

Системы программирования.

лекции 04.04.2011

Синтаксический анализ. Метод рекурсивного спуска.





Этапы трансляции

- лексический анализ
- **синтаксический анализ**
- семантический анализ
- генерация внутреннего представления программы
- оптимизация
- генерация объектной программы



Синтаксический анализ

Входные данные: последовательность лексем

Задачи :

- Установить, имеет ли цепочка лексем структуру, заданную синтаксисом языка.
- Зафиксировать эту структуру.

Используемые грамматики: контекстно-свободные

Корректный анализатор

- завершает работу для любой входной цепочки
- выдает верный ответ о принадлежности цепочки языку

Какие ошибки в программе может обнаружить синтаксический анализатор?

Метод рекурсивного спуска

Пример 1.

$S \rightarrow ABd$

$A \rightarrow a \mid cA$

$B \rightarrow b$

Вопрос : порождает ли грамматика цепочку `cabd` ?


Каждому нетерминалу сопоставим процедуру, носящую его имя. Задача процедуры: начиная с указанного места исходной цепочки найти подцепочку, которая выводится из соответствующего нетерминала. Если такой цепочки не нашлось — зафиксировать ошибку.

Соглашения

`s` - текущая анализируемая лексема.

Перед входом в каждую из процедур в переменной `s` хранится лексема, которую еще не анализировали.

Если процедура РС находит ошибку, она генерирует исключение.⁴



Достаточное условие применимости метода рекурсивного спуска

Для применимости метода рекурсивного спуска достаточно, чтобы каждое правило в грамматике имело вид:

(а) либо $X \rightarrow \alpha$,

где $\alpha \in (T \cup N)^*$ и это единственное правило вывода для этого нетерминала;

(б) либо $X \rightarrow a_1\alpha_1 \mid a_2\alpha_2 \mid \dots \mid a_n\alpha_n$,

где $a_i \in T$ для всех $i = 1, 2, \dots, n$; $a_i \neq a_j$ для $i \neq j$; $\alpha_i \in (T \cup N)^*$,

т. е. если для нетерминала X правил вывода несколько, то они должны начинаться с терминалов, причем все эти терминалы должны быть различными.

Это условие не является необходимым.

РС-анализатор для грамматики с итерациями и ϵ - правилами

Вопрос:

В чем отличие функций анализатора для правил $V \rightarrow b$ и $V \rightarrow b \mid \epsilon$?

Обозначение итерации

$S \rightarrow a \{bV\}$ эквивалентно $S \rightarrow aN$
 $N \rightarrow bVN \mid \epsilon$

Пример 2:

$E \rightarrow T \{ [+ \mid -] T \}$

$T \rightarrow F \{ [* \mid /] F \}$

$F \rightarrow 0 \mid 1 \mid \dots \mid 9 \mid - F \mid (E)$

РС-анализатор для грамматики с итерациями и ε - правилами (продолжение)

Пример 3:

$S \rightarrow Ab$
 $A \rightarrow c\{bb\}$

$S \rightarrow Ab$
 $A \rightarrow cN$
 $N \rightarrow bbN \mid \varepsilon$

Пример 4:

$S \rightarrow Ac$
 $A \rightarrow c\{bb\}$

$S \rightarrow Ac$
 $A \rightarrow cN$
 $N \rightarrow bbN \mid \varepsilon$

РС-анализатор для грамматики с итерациями и ε -правилами (продолжение)

Множество $first(\alpha)$ — это множество терминальных символов, которыми начинаются цепочки, выводимые из цепочки α в грамматике $G = \langle T, N, P, S \rangle$, т. е.

$$first(\alpha) = \{ a \in T \mid \alpha \Rightarrow a\alpha', \text{ где } \alpha \in (T \cup N)^+, \alpha' \in (T \cup N)^* \}$$

Множество $follow(A)$ — это множество терминальных символов, которые следуют за цепочками, выводимыми из A в грамматике $G = \langle T, N, P, S \rangle$, т. е.

$$follow(A) = \{ a \in T \mid S \Rightarrow \alpha A \beta, \beta \Rightarrow a \gamma, A \in N, \alpha, \beta, \gamma \in (T \cup N)^* \}$$

Если в грамматике есть пара правил $X \rightarrow \alpha \mid \beta$, таких что $\beta \Rightarrow \varepsilon$,

$$first(\alpha) \cap follow(X) \neq \emptyset,$$

то метод рекурсивного спуска неприменим к данной грамматике.



Канонический вид грамматики для метода РС

Достаточное условие применимости. Такой вид удобен для построения процедур анализатора (но необходимым условием не является)

(1) либо $X \rightarrow \alpha$,

где $\alpha \in (T \cup N)^*$ и это единственное правило вывода для этого нетерминала;

(2) либо $X \rightarrow a_1\alpha_1 \mid a_2\alpha_2 \mid \dots \mid a_n\alpha_n$,

где $a_i \in T$ для всех $i = 1, 2, \dots, n$; $a_i \neq a_j$ для $i \neq j$;

$\alpha_i \in (T \cup N)^*$, т. е. если для нетерминала X правил вывода несколько, то они должны начинаться с терминалов, причем все эти терминалы должны быть попарно различными;

(3) либо $X \rightarrow a_1\alpha_1 \mid a_2\alpha_2 \mid \dots \mid a_n\alpha_n \mid \varepsilon$,

где $a_i \in T$ для всех $i = 1, 2, \dots, n$; $a_i \neq a_j$ для $i \neq j$; $\alpha_i \in (T \cup N)^*$,

и $first(X) \cap follow(X) = \emptyset$.

Применимость РС

Пример 5

$S \rightarrow A \mid B$

$A \rightarrow aA \mid d$

$B \rightarrow aB \mid b$

Пример 6

$S \rightarrow aA \mid BDc$

$A \rightarrow BAa \mid aB \mid b$

$B \rightarrow \varepsilon$

$D \rightarrow B \mid b$

Наличие в грамматике правил вида $X \rightarrow \alpha \mid \beta$, таких что $first(\alpha) \cap first(\beta) \neq \emptyset$, делает метод рекурсивного спуска неприменимым.



Критерий применимости РС

Пусть G — КС-грамматика.

Метод рекурсивного спуска применим к G , если и только если для любой пары альтернатив вида

$$X \rightarrow \alpha \mid \beta$$

выполняются следующие условия:

(1) $\text{first}(\alpha) \cap \text{first}(\beta) = \emptyset$;

(2) справедливо не более чем одно из двух соотношений:

$$\alpha \Rightarrow \varepsilon, \beta \Rightarrow \varepsilon ;$$

(3) если $\beta \Rightarrow \varepsilon$, то $\text{first}(\alpha) \cap \text{follow}(X) = \emptyset$.



Применимость метода РС

Если обнаружено, что метод РС не применим, можно ли написать эквивалентную грамматику, допускающую анализ методом РС?

В общем случае задача алгоритмически неразрешима.



Преобразования грамматик для применения РС

Все перечисленные преобразования являются эквивалентными

1. Удаление левой рекурсии
2. Объединение альтернатив с одинаковыми началами
3. Подстановка
4. Решение проблемы с пересечением first и follow

Преобразования грамматик для применения РС

Все перечисленные преобразования являются эквивалентными

1. Удаление левой рекурсии

2. Объединение альтернатив с одинаковыми началами

3. Подстановка

4. Решение проблемы с пересечением first и follow

Если в грамматике есть правила

$$A \rightarrow \alpha_1 A \mid \dots \mid \alpha_n A \mid \beta_1 \mid \dots \mid \beta_m \mid \varepsilon$$

$$B \rightarrow \alpha A \beta$$

и $FIRST(A) \cap FOLLOW(A) \neq \emptyset$ (из-за вхождения A в правило вывода для B), то можно преобразовать их в такие:

$$B \rightarrow \alpha A'$$

$$A' \rightarrow \alpha_1 A' \mid \dots \mid \alpha_n A' \mid \beta_1 \beta \mid \dots \mid \beta_m \beta \mid \beta$$



Синтаксический анализ по неоднозначным грамматикам

Пример из грамматики языка Паскаль

$S \rightarrow \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S \mid I := E$



Грамматика с действиями

Задача:

Описать с помощью грамматики язык

$L = \{\text{любые цепочки из } 0 \text{ и } 1 \text{ с неравным количеством } 0 \text{ и } 1\}$

Написать по грамматике анализатор.