Лекция 8. Контекстно-свободные грамматики и языки. Часть 2

Главные вопросы, которые мы обсуждаем, представлены на СЛАЙДЕ 1. Сначала мы определим ограничения на структуру продукций КСГ и докажем, что любой КСЯ имеет грамматику специального вида. Затем мы ознакомимся с леммой о разрастании КСЯ, и далее рассмотрим ряд свойств КСЯ, а именно свойства замкнутости.

8.1 Нормальные формы контекстно-свободных грамматик

Давайте покажем, что любой КСЯ без ε порождается грамматикой, все продукции которой имеют вид $A \to BC$ и $A \to d$, где A, B, C – нетерминалы, а d – терминал. Иными словами, если в RHS любой продукции два нетерминала, либо один терминальный символ, то такая форма записи КСГ называется нормальной формой Хомского (НФХ, *Chomsky Normal Form*, *CNF*). СЛАЙД 2.

Пример 52.

Грамматика с продукциями $S \to AS \mid a$, $A \to SA \mid b$ находится в НФХ, а грамматика $S \to AS \mid AAS$, $A \to SA \mid aa$ – нет.

Для того чтобы получить НФХ, необходимо предварительно выполнить несколько преобразований, которые имеют самостоятельное значение:

- Нужно удалить все **бесполезные** символы.
- Следует удалить все **ε-продукции** ($A \to \varepsilon$).
- Следует удалить все **цепные продукции** ($A \to B$ с нетерминалами A и B).

Удаление бесполезных символов

Символ X называется **полезным**, если существует некоторое порождение вида $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$ из V_T^*. Символ X может быть как нетерминальным, так и терминальным, а выводимая строка $\alpha X \beta$ может быть первой или последней в порождении. Символ X называется **бесполезным**, если он не является полезным. Исключение бесполезных символов из КСГ не изменяет порождаемого ею языка, поэтому все бесполезные символы можно обнаружить и удалить. СЛАЙД 3.

Обратим внимание, что полезным символам присущи следующие свойства.

1. Символ X является **порождающим**, если $X \Rightarrow^* w$ для некоторой терминальной строки w. Каждый терминал является порождающим, т.к. w может быть этим терминалом, порожденным за 0 шагов.

2. Символ X является **достижимым**, если существует порождение $S \Rightarrow^* \alpha X \beta$ для некоторых α и β.

Очевидно, что полезный символ будет одновременно порождающим и достижимым. Тогда, если из КСГ удалить непорождающие символы, а затем недостижимые, то останутся только полезные.

Пример 53.

В грамматике с продукциями $S \to AB \mid a$, $A \to b$ все символы, кроме B являются порождающими. Если удалить B, то придется удалить и первую продукцию в грамматике. Это приводит к продукциям $S \to a$, $A \to b$. Из S достижимы только a и S. Удаление символов a и b приведет к единственной продукции $S \to a$. С помощью этой КСГ можно

получить язык с единственной строкой *a*, также как с использованием исходной грамматики.

Если мы изменим порядок проверки, то окажется, что все символы исходной грамматики являются достижимыми. Удаляя *B*, как непорождающий символ, мы останемся с грамматикой с двумя бесполезными символами *A* и *b*.

Продукция называется **бесполезной**, если в ней есть хотя бы один бесполезный символ.

Процедура удаления бесполезных символов и продукций основывается на распознавании двух рассмотренных выше ситуаций.

Теорема 8.1. Пусть G – КСГ и $L(G)$ – непустой язык, т.е. эта грамматика порождает хотя бы одну непустую строку. Тогда существует эквивалентная ей КСГ G_1, не содержащая бесполезных символов или продукций. СЛАЙД 5.

Доказательство. Грамматику G_1 можно получить из G путем выполнения алгоритма, состоящего из двух основных частей.

В первой части конструируется промежуточная грамматика $G_2 = (V_{T,2}, V_{N,2}, P_2, S)$, такая что $V_{N,2}$ содержит только такие нетерминалы A, для которых возможно $A =>^* w$ из T^*. Шаги алгоритма:

1. Установить $V_{N,2} = \varnothing$.
2. Пока не останется символов для попадания в $V_{N,2}$, повторять добавление такого нетерминала A из V_N, у которых продукции имеют следующую форму:

$A \to x_1 \ldots x_n$, все x_i из $V_{N,2} \cup V_T$.

3. Принять P_2 как все продукции из P, чьи символы все в $(V_{N,2} \cup V_T)$.

Очевидно, что эта процедура конечна. Также очевидно, что если A принадлежит $V_{N,2}$, то он является порождающим символом для G_2. Остался вопрос о том, а все ли такие символы, которые $A =>^* w = ab\ldots$, попадают в $V_{N,2}$ перед тем, как эта часть алгоритма прекращается. Чтобы это увидеть, представим любой такой A и посмотрим на фрагмент дерева разбора для данного порождения (СЛАЙД 6, слева). На уровне k есть только терминальные символы, так что все нетерминалы A_i на уровне k-1 будут добавлены к $V_{N,2}$ на первой итерации второго шага алгоритма. Аналогично, нетерминалы уровня k-2 добавляются на второй итерации и так далее. Алгоритм не прерывается, пока в дереве есть нетерминалы, которых еще нет в $V_{N,2}$. Следовательно, A рано или поздно появится в этом множестве, в котором остаются только достижимые символы и нет недостижимых.

Во второй части мы должны из G_2 получить грамматику G_1. Можно построить так называемый граф зависимостей нетерминалов грамматики G_2.

Граф зависимостей (далее – ГЗ) – это способ визуализации отношений, которым можно найти множество применений. Так, для КСГ ГЗ содержит узлы C и D, помеченные нетерминалами, которые соединяются дугой, если и только если существует продукция вида $C \to xDy$. Пример ГЗ приведен на СЛАЙДЕ 8.

Используя ГЗ грамматики G_2, следует найти все недостижимые из аксиомы символы и удалить их множества нетерминалов. Также нужно удалить все продукции, в которые вовлечены такие символы. Терминальные символы, которые не встречаются ни в одной из полезных продукций, исключаются из соответствующего множества. Результатом будет грамматика G_1, в которой не будет бесполезных символов и продукций.

Для всех w из $L(G)$ мы имеем порождение $S =>^* xAy =>^* w$. Поскольку построение G_1 сохраняет нетерминал A и все связанные с ним продукции, то у нас есть все, чтобы с помощью G_1 получить то же самое порождение. Из произвольности нетерминала A и строки w следует $L(G) \subseteq L(G_1)$

Грамматика G_1 получена из G путем удаления продукций, так что $P_1 \subseteq P$. Значит, $L(G_1) \subseteq L(G)$. Объединив оба результата, получаем доказательство эквивалентности

исходной и результирующей грамматик. СЛАЙД 9.

Удаление ε-продукций

Далее покажем, что такие продукции, хотя и удобны в задачах построения грамматик, но существенными не являются. При этом мы должны осознавать, что без продукции с RHS, равной ε, невозможно породить пустую строку как элемент языка. Иначе говоря, мы на самом деле доказываем, что если язык L задается КСГ, то L-$\{\varepsilon\}$ имеет КСГ без ε-продукций. Такой язык называется ***ε*-свободным**.

Продукция вида $A \to \varepsilon$ называется ***ε*-продукцией**. Нетерминал A, для которого возможно порождение $A \Rightarrow^* \varepsilon$, называется **допускающим пустоту**, или ***ε*-порождающим** (*nullable*). СЛАЙД 10.

Пример 54.

Грамматика с продукциями $S \to aS_1b$, $S_1 \to aS_1b \mid \varepsilon$ генерирует ε-свободный язык $\{a^nb^n : n \geq 1\}$. ε-продукцию из КСГ можно удалить после добавления правил, полученных заменой ε на S_1 там, где он появляется справа. Выполнив это, мы получим следующие продукции.

$S \to aS_1b \mid ab$, $S_1 \to aS_1b \mid ab$.

Не сложно доказать, что эта грамматика, эквивалентна исходной с учетом ε-свободы языка.

Теорема 8.2. Пусть G – КСГ и $L(G)$ не содержит ε. Тогда существует эквивалентная ей КСГ G_1, не содержащая ε-продукций.

Доказательство. Сначала нужно найти множество NN всех ε-порождающих нетерминалов в G. Для этого выполняются два следующих шага.

1. Для всех продукций $A \to \varepsilon$ добавляем A во множество NN.
2. Пока не останется нетерминалов для добавления в NN повторять для всех продукций вида $B \to A_1 \ldots A_N$, где $A_1, \ldots, A_N$ есть в NN, положить B в это множество.

Когда NN найдено, мы готовы к конструированию P_1. Чтобы это сделать, мы взглянем на такие продукции в P, которые имеют вид $A \to x_1 \ldots x_m$, $m > 0$, для каждого x_i из $(V_N \cup V_T)$. Каждую такую продукцию мы размещаем в P_1 также как все продукции, сгенерированные заменой ε-порождающих нетерминалов с ε во всех возможных комбинациях. Например, если x_i и x_j – ε-порождающие нетерминалы, то в P_1 будет одна продукция с x_i, замененным на ε, одна – с x_j, замененным на ε, и еще одна где оба символа заменены на ε.

Есть одно **исключение из этого правила**: если все x_i – ε-порождающие нетерминалы, то продукция $A \to \varepsilon$ не добавляется в множество P_1.

Доказательство эквивалентности грамматик оставляем в качестве самостоятельного упражнения. СЛАЙД 11.

Пример 55.

Устраним ε-порождающие нетерминалы из грамматики с продукциями $S \to ABaC$, $A \to BC$, $B \to b \mid \varepsilon$, $C \to D \mid \varepsilon$, $D \to d$. Применив первую часть теоремы 8.2, мы обнаружим, что ε-порождающими нетерминалами являются A, B, C. Применив вторую часть этой теоремы, получаем эквивалентную грамматику с продукциями $S \to ABaC \mid BaC \mid AaC \mid ABa \mid aC \mid Aa \mid Ba \mid a$, $A \to BC \mid B \mid C$, $B \to b$, $C \to D$, $D \to d$.

Удаление цепных продукций

Цепная продукция – это продукция вида $A \to B$, где и A, и B являются нетерминалами. Эти продукции могут быть полезными, например, для получения однозначных грамматик. Однако они могут усложнять некоторые из доказательств и создавать лишние шаги порождений.

Для удаления можно использовать правило подстановки, сформулированное в виде теоремы, принимаемой без доказательства. СЛАЙД 13.

Теорема 8.3. Продукция вида $A \to x_1Bx_2$ может быть удалена из грамматики при условии $A \neq B$, если заменим ее набором продукций, в которых B заменяется всеми строками, порождаемыми ею за один шаг.

Пример 56.

Пусть в грамматике G есть продукции $A \to a \mid aaA \mid abBc$, $B \to abbA \mid b$. Используя правило подстановки для нетерминала B, мы получим грамматику G_1 с продукциями $A \to a \mid aaA \mid ababbA \mid abbc$, $B \to abbA \mid b$. СЛАЙД 14.

Применяя правило подстановки с известной долей осторожности, мы можем избавиться от цепных правил.

Теорема 8.4. Пусть G – КСГ и $L(G)$ не содержит ε. Тогда существует эквивалентная ей КСГ G_1, не содержащая цепных продукций. СЛАЙД 15.

Доказательство. Очевидно, что любая продукция вида $A \to A$ может быть безопасно удалена из грамматики, и нам остается рассматривать случай $A \to B$. На первый взгляд может показаться, что можно применять теорему 8.3 напрямую, допустив $x_1 = x_2 = \varepsilon$, чтобы заменить $A \to B$ чем-то вроде $A \to y_1 \mid \ldots \mid y_n$. Однако это не всегда дает нужный эффект. Скажем, из продукций $A \to B$, $B \to A$ нельзя удалить цепные продукции. Чтобы обойти эту проблему, мы должны для всех нетерминалов A найти нетерминалы B такие что:

$$A \Rightarrow^* B. \qquad (8.1)$$

Мы можем сделать это с помощью ГЗ добавлением в него ребра (C, D) всякий раз, когда есть продукция $C \to D$. Это распространяется и на (8.1), т.к. это выражение хранит путь от A к B. Грамматика G_1 конструируется из G во-первых, копированием в P_1 всех нецепных продукций из P, и во-вторых, для всех A и B, удовлетворяющих условию (8.1), мы добавляем в P_1 продукции $A \to y_1 \mid \ldots \mid y_n$, где $B \to y_1 \mid \ldots \mid y_n$ – это набор продукций в P_1, имеющих нетерминал B в LHS. Важно отметить, что эти продукции из P_1, поэтому ни один из y_i не может быть одиночным нетерминалом, так что на последнем шаге не создается ни одной цепной продукции.

Эквивалентность результирующей и исходной грамматик может быть доказана использованием теоремы 8.3.

Пример 57.

Удалим все цепные продукции из грамматики с продукциями $S \to Aa \mid B$, $B \to A \mid bb$, $A \to a \mid bc \mid B$. Граф зависимостей приведен на СЛАЙДЕ 16. Из него мы видим, что $S \Rightarrow^* A$, $S \Rightarrow^* B$, $B \Rightarrow^* A$, $A \Rightarrow^* B$. Следовательно, к первоначальным нецепным продукциям $S \to Aa$, $B \to bb$, $A \to a \mid bc$ мы добавляем новые правила $S \to a \mid bc \mid bb$, $A \to bb$, $B \to a \mid bc$. В итоге получаем следующие продукции $S \to Aa \mid a \mid bc \mid bb$, $A \to a \mid bb \mid bc$, $B \to a \mid bb \mid bc$.

Так же видно, что удаление цепных продукций привело к тому, что нетерминал B и связанные с ним продукции сделались бесполезными.

Мы можем объединить полученные выводы, чтобы показать, что грамматики для КСЯ могут быть свободными от бесполезных, ε- и цепных продукций.

Теорема 8.5. Пусть L – это КСЯ, который не содержит ε. Тогда существует КСГ G такая, $L = L(G)$, и в G нет бесполезных, ε- и цепных продукций. СЛАЙД 17.

Доказательство. Процедуры, сформулированные в теоремах 8.4, 8.2 и 8.1, поочередно удаляют все названные типы продукций. Единственный момент, который нуждается в пояснении, заключается в том, что удаление одного типа продукций может

приводить к появлению продукций другого типа. Например, устранение ε-продукций может давать в результате цепные продукции. Кроме того, теорема 8.4 требует, чтобы в КСГ не было ε-продукций. С другой стороны, легко проверить, что удаление цепных продукций не создает ε-продукций, а удаление бесполезных продукций не приводит к созданию ε- и цепных продукций. Следовательно, мы можем избавиться от этих нежелательных элементов, используя следующий порядок выполняемых шагов.

1. Удаляются ε- продукции.
2. Удаляются цепные продукции.
3. Удаляются бесполезные продукции.

После третьего шага КСГ не будет содержать ни одну из этих продукций, и теорема доказана.

Нормальная форма Хомского

Теорема 8.6. Любая КСГ G для языка L такого, что $L(G)$ не содержит ε, имеет эквивалентную ей КСГ G_{cnf} в НФХ. СЛАЙД 18.

Доказательство. По теореме 8.5 мы можем допустить без потери общности, что G не имеет ε- и цепных продукций. Конструирование G_{cnf} можно выполнить за два шага.

Шаг 1. На основе G создадим грамматику $G_1 = (V_T, V_{N,1}, S, P_1)$ путем рассмотрения в P всех продукций вида

$$A \to x_1 \ldots x_n, \tag{8.2}$$

где каждый x_i – это символ из множества терминальных или нетерминальных символов. Если $n = 1$, то x_i является терминалом, т.к. в нашей грамматике нет цепных продукций. В этом случае копируем эту продукцию в P_1. Если $n > 1$, то вводим новый нетерминал B_a для всех a из V_T. Для каждой продукции в P, имеющей форму (8.2), мы размещаем в P_1 продукцию $A \to C_1 \ldots C_n$, где $C_i = x_i$, если последний является нетерминалом, и $C_i = B_a$, если $x_i = a$.

Для всех B_a мы также создаем в P_1 продукции вида $B_a \to a$.

В этой части алгоритма будут удалены все терминалы из тех продукций, чьи RHS имеет длину более одного символа. Они заменяются на вновь создаваемые нетерминалы. Иначе говоря, по окончании первого шага будет получена G_1, все продукции которой имеют форму

$$A \to a, \tag{8.3}$$

либо

$$A \to C_1 \ldots C_n, \tag{8.4}$$

где C_i принадлежит $V_{N,1}$.

Это следует из теоремы 8.3, что $L(G_1) = L(G)$.

Шаг 2. В этой части алгоритма мы вводим дополнительные нетерминалы, чтобы уменьшить длину RHS продукций, если это необходимо. Первым делом мы копируем в P_{cnf} все продукции вида (8.3), а также все продукции вида (8.4) для $n = 2$. Если $n > 2$, то вводятся новые нетерминалы $D_1, D_2, \ldots$ и в P_{cnf} вводятся продукции $A \to C_1 D_1$, $D_1 \to C_2 D_2$, ..., $D_{n-2} \to C_{n-1} C_n$. Очевидно, результирующая грамматика G_{cnf} находится в НФХ. Повторное применение теоремы 8.3 покажет, что $L(G_1) = L(G_{cnf})$, так что $L(G) = L(G_{cnf})$. Этот в какой-то степени неформальный аргумент можно уточнить, но это мы оставляем на самостоятельное изучение.

Пример 58.

Преобразуем в НФХ грамматику с продукциями $S \to ABa$, $A \to aab$, $B \to Ac$. См. демонстрацию в JFLAP.

Нормальная форма Грейбах

В этой нормальной форме (НФГ) ограничения накладываются не на длину RHS, а на позиции символов в ней. Аргументы, обосновывающие НФГ немного сложнее и менее

прозрачны для понимания. Аналогичным образом, поиск НФГ, эквивалентной заданной КСГ, может оказаться утомительным. Мы рассмотрим ее очень бегло, хотя у этой формы имеются интересные теоретические и практические следствия.

Говорят, что КСГ находится в НФГ, если все продукции имеют вид $A \to ax$, где a – терминал, а x – это нетерминал либо ε.

Если КСГ не находится в НФГ, то ее можно переписать с использованием обсуждавшихся выше приемов. СЛАЙД 19.

Пример 59.

Преобразуем в НФГ грамматику с продукциями $S \to AB$, $A \to aA \mid bB \mid b$, $B \to b$. Она очевидно, не НФГ. Используя теорему 8.3, мы немедленно получим нужный результат – НФГ-грамматику $S \to aAB \mid bBb \mid bB$, $A \to aA \mid bB \mid b$, $B \to b$. СЛАЙД 20.

Пример 60.

Преобразуем в НФГ грамматику с продукциями $S \to abSb \mid aa$. Воспользуемся приемом, показанным в теореме 8.6, и введем новые нетерминалы A и B, которые всего лишь синонимы для терминалов a и b. Заменив терминалы, ассоциированными с ними нетерминалами, мы придем к нужному виду грамматики. Продукции: $S \to aBSB \mid aA$, $A \to a$, $B \to b$. СЛАЙД 21.

Вообще говоря, ни само преобразование КСГ в НФГ, ни доказательство правильности этого не делается обычно легким образом. Однако концептуальной роли для нас НФГ играть в дальнейшем не будет, поэтому следующее утверждение дается без доказательства.

Теорема 8.7. Любая КСГ G для языка L такого, что $L(G)$ не содержит ε, имеет эквивалентную ей КСГ G_{gnf} в НФГ.

8.2 Лемма о разрастании контекстно-свободных языков

Первый шаг на пути к лемме о разрастании для КСЯ состоит в том, чтобы рассмотреть вид и размер деревьев разбора. Одно из практически полезных применений НФХ – преобразование сильноветвящихся деревьев разбора в двоичные деревья. У них есть полезные свойства, и одно из них формулируется следующей теоремой.

Теорема 8.8. Пусть дано дерево разбора, соответствующее КСГ G в НФХ, и пусть кроной дерева является терминальная строка w. Если n – наибольшая длина пути от корня к листьям, то $|w| \le 2^{n-1}$. СЛАЙД 22.

Доказательство. Используем индукцию по n. В базисной части предполагаем, что $n = 1$. Поскольку длина пути отличается от количества ребер на 1, то дерево максимальной длиной пути 1 состоит из корня и листа, отмеченного терминалом. Строка w является этим терминалом, и $|w| = 1$. $2^{n-1} = 2^0 = 1$, значит, базис доказан.

В индуктивной части доказательства предполагаем, что самый длинный путь имеет длину n, и $n > 1$. Корень дерева использует продукцию, которая должна иметь вид $A \to BC$, поскольку $n > 1$. Ни один из путей в поддеревьях с корнями в B и C не может иметь длину больше n-1, т.к. в этих путях нет ребра от корня к потомку, помеченному B или C. По предположению индукции эти два дерева имеют кроны длины не более 2^{n-2}. Крона всего дерева представляет собой конкатенацию этих двух крон, поэтому имеет длину не более $2^{n-2} + 2^{n-2} = 2^{n-1}$. Доказана и индуктивная часть утверждения.

Лемма о разрастании для КСЯ похожа на аналогичную лемму для РЯ, только каждая строка z разбивается на пять частей, и совместно растут вторая и четвертая из них.

Теорема 8.9. «Лемма о разрастании для КСЯ». Пусть L – КСЯ, тогда существует

такое число n, что если z – произвольная строка из L, длина которой не меньше n, то можно записать $z = uvwxy$, причем выполняются следующие условия.

1. $|vwx| \le n$. Таким образом, средняя часть не слишком длинная.

2. $vx \ne \varepsilon$. По причине того, что v и x подстроки, которые должны разрастись, одна из них не может быть пустой.

3. uv^iwx^iy принадлежит L для всех $i \ge 0$. Две строки v и x могут разрастаться произвольное число раз, включая 0, и полученная при этом строка также будет принадлежать L.

Доказательство. Сначала для L найдем грамматику G в НФХ. Это невозможно если $L = \{\varepsilon\}$ или $L = \varnothing$. Во втором из этих случаев утверждение леммы не может быть нарушено, т.к. строки z нет в $\varnothing$. НФХ грамматики G в действительности порождает L-$\{\varepsilon\}$, но это не имеет значения, т.к. выбирается $n > 0$, и z не может быть ε.

Итак, пусть в НФХ-грамматика G имеет m нетерминалов и порождает язык $L(G) = L$-$\{\varepsilon\}$. Выбираем $n = 2^m$. Предположим, что z из L имеет длину не менее n. По теореме 8.8 любое дерево разбора, наибольшая длина путей в котором не превышает m, должно иметь крону длиной не более $2^{m-1} = n/2$. Такое дерево разбора не может иметь крону z, т.к. для этого она слишком длинная. Таким образом, любое дерево разбора с кроной z имеет путь длины не менее m+1. СЛАЙД 24.

На СЛАЙДЕ 25 представлен самый длинный пусть в дереве для z. Его длина равна k+1, где $k \ge m$, поэтому на пути встречается не менее m+1 нетерминалов A_0, A_1, …, A_k. Однако V_N содержит, как мы условились, всего m различных символов, поэтому хотя бы два из m+1 последних нетерминалов на пути от A_{k-m} до A_k (включительно) должны совпадать. Пусть $A_i = A_j$, где $k\text{-}m \le i \le j \le k$.

Тогда дерево можно разделить так, как показано СЛАЙДЕ 26. Строка w является кроной поддерева с корнем A_j. Строки v и x – это цепочки соответственно слева и справа от w в кроне большего поддерева с корнем в A_i. В G нет цепных продукций, поэтому v и x не могут быть одновременно пустыми. Цепочки u и y образуют части z, лежащие слева и справа от поддерева с корнем A_i.

Если $A_i = A_j = A$, то по исходному дереву разбора можно построить новое дерево, как показано на СЛАЙДЕ 27. Сначала можно заменить поддерево с корнем A_i, имеющее крону vwx, поддеревом с корнем A_j с кроной w. Это допустимо, т.к. корни помечены одним и тем же символом A. Полученное дерево представлено там же в части (б). Оно имеет крону и соответствует случаю $i = 0$ в строке uv^iwx^iy.

В части (в) на СЛАЙДЕ 28 представлена еще одна возможность. Там поддерево с корнем A_j заменено поддеревом с корнем A_i. Это допустимо по той же причине, что и в части (а). Крона поддерева – uv^2wx^2y. Если бы мы потом заменили поддерево с кроной w бо̲льшим поддеревом с кроной vwx, то получили бы дерево с кроной uv^3wx^3y. Этот процесс можно продолжать для любого i. Итак, в нашей грамматике имеются деревья разбора для всех строк указанного вида (uv^iwx^iy).

Мы можем также успешно расправиться с условием 1, где $|vwx| \le n$. Мы выбирали A_i как можно ближе к кроне дерева, поэтому $k\text{-}i \le m$. Тогда самый длинный путь в поддереве с корнем A_i имеет длину не более m+1. Согласно теореме 8.8 поддерево с корнем A_i имеет крону, длина которой не больше $2^m = n$. Лемма о разрастании доказана. СЛАЙД 20.

Лемму о разрастании КСЯ можно использовать в виде игры с противником следующим образом (СЛАЙД 30).

1. Мы выбираем язык L, желая доказать, что он не КСЯ.

2. Противник выбирает заранее неизвестное n, поэтому мы можем рассчитывать на любое возможное значение.

3. Мы выбираем z и при желании используем n как параметр.

4. Противник разбивает z на 5 частей, соблюдая ограничения $|vwx| \le n$, $vx \ne \varepsilon$.

5. Мы выигрываем, если смогли, выбрав i, показать, что uv^iwx^iy не принадлежит

языку L.

Пример 61. СЛАЙД 31.

Пусть $L = \{0^n1^n2^n \mid n \geq 1\}$. Предполагаем, что L – КСЯ, тогда существует n из леммы о разрастании. Выбираем $z = 0^n1^n2^n$.

Наш противник разбивает z на пять частей с соблюдением всех условий. Тогда нам известно, что vwx не может включать одновременно нули и двойки, т.к. последний нуль и первая двойка разделены n+1 позициями. Докажем, что L содержит некоторую строку, которая не может быть в L. Возможны следующие случаи.

1. vwx не имеет двоек, т.е. vx состоит только из 0 и 1 и содержит хотя бы один из этих символов. Тогда строка uwy, которая по лемме должна быть в L, имеет n двоек, но меньше, чем n нулей и единиц. Значит, она не принадлежит L, и в этом случае L – не КСЯ.

2. vwx не имеет нулей. Аналогично, uwy имеет n нулей, но меньше двоек или единиц, поэтому не принадлежит L.

В любом случае мы приходим к выводу, что L содержит строку, которая не может ему принадлежать. Мы пришли к противоречию, которое позволяет заключить, что наше предположение ложно. Следовательно, L не является КСЯ.

8.3 Свойства замкнутости контекстно-свободных языков

Сразу же оговоримся, но без должной аргументации, что КСЯ не замкнуты относительно пересечения и разности. При этом пересечение и разность КСЯ и РЯ всегда в результате дает КСЯ.

Для исследования свойств замкнутости КСЯ необходимо ввести операцию подстановки, по которой каждый символ в цепочках из одного языка заменяется целым языком.

Подстановки

Пусть Σ – это алфавит. Пусть для каждого символа a из алфавита выбран язык L_a. Выбранные языки могут быть в любых алфавитах, не обязательно одинаковых и не обязательно совпадающих с Σ. Выбор языков определяет функцию s (*substitution*, **подстановка**) на Σ, и L_a обозначается как $s(a)$ для всех a.

Если $w = a_1 \ldots a_n$ – строка из Σ^*, то $s(w)$ представляет собой язык всех строк $x_1 \ldots x_n$, у которых x_i принадлежит языку $s(a_i)$. То есть $s(w)$ является конкатенацией языков $s(a_1) \ldots s(a_n)$. Это определение можно распространить и на языки. $s(L)$ – это объединение $s(w)$ по всем строкам из L. СЛАЙД 32.

Пример 62. СЛАЙД 33.

Пусть $s(0)=\{a^nb^n : n > 0\}$, $s(1) = \{aa, bb\}$. Подстановка определяется на алфавите $\{0,1\}$. Язык $s(0)$ представляется собой множество строк с одним или несколькими символами a, за которыми идет такое же количество b, а $s(1)$ – конечный язык из двух строк aa и bb. Пусть $w = 01$, тогда $s(w) = s(0)s(1)$. Более точно, $s(w)$ состоит из всех строк вида a^nb^naa и a^nb^{n+2}, где $n > 0$.

Предположим, $L = L(0^*)$. Это набор строк из произвольного количества нулей. Тогда $s(L) = (s(0))^*$. Этот язык является набором цепочек вида $a^{n1}b^{n1} \ldots a^{nk}b^{nk}$ для некоторого $k \geq 0$ и произвольной последовательности положительных целых чисел $n_1, \ldots, n_k$.

Теорема 8.10. Если L – КСЯ в алфавите Σ, а s – подстановка на Σ, при которой $s(a)$ является КСЯ для каждого a из Σ, то $s(L)$ также является КСЯ. СЛАЙД 34.

Доказательство. Дадим общую идею. Берется КСГ для L, и каждый терминал a заменяется аксиомой грамматики для языка $s(a)$. В результате получится единственная КСГ, порождающая $s(L)$.

Пусть грамматика $G = (\Sigma, V_N, P, S)$ задает язык L, а КСГ $G_a = (V_{T,a}, V_{N,a}, P_a, S_a)$ – язык,

подставляемый вместо каждого a из Σ. Поскольку для нетерминалов можно выбирать любые имена, то обеспечим непересечение множеств имен символов в V_N и $V_{N,a}$. Цель – гарантировать, что при сборе продукций из разных грамматик в одном множестве случайно не смешаются продукции двух грамматик, и таким образом, получить порождения, невозможные в данных грамматиках.

Новая грамматика $G' = (V_T', V_N', P', S)$ для $s(L)$ по следующим правилам.

1. V_N' представляет собой объединение V_N и $V_{N,a}$ по всем a.

2. V_T' является объединением $V_{T,a}$ по a из Σ.

3. P' состоит из всех продукций каждого из P_a для a из Σ и всех продукций P с заменой в их RHS каждого терминала a на S_a.

Таким образом, все деревья разбора в G' начинаются как деревья разбора в G, но вместо порождения кроны в Σ^* он содержат границу, на которой все узлы отмечены нетерминалами S_a вместо a. Каждый такой узел является корнем дерева в G_a, крона которого представляет собой терминальную строку из $s(a)$.

Требуется доказать, что эта конструкция правильна в том смысле, что G' порождает язык $s(L)$. Мы оставим это для самостоятельного изучения. Крона дерева приведена на СЛАЙДЕ 35.

Теорема 8.11. КСЯ замкнуты относительно операций объединения, конкатенации, замыкания, транзитивного замыкания и гомоморфизма. СЛАЙД 36.

Доказательство. Для каждой операции требуется определение соответствующей подстановки и использование теоремы 8.10.

Пусть L_1 и L_2 – КСЯ. Тогда $L_1 \cup L_2$ – является языком $s(L)$, где L – язык $\{1, 2\}$, а s – подстановка, определяемая как $s(1) = L_1$ и $s(2) = L_2$.

L_1L_2 – также является языком $s(L)$, где L – язык $\{12\}$, а s – подстановка, определяемая как $s(1) = L_1$ и $s(2) = L_2$.

Если L_1 – КСЯ, L – язык $\{1\}^*$, а s – подстановка, определяемая как $s(1) = L_1$, то $L_1^* = s(L)$.

Если L_1 – КСЯ, L – язык $\{1\}^+$, а s – подстановка, определяемая как $s(1) = L_1$, то $L_1^+ = s(L)$.

Пусть L – КСЯ над алфавитом Σ, а h – гомоморфизм на алфавите. Пусть s – это подстановка, заменяющая каждый символ a из Σ языком, состоящим из единственной строки $h(a)$. Иначе говоря, $s(a)=\{h(a)\}$ для всех a. Тогда $h(L) = s(L)$.

КСЯ также замкнуты относительно обращения. Для доказательства использовать теорему 8.10 нельзя, но есть простая конструкция на основе грамматик. СЛАЙД 37.

Теорема 8.12. Если L – КСЯ, то L^R – тоже КСЯ.

Доказательство. Снова даем только общую идею. Пусть $L = L(G)$ для КСГ $G = (V_T, V_N, P, S)$. Построим $G^R = (V_T, V_N, P^R, S)$, где продукции P^R представляют собой обращения продукций из P. Таким образом, если $A \to \alpha$ – продукция в G, то $A \to \alpha^R$ – продукция G^R. Используя индукцию по длине порождений в G и в G^R, нетрудно показать, что $L(G^R) = L^R$. По сути, все выводимые в G^R строки являются обращениями строк, выводимых в G и наоборот.

Теорема 8.13. Если L – КСЯ, а R – РЯ, то их пересечение – КСЯ.

Доказательство. Здесь можно исходить из моделирования КСЯ с помощью МПА, а РЯ – с помощью КА. Они запускаются «параллельно», и в результате должен быть получен новый МПА.

Теорема 8.14. Если L – КСЯ, а h – гомоморфизм. Тогда $h^{-1}(L)$ – КСЯ.

Приводим без доказательства, которое также сводится к моделированию поведения МПА и индукции по количеству переходов, совершаемых исходным и результирующим автоматами.

Литература к лекции 8

1. Контекстно-свободная грамматика - http://ru.wikipedia.org/wiki/Контекстно-свободная_грамматика
2. Короткова, М.А. Математическая теория автоматов. Учебное пособие / М.А. Короткова. – М.: МИФИ, 2008. – 116 с.
3. Кузнецов, А.С. Теория вычислительных процессов [Текст] : учеб. пособие / А. С. Кузнецов, М. А. Русаков, Р. Ю. Царев ; Сиб. федерал. ун-т. - Красноярск: ИПК СФУ, 2008. – 184 с.
4. Молчанов, А. Ю. Системное программное обеспечение. 3-е изд. / А.Ю. Молчанов. – СПб.: Питер, 2010. – 400 с.
5. Серебряков В. А., Галочкин М. П., Гончар Д. Р., Фуругян М. Г. Теория и реализация языков программирования — М.: МЗ-Пресс, 2006 г., 2-е изд. - http://trpl7.ru/t-books/TRYAP_BOOK_Details.htm
6. Теория автоматов / Э. А. Якубайтис, В. О. Васюкевич, А. Ю. Гобземис, Н. Е. Зазнова, А. А. Курмит, А. А. Лоренц, А. Ф. Петренко, В. П. Чапенко // Теория вероятностей. Математическая статистика. Теоретическая кибернетика. — М.: ВИНИТИ, 1976. — Т. 13. — С. 109–188. — URL http://www.mathnet.ru/php/getFT.phtml?jrnid=intv&paperid=28&what=fullt&option_lang=rus
7. Нормальная форма Хомского - http://ru.wikipedia.org/wiki/Нормальная_форма_Хомского
8. Свойства контекстно-свободных языков - http://www.williamspublishing.com/PDF/978-5-8459-1347-0/part7.pdf
9. Контекстно-свободные грамматики - http://www.math.spbu.ru/user/mbk/PDF/Ch-4.pdf