» язык $\{a^m b^n c^n d^n \mid m > 0\} \cup \{b^i c^j d^k\}$ с помощью леммы Огдена. Абеляр (антагонист) выбирает n. Элоиза (т.е. мы) строит слово $ab^{2n}c^{2n}d^{2n}$ и отмечает n последних букв d. Абеляр может разбить слово $ab^{2n}c^{2n}d^{2n}$ на $x_1 y_1 z y_2 x_2$ двумя способами:

- отмечены x_1, y_1, z, накачиваться может только d^{2n}.
- отмечены x_2, y_2, z, накачивается либо d^{2n}, либо d^{2n} совместно с c^{2n}, b^{2n} или a.

Оба типа накачки выводят из языка, поскольку при любой положительной накачке число вхождений букв b или c расходится с числом вхождений d в слово.

Н.ф. Хомского и левосторонний вывод

- Могут быть непродуктивные левосторонние цепочки: $A \to AB \to \ldots AB^n \to \ldots$.
- Есть гарантия роста слова при развёртке, но нет определённости, по какому префиксу.

Нормальная форма Грейбах

Определение

Грамматика G ($\varepsilon \notin L(G)$) находится в GNF (н.ф. Грейбах) $\Leftrightarrow$ каждое её правило имеет вид $A_i \to a_j\alpha$, где $A_i \in N$, $\alpha \in N^*$, $a_j \in \Sigma$.

- Левосторонний разбор по грамматике в GNF на каждом шагу переписывания порождает терминальный символ.

Нормальная форма Грейбах

Определение

Грамматика G ($\varepsilon \notin L(G)$) находится в GNF (н.ф. Грейбах) $\Leftrightarrow$ каждое её правило имеет вид $A_i \to a_j\alpha$, где $A_i \in N$, $\alpha \in N^*$, $a_j \in \Sigma$.

- Левосторонний разбор по грамматике в GNF на каждом шагу переписывания порождает терминальный символ.
- Для приведения к GNF нужно «вытащить из рекурсии» возможные first-терминалы, порождаемые нетерминалами грамматики.

Нормальная форма Грейбах

Определение

Грамматика G ($\varepsilon \notin L(G)$) находится в GNF (н.ф. Грейбах) $\Leftrightarrow$ каждое её правило имеет вид $A_i \to a_j\alpha$, где $A_i \in N$, $\alpha \in N^*$, $a_j \in \Sigma$.

- Левосторонний разбор по грамматике в GNF на каждом шагу переписывания порождает терминальный символ.
- Для приведения к GNF нужно «вытащить из рекурсии» возможные first-терминалы, порождаемые нетерминалами грамматики.
    - явно найти все завершающиеся цепочки вывода;

Нормальная форма Грейбах

Определение

Грамматика G ($\varepsilon \notin L(G)$) находится в GNF (н.ф. Грейбах) $\Leftrightarrow$ каждое её правило имеет вид $A_i \to a_j\alpha$, где $A_i \in N$, $\alpha \in N^*$, $a_j \in \Sigma$.

- Левосторонний разбор по грамматике в GNF на каждом шагу переписывания порождает терминальный символ.
- Для приведения к GNF нужно «вытащить из рекурсии» возможные first-терминалы, порождаемые нетерминалами грамматики.
    - явно найти все завершающиеся цепочки вывода;
    - рассмотреть язык-реверс сентенциальных форм.
- По умолчанию считаем, что к GNF приводится CNF (н.ф. Хомского).

Первый способ приведения к GNF

1. Пронумеровать нетерминалы в правых частях правил в порядке их вхождения;
2. (по исчерпанию) Если имеется правило вида $A_i \to B_j \beta$, где $j < i$, тогда подставить вместо B_j все правые части α_k правил вида $B_j \to \alpha_k$.
3. Если после этого все правила имеют вид либо $A_i \to a\alpha$, $a \in \Sigma$, либо $A_i \to B_j \beta$, причём $i < j$, тогда GNF получается последовательной развёрткой B_j.

Первый способ приведения к GNF

1. Пронумеровать нетерминалы в правых частях правил в порядке их вхождения;
2. (по исчерпанию) Если имеется правило вида $A_i \to B_j\beta$, где $j < i$, тогда подставить вместо B_j все правые части α_k правил вида $B_j \to \alpha_k$.
3. Если после этого все правила имеют вид либо $A_i \to a\alpha$, $a \in \Sigma$, либо $A_i \to B_j\beta$, причём $i < j$, тогда GNF получается последовательной развёрткой B_j. Существует лексикографический порядок на функциональных символах из N.

Первый способ приведения к GNF

1. Пронумеровать нетерминалы в правых частях правил в порядке их вхождения;
2. (по исчерпанию) Если имеется правило вида $A_i \to B_j \beta$, где $j < i$, тогда подставить вместо B_j все правые части α_k правил вида $B_j \to \alpha_k$.
3. Если после этого все правила имеют вид либо $A_i \to a\alpha$, $a \in \Sigma$, либо $A_i \to B_j \beta$, причём $i < j$, тогда GNF получается последовательной развёрткой B_j. Существует лексикографический порядок на функциональных символах из N.
4. Если есть правила вида $A_i \to A_i \alpha$, тогда устраняем левую рекурсию.

Устранение левой рекурсии

1. Предположим, для A_i нашлось n леворекурсивных правил и m упорядоченных лексикографически:

$$
\begin{array}{ll}
A_i \to A_i\alpha_1 & A_i \to \beta_1 \\
\ldots & \ldots \\
A_i \to A_i\alpha_n & A_i \to \beta_m
\end{array}
$$

2. Вводим новый нетерминал A_i' такой, что его вес меньше всех прочих, и заменяем правила на:

$$
\begin{array}{ll}
A_i' \to \alpha_1 A_i' \,|\, \alpha_1 & A_i \to \beta_1 \,|\, \beta_1 A_i' \\
\ldots & \ldots \\
A_i' \to \alpha_n A_i' \,|\, \alpha_n & A_i \to \beta_m \,|\, \beta_m A_i'
\end{array}
$$

3. После всех таких замен грамматика лексикографически упорядочена по левому разбору, и GNF получается последовательной левой развёрткой.

Второй способ приведения к GNF

Алгоритм Блюма–Коха (1999).

Неформальное описание

- Рассмотрим язык сентенциальных форм с переписыванием только по левому разбору. Он регулярен, и в конечное состояние его NFA ведут стрелки, помеченные терминалами.
- Для такого языка легко построить инверсный $\Rightarrow$ множество терминалов-префиксов, которые может породить данный нетерминал.

Второй способ: порождение NFA

1. По каждому нетерминалу B строим автомат $M_B = \langle N_B \cup \{S_B\}, \Sigma \cup N, B_B, \{S_B\}, \delta\rangle$ (S_B — новое состояние, N_B — множество нетерминалов CFG, индексированное нетерминалом B). Правила перехода δ:
   - $\langle C_B, E, M\rangle \Leftrightarrow M = \{D_B \mid C \to DE \in P\}$;
   - $\langle C_B, a, \{S_B\}\rangle \Leftrightarrow C \to a \in P$.
2. Строим реверс к M_B, получаем NFA M_B^R.
3. Строим грамматику $G'_B = \langle N_B \cup \{S_B\}, \Sigma \cup N, R'_B, S_B\rangle$ для M_B^R с правилами переписывания:
   - $S_B \to aC_B \Leftrightarrow \langle S_B, a, C_B\rangle \in \delta^R$ и $C_B \neq B_B$ либо из B_B есть стрелки в M_B^R;
   - $S_B \to a \Leftrightarrow \langle S_B, a, B_B\rangle \in \delta^R$;
   - $D_B \to EC_B \Leftrightarrow \langle S_B, E, C_B\rangle \in \delta^R$ и $C_B \neq B_B$ либо из B_B есть стрелки в M_B^R;
   - $C_B \to E \Leftrightarrow \langle C_B, E, B_B\rangle \in \delta^R$.

Окончание конструкции

Построение GNF

Теперь по всем G'_i строим окончательный вариант грамматики $G_B = \langle N_B \cup \{S_B\}, \Sigma, R_B, S_B \rangle$ с правилами:

- $S_B \to aC_B,\ S_B \to aC_B \in R'_B$;
- $S_B \to a\ S_B \to a \in R'_B$;
- $D_B \to \alpha C_B \Leftrightarrow D_B \to EC_B \in R'_B\ \&\ S_E \to \alpha$ (по всем таким α и E);
- $D_B \to \alpha \Leftrightarrow D_B \to E \in R'_B\ \&\ S_E \to \alpha$ (по всем таким α и E).

Грамматика $\bigcup\limits_{i \in N} G_i$ со стартовым символом S_S — это искомая GNF для исходной грамматики G.

Пример преобразования грамматики по Блюму–Коху

Привести к GNF грамматику некорректных сумм двоичных чисел (почему некорректных?)

$$S \to S + S \mid D \qquad D \to D0 \mid D1 \mid 1 \mid (S)$$

Пример преобразования грамматики по Блюму–Коху

Привести к GNF грамматику некорректных сумм двоичных чисел (почему некорректных?)

$$\mathrm{S} \to \mathrm{S} + \mathrm{S} \,|\, \mathrm{D} \qquad \mathrm{D} \to \mathrm{D}0 \,|\, \mathrm{D}1 \,|\, 1 \,|\, (\mathrm{S})$$

Сначала избавляемся от цепного правила $S \to \mathrm{D}$. Потом строим порождающую структуру A_{V} сентенциальных форм по левостороннему разбору с финальным состоянием N_{V} и стартовым V_{V}. Каждому нетерминалу V соответствует своя структура.

Для A_{S} :

$$\mathrm{S}_{\mathrm{S}} \xrightarrow{+\mathrm{S}} \mathrm{S}_{\mathrm{S}} \quad \mathrm{S}_{\mathrm{S}} \xrightarrow{0} \mathrm{D}_{\mathrm{S}} \quad \mathrm{S}_{\mathrm{S}} \xrightarrow{1} \mathrm{D}_{\mathrm{S}} \quad \mathrm{S}_{\mathrm{S}} \xrightarrow{1} \mathrm{N}_{\mathrm{S}}$$

$$\mathrm{S}_{\mathrm{S}} \xrightarrow{(\mathrm{S})} \mathrm{N}_{\mathrm{S}} \quad \mathrm{D}_{\mathrm{S}} \xrightarrow{0} \mathrm{D}_{\mathrm{S}} \quad \mathrm{D}_{\mathrm{S}} \xrightarrow{1} \mathrm{D}_{\mathrm{S}} \quad \mathrm{D}_{\mathrm{S}} \xrightarrow{1} \mathrm{N}_{\mathrm{S}} \quad \mathrm{D}_{\mathrm{S}} \xrightarrow{(\mathrm{S})} \mathrm{N}_{\mathrm{S}}$$

Пример преобразования грамматики по Блюму–Коху

Привести к GNF грамматику некорректных сумм двоичных чисел (почему некорректных?)

$$S \to S + S \mid D \qquad D \to D0 \mid D1 \mid 1 \mid (S)$$

Сначала избавляемся от цепного правила $S \to D$. Потом строим порождающую структуру $\mathcal{A}_V$ сентенциальных форм по левостороннему разбору с финальным состоянием N_V и стартовым V_V. Каждому нетерминалу V соответствует своя структура.

Для $\mathcal{A}_S$: $\quad S_S \xrightarrow{+S} S_S \quad S_S \xrightarrow{0} D_S \quad S_S \xrightarrow{1} D_S \quad S_S \xrightarrow{1} N_S$

$S_S \xrightarrow{(S)} N_S \quad D_S \xrightarrow{0} D_S \quad D_S \xrightarrow{1} D_S \quad D_S \xrightarrow{1} N_S \quad D_S \xrightarrow{(S)} N_S$

Для $\mathcal{A}_D$: $\quad D_D \xrightarrow{0} D_D \quad D_D \xrightarrow{1} D_D \quad D_D \xrightarrow{1} N_D \quad D_D \xrightarrow{(S)} N_D$

Пример преобразования грамматики по Блюму–Коху

Превращаем структуры в праволинейные (меняя местами нетерминалы левых и правых частей правил и стартовые состояния с финальными):

$$\text{Для } A_S: \quad S_S \xrightarrow{+S} S_S \quad D_S \xrightarrow{0} S_S \quad D_S \xrightarrow{1} D_S \quad N_S \xrightarrow{1} S_S$$
$$N_S \xrightarrow{(S)} S_S \quad D_S \xrightarrow{0} D_S \quad D_S \xrightarrow{1} D_S \quad N_S \xrightarrow{1} D_S \quad N_S \xrightarrow{(S)} D_S$$

$$\text{Для } A_D: \quad D_D \xrightarrow{0} D_D \quad D_D \xrightarrow{1} D_D \quad N_D \xrightarrow{1} D_D \quad N_D \xrightarrow{(S)} D_D$$

Извлекаем праволинейные (почти) грамматики. В правой части правила может не быть нетерминалов, если там стоял нетерминал V_V.

q-RLG G_S :	$S_S \to +SS_S$	$D_S \to 0S_S$	$D_S \to 1D_S$	$N_S \to 1S_S$
$N_S \to (S)S_S$	$D_S \to 0D_S$	$D_S \to 1D_S$	$N_S \to 1D_S$	$N_S \to (S)D_S$
$S_S \to +S$	$D_S \to 0$	$D_S \to 1$	$N_S \to 1$	$N_S \to (S)$

Извлекаем праволинейные (почти) грамматики. В правой части правила может не быть нетерминалов, если там стоял нетерминал V_V.

q-RLG G_S :	$S_S \to +SS_S$	$D_S \to 0S_S$	$D_S \to 1D_S$	$N_S \to 1S_S$
$N_S \to (S)S_S$	$D_S \to 0D_S$	$D_S \to 1D_S$	$N_S \to 1D_S$	$N_S \to (S)D_S$
$S_S \to +S$	$D_S \to 0$	$D_S \to 1$	$N_S \to 1$	$N_S \to (S)$

q-RLG G_D :	$D_D \to 0D_D$	$D_D \to 1D_D$	$N_D \to 1D_D$	$N_D \to (S)D_D$
$D_D \to 0$	$D_D \to 1$	$N_D \to 1$	$N_D \to (S)$	

Извлекаем праволинейные (почти) грамматики. В правой части правила может не быть нетерминалов, если там стоял нетерминал V_V.

q-RLG $\mathrm{G_S}$: $\quad S_S \to +SS_S \quad \mathrm{D_S} \to 0S_S \quad \mathrm{D_S} \to 1\mathrm{D_S} \quad \mathrm{N_S} \to 1S_S$

$\mathrm{N_S} \to (S)S_S \quad \mathrm{D_S} \to 0\mathrm{D_S} \quad \mathrm{D_S} \to 1\mathrm{D_S} \quad \mathrm{N_S} \to 1\mathrm{D_S} \quad \mathrm{N_S} \to (S)\mathrm{D_S}$

$S_S \to +S \quad \mathrm{D_S} \to 0 \quad \mathrm{D_S} \to 1 \quad \mathrm{N_S} \to 1 \quad \mathrm{N_S} \to (S)$

q-RLG $\mathrm{G_D}$: $\quad \mathrm{D_D} \to 0\mathrm{D_D} \quad \mathrm{D_D} \to 1\mathrm{D_D} \quad \mathrm{N_D} \to 1\mathrm{D_D} \quad \mathrm{N_D} \to (S)\mathrm{D_D}$

$\mathrm{D_D} \to 0 \quad \mathrm{D_D} \to 1 \quad \mathrm{N_D} \to 1 \quad \mathrm{N_D} \to (S)$

Заменяем неразмеченные нетерминальные символы V исходной грамматики на $\mathrm{N_V}$. В данном случае нет правил, в которых неразмеченные нетерминалы стояли бы первыми в правых частях, поэтому достаточно просто заменить их на $\mathrm{N_V}$. Иначе пришлось бы заменять их на все возможные правые части α правил вида $\mathrm{N_V} \to \alpha$. Стартовый символ — $\mathrm{N_S}$. GNF почти построена!

Извлекаем праволинейные (почти) грамматики. В правой части правила может не быть нетерминалов, если там стоял нетерминал V_V.
Заменяем неразмеченные нетерминальные символы V исходной грамматики на N_V. В данном случае нет правил, в которых неразмеченные нетерминалы стояли бы первыми в правых частях, поэтому достаточно просто заменить их на N_V. Иначе пришлось бы заменять их на все возможные правые части α правил вида $\mathrm{N}_V \to \alpha$. Стартовый символ — N_S. GNF почти построена!

q-GNF для G :	$S_S \to +\mathrm{N}_S S_S$	$\mathrm{D}_S \to 0S_S$	$\mathrm{D}_S \to 1\mathrm{D}_S$	$\mathrm{N}_S \to 1S_S$
$\mathrm{N}_S \to (\mathrm{N}_S)S_S$	$\mathrm{D}_S \to 0\mathrm{D}_S$	$\mathrm{D}_S \to 1\mathrm{D}_S$	$\mathrm{N}_S \to 1\mathrm{D}_S$	$\mathrm{N}_S \to (\mathrm{N}_S)\mathrm{D}_S$
$S_S \to +\mathrm{N}_S$	$\mathrm{D}_S \to 0$	$\mathrm{D}_S \to 1$	$\mathrm{N}_S \to 1$	$\mathrm{N}_S \to (\mathrm{N}_S)$

Извлекаем праволинейные (почти) грамматики. В правой части правила может не быть нетерминалов, если там стоял нетерминал V_V.

Заменяем неразмеченные нетерминальные символы V исходной грамматики на N_V. В данном случае нет правил, в которых неразмеченные нетерминалы стояли бы первыми в правых частях, поэтому достаточно просто заменить их на N_V. Иначе пришлось бы заменять их на все возможные правые части α правил вида $\mathrm{N}_V \to \alpha$. Стартовый символ — N_S. GNF почти построена!

q-GNF для G : $S_S \to +\mathrm{N}_S S_S$	$\mathrm{D}_S \to 0S_S$	$\mathrm{D}_S \to 1\mathrm{D}_S$	$\mathrm{N}_S \to 1S_S$	
$\mathrm{N}_S \to (\mathrm{N}_S)S_S$	$\mathrm{D}_S \to 0\mathrm{D}_S$	$\mathrm{D}_S \to 1\mathrm{D}_S$	$\mathrm{N}_S \to 1\mathrm{D}_S$	$\mathrm{N}_S \to (\mathrm{N}_S)\mathrm{D}_S$
$S_S \to +\mathrm{N}_S$	$\mathrm{D}_S \to 0$	$\mathrm{D}_S \to 1$	$\mathrm{N}_S \to 1$	$\mathrm{N}_S \to (\mathrm{N}_S)$

Осталось обернуть в delay-нетерминалы терминальные символы правых частей правил, кроме первого. Здесь это символ).

Связь КС-грамматик и АПГ*

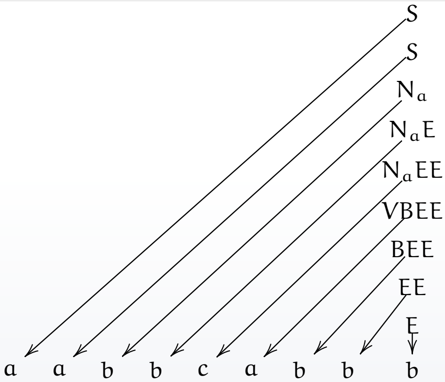
Как и в технике Блюма–Коха, рассмотрим язык сентенциальных форм, но только для грамматик в ГНФ, и без учёта терминальных символов. Получим алфавитную префиксную грамматику (АПГ, см. лекцию 4). При этом каждое применение правила переписывания такой грамматики выбрасывает ровно один терминальный символ слева.

Рассмотрим КС-грамматику, соответствующую ей АПГ и путь вывода слова, получаемый с помощью АПГ.

$$
\begin{array}{lcl}
S & \to & aS \mid bS \mid aN_a \mid bN_b \\
N_a & \to & aN_aE \mid bN_aE \mid cVB \\
N_b & \to & aN_bE \mid bN_bE \mid cVA \\
V & \to & aV \mid bV \mid a \mid b \\
\multicolumn{3}{l}{A \to a \quad B \to b \quad E \to a \mid b}
\end{array}
\qquad
\begin{array}{lcl}
S & \to & S \mid N_a \mid N_b \\
N_a & \to & N_aE \mid VB \\
N_b & \to & N_bE \mid VA \\
V & \to & V \mid \varepsilon \\
\multicolumn{3}{l}{A \to \varepsilon \quad B \to \varepsilon \quad E \to \varepsilon}
\end{array}
$$

Связь КС-грамматик и АПГ*

S	$\to$	$aS \mid bS \mid aN_a \mid bN_b$	S	$\to$	$S \mid N_a \mid N_b$
N_a	$\to$	$aN_aE \mid bN_aE \mid cVB$	N_a	$\to$	$N_aE \mid VB$
N_b	$\to$	$aN_bE \mid bN_bE \mid cVA$	N_b	$\to$	$N_bE \mid VA$
V	$\to$	$aV \mid bV \mid a \mid b$	V	$\to$	$V \mid \varepsilon$
$A \to a$		$B \to b \quad E \to a \mid b$	$A \to \varepsilon$		$B \to \varepsilon \quad E \to \varepsilon$



Рассмотрим $\{a^n b^m \mid n \neq m\} \cup \{a^n b^n \mid n$ — простое число$\}$. Этот язык накачивается любыми леммами Огдена! Но множественный анализ накачек его берёт.