Лекция 11

Теория формальных языков (III)

(Конспект: В. Вялов)

11.1 Бесконтекстные языки

11.1.1 Нормальные формы бесконтекстных языков

Определение 11.1. Будем говорить, что грамматика — в *нормальной форме Хомского* (*НФХ*), если она содержит только правила одного из следующих типов.

1. $A \to BC$.

2. $A \to a$.

3. $S \to \epsilon$.

Причем

4. Стартовый символ S в правых частях не встречается (т.е. $B, C \neq S$ в пункте 1).

Теорема 11.1. *Для любой бесконтекстной грамматики G существует грамматика G^* в НФХ, такая, что $L(G) = L(G^*)$.*

Доказательство. 1. Добавим новый стартовый символ S^* и правило $S^* \to S$. Тем самым мы заработаем пункт 4.

2. Для каждого терминала a добавим нетерминал A_a и правило $A_a \to a$. Также заменим a на A_a в остальных правилах. Теперь в правых частях правил у нас встречается либо только один терминал (как в пункте 2), либо цепочка нетерминалов (почти как в пунктах 1 и 3). Следующим шагом мы уменьшим длину этой цепочки до не более, чем двух символов.

3. Пусть у нас есть правило $A \to BC\alpha$, где α — непустая цепочка из нетерминалов. Добавим новый нетерминал Z, а правило заменим на два других: $A \to BZ$ и $Z \to C\alpha$. Тем самым мы сократим правую часть правила на один символ. Продолжая действовать таким образом, добьемся нужной длины правой части.

4. Остается избавиться от правил вида $A \to \epsilon$ (где $A \neq S^*$) и $A \to B$. Аналогично лемме о праволинейных грамматиках построим множества $S(B) = \{A \,|\, A \to^* B\}$ и $S(\epsilon) = \{A \,|\, A \to^* \epsilon\}$. Заметим, что $A \in S(B)$ означает, что выполняется хотя бы одно из трех условий:

(1) $A = B$;

(2) $A \to C$, где $C \in S(B)$;

(3) $A \to CD$, где $C \in S(B)$, а $D \in S(\epsilon)$ (или наоборот).

Аналогично для $S(\epsilon)$. Отсюда видно, что множества $S(\ldots)$ можно построить за конечное число шагов: на каждом шаге мы будем для каждой пары нетерминалов (A, B) проверять условия (1)–(3) и добавлять A в $S(B)$, если одно из условий выполнится (и то же самое — для $S(\epsilon)$). Если на очередном шаге мы ничего не добавим ни в одно из множеств, то множества $S(\ldots)$ построены.

Теперь для каждого $A \in S(B)$ и правила $B \to \alpha$ добавим правило $A \to \alpha$. Также, если $S^* \in S(\epsilon)$, то добавим правило $S^* \to \epsilon$. После этого удалим все «плохие» правила, т.е. правила вида $A \to \epsilon$ (где $A \neq S^*$) и $A \to B$.

Заметим, что в результате наших действий язык не изменился. □

Определение 11.2. Будем говорить, что грамматика — в *нормальной форме Грейбах* (*НФГ*), если она содержит только правила одного из следующих типов.

1. $A \to a$.

2. $A \to aB$.

3. $A \to aBC$.

4. $S \to \epsilon$.

Причем

5. Стартовый символ S в правых частях не встречается.

Задача 11.1. Доказать аналогичную теорему для НФГ.

11.1.2 Автоматы с магазинной памятью

Как уже было доказано, любой праволинейной грамматике соответствует конечный автомат. Для бесконтекстной грамматики конечного автомата мало, поэтому мы усложним конструкцию автомата.

Наш автомат будет обладать памятью, а именно, магазином. *Магазин* представляет из себя стек, с которым мы можем делать две вещи:

(1) достать верхний элемент;

(2) положить наверх данный элемент.

Определение 11.3. *Недетерминированный автомат с магазинной памятью* состоит из тех же частей, что и недетерминированный конечный автомат, но надо добавить еще магазинный алфавит M и начальный символ магазина $Z \in M$. Функция перехода теперь такая:

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times M \to 2^{Q \times M^*},$$

при этом требуется, чтобы значения δ были *конечными* множествами.

Переход такого автомата мы будем записывать следующим образом: $q \overset{a,A/\alpha}{\longrightarrow} p$. Это значит, что $(p, \alpha) \in \delta(q, a, A)$. Фактически это означает следующее: мы переходим из состояния q в состояние p, считывая из строки символ a (возможно, никакого символа вовсе не считываем: $a = \epsilon$). При этом мы снимаем с магазина символ A, а на его место помещаем строку α.

Конфигурацией нашего автомата назовем тройку (q, u, γ), где $q \in Q$ — текущее состояние, u — оставшаяся (непрочитанная) часть строки, γ — содержимое магазина (верхний символ записывается слева). Такт работы нашего автомата выглядит так:

$$(q, au, A\beta) \vdash (p, u, \alpha\beta),$$

если $q \overset{a,A/\alpha}{\longrightarrow} p$ (здесь $a \in \Sigma \cup \{\epsilon\}$). Будем говорить, что строка x *принимается автоматом*, если мы сможем из состояния (q_s, x, Z) за конечное число шагов попасть в конечное состояние, исчерпав всю строку, т.е.

$$(q_s, x, Z) \vdash^* (q, \epsilon, \alpha) \qquad (\text{где } q \in F).$$

При этом нам неважно, что именно у нас осталось в магазине.

Пример 11.1. Построим автомат, задающий язык $\{0^n 1^n \mid n \in \mathbb{N} \cup \{0\}\}$. Он будет иметь четыре состояния: начальное q_s, конечное q_f, и два промежуточных q_1, q_2; а также следующие правила:

(1) $q_s \xrightarrow{\epsilon,Z/} q_f$ (чтобы не забыть о случае $n = 0$);

(2) $q_s \xrightarrow{0,Z/0Z} q_1$;

(3) $q_1 \xrightarrow{0,0/00} q_1$;

(4) $q_1 \xrightarrow{1,0/} q_2$;

(5) $q_2 \xrightarrow{1,0/} q_2$;

(6) $q_2 \xrightarrow{\epsilon,Z/} q_f$.

Наш автомат действует следующим образом: считывая на очередном шаге 0, мы кладем его в магазин; считывая 1, вынимаем 0 из магазина. При этом, если единиц будет больше, чем нулей, то мы не сможем дочитать строку до конца; а если наоборот, то мы не увидим дна магазина и не попадем в конечное состояние. □

Теорема 11.2. *По любой бесконтекстной грамматике можно построить недетерминированный магазинный автомат, задающий тот же язык.*

Доказательство. 1. Приведем грамматику к НФХ.

2. В магазин автомата мы будем складывать нетерминалы; наши состояния — q_s (стартовое), q («рабочее») и q_f (конечное); а правилам вида (1)–(3) сопоставим, соответственно, такие переходы:

(1) $q \xrightarrow{\epsilon,A/BC} q$;

(2) $q \xrightarrow{a,A/\epsilon} q$;

(3) $q \xrightarrow{\epsilon,S/\epsilon} q$.

А также добавим еще два правила: $q_s \xrightarrow{\epsilon,Z/SZ} q$ («начинаем вывод со стартового символа») и $q \xrightarrow{\epsilon,Z/\epsilon} q_f$ («увидев дно магазина, следует завершить вывод»). Для того, чтобы убедиться, что полученный автомат задает тот же самый язык, достаточно доказать по индукции (упражнение) следующую лемму.

Лемма 11.1. $(q, uv, \alpha\beta) \vdash^* (q, v, \beta)$ *для нашего автомата выполняется тогда и только тогда, когда* $\alpha \Rightarrow^* u$ *для исходной грамматики (в НФХ).*

□

Задача 11.2. Доказать обратную теорему.

11.1.3 Алгоритмические проблемы, связанные с бесконтекстными языками

Алгоритм проверки принадлежности слова языку. Цель — проверить, принадлежит ли слово $a_1 a_2 \dots a_n$ языку, задаваемому грамматикой G (очевидно, можно считать, что она — в НФХ).

Решать эту задачу мы будем методом динамического программирования. Обозначим t_{ij} множество тех нетерминалов, из которых можно получить строку $a_i \dots a_{i+j-1}$. Множество t_{i1} состоит из терминалов, для которых в G имеется правило $A \to a_i$. Для того, чтобы построить множество t_{ij} будем разбивать строку по позиции $i+k-1$ и смотреть, из чего выводится каждая из половинок. Иначе говоря, $A \in t_{ij}$, если существует правило $A \to BC$, где $B \in t_{ik}$ и $C \in t_{i+k,j-k}$. Отсюда видно, что все множества t_{ij} можно построить за кубическое от длины строки количество шагов. При этом исходная строка принадлежит языку, если стартовый символ лежит в множестве t_{1n}.

Проверка пустоты языка. Для решения этой задачи докажем следующую лемму:

Лемма 11.2 (лемма о разрастании для бесконтекстных языков). *Для любого бесконтекстного языка L существует число k, такое, что, если длина слова α больше чем k, то его можно представить в виде $\alpha = xuvwy$, где длина слова uvw — менее k, слово uw не пусто, и $xu^i v w^i y \in L$ для всех $i \geqslant 0$.*

Доказательство. Рассмотрим грамматику в НФХ, порождающую этот язык. В качестве k возьмем $2^{|N|+2}$.

Построим дерево вывода строки α. В корень поместим стартовый символ S. Потомками вершины, помеченной символом X, будут вершины, помеченные символами правой части правила, которое было применено для X при выводе строки α. (Упорядочим их так же, как они были расположены в правой части правила.) Листьями будут терминалы (символы строки α).

По условию, в дереве $\geqslant 2^{|N|+2}$ листьев, поэтому его высота — не менее $|N|+2$. Значит, существует путь из корня в лист такой длины. Тогда на этом пути какой-то нетерминал B встретится дважды.

Рассмотрим основное дерево, а также получившиеся два поддерева с корнями в вершинах, где сидит этот символ. В результате слово разобьется на необходимые нам пять частей (см. рис. 11.1). По построению, $B \Rightarrow^* uBw$ и $B \Rightarrow^* v$. Слово uw непусто, поскольку грамматика была в НФХ.

```
          S
         /|\
        / | \
       /  |  \
      /   B   \
     /   /|\   \
    /   / | \   \
   /   /  |  \   \
  /   /   B   \   \
 /   /   / \   \   \
/   /   /   \   \   \
 x   u    v   w    y
```

Рис. 11.1: Разбиение $\alpha = xuvwy$. Пунктирная черта обозначает «границы» соответствующего поддерева.

Осталось добиться того, чтобы выполнялось $|uvw| < k$.

Задача 11.3. Сделать это.

$\square$

Теперь легко проверить пустоту языка. Если язык не является пустым, то в нем должна существовать строка длины меньшей, чем k (пока строка удовлетворяет лемме, ее можно при помощи леммы заменять на другую строку, меньшую по длине), а количество таких строк конечно. Итого нам надо будет лишь конечное число раз запустить алгоритм, распознающий принадлежность строки нашему языку.