

Теория автоматов и формальных языков

Контекстно-свободные языки: нисходящий анализ

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

16 ноября 2021

- Контекстно-свободные грамматики (все правила вида $A \rightarrow \alpha$)
- КС языки и разрешимость проверки пустоты
- Нормальная форма Хомского
- Алгоритм СЮК

В предыдущей серии: НФХ

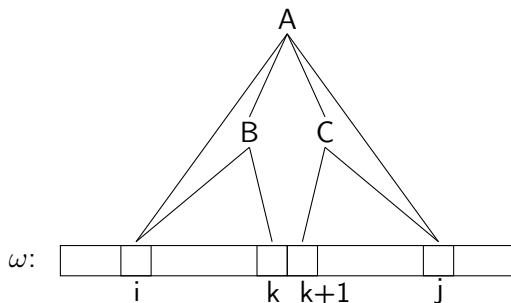
КС грамматика находится в **нормальной форме Хомского**, если все ее правила имеют вид:

- $A \rightarrow BC$, где $A, B, C \in V_N$
- $A \rightarrow a$, где $A \in V_N, a \in V_T$
- $S \rightarrow \varepsilon$, если в языке есть пустое слово, где S — стартовый нетерминал

- 1 Удалить стартовый нетерминал из правых частей правил
- 2 Избавиться от неодинокных терминалов в правых частях
- 3 Удалить длинные правила (длины больше 2)
- 4 Удалить непродуктивные правила (ε -правила)
- 5 Удалить цепные правила

В предыдущей серии: СΥΚ

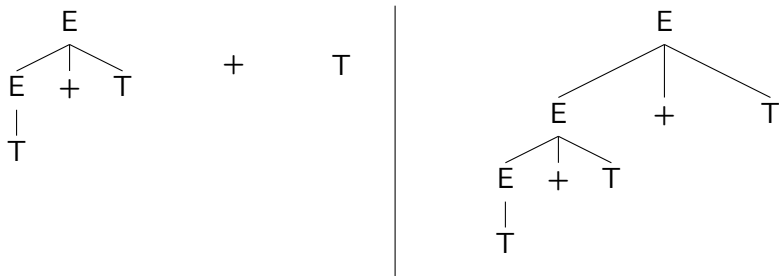
- Алгоритм синтаксического анализа, работающий с грамматиками в НФХ
- Динамическое программирование



- Дано: строка ω длины n , грамматика $G = \langle V_T, V_N, P, S \rangle$ в НФХ
- Используем трехмерный массив d булевых значений размером $|V_N| \times n \times n$, $d[A][i][j] = \text{true} \Leftrightarrow A \Rightarrow \omega[i \dots j]$
- Инициализация: $i = j$
 - ▶ $d[A][i][i] = \text{true}$, если в грамматике есть правило $A \rightarrow \omega[i]$
 - ▶ $d[A][i][i] = \text{false}$, иначе
- Динамика. Предполагаем, d построен для всех нетерминалов и пар $\{(i', j') \mid j' - i' < m\}$
 - ▶ $d[A][i][j] = \bigvee_{A \rightarrow BC} \bigvee_{k=i}^j d[B][i][k] \wedge d[C][k+1][j]$
- В конце работы алгоритма в $d[S][1][n]$ записан ответ, выводится ли ω в данной грамматике

СНК — алгоритм восходящего анализа

Восходящий анализ: начинаем с символов входной строки, строим дерево вывода до стартового нетерминала



Восходящий анализ контринтуитивен
(особенно при диагностике ошибок)

Нисходящий синтаксический анализ

- 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

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd$$

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

S

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

$$S \Rightarrow aB$$

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

$$S \Rightarrow aB \Rightarrow accd$$

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

$$S \Rightarrow aB \Rightarrow accd$$

не подходит, откатываемся

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

$$S \Rightarrow aB \Rightarrow accd$$

$$S \Rightarrow aB$$

не подходит, откатываемся

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

$$S \Rightarrow aB \Rightarrow accd$$

$$S \Rightarrow aB \Rightarrow addc$$

не подходит, откатываемся

не подходит, откатываемся

не подходит, откатываемся

Bruteforce parsing: пример

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

$$\omega = addc$$

$$S \Rightarrow aAd \Rightarrow abd$$

$$S \Rightarrow aAd \Rightarrow acd$$

$$S \Rightarrow aB \Rightarrow accd$$

$$S \Rightarrow aB \Rightarrow addc$$

не подходит, откатываемся

не подходит, откатываемся

не подходит, откатываемся

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

```
parse_S(word) =  
  if (isEmpty(word))  
  then (true, word)  
  else  
    let (r, w') = parse_lbr(word) in  
    if (r)  
    then  
      let (r, w'') = parse_S(w') in  
      if (r)  
      then parser_rbr(w'')  
      else (false, w')  
    else (false, word)
```


Нисходящий синтаксический анализ без отката: LL(1)

- Идея: откат запрещен, но разрешен предпросмотр
- По следующему терминалу 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

- 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

Управляет процессом разбора: показывает, какую продукцию применять, если во время анализа рассматривается нетерминал A , а следующий символ входа — t

| | ... | t | ... | \$ |
|-----|-----|------------------------|-----|-----|
| | ... | ... | ... | ... |
| A | ... | $A \rightarrow \alpha$ | ... | ... |
| | ... | ... | ... | ... |

Для заполнения таблицы надо научиться считать множества символов, которые можно встретить во время анализа

Множество FIRST

Множество символов, которые могут появиться первыми во время вывода из данной сентенциальной формы

- $FIRST(a\alpha) = \{a\}$, если $a \in V_T, \alpha \in (V_T \cup V_N)^*$
- $FIRST(\varepsilon) = \{\varepsilon\}$
- $FIRST(\alpha\beta) = FIRST(\alpha) \cup (FIRST(\beta)$, если $\varepsilon \in FIRST(\alpha)$)
- $FIRST(S) = FIRST(\alpha) \cup FIRST(\beta)$, если есть правило $S \rightarrow \alpha \mid \beta$

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

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

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

$$\begin{aligned} (S &\rightarrow aS') \\ (S' &\rightarrow AbBS') \end{aligned}$$

Множество FOLLOW: пример

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

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

$$(S \rightarrow aS')$$

$$(S' \rightarrow AbBS')$$

$$(A \rightarrow aA')$$

Множество FOLLOW: пример

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

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

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

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

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

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

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

LL(1)-анализ

- Инициализация: указатель в строке на первый символ, в стек помещаем \$ и стартовый нетерминал
- Пока стек не пуст
 - ▶ Если на вершине стека нетерминал N , указатель в строке на символе t , смотрим на содержимое ячейки (N, t) управляющей таблицы
 - ★ Если ячейка пуста, сообщаем об ошибке анализа
 - ★ Если в ячейке продукция $N \rightarrow \varepsilon$, снимаем со стека N
 - ★ Если в ячейке продукция $N \rightarrow \alpha$, снимаем со стека N , символы α кладем на стек в обратном порядке
 - ▶ Если на вершине стека терминал t
 - ★ Если указатель в строке на терминале t , снимаем со стека вершину, двигаем указатель на следующий символ
 - ★ Если указатель в строке на любом другом терминале, сообщаем об ошибке
- Если строка прочитана полностью, анализ завершен успешно. Иначе — полагается сообщить об ошибке

Пример (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
 - ★ $FIRST(A) \cap FOLLOW(A) \neq \emptyset$
 - ★ $S \rightarrow Aab, A \rightarrow a \mid \varepsilon$
- Как с этим бороться?
 - ▶ Избавиться от левой рекурсии
 - ▶ Избавиться от недетерминизма
 - ▶ Факторизовать грамматику
 - ▶ Использовать аннотации (если есть)
 - ▶ Переписать грамматику
 - ▶ Использовать более одного символа предпросмотра