Корректность алгоритма разбора LC-грамматик

Шпилевой Денис Б05-153
Руководитель - Павел Ахтямов

Мотивация

- Выявление закономерностей в алгоритмах разбора с заранее определенными параметрами - порядка разбора и известной информацией (в частном случае - аванцепочкой), с целью построить обобщение на произвольные параметры разбора.

- Данный алгоритм разбора встречается намного реже в исследованиях и публикациях, чем другие алгоритмы разбора - (например, LR).

- Отсутствие доказательства корректности в оригинальной статье.

Алгоритм построения анализатора для LC(1)-грамматики

Вход: LC(1)-грамматика $G = (N, \Sigma, P, S)$.

Выход: Корректная управляющая таблица разбора T.

1. Пусть B -> x - правило с номером i.

   а) Если x = Cy, где C - нетерминал, то T([A, C], a) = (y[A, B], i) для всех A є N, x є FIRST1(yde), таких, что S => wAe и A => Bd. x, y, d, e є (N ∪ Σ)*, a є N.

   б) Если x не начинается нетерминалом, то T(A, a) = (x[A, B], i) для всех A є N, x є FIRSTk(xde), таких, что S => wAe и A => Bd. x, d, e є (N ∪ Σ)*, a є N.

2. T([A, A], a) = (ε, ε), для всех A є N, a є FIRSTk(e), таких, что S => wAe.
3. T(a, av) = **выброс**, для всех a є Σ.
4. T($, ε) = **допуск.**
5. Иначе T(X, a) = **ошибка.**

Алгоритм разбора при помощи анализатора

1. Если T (X, a) = (β, i), где $X \in N \cup (N \times N)$, то будем писать (aω, Xα$, π) ⊢ (aω, βα$, πi);
2. Если T (a, a) = выброс, то будем писать (aω, aα$, π) ⊢ (ω, α$, π);
3. Будем говорить, что π - разбор слова ω, если (ω, S, ε) ⊢ (ε, $, π) , (ω, S$, ε) - стартовая, а (ε, $, π) - завершающая конфигурация;
4. В случае, если завершающая конфигурация недостижима из стартовой, алгоритм при первом достижении ошибочной конфигурации выводит ошибку, что означает, что сло-во не лежит в языке.

Пример работы алгоритма

По LC(1)-грамматике $G = (N, \Sigma, P, S)$ построим анализатор, представленный в виде таблицы, и разберем по ней некоторое слово.

N = {A, B, S} Σ = {+, *, <, >, a}

P состоит из 6 правил:

1. S -> S + A
2. S -> A
3. A -> A * B
4. A -> B
5. B -> < S >
6. B -> a

Пример работы алгоритма

магазинный символ	входной символ a	<	>	+	*	e
S	a[S, B], 6	<S>[S, B], 5				
A	a[A, B], 6	<S>[A, B], 5				
B	a[B, B], 6	<S>[B, B], 5				
[S, S]			e, e	+A[S, S], 1		e, e
[S, A]			[S, S], 2	[S, S], 2	*B[S, A], 3	[S, S], 2
[S, B]			[S, A], 4	[S, A], 4	[S, A], 4	[S, A], 4
[A, A]			e, e	e, e	*B[A, A], 3	e, e
[A, B]			[A, A], 4	[A, A], 4	[A, A], 4	[A, A], 4
[B, B]			e, e	e, e	e, e	e, e
a	выброс					
<		выброс				
>			выброс			
+				выброс		
*					выброс	
$						допуск

Возьмём слово w = < a * a >

(<a*a>, S$, ε) ⊢ (<a*a>, <S>[S, B]$, 5) ⊢

(a*a>, S>[S, B]$, 5) ⊢ (a*a>, a[S, B]>[S, B]$, 56) ⊢

(*a>, [S, B]>[S, B]$, 56) ⊢ (*a>, [S, A]>[S, B]$, 564) ⊢

(*a>, *B[S, A]>[S, B]$, 5643) ⊢ (a>, [S, A]>[S, B]$, 5643) ⊢

(a>,a[B, B][S,A]>[S,B]$, 56436) ⊢ (>, [B, B][S, A]>[S, B]$, 56436) ⊢

(>, [S, A]>[S, B]$, 56436) ⊢ (>, [S, S]>[S, B]$, 564362) ⊢

(>, >[S, B]$, 564362) ⊢ (ε, [S, B]$, 564362) ⊢

(ε, [S, A]$, 5643624) ⊢ (ε, [S, S]$, 56436242) ⊢

(ε, $, 56436242)

Результат работы

Доказана корректность работы алгоритма построения анализатора для LC(1)-грамматик.

Доказательство проводилось при помощи вспомогательной леммы:

$$(xy, \alpha\gamma, \pi_1) \vdash^* (y, \beta\gamma, \pi_1\pi_2) \Leftrightarrow \alpha \ {}_{\pi_2}\!\Rightarrow^*_{lc} x\beta$$

Данная формулировка показывает корректность работы на каждом шаге алгоритма разбора, частным случаем которого является сама теорема:

$$(\omega, S\$, \varepsilon) \vdash^* (\varepsilon, \$, \pi) \Leftrightarrow S \ {}_{\pi}\!\Rightarrow^*_{lc} \omega.$$

Идея доказательства леммы

В прямую сторону.

1. Задаем гомоморфизм на множестве состояний магазина для того, чтобы избавится от символов, не представленных в множестве нетерминалов.
2. Доказательство индукцией по количеству шагов алгоритма.
3. Разбор всех возможных случаев перехода между конфигурациями с поправкой на алгоритм построения анализатора.

В обратную сторону доказываем индукцией по количеству шагов вывода.

Источники и литература

- Ахо А. Ульман Дж., Теория синтаксического анализа, перевода и компиляции, том 1, Синтаксический анализ.
- 11th Annual Symposium on Switching and Automata Theory (swat 1970), deterministic left corner parsing, pages 139-152.