

Лабораторная работа №4

Устранить из КС-грамматики бесполезные символы и эпсилон-правила

Исходная грамматика

$$P = \{ S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon \}$$

$$G = (\{b, c\}, \{A, B, S\}, P, S)$$

Удаление непроезводящих символов.

$$\text{Шаг 1. } V_p^0 = \emptyset$$

$$\text{Шаг 2. } V_p^1 = \{S, B\}$$

$$\text{Шаг 3. } V_p^2 = \{S, B\}, V_p^2 = V_p^1$$

$$P^* = \{S \rightarrow b, B \rightarrow cB, B \rightarrow \varepsilon\}, G^* = (\{b, c\}, \{S, B\}, P^*, S)$$

Удаление недостижимых символов.

$$\text{Шаг 1. } VT_r^0 = \{S\}$$

$$\text{Шаг 2. } VT_r^1 = \{S, b\}$$

$$\text{Шаг 3. } VT_r^2 = \{S, b\}, VT_r^1 = VT_r^2$$

$$VT_r = \{S, b\}, T' = T \cap VT_r = \{b\}, V' = V \cap VT_r = \{S\}$$

$$P' = \{S \rightarrow b\}, G' = (\{b\}, \{S\}, P', S)$$

Построение множества укорачивающих символов.

Возьму исходную грамматику $G = (\{b, c\}, \{A, B, S\}, P, S)$

$$P = \{S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon\}$$

$$\text{Шаг 1. } V_\varepsilon^0 = \emptyset$$

$$\text{Шаг 2. } V_\varepsilon^1 = \{B\}$$

$$\text{Шаг 3. } V_\varepsilon^2 = \{B\}, V_\varepsilon^1 = V_\varepsilon^2$$

Удаление эпсилон-правил.

Эпсилон-правило: $B \rightarrow \varepsilon$

Заменим правила так, чтобы получить эквивалентную грамматику. Получим новые правила: $S \rightarrow cA$ и $B \rightarrow c$.

$$P' = \{S \rightarrow b, S \rightarrow cAB, S \rightarrow cA, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow c\}$$

$$G = (\{b, c\}, \{A, B, S\}, P', S)$$

Лабораторная работа №5

Устранить из КС-грамматики цепные правила и устранить левую рекурсию

Исходная грамматика

$$P = \{S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon\}$$

$$G = (\{b, c\}, \{A, B, S\}, P, S)$$

Удаление цепных правил. В моей грамматике нет цепных, добавлю правило $A \rightarrow B$

$$\tilde{P} = \{S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon, A \rightarrow B\}$$

$$V_S = \{S\}, V_A = \{A\}, V_B = \{B\}$$

Шаг 1. $V_S = \{S\}$ и $V_B = \{B\}$ не изменяются, $V_A = \{A\}$ изменится из-за $A \rightarrow B$

Получим $V_A = \{A, B\}$

Шаг 2. Цепных правил не осталось

$$V_A = \{A, B\}, \text{ добавим правил } A \rightarrow cB, A \rightarrow \varepsilon, \text{ удалим } A \rightarrow B.$$

Получим: $\tilde{P}' = \{S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon, A \rightarrow cB, A \rightarrow \varepsilon\}$

Удаление левой рекурсии

Возьмём правила \tilde{P}' , но без эpsilon-правил:

$$\tilde{\tilde{P}} = \{S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, A \rightarrow cB\}$$

Упорядочим нетерминалы V по возрастанию $V = \{A, B, S\}$

Шаг 1. $i=1$ $A \rightarrow Ab | cA | cB$

Добавим A' : $A \rightarrow b | bA', A' \rightarrow cA | cB | cAA' | cBA'$

Шаг 2. Больше левой рекурсии нет

После удаления левой рекурсии получим

$$\tilde{\tilde{P}}' = \{S \rightarrow b, S \rightarrow cAB, A \rightarrow cA, B \rightarrow cB, A \rightarrow cB, A' \rightarrow cA, A' \rightarrow cB, A' \rightarrow cAA', A' \rightarrow cBA', A \rightarrow b, A \rightarrow bA'\}$$

Лабораторная работа №6

Исходная грамматика

$$P = \{ S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon \}$$

$$G = (\{b, c\}, \{A, B, S\}, P, S)$$

Вывод цепочек исходной грамматики:

$$S \Rightarrow b, S \Rightarrow cAB \Rightarrow cAbB \Rightarrow cAbbB \Rightarrow \dots \Rightarrow cAb\dots bB \Rightarrow ccAb\dots bB \Rightarrow \dots \Rightarrow c\dots cAb\dots bB \Rightarrow \dots$$

После приведения: $P' = \{ S \rightarrow b \}$, $G' = (\{b\}, \{S\}, P', S)$

Вывод цепочек приведённой грамматики: $S \Rightarrow b$

Язык, порождаемый грамматикой G , эквивалентен языку, порождаемому G' .

Лабораторная работа 7

Спроектировать МП автомат для приведённой КС-грамматики

Исходная грамматика

$$P = \{ S \rightarrow b, S \rightarrow cAB, A \rightarrow Ab, A \rightarrow cA, B \rightarrow cB, B \rightarrow \varepsilon \}$$

$$G = (\{b, c\}, \{A, B, S\}, P, S)$$

После приведения: $P' = \{ S \rightarrow b \}$, $G' = (\{b\}, \{S\}, P', S)$

МП-автомат: $МП = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

Q – конечное множество состояний устройства управления

Σ – конечный алфавит входных символов

Γ – конечный алфавит магазинных символов

δ – функция переходов, отображает множества $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ в множество конечных подмножеств множества $Q \times \Gamma^*$, $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$

q_0 – начальное состояние, $q_0 \in Q$

z_0 – начальный символ магазина, $z_0 \in \Gamma$

F – множество заключительных состояний, $F \subseteq Q$

$$МП = (\{q\}, \{b\}, \{b, S\}, \delta, q, S, \{q\})$$

δ :

1. $\delta(q, \varepsilon, S) = \{(q, b)\}$
2. $\delta(q, b, b) = \{(q, \varepsilon)\}$

Распознавание цепочки:

$$(q, b, S) \vdash^1 (q, b, b) \vdash^2 (q, \varepsilon, \varepsilon)$$

Лабораторная работа №8

Реализовать МП автомат для приведённой КС-грамматики

Debug Mp Deltarules :

$\text{delta}(Q\ q, T\ , Z\ S) = (Q\ q, Z\ b)$

$\text{delta}(Q\ q, T\ b, Z\ b) = (Q\ q, Z\)$

Enter the line :

b

True

Debug Mp Deltarules :

$\text{delta}(Q\ q, T\ , Z\ S) = (Q\ q, Z\ b)$

$\text{delta}(Q\ q, T\ b, Z\ b) = (Q\ q, Z\)$

Enter the line :

cb

False

Код

```
myMp MP = new myMp(new ArrayList() { "q" },
                    new ArrayList() { "b" },
                    new ArrayList() { "b", "S" },
                    "q",
                    "S",
                    new ArrayList() { "q" });

MP.addDeltaRule("q", "", "S", new ArrayList() { "q" }, new ArrayList() { "b" });
MP.addDeltaRule("q", "b", "b", new ArrayList() { "q" }, new ArrayList() { "" });

Console.WriteLine("Debug Mp ");
MP.debugDelta();

Console.WriteLine("\nEnter the line :");
Console.WriteLine(MP.Execute(Console.ReadLine()).ToString());
```