Санкт-Петербургский государственный университет

ОЛЬХОВСКИЙ Илья Сергеевич
Выпускная квалификационная работа

Сложность преобразования LL(k) линейных грамматик и конъюнктивных LL(k) линейных к LL(1) линейным

Образовательная программа бакалавриат «Математика»
Направление и код: 01.03.01 «Математика»
Шифр ОП: СВ.5000.2016

Научный руководитель:
профессор
факультета математики
и компьютерных наук СПбГУ
к.ф.-м.н., Ph.D.
А. С. Охотин

Рецензент: доцент
математико-механического
факультета СПбГУ
к.ф.-м.н.
С. В. Григорьев

Санкт-Петербург
2020 год

Содержание

1 Введение

LL(k)-анализ — один из самых известных методов синтаксического анализа, работающих за линейное время. В этом методе дерево разбора последовательно восстанавливается сверху вниз, по мере чтения строки слева направо. При этом нужное правило вывода синтаксический анализатор выбирает, заглядывая вперёд не более, чем на k символов. Класс LL(k) грамматик, к которому он применим, был изучен в работах Кнута [1971], Льюиса и Стирнса [1968], Розенкранца и Стирнса [1970]. При этом Розенкранц и Стирнс, а также Курки-Суонио [1969] установили, что, для всякого k, грамматики из класса LL($k+1$) порождают больше языков, чем грамматики из класса LL(k), и таким образом появляется иерархия классов LL грамматик.

Важный подкласс *линейных LL(k) грамматик* впервые изучался Ибаррой и др. [1988] и Хольцером и Ланге [1993], получивших ряд результатов о вычислительной сложности языков, задаваемых такими грамматиками. Однако очевидный вопрос о существовании иерархии классов LL(k)-линейных грамматик остался нерассмотренным.

В данной работе даётся отрицательный ответ на этот вопрос: оказывается, что в линейном случае иерархия по k обрушивается, то есть все языки, задаваемые грамматиками из класса LL(k)-линейных для некоторого k, задаются и грамматиками из класса LL(1)-линейных. Доказательство проводится эффективным преобразованием, состоящем из двух этапов и описанном в разделах 3–5: на первом этапе произвольная LL(k)-линейная грамматика преобразуется к некоторому нормальному виду, а на втором этапе — далее к LL(1)-линейной.

Полученное построение приводит к сильному росту числа нетерминальных символов грамматики: их становится больше примерно в $|\Sigma|^{2k}$ раз, где Σ — алфавит языка. Поэтому естественным образом возникает вопрос об эффективности приведённого построения и о нижней оценке количества нетерминальных символов LL(1)-линейной грамматики, задающей тот же язык, что и данная LL(k)-линейная. Нельзя ли разработать другое, улучшенное построение, которое не будет приводить к столь значительному росту?

Оказывается, что нет. В разделе 6 устанавливается, что такой рост количества нетерминальных символов неизбежен, и приводимое построение на самом деле близко к оптимальному: строятся примеры LL(k)-линейных грамматик с n нетерминальными символами, для которых доказывается, что всякая LL(1)-линейная грамматика, задающая тот же язык, должна иметь не менее чем $n \cdot |\Sigma|^{2k-O(\log k)}$ нетерминальных символов.

Третий результат этой работы посвящён более мощному классу формальных грамматик — *конъюнктивным грамматикам.* Конъюнктивные грамматики, введённые Охотиным [2001], обобщают обыкновенные грамматики введением операции конъюнкции в правила. Для них также определены LL(k)-подкласс и линейный подкласс, и, таким образом, можно рассматривать *LL(k)-линейные конъюнктивные грамматики*; их определение приведено в разделе 7. Ранее Охотин [2011] определил простейшие ограничения выразительной мощности этого подкласса, но в целом класс остаётся малоизученным. В частности, вопрос о существовании иерархии по k не рассматривался.

В разделах 8–9 этой работы установлено, что LL(k)-линейные конънктивные грамматики можно перевести в LL(1)-линейные конъюнктивные. Результат доказывается обобщением построения для случая обыкновенных грамматик.

Результаты разделов 3–6 данной работы представлены в двух статьях, опубликованных в сборниках докладов конференций RuFiDiM 2019 и CSR 2020.

2 Основные определения

Определение 1. ***Линейной (формальной) грамматикой*** *называется четвёрка $G = (\Sigma, N, R, S)$, состоящая из следующих компонентов:*

1. Σ — *конечное множество символов, называемое* ***алфавитом***.
2. N — *конечное множество* ***нетерминалов***. *Каждый нетерминал обозначает некоторое свойство, которым строка из* Σ^* *может обладать или не обладать.*
3. R — *конечное множество* ***правил*** *грамматики, каждое из которых описывает возможную структуру строк со свойством* $A \in N$. *Каждое правило имеет вид* $A \to w_1 B w_2$, *где* $B \in N$ — *нетерминал, а* $w_1, w_2 \in \Sigma^*$ — *строки, или вид* $A \to x$, *где* $x \in \Sigma^*$.

   Правило $A \to x$ *означает, что строка* x *обладает свойством* A, *а наличие правила* $A \to w_1 B w_2$ *означает, что если строка* s *обладает свойством* B, *то строка* $w_1 s w_2$ *обладает свойством* A.
4. $S \in N$ — *выделенный* ***начальный*** *нетерминал.*

Определение 2. ***Деревом разбора*** *линейной грамматики* $G = (\Sigma, N, R, S)$ *называется упорядоченное корневое дерево, листья которого помечены символами алфавита* Σ, *а внутренние вершины — нетерминальными символами из* N. *Если внутренняя вершина помеченна нетерминалом* $A \in N$, *а её потомки — символами* $X_1, \ldots, X_l \in \Sigma \cup N$, *то в грамматике должно присутствовать правило* $A \to X_1 \ldots X_l$. *Так как среди потомков* $X_1 \ldots X_l$ *не более одного является нетерминалом, то дерево разбора представляет из себя путь из нетерминальных вершин, от которых влево и вправо отходят листья, как показано на рисунке 1(слева).*

Пусть w — *строка из символов всех листьев, а* $A \in N$ — *нетерминал, которым помечен корень. Тогда соответствующее дерево разбора называют* ***деревом разбора*** w ***из*** A.

Язык, задаваемый нетерминалом A, *определяется как множество всех строк* w, *для которых существует дерево разбора* w *из* A. *Мы будем обозначать его* $L_G(A)$ *или* $L(A)$, *если ясно, о какой грамматике идёт речь.*

Язык, задаваемый грамматикой, *определяется как язык, задаваемый её начальным нетерминалом* S. *Мы будем обозначать его* $L(G)$.

Замечание 1. *Если нетерминал, которым помечен корень дерева разбора, не указан явно, то подразумевается, что им является начальный нетерминал* S.

Замечание 2. *Пусть в дереве разбора* w *из* A *к корню применяется правило* $A \to \alpha$. *Тогда будем говорить, что в этом дереве разбора* A *задаёт* w *по правилу* $A \to \alpha$.

Замечание 3. *Мы будем допускать некую вольность речи и писать "поддерево, помеченное* A*"или просто "поддерево* A *имея в виду "поддерево, корень которого помечен нетерминалом* A*".*

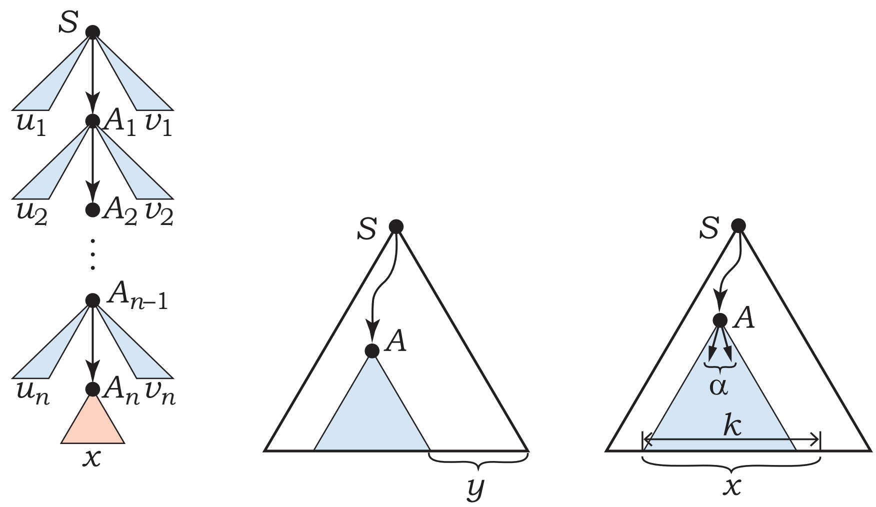

Рис. 1: Слева направо: дерево разбора линейной грамматики, иллюстрация к определению $\mathrm{Follow}(A)$, иллюстрация к определению LL-таблицы

Определение 3. *Пусть $G = (\Sigma, N, R, S)$ — формальная грамматика.*

Пусть зафиксировано некоторое дерево разбора G и поддерево в нём.

Будем говорить, что строка $y \in \Sigma^*$ ***следует*** *за этим поддеревом, если все листья справа от поддерева образуют строку y.*

Будем говорить, что y ***следует за нетерминалом*** *$A \in N$, если существует поддерево некоторого дерева разбора, помеченное символом A, за которым следует y, как на рисунке 1(в середине).*

Обозначим:

$$\mathrm{Follow}(A) = \{\, v \mid v \text{ следует за } A \,\}$$

Синтаксическим анализом строки называется построение дерева разбора этой строки. Класс грамматик $\mathrm{LL}(k)$, рассматриваемый в этой работе, допускает синтаксический анализ за линейное время с помощью специального алгоритма.

Сначала будет описан этот алгоритм, а потом уже точно определён класс $\mathrm{LL}(k)$.

Определение 4. ***Синтаксическим анализатором*** *для линейной $LL(k)$ грамматики $G = (\Sigma, N, R, S)$ называется алгоритм, который по данной строке w строит её дерево разбора.*

Строка w при этом читается слева направо, а дерево разбора строится сверху вниз.

На каждом шаге ***состоянием*** *(или* ***конфигурацией****) алгоритма является пара (uAv, x), где $x \in \Sigma^*$ — ещё не разобранный суффикс стро-*

ки w, а uAv, где $A \in N$, $u, v \in \Sigma^*$, — содержимое ***стека*** *синтаксического анализатора.*

Инвариантом алгоритма является то, что строка w задаётся грамматикой тогда и только тогда, когда строка x представима в виде конкатенации uyv, где $y \in L(A)$.

В начале работы алгоритм находится в состоянии (S, w).

На каждом шаге алгоритм видит символ на вершине стека и первые k символов строки x.

Если на вершине стека находится символ алфавита $a \in \Sigma$, и конфигурация анализатора имеет вид $(auAv, x)$, синтаксический анализатор проверяет, что строка x начинается с a, и если так и есть, то переходит в конфигурацию $(auAv, x')$, где $ax' = x$, а если x пусто или начинается с другого символа, то выдаёт ошибку.

Если же на вершине стека находится нетерминал $A \in N$, и конфигурация анализатора имеет вид (Av, x), синтаксический анализатор неким способом, связанным со спецификой класса LL(k), определяет правило $A \to sBt$, которое следует применить к нетерминалу A, и переходит в состояние $(sBtv, x)$.

Принадлежность грамматики G к классу LL(k) позволяет описанному выше синтаксическому анализатору определять правило, которое следует применить к нетерминалу на вершине стека, путём обращения к специальной таблице:

Определение 5. *LL(k)-таблицей для грамматики $G = (\Sigma, N, R, S)$ называется частично определённая функция $T : N \times \Sigma^{\leqslant k} \to R$, удовлетворяющая следующему условию. $T(A, x) = (A \to \alpha)$, где $A \in N, x \in \Sigma^{\leqslant k}$, тогда и только тогда, когда существует поддерево некоторого дерева разбора, помеченное нетерминалом A, такое что первые k листьев всего дерева, начиная с самого левого листа поддерева, образуют строку x, и к корню поддерева применяется правило $A \to \alpha$, как показано на рисунке 1(справа).*

Если для грамматики G существует LL(k)-таблица, и каждый нетерминал G встречается в некотором дереве разбора, то говорят, что G принадлежит классу LL(k).

Таким образом, если G принадлежит классу LL(k), то для любого поддерева дерева разбора G правило, применяемое к корню этого поддерева, однозначно определяется строкой, составленной из символов первых k листьев, начиная с самого левого листа поддерева.

Требование, того чтобы каждый нетерминал грамматики встречался в некотором дереве разбора нужно чтобы исключить "вырожденные грамматики". Если грамматика задаёт хотя бы одну строку, его легко удовлетворив, просто убрав из грамматики все нетерминалы, которые не встречаются ни в одном дереве разбора.

Следовательно, когда синтаксический анализатор находится в состоянии (Av, x), он понимает, что к A следует применить правило $T(A, x')$, где x' — первые k символов строки x.

Если $T(A, x')$ не определено, то анализатор выдаёт ошибку.

Определение 6. *Пусть $w \in \Sigma^*$. Обозначим за $\mathrm{First}_k(w)$ префикс w длины k. Если $|w| < k$, то $\mathrm{First}_k(w) = w$.*

Это обозначение обобщается на языки. Пусть $L \subseteq \Sigma^*$. Тогда

$$\mathrm{First}_k(L) = \{\, \mathrm{First}_k(w) \mid w \in L \,\}$$

Далее в доказательствах будет использоваться следующее известное утверждение.

Факт 1. *Следующие два утверждения эквивалентны.*

1. *Грамматика $G = (\Sigma, N, R, S)$ принадлежит классу LL(k).*

2. *Для любого нетерминала $A \in N$ и для любых двух различных правил $A \to \alpha_1$ и $A \to \alpha_2$ множества $\mathrm{First}_k(L_G(\alpha_1)\mathrm{Follow}(A))$ и $\mathrm{First}_k(L_G(\alpha_2)\mathrm{Follow}(A))$ не пересекаются.*

3 План преобразования линейной LL(k) грамматики к виду LL(1)

Первый результат данной работы состоит в том, что по произвольной линейной LL(k) грамматике G можно построить линейную LL(1) грамматику G', задающую тот же язык.

Естественно пытаться определить G' так, чтобы вывод каждой строки w в G' каким-нибудь образом повторял вывод той же строки w в G, и, соответственно, каждое вычисление синтаксического анализатора G' также каким-нибудь образом повторяло аналогичное вычисление синтаксического анализатора G.

Синтаксический анализатор для исходной грамматики G определял правило, которое нужно применить на очередном шаге, исходя из первых k символов ещё не прочитанной части входной строки.

Чтобы повторить вычисление анализатора G, анализатору G' тоже необходимо каким-то образом узнать следующие k символов входной строки.

Основная идея преобразования заключается в использовании *буфера* для не более, чем $k - 1$ символов: будучи не в силах сразу определить правило, которое следует применить для нетерминала на вершине стека, анализатор продолжает читать символы входной строки, занося их в буфер, пока их не наберётся достаточно, чтобы определить то самое правило.

Реализуется это с помощью введения новых нетерминалов ${}_uA$, где $A \in N$ — нетерминал исходной грамматики, а строка $u \in \Sigma^{\leqslant k-1}$ выполняет роль буфера. Каждый нетерминал ${}_uA$ задаёт язык $L_{G'}({}_uA) = \{w \mid uw \in L_G(A)\}$.

Каждому правилу $A \to sBt$ грамматики G, такому, что одна из строк u, s является префиксом другой, в грамматике G' соответствует цепочка из k правил:

$$\begin{aligned} A &\to u_1\, {}_{u_1}A \\ {}_{u_1}A &\to u_2\, {}_{u_1u_2}A \\ &\vdots \\ {}_{u_1\dots u_{k-2}}A &\to u_{k-1}\, {}_uA \\ {}_uA &\to \alpha \end{aligned}$$

Первые $k-1$ правил наращивают буфер, и когда буфер достигает размера $k-1$, что, вместе с одним символов, доступным анализатору G' напрямую, даёт необходимые ему k символов, применяется нужное правило α. Последнее получается откусыванием уже прочтённой строки u из начала правила $A \to sBt$: общие префиксы u и s сокращаются, и оставшиеся символы u, если они есть, уходят в буфер нетерминала B. Таким образом, если $u = sv$, то $\alpha = {}_vBt$, а если $s = us'$, то $\alpha = s'\,{}_\varepsilon Bt$. Это проиллюстрировано на рисунках 5 и 6.

Однако у такого плана есть существенный изъян. Если в исходной грамматике есть *короткое* правило $A \to s$, где $s \in \Sigma^{<k-1}$, то синтаксическому анализатору G', чтобы определить это правило, может понадобиться занести в буфер не только всю строку s, но и некоторые символы, идущие после неё, не порождённые нетерминалом A.

В таком случае в момент, когда анализатор поймёт, что надо применить правило $A \to s$, на вершине его стека будет находится нетерминал ${}_{sx}A$, где $|x| > 0$, и удалить буфер sx из правой части правила $A \to s$ будет, очевидно, невозможно.

Пример 1. *Рассмотрим следующую линейную LL(3) грамматику.*

$$\begin{aligned} &S \to aabSaa \\ &S \to a && \text{(}\textit{короткое правило}\text{)} \end{aligned}$$

Чтобы понять, какое из этих правил следует применить, гипотетический LL(1) анализатор заносит в буфер до двух первых символов с помощью следующих правил.

$$\begin{aligned} &{}_\varepsilon S \to a\, {}_aS \\ &{}_aS \to a\, {}_{aa}S \end{aligned}$$

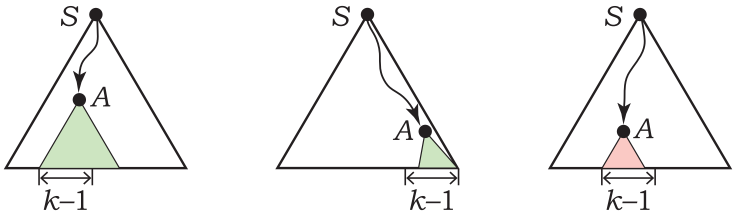

Рис. 2: Третье правило короткое, а первые два — нет.

Когда в буфере находится строка aa, и анализатор видит, что следующий символ — b, он понимает, что следует применить правило $S \to aabSaa$*, и применяет его, предварительно удалив из начала уже прочтённые символы aa.*

$$_{aa}S \to b\,{_\varepsilon}Saa$$

Однако если анализатор видит, что следующий символ — a, он понимает, что следует применить правило $S \to a$*, но совершенно не ясно, чему поставить в соответствие уже прочтённые символы aa.*

Корень проблемы кроется в наличии *короткого правила*, порождающего подстроку длины менее, чем $k-1$, где-то в середине строки.

Далее будет показано, что в случае отсутствия в грамматике коротких правил намеченная конструкция будет работать.

Соответственно, первым шагом преобразования произвольной линейной LL(k) грамматики к виду LL(1) будет устранение коротких правил.

4 Устранение «коротких» правил

Определение 7. ***Коротким правилом*** *будем называть правило вида* $A \to w$*, где* $w \in \Sigma^*$, $|w| < k-1$ *и* $\mathrm{Follow}(A) \neq \{\varepsilon\}$*, как показано на рисунке 2 (справа).*

Лемма 1. *Для каждой линейной LL(k) грамматики* $G = (\Sigma, N, R, S)$ *существует линейная LL(k) грамматика* G' *без коротких правил, задающая тот же язык. Количество нетерминалов* G' *не превосходит* $|\Sigma^{\leqslant k-1}| \cdot |N|$.

Доказательство. Опишем структуру новой грамматики $G' = (\Sigma, N', R', S_\varepsilon)$. Нетерминалы G' имеют вид A_u, где $A \in N$ и $u \in \Sigma^{\leqslant k-1}$.

Грамматика G' будет определена так, что каждый нетерминал A_u будет задавать все строки, задаваемые A в G, с приписанным в конце суффиксом u: $L_{G'}(A_u) = \{wu \mid w \in L_G(A)\}$.

Для каждого нетерминала A_u и для каждого правила $A \to w_1 B w_2 \in R$, в новой грамматике будет правило $A_u \to \alpha$, определённое ниже.

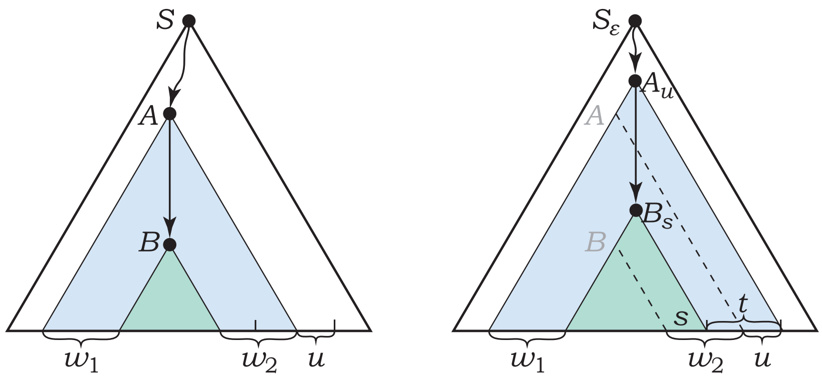

Рис. 3: Получение правила $A_u \to w_1 B_s t$ в G' из правила $A \to w_1 B w_2$ в G

Пусть s обозначает первые $k-1$ символов строки $w_2 u$, а t — оставшийся суффикс $w_2 u$, так что $s = \text{First}_{k-1}(w_2 u)$ и $st = w_2 u$.

Правило $A_u \to \alpha$, получается из правила $A \to w_1 B w_2 \in R$ следующим образом: строка u дописывается в конец правила, так что получается правило $A_u \to w_1 B w_2 u$, а затем первые $k-1$ символов строки $w_2 u$ прикрепляются к нетерминалу B в качестве нижнего индекса.

$$A_u \to w_1 B_s t$$

Это проиллюстрировано на рисунке 3.

Для правил вида $A \to x$, где $x \in \Sigma^*$, соответствующие правила для нетерминалов A_u новой грамматики получаются просто приписыванием строки u в конец.

$$A_u \to xu$$

Заметим, что по каждому правилу $A_u \to \alpha$ новой грамматики всегда можно однозначно восстановить правило $A \to \gamma$ исходной грамматики, из которого оно получено: если $\alpha \in \Sigma^*$, то γ получается откусыванием у α суффикса u, а если $\alpha = w_1 B_s t$, то γ получается откусыванием суффикса u у строки $w_1 B s t$.

Доказательство корректности описанного построения естественно разбивается на несколько отдельных утверждений: а именно, что G' является линейной LL(k) грамматикой, задаёт тот же язык, что и G, и не содержит коротких правил.

Начнём с доказательства уже упомянутого факта: $L_{G'}(A_u) = \{xu \mid x \in L_G(A)\}$.

Как обычно, чтобы установить равенство множеств, проверим два включения.

Утверждение 1. *Если строка w задаётся нетерминалом A_u в новой грамматике, то $w = xu$, где x задаётся нетерминалом A в исходной грамматике.*

Доказательство. Индукция по высоте дерева разбора строки w из нетерминала A_u.

Базовый случай. Пусть A_u задаёт w по правилу $A_u \to w$.

По построению правило $A_u \to w$ получено из некоторого правила $A \to x$ исходной грамматики и имеет вид $A_u \to xu$.

Таким образом $x \in L_G(A)$.

Индукционный переход. Пусть A_u задаёт w по правилу $A_u \to w_1 B_s t$, полученному из правила $A \to w_1 B w_2$ исходной грамматики, где $w_2 u = st$.

Тогда $w = w_1 y t$, где $y \in L_{G'}(B_s)$, и высота дерева разбора y из нетерминала B_s меньше, чем у дерева разбора w.

По индукционному предположению, $y = zs$, где $z \in L_G(B)$.

Таким образом получаем $w = w_1 z s t = w_1 z w_2 u$, и так как строка $w_1 z w_2$ задаётся нетерминалом A с помощью правила $A \to w_1 B w_2$, значит $w \in L_G(A)u$.

□

Утверждение 2. *Если строка x задаётся нетерминалом A исходной грамматики, то в новой грамматике A_u задаёт xu.*

Доказательство. Индукция по высоте дерева разбора строки x из A.

Базовый случай. Пусть A задаёт x по правилу $A \to x$.

По построению, в грамматике G' есть правило $A_u \to xu$, значит $xu \in L_{G'}(A_u)$.

Индукционный переход. Пусть A задаёт x по правилу $A \to w_1 B w_2$.
Тогда $x = w_1 y w_2$, где $y \in L_G(B)$.

Положим $s = \mathrm{First}_{k-1}(w_2 u)$.

Высота дерева разбора y из B меньше, чем у дерева разбора x, значит по предположению индукции $ys \in L_{G'}(B_s)$.

Грамматика G' содержит правило $A_u \to w_1 B_s t$, полученное из правила $A \to w_1 B w_2$ исходной грамматики, поэтому $w_1 y s t \in L_{G'}(A_u)$.

Так как $w_1 y s t = w_1 y w_2 u = xu$, то строка xu лежит в языке $L_{G'}(A_u)$.

□

Итак, из последних двух утверждений следует, что для каждого нетерминала $A_u \in N'$ выполняется $L_{G'}(A_u) = L_G(A)u$. В частности $L(G') = L_{G'}(S_\varepsilon) = L_G(S) = L(G)$, то есть G' задаёт тот же язык, что и G.

Аналогичное равенство имеет место и для соответствующих правил.

Утверждение 3. *Для любого правила $A_u \to \alpha$, $\alpha \in (\Sigma \cup N')^*$ новой грамматики, полученного из правила $A \to \gamma$ исходной грамматики, выполняется $L_{G'}(\alpha) = L_G(\gamma)u$.*

Доказательство. Пусть правило G' имеет вид $A_u \to w_1 B_s t$ и получено из правила $A \to w_1 B w_2$ исходной грамматики G.

По построению, $st = w_2 u$, и по предыдущим двум утверждениям $L_{G'}(B_s) = L_G(B)s$.

Таким образом, $L_{G'}(w_1 B_s t) = w_1 L_G(B) st = L_G(w_1 B w_2)u$.

Если же правило G' имеет вид $A_u \to xu$, то оно получено из правила $A \to x$, и утверждение тривиально.

□

Теперь докажем, что в G' нет коротких правил.

Уже известно, что $L_{G'}(A_u) = L_G(A)u$, поэтому, если $|u| = k-1$, то A_u не задаёт строк длины менее $k-1$, и, следовательно, для таких нетерминалов коротких правил нет.

Чтобы показать, что нет коротких правил и для нетерминалов $A_u \in N'$ с $|u| < k-1$, потребуется разобраться, как устроены множества $\mathrm{Follow}(A_u)$.

Для начала докажем ещё пару несложных вспомогательных утверждений.

Утверждение 4. *Для любого правила $A_u \to w_1 B_s t$ в G' верно*

- $|s| \geqslant |u|$
- *Если $|t| > 0$, то $|s| = k-1$.*

Доказательство. По построению правило $A_u \to w_1 B_s t$ было получено из некоторого правила $A \to w_1 B w_2$, причём $s = \mathrm{First}_{k-1}(w_2 u)$ и $st = w_2 u$.

Поскольку $|u| \leqslant k-1$, то $|u| = |\mathrm{First}_{k-1}(u)| \leqslant |\mathrm{First}_{k-1}(w_2 u)| = |s|$, и тем самым первая часть доказана.

Если $|t| > 0$, то, $|\mathrm{First}_{k-1}(w_2 u)| < |w_2 u|$, и значит $|s| = |\mathrm{First}_{k-1}(w_2 u)| = k-1$.

□

Утверждение 5. *Пусть $A \in N$, $v \in \mathrm{Follow}(A)$, и в грамматике есть правило $A \to w_1 B w_2$. Тогда $w_2 v \in \mathrm{Follow}(B)$.*

Доказательство. По определению $\mathrm{Follow}(A)$, существует некоторое дерево разбора с поддеревом A, за которым следует строка v. Обозначим это поддерево как T_A.

Пусть x — любая строка из $L_G(B)$. Тогда A задаёт $w_1 x w_2$ по правилу $A \to w_1 B w_2$. Если вставить на место T_A соответствующее дерево вывода $w_1 x w_2$ из A, получится дерево разбора с поддеревом B, за которым следует строка $w_2 v$. Тогда по определению $w_2 v \in \mathrm{Follow}(B)$.

□

Следующее утверждение проливает свет на устройство множеств $\mathrm{Follow}(A_u)$.

Утверждение 6. *Для каждого нетерминала* $B_s \in N'$, *если* $y \in \mathrm{Follow}(B_s)$, *то* $sy \in \mathrm{Follow}(B)$. *Кроме того, если* $|s| < k - 1$, *то* $y = \varepsilon$.

Доказательство. Пусть $y \in \mathrm{Follow}(B_s)$. По определению существует поддерево B_s дерева разбора, за которым следует строка y.

Применим индукцию по глубине этого поддерева в дереве разбора.

Базовый случай. Пусть B_s является корнем всего дерева разбора, то есть $B_s = S_\varepsilon$. Так как правее всего дерева разбора листьев нет, то $y = \varepsilon$ и значит $sy = \varepsilon \in \mathrm{Follow}(S)$.

Индукционный переход. Пусть A_u — родитель B_s в дереве разбора. К A_u применяется некоторое правило $A_u \to w_1 B_s t$, полученное из правила $A \to w_1 B w_2$ исходной грамматики, где $w_2 u = st$.

Обозначим за y' строку, которая следует за поддеревом A_u, так что $y = ty'$ и $y' \in \mathrm{Follow}(A_u)$ (см. рисунок 3).

Так как глубина B_s строго больше глубины A_u в дереве разбора, то к A_u применимо индукционное предположение, и тем самым $uy' \in \mathrm{Follow}(A)$.

По утверждению 5 $w_2 u y' = s t y' = sy \in \mathrm{Follow}(B)$, и первая часть утверждения 6 доказана.

Пусть теперь $|s| < k - 1$. По утверждению 4 $|s| \geqslant |u|$, причём если $|t| > 0$, то $|s| = k - 1$.

Следовательно $|u| < k - 1$ и $t = \varepsilon$. К A_u применимо индукционное предположение, и значит из $|u| < k - 1$ следует $y' = \varepsilon$. Таким образом $y = ty' = \varepsilon$, и второе утверждение доказано.

□

По предыдущему утверждению для нетерминалов A_u с $|u| < k-1$ выполняется $\mathrm{Follow}(A_u) = \{\varepsilon\}$, и тем самым в грамматике G' отсутствуют короткие правила.

Остаётся доказать, что G' является линейной LL(k) грамматикой. Линейность G' видна из построения.

Утверждение 7. *Если G принадлежит классу $LL(k)$, то G' — тоже.*

Доказательство. Пусть A_u — нетерминал G' со следующими правилами.

$$\begin{aligned} A_u &\to \alpha_1 \\ A_u &\to \alpha_2 \\ &\vdots \\ A_u &\to \alpha_n \end{aligned}$$

По определению, грамматика G' обладает свойством LL(k) тогда и только тогда, когда для каждого нетерминала A_u, множества $\mathrm{First}_k(L(\alpha_j)\mathrm{Follow}(A_u))$ попарно не пересекаются.

По построению, каждому правилу $A_u \to \alpha_j$ однозначно соответствует правило $A \to \gamma_j$ исходной грамматики G. Так G принадлежит классу $LL(k)$, то множества $\mathrm{First}_k(L(\gamma_j)\mathrm{Follow}(A))$ попарно не пересекаются.

Таким образом, чтобы доказать, что G' тоже принадлежит классу $LL(k)$, достаточно показать, что $\mathrm{First}_k(L(\alpha_j)\mathrm{Follow}(A_u)) \subseteq \mathrm{First}_k(L(\gamma_j)\mathrm{Follow}(A))$ для каждого правила α_j.

По утверждению 3 $L_{G'}(\alpha_j) = L_G(\gamma_j)u$ для каждого правила $A_u \to \alpha_j$, и по утверждению 6 $u\mathrm{Follow}(A_u) \subseteq \mathrm{Follow}(A)$.

Следовательно,

$$\mathrm{First}_k(L(\alpha_j)\mathrm{Follow}(A_u)) = \mathrm{First}_k(L(\gamma_j)u\mathrm{Follow}(A_u)) \subseteq \mathrm{First}_k(L(\gamma_j)\mathrm{Follow}(A))$$

□

Таким образом грамматика G' принадлежит классу LL(k), что завершает доказательство корректности построения.

□

Пример 2. *Применим описанное построение к грамматике из примера 1.*

$$\begin{aligned} S &\to aabSaa \\ S &\to a && (\textit{короткое правило}) \end{aligned}$$

Получится следующая грамматика без коротких правил.

$$\begin{aligned} S_\varepsilon &\to aabS_a\,a \\ S_\varepsilon &\to a \\ S_a &\to aabS_a\,aa \\ S_a &\to aa \end{aligned}$$

Заметим, что $\operatorname{Follow}(S_\varepsilon) = \{\varepsilon\}$, *и поэтому правило* $S_\varepsilon \to a$ *не короткое.*

5 Приведение к виду LL(1)

После того, как из грамматики удалены все короткие правила, её можно преобразовать к виду LL(1) с помощью конструкции, кратко намеченной в начале.

Однако, перед этим удобно заранее избавиться от правил вида $A \to B$, чтобы избежать некоторых технических проблем.

Лемма 2. *Для каждой линейной LL(k) грамматики* $G = (\Sigma, N, R, S)$ *без коротких правил существует линейная LL(k) грамматика* $G' = (\Sigma, N, R', S)$ *без коротких правил, задающая тот же язык и не содержащая правил вида* $A \to B$*. Количество нетерминалов* G' *такое же, как у грамматики* G*.*

Доказательство. Правила вида $A \to B$ естественным образом задают ориентированный граф на множестве N: в графе есть ребро из A в B тогда и только тогда, когда в G есть правило $A \to B$.

Обозначим $c(A)$ за множество всех вершин, достижимых из нетерминала A.

Правила грамматики G' определяются следующим образом: Пусть R_A — множество правил грамматики G для нетерминала A, кроме правил вида $A \to B$.

Положим $R'_A = \bigcup_{B \in (A)} \{A \to \alpha \mid (B \to \alpha) \in R_B\}$. Тогда $R' = \bigcup_{A \in N} R'_A$.

Чтобы доказать лемму, нужно проверить четыре утверждения.

1. $L_{G'}(A) = L_G(A)$.
2. G' не содержит коротких правил.
3. G' — линейная LL(k) грамматика.
4. G' не содержит правил вида $A \to B$.

Приведённое построение классическое, и хоть отсутствие коротких правил и сохранение свойства LL(k) и требуют некоторых пояснений, читатель, знакомый с ним, вряд ли найдёт здесь для себя что-то новое.

Покажем, что деревья разбора грамматик G и G' находятся в естественном взаимно однозначном соответствии.

Любое дерево разбора G представляет из себя путь по вершинам, помеченным нетерминалами, от которых влево и вправо отходят листья, как показано на рисунке 1(слева). Пусть $A_1 \to A_2 \to A_3 \to \ldots \to A_n$ — этот путь, и к A_n применяется правило $A_n \to x$, где $x \in \Sigma^*$.

Тогда соответствущая последовательность правил выглядит так:

$$
\begin{aligned}
A_1 = A_{n_0} \to A_{n_0+1}, \quad A_{n_0+1} \to A_{n_0+2}, \quad \cdots, \quad A_{n_1-1} \to u_1 A_{n_1} v_1,\\
A_{n_1} \to A_{n_1+1}, \quad A_{n_1+1} \to A_{n_1+2}, \quad \cdots, \quad A_{n_2-1} \to u_2 A_{n_2} v_2,\\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad\\
A_{n_{k-1}} \to A_{n_{k-1}+1}, \quad A_{n_{k-1}+1} \to A_{n_{k-1}+2}, \quad \cdots, \quad A_{n_k-1} = A_n \to x
\end{aligned}
$$

В грамматике G' каждая строчка сжимается в одно правило, чему соответствует последовательность правил

$$
\begin{aligned}
A_0 = A_{n_0} &\to u_1 A_{n_1} v_1,\\
A_{n_1} &\to u_2 A_{n_2} v_2,\\
&\vdots\\
A_{n_{k-1}} &\to x
\end{aligned}
$$

Аналогично, по любому дереву разбора G' можно восстановить дерево разбора G. Пусть $B_1 \to B_2 \to \ldots \to B_m$ — путь из нетерминалов в дереве разбора G'. Соответствующий путь в дереве разбора G получается вставкой некоторых правил вида $A \to B$: пусть поддерево B_j задаёт строку x_j, и за этим поддеревом следует строка y_j. Тогда, чтобы получить путь в дереве разбора G, надо на место каждого правила $B_j \to w_1 B_{j+1} w_2$ вставить цепочку правил $B_j = B_{j_1} \to B_{j_2}, \quad \cdots, \quad B_{j_{k-1}} \to B_{j_k} = B_{j+1}$, где

$$
\begin{aligned}
&T(B_{j_1}, \mathrm{First}_k(x_j y_j)) = B_{j_1} \to B_{j_2}\\
&\vdots\\
&T(B_{j_{k-2}}, \mathrm{First}_k(x_j y_j)) = B_{j_{k-2}} \to B_{j_{k-1}}\\
&T(B_{j_{k-1}}, \mathrm{First}_k(x_j y_j)) = B_{j_{k-1}} \to w_1 B_{j_k} w_2
\end{aligned}
$$

Из описанного соответствия деревьев разбора сразу следует, что $L_G(A) = L_{G'}(A)$ для каждого нетерминала $A \in N$. Также несложно заметить, что если строка y следует за нетерминалом A в грамматике G', то в соответствующем дереве разбора y следует за A в грамматике G, и таким образом $\mathrm{Follow}(A)_{G'} \subseteq \mathrm{Follow}(A)_G$.

Значит, если в грамматике G' нетерминал A задаёт строку длины менее $k-1$, и $\mathrm{Follow}(A)_{G'} \neq \{\varepsilon\}$, то же самое происходит и в грамматике G. Поэтому из отсутствия в G коротких правил следует отсутсвие коротких правил в G'.

Линейность G' и отсутствие в неё правил вида $A \to B$ видны из построения, а $\mathrm{LL}(k)$ таблица для G' получается из $\mathrm{LL}(k)$ таблицы G следующим образом: пусть $T_G(A_1, u) = A_1 \to A_2$, $T_G(A_2, u) = A_2 \to$

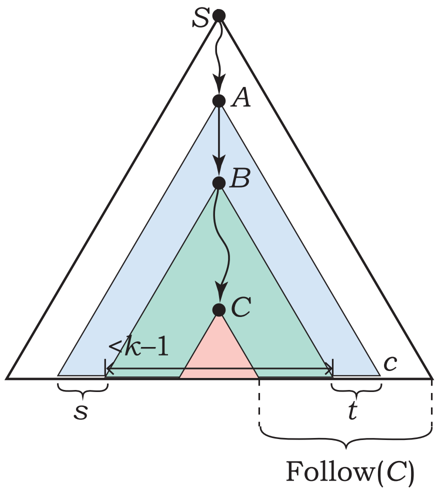

Рис. 4: Иллюстрация к доказательству утверждения 8.

$A_3, \dots, T_G(A_{k-1}, u) = A_{k-1} \to w_1 A_k w_2$. Тогда $T_{G'}(A_1, u) = w_1 A_k w_2$. Если $T_G(A_1, u)$ не определено, то не определено и $T_{G'}(A_1, u)$.

$\square$

После того, как правила вида $A \to B$ устранены, у грамматики появляется хорошее свойство, которое нам потом понадобится.

Утверждение 8. *Пусть $G = (\Sigma, N, R, S)$ — линейная LL(k) грамматика без коротких правил и правил вида $A \to B$. Тогда для любого нетерминала $A \in N$ если $x \in L(A)$ и $|x| = k - 1$, то либо* $\mathrm{Follow}(A) = \{\varepsilon\}$, *либо в R есть правило $A \to x$.*

Доказательство. Пусть $x \in L(A)$, $\mathrm{Follow}(A) \neq \{\varepsilon\}$ и $A \to \alpha$ — правило, по которому A задаёт x. Пусть $c \in \Sigma$ — любой символ из $\mathrm{First}_1(\mathrm{Follow}(A))$.

Тогда $T(A, xc) = A \to \alpha$. Если $\alpha \in \Sigma^*$, то $\alpha = x$ и всё доказано.

Предположим, $\alpha = sBt$ для некоторых $s, t \in \Sigma^*$. Тогда $x = syt$, где $y \in L(B)$. Так как в G нет правил вида $A \to B$, то $|s| + |t| > 0$, и значит $|y| < |x| = k - 1$.

Рассмотрим дерево разбора y из B. Последнее правило в нём имеет вид $C \to z$, где $|z| \leqslant |y| < k - 1$.

Так как $\mathrm{Follow}(A) \neq \varepsilon$, то и $\mathrm{Follow}(B) \neq \varepsilon$, и $\mathrm{Follow}(C) \neq \varepsilon$ (см. рисунок 4).

Значит правило $C \to y'$ короткое, что противоречит отсутствию в G коротких правил.

$\square$

Итак, наконец, пришла пора описать основное построение.

Лемма 3. *Для каждой линейной LL(k) грамматики* $G = (\Sigma, N, R, S)$ *без коротких правил и правил вида* $A \to B$ *существует линейная LL(1) грамматика* G', *задающая тот же язык. Количество нетерминалов* G' *не превосходит* $|\Sigma^{\leqslant k-1}| \cdot |N|$.

Доказательство. Нетерминалы новой грамматики $G' = (\Sigma, N', R', {}_{\varepsilon}S)$, имеют вид ${}_uA$, где $A \in N$ и $u \in \Sigma^{\leqslant k-1}$.

Левый нижний индекс u нетерминала ${}_uA$ выполняет роль буфера, в котором хранится до $k-1$ последних символов, прочтённых синтаксическим анализатором.

Начальным символом G' является ${}_{\varepsilon}S$, что соответствует S с пустым буфером.

Грамматика G' будет определена таким образом, что $L_{G'}({}_uA) = \{w \mid uw \in L_G(A)\}$.

До тех пор, пока буфер не заполнен, анализатор читает символы входной строки и заносит их в буфер. Когда в буфере накопится $k-1$ символов, на вершине стека анализатора будет некий нетерминал ${}_uA$, где $u \in \Sigma^{k-1}$. Строка u и следующий символ a входной строки, доступный анализатору напрямую, вместе образуют k символов, необходимые, чтобы определить нужное правило для нетерминала A. Это правило находится с помощью обращения к элементу $T(A, ua)$ в LL(k) таблице грамматики G.

Итак, $N' = \{{}_uA \mid A \in N\, u \in \Sigma^{\leqslant k-1}\}$. Правила грамматики G' представляются объединением трёх множеств R_1,R_2 и R_3.

Множество правил R_1 реализуют заполнение буфера. Для каждого нетерминала ${}_uA$ с $|u| < k-1$, и для каждого символа $a \in \Sigma$, в G' есть правило, заносящее этот символ в буфер.

$${}_uA \to a\, {}_{ua}A$$

Правила R_2 используются, когда буфер уже заполнен, и синтаксический анализатор G' может понять, какое правило исходной грамматики следует применить. Для каждого ${}_uA \in N'$ и $a \in \Sigma$, где $|u| = k-1$ и значение $T(A, ua)$ определено, в G' есть правило, получающееся *откусыванием* строки u из правила $(A \to \alpha) = T(A, ua)$. Если $\alpha \in \Sigma^*$, тогда $\alpha = us$, где $s \in \Sigma^*$, и соответствующее правило в G' имеет вид

$${}_uA \to s$$

Если α содержит нетерминал в правой части, то есть $\alpha = sBt$, где $s, t \in \Sigma^*$, $B \in N$, то одна из строк u и s является префиксом другой, и имеем два случая:

$$\begin{aligned} {}_uA &\to s'\,{}_{\varepsilon}Bt, && \text{если } s = us',\ s' \in \Sigma^* \\ {}_uA &\to {}_vBt, && \text{если } u = sv,\ v \in \Sigma^+ \end{aligned}$$

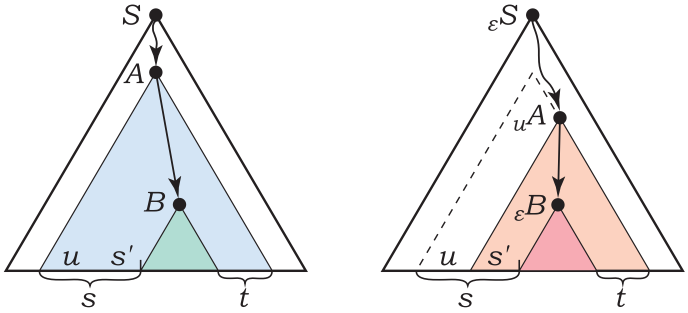

Рис. 5: Откусывание строки u из начала правила $A \to sBt$ грамматики G: случай $|u| \leqslant |s|$.

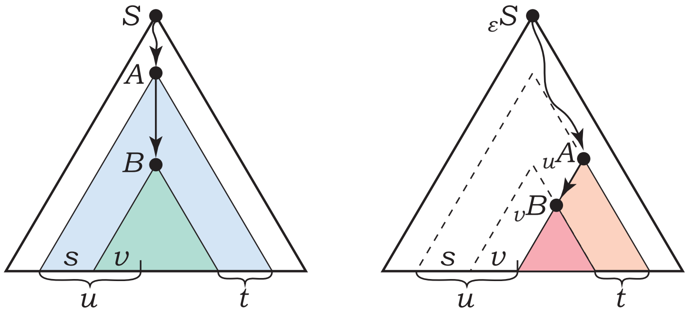

Рис. 6: Откусывание строки u из начала правила $A \to sBt$ грамматики G: случай $|u| > |s|$.

Последние два правила показаны на рисунках 5 и 6. Наконец, правила R_3 нужны для случая, когда вся строка уже прочитана анализатором, и ему нечего заносить в буфер. А именно, для каждого ${}_uA \in N'$, где $|u| \leqslant k-1$ и значение $T(A, u)$ определено, грамматика G' содержит пустое правило.

$$ {}_uA \to \varepsilon $$

Заметим, что $R_1 \cap (R_2 \cup R_3) = \varnothing$, однако множества R_2 и R_3 могут пересекаться по некоторым правилам вида ${}_uA \to \varepsilon$, $|u| = k-1$. Если известно, что правило ${}_uA \to \alpha$ лежит в R_2, то всегда можно однозначно восстановить правило $A \to \gamma$ исходной грамматики, из которого оно получено, исходя из определения откусывания u. Если $\alpha \in \Sigma^*$, то $\gamma = u\alpha$, если $\alpha = s' {}_\varepsilon Bt$, то $\gamma = us'Bt$, и, наконец, если $\alpha = {}_vBt$, то имеем $u = sv$, и $\gamma = sBt$.

Доказательство того, что G' обладает свойством $LL(1)$ и задаёт тот же язык, что и G, даётся в серии утверждений.

Сначала установим заявленное равенство $L_{G'}({}_uA) = \{w \mid uw \in L_G(A)\}$. Как обычно, докажем два включения.

Утверждение 9. *Если $x \in L_{G'}({}_uA)$, то $ux \in L_G(A)$.*

Доказательство. Доказательство проводится индукцией по высоте дерева разбора x из ${}_uA$.

Базовый случай: x задаётся одним правилом ${}_uA \to x$. Все правила R_1 содержат нетерминал, поэтому ${}_uA \to x$ может быть либо из R_2, либо из R_3.

В первом случае ${}_uA \to x$, получено из правила $A \to ux$ в G. Тогда, очевидно, $ux \in L_G(A)$.

Во втором случае $T(A, u)$ определено и $x = \varepsilon$. Так как $T(A, u)$ определено и $u < k$, то u — суффикс входной строки.

Значит $u = x't$, где $x' \in L_G(A)$ и $t \in \mathrm{Follow}(A)$.

Если $|x'| = k - 1$, то, поскольку $|u| \leqslant k - 1$, то $x' = u$. Если же $|x'| < k - 1$, то из отсутствия в G коротких правил, следует, что $t = \varepsilon$ и тем самым снова $x' = x't = u$.

Так или иначе, получаем $ux = u = x' \in L_G(A)$

Индукционный переход. Пусть ${}_uA$ задаёт x по правилу ${}_uA \to \gamma$, и γ содержит нетерминал.

Правило ${}_uA \to \gamma$ лежит либо в R_1, либо в R_2, соответственно имеем два случая.

Пусть ${}_uA \to \gamma$ лежит в R_2 и получено откусыванием u из правила $A \to sBt$ грамматики G. В соответствии с определением R_2, имеем два случая, в зависимости от того, какая из строк u и s длиннее.

- Если $|u| > |s|$, то $\gamma = {}_vBt$, где $u = sv$. Тогда $x = yt$ для некоторого $y \in L_{G'}({}_vB)$. Высота дерева разбора y из ${}_vB$ меньше высоты дерева разбора x из ${}_uA$. Тогда, по предположению индукции, $vy \in L_G(B)$. Следовательно, $ux = uyt = svyt \in L_G(sBt) \in L_G(A)$.
- Если $|u| \leqslant |s|$, то $\gamma = s'{}_\varepsilon Bt$ где $us' = s$. Тогда $x = s'yt$, где $y \in L_{G'}({}_\varepsilon B)$.

  По предположению индукции, $y \in L_G(B)$. Значит $ux = us'yt = syt \in L_G(sBt) \in L_G(A)$.

Пусть $({}_uA \to \gamma) \in R_1$ — правило, наращивающее буфер.

Тогда $\gamma = a_{ua}A$ для некоторого $a \in \Sigma$. Значит $x = ay$, где $y \in L_{G'}({}_{ua}A)$.

Следовательно, по предположению индукции, $uay = ux \in L_G(A)$.

□

Можно доказать похожее включение и для правил.

Утверждение 10. *Пусть правило* $({}_uA \to \gamma) \in R_2$ *получено из правила* $A \to \alpha$. *Тогда, если* $x \in L_{G'}(\gamma)$, *то* $ux \in L_G(\alpha)$.

Доказательство. Если $\alpha \in \Sigma^*$, то по построению $\gamma = x$ и $\alpha = ux$. Пусть теперь $\alpha = sBt$, где $s, t \in \Sigma^*$ и $B \in N$.

В соответствии с определением R_2 имеем два случая.

Если $|u| \leqslant |s|$, то $\gamma = s' {}_\varepsilon Bt$, где $us' = s$. Тогда $x = s'yt$, где $y \in L_{G'}({}_\varepsilon B)$. По утверждению 9 $y \in L_G(B)$, и значит $ux = us'yt = syt \in L_G(sBt)$.

Если $|u| > |s|$, то $\gamma = {}_vBt$ где $u = sv$. Тогда $x = yt$, где $y \in L_{G'}({}_vB)$, и по утверждению 9 $vy \in L_G(B)$. Значит $ux = svyt \in L_G(sBt)$.

□

Утверждение 11. *Если* $ux \in L_G(A)$, *то* $x \in L_{G'}({}_uA)$.

Доказательство. Сначала разберём случай, когда $|ux| < k$ и $T(A, ux)$ определено. Тогда, по построению G' содержит правило ${}_{ux}A \to \varepsilon$.

Пусть $x = x_1 \dots x_m$. Буфер u нетерминала ${}_uA$ может быть расширен до ux с помощью следующих правил.

$$
\begin{aligned}
{}_uA &\to x_1\ {}_{ux_1}A \\
{}_{ux_1}A &\to x_2\ {}_{ux_1x_2}A \\
{}_{ux_1\dots x_{m-1}}A &\to x_m\ {}_{ux}A
\end{aligned}
$$

Эта последовательность в правил, вместе с правилом ${}_{ux}A \to \varepsilon$ составляют вывод x из ${}_uA$. Таким образом, $x \in L_{G'}({}_uA)$.

Теперь предположим, что либо $|ux| \geqslant k$, либо $|ux| = k-1$, но значение $T(A, ux)$ не определено.

Если $|ux| = k - 1$, то из того, что $T(A, ux)$ не определено, следует, что $\mathrm{Follow}(A) \neq \{\varepsilon\}$, и поэтому существует некий символ $c \in \mathrm{Follow}_1(A)$.

Положим $n = k - |u| - 1$ и пусть a будет либо x_{n+1}, в случае $|ux| \geqslant k$, либо c, в случае $|ux| = k - 1$.

Обозначим за u' строку $ux_1 \dots x_n$.

Тогда $|u'| = k - 1$ и $T(A, u'a) = A \to \gamma$, где $A \to \gamma$ — правило, по которому A задаёт ux.

Буфер ${}_uA$ может быть пополнен до u' с помощью следующих правил.

$$
\begin{aligned}
{}_uA &\to x_1\ {}_{ux_1}A \\
{}_{ux_1}A &\to x_2\ {}_{ux_1x_2}A \\
{}_{ux_1\dots x_{n-1}}A &\to x_n\ {}_{u'}A
\end{aligned}
$$

Тогда, так как $T(A, u'a) = A \to \gamma$, грамматика G' содержит правило ${}_{u'}A \to \alpha$, где α получается откусыванием u из γ.

Остаётся доказать, что α задаёт строку $x_{n+1} \dots x_m$. Воспользуемся индукцией по высоте дерева разбора ux из A.

Заметим, что если α задаёт строку $x_{n+1} \dots x_m$, то последовательность правил выше, вместе с правилом ${}_{u'}A \to \alpha$, составляет вывод строки $x_1 \dots x_m = x$ из ${}_uA$, и доказательство будет окончено. Поэтому можно считать, что предположением индукции служит всё доказываемое утверждение $ux \in L_G(A) \Rightarrow x \in L_{G'}({}_uA)$.

Базовый случай: ux **задаётся по правилу** $A \to ux$**.** Тогда $\gamma = T(A, u'a) = A \to ux$, и $\alpha = x_{n+1} \dots x_m$.

Индукционный переход: ux **задаётся по правилу** $A \to sBt$**.**

Тогда $ux = syt$, где $y \in L_G(B)$. и α получается откусыванием u' из строки sBt.

В соответствии с определением R_2 имеем два случая, в зависимости от того, какая из строк u' и s длиннее.

Если $|s| \geqslant |u'|$, то $\alpha = s'{}_{\varepsilon}Bt$, где $u's' = s$.

Высота дерева разбора y из B меньше высоты дерева разбора ux из A. Тогда по предположению индукции $y \in L_{G'}({}_{\varepsilon}B)$.

Следовательно, $u's'yt = syt = ux$, и значит $s'yt = x_{n+1} \dots x_m \in L_{G'}(\alpha)$, что и требовалось доказать.

Если же $|s| < |u'|$, то $\alpha = {}_vBt$, где $sv = u'$.

Чтобы воспользоваться предположением индукции для нетерминалов B и ${}_vB$ и строки y, необходимо показать, что v является префиксом y.

Так как в G нет коротких правил, то либо $|y| \geqslant k - 1$, либо $t = \varepsilon$.

Если $t = \varepsilon$, то $ux = sy$, и поскольку $sv = u'$ является префиксом ux, то v является префиксом y.

Если же $|y| = k - 1$, то и $|sy| \geqslant k - 1$. Поскольку u' — это префикс $ux = syt$, и $|u'| \leqslant k - 1$, то строка $u' = sv$ является префиксом sy, и следовательно v — префикс y.

Таким образом, $y = vy'$ для некоторой строки $y' \in \Sigma^*$, и по предположению индукции $y' \in L_{G'}({}_vB)$.

Получаем $u'y't = u'syt = ux$, и значит $y't = x_{n+1} \dots x_m \in L_{G'}(\alpha)$, что и требовалось доказать.

$\square$

Следующее утверждение устанавливает включение $\mathrm{Follow}({}_uA) \subseteq \mathrm{Follow}(A)$.

Утверждение 12. *Если* $y \in \mathrm{Follow}({}_uA)$*, то* $y \in \mathrm{Follow}(A)$.

Доказательство. По определению $\mathrm{Follow}({}_uA)$ существует поддерево ${}_uA$ некоторого дерева разбора, за которым следует строка y. Доказательство того, что $y \in \mathrm{Follow}(A)$ проводится индукцией по глубине поддерева.

Базовый случай. Пусть ${}_uA$ является корнем всего дерева разбора, то есть ${}_uA = {}_\varepsilon S$.

В этом случае $y = \varepsilon$, поскольку справа от всего дерева разбора листьев нет. Таким образом, $y = \varepsilon \in \mathrm{Follow}(S)$.

Индукционный переход. Пусть ${}_vB$ — родитель ${}_uA$ в дереве разбора, пусть ${}_vB \to \gamma$ — соответствующее правило, и за поддеревом ${}_vB$ следует строка y'.

Глубина поддерева с корнем ${}_vB$ меньше глубины поддерева с корнем ${}_uA$, и так как $y' \in \mathrm{Follow}({}_vB)$, то по индукционному предположению $y' \in \mathrm{Follow}(B)$

Поскольку γ содержит нетерминал ${}_uA$, правило ${}_vB \to \gamma$ лежит либо в R_1, либо в R_2.

В первом случае правило ${}_vB \to \gamma$ наращивает буфер, и следовательно $B = A$ и $u = va$ для некоторого $a \in \Sigma$. Соответственно, ${}_vB_{\to}\gamma = {}_vA \to a\,{}_{va}A$. Поэтому $y' = y$, и значит $y \in \mathrm{Follow}(A)$.

Во втором случае γ получается откусыванием u из w_1Aw_2, где $B \to w_1Aw_2$ — правило исходной грамматики. Поэтому $y = w_2y'$. Так как $y' \in \mathrm{Follow}(B)$, то по утверждению 5 $w_2y' = y \in \mathrm{Follow}(A)$.

□

Грамматика G' линейна по построению и по утверждениям 9 и 11 $L(G') = L_{G'}({}_\varepsilon S) = L_G(S) = L(G)$. Остаётся показать, что G' принадлежит классу $LL(1)$.

Утверждение 13. *Грамматика* G' *принадлежит классу LL(1).*

Рассмотрим любые два правила ${}_uA \to \gamma_1$ и ${}_uA \to \gamma_2$ и покажем, если множества $\mathrm{First}_1(L(\gamma_1)\mathrm{Follow}({}_uA))$ и $\mathrm{First}_1(L(\gamma_2)\mathrm{Follow}({}_uA))$ пересекаются, то $\gamma_1 = \gamma_2$.

Доказательство проводится отдельно для случаев $|u| < k-1$ и $|u| = k-1$. Пусть сначала $|u| < k-1$.

Тогда каждое из правил ${}_uA \to \gamma_1$ и ${}_uA \to \gamma_2$ лежит в R_1 или R_3. Разберём случаи.

- Если оба правила лежат в R_3, то $\gamma_1 = \gamma_2 = \varepsilon$.

- Если оба правила лежат в R_1, то $\gamma_1 = a\,{}_{ua}A$ и $\gamma_2 = b\,{}_{ub}A$ для некоторых символов a и b. Тогда $\mathrm{First}_1(L(\gamma_1)\mathrm{Follow}({}_uA)) = \{a\}$ и $\mathrm{First}_1(L(\gamma_2)\mathrm{Follow}({}_uA)) = \{b\}$, и очевидно, что если последние множества пересекаются, то $\gamma_1 = \gamma_2$

- Пусть одно из правил лежит в R_1, а другое — в R_3. Не умаляя общности, $\gamma_1 = a\,{}_{ua}A$ для некоторого $a \in \Sigma$, а $\gamma_2 = \varepsilon$. Тогда $\varepsilon \in L_{G'}({}_uA)$, и $u \in L_G(A)$ по утверждению 9. Поскольку в G нет коротких правил, и $|u| < k-1$, то $\mathrm{Follow}(A) = \{\varepsilon\}$, а значит $\mathrm{Follow}({}_uA) = \{\varepsilon\}$ по утверждению 12. Тогда $\mathrm{First}_1(L(\gamma_2)\mathrm{Follow}({}_uA)) = \{\varepsilon\}$, и так как $\mathrm{First}_1(L(\gamma_1)\mathrm{Follow}({}_uA)) = \{a\}$, то последние множества снова не пересекаются.

Пусть теперь $|u| = k-1$. Тогда, каждое из правил ${}_uA \to \gamma_1$, ${}_uA \to \gamma_2$ лежит в R_2 или в R_3. Снова разберём случаи.

- Если оба правила лежат в R_3, то $\gamma_1 = \gamma_2 = \varepsilon$.

- Пусть, скажем, ${}_uA \to \gamma_1$, лежит в R_2 и получено из правила $A \to \alpha$, а ${}_uA \to \gamma_2$ лежит в R_3. Тогда $\gamma_2 = \varepsilon$, и значит, во-первых, $u \in L_G(A)$, и во-вторых, $\mathrm{First}_1(L_{G'}(\gamma_2)\mathrm{Follow}({}_uA)) = \mathrm{First}_1(\mathrm{Follow}({}_uA))$.

  Предположим, γ_1 задаёт пустую строку. Тогда α задаёт u по утверждению 10. По определению R_2 имеем $(A \to \alpha) = T(A, ua)$ для некоторого $a \in \Sigma$, и поэтому $\mathrm{Follow}(A) \neq \{\varepsilon\}$. Тогда $\alpha = u$ по утверждению 8, и так как γ_1 получается откусыванием u из α, то $\gamma_1 = \varepsilon = \gamma_2$.

  Пусть теперь γ_1 не задаёт пустой строки, и тем самым $\mathrm{First}_1(L_{G'}(\gamma_1)\mathrm{Follow}({}_uA)) = \mathrm{First}_1(L_{G'}(\gamma_1))$.

  То, что $\varepsilon \notin \mathrm{First}_1(L_{G'}(\gamma_1))$ уже предполагается. Пусть $c \in \mathrm{First}_1(L_{G'}(\gamma_1))$, где $c \in \Sigma$. Тогда $T(A, uc) = (A \to \alpha)$.

  Предположим, что $c \in \mathrm{First}_1(L_{G'}(\gamma_2)\mathrm{Follow}({}_uA)) = \mathrm{First}_1(\mathrm{Follow}({}_uA))$ Тогда из утверждения 12 следует, что $c \in \mathrm{First}_1(\mathrm{Follow}(A))$. Рассмотрим дерево разбора с поддеревом A, за которым следует строка, начинающаяся с символа c.

  Так как $u \in L_G(A)$, то можно считать, что это поддерево задаёт строку u, как показано на рисунке 7.

  Так как $|u| = k-1$ и $\mathrm{Follow}(A) \neq \{\varepsilon\}$, то из утверждения 8 следует, что A задаёт u по правилу $A \to u$.

  Тогда по определению LL(k)-таблицы имеем $T(A, uc) = (A \to u)$. Поэтому $\alpha = u$, и снова $\gamma_1 = \varepsilon = \gamma_2$.

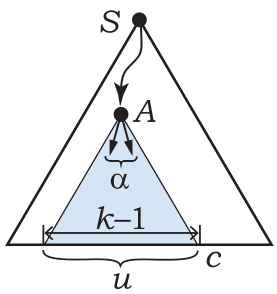

Рис. 7: Иллюстрация к доказательству того, что G' принадлежит классу LL(1)

- Пусть теперь оба правила ${}_uA \to \gamma_1$ и ${}_uA \to \gamma_2$ лежат в R_2 и получены из правил $A \to \alpha_1$ и $A \to \alpha_2$ соответственно.

  Пусть $x \in \mathrm{First}_1(L(\gamma_1)\mathrm{Follow}({}_uA)) \cap \mathrm{First}_1(L(\gamma_2)\mathrm{Follow}({}_uA))$, где $x \in \Sigma$ или $x = \varepsilon$.

  Из утверждений 10 и 12 получаем $ux \in \mathrm{First}_k(L(\alpha_1)\mathrm{Follow}(A)) \cap \mathrm{First}_k(L(\alpha_2)\mathrm{Follow}(A))$.

  Тогда, так как G принадлежит классу LL(k), то $\alpha_1 = \alpha_2$, и значит снова имеем $\gamma_1 = \gamma_2$. $\square$

Построения в утверждениях 1 и 3 приводят к увеличению количества нетерминалов в $|\Sigma^{\leqslant k-1}|$ раз, а построение леммы 2 не меняет числа нетерминалов.

Таким образом, вместе упомянутые леммы влекут заявленный результат.

Теорема 1. *Для каждой линейной LL(k) грамматики $G = (\Sigma, N, R, S)$ существует линейная LL(1) грамматика с $|N| \cdot |\Sigma^{\leqslant k-1}|^2$ нетерминальными символами, которая задаёт тот же язык.*

6 Нижняя оценка

Ранее показывалось, как по линейной LL(k) грамматике G с множеством нетерминальных символов N построить линейную LL(1) грамматику с $|N| \cdot |\Sigma^{\leqslant k-1}|^2$ нетерминальными символами, задающую тот же язык.

Покажем, что приведённое построение близко к оптимальному, и в худшем случае необходимо почти столько же нетерминальных символов.

Теорема 2. *Для каждых натуральных $m \geqslant 3$, $k \geqslant 4$ и $n \geqslant 1$ существует язык над m-символьным алфавитом, задаваемый линейной LL(k) грамматикой G с n нетерминальными символами, такой что каждая линейная LL(1) грамматика, задающая тот же язык, имеет не менее $n \cdot (m-1)^{2k-3-\lceil \log_{m-1} k \rceil}$ нетерминалов.*

Линейная LL(k) грамматика G, задающая искомый язык будет определена над m-символьным алфавитом $\Sigma \cup \{\#\}$, где $\#$ — специальный символ, не лежащий в Σ.

Грамматика будет содержать правила вида $A \to xAf(x)$, где $x \in \Sigma^{k-1}\#$, а $f\colon \Sigma^{k-1}\# \to \Sigma$ — некоторая функция, которая будет определена позже, а также правила $A \to \varepsilon$.

Функция f будет устроена таким образом, чтобы любому синтаксическому анализатору для линейной LL(1) грамматики, задающей тот же язык, что и G, было необходимо хранить много информации в своём стеке.

Пусть $1 \leqslant C \leqslant k-1$ — фиксированная константа.

Рассмотрим момент, когда LL(1) анализатор прочёл первые $k-C$ символов $c_1 \dots c_{k-C}$ очередного блока $x \in \Sigma^{k-1}\#$ входной строки, и до конца блока ему осталось прочесть ещё C символов $d_{k-C+1} \dots d_{k-1}\#$.

Чтобы определить значение $f(x)$ после прочтения блока x, в указанный момент LL(1) анализатору необходимо помнить проекцию функции f на свои последние C аргументов $d_{k-C+1} \dots d_{k-1}\#$

Обозначим последнюю проекцию за g, то есть $g(d_{k-C+1} \dots d_{k-1}\#) = f(c_1 \dots c_{k-C} d_{k-C+1} \dots d_{k-1}\#)$.

Постараемся выбрать функцию f и константу C так, чтобы количество возможных её возможных проекций $g\colon \Sigma^{C-1}\# \to \Sigma$ было как можно бо́льшим

Лемма 4. *Для $C = \lceil \log_{|\Sigma|} k \rceil + 1$ существует сюръективная функция $f\colon \Sigma^{k-1}\# \to \Sigma$, для которой все проекции g_s, получаемые подстановкой на место первых $k-C$ аргументов f символов s (то есть $g_s(t) = f(st)$), где $s \in \Sigma^{k-C}$, попарно различны.*

Доказательство. Всего существует $|\Sigma|^{|\Sigma|^{C-1}}$ возможных функций $g\colon \Sigma^{C-1}\# \to \Sigma$.

Чтобы различным строкам $s \in |\Sigma|^{k-C}$ поставить в соответствие различные функции g_s, количество строк не должно превышать количество возможных функций.

$$|\Sigma|^{k-C} \leqslant |\Sigma|^{|\Sigma|^{C-1}}$$

Это неравенство выполняется, так как $C = \lceil \log_{|\Sigma|} \rceil k + 1$ влечёт $k - C \leqslant |\Sigma|^{C-1}$.

Тогда строки $s \in \Sigma^{k-C}$ возможно инъективно отобразить в функции $g_s\colon \Sigma^{C-1}\# \to \Sigma$, и искомая функция f определяется как $f(st) = g_s(t)$, для всех $s \in \Sigma^{k-C}$ и $t \in \Sigma^{C-1}\#$.

Предположим, что построенная таким образом функция f не сюръективна, то есть существует множество символов $\{a_1, \dots, a_r\} \subset \Sigma$, где $r \geqslant 1$, такое что ни одна из строк $\Sigma^{k-1}\#$ не отображается f ни в один из этих символов.

Рассмотрим прообраз $f^{-1}(b)$ каждого символа $b \in \Sigma$. По крайней мере один прообраз $f^{-1}(b)$ содержит не менее $\frac{|\Sigma^{k-1}\#|}{|\Sigma|} > |\Sigma| > r$ строк.

Пусть $x_1, \ldots, x_r$ — любые r различных строк из $f^{-1}(b)$. Рассмотрим функцию $f' \colon \Sigma^{k-1}\# \to \Sigma$ определяемую следующим образом: если $x \neq x_1, \ldots, x_r$, то $f'(x) = f(x)$, и $f(x_j) = a_j$.

Как несложно убедиться, функция f' сюръективна: для символов $a \in \Sigma \setminus \{a_1, \ldots, a_r, b\}$ имеем $f'^{-1}(a) = f^{-1}(a) \neq \varnothing$, $f'^{-1}(b) = f^{-1}(b) \setminus \{x_1, \ldots, x_r\} \neq \varnothing$ и $f'^{-1}(a_j) = x_j$.

Покажем, что все проекции $g'_s(t) = f'(st)$ попарно различны. Рассмотрим любые две различные строки $s_1, s_2 \in \Sigma^{k-C}$. По построению значения функций g_{s_1} и g_{s_2} различаются на некоторой строке t и, соответственно, имеем три случая:

1. Ни одна из строк s_1t и s_2t не совпадает с какой-то из $x_1, \ldots, x_r$. Тогда $g'_{s_1}(t) = f'(s_1t) = f(s_1t) = g_{s_1}(t)$ и $g'_{s_2}(t) = f'(s_2t) = f(s_2t) = g_{s_2}(t)$, значит $g'_{s_1}(t) \neq g'_{s_2}(t)$.

2. Только одна из строк s_1t, s_2t, скажем s_1t, совпадает с некоторой строкой x_j. Тогда $f'(s_1t) = a_j$ и $f'(s_2t) = f(s_2t)$. Так как $f(s_2t) \notin \{a_1, \ldots, a_r\}$, то $f'(s_1t) \neq f'(s_2t)$ и значит $g'_{s_1}(t) \neq g'_{s_2}(t)$.

3. Пусть $s_1t = x_i$ и $s_2t = x_j$ для некоторых $i, j \in \{1, \ldots, r\}$. Так как строки s_1t and s_2t различны, то $x_i \neq x_j$, и значит $f'(s_1t) = a_i \neq a_j = f'(s_2t)$.

$\square$

Пусть далее f — некая фиксированная функция из леммы 4. Заявленная грамматика $G = (\Sigma \cup \{\#\}, N, R, S)$ имеет множество нетерминалов $N = \{A_1, \ldots, A_n\}$, где $S = A_1$.

Каждый нетерминал A_i ставит в соответствие блокам $x \in \Sigma^{k-1}\#$ в начале строки символы $f(x)$ в конце строки.

$$A_i \to xA_if(x) \qquad (1 \leqslant i \leqslant n,\ x \in \Sigma^{k-1}\#)$$

Внизу дерева разбора есть две возможности. Во-первых, нетерминал может задавать пустую строку.

$$A_i \to \varepsilon \qquad (1 \leqslant i \leqslant n)$$

То, что блоки вида $\Sigma^{k-1}\#$ закончились, LL(1) анализатор сможет понять лишь спустя k символов, когда в конце очередного предполагаемого блока не окажется символа $\#$. Во-вторых, может быть явно выписан номер нетерминального символа.

$$A_i \to b\#^i \qquad (1 \leqslant i \leqslant n)$$

Последние правила нужны, чтобы нетерминалы LL(1) грамматики так или иначе хранили информацию о номере i. Наконец, номер нетерминального символа может быть изменён с помощью правила

$$A_i \to \# A_{i+1} \qquad\qquad (1 \leqslant i < n)$$

Так определяется линейная LL(k) грамматика G, заявленная в Теореме 2.

Опишем её LL таблицу. Пусть A_i — нетерминал на вершине стека LL(k) анализатора.

- Если анализатор видит в начале непрочтённого суффикса строки $\#$, то он понимает, что должно быть применено правило $A_i \to \# A_{i+1}$, поэтому $T(A_i, \# w) = (A_i \to \# A_{i+1})$ для всех строк $w \in (\Sigma \cup \{\#\})^*$, таких что $\# w \in \mathrm{First}_k(L_G(A_i)\mathrm{Follow}(A_i))$ (для остальных w значения $T(A_i, \# w)$ не определены).

- Если анализатор перед собой $b\#$, то он понимает, что должно быть применено правило $A_i \to b\#^i$, поэтому $T(A_i, b\#^i w) = (A_i \to b\#^i)$ для всех строк $w \in \Sigma^{\leqslant k-i-1}$.

- Если анализатор видит перед собой блок $x \in \Sigma^{k-1}\#$, то нужное правило — $A_i \to x A_i f(x)$, и поэтому $T(A_i, x) = (A_i \to x A_i f(x))$ для всех $x \in \Sigma^{k-1}\#$.

- Наконец, если анализатор не видит впереди ни одного символа $\#$, то нужное правило $A_i \to \varepsilon$, и тем самым $T(A_i, w) = (A_i \to \varepsilon)$ для всех $w \in \Sigma^{\leqslant k}$.

Докажем, что каждая линейная LL(1) грамматика $G' = (\Sigma \cup \{\#\}, N', R', S')$, задающая тот же язык, что и G, должна иметь по крайней мере столько нетерминальных символов, сколько указано в теореме.

Доказательство использует следующие строки из языка $L(G)$ (грамматика G порождает и другие строки).

$$\#^{i-1} x_1 \dots x_\ell f(x_\ell) \dots f(x_1), \qquad \text{где } i, \ell \geqslant 1,\ x_1, \dots, x_\ell \in \Sigma^{k-1}\#$$
$$\#^{i-1} x_1 \dots x_\ell b\#^i f(x_\ell) \dots f(x_1), \qquad \text{где } i, \ell \geqslant 1,\ x_1, \dots, x_\ell \in \Sigma^{k-1}\#$$

Рассмотрим префикс строки указанного выше вида.

Пусть $i \in \{1, \dots, n\}$, $x_1, \dots, x_k \in \Sigma^{k-1}\#$, и $a_1, \dots, a_{k-C} \in \Sigma$.

Положим

$$u = \#^{i-1} x_1 \dots x_k a_1 \dots a_{k-C}$$

Так как в строках из $L(G)$ после префикса u может идти любой символ Σ, то после прочтения LL(1) анализатором для G' строки u содержимое его стека начинается с нетерминала, и значит имеет вид Av, где $A \in N'$, а $v \in (\Sigma \cup \{\#\})^*$.

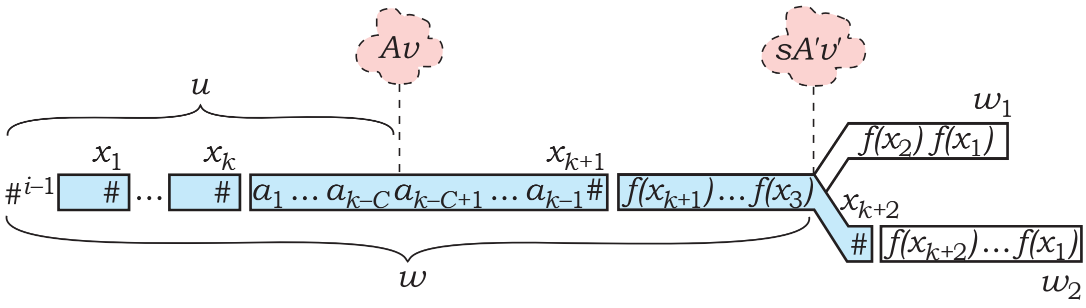

Рис. 8: Иллюстрация к доказательству леммы 5

Мы покажем, что для большого множества строк u указанного вида соответствующие нетерминалы A на вершине стека должны быть попарно различны.

Сперва покажем, что строка v короткая, и таким образом бо́льшая часть информации, находящейся в стеке анализатора, закодирована в нетерминале A.

Лемма 5. *Пусть линейная LL(1) грамматика* $G' = (\Sigma \cup \{\#\}, N', R', S')$ *задаёт язык* $L(G)$*, и пусть в стеке синтаксического анализатора для* G' *после прочтения строки вида* $u = \#^{i-1} x_1 \dots x_k a_1 \dots a_{k-C}$*, где* $i \in \{1, \dots, n\}$*,* $x_1, \dots, x_k \in \Sigma^{k-1}\#$*, и* $a_1, \dots, a_{k-C} \in \Sigma$*, находится* Av*. Тогда* $|v| \leqslant 2$*.*

Доказательство. Пусть $a_{k-C+1}, \dots, a_{k-1} \in \Sigma$ любые символы. Рассмотрим следующие два блока из $\Sigma^{k-1}\#$.

$$x_{k+1} = a_1 \dots a_{k-1}\#$$
$$x_{k+2} = f(x_{k+1}) f(x_k) \dots f(x_3)\#$$

Заметим, что начало x_{k+2} выглядит так, будто в исходной грамматике было применено правило $A_i \to \varepsilon$.

Рассмотрим следующие две строки, получающиеся дописыванием в конец u сначала, соответственно, одного блока x_{k+1} и двух блоков x_{k+1} и x_{k+2}, а затем — образов всех блоков под действием f (см. рисунок 8).

$$w_1 = \#^{i-1} x_1 \dots x_k \boldsymbol{x_{k+1}} f(x_{k+1}) f(x_k) \dots f(x_3) f(x_2) f(x_1)$$
$$w_2 = \#^{i-1} x_1 \dots x_k \boldsymbol{x_{k+1}} \boldsymbol{x_{k+2}} f(x_{k+2}) f(x_{k+1}) f(x_k) \dots f(x_3) f(x_2) f(x_1)$$

Обе строки начинаются с u, лежат в $L(G)$ и имеют общий префикс

$$w = \#^{i-1} x_1 \dots x_k x_{k+1} f(x_{k+1}) f(x_k) \dots f(x_4) f(x_3)$$

Так как по крайней мере две строки из $L(G)$, а именно, w_1 и w_2, начинаются с w, то в стеке LL(1) анализатора после прочтения w есть

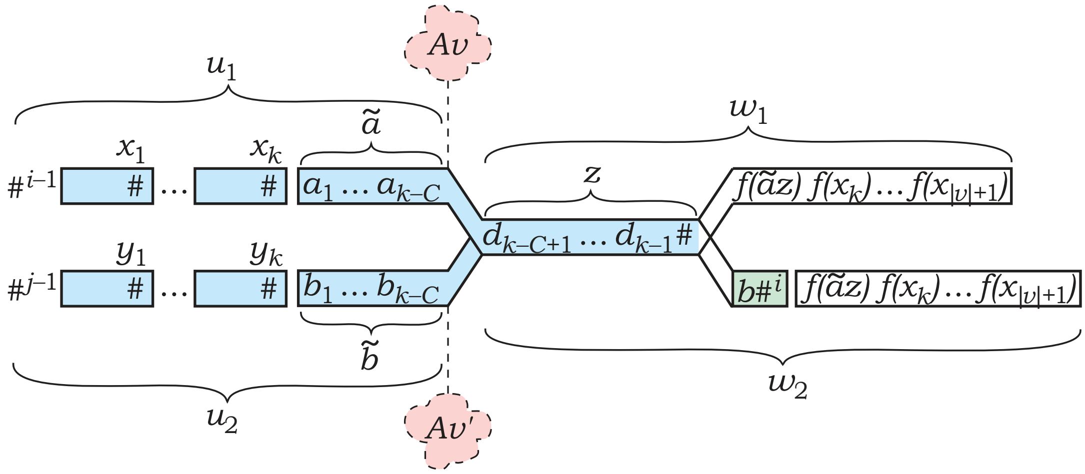

Рис. 9: Иллюстрация к доказательству леммы 6

нетерминал, то есть там лежит строка вида $sA'v'$, где v' заканчивается на v.

Тогда суффикс $f(x_2)f(x_1)$ строки w_1 должен представляться в виде syv', где $y \in L_{G'}(A')$.

Значит $|s| + |v'| \leqslant 2$, и так как v — суффикс v', то получаем $|v| \leqslant 2$. $\square$

Рассмотрим теперь *две* различные строки того же вида, что и строка u из леммы 5

Покажем, что если эти строки существенно различаются, то и нетерминалы, находящиеся на вершине стека анализатора после прочтения им этих строк, должны быть различны.

Лемма 6. *Пусть $G' = (\Sigma \cup \{\#\}, N', R', S')$ — линейная LL(1) грамматика, задающая язык $L(G)$. Рассмотрим две строки следующего вида.*

$$u_1 = \#^{i-1}x_1 \dots x_k a_1 \dots a_{k-C} \quad (A_i \in N,\ x_1, \dots, x_k \in \Sigma^{k-1}\#,\ a_1, \dots, a_{k-C} \in \Sigma)$$
$$u_2 = \#^{j-1}y_1 \dots y_k b_1 \dots b_{k-C} \quad (A_j \in N,\ y_1, \dots, y_k \in \Sigma^{k-1}\#,\ b_1, \dots, b_{k-C} \in \Sigma)$$

Пусть либо $i \neq j$, либо $f(x_3) \dots f(x_k) \neq f(y_3) \dots f(y_k)$, либо $a_1 \dots a_{k-C} \neq b_1 \dots b_{k-C}$, и после прочтения строк u_1 и u_2 на в стеке анализатора лежат строки Av и Bv' соответственно. Тогда $A \neq B$.

Доказательство. Предположим, что $A = B$, и значит после прочтения u_1 и u_2 в стеке лежит Av и Av' соответственно.

Положим $\widetilde{a} = a_1 \dots a_{k-C}$ и $\widetilde{b} = b_1 \dots b_{k-C}$ — последние незавершенные блоки строк u_1 и u_2.

Если $\widetilde{a} \neq \widetilde{b}$, то по построению f блоки $\widetilde{a}$ и $\widetilde{b}$ можно продолжить некоторой строкой $z = d_{k-C+1} \dots d_{k-1}\#$ длины C, где $d_{k-C+1}, \dots, d_{k-1} \in \Sigma$, так что $f(\widetilde{a}z) \neq f(\widetilde{b}z)$.

Если же $\widetilde{a} = \widetilde{b}$, то продолжим блоки $\widetilde{a}$ и $\widetilde{b}$ любой строкой $z = d_{k-C+1} \dots d_{k-1}\#$, где $d_{k-C+1}, \dots, d_{k-1} \in \Sigma$.

Рассмотрим следующие две строки из $L(G)$.

$$u_1 z f(\widetilde{a}z) f(x_k) \dots f(x_1)$$
$$u_1 z b \#^i f(\widetilde{a}z) f(x_k) \dots f(x_1)$$

Так как после прочтения анализатором их общего префикса u_1, в стеке лежит Av, то оба оставшихся суффикса должны лежать в $L_{G'}(Av)$ (см. рисунок 9).

Тогда префиксы этих суффиксов, содержащие все символы, кроме последних $|v|$, лежат в $L_{G'}(A)$. Обозначим последние строки за w_1 и w_2 соответственно.

$$w_1 = z f(\widetilde{a}z) f(x_k) \dots f(x_{|v|+1})$$
$$w_2 = z b \#^i f(\widetilde{a}z) f(x_k) \dots f(x_{|v|+1})$$

(заметим, что $|v| \leqslant 2$ по лемме 5).

По условию леммы, u_1 и u_2 должны различаться либо в количестве знаков $\#$ в начале ($i \neq j$), либо в каком-то из образов последних $k-2$ завершенных блоков ($f(x_3) \dots f(x_k) \neq f(y_3) \dots f(y_k)$), либо в каком-то из символов последнего незавершенного блока ($\widetilde{a} \neq \widetilde{b}$). Соответственно, нужно рассмотреть три случая.

- Пусть $i \neq j$. Так как в стеке анализатора после прочтения им u_2 лежит строка Av', то строка $u_2 w_2 v'$ должна принадлежать языку $L(G)$.

  $$u_2 w_2 v' = \#^{j-1} y_1 \dots y_k \widetilde{b} z b \#^i f(\widetilde{a}z) f(x_k) \dots f(x_{|v|+1}) v'$$

  Предполагаемое дерево разбора $u_2 w_2 v'$ несложно восстановить, пользуясь LL(k) таблицей G: сначала применяется $j-1$ правил $A_1 \to \# A_2, \dots, A_{j-1} \to \# A_j$, увеличивающих номер нетерминала, затем $k+1$ правил $A_j \to y_1 A_j f(y_1), \dots, A_j \to y_k A_j f(y_k)$, $A_j \to \widetilde{b} z A_j f(\widetilde{b}z)$, сопоставляющих блокам $y_1, \dots, y_k, \widetilde{b}z \in \Sigma^{k-1}\#$ символы $f(y_1), \dots, f(y_k), f(\widetilde{b}z)$, и, наконец, правило $A_j \to b\#^j$.

  Таким образом должно выполняться $i = j$, что противоречит сделанному предположению.

- В случае, когда $f(x_3) \dots f(x_k) \neq f(y_3) \dots f(y_k)$, аналогично предыдущему пункту получаем, что строка $u_2 w_1 v'$ лежит в $L(G)$.

  $$u_2 w_1 v' = \#^{j-1} y_1 \dots y_k b_1 \dots b_{k-C} z f(\widetilde{a}z) f(x_k) \dots f(x_{|v|+1}) v'$$

  Как и в предыдущем случае, можно явно рассмотреть предполагаемое дерево разбора $u_2 w_1 v'$ и прийти к противоречию с тем, что$f(x_3) \dots f(x_k) \neq f(y_3) \dots f(y_k)$,

- Пусть $\widetilde{a} \neq \widetilde{b}$. Тогда с одной стороны, как и в предыдущем случае, строка $u_2 w_1 v'$ лежит в $L(G)$, но с другой стороны по выбору z имеем $f(\widetilde{a}z) \neq f(\widetilde{b}z)$, и можно снова рассмотреть предполагаемое дерево разбора $u_2 w_1 v'$ и прийти к противоречию.

□

Теоремы 2. Пусть $i \in \{1, \dots, n\}$, $d_3, \dots, d_k \in \Sigma$, $a_1, \dots, a_{k-C} \in \Sigma$. Так как f сюръективна, существуют строки $x_1, \dots, x_k \in \Sigma^{k-1}\#$, такие что $f(x_j) = d_j$ для всех $j \in \{3, \dots, n\}$.

Определим строки $u_{i;d_3,\dots,d_k;a_1,\dots,a_{k-C}}$ следующим образом.

$$u_{i;d_3,\dots,d_k;a_1,\dots,a_{k-C}} = \#^{i-1} x_1 \dots x_k a_1 \dots a_{k-C}$$

По лемме 6, после прочтения LL(1) анализатором различных строк указанного вида, на вершине его стека должны находиться различные нетерминальные символы.

Следовательно, в грамматике G' должно быть не меньше нетерминалов, чем существует различных строк $u_{i;d_3,\dots,d_k;a_1,\dots,a_{k-C}}$, то есть $|N'| \geqslant n \cdot (m-1)^{2k-C-2}$, как и заявлено.

□

7 Конъюнктивные линейные LL(k) грамматики

Языки, задаваемые нетерминалами формальной грамматики, можно определять как решения некоторых языковых уравнений.

Так, каждое правило $A \to uBv$ означает, что $L(A) \supseteq L(uBv)$, и если $A \to u_1 B_1 v_1$, $\dots$, $A \to u_k B_k v_k$ — все правила для нетерминала A, то язык, задаваемый A, определяется как $L(A) = L(u_1 B_1 v_1) \cup \dots \cup L(u_k B_k v_k)$.

Переменными в таких уравнениях являются языки, соответствующие нетерминалам грамматики, а также константные языки $\{s\}$, $s \in \Sigma^*$, а допустимыми операциями — конкатенация и объединение.

Конъюнктивные грамматики обобщают обычные грамматики путём добавления операции *пересечения*. Правила конъюнктивной грамматики имеют вид $A \to u_1 B_1 v_1 \,\&\, \dots \,\&\, u_k B_k v_k$ и означают, что $L(A) \supseteq L(u_1 B_1 v_1) \cap \dots \cap L(u_k B_k v_k)$. Как и прежде, если $A \to \gamma_1, \dots, A \to \gamma_k$ — все правила для нетерминала A, то язык, задаваемый A, определяется как $L(A) = L(\gamma_1) \cup \dots \cup L(\gamma_k)$.

Дадим формальное определение.

Определение 8. ***Конъюнктивной линейной (формальной) грамматикой*** *называется четвёрка $G = (\Sigma, N, R, S)$, состоящая из следующих компонентов:*

1. *Σ — конечное множество символов, называемое* ***алфавитом****.*

2. *N — конечное множество* ***нетерминалов****. Каждый нетерминал обозначает некоторое свойство, которым строка из Σ^* может обладать или не обладать.*

3. *R — конечное множество* ***правил*** *грамматики, каждое из которых описывает возможную структуру строк со свойством $A \in N$. Каждое правило имеет вид $A \to u_1B_1v_1 \ \& \ u_2B_2v_2 \ \& \ \cdots \ \& \ u_rB_rv_r$, где $B_1, \ldots, B_r \in N$ — нетерминалы, а $u_1, v_1, \ldots, u_r, v_r \in \Sigma^*$ — строки или вид $A \to x$, где $x \in \Sigma^*$. Строки вида uBv, входящие в правила, называются* ***конъюнктами****.*

   Наличие правила $A \to u_1B_1v_1 \ \& \ u_2B_2v_2 \ \& \ \cdots \ \& \ u_rB_rv_r$ означает, что если строку w можно представить в виде конкатенации $w = u_is_iv_i$, где s_i обладает свойством B_i, для каждого $i = 1, \ldots, r$, то строка w обладает свойством A. Правила вида $A \to x$, где $x \in \Sigma^*$, как и стоит ожидать, означают, что x обладает свойством A.

4. *$S \in N$ — выделенный* ***начальный*** *нетерминал.*

Столь же важным, сколь и в неконъюнктивном случае, является понятие *дерева разбора*. Стоит заметить, что деревья разбора конъюнктивной грамматики получаются склеиванием некоторых листьев упорядоченного корневого дерева, и тем самым представляют из себя графы, в строгом смысле деревьями не являющиеся. Однако мы всё равно будем называть их деревьями, их самые нижние вершины — листьями, а также использовать и другую «древесную» терминологию.

Определение 9. *Деревья разбора конъюнктивной грамматики проще всего определить индуктивно. Каждому правилу вида $A \to w$, $w \in \Sigma^*$ соответствует дерево разбора w из A высоты 1, такое же, как и в неконъюнктивном случае: вершина, помеченная A, является корнем, и от неё отходят листья, образующие строку w.*

Пусть теперь в G есть правило $A \to u_1B_1v_1 \ \& \cdots \& \ u_rB_rv_r$, нетерминалы $B_1, \ldots, B_r$ задают строки $s_1, \ldots, s_r$, и соответствующие деревья разбора имеют высоту не более $n-1$, причём $u_1s_1v_1 = \cdots = u_rs_rv_r = w$. Тогда будем говорить, что A задаёт w по правилу $A \to u_1B_1v_1 \ \& \cdots \& \ u_kB_kv_k$, и соответствующее дерево разбора имеет высоту n и получается из деревьев разбора s_j из B_j следующим образом.

1. *Каждому дереву разбора s_j из B_j ставится в соответствие «дерево разбора» w из A по «правилу» $A \to u_jB_jv_j$, как показано на рисунке 10. Заметим, что правила $A \to u_jB_jv_j$ в грамматике нет, поэтому данное дерево разбора сугубо промежуточное и не является корректным деревом разбора w из A.*

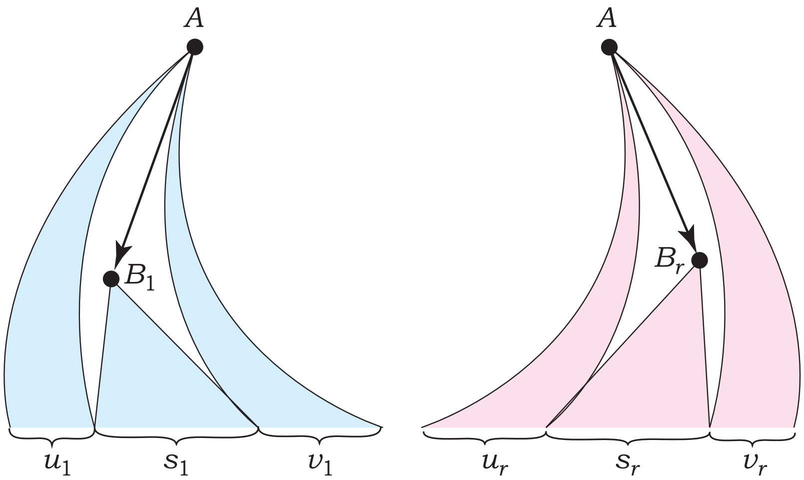

Рис. 10: Промежуточные деревья разбора

2. *Корни и листья всех этих деревьев склеиваются, как показано на рисунке 11. Таким образом получается, что, A задаёт w по всем конъюнктам $u_j B_j v_j$ сразу.*

Язык задаваемый нетерминалом A обозначается $L(A)$ и определяется как множество всех строк w, задаваемых A за некоторое число шагов. Соответствующие деревья разбора называются деревьями разбора w из A.

Язык задаваемый грамматикой обозначается L_G и определяется как язык, задаваемый её начальным нетерминалом S.

Язык, задаваемый конъюнкцией $\varphi = \alpha_1 \,\&\, \ldots \,\&\, \alpha_r$, определяется как $L(\varphi) = L(\alpha_1) \cap \cdots \cap L(\alpha_r)$.

Определение 10. *Пусть $G = (\Sigma, N, R, S)$ — конъюнктивная формальная грамматика.*

Пусть зафиксировано некоторое дерево разбора G и поддерево в нём.

Будем говорить, что строка $y \in \Sigma^*$ ***следует*** *за этим поддеревом, если все листья справа от поддерева образуют строку y, как на рисунке 1(в середине).*

Так же, как и в неконъюнктивном случае, определяется класс LL(k).

Определение 11. *LL(k)-таблицей для конъюнктивной грамматики $G = (\Sigma, N, R, S)$ называется частично определённая функция $T : N \times \Sigma^{\leqslant k} \to R$, удовлетворяющая следующему условию. $T(A, x) = (A \to \alpha)$, где $A \in N, x \in \Sigma^{\leqslant k}$, тогда и только тогда, когда существует поддерево некоторого дерева разбора, помеченное нетерминалом A, такое что первые k листьев всего дерева, начиная с самого левого листа поддерева,*

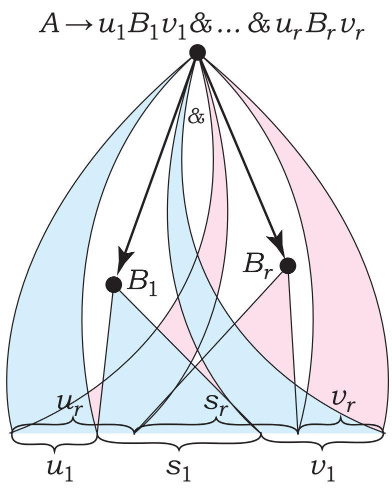

Рис. 11: Дерево разбора w из A

образуют строку x, и к корню поддерева применяется правило $A \to \alpha$, как показано на рисунке 1(справа).

Если для грамматики G существует LL(k)-таблица, то говорят, что G принадлежит классу LL(k).

Известно, что выразительные способности конъюнктивных LL-линейных грамматик сильно ограничены: как установлено Охотиным [], они не могут задать никакой язык вида $L \cdot \{a, b\}$, где L — нерегулярный. Поэтому следующий пример нетривиального языка, задаваемого такими грамматиками, представляется небезынтересным.

Пример 3. *Пусть $\Sigma = \{a, b, c, \#\}$, и пусть $h \colon \Sigma^* \to \{a, b\}^*$ — гомоморфизм, задаваемый соотношениями $h(a) = aa$, $h(\#) = ab$, $h(c) = ba$, $h(b) = bb$. Тогда следующая конъюнктивная LL(1)-линейная грамматика задаёт язык, состоящий из всех строк w_i, где $w_0 = c$, $w_{i+1} = h(w_i^R)\#w_i$.*

$$
\begin{aligned}
&S \to X\&R \mid c \\
&X \to h(s)Xs && (s \in \Sigma) \\
&X \to \# \\
&R \to aR \mid bR \mid \#S
\end{aligned}
$$

Следующий результат данной работы заключается в том, что для конъюнктивных грамматик верен аналог теоремы 1.

Теорема 3. *Для каждой конъюнктивной линейной LL(k) грамматики $G = (\Sigma, N, R, S)$ существует конъюнктивная линейная LL(1) грамматика, которая задаёт тот же язык.*

Доказательство последней теоремы почти дословно повторяет аналогичное доказательство для неконъюнктивного случая, и поэтому некоторые детали будут опускаться. Читатель без труда сможет их восстановить, обратившись к соответствующим местам доказательства теоремы 1.

Сперва из грамматики G удаляются «короткие правила», затем удаляются правила вида $A \to B$ и, наконец, грамматика приводится к виду LL(1) с помощью прикрепления к каждому нетерминалу специального буфера.

Стоит заметить, что для конъюнктивных грамматик не верен аналог факта 1, поэтому принадлежность классу LL(k) будет проверяться напрямую по определению, без использования множеств $\mathrm{First}_k(A)$ и $\mathrm{Follow}(A)$ нетерминалов $A \in N$.

8 Конъюнктивные грамматики: устранение «коротких» правил

Определение 12. ***Коротким правилом*** *будем называть правило вида $A \to w$, такое что $w \in \Sigma^*$, $|w| < k-1$ и существует дерево разбора с поддеревом A, за которым следует некая непустая строка, как показано на рисунке 2 (справа).*

Лемма 7. *Для каждой конъюнктивной линейной LL(k) грамматики $G = (\Sigma, N, R, S)$ существует конъюнктивная линейная LL(k) грамматика G' без коротких правил, задающая тот же язык.*

Доказательство. Как и в лемме 1, нетерминалы грамматики $G' = (\Sigma, N', R', S_\varepsilon)$ имеют вид A_u, где $A \in N$ и $u \in \Sigma^{\leqslant k-1}$.

Для каждого нетерминала A_u и для каждого правила $A \to \gamma_1 \,\&\, \ldots \,\&\, \gamma_r \in R$, в новой грамматике будет правило $A_u \to \alpha_1 \,\&\, \ldots \,\&\, \alpha_r$, где каждый конъюнкт α_j получается из конъюнкта γ_j следующим образом.

Пусть $\gamma_j = w_1 B w_2$. Обозначим за s первые $k-1$ символов строки $w_2 u$, а за t — оставшийся суффикс $w_2 u$, так что $s = \mathrm{First}_{k-1}(w_2 u)$ и $st = w_2 u$. Тогда положим $\alpha_j = w_1 B_s t$.

Для правил вида $A \to x$, где $x \in \Sigma^*$, соответствующие правила для нетерминалов A_u новой грамматики получаются просто приписыванием строки u в конец.

$$A_u \to xu$$

Заметим, что по каждому правилу $A_u \to \alpha_1 \,\&\, \ldots \,\&\, \alpha_k$ новой грамматики всегда можно однозначно восстановить правило $A \to \gamma_1 \,\&\, \ldots \,\&\, \gamma_k$ исходной грамматики, из которого оно получено: если $\alpha_j \in \Sigma^*$, то γ_j

получается откусыванием у α_j суффикса u, а если $\alpha_j = w_1 B_s t$, то γ_j получается откусыванием суффикса u у строки $w_1 B st$.

Доказательство корректности описанного построения естественно разбивается на несколько отдельных утверждений: а именно, что G' является конъюнктивной линейной $\mathrm{LL}(k)$ грамматикой, задаёт тот же язык, что и G, и не содержит коротких правил.

Сначала докажем, что $L_{G'}(A_u) = \{xu \mid x \in L_G(A)\}$.

Утверждение 14. *Если строка w задаётся нетерминалом A_u в новой грамматике, то $w = xu$, где x задаётся нетерминалом A в исходной грамматике.*

Доказательство. Индукция по высоте дерева разбора строки w из нетерминала A_u.

Базовый случай. Пусть A_u задаёт w по правилу $A_u \to w$.

По построению правило $A_u \to w$ получено из некоторого правила $A \to x$ исходной грамматики и имеет вид $A_u \to xu$.

Таким образом $x \in L_G(A)$.

Индукционный переход. Пусть A_u задаёт w по правилу $A_u \to \alpha_1 \,\&\, \dots \,\&\, \alpha_r$, полученному из правила $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$.

Зафиксируем $j \in \{1, \dots, r\}$. Пусть $\alpha_j = w_1 B_s t$. Тогда $\gamma_j = w_1 B w_2$, где $w_2 u = st$.

Значит $w = w_1 y t$, где $y \in L_{G'}(B_s)$, и высота дерева разбора y из нетерминала B_s меньше, чем у дерева разбора w из A_u.

По индукционному предположению, $y = zs$, где $z \in L_G(B)$.

Таким образом получаем $w = w_1 z s t = w_1 z w_2 u$. Значит строка w имеет вид xu, где $x = w_1 z w_2$ лежит в $L_G(\gamma)$. Так как j выбиралось произвольным, то x лежит в каждом из языков $L_G(\gamma_1), \dots, L_G(\gamma_r)$, и значит задаётся нетерминалом A с помощью правила $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$. В частности $w \in L_G(A)u$.

$\square$

Утверждение 15. *Если строка x задаётся нетерминалом A исходной грамматики, то в новой грамматике A_u задаёт xu.*

Доказательство. Индукция по высоте дерева разбора строки x из A.

Базовый случай. Пусть A задаёт x по правилу $A \to x$.

По построению, в грамматике G' есть правило $A_u \to xu$, значит $xu \in L_{G'}(A_u)$.

Индукционный переход. Пусть A задаёт x по правилу $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$. Зафиксируем $j \in \{1, \dots, r\}$. Пусть $\gamma_j = w_1 B w_2$. Тогда $x = w_1 y w_2$, где $y \in L_G(B)$.

Положим $s = \mathrm{First}_k(w_2 u)$.

Высота дерева разбора y из B меньше, чем у дерева разбора x, значит по предположению индукции $ys \in L_{G'}(B_s)$.

Грамматика G' содержит правило $A_u \to \alpha_1 \,\&\, \dots \,\&\, \alpha_r$, где каждый конъюнкт α_i получен из соотвутствующего конъюнкта γ_i, так что $\alpha_j = w_1 B_s t$, где $st = w_2 u$. Значит $w_1 yst \in L_{G'}(\alpha_j)$, и так как Так как $w_1 yst = w_1 y w_2 u = xu$, то строка xu лежит в языке $L_{G'}(\alpha_j)$.

Поскольку j выбиралось произвольно, то $xu \in L_{G'}(\alpha_1) \cap \dots \cap L_{G'}(\alpha_r)$, и тем самым A_u задаёт xu по правилу $A_u \to \alpha_1 \,\&\, \dots \,\&\, \alpha_r$.

□

Итак, из последних двух утверждений следует, что для каждого нетерминала $A_u \in N'$ выполняется $L_{G'}(A_u) = L_G(A)u$. В частности $L(G') = L_{G'}(S_\varepsilon) = L_G(S) = L(G)$, то есть G' задаёт тот же язык, что и G.

Аналогичное равенство имеет место и для соответствующих правил.

Утверждение 16. *Для любого правила $A_u \to \varphi$ новой грамматики, полученного из правила $A \to \varphi'$ исходной грамматики, выполняется $L_{G'}(\varphi) = L_G(\varphi')u$.*

Доказательство. Пусть $\varphi = \alpha_1 \,\&\, \dots \,\&\, \alpha_r$, и $\varphi' = \gamma_1 \,\&\, \dots \,\&\, \gamma_r$. Зафиксируем $j \in \{1, \dots, r\}$. Пусть $\alpha_j = w_1 B_s t$. Конъюнкт α_j получен из конъюнкта $\gamma_j = w_1 B w_2$ правила исходной грамматики G.

По построению, $st = w_2 u$, и по предыдущим двум утверждениям $L_{G'}(B_s) = L_G(B)s$.

Таким образом, $L_{G'}(\alpha_j) = L_{G'}(w_1 B_s t) = w_1 L_G(B) st = L_G(w_1 B w_2)u = L_G(\gamma_j)u$.

Так как такое равенство имеет место для всех j, то получаем

$$L_{G'}(\alpha_1) \cap \dots \cap L_{G'}(\alpha_r) = L_G(\gamma_1)u \cap \dots \cap L_G(\gamma_r)u$$

Что и требовалось доказать.

Если же $\varphi = xu$, то $\varphi' = x$, и утверждение тривиально. □

Теперь докажем, что в G' нет коротких правил.

Уже известно, что $L_{G'}(A_u) = L_G(A)u$, поэтому, если $|u| = k - 1$, то A_u не задаёт строк длины менее $k - 1$, и, следовательно, для таких нетерминалов коротких правил нет.

Покажем, что нет коротких правил и для нетерминалов $A_u \in N'$ с $|u| < k - 1$.

Для начала докажем ещё пару несложных вспомогательных утверждений.

Утверждение 17. *Для каждого конъюнкта* $\alpha_j = w_1 B_s t$ *правила* $A_u \to \alpha_1 \,\&\, \dots \,\&\, \alpha_r$ *в* G' *верно*

- $|s| \geqslant |u|$
- *Если* $|t| > 0$*, то* $|s| = k - 1$.

Доказательство. По построению конъюнкт $\alpha_j = w_1 B_s t$ был получен из конъюнкта $\gamma_j = w_1 B w_2$ некоторого правила $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$, причём $s = \mathrm{First}_{k-1}(w_2 u)$ и $st = w_2 u$.

Поскольку $|u| \leqslant k-1$, то $|u| = |\mathrm{First}_{k-1}(u)| \leqslant |\mathrm{First}_{k-1}(w_2 u)| = |s|$, и тем самым первая часть доказана.

Если $|t| > 0$, то, $|\mathrm{First}_{k-1}(w_2 u)| < |w_2 u|$, и значит $|s| = |\mathrm{First}_{k-1}(w_2 u)| = k - 1$. ☐

Утверждение 18. *Пусть существует дерево разбора* G'*, в котором* B_s *задаёт* xs *по правилу* $B_s \to \varphi$*, полученному из правила* $B \to \varphi'$ *исходной грамматики, и за поддеревом* B_s *следует строка* y.

Тогда существует дерево разбора G*, в котором* B *задаёт* x *по правилу* $B \to \varphi'$*, и за поддеревом* B *следует строка* sy.

Кроме того, если $|s| < k - 1$*, то* $y = \varepsilon$.

Доказательство. Применим индукцию по глубине поддерева B_s в дереве разбора.

Базовый случай. Пусть B_s является корнем всего дерева разбора, то есть $B_s = S_\varepsilon$. Так как правее всего дерева разбора листьев нет, то $y = \varepsilon$.

Так как $xs \in L_{G'}(\varphi)$, то из утверждения 16 следует, что $x \in L_G(\varphi')$, и тем самым S задаёт x по правилу φ'.

В соответствующем дереве разбора S является корнем, и поэтому за ним следует строка $\varepsilon = sy$, что и требовалось доказать.

Индукционный переход. Пусть A_u — родитель B_s в дереве разбора. К A_u применяется некоторое правило $A_u \to w_1 B_s t \,\&\, \alpha_1 \,\&\, \cdots \,\&\, \alpha_r$, полученное из правила $A \to w_1 B w_2 \,\&\, \gamma_1 \,\&\, \cdots \,\&\, \gamma_r$ исходной грамматики, где $st = w_2 u$.

Обозначим за y' строку, которая следует за поддеревом A_u, так что $y = ty'$.

Так как глубина B_s строго больше глубины A_u в дереве разбора, то к A_u применимо индукционное предположение, и тем самым существует дерево разбора с поддеревом A, такое что A задаёт $w_1 x w_2$

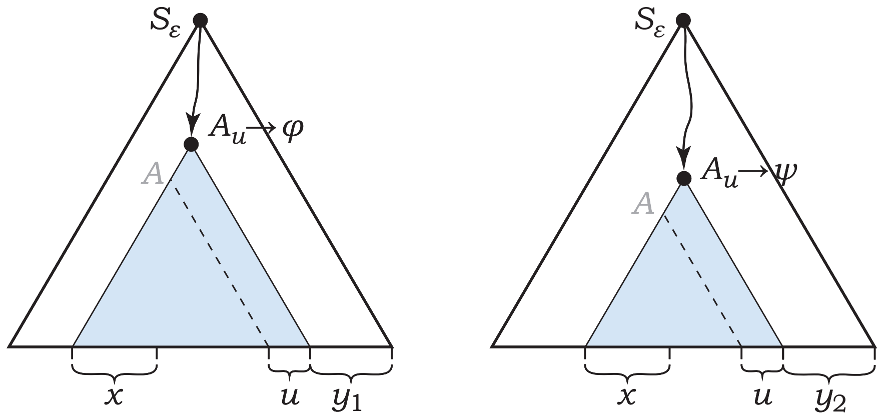

Рис. 12: Деревья D_1 и D_2 из утверждения 19

по правилу $A \to w_1 B w_2 \,\&\, \gamma_1 \,\&\, \cdots \,\&\, \gamma_r$, и за поддеревом следует строка $uy' \in \text{Follow}(A)$.

В этом дереве разбора поддерево B задаёт строку x, и за поддеревом B следует строка $w_2uy' = sty' = sy$, и первая часть утверждения 18 доказана.

Пусть теперь $|s| < k-1$. Из утверждения 17 следует, что $|s| \geqslant |u|$, причём если $|t| > 0$, то $|s| = k-1$.

Следовательно $|u| < k-1$ и $t = \varepsilon$. К A_u применимо индукционное предположение, и значит из $|u| < k-1$ следует $y' = \varepsilon$. Таким образом $y = ty' = \varepsilon$, и второе утверждение доказано.

□

По предыдущему утверждению в деревьях разбора G' за поддеревьями A_u с $|u| < k-1$ не может следовать непустых строк, и тем самым в грамматике G' отсутствуют короткие правила.

Остаётся доказать, что G' является линейной LL(k) грамматикой. Линейность G' видна из построения.

Утверждение 19. *Если G принадлежит классу LL(k), то G' — тоже.*

Доказательство. Рассмотрим два дерева разбора D_1 и D_2 грамматики G', такие что в каждом из них есть поддеревья A_u, и обоих деревьях первые k листьев, начиная с самых левых листьев поддеревьев A_u образуют одну и ту же строку x.

Пусть в дереве D_1 поддерево A_u задаёт строку pu по правилу $A_u \to \varphi$, и за поддеревом A_u следует строка y_1, а в дереве D_2 поддерево A_u задаёт строку qu по правилу $A_u \to \psi$, и за поддеревом A_u следует строка y_2, как показано на рисунке 12. Покажем, что $\varphi = \psi$.

Правила $A_u \to \varphi$ и $A_u \to \psi$ были получены из правил исходной грамматики $A \to \varphi'$ и $A \to \psi'$ соответственно. По утверждению 18 существуют деревья разбора D'_1 и D'_2, такие что в каждом из них есть поддерево A, в D'_1 поддерево A задаёт p по правилу $A \to \varphi'$, и за поддеревом A следует строка uy_1, а в D'_2 поддерево A задаёт q по правилу $A \to \psi'$, и за поддеревом A следует строка uy_2.

Тогда первые k листьев этих деревьев разбора, начиная с самых левых листьев поддеревьев A в деревьях D'_1 и D'_2 образуют строки $\mathrm{First}_k(puy_1)$ и $\mathrm{First}_k(quy_2)$ соответственно.

Так как $\mathrm{First}_k(puy_1) = \mathrm{First}_k(quy_2) = x$, и грамматика G принадлежит классу $\mathrm{LL}(k)$, то $\varphi' = \psi'$, а значит и $\varphi = \psi$.

Тогда можно положить $T_{G'}(A_u, x) = A_u \to \varphi$, и построенная таким образом частичная функция будет корректной $\mathrm{LL}(k)$ таблицей для грамматики G'. □

Таким образом грамматика G' принадлежит классу $\mathrm{LL}(k)$, что завершает доказательство корректности построения. □

9 Конъюнктивные грамматики: приведение к виду LL(1)

После того, как из грамматики удалены все короткие правила, её можно преобразовать к виду $\mathrm{LL}(1)$ так же, как это делалось в неконъюнктивном случае.

Сперва снова избавимся от так называемых цепных правил.

Определение 13. *Правило $A \to \alpha_1 \& \ldots \& \alpha_r$ конъюнктивной грамматики $G = (\Sigma, N, R, S)$ называется* ***цепным****, если каждый конъюнкт α_j, $j \in \{1, \ldots, r\}$ состоит из единственного нетерминала, то есть $\alpha_j = B$, $B \in N$.*

Лемма 8. *Для каждой конъюнктивных линейной $LL(k)$ грамматики $G = (\Sigma, N, R, S)$ без коротких правил существует конъюнктивная линейная $LL(k)$ грамматика $G' = (\Sigma, N, R', S)$ без коротких правил, задающая тот же язык и не содержащая цепных правил.*

Доказательство. Доказательство непосредственно обобщает аналогичное доказательство леммы 2. □

Докажем теперь аналог утверждения 8.

Утверждение 20. *Пусть $G = (\Sigma, N, R, S)$ — конъюнктивная линейная $LL(k)$ грамматика без коротких и цепных правил. Тогда для любого нетерминала $A \in N$ если $x \in L(A)$ и $|x| = k - 1$, то либо в G не существует деревьев разбора, в которых есть поддерево A, задающее* *строку x, и за поддеревом A следует непустая строка, либо в R есть правило $A \to x$.*

Доказательство. Предположим противное. Пусть существует дерево разбора с поддеревом A, такое что A задаёт x по некоторому правилу $A \to \varphi$, $|x| = k - 1$, и за поддеревом A следует некоторая непустая строка. Обозначим за c первый символ этой строки.

Тогда $T(A, xc) = A \to \varphi$. Если $\varphi \in \Sigma^*$, то $\alpha = x$ и всё доказано.

Предположим, $\varphi = \alpha_1 \,\&\, \dots \,\&\, \alpha_r$. Так как в G отсутствуют цепные правила, то для некоторого конъюнкта $\alpha_j = sBt$ выполняется $|s|+|t| > 0$. Тогда $x = syt$, где $y \in L(B)$, и значит $|y| < |x| = k - 1$.

Рассмотрим дерево разбора y из B. Любое самое нижнее правило в нём имеет вид $C \to z$, где $|z| \leqslant |y| < k - 1$.

Так как вершина C — потомок A в дереве разбора, и за A следует непустая строка, то за поддеревом C также следует непустая строка.

Тем самым правило $C \to z$ короткое, что противоречит отсутствию в G коротких правил. $\square$

Опишем теперь основное построение.

Лемма 9. *Для каждой конъюнктивной линейной LL(k) грамматики $G = (\Sigma, N, R, S)$ без коротких и цепных правил существует конъюнктивная линейная LL(1) грамматика G', задающая тот же язык.*

Доказательство. Как и в лемме 3, нетерминалы новой грамматики $G' = (\Sigma, N', R', {}_\varepsilon S)$, имеют вид ${}_uA$, где $A \in N$ и $u \in \Sigma^{\leqslant k-1}$.

Аналогично неконъюнктивному случаю можно определить синтаксический анализатор, который строит дерево разбора строки по мере прочтения её слева направо, определяя нужные правила по очередным k символам строки с помощью LL(k) таблицы.

Левый нижний индекс u нетерминала ${}_uA$ будет выполнять роль буфера, в котором хранится до $k - 1$ последних символов, прочтённых синтаксическим анализатором.

Начальным символом G' является ${}_\varepsilon S$, что соответствует S с пустым буфером.

Итак, $N' = \{{}_uA \mid A \in N\, u \in \Sigma^{\leqslant k-1}\}$. Правила грамматики G' представляются объединением трёх множеств R_1,R_2 и R_3.

Множество правил R_1 реализуют заполнение буфера. Для каждого нетерминала ${}_uA$ с $|u| < k - 1$, и для каждого символа $a \in \Sigma$, в G' есть правило, заносящее этот символ в буфер.

$$ {}_uA \to a\, {}_{ua}A $$

Правила R_2 используются, когда буфер уже заполнен, и синтаксический анализатор G' может понять, какое правило исходной грамматики следует применить. Для каждого ${}_uA \in N'$ и $a \in \Sigma$, где $|u| = k-1$ и значение $T(A, ua)$ определено, в G' есть правило, получающееся *откусыванием* строки u из правила $(A \to \varphi) = T(A, ua)$. Если $\varphi \in \Sigma^*$, тогда $\varphi = us$, где $s \in \Sigma^*$, и соответствующее правило в G' имеет вид

$$ {}_uA \to s $$

Если φ имеет вид $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$, то соответствующее правило G' имеет вид ${}_uA \to \alpha_1 \,\&\, \dots \,\&\, \alpha_r$, где каждый конъюнкт α_j получается откусыванием u из начала конъюнкта γ_j, как это было в лемме 3. Если $\gamma_j = sBt$, где $s, t \in \Sigma^*$, $B \in N$, то одна из строк u и s является префиксом другой, и имеем два случая:

$$\begin{aligned} \alpha_j &= s'\,{}_\varepsilon Bt, && \text{если } s = us',\ s' \in \Sigma^* \\ \alpha_j &= {}_vBt, && \text{если } u = sv,\ v \in \Sigma^+ \end{aligned}$$

Наконец, правила R_3 нужны для случая, когда вся строка уже прочитана анализатором, и ему нечего заносить в буфер. А именно, для каждого ${}_uA \in N'$, где $|u| \leqslant k-1$ и значение $T(A, u)$ определено, грамматика G' содержит пустое правило.

$$ {}_uA \to \varepsilon $$

Заметим, что $R_1 \cap (R_2 \cup R_3) = \varnothing$, однако множества R_2 и R_3 могут пересекаться по некоторым правилам вида ${}_uA \to \varepsilon$, $|u| = k-1$. Если известно, что правило ${}_uA \to \varphi$ лежит в R_2, то всегда можно однозначно восстановить правило $A \to \varphi'$ исходной грамматики, из которого оно получено, исходя из определения. Если $\varphi \in \Sigma^*$, то $\varphi' = u\alpha$, если $\varphi = \alpha \,\&\, \dots \,\&\, \alpha_r$, то $\varphi' = \gamma_1 \,\&\, \dots \,\&\, \gamma_r$, где каждый конъюнкт γ_j получается из конъюнкта α_j следующим образом. Если $\alpha_j = s'\,{}_\varepsilon Bt$, то $\gamma_j = us'Bt$, и если $\alpha_j = {}_vBt$, то имеем $u = sv$, и $\gamma_j = sBt$.

Доказательство того, что G' обладает свойством $LL(1)$ и задаёт тот же язык, что и G, даётся в серии утверждений.

Сначала установим связь между языками $L_G(A)$ и $L_{G'}({}_uA)$. В конъюнктивном случае равенство $L_{G'}({}_uA) = \{\, w \mid uw \in L_G(A) \,\}$, вообще говоря, не выполняется. Причина этого кроется в том, что, в отличие от неконъюнктивного случая, может оказаться, что некоторая строка w задаётся нетерминалом A, однако дерева разбора с корнем S и поддеревом A, задающим w не существует.

Следующие два утверждения показывают, как соотносятся между собой множества $L_{G'}({}_uA)$ и $L_G(A)$.

Сначала определим соответствие между деревьями разбора G' и деревьями разбора G.

Определение 14. *Пусть D и D' — деревья разбора G и G' соответственно. Корни D и D' не обязательно помечены начальными символами грамматик. Пусть V и V' — множества вершин D и D' соответственно.*

*Отображение $f : V' \to V$ назовём назовём **гомоморфизмом** деревьев D и D', если оно обладает следующим свойством. Пусть вершина $v' \in V'$ помечена нетерминалом ${}_uA \in N'$, и её поддерево задаёт строку x. Тогда:*

1. *Вершина $v = f(v')$ помечена нетерминалом $A \in N$ и поддерево v задаёт строку ux.*

2. *Пусть путь от вершины $v' \in V'$, помеченной нетерминалом ${}_uA \in N'$, к некоторому листу начинается с нескольких (возможно, нуля) правил, наращивающих буфер, а затем — правила ${}_{u'}A \to \varphi$, полученного из правила $A \to \varphi'$ исходной грамматики (соответствующие правила выписаны ниже).*

$$\begin{aligned} {}_uA &\to a_1 \, {}_{ua_1}A \\ {}_{ua_1}A &\to a_2 \, {}_{ua_1a_2}A \\ &\vdots \\ {}_{ua_1\dots a_{m-1}}A &\to a_m \, {}_{u'}A \\ {}_{u'}A &\to \varphi \end{aligned}$$

   Тогда к вершине $f(v)$ применяется правило $A \to \varphi'$.

*Если такое отображение f существует, то деревья D и D' будем называть **гомоморфными**.*

Утверждение 21. *Если существует дерево разбора D' строки x из нетерминала ${}_uA \in N'$, то существует гомоморфное ему дерево разбора D строки ux из нетерминала $A \in N$.*

Доказательство. Доказательство проводится индукцией по высоте дерева разбора x из ${}_uA$.

Базовый случай: x задаётся одним правилом ${}_uA \to x$. Все правила R_1 содержат нетерминал, поэтому ${}_uA \to x$ может быть либо из R_2, либо из R_3.

В первом случае ${}_uA \to x$, получено из правила $A \to ux$ в G, и несложно проверить, что соответствующие поддеревья высоты 1 гомоморфны.

Во втором случае $T(A, u)$ определено и $x = \varepsilon$. Так как $T(A, u)$ определено и $u < k$, то u — суффикс входной строки.

Тогда по определению LL-таблицы существует дерево разбора с поддеревом A, такое что A задаёт некоторую строку x', и за поддеревом следует строка t, так что $x't = u$.

Если $|x'| = k-1$, то, поскольку $|u| \leqslant k-1$, то $x' = u$. Если же $|x'| < k-1$, то из отсутствия в G коротких правил, следует, что $t = \varepsilon$ и тем самым снова $x' = x't = u$.

Так или иначе, получаем, что $ux = u = x'$ задаётся A, и отображение, переводящее вершину ${}_uA$ в корень поддерева A, задающего ux, есть требуемый гомоморфизм деревьев. Заметим, что в этом случае второе условие из определения гомоморфизма выполняется тривиальным образом, так как в дереве ${}_uA$ просто нет правил из R_2.

Индукционный переход. Пусть ${}_uA$ задаёт x по правилу ${}_uA \to \varphi$, и φ содержит нетерминал.

Правило ${}_uA \to \varphi$ лежит либо в R_1, либо в R_2, соответственно нужно разобрать два случая.

Пусть ${}_uA \to \varphi$ лежит в R_2, то есть ${}_uA \to \alpha_1 \,\&\, \dots \,\&\, \alpha_r$, где каждый конъюнкт α_j получен откусыванием u из начала конъюнкта γ_j соответствующего правила $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$ грамматики G.

Зафиксируем $j \in \{1, \dots, r\}$. Пусть $\gamma_j = sBt$. В соответствии с определением R_2, есть два случая, в зависимости от того, какая из строк u и s длиннее.

- Если $|u| > |s|$, то $\alpha_j = {}_vBt$, где $u = sv$. Тогда $x = yt$ для некоторого $y \in L_{G'}({}_vB)$. Высота дерева разбора y из ${}_vB$ меньше высоты дерева разбора x из ${}_uA$.

  Тогда, по предположению индукции, существует дерево разбора vy из B, гомоморфное поддереву ${}_vB$, и в частности $ux = uyt = svyt \in L_G(sBt) = L_G(\gamma_j)$.
- Если $|u| \leqslant |s|$, то $\alpha_j = s'{}_\varepsilon Bt$ где $us' = s$. Тогда $x = s'yt$, где $y \in L_{G'}({}_\varepsilon B)$.

  По предположению индукции, существует дерево разбора y из B, гомоморфное поддереву ${}_\varepsilon B$, и в частности $ux = us'yt = syt \in L_G(sBt) \in L_G(\gamma_j)$.

Пусть D_j обозначает дерево разбора с корнем B, гомоморфное поддереву D'_j с корнем ${}_vB$ или ${}_\varepsilon B$, в зависимости от того, какой из случаев имеет место.

Рассмотрим дерево разбора D с корнем A, такое что к A применяется правило $A \to \gamma_1 \,\&\, \dots \,\&\, \gamma_r$, и нетерминалу из каждого конъюнкта γ_j соответствует поддерево D_j.

Тогда все конъюнкты γ_j задают одну и ту же строку ux, и тем самым D является корректным деревом разбора ux из A.

Гомоморфизм между деревьями D' и D получается следующим образом: корень D' отображается в корень D, а все остальные вершины — согласно гомоморфизмам между D'_j и D_j.

Пусть теперь $({}_uA \to \varphi) \in R_1$ — правило, наращивающее буфер.

Тогда $\varphi = a_{ua}A$ для некоторого $a \in \Sigma$. Значит $x = ay$, где $y \in L_{G'}({}_{ua}A)$.

Следовательно, по предположению индукции, существует гомоморфизм между поддеревом ${}_{ua}A$ и некоторым деревом разбора $uay = ux$ из A. Этот гомоморфизм можно продолжить на корень ${}_uA$, отобразив его в корень дерева разбора ux из A.

$\square$

Из последнего утверждения сразу вытекает полезное следствие:

Утверждение 22. *Пусть существует дерево разбора G', в котором ${}_vB$ задаёт x по правилу ${}_vB \to \varphi$, и за поддеревом ${}_vB$ следует строка y.*

1. *Тогда существует дерево разбора G, в котором B задаёт vx и за поддеревом B следует строка y.*

2. *Если, к тому же, правило ${}_vB \to \varphi$ получено из правила исходной грамматики $B \to \varphi'$, то существует дерево разбора G, в котором A задаёт ux по правилу $A \to \varphi'$ и за поддеревом A следует строка y.*

Утверждение 23. *Если существует дерево разбора G с поддеревом A, задающим строку ux, то $x \in L_{G'}({}_uA)$.*

Доказательство. Сначала разберём случай, когда $|ux| < k$ и $T(A, ux)$ определено. Тогда, по построению G' содержит правило ${}_{ux}A \to \varepsilon$.

Пусть $x = x_1 \dots x_m$. Буфер u нетерминала ${}_uA$ может быть расширен до ux с помощью следующих правил.

$$
\begin{aligned}
{}_uA &\to x_1\ {}_{ux_1}A \\
{}_{ux_1}A &\to x_2\ {}_{ux_1x_2}A \\
{}_{ux_1\dots x_{m-1}}A &\to x_m\ {}_{ux}A
\end{aligned}
$$

Эта последовательность правил, вместе с правилом ${}_{ux}A \to \varepsilon$ составляют вывод x из ${}_uA$. Таким образом, $x \in L_{G'}({}_uA)$.

Теперь предположим, что либо $|ux| \geqslant k$, либо $|ux| = k-1$, но значение $T(A, ux)$ не определено.

Если $|ux| = k-1$, то из того, что $T(A, ux)$ не определено, следует, что за поддеревом A, задающим ux, следует некоторая непустая строка. Определим c как первый символ этой строки.

Положим $n = k - |u| - 1$ и пусть a будет либо x_{n+1}, в случае $|ux| \geqslant k$, либо c, в случае $|ux| = k-1$.

Обозначим за u' строку $ux_1 \dots x_n$.

Тогда $|u'| = k-1$ и $T(A, u'a) = A \to \varphi'$, где $A \to \varphi'$ — правило, по которому A задаёт ux.

Буфер ${}_uA$ может быть пополнен до u' с помощью следующих правил.

$$
\begin{aligned}
{}_uA &\to x_1\ {}_{ux_1}A \\
{}_{ux_1}A &\to x_2\ {}_{ux_1x_2}A \\
{}_{ux_1\dots x_{n-1}}A &\to x_n\ {}_{u'}A
\end{aligned}
$$

Тогда, так как $T(A, u'a) = A \to \varphi'$, грамматика G' содержит правило ${}_{u'}A \to \varphi$, где каждый конъюнкт φ получается откусыванием u из соответствующего конъюнкта φ'.

Остаётся доказать, что φ' задаёт строку $x_{n+1} \dots x_m$. Воспользуемся индукцией по высоте дерева разбора ux из A.

Заметим, что если φ задаёт строку $x_{n+1} \dots x_m$, то последовательность правил выше, вместе с правилом ${}_{u'}A \to \varphi$, составляет вывод строки $x_1 \dots x_m = x$ из ${}_uA$, и доказательство будет окончено. Поэтому можно считать, что предположением индукции служит всё доказываемое утверждение.

Базовый случай: ux задаётся по правилу $A \to ux$. Тогда $\varphi' = T(A, u'a) = A \to ux$, и $\varphi = x_{n+1} \dots x_m$.

Индукционный переход: ux задаётся по правилу $A \to \gamma_1 \& \dots \& \gamma_r$. Тогда $\varphi = \alpha_1 \& \dots \& \alpha_r$, где каждый конъюнкт α_j получается откусыванием u' из конъюнкта γ_j.

Зафиксируем $j \in \{1, \dots, r\}$. Пусть $\gamma_j = sBt$.

Тогда $ux = syt$, где $y \in L_G(B)$. и α_j получается откусыванием u' из строки sBt.

В соответствии с определением R_2 имеем два случая, в зависимости от того, какая из строк u' и s длиннее.

Если $|s| \geqslant |u'|$, то $\alpha_j = s'{}_\varepsilon Bt$, где $u's' = s$.

Высота дерева разбора y из B меньше высоты дерева разбора ux из A. Тогда по предположению индукции $y \in L_{G'}({}_\varepsilon B)$.

Следовательно, $u's'yt = syt = ux$, и значит $s'yt = x_{n+1} \dots x_m \in L_{G'}(\alpha_j)$.

Если же $|s| < |u'|$, то $\alpha_j = {}_vBt$, где $sv = u'$.

Чтобы воспользоваться предположением индукции для нетерминалов B и ${}_vB$ и строки y, необходимо показать, что v является префиксом y.

Так как в G нет коротких правил, то либо $|y| \geqslant k-1$, либо $t = \varepsilon$.

Если $t = \varepsilon$, то $ux = sy$, и поскольку $sv = u'$ является префиксом ux, то v является префиксом y.

Если же $|y| = k-1$, то и $|sy| \geqslant k-1$. Поскольку u' — это префикс $ux = syt$, и $|u'| \leqslant k-1$, то строка $u' = sv$ является префиксом sy, и следовательно v — префикс y.

Таким образом, $y = vy'$ для некоторой строки $y' \in \Sigma^*$, и по предположению индукции $y' \in L_{G'}({}_vB)$.

Получаем $u'y't = u'syt = ux$, и значит $y't = x_{n+1} \dots x_m \in L_{G'}(\alpha_j)$.

Следовательно $x_{n+1} \dots x_m \in L_{G'}(\alpha_j)$ для каждого $j \in \{1, \dots, r\}$. Значит ${}_{u'}A$ задаёт $x_{n+1} \dots x_m$ по правилу ${}_{u'}A \to \alpha_1 \mathbin{\&} \dots \mathbin{\&} \alpha_r = \varphi$, что и требовалось доказать.

□

Грамматика G' линейна по построению и по утверждениям 21 и 23 $L(G') = L_{G'}({}_\varepsilon S) = L_G(S) = L(G)$. Остаётся показать, что G' принадлежит классу $LL(1)$.

Утверждение 24. *Грамматика G' принадлежит классу LL(1).*

Доказательство. Рассмотрим два дерева разбора D_1 и D_2 грамматики G', в каждом из которых есть поддерево ${}_uA$, и пусть строки, начинающиеся с самых левых листьев этих поддеревьев, начинаются на одну и ту же букву или обе пусты. Обозначим эту букву за x (если обе строки пусты, то $x = \varepsilon$).

Пусть в D_1 поддерево ${}_uA$ задаёт строку p по правилу ${}_uA \to \varphi$, и за поддеревом ${}_uA$ следует строка y_1, а в D_2 поддерево ${}_uA$ задаёт строку q по правилу ${}_uA \to \psi$, и за поддеревом ${}_uA$ следует строка y_2. Покажем, что $\varphi = \psi$.

Доказательство проводится отдельно для случаев $|u| < k-1$ и $|u| = k-1$. Пусть сначала $|u| < k-1$.

Тогда каждое из правил ${}_uA \to \varphi$ и ${}_uA \to \psi$ лежит в R_1 или R_3. Разберём случаи.

- Если оба правила лежат в R_3, то $\varphi = \psi = \varepsilon$.

- Если оба правила лежат в R_1, то $\varphi = a\, {}_{ua}A$ и $\psi = b\, {}_{ub}A$ для некоторых символов a и b. Тогда $x = a = b$, и значит $\varphi = \psi$.

- Пусть одно из правил лежит в R_1, а другое — в R_3. Не умаляя общности, $\varphi = a\, {}_{ua}A$ для некоторого $a \in \Sigma$, а $\psi = \varepsilon$.

  Тогда $\varepsilon \in L_{G'}({}_uA)$, и $u \in L_G(A)$ по утверждению 21. Поскольку в G нет коротких правил, и $|u| < k - 1$, то $y_2 = \varepsilon$, и так как $x \in \mathrm{First}_k(L_{G'}(\psi)y_2)$, то $x = \varepsilon$. В то же время из того, что $\varphi = a\, {}_{ua}A$ следует, что $x = a$. Полученное противоречие показывает, что этот случай невозможен.

Пусть теперь $|u| = k - 1$. Тогда, каждое из правил ${}_uA \to \varphi$, ${}_uA \to \psi$ лежит в R_2 или в R_3. Снова разберём случаи.

- Если оба правила лежат в R_3, то $\varphi = \psi = \varepsilon$.

- Пусть, скажем, ${}_uA \to \varphi$, лежит в R_2 и получено из правила $A \to \varphi'$,а ${}_uA \to \psi$ лежит в R_3. Тогда $\psi = \varepsilon$, и значит, во-первых, $u \in L_G(A)$, и во-вторых, $x = \mathrm{First}_1(y_2)$.

  Предположим, φ задаёт пустую строку. Тогда φ' задаёт u по утверждению 10. По определению R_2 имеем $(A \to \varphi') = T(A, ua)$ для некоторого $a \in \Sigma$, и поэтому существует дерево разбора с поддеревом A, такое что A задаёт u по правилу $A \to \varphi'$, и строка, которая следует за поддеревом A, начинается с символа a.

  Тогда $\alpha = u$ по утверждению 20, и так как φ получается откусыванием u из φ', то $\varphi = \varepsilon = \psi$.

  Пусть теперь φ не задаёт пустой строки, и тем самым $x = \mathrm{First}_1(L_{G'}(\varphi))$.

  То, что $\varepsilon \notin \mathrm{First}_1(L_{G'}(\varphi))$ уже предполагается. Пусть $c \in \mathrm{First}_1(L_{G'}(\gamma_1))$, где $c \in \Sigma$. Тогда $T(A, uc) = (A \to \varphi')$.

  Из утверждения 22 следует, существует дерево разбора с поддеревом A, такое что A задаёт u, и за поддеревом A следует строка y_2.

  Так как $|u| = k - 1$ и $y_2 \neq \varepsilon$, то из утверждения 8 следует, что A задаёт u по правилу $A \to u$.

  Предположим, что $\mathrm{First}_1(y_2) = c$. Тогда по определению LL(k)-таблицы имеем $T(A, uc) = (A \to u)$. Поэтому $\varphi' = u$, и снова $\varphi = \varepsilon = \psi$.

- Пусть теперь правила $A \to \varphi$ и $A \to \psi$ были получены из правил исходной грамматики $A \to \varphi'$ и $A \to \psi'$ соответственно. По утверждению 22 существуют деревья разбора D_1' и D_2', такие что в обоих деревьях есть поддерево A, в D_1 поддерево A задаёт up по правилу

$A \to \varphi'$, и за поддеревом A следует строка y_1, а в D_2 поддерево A задаёт uq по правилу $A \to \psi'$, и за поддеревом A следует строка y_2.

Тогда первые k листьев этих деревьев разбора, начиная с самых левых листьев поддеревьев A и A образуют строки $\mathrm{First}_k(py_1)$ и $\mathrm{First}_k(qy_2)$ соответственно.

Так как $\mathrm{First}_k(py_1) = \mathrm{First}_k(qy_2)$, и грамматика G принадлежит классу LL(k), то $\varphi' = \psi'$, а значит и $\varphi = \psi$.

Тогда можно положить $T_{G'}({}_uA, x) = {}_uA \to \varphi$, и построенная таким образом частичная функция будет корректной LL(k) таблицей для грамматики G'. $\square$

Последнее утверждение завершает доказательство леммы 9. $\square$

Вместе леммы 7, 8 и 9 составляют доказательство теоремы 3.

10 Заключение

Полученные результаты заставляют задуматься о нескольких близких вопросах. Неизбежен ли рост размера грамматики пи преобразовании LL(k)-линейных конънктивных грамматик в LL(1)-линейные конъюнктивные? Существует ли иерархия LL(k)-конъюнктивных грамматик (без свойства линейности) по k? Вообще, о классе LL(k)-конъюнктивных грамматик известно очень мало, и он ждёт своего исследователя.

Список литературы

[1993] M. Holzer, K.-J. Lange, "On the complexities of linear LL(1) and LR(1) grammars", *Fundamentals of Computation Theory* (FCT 1993, Hungary, August 23–27, 1993), LNCS 710, 299–308.

[1988] O. H. Ibarra, T. Jiang, B. Ravikumar, "Some subclasses of context-free languages in NC^1", *Information Processing Letters*, 29:3 (1988), 111–117.

[1971] D. E. Knuth, "Top-down syntax analysis", *Acta Informatica*, 1 (1971), 79–110.

[1969] R. Kurki-Suonio, "Notes on top-down languages", *BIT Numerical Mathematics*, 9:3 (1969), 225–238.

[1968] P. M. Lewis II, R. E. Stearns, "Syntax-directed transduction", *Journal of the ACM*, 15:3 (1968), 465–488.

[2001] A. Okhotin, “Conjunctive grammars”, *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.

[2011] A. Okhotin, “Expressive power of LL(k) Boolean grammars”, *Theoretical Computer Science*, 412:39 (2011), 5132–5155.

[2019] A. Okhotin, I. Olkhovsky, “LL(1) linear grammars are as powerful as LL(k) linear grammars”, *Fifth Russian-Finnish Symposium on Discrete Mathematics* (RuFiDiM V, Veliky Novgorod, 19–22 May 2019).

[2020] A. Okhotin, I. Olkhovsky, “On the transformation of LL(k)-linear grammars to LL(1)-linear”, *Computer Science in Russia* (CSR 2020, Ekaterinburg, Russia, 29 June–3 July 2020), LNCS 12159, to appear.

[1970] D. J. Rosenkrantz, R. E. Stearns, “Properties of deterministic top-down grammars”, *Information and Control*, 17 (1970), 226–256.