

Рекурсивный спуск

В. С. Полозов

Кафедра системного программирования СПбГУ



Теория автоматов и формальных языков

Рекурсивный спуск

Возьмём правило для нетерминала A :

$$A \rightarrow \alpha_1 | \dots | \alpha_n$$

Определим:

```
procedure A;  
  if lookahead  $\in$  FIRST( $\alpha_1$ FOLLOW(A)) then  
    code for  $\alpha_1$  ...  
  else if lookahead  $\in$  FIRST( $\alpha_2$ FOLLOW(A)) then  
    code for  $\alpha_2$  ...  
  .  
  .  
  else if lookahead  $\in$  FIRST( $\alpha_n$ FOLLOW(A)) then  
    code for  $\alpha_n$  ...  
  else ERROR;  
endA;
```

Рекурсивный спуск

Где код для продукции состоит из последовательности вызовов процедур соответствующих нетерминалу, или MATCH для терминального символа:

```
procedure MATCH(sym);  
    if lookahead = sym then  
        lookahead := NEXTSYM  
    else ERROR;  
end MATCH;
```

Почему рекурсивный спуск?

- Легко встроить семантические действия в код.
- Параметризация и атрибуты получаются практически "сами по себе".
- Рекурсивный спуск без возвратов довольно эффективен. Часто – даже эффективнее табличных методов.
- Проще реализовать динамическое разрешение конфликтов.
- Основной минус - размер.

Расширенные $LL(1)$ грамматики

Аналогично введению расширений для CF -грамматик, можно ввести расширения и для $LL(1)$ -грамматик, устранив $LL(1)$ -конфликты. Например,

- $A^+ \rightarrow A \mid AA^+$

приводит к $LL(1)$ -конфликту по $FIRST(A)$. Положим:

- $A^* \rightarrow \varepsilon \mid AA^*$

- $A^+ \rightarrow AA^*$

- $A^? \rightarrow \varepsilon \mid A$

Проекции соответствуют циклам **while**, **repeat** и условному оператору **if**, соответственно.

Примеры разбора рекурсивным спуском

Возьмём грамматику арифметических выражений,

$$\begin{array}{lll} E & \rightarrow & T \quad \{T\} \\ E & \rightarrow & E + F \quad \{E + F\} \\ E & \rightarrow & E - F \quad \{E - F\} \\ F & \rightarrow & F \quad \{F\} \\ F & \rightarrow & F * T \quad \{F * T\} \\ F & \rightarrow & F / T \quad \{F / T\} \\ T & \rightarrow & \textit{number} \quad \{\textit{number}\} \\ T & \rightarrow & '(E)' \quad \{E\} \end{array}$$

где код в фигурных скобках обозначает возвращаемое значение соответствующей функции рекурсивного спуска, а символы в них - ссылки на соответствующие результаты вызовов функций рекурсивного спуска.

Примеры L-атрибутного разбора рекурсивным спуском

Приведём грамматику к $LL(1)$ -виду, устранив левую рекурсию:

$$E \rightarrow TE' \{E'(T)\}$$

$$E'(v) \rightarrow \varepsilon \{v\}$$

$$E'(v) \rightarrow +TE' \{E'(v+T)\}$$

$$E'(v) \rightarrow -TE' \{E'(v-T)\}$$

...

где " $E'(v) \rightarrow \dots \{...v...\}$ " обозначает дополнительный аргумент v функции E' рекурсивного спуска, фактическое значение которого может быть использовано в action-коде $\{...v...\}$.

Вопросы