

КС-грамматики

Разбор цепочки - процесс построения вывода цепочки α из цели S грамматики $G = (T, N, P, S)$.

Вывод цепочки $\beta \in T^*$ из $S \in N$ в КС-грамматике $G = (T, N, P, S)$, называется:

- **левосторонним**, если в нем каждая очередная сентенциальная форма получается из предыдущей заменой самого левого нетерминала.

- **правосторонним**, если в нем каждая очередная сентенциальная форма получается из предыдущей заменой самого правого нетерминала.

Например, для цепочки $a+b+a$ в грамматике

$$G = (\{a, b, +\}, \{ S, T \}, \{ S \rightarrow T \mid T+S; T \rightarrow a \mid b \}, S)$$

можно построить выводы:

(1) $S \rightarrow T+S \rightarrow T+T+S \rightarrow T+T+T \rightarrow a+T+T \rightarrow a+b+T \rightarrow a+b+a$ - произвольный

(2) $S \rightarrow T+S \rightarrow a+S \rightarrow a+T+S \rightarrow a+b+S \rightarrow a+b+T \rightarrow a+b+a$ - левый

(3) $S \rightarrow T+S \rightarrow T+T+S \rightarrow T+T+T \rightarrow T+T+a \rightarrow T+b+a \rightarrow a+b+a$ - правый

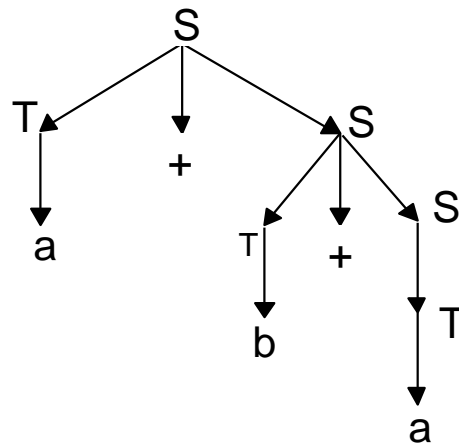
Выводы (1) – (3) являются **эквивалентными** в том смысле, что в них в одних и тех же местах применяются одни и те же правила вывода, но в различном порядке.

Дерево вывода

Дерево вывода (или **дерево разбора**) в КС-грамматике $G = (T, N, P, S)$ – дерево, для которого выполнены следующие условия:

- (1) дерево ориентировано и упорядочено;
- (2) каждая вершина дерева помечена символом из множества $N \cup T \cup \{\varepsilon\}$, при этом корень дерева помечен символом S ; листья - символами из $T \cup \{\varepsilon\}$;
- (3) если вершина дерева помечена символом $A \in N$, а ее непосредственные потомки - символами a_1, a_2, \dots, a_n , где каждое $a_i \in T \cup N$, то $A \rightarrow a_1 a_2 \dots a_n$ - правило вывода в этой грамматике;
- (4) если вершина дерева помечена символом $A \in N$, а ее единственный непосредственный потомок помечен символом ε , то $A \rightarrow \varepsilon$ - правило вывода в этой грамматике.

Пример дерева вывода для цепочки $a + b + a$ в грамматике G :



Дерево вывода можно строить **нисходящим** либо **восходящим** способом.

Неоднозначность грамматик

КС-грамматика G неоднозначная, если существует **хотя бы одна** цепочка $\alpha \in L(G)$, для которой может быть построено два или более различных деревьев вывода.

В противном случае грамматика является **однозначной**.

Если грамматика однозначная, то при любом способе построения, нисходящем или восходящем, будет получено одно и то же дерево разбора.

Пример неоднозначной грамматики:

$G_{\text{if}} = (\{ \text{if, then, else, a, b} \}, \{ S \}, P, S)$,

где $P = \{ S \rightarrow \text{if } b \text{ then } S \text{ else } S \mid \text{if } b \text{ then } S \mid a \}$.

В этой грамматике для цепочки

if b then if b then a else a

можно построить два различных дерева вывода.

Неоднозначность грамматик

Неоднозначность - это свойство грамматики, а не языка.

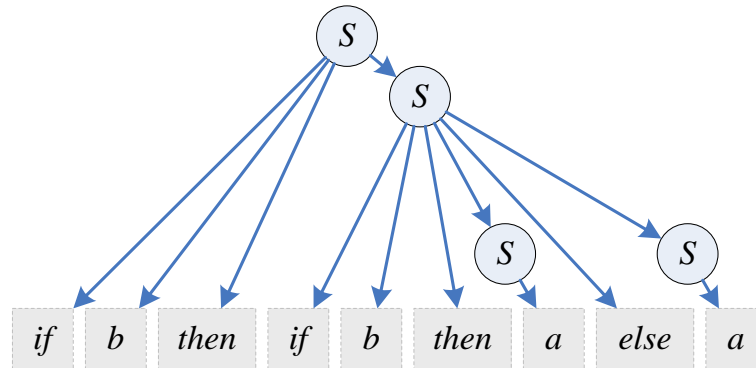
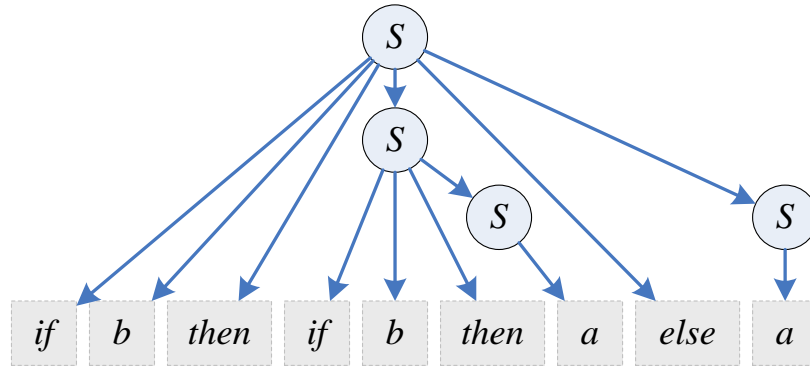
Если грамматика используется для определения языка программирования, то она должна быть однозначной.

Можно преобразовать грамматику G_{if} , устранив неоднозначность:

$$S \rightarrow \text{if } b \text{ then } S \mid T$$
$$T \rightarrow \text{if } b \text{ then } T \text{ else } S \mid a$$

Проблема определения, является ли заданная КС-грамматика однозначной, является **алгоритмически неразрешимой**.

Деревья вывода для цепочки *if b then if b then a else a* в грамматике G_{if}



Грамматика G_{if} неоднозначна, однако, это **не** означает, что язык $L(G_{if})$ неоднозначный.

Преобразование неоднозначных грамматик

Некоторые виды правил вывода, которые приводят к неоднозначности и некоторые способы эквивалентных преобразований неоднозначных грамматик к однозначным:

$$1. A \rightarrow AA \mid \alpha \quad \Rightarrow \quad A \rightarrow \alpha A \mid \alpha$$

(док-во для $\alpha\alpha\alpha$) – порождаются подцепочки α^n ($n \geq 1$);

$$2. A \rightarrow A\alpha A \mid \beta \quad \Rightarrow \quad A \rightarrow \beta\alpha A \mid \beta$$

(док-во для $\beta\alpha\beta\alpha\beta$) – порождаются подцепочки $\beta(\alpha\beta)^n$ ($n \geq 0$);

$$3. A \rightarrow \alpha A \mid A\beta \mid \gamma \quad \Rightarrow \quad A \rightarrow \alpha A \mid B;$$
$$B \rightarrow B\beta \mid \gamma, \quad B \notin N$$

(док-во для $\alpha\gamma\beta$) – порождаются подцепочки $\alpha^n \gamma \beta^m$ ($n, m \geq 0$);

$$4. A \rightarrow \alpha A \mid \alpha A\beta A \mid \gamma \Rightarrow \quad A \rightarrow \alpha A \mid B;$$
$$B \rightarrow \alpha B\beta A \mid \gamma, \quad B \notin N$$

(док-во для $\alpha\alpha\gamma\beta\gamma$) – порождаются подцепочки $\delta = \alpha^n \alpha^m \gamma (\beta\delta)^m$ ($n, m \geq 0$)

Таким приемом преобразована грамматика G_{if} :

($\alpha \equiv \text{if_b_then}$, $\beta \equiv \text{else}$, $a \equiv \gamma$, $A \equiv S$, $B \equiv T$).

Неоднозначные языки

Язык, порождаемый грамматикой **неоднозначный**, если он не может быть порожден никакой однозначной грамматикой.

Проблема определения, порождает ли данная КС-грамматика **однозначный язык** (т.е. существует ли эквивалентная ей однозначная грамматика), является **алгоритмически неразрешимой**.

Пример неоднозначного КС-языка:

$$L = \{a^i b^j c^k \mid i = j \text{ или } j = k\} .$$

Одна из грамматик, порождающих L , такова:

$$S \rightarrow AB \mid DC \quad (\text{док-во для цепочки } abc)$$

$$A \rightarrow aA \mid \varepsilon$$

$$B \rightarrow bBc \mid \varepsilon$$

$$C \rightarrow cC \mid \varepsilon$$

$$D \rightarrow aDb \mid \varepsilon$$

Бесплодные символы грамматики.

Символ $A \in N$ является **бесплодным** в грамматике $G = (T, N, P, S)$, если множество $\{ \alpha \in T^* \mid A \Rightarrow \alpha \}$ пусто.

Алгоритм удаления бесплодных символов:

Вход: КС-грамматика $G = (T, N, P, S)$,

Выход: КС-грамматика $G' = (T, N', P', S)$, не содержащая бесплодных символов, для которой $L(G) = L(G')$.

Метод:

Строим множества N_0, N_1, \dots

1. $N_0 := \emptyset$; $i := 1$.

2. $N_i := N_{i-1} \cup \{ A \mid A \rightarrow \alpha \in P \text{ и } \alpha \in (T \cup N_{i-1})^* \}$.

Если $N_i \neq N_{i-1}$, то $i := i + 1$ и переходим к шагу 2, иначе $N' := N_i$; P' состоит из правил множества P , содержащих только символы из $N_i \cup T$; $G' := (T, N', P', S)$.

Удаление бесплодных символов грамматики.

Пример.

$S \rightarrow AC \mid Bb \mid \varepsilon$

$A \rightarrow aCb$

$B \rightarrow bB$

$C \rightarrow cCc \mid c$

$D \rightarrow Aa \mid Bb \mid d$

Шаг 0: $N_0 := \emptyset; i := 1.$

Шаг 1: $N_1 := \{S, C, D\}; i := 2.$

Шаг 2: $N_2 := \{S, C, D, A\}; i := 3.$

Шаг 3: $N_3 := \{S, C, D, A\} = N_2$, т.е. искомое множество построено.

Удаляем все правила, содержащие нетерминал B , не вошедший в построенное множество:

$S \rightarrow AC \mid \varepsilon$

$A \rightarrow aCb$

$C \rightarrow cCc \mid c$

$D \rightarrow Aa \mid d$

Недостижимые символы грамматики

Символ $x \in (T \cup N)$ является **недостижимым** в грамматике $G = (T, N, P, S)$, если он не появляется ни в одной сентенциальной форме этой грамматики.

Алгоритм удаления недостижимых символов:

Вход: КС-грамматика $G = (T, N, P, S)$,

Выход: КС-грамматика $G' = (T', N', P', S)$, не содержащая недостижимых символов, для которой $L(G) = L(G')$.

Метод:

Строим множества V_0, V_1, \dots

1. $V_0 := \{S\}; i := 1.$

2. $V_i := V_{i-1} \cup \{x \mid x \in T \cup N, A \rightarrow \alpha x \beta \in P, A \in V_{i-1}, \alpha, \beta \in (T \cup N)^*\}.$

Если $V_i \neq V_{i-1}$, то $i := i + 1$ и переходим к шагу 2,

иначе $N' := V_i \cap N; T' := V_i \cap T; P'$ состоит из правил множества P , содержащих только символы из $V_i; G' := (T', N', P', S).$

Удаление недостижимых символов грамматики.

Пример.

$S \rightarrow AC \mid \varepsilon$

$A \rightarrow aCb$

$C \rightarrow cCc \mid c$

$D \rightarrow Aa \mid d$

Шаг 0: $V_0 := \{S\}; i := 1.$

Шаг 1: $V_1 := \{S, A, C, \varepsilon\}; i := 2.$

Шаг 2: $V_2 := \{S, A, C, \varepsilon, a, b, c\}; i := 3.$

Шаг 3: $V_3 := \{S, A, C, \varepsilon, a, b, c\} = V_2,$ т.е. искомое множество построено

Удаляем все правила, содержащие символы D и d, не вошедшие в построенное множество:

$S \rightarrow AC \mid \varepsilon$

$A \rightarrow aCb$

$C \rightarrow cCc \mid c$

Приведенные грамматики

Недостижимые и бесплодные символы в грамматике $G = (T, N, P, S)$ называются **бесполезными** символами в этой грамматике.

КС-грамматика G называется **приведенной**, если в ней нет бесполезных символов.

Алгоритм приведения грамматики:

- 1) обнаруживаются и удаляются все **бесплодные** нетерминалы.
- 2) обнаруживаются и удаляются все **недостижимые** символы.

Удаление символов сопровождается удалением правил вывода, содержащих эти символы.

Если в алгоритме переставить шаги 1) и 2), то не всегда результатом будет приведенная грамматика. Например, при такой перестановке шагов грамматика

$$S \rightarrow AB \mid a$$
$$A \rightarrow b$$
$$B \rightarrow BA$$

останется неприведенной.

Алгоритм устранения правил с пустой правой частью

Вход: КС-грамматика $G = (T, N, P, S)$.

Выход: КС-грамматика $G' = (T, N', P', S')$ - неукорачивающая, $L(G') = L(G)$.

Метод:

1. Построить множество $X = \{A \in N \mid A \Rightarrow \varepsilon\}$; $N' := N$.
2. Построить P' , удалив из множества правил P все правила с пустой правой частью.
3. Если $S \in X$, то ввести новый начальный символ S' , добавив его в N' , и в множество правил P' добавить правило $S' \rightarrow S \mid \varepsilon$.
Иначе просто переименовать S в S' .
4. Изменить P' следующим образом. Каждое правило вида $B \rightarrow \alpha_1 A_1 \alpha_2 A_2 \dots \alpha_n A_n \alpha_{n+1}$, где $A_i \in X$ для $i = 1, \dots, n$, $\alpha_j \in ((N' - X) \cup T)^*$ для $j = 1, \dots, n + 1$ (т. е. α_j — цепочка, не содержащая символов из X), заменить 2^n правилами, соответствующими всем возможным комбинациям вхождений A_i между α_j :

$$B \rightarrow \alpha_1 \alpha_2 \dots \alpha_n \alpha_{n+1}$$

$$B \rightarrow \alpha_1 \alpha_2 \dots \alpha_n A_n \alpha_{n+1}$$

...

$$B \rightarrow \alpha_1 \alpha_2 A_2 \dots \alpha_n A_n \alpha_{n+1}$$

$$B \rightarrow \alpha_1 A_1 \alpha_2 A_2 \dots \alpha_n A_n \alpha_{n+1}$$

Если $\alpha_j = \varepsilon$ для всех $i = 1, \dots, n + 1$, то получившееся на данном шаге правило $B \rightarrow \varepsilon$ не включать в множество P' .

5. Удалить бесполезные символы и правила, их содержащие.

Устранение правил с пустой правой частью.

Пример.

$$S \rightarrow BC \mid Ab \mid AB$$

$$A \rightarrow Aa \mid \varepsilon$$

$$B \rightarrow \varepsilon$$

$$C \rightarrow c$$

Шаг 1: $N_1 := \{ A, B \};$

Шаг 2: $N_2 := \{ A, B, S \};$

$$S1 \rightarrow S \mid \varepsilon$$

$$S \rightarrow BC \mid C \mid Ab \mid b \mid AB \mid A \mid B$$

$$A \rightarrow Aa \mid a$$

$$C \rightarrow c$$

Приводим грамматику:

$$S1 \rightarrow S \mid \varepsilon$$

$$S \rightarrow C \mid Ab \mid b \mid A$$

$$A \rightarrow Aa \mid a$$

$$C \rightarrow c$$