

Теория автоматов и формальных языков

Контекстно-свободные языки: нисходящий анализ

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

25 октября 2016г.

В предыдущей серии

- Восходящий и нисходящий синтаксический анализ
- LL-грамматики

- Top-down parsing
- Начинаем разбирать со стартового нетерминала, применяем правила грамматики, пока не получим строку
 - ▶ С откатом ([full] backtracking)
 - ▶ Без отката (without backtracking)

Нисходящий синтаксический анализ с откатом

- Метод грубой силы, bruteforce
- Перебираем все возможные варианты разбора, если что-то пошло не так — возвращаемся к началу и пробуем снова

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

S

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$$S \Rightarrow aAd$$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$$S \Rightarrow aAd$$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$$S \Rightarrow aAd \Rightarrow acd$$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

S

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

$S \Rightarrow aB$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow accd$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow accd$ — не подходит, откатываемся

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow accd$ — не подходит, откатываемся

$S \Rightarrow aB$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow accd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow addc$

Bruteforce parsing: пример

$$\begin{array}{lcl} S & \rightarrow & aAd \mid aB \\ A & \rightarrow & b \mid c \\ B & \rightarrow & ccd \mid ddc \end{array}$$

$$\omega = addc$$

$S \Rightarrow aAd \Rightarrow abd$ — не подходит, откатываемся

$S \Rightarrow aAd \Rightarrow acd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow accd$ — не подходит, откатываемся

$S \Rightarrow aB \Rightarrow addc$ — ура!

Проблема: ну очень уж долго работает: экспоненциальное время!

Нисходящий синтаксический анализ без отката

- Рекурсивный спуск (recursive descent parsing)
 - ▶ Для каждого нетерминала написана функция
 - ▶ Функции для нетерминалов рекурсивно вызывают друг друга

Рекурсивный спуск: пример

Код

Нисходящий синтаксический анализ без отката: LL(k)

- Идея: откат запрещен, но разрешен предпросмотр
- По (нескольким) следующим терминалам принять решение о том, какую продукцию использовать
- Как и предыдущие 2 подхода не может обрабатывать леворекурсивные правила грамматики
- Достаточно хорош для используемых на практике языков

Леворекурсивные правила грамматики

- Явная (непосредственная) левая рекурсия
 - ▶ $A \rightarrow A\beta$
- Неявная левая рекурсия
 - ▶ $A \rightarrow \alpha A\beta, \alpha \xRightarrow{*} \varepsilon$
- Взаимная рекурсия
 - ▶ $A \rightarrow \alpha B\beta, B \rightarrow \gamma A\delta, \alpha \xRightarrow{*} \varepsilon, \gamma \xRightarrow{*} \varepsilon$

Избавление от левой рекурсии

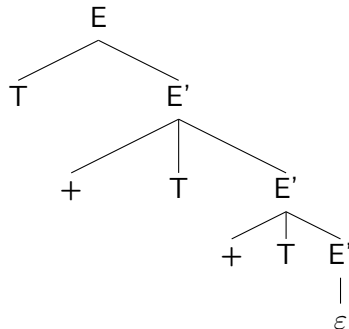
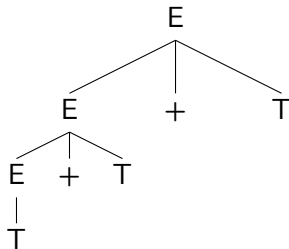
- $A \rightarrow A\alpha \mid \beta \Leftrightarrow A \rightarrow \beta A', A' \rightarrow \varepsilon \mid \alpha A'$

Избавление от левой рекурсии

- $A \rightarrow A\alpha \mid \beta \Leftrightarrow A \rightarrow \beta A', A' \rightarrow \varepsilon \mid \alpha A'$
- $E \rightarrow E + T \mid T \Leftrightarrow E \rightarrow TE', E' \rightarrow \varepsilon \mid + TE'$

Избавление от левой рекурсии

- $A \rightarrow A\alpha \mid \beta \Leftrightarrow A \rightarrow \beta A', A' \rightarrow \varepsilon \mid \alpha A'$
- $E \rightarrow E + T \mid T \Leftrightarrow E \rightarrow TE', E' \rightarrow \varepsilon \mid + TE'$



Избавление от левой рекурсии: более общий случай

- $A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_n \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_k$
- $A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_k A'$
- $A' \rightarrow \varepsilon \mid \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_n A'$

Избавление от взаимной левой рекурсии

- Избавляемся от ε -продукций
- Упорядочиваем правила не индексу нетерминала
- Добиваемся того, чтобы не было правил вида $A_i \rightarrow A_j \alpha, j \leq i$
 - ▶ Перебираем все A_i
 - ▶ Перебираем все $A_j, 1 \leq j < i$
 - ▶ Для каждого правила $p : A_i \rightarrow A_j \gamma$
 - ★ Удалить правило p
 - ★ Для каждого правила $A_j \rightarrow \delta_1 \mid \dots \mid \delta_k$ Добавить правила $A_i \rightarrow \delta_l$
 - ▶ Устранить непосредственную левую рекурсию для A_i

Левая факторизация грамматики

- Выделяем наибольший общий префикс продукций
 $A \rightarrow \alpha\beta \mid \alpha\gamma \Rightarrow A \rightarrow \alpha A', A' \rightarrow \beta \mid \gamma$

Пример

$$\begin{array}{lcl} S & \rightarrow & aSSbS \\ & | & aSaSb \\ & | & abb \\ & | & b \end{array}$$

Пример

$$\begin{array}{lcl} S & \rightarrow & aSSbS \\ & | & aSaSb \\ & | & abb \\ & | & b \end{array}$$

$$\begin{array}{lcl} S & \rightarrow & aS' \\ & | & b \\ S' & \rightarrow & SSbS \\ & | & SaSb \\ & | & bb \end{array}$$

Пример

$$\begin{array}{lcl} S & \rightarrow & aSSbS \\ & | & aSaSb \\ & | & abb \\ & | & b \end{array}$$

$$\begin{array}{lcl} S & \rightarrow & aS' \\ & | & b \\ S' & \rightarrow & SSbS \\ & | & SaSb \\ & | & bb \end{array}$$

$$\begin{array}{lcl} S & \rightarrow & aS' \mid b \\ S' & \rightarrow & SS'' \mid bb \\ S'' & \rightarrow & SbS \mid aSb \end{array}$$

Множество FIRST

- Множество символов, которые могут появиться первыми во время вывода из данной сентенциальной формы
- $FIRST(a\alpha) = \{a\}, a \in V_T, \alpha \in (V_T \cup V_N)^*$
- $FIRST(\varepsilon) = \{\varepsilon\}$
- $FIRST(\alpha\beta) = FIRST(\alpha) \cup (FIRST(\beta), \text{if } \varepsilon \in FIRST(\alpha))$
- $FIRST(S) = FIRST(\alpha) \cup FIRST(\beta), S \rightarrow \alpha \mid \beta$

Множество FIRST: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

Множество FIRST: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FIRST(S) = \{a\}$

Множество FIRST: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FIRST(S) = \{a\}$
- $FIRST(A) = \{a, \varepsilon\}$

Множество FIRST: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FIRST(S) = \{a\}$
- $FIRST(A) = \{a, \varepsilon\}$
- $FIRST(A') = \{a, b\}$

Множество FIRST: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FIRST(S) = \{a\}$
- $FIRST(A) = \{a, \varepsilon\}$
- $FIRST(A') = \{a, b\}$
- $FIRST(B) = \{c, \varepsilon\}$

Множество FIRST: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FIRST(S) = \{a\}$
- $FIRST(A) = \{a, \varepsilon\}$
- $FIRST(A') = \{a, b\}$
- $FIRST(B) = \{c, \varepsilon\}$
- $FIRST(S') = \{a, b, \varepsilon\}$

Множество FOLLOW

- Множество символов, которые могут появиться в некотором выводе сразу после данной сентенциальной формы
- Положим $FOLLOW(X) = \emptyset$
- Если X — стартовый нетерминал,
 $FOLLOW(X) = FOLLOW(X) \cup \{\$$ — символ конца строки
- Для всех правил вида $A \rightarrow \alpha X \beta$,
 $FOLLOW(X) = FOLLOW(X) \cup (FIRST(\beta) \setminus \{\epsilon\})$
- Для всех правил вида $A \rightarrow \alpha X$ и $A \rightarrow \alpha X \beta$, где $\epsilon \in FIRST(\beta)$,
 $FOLLOW(X) = FOLLOW(X) \cup FOLLOW(A)$
- Повторять последние 2 пункта, пока можно что-то добавлять

Множество FOLLOW: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

Множество FOLLOW: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FOLLOW(S) = \{\$ \}$

Множество FOLLOW: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FOLLOW(S) = \{\$ \}$
- $FOLLOW(S') = \{\$ \}$ ($S \rightarrow aS'$)

Множество FOLLOW: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FOLLOW(S) = \{\$ \}$
- $FOLLOW(S') = \{\$ \}$ ($S \rightarrow aS'$)
- $FOLLOW(A) = \{b\}$ ($S' \rightarrow AbBS'$)

Множество FOLLOW: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FOLLOW(S) = \{\$ \}$
- $FOLLOW(S') = \{\$ \}$ ($S \rightarrow aS'$)
- $FOLLOW(A) = \{b\}$ ($S' \rightarrow AbBS'$)
- $FOLLOW(A') = \{b\}$ ($A \rightarrow aA'$)

Множество FOLLOW: пример

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow AbBS' \mid \varepsilon \\ A &\rightarrow aA' \mid \varepsilon \\ A' &\rightarrow b \mid a \\ B &\rightarrow c \mid \varepsilon \end{aligned}$$

- $FOLLOW(S) = \{\$ \}$
- $FOLLOW(S') = \{\$ \}$ ($S \rightarrow aS'$)
- $FOLLOW(A) = \{b\}$ ($S' \rightarrow AbBS'$)
- $FOLLOW(A') = \{b\}$ ($A \rightarrow aA'$)
- $FOLLOW(B) = \{a, b, \$ \}$ ($S' \rightarrow AbBS', \varepsilon \in FIRST(S')$)

- Нисходящий синтаксический анализ с предпросмотром одного символа
- Читает вход слева направо (L: left-to-right), строит левый вывод в грамматике (L: leftmost)
- Состоит из:
 - ▶ Входного буфера (откуда читается входная строка)
 - ▶ Стека (для промежуточных данных)
 - ▶ Таблицы анализатора (управляет процессом разбора)
- Работает за $O(n)$, где n — длина входной строки

Таблица LL(1)-анализатора

$$S \rightarrow (S) \mid \varepsilon$$

Размещаем productions в таблице (по горизонтали — нетерминалы; по вертикали — терминалы + \$)

- Productions вида $A \rightarrow \alpha$ — в ячейки (A, a) , где $a \in FIRST(A)$
- Productions вида $A \rightarrow \varepsilon$ — в ячейки (A, a) , где $a \in FOLLOW(A)$

N	FIRST	FOLLOW	()	\$
S					

Таблица LL(1)-анализатора

$$S \rightarrow (S) \mid \varepsilon$$

Размещаем productions в таблице (по горизонтали — нетерминалы; по вертикали — терминалы + \$)

- Productions вида $A \rightarrow \alpha$ — в ячейки (A, a) , где $a \in FIRST(A)$
- Productions вида $A \rightarrow \varepsilon$ — в ячейки (A, a) , где $a \in FOLLOW(A)$

N	FIRST	FOLLOW	()	\$
S	{(, ε }	{), \$}			

Таблица LL(1)-анализатора

$$S \rightarrow (S) \mid \varepsilon$$

Размещаем productions в таблице (по горизонтали — нетерминалы; по вертикали — терминалы + \$)

- Productions вида $A \rightarrow \alpha$ — в ячейки (A, a) , где $a \in FIRST(A)$
- Productions вида $A \rightarrow \varepsilon$ — в ячейки (A, a) , где $a \in FOLLOW(A)$

N	FIRST	FOLLOW	()	\$
S	{(, ε }	{), \$}	$S \rightarrow (S)$		

Таблица LL(1)-анализатора

$$S \rightarrow (S) \mid \varepsilon$$

Размещаем productions в таблице (по горизонтали — нетерминалы; по вертикали — терминалы + \$)

- Productions вида $A \rightarrow \alpha$ — в ячейки (A, a) , где $a \in FIRST(A)$
- Productions вида $A \rightarrow \varepsilon$ — в ячейки (A, a) , где $a \in FOLLOW(A)$

N	FIRST	FOLLOW	()	\$
S	{(, ε }	{), \$}	$S \rightarrow (S)$	$S \rightarrow \varepsilon$	$S \rightarrow \varepsilon$

Синтаксический анализ (доска)

$$S \rightarrow (S) \mid \varepsilon$$

N	FIRST	FOLLOW	()	\$
S	{(, ε }	{), \$}	$S \rightarrow (S)$	$S \rightarrow \varepsilon$	$S \rightarrow \varepsilon$

$$\omega = (())\$$$

Стек: \$, S,), S, (,), S, (

Когда LL-анализ не возможен

- Леворекурсивные правила
- Когда при построении таблицы в одну ячейку нужно записать больше одной записи
 - ▶ FIRST-FIRST конфликт
 - ★ $A \rightarrow \alpha \mid \beta, FIRST(\alpha) \cap FIRST(\beta) \neq \emptyset$
 - ★ $E \rightarrow T + E \mid T * E$
 - ▶ FIRST-FOLLOW конфликт
 - ★ $FIRST(A) \cap FOLLOW(A) \neq \emptyset$
 - ★ $S \rightarrow Aab, A \rightarrow a \mid \epsilon$
- Как с этим бороться?
 - ▶ Избавиться от левой рекурсии
 - ▶ Избавиться от недетерминизма
 - ▶ Факторизовать грамматику
 - ▶ Использовать аннотации (если есть)
 - ▶ Переписать грамматику
 - ▶ Использовать более одного символа предпросмотра