Теория автоматов и формальных языков

Контекстно-свободные языки

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

27 сентября 2016г.

В предыдущей серии

- Регулярные выражения, регулярные грамматики и конечные автоматы задают класс регулярных языков
- Класс регулярных языков замкнут относительно теоретико-множественных операций, конкатенации, итерации, гомоморфизма цепочек
- Определение принадлежности слова языку осуществляется за $O(n)$ операций
- Однако класс регулярных языков достаточно узок, ни один используемый в промышленности язык программирования не является регулярным
  - Лемма о накачке для доказательства нерегулярности языка
  - Язык правильных скобочных последовательностей, язык палиндромов не являются регулярными

Контекстно-свободная грамматика

Четверка $\langle V_T, V_N, P, S\rangle$

- V_T — алфавит терминальных символов (терминалов)
- V_N — алфавит нетерминальных символов (нетерминалов)
  - $V_T \cap V_N = \emptyset$
  - $V ::= V_T \cup V_N$
- P — конечное множество правил вида $A \to \alpha$
  - $A \in V_N$
  - $\alpha \in V^*$
- S — начальный нетерминал грамматики, $S \in V_N$

Пример: арифметические выражения

$$
\begin{array}{lcl}
E & \to & E + E \,|\, E * E \,|\, N \\
N & \to & 0 \,|\, 1 \,|\, \ldots \,|\, 9
\end{array}
$$

Вывод в грамматике

- **Отношение выводимости**:
  $\forall \alpha, \gamma, \delta \in V^*, A \in V_N : A \to \alpha \in P. \gamma A \delta \Rightarrow \gamma \alpha \delta$
- **Вывод** — транизитивное, рефлексивное замыкание отношения выводимости ($\overset{*}{\Rightarrow}, \overset{+}{\Rightarrow}, \overset{k}{\Rightarrow}$)
- **Левосторонний (правосторонний) вывод** — на каждом шаге заменяем самый левый (правый) нетерминал
  - Если не специфицируется, подразумевается левосторонний вывод
- По сути, правила грамматики рассматриваются как правила переписывания

Пример вывода

Построим левосторонний вывод цепочки $2 + 3 * 4$ в грамматике $\langle\{0, 1, \ldots, 9, +, *\}, \{E, N\}, P, E\rangle$

$$
\begin{array}{lcl}
E & \to & E + N \,|\, E * N \,|\, N \\
N & \to & 0 \,|\, 1 \,|\, \ldots \,|\, 9
\end{array}
$$

$$E \Rightarrow E * N \Rightarrow E + N * N \Rightarrow N + N * N \Rightarrow 2 + N * N \overset{2}{\Rightarrow} 2 + 3 * 4$$

Существование левостороннего вывода

Теорема

Если для цепочки ω существует некоторый вывод $S \overset{*}{\Rightarrow} \omega$, то существует и левосторонний вывод для этой цепочки $S \underset{l}{\overset{*}{\Rightarrow}} \omega$

Доказательство.

Докажем более общее утверждение: если существует $A \overset{*}{\Rightarrow} \omega$, то существует $A \underset{l}{\overset{*}{\Rightarrow}} \omega$, где $A \in V_N$.

Доказываем по индукции по длине вывода k

$k = 1 : A \Rightarrow \omega$ — тривиально.

$k \to k + 1 : \lhd A \Rightarrow \alpha \overset{*}{\Rightarrow} \omega$.

Обозначим $\alpha = B_1 B_2 \ldots B_m \overset{*}{\Rightarrow} \omega_1 \omega_2 \ldots \omega_m = \omega; \forall i. B_i \overset{l_i}{\Rightarrow} \omega_i, l_i \leq n$

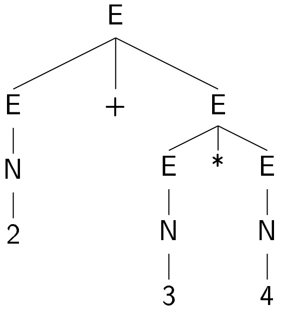
По индукционному предположению $\forall i.\ B_i \underset{l}{\overset{*}{\Rightarrow}} \omega_i$

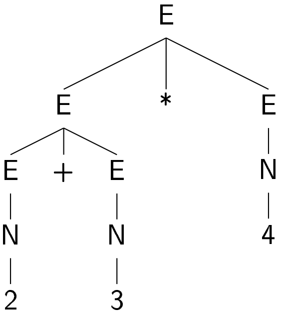
$\models\!\Rightarrow: A \Rightarrow B_1 B_2 \ldots B_m \underset{l}{\overset{*}{\Rightarrow}} \omega_1 B_2 \ldots B_m \underset{l}{\overset{*}{\Rightarrow}} \omega$ — левосторонний вывод $\square$

Единственность вывода

Не всегда (левосторонний) вывод единственен: 2 вывода строки $2 + 3 * 4$

$$
\begin{array}{lcl}
E & \to & E + E \mid E * E \mid N \\
N & \to & 0 \mid 1 \mid \ldots \mid 9
\end{array}
$$





Однозначность грамматики

- Грамматика называется **однозначной**, если для *любого* слова языка существует *единственный* (левосторонний) вывод
- Грамматика называется **неоднозначной**, если *существует* слово языка, такое что для него *существует несколько* (левосторонних) выводов
- По однозначной грамматике можно тривиальным образом построить неоднозначную: продублировать правило
  - $S \to A; A \to a$
  - $S \to A|B; A \to a; B \to a$
- Не существует общего алгоритма преобразования неоднозначной грамматики в однозначную

Примеры однозначной и неоднозначной грамматики

- Неоднозначная грамматика

$$\begin{array}{rcl} E & \to & E + E \mid E * E \mid N \\ N & \to & 0 \mid 1 \mid \ldots \mid 9 \end{array}$$

- Однозначная грамматика

$$\begin{array}{rcl} E & \to & E + N \mid E * N \mid N \\ N & \to & 0 \mid 1 \mid \ldots \mid 9 \end{array}$$

Проверка однозначности грамматики — неразрешимая задача

- Проверка однозначности грамматик сводится к задаче соответствий Поста
- Задача соответствий Поста: Даны списки $A = (a_1, \ldots, a_n)$ и $B = (b_1, \ldots, b_n)$, где $\forall i. \, a_i \in \Sigma^*$ и $b_i \in \Sigma^*$.Существует ли непустая последовательность $(i_1, \ldots, i_k)$, удовлетворяющая условию $a_{i_1} \ldots a_{i_k} = b_{i_1} \ldots b_{i_k}$, где $\forall j. \, 1 \leq i_j \leq n$

Контекстно-свободный язык

- Язык называется **контекстно-свободным**, если для него *существует* контекстно-свободная грамматика
- Язык, задаваемый КС грамматикой $\langle V_T, V_N, P, S \rangle$: $\{\omega \in V_T^* | S \overset{*}{\Rightarrow} \omega\}$
- КС язык называется **существенно неоднозначным**, если для него не существует однозначной грамматики

Пустота КС языка

Теорема

Существует алгоритм, определяющий, является ли язык, порождаемый КС грамматикой, пустым

Доказательство.

Для доказательства потребуется следующая лемма □

Лемма

Теорема

Если в данной грамматике выводится некоторая цепочка, то существует цепочка, дерево вывода которой не содержит ветвей длиннее m, где m — количество нетерминалов грамматики

Доказательство.

Рассмотрим дерево вывода цепочки ω. Если в нем есть 2 узла, соответствующих одному нетерминалу A, обозначим их n_1 и n_2. Предположим, n_1 расположен ближе к корню дерева, чем n_2; $A_{n_1} \overset{*}{\Rightarrow} \alpha\omega_1\beta$; $A_{n_2} \overset{*}{\Rightarrow} \gamma\omega_2\delta$. При этом ω_2 является подцепочкой ω_1. Заменим в изначальном дереве узел n_1 на n_2. Полученное дерево является деревом вывода $\alpha\omega_2\delta$. Повторяем процесс замены одинаковых нетерминалов до тех пор, пока в дереве не останутся только уникальные нетерминалы.
В полученном дереве не может быть ветвей длины большей, чем m. По постороению оно является деревом вывода. $\square$

Алгоритм проверки пустоты КС языка

Доказательство.

Строим коллекцию деревьев, представляющих вывод в грамматике.

1. Инициализируем коллекцию деревом из одного узла S
2. Добавляем в коллекцию дерево, полученное применением единственного правила грамматики из какого-нибудь дерева из коллекции, если его в нем еще нет, и самая длинная ветвь не длиннее m
3. Если после окончания построения коллекции в ней существует дерево, являющееся деревом вывода некоторой цепочки терминалов, значит, язык не пуст

□