

Теория автоматов и формальных языков

Контекстно-свободные языки: LR-анализ

Лектор: Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

07 ноября 2019

- Нисходящий анализ
- Алгоритм синтаксического анализа $LL(1)$
- $LL(k)$ грамматики и языки

Восходящий синтаксический анализ

- Начинаем с символов входной строки, строим дерево вывода до стартового нетерминала
- СҮК — один из примеров восходящего синтаксического анализа

- Left-to-right, Right-most вывод
- Разрешен предпросмотр
- Предиктивен
- Обрабатывает леворекурсивные грамматики
- Достаточно хорош для используемых на практике языков

- Используют:
 - ▶ Входной буфер (откуда читается входная строка)
 - ▶ Стек (для промежуточных данных)
 - ▶ **Таблицы** анализатора (управляет процессом разбора)
 - ★ Разные модификации используют разные таблицы
 - ★ Таблица определяет “мощность” анализатора
- Оперирует состояниями
- Работает за $O(n)$, где n — длина входной строки

Таблица LR-анализатора

Управляет процессом разбора

		...	t	...	$\$$...	B	...
...	
13		...	s_i	...	r_k		...	j	...
...	

- s_i — shift
- r_k — reduce
- j — goto
- acc — accept

- Разбирает наименьший класс языков
- Использует $LR(0)$ пункты: $A \rightarrow \alpha \cdot \beta$, где $A \rightarrow \alpha\beta$ — правило грамматики
- Множества $LR(0)$ пунктов суть состояния анализатора

Используется при вычислении множеств $LR(0)$ пунктов, которые могут быть применены на данном этапе во время синтаксического анализа

- Все пункты из I в $\text{closure}(I)$
- Если $A \rightarrow \alpha \cdot B\beta \in \text{closure}(I)$ и $B \rightarrow \gamma$ — правило грамматики, то $B \rightarrow \cdot \gamma \in \text{closure}(I)$

Пример *closure*

$$S' \rightarrow S$$

$$S \rightarrow AA$$

$$A \rightarrow aA \mid b$$

$$\text{closure}(\{S' \rightarrow \cdot S\}) = \left\{ \begin{array}{l} S' \rightarrow \cdot S \\ S \rightarrow \cdot AA \\ A \rightarrow \cdot aA \\ A \rightarrow \cdot b \end{array} \right\}$$

$$\text{closure}(\{A \rightarrow a \cdot A\}) = \left\{ \begin{array}{l} A \rightarrow a \cdot A \\ A \rightarrow \cdot aA \\ A \rightarrow \cdot b \end{array} \right\}$$

Пример *closure*

$goto(I, X)$ — передвигаем точку за символ X во всех пунктах в I

Если $A \rightarrow \alpha \cdot X\beta \in I$, добавляем $closure(\{A \rightarrow \alpha X \cdot \beta\})$ в $goto(I, X)$

Пример *goto*

$$S' \rightarrow S$$

$$S \rightarrow AA$$

$$A \rightarrow aA \mid b$$

$$I = \left\{ \begin{array}{l} S' \rightarrow \cdot S \\ S \rightarrow \cdot AA \\ A \rightarrow \cdot aA \\ A \rightarrow \cdot b \end{array} \right\}$$

$$\text{goto}(I, A) = \left\{ \begin{array}{l} S \rightarrow A \cdot A \\ A \rightarrow \cdot aA \\ A \rightarrow \cdot b \end{array} \right\}$$

- Состояния — множества пунктов
- Переходы по символам грамматики
- Начальное состояние — $\text{closure}(\{S' \rightarrow \cdot S\})$
- Следующие состояния считаются при помощи $\text{goto}(I, X)$

Пример

Пример LR(0)-автомата

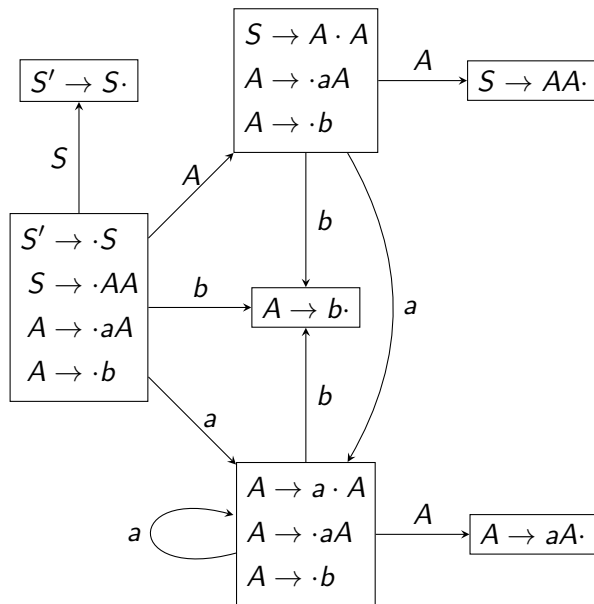


Таблица LR(0)-анализатора

- По горизонтали: состояния
- По вертикали: терминалы + \$ + нетерминалы
- acc в ячейку, соответствующую стартовому состоянию и \$
- s_i в ячейку (j, t) , если в автомате есть переход из состояния j по терминалу t в состояние i
- i в ячейку (j, N) , если в автомате есть переход из состояния j по нетерминалу N в состояние i
- r_k в ячейку (j, t) , если в состоянии j есть пункт $A \rightarrow \alpha \cdot$, где $A \rightarrow \alpha$ — k -ое правило грамматики, t — терминал грамматики

Таблица LR(0)-анализатора

	<i>a</i>	<i>b</i>	<i>§</i>	<i>A</i>	<i>S</i>
0	<i>s</i> ₃	<i>s</i> ₄		2	1
1			<i>acc</i>		
2	<i>s</i> ₃	<i>s</i> ₄		5	
3	<i>s</i> ₃	<i>s</i> ₄		6	
4	<i>r</i> ₃	<i>r</i> ₃	<i>r</i> ₃		
5	<i>r</i> ₁	<i>r</i> ₁	<i>r</i> ₁		
6	<i>r</i> ₂	<i>r</i> ₂	<i>r</i> ₂		

Таблица SLR(1)-анализатора

r_k в ячейку (j, t) , если в состоянии j есть пункт $A \rightarrow \alpha \cdot$,
где $A \rightarrow \alpha$ — k -ое правило грамматики, $t \in \text{Follow}(A)$

	a	b	$\$$	A	S
0	s_3	s_4		2	1
1			acc		
2	s_3	s_4		5	
3	s_3	s_4		6	
4	r_3	r_3	r_3		
5			r_1		
6	r_2	r_2	r_2		