БУ ВО Ханты-Мансийского автономного округа – Югры
«Сургутский государственный университет»

Теория языков программирования и методы трансляции

Часть 5

- Факторизация грамматики
- Устранение левой рекурсии
- Построение приведенной грамматики

Гришмановский Павел Валерьевич
доцент кафедры автоматики и компьютерных систем, к.т.н., доцент

Сургут, 2018

Преобразования контекстно-свободных грамматик

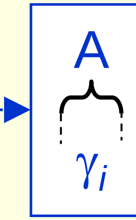
- **Факторизация** (левая факторизация) – устранение правил, правые части которых начинаются с одинаковых подцепочек (затрудняют применение нисходящих и восходящих алгоритмов)
- Устранение **левой рекурсии** (затрудняет применение нисходящих алгоритмов)
  - Устранение прямой левой рекурсии (леворекурсивных правил вида $A \to A\alpha$)
  - Устранение левой рекурсии в грамматике (группы правил, допускающих вывод вида $A \Rightarrow^* A\alpha$, например $\{ A \to B\gamma_1, B \to C\gamma_2, C \to A\gamma_3 \}$)
- Построение **приведенной грамматики** – упрощает применение алгоритмов разбора и преобразование к нормальным формам
  1. Удаление непорождающих (непроизводящих, бесплодных) символов
  2. Удаление недостижимых (невыводимых) символов
     - Непорождающие и недостижимые символы называются **несущественными**.
     - Грамматика, не содержащая несущественных символов, называется **редуцированной**
  3. Удаление λ-правил
  4. Удаление цепных правил
     - Грамматика, не содержащая несущественных символов, цепных и λ-правил называется **приведенной**
     - Порядок этих преобразований должен быть **именно таким (всех 4-х)**

Факторизация (левая факторизация)

Пусть дана грамматика G = ($\mathbf{V}^T$, $\mathbf{V}^N$, **P**, S), множество правил **P** которой содержит правила вида

$$A \to \alpha\beta_1 \mid \alpha\beta_2 \mid \ldots \mid \alpha\beta_n \mid \gamma_1 \mid \gamma_2 \mid \ldots \mid \gamma_m,$$

$A \in \mathbf{V}^N$, $\alpha \in \mathbf{V}^+$, $\beta_1, \beta_2, \ldots, \beta_n, \gamma_1, \gamma_2, \ldots, \gamma_m \in \mathbf{V}^*$



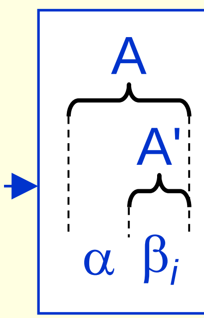
По первым символам нельзя однозначно выбрать правило

Тогда <u>вместо</u> правил

$$A \to \alpha\beta_1 \mid \alpha\beta_2 \mid \ldots \mid \alpha\beta_n$$

записывают правила

$$A \to \alpha A'$$
$$A' \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$$



где A' – новый нетерминальный символ

Факторизация (**левая**) может выполняться как для контекстно-свободных, так и для регулярных **право**линейных грамматик

Устранение левой рекурсии

В грамматике $G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$ **рекурсивным** называют такой символ $A \in \mathbf{V}^N$, для которого возможен вывод $A \Rightarrow^* \alpha A \beta$, $\alpha, \beta \in \mathbf{V}^*$

- при $\alpha = \lambda$ – левая рекурсия; при $\beta = \lambda$ – правая рекурсия

Грамматика G называется **леворекурсивной** (**праворекурсивной**), если в ней возможен вывод $A \Rightarrow^* A\beta$ ($A \Rightarrow^* \alpha A$), $A \in \mathbf{V}^N$, $\alpha, \beta \in \mathbf{V}^*$, называемый левой (правой) рекурсией

- Грамматика может одновременно являться лево- и праворекурсивной относительно разных нетерминальных символов (если относительного одного – это неоднозначность!)
- Практический интерес представляет **устранение левой рекурсии**, т.к. она делает невозможным нисходящий анализ, при котором восстанавливается цепочка левостороннего вывода

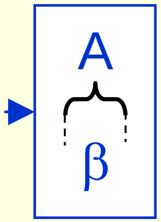
Левая рекурсия является следствием наличия во множестве **P** грамматики правил вида

- $A \to A\alpha$ – прямая левая рекурсия
- $A \to A_1\alpha_1$, $A_1 \to A_2\alpha_2$, … $A_k \to A\alpha$ – непрямая левая рекурсия

Устранение прямой левой рекурсии

Пусть дана грамматика G = ($\mathbf{V}^T$, $\mathbf{V}^N$, **P**, S), множество правил **P** которой содержит правила вида

$$A \to \boxed{A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_n} \mid \boxed{\beta_1 \mid \beta_2 \mid \ldots \mid \beta_m},$$



$$A \in \mathbf{V}^N,\ \alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_m \in \mathbf{V}^*$$

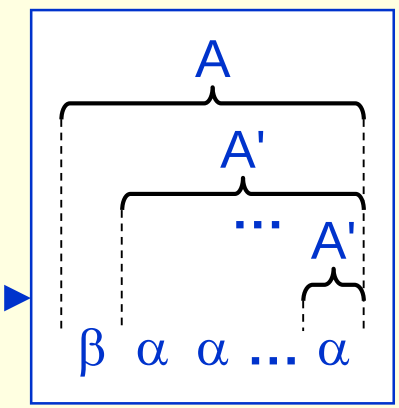
Из A могут быть выведены цепочки вида β_i, $\beta_i\,\alpha_i$, $\beta_i\,\alpha_i\,\alpha_i$ и т.д.

Тогда <u>вместо леворекурсивных</u> правил

$$A \to A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_n$$

записывают правила

$$\boxed{\begin{array}{l} A \to \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_m A' \\ A' \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n \mid \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_n A' \end{array}}$$



где A' – новый нетерминальный символ

Левая рекурсия заменяется на правую, но возможный вывод при этом сохраняется

Устранение прямой левой рекурсии (пример)

Дано: контекстно-свободная леворекурсивная грамматика $G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$:

$\mathbf{V}^T$ = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, _, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, =, ";" }

$\mathbf{V}^N$ = { S, I, E, T, M, C, A, K, D }

P = {
S → I=E;
E → E+T | E-T | T
T → T*M | T/M | M
M → (E) | -M | I | C
I → A | AK
K → A | AK | D | DK
C → D | CD
A → a | b | ... | z | _
D → 0 | 1 | ... | 9 }

Задача: привести грамматику к нелеворекурсивной (не содержащей леворекурсивных правил)

Устранение прямой левой рекурсии
(пример – продолжение)

Исходная грамматика	Удаляемые правила	Добавляемые правила
S → I=E;	–	–
E → E+T|E-T|T	E → E+T|E-T	E → TE′ E′→ +T|-T|+TE′|-TE′
T → T*M|T/M|M	T → T*M|T/M	T → MT′ T′→ *M|/M|*MT′|/MT′
M → (E)|-M|I|C	–	–
I → A|AK	–	–
K → A|AK|D|DK	–	–
C → D|CD	C → CD	C → DC′ C′→ D|DC′
A → a|b|c|…|z|_	–	–
D → 0|1|…|9	–	–

Обозначения: Леворекурсивный символ
Остаток правой части леворекурсивного правила
Новый нетерминальный символ

Устранение прямой левой рекурсии (пример – окончание)

Результат: контекстно-свободная **<u>не</u>**леворекурсивная грамматика G' = ($\mathbf{V}^T$, $\mathbf{V}^N$, **P**, S):

$\mathbf{V}^T$ = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, _,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, =, ";" }

$\mathbf{V}^N$ = { S, I, E, E', T, T', M, C, C', A, K, D }

P = {
S → I=E;
E → T | TE'
E' → +T | -T | +TE' | -TE'
T → M | MT'
T' → *M | /M | *MT' | /MT'
M → (E) | -M | I | C
I → A | AK
K → A | AK | D | DK
C → D | DC'
C' → D | DC'
A → a | b | ... | z | _
D → 0 | 1 | ... | 9
}

C → D | DC

Примечание: можно упростить полученную грамматику, исключив правила C' → D | DC' и нетерминальный символ C', а вместо правила C → DC' записать C → DC (совпадают правые части всех правил для C и C')

Преобразование грамматики в нелеворекурсивную

Вход: леворекурсивная контекстно-свободная приведенная грамматика

$G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$

Выход: нелеворекурсивная контекстно-свободная приведенная грамматика

$G' = (\mathbf{V}^{T\prime}, \mathbf{V}^{N\prime}, \mathbf{P}', S')$

Суть алгоритма преобразования грамматики в нелеворекурсивную заключается в следующем:

- определить отношение порядка на множестве нетерминальных символов $\mathbf{V}^N$ (занумеровать нетерминальные символы)
- устранить прямую левую рекурсию для каждого нетерминального символа (рассмотренный выше алгоритм)
- путем замены правил обеспечить, чтобы для каждого нетерминального символа правая часть правил начиналась:
  - с терминального символа
  - с большего по номеру нетерминального символа

  (по порядку подставлять в правые части правил вместо нетерминальных символов с меньшими номерами, если они являются первыми символами, все правые части их правил)

Преобразование грамматики в нелеворекурсивную

Алгоритм:

I. $\mathbf{V}^{T'} = \mathbf{V}^{T}$, $\mathbf{V}^{N'} = \{ A_i \in \mathbf{V}^{N} : i = \underline{1..n},\ n = |\mathbf{V}^{N}| \}$, $\mathbf{P}' = \mathbf{P}$, $S' = S$

(все множества переносятся, нетерминальные символы нумеруются)

II. Для $\forall i = \underline{1..n}$ (последовательно от 1 до n):

1. Если $(A_i \to A_i\alpha_1 \,|\, A_i\alpha_2 \,|\, \ldots \,|\, A_i\alpha_k) \in \mathbf{P}'$, то исключить их и в $\mathbf{P}'$ добавить

   $A_i \to \beta_1 A_i' \,|\, \beta_2 A_i' \,|\, \ldots \,|\, \beta_m A_i'$,

   $A_i' \to \alpha_1 \,|\, \alpha_2 \,|\, \ldots \,|\, \alpha_k \,|\, \alpha_1 A_i' \,|\, \alpha_2 A_i' \,|\, \ldots \,|\, \alpha_k A_i'$,

   где $\beta_1,\beta_2\ldots\beta_m : (A_i \to \beta_1 \,|\, \beta_2 \,|\, \ldots \,|\, \beta_m) \in \mathbf{P}'$

   (правые части всех нелеворекурсивных правил для A_i),

   A_i' – новый нетерминальный символ

   (добавить в $\mathbf{V}^{N'}$ следующим по порядку за A_i)

2. Для $\forall j = \underline{1..i\text{-}1}$ (последовательно от 1 до i-1 – все предшествующие A_i):

   Если $(A_i \to A_j\alpha_1 \,|\, A_j\alpha_2 \,|\, \ldots \,|\, A_j\alpha_k) \in \mathbf{P}'$, то исключить их и в $\mathbf{P}'$ добавить

   $A_i \to \beta_1\alpha_1 \,|\, \beta_2\alpha_1 \,|\, \ldots \,|\, \beta_m\alpha_1, \quad \ldots \quad A_i \to \beta_1\alpha_k \,|\, \beta_2\alpha_k \,|\, \ldots \,|\, \beta_m\alpha_k$,

   где $\beta_1,\beta_2\ldots\beta_m : (A_j \to \beta_1 \,|\, \beta_2 \,|\, \ldots \,|\, \beta_m) \in \mathbf{P}'$ (все правила для A_j)

3. Если в п.2 выполнены преобразования, то повторить п.1-2 для того же i (при $(A_i \to A_j\alpha) \in \mathbf{P}'$ и $(A_j \to A_t\,\beta) \in \mathbf{P}'$, $t>j$, будет добавлено $A_i \to A_p\beta\alpha$, $t \le i$, и оно будет заменено на шаге $j=t$, следовательно, остается возможным случай $A_i \to A_i\,\gamma$ – леворекурсивное правило)

Удаление непорождающих символов

Непорождающий (бесплодный, непроизводящий) **символ** – это такой нетерминальный символ $A \in \mathbf{V}^N$, из которого нельзя вывести ни одной цепочки терминальных символов $\alpha \in \mathbf{V}^{T*}$:

$$A \in \mathbf{V}^N : \{ \alpha : \exists A \Rightarrow^* \alpha, \alpha \in \mathbf{V}^{T*} \} = \varnothing$$

Вход: произвольная контекстно-свободная грамматика

$G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$

Выход: контекстно-свободная грамматика, не содержащая непорождающих символов

$G' = (\mathbf{V}^{T'}, \mathbf{V}^{N'}, \mathbf{P}', S')$

Суть алгоритма удаления непорождающих символов заключается в итерационном составлении вспомогательного множества **порождающих** (производящих) символов $\mathbf{Y} = \{A \in \mathbf{V}^N : \exists A \Rightarrow^* \alpha, \alpha \in \mathbf{V}^{T*}\}$

В новую грамматику переносятся только те правила, левые и правые части которых не содержат нетерминальных символов, не входящих в **Y** (содержат только те нетерминальные символы, которые входят в **Y**)

Удаление непорождающих символов

Алгоритм:

1. $\mathbf{Y}_0 = \varnothing$
2. $\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall A \in \mathbf{V}^N : \exists (A \to \alpha) \in \mathbf{P}, \alpha \in (\mathbf{Y}_{i\text{-}1} \cup \mathbf{V}^T)^* \}$
   - на первой итерации в **Y** включаются только нетерминальные символы $\forall A \in \mathbf{V}^N$, для которых существуют правила, содержащие в правой части только терминальные символы
   - на второй итерации – только те, для которых существуют правила, содержащие только терминальные символы и те нетерминальные символы, которые уже содержаться в **Y** и т.д.
3. Если $\mathbf{Y}_i \neq \mathbf{Y}_{i\text{-}1}$, то п.2, иначе $\mathbf{Y} = \mathbf{Y}_i$
4. $\mathbf{V}^{T'} = \mathbf{V}^T$, $\mathbf{V}^{N'} = \mathbf{Y}$, $S' = S$
   - множество Y на последней итерации содержит только порождающие символы, причем все порождающие символы
5. $\mathbf{P}' = \{ \forall (A \to \alpha) \in \mathbf{P} : A \in \mathbf{V}^{N'}, \alpha \in (\mathbf{V}^{N'} \cup \mathbf{V}^{T'})^* \}$
   - переносятся только те правила, в левой части которых порождающий символ, а в правой – только порождающие и терминальные символы

Удаление недостижимых символов

Недостижимый (невыводимый) **символ** – это такой терминальный или нетерминальный символ $x \in \mathbf{V}$, который не может встретиться в сентенциальной форме (цепочке, выводимой из начального символа грамматики S)

$$x \in \mathbf{V} : \{ \forall\gamma : S \Rightarrow^* \gamma,\ \gamma = \alpha x\beta,\ \alpha,\beta \in \mathbf{V}^*,\ x \in \mathbf{V} \} = \varnothing$$

Вход: контекстно-свободная грамматика, не содержащая непорождающих символов

$$G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$$

Выход: редуцированная контекстно-свободная грамматика (не содержащая несущественных символов)

$$G' = (\mathbf{V}^{T\prime}, \mathbf{V}^{N\prime}, \mathbf{P}', S')$$

<u>Суть</u> алгоритма удаления недостижимых символов заключается в итерационном составлении вспомогательного множества **достижимых** (выводимых) символов $\mathbf{Y} = \{ x \in \mathbf{V} : \exists\ S \Rightarrow^* \alpha x\beta,\ \alpha,\beta \in \mathbf{V}^* \}$.

В новую грамматику переносятся только правила, левые и правые части которых целиком состоят только из символов, входящих в **Y**

Удаление недостижимых символов

Алгоритм:

1. $\mathbf{Y}_0 = \{ S \}$
2. $\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall x \in \mathbf{V} : \exists (A \to \alpha x \beta) \in \mathbf{P}, A \in \mathbf{Y}_{i\text{-}1}, \alpha,\beta \in \mathbf{V}^* \}$
   - на первой итерации в **Y** включаются все те символы, которые присутствуют в правых частях правил для начального символа S
   - на второй итерации – все символы, присутствующие в правых частях правил, в левой части которых только нетерминальные символы, уже содержащиеся в **Y** и т.д.
3. Если $\mathbf{Y}_i \neq \mathbf{Y}_{i\text{-}1}$, то п.2, иначе $\mathbf{Y} = \mathbf{Y}_i$
4. $\mathbf{V}^{T'} = \mathbf{V}^T \cap \mathbf{Y}$, $\mathbf{V}^{N'} = \mathbf{V}^N \cap \mathbf{Y}$, $S' = S$
   - множество **Y** на последней итерации содержит только достижимые символы, причем все достижимые символы
5. $\mathbf{P}' = \{ \forall (A \to \alpha) \in \mathbf{P} : A \in \mathbf{V}^{N'}, \alpha \in (\mathbf{V}^{N'} \cup \mathbf{V}^{T'})^* \}$
   - переносятся только те правила, левые и правые части которых содержат только достижимые символы (терминальные и нетерминальные)

Удаление несущественных символов (пример)

Исходная грамматика

$\mathbf{V}^T = \{ a, b, c, d, e, f \}$

$\mathbf{V}^N = \{ S, A, B, C, D, E \}$

$\mathbf{P} = \{ S \to aA$

$A \to bAc \mid Cd \mid cd$

$B \to f$

$C \to CDe$

$D \to De \mid f$

$E \to AB \mid ab \ \}$

I. Удаление непорождающих символов

$\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall A \in \mathbf{V}^N : \exists\ (A \to \alpha) \in \mathbf{P},\ \alpha \in (\mathbf{Y}_{i\text{-}1} \cup \mathbf{V}^T)^* \}$

$\mathbf{Y}_0 = \varnothing$

$\mathbf{Y}_1 = \{ A, B, D, E \}$ $A \to c$

$\mathbf{Y}_2 = \{ S, A, B, D, E \}$ $S \to a$

Результат I

$\mathbf{V}^T = \{ a, b, c, d, e, f \}$

$\mathbf{V}^N = \{ S, A, B, D, E \}$

$\mathbf{P} = \{ S \to aA$

$A \to bAc \mid cd$

$B \to f$

$D \to De \mid f$

$E \to AB \mid ab \ \}$

II. Удаление недостижимых символов

$\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall x \in \mathbf{V} : \exists\ (A \to \alpha x \beta) \in \mathbf{P},\ A \in \mathbf{Y}_{i\text{-}1},\ \alpha,\beta \in \mathbf{V}^* \}$

$\mathbf{Y}_0 = \{ S \}$

$\mathbf{Y}_1 = \{ S, A, a \}$

$\mathbf{Y}_2 = \{ S, A, a, b, c, d \}$

Результат II – редуцированная грамматика

$\mathbf{V}^T = \{ a, b, c, d \}$

$\mathbf{V}^N = \{ S, A \}$

$\mathbf{P} = \{ S \to aA$

$A \to bAc \mid cd \ \}$

Удаление несущественных символов
(пример – неправильный порядок)

Исходная грамматика

$\mathbf{V}^T = \{ a, b, c, d, e, f \}$

$\mathbf{V}^N = \{ S, A, B, C, D, E \}$

$\mathbf{P} = \{ S \to aA$

$A \to bAc \mid Cd \mid cd$

$B \to f$

$C \to CDe$

$D \to De \mid f$

$E \to AB \mid ab \; \}$

I. Удаление недостижимых символов

$\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall x \in \mathbf{V} : \exists (A \to \alpha x \beta) \in \mathbf{P}, A \in \mathbf{Y}_{i\text{-}1},$
$\alpha, \beta \in \mathbf{V}^* \}$

$\mathbf{Y}_0 = \{ S \}$

$\mathbf{Y}_1 = \{ S, A, a \}$

$\mathbf{Y}_2 = \{ S, A, C, a, b, c, d \}$

$\mathbf{Y}_3 = \{ S, A, C, D, a, b, c, d,$

$\mathbf{Y}_4 = \{ S, A, C, D, a, b, c, d,$

Результат I

$\mathbf{V}^T = \{ a, b, c, d, e, f \}$

$\mathbf{V}^N = \{ S, A, C, D \}$

$\mathbf{P} = \{ S \to aA$

$A \to bAc \mid Cd \mid cd$

$C \to CDe$

$D \to De \mid f \; \}$

II. Удаление непорождающих символов

$\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall A \in \mathbf{V}^N : \exists (A \to \alpha) \in \mathbf{P}, \alpha \in (\mathbf{Y}_{i\text{-}1} \cup \mathbf{V}^T)^* \}$

$\mathbf{Y}_0 = \varnothing$

$\mathbf{Y}_1 = \{ A, D \}$ $\quad A \to c$

$\mathbf{Y}_2 = \{ S, A, D \}$ $\quad S \to a$

Результат II – *не* редуцированная грамматика

$\mathbf{V}^T = \{ a, b, c, d, e, f \}$

$\mathbf{V}^N = \{ S, A, D \}$

$\mathbf{P} = \{ S \to aA$

$A \to bAc \mid cd$

$D \to De \mid f \; \}$

Удаление несущественных символов (пример – сравнение результатов)

Исходная грамматика

V^T = { a, b, c, d, e, f }
V^N = { S, A, B, C, D, E }
P = { $S \to aA$
$A \to bAc \mid Cd \mid cd$
$B \to f$
$C \to CDe$
$D \to De \mid f$
$E \to AB \mid ab$ }

Редуцированная грамматика

V^T = { a, b, c, d }
V^N = { S, A }
P = { $S \to aA$
$A \to bAc \mid cd$ }

Преобразованная грамматика,
но ***не*** редуцированная (неправильный порядок преобразований)

V^T = { a, b, c, d, e, f }
V^N = { S, A, D }
P = { $S \to aA$
$A \to bAc \mid cd$
$D \to De \mid f$ }

Недостижимые символы
e,f $\in$ V^T, D $\in$ V^N
Лишние правила
$D \to De \mid f$

Вывод: Исключение правил, содержащих непроизводящие символы, может привести к появлению недостижимых символов, но не наоборот (символ может стать недостижимым, если он находится в правой части правила, содержащего непроизводщий символ)

Удаление λ-правил

λ-правило (правило с пустой цепочкой) – это любое правило грамматики вида $A \to \lambda$, $A \in \mathbf{V}^N$ (в т.ч. $S \to \lambda$)

Контекстно-свободная грамматика называется **неукорачивающей**, если она не содержит λ-правил

$$\{ (A \to \lambda) \in \mathbf{P} : \forall A \in \mathbf{V}^N \} = \varnothing$$

λ-свободной грамматикой (грамматикой без λ-правил) называют грамматику, которая не содержит λ-правил, и может существовать только правило $S \to \lambda$, если $\lambda \in L(G)$, причем S не встречается в правых частях правил

$$\{ (A \to \lambda) \in \mathbf{P} : \forall A \in \mathbf{V}^N, A \neq S \} = \varnothing \wedge$$
$$\wedge \{ (A \to \alpha S \beta) \in \mathbf{P} : \forall A \in \mathbf{V}^N, \exists (S \to \lambda) \in \mathbf{P}, \alpha,\beta \in \mathbf{V}^* \} = \varnothing$$

Для любой контекстно-свободной грамматики G может быть построена эквивалентная λ-свободная грамматика G'

Для любой контекстно-свободной грамматики G, $\lambda \notin L(G)$ может быть построена эквивалентная неукорачивающая грамматика G'

Для любой λ-свободной грамматики G может быть построена почти эквивалентная неукорачивающая грамматика G' и наоборот

Удаление λ-правил

Вход: контекстно-свободная редуцированная грамматика

$G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$

Выход: контекстно-свободная λ-свободная грамматика

$G' = (\mathbf{V}^{T'}, \mathbf{V}^{N'}, \mathbf{P}', S')$

Суть алгоритма удаления λ-правил заключается в итерационном составлении вспомогательного множества **Y** таких нетерминальных символов, из которых возможен вывод пустой цепочки λ

$\mathbf{Y} = \{ \forall A \in \mathbf{V}^N : \exists A \Rightarrow^* \lambda \}$

В новую грамматику переносятся все правила, кроме λ-правил, и добавляются новые правила, образованные исключением из правых частей правил символов, входящих в **Y**, во всех возможных комбинациях

Если $S \in \mathbf{Y}$, то $\lambda \in L(G)$, т.к. $\exists S \Rightarrow^* \lambda$

Удаление λ-правил

Алгоритм:

1. $\mathbf{Y}_0 = \{ \forall A \in \mathbf{V}^N : (A \to \lambda) \in \mathbf{P} \}$
   - в **Y** включаются все нетерминальные символы, которые присутствуют в левых частях λ-правил, т.е. $\exists\, A \Rightarrow \lambda$
2. $\mathbf{Y}_i = \mathbf{Y}_{i\text{-}1} \cup \{ \forall A \in \mathbf{V}^N : (A \to \alpha) \in \mathbf{P},\ \alpha \in \mathbf{Y}_{i\text{-}1}^{\ +} \}$
   - в **Y** включаются все нетерминальные символы, которые присутствуют в левых частях правил, правые части которых целиком состоят из символов, уже присутствующих в **Y**, т.е. $\exists\, A \Rightarrow^+ \lambda$, и т.д.
3. Если $\mathbf{Y}_i \neq \mathbf{Y}_{i\text{-}1}$, то п.2, иначе $\mathbf{Y} = \mathbf{Y}_i$
4. $\mathbf{V}^{T'} = \mathbf{V}^T$, $\mathbf{V}^{N'} = \mathbf{V}^N$, $\mathbf{P'}_0 = \{ (A \to \alpha) \in \mathbf{P} : \alpha \in \mathbf{V}^+ \}$
   - во множество **P'** переносятся все правила, кроме λ-правил
5. $\mathbf{P'}_i = \mathbf{P'}_{i\text{-}1} \cup \{ (A \to \alpha\beta) :$
   $(A \to \alpha X \beta) \in \mathbf{P'}_{i\text{-}1},\ \forall A \in \mathbf{V}^{N'},\ \forall X \in \mathbf{Y},\ \alpha,\beta \in \mathbf{V'}^*,\ |\alpha\beta| > 0 \}$
   - добавляются новые правила, образованные путем исключения из правых частей правил, содержащих символы из **Y**, всех возможных комбинаций этих символов, кроме получающихся λ-правил, т.к. $|\alpha\beta| > 0$
6. Если $\mathbf{P'}_i \neq \mathbf{P'}_{i\text{-}1}$, то п.5
7. Если $S \in \mathbf{Y}$, то $\mathbf{P'} = \mathbf{P'}_i \cup \{ S' \to \lambda | S \}$, где S' – новый нетерминальный символ, иначе $\mathbf{P'} = \mathbf{P'}_i$, $S' = S$
   - добавляется $S' \to \lambda$, если $\lambda \in L(G)$

Удаление λ-правил (пример)

Дано: контекстно-свободная редуцированная грамматика $G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$:

$\mathbf{V}^{T'} = \{ a, b, c, d, e, f, g \}$

$\mathbf{V}^{N'} = \{ S, A, B, C, D, E, F \}$

$$
\mathbf{P}' = \{ \begin{array}{lll}
S & \to & A \mid BC \mid bcd \\
A & \to & aBCdDefFgh \mid aBc \\
B & \to & b \mid bc \\
C & \to & c \mid \lambda \\
D & \to & d \mid dAe \\
F & \to & Ff \mid g \mid \lambda
\end{array} \}
$$

Найти: контекстно-свободную λ-свободную грамматику $G' = (\mathbf{V}^{T'}, \mathbf{V}^{N'}, \mathbf{P}', S')$:

Удаление λ-правил (пример)

Исходная грамматика

$$
\begin{aligned}
P = \{\ & S \to A \mid \boxed{BC} \mid bcd \\
& A \to aBCdDefFgh \mid aBc \\
& B \to b \mid bc \mid \boxed{C} \\
& C \to c \mid \boxed{\lambda} \\
& D \to d \mid dAe \\
& F \to Ff \mid g \mid \boxed{\lambda} \quad \}
\end{aligned}
$$

1-3. Составление вспомогательного множества

$$Y_i = Y_{i\text{-}1} \cup \{\ \forall A \in V^N : (A \to \alpha) \in \mathbf{P},\ \alpha \in Y_{i\text{-}1}{}^*\ \}$$

$Y_0 = \{ C, F \}$	$C \to \lambda,\ F \to \lambda$
$Y_1 = \{ C, F, B \}$	$B \to C$
$Y_2 = \{ C, F, B, S \}$	$S \to BC$

4. Перенос правил	5-6. Добавление правил (по итерациям)			7. λ∈L(G)
P'_0	P'_1	P'_2	P'_3	
S → A | BC | bcd	| B | C	~~| λ~~	–	S′ → S | λ
A → aBCdDefFgh | aBc	| aCdDefFgh | aBdDefFgh | aBCdDefgh | ac	| adDefFgh | aCdDefgh | aBdDefgh	| adDefgh	
B → b | bc | C	~~| λ~~	–	–	
C → c	–	–	–	
D → d | dAe	–	–	–	
F → Ff | g	| f	–	–	

Удаление λ-правил (пример)

Результат: контекстно-свободная λ-свободная грамматика $G' = (\mathbf{V}^{T'}, \mathbf{V}^{N'}, \mathbf{P}', S')$:

$\mathbf{V}^{T'} = \{ a, b, c, d, e, f, g \}$

$\mathbf{V}^{N'} = \{ S', S, A, B, C, D, E, F \}$

$\mathbf{P}' = \{ S' \to S \mid \lambda$

$S \to A \mid BC \mid bcd \mid B \mid C$

$A \to aBCdDefFgh \mid aBc \mid aCdDefFgh \mid aBdDefFgh \mid aBCdDefgh \mid adDefFgh \mid aCdDefgh \mid aBdDefgh \mid adDefgh \mid ac$

$B \to b \mid bc \mid C$

$C \to c$

$D \to d \mid dAe$

$F \to Ff \mid g \mid f \quad \}$

Вывод: исключение λ-правил может приводить к появлению цепных правил

Удаление цепных правил

Цепное (сингулярное) **правило** – это правило $(A \to B) \in \mathbf{P}$, $A,B \in \mathbf{V}^N$
Цепные правила являются необходимым условием циклов, т.е. вывода вида $A \Rightarrow^* A$

Вход: контекстно-свободная λ-свободная грамматика

$$G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$$

Выход: контекстно-свободная приведенная грамматика

$$G' = (\mathbf{V}^{T'}, \mathbf{V}^{N'}, \mathbf{P}', S')$$

<u>Суть</u> алгоритма удаления цепных правил заключается в итерационном составлении для каждого нетерминального символа $A \in \mathbf{V}^N$ такого вспомогательного множества $\mathbf{Y}^A$ цепных символов $B \in \mathbf{V}^N$, которые связаны возможным выводом $A \Rightarrow^* B$

$$\mathbf{Y}^A = \{ \forall B \in \mathbf{V}^N : \exists\, A \Rightarrow^* B \}$$

В новую грамматику переносятся все правила, кроме цепных, и для каждого символа $A \in \mathbf{V}^N$ добавляются новые правила, образованные всеми правыми частями существующих правил (кроме цепных) для символов, входящих в $\mathbf{Y}^A$

$$A \to \alpha : \exists\, (B \to \alpha) \in \mathbf{P},\ \forall A \in \mathbf{V}^N,\ \forall B \in \mathbf{Y}^A$$

Удаление цепных правил

Алгоритм:

1. Для $\forall A \in \mathbf{V}^N$:
   1. $\mathbf{Y}^A{}_0 = \{\ \forall B \in \mathbf{V}^N : (A \to B) \in \mathbf{P}\ \}$
      - в $\mathbf{Y}^A$ включаются все нетерминальные символы B, которые составляют правые части цепных правил для A, т.е. $\exists\ A \Rightarrow B$
   2. $\mathbf{Y}^A{}_i = \mathbf{Y}^A{}_{i\text{-}1} \cup \{\ \forall B \in \mathbf{V}^N : (X \to B) \in \mathbf{P},\ X \in \mathbf{Y}^A{}_{i\text{-}1}\ \}$
      - в $\mathbf{Y}^A$ включаются все нетерминальные символы B, для которых возможен вывод $A \Rightarrow^+ B$
   3. Если $\mathbf{Y}^A{}_i \neq \mathbf{Y}^A{}_{i\text{-}1}$, то п.1.2, иначе $\mathbf{Y}^A = \mathbf{Y}^A{}_i$
2. $\mathbf{V}^{T'} = \mathbf{V}^T,\ \mathbf{V}^{N'} = \mathbf{V}^N,\ S' = S$
   - перенос терминальных, нетерминальных и целевого символа
3. $\mathbf{P}'_0 = \{\ (A \to \alpha) \in \mathbf{P}\ \} \setminus \{\ (A \to B) \in \mathbf{P}\ \},\ \forall A,B \in \mathbf{V}^N,\ \alpha \in \mathbf{V}^+$
   - во множество $\mathbf{P}'$ переносятся все правила, за исключением цепных
4. $\mathbf{P}' = \mathbf{P}'_0 \cup \{\ (A \to \alpha) : (B \to \alpha) \in \mathbf{P}'_0,\ \forall A \in \mathbf{V}^{N'},\ \forall B \in \mathbf{Y}^A,\ \alpha \in \mathbf{V}'^+\ \}$
   - добавляются новые правила, образованные правыми частями всех не цепных правил всех цепных символов

Удаление цепных правил
(пример)

Дано: контекстно-свободная λ-свободная грамматика $G = (\mathbf{V}^T, \mathbf{V}^N, \mathbf{P}, S)$:

$\mathbf{V}^T$ = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, _, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, =, ";" }

$\mathbf{V}^N$ = { S, I, E, E', T, T', M, C, A, K, D }

$\mathbf{P}$ = {

$S \to I{=}E;$

$E \to T \mid TE'$

$E' \to +T \mid -T \mid +TE' \mid -TE'$

$T \to M \mid MT'$

$T' \to *M \mid /M \mid *MT' \mid /MT'$

$M \to (E) \mid -M \mid I \mid C$

$I \to A \mid AK$

$K \to A \mid AK \mid D \mid DK$

$C \to D \mid DC$

$A \to a \mid b \mid \ldots \mid z \mid \_$

$D \to 0 \mid 1 \mid \ldots \mid 9$

}

Получить: контекстно-свободную приведенную грамматику $G' = (\mathbf{V}^{T\prime}, \mathbf{V}^{N\prime}, \mathbf{P}', S')$

Удаление цепных правил
(пример – 1. Составление множеств цепных символов)

Исходная грамматика	Цепные символы	Результирующая грамматика
S → I=E;	$Y^S = \varnothing$	
E → T | TE′	$Y^E = \{ T, M, I, C, A, D \}$	
E′ → +T | -T | +TE′ | -TE′	$Y^{E'} = \varnothing$	
T → M | MT′	$Y^T = \{ M, I, C, A, D \}$	
T′ → *M | /M | *MT′ | /MT′	$Y^{T'} = \varnothing$	
M → (E) | -M | I | C	$Y^M = \{ I, C, A, D \}$	
I → A | AK	$Y^I = \{ A \}$	
K → A | AK | D | DK	$Y^K = \{ A, D \}$	
C → D | DC	$Y^C = \{ D \}$	
A → a | b | c | … | z | _	$Y^A = \varnothing$	
D → 0 | 1 | … | 9	$Y^D = \varnothing$	

Обозначения: Удаляемое правило
Добавляемое правило

Удаление цепных правил
(пример – 3. Перенос нецепных правил)

Исходная грамматика	Цепные символы	Результирующая грамматика
S → I=E;	$Y^S = \varnothing$	S → I=E;
E → T | TE′	$Y^E = \{ T, M, I, C, A, D \}$	E → TE′
E′ → +T | –T | +TE′ | –TE′	$Y^{E'} = \varnothing$	E′ → +T | –T | +TE′ | –TE′
T → M | MT′	$Y^T = \{ M, I, C, A, D \}$	T → MT′
T′ → *M | /M | *MT′ | /MT′	$Y^{T'} = \varnothing$	T′ → *M | /M | *MT′ | /MT′
M → (E) | –M | I | C	$Y^M = \{ I, C, A, D \}$	M → (E) | –M
I → A | AK	$Y^I = \{ A \}$	I → AK
K → A | AK | D | DK	$Y^K = \{ A, D \}$	K → AK | DK
C → D | DC	$Y^C = \{ D \}$	C → DC
A → a | b | c | … | z | _	$Y^A = \varnothing$	A → a | b | c | … | z | _
D → 0 | 1 | … | 9	$Y^D = \varnothing$	D → 0 | 1 | … | 9

Обозначения: Удаляемое правило
Добавляемое правило

Удаление цепных правил
(пример – 4. Добавление новых правил)

Исходная грамматика	Цепные символы	Результирующая грамматика
S → I=E;	$Y^S = \varnothing$	S → I=E;
E → T | TE′	$Y^E = \{T, M, I, C, A, D\}$	E → TE′ | MT′ | (E) | –M | AK | DC | | a | b | c | … | z | _ | 0 | 1 | … | 9
E′ → +T | –T | +TE′ | –TE′	$Y^{E'} = \varnothing$	E′ → +T | –T | +TE′ | –TE′
T → M | MT′	$Y^T = \{M, I, C, A, D\}$	T → MT′ | (E) | –M | AK | DC | | a | b | c | … | z | _ | 0 | 1 | … | 9
T′ → *M | /M | *MT′ | /MT′	$Y^{T'} = \varnothing$	T′ → *M | /M | *MT′ | /MT′
M → (E) | –M | I | C	$Y^M = \{I, C, A, D\}$	M → (E) | –M | AK | DC | | a | b | c | … | z | _ | 0 | 1 | … | 9
I → A | AK	$Y^I = \{A\}$	I → AK | a | b | c | … | z | _
K → A | AK | D | DK	$Y^K = \{A, D\}$	K → AK | DK | | a | b | c | … | z | _ | 0 | 1 | … | 9
C → D | DC	$Y^C = \{D\}$	C → DC | 0 | 1 | … | 9
A → a | b | c | … | z | _	$Y^A = \varnothing$	A → a | b | c | … | z | _
D → 0 | 1 | … | 9	$Y^D = \varnothing$	D → 0 | 1 | … | 9

Обозначения: Удаляемое правило
Добавляемое правило

Удаление цепных правил
(пример – окончание)

Результат: контекстно-свободная приведенная грамматика

G' = (**V**^{T}', **V**^{N}', **P**', S'):

V^{T} = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, _,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, =, ";" }

V^{N} = { S, I, E, E', T, T', M, C, A, K, D }

P = { S → I=E;
E → TE' | MT' | (E) | -M | AK | DC | a | b | c |…| z | _ | 0 | 1 |…| 9
E' → +T | -T | +TE' | -TE'
T → MT' | (E) | -M | AK | DC | a | b | c |…| z | _ | 0 | 1 |…| 9
T' → *M | /M | *MT' | /MT'
M → (E) | -M | AK | DC | a | b | c |…| z | _ | 0 | 1 |…| 9
I → AK | a | b | c |…| z | _
K → AK | DK | a | b | c |…| z | _ | 0 | 1 |…| 9
C → DC | 0 | 1 |…| 9
A → a | b | ... | z | _
D → 0 | 1 | ... | 9 }