

Классификация грамматики

Согласно классификации Хомского, грамматика $G[Expr]$ является конетекстно-свободной.

S-грамматики – это подкласс контекстно-свободных грамматик, таких что:

1. Правая часть каждого правила начинается с терминала $A \rightarrow a$
2. Если в грамматике есть альтернативные правила, то они обязательно начинаются с разных терминальных символов

$A \rightarrow a \mid ba, a \neq b$

1. $S \rightarrow \langle Expr \rangle$
2. $ORExpr \rightarrow \langle ORExpr \rangle or \langle AndExpr \rangle \mid \langle AndExpr \rangle$
3. $AndExpr \rightarrow \langle AndExpr \rangle and \langle NotExpr \rangle \mid \langle NotExpr \rangle$
4. $NotExpr \rightarrow \langle NotExpr \rangle \mid \langle RelExpr \rangle$
5. $RelExpr \rightarrow \langle AddExpr \rangle (RelOp \ AddExpr)^*$
6. $AddExpr \rightarrow \langle MulExpr \rangle (AddOp \ MulExpr)^*$
7. $MulExpr \rightarrow \langle Unary \rangle (MulOp \ UnaryExpr)^*$
8. $UnaryExpr \rightarrow \langle Factor \rangle \mid (AddOp \ UnaryExpr)$
9. $Factor \rightarrow \langle Identifier \rangle \mid Const \mid \langle \text{`Expr`} \rangle$

$\langle Digit \rangle \rightarrow "0" \mid "1" \mid "2" \mid "3" \mid "4" \mid "5" \mid "6" \mid "7" \mid "8" \mid "9"$

$\langle Letter \rangle \rightarrow "A" \mid "B" \mid "C" \mid \dots \mid "Z" \mid "a" \mid "b" \mid "c" \mid \dots \mid "z"$

ледует заметить, что данная грамматика реализуется с помощью метода рекурсивного спуска. Основная идея этого метода состоит в том, что каждому нетерминалу грамматики ставится в соответствие определенная программная единица, процедура или функция, которая распознает цепочку, порождаемую этим нетерминалом

ледует заметить, что данная грамматика реализуется с помощью метода рекурсивного спуска. Основная идея этого метода состоит в том, что каждому нетерминалу грамматики ставится в соответствие определенная

программная единица, процедура

программная единица, процедура или функция, которая распознает цепочку, порождаемую этим нетерминалом

Данная грамматика реализуется с помощью метода рекурсивного спуска.

Основная идея этого метода состоит в том, что каждому нетерминалу грамматики ставится в соответствие определенная программная единица, процедура или функция, которая распознает цепочку, порождаемую этим нетерминалом.