Лекции по теории формальных языков

Лекция 5.
Операции над контекстно-свободными языками. Контекстно-свободные языки и автоматы с магазинной памятью. Контекстно-свободные грамматики и языки программирования. Аннулирующие нетерминалы контекстно-свободной грамматики

Александр Сергеевич Герасимов
http://gas-teach.narod.ru

Кафедра математических и информационных технологий
Санкт-Петербургского академического университета
Российской академии наук.
Весенний семестр 2010/11 учебного года

11 марта 2011 г.

План

План

Предложение о замкнутости класса КС-языков относительно объединения, произведения и итерации

Предложение

Класс КС-языков замкнут относительно объединения, произведения и итерации.

Доказательство. (Было дано в доказательстве теоремы о языках в однобуквенных алфавитах и периодических множествах, см. лекцию 4).

- Пусть языки L_1 и L_2 порождаются КС-грамматиками $G_1 = (\Sigma_1, \Gamma_1, P_1, S_1)$ и $G_2 = (\Sigma_2, \Gamma_2, P_2, S_2)$ соответственно. Можно считать, что $\Gamma_1 \cap \Gamma_2 = \emptyset$.
- Построим КС-грамматики, порождающие языки $L_1 \cup L_2$, $L_1 L_2$ и L_1^*. Выберем новый нетерминал $S \notin \Gamma_1 \cup \Gamma_2$.
  - $L_1 \cup L_2$ порождается КС-грамматикой $\{S \to S_1 | S_2\} \cup P_1 \cup P_2$.
  - $L_1 L_2$ порождается КС-грамматикой $\{S \to S_1 S_2\} \cup P_1 \cup P_2$.
  - L_1^* порождается КС-грамматикой $\{S \to S_1 S | \varepsilon\} \cup P_1$.

Теорема о замкнутости класса КС-языков относительно подстановок

Теорема

Пусть Σ, Δ — алфавиты, τ — подстановка из Σ в Δ, для любого символа $\mathrm{a} \in \Sigma$ язык $\tau(\mathrm{a})$ является контекстно-свободным. Тогда если язык $L \subseteq \Sigma^*$ является контекстно-свободным, то и язык $\tau(L)$ является контекстно-свободным.

Доказательство.

- Пусть $\Sigma = \{\mathrm{a}_1, \ldots, \mathrm{a}_n\}$, язык L порождается КС-грамматикой $G = (\Sigma, \Gamma, P, S)$, и для каждого $i = 1, \ldots, n$ язык $\tau(\mathrm{a}_i)$ порождается КС-грамматикой $G_i = (\Sigma_i, \Gamma_i, P_i, S_i)$. Можно считать, что Γ, Γ_1, …, Γ_n попарно не пересекаются.

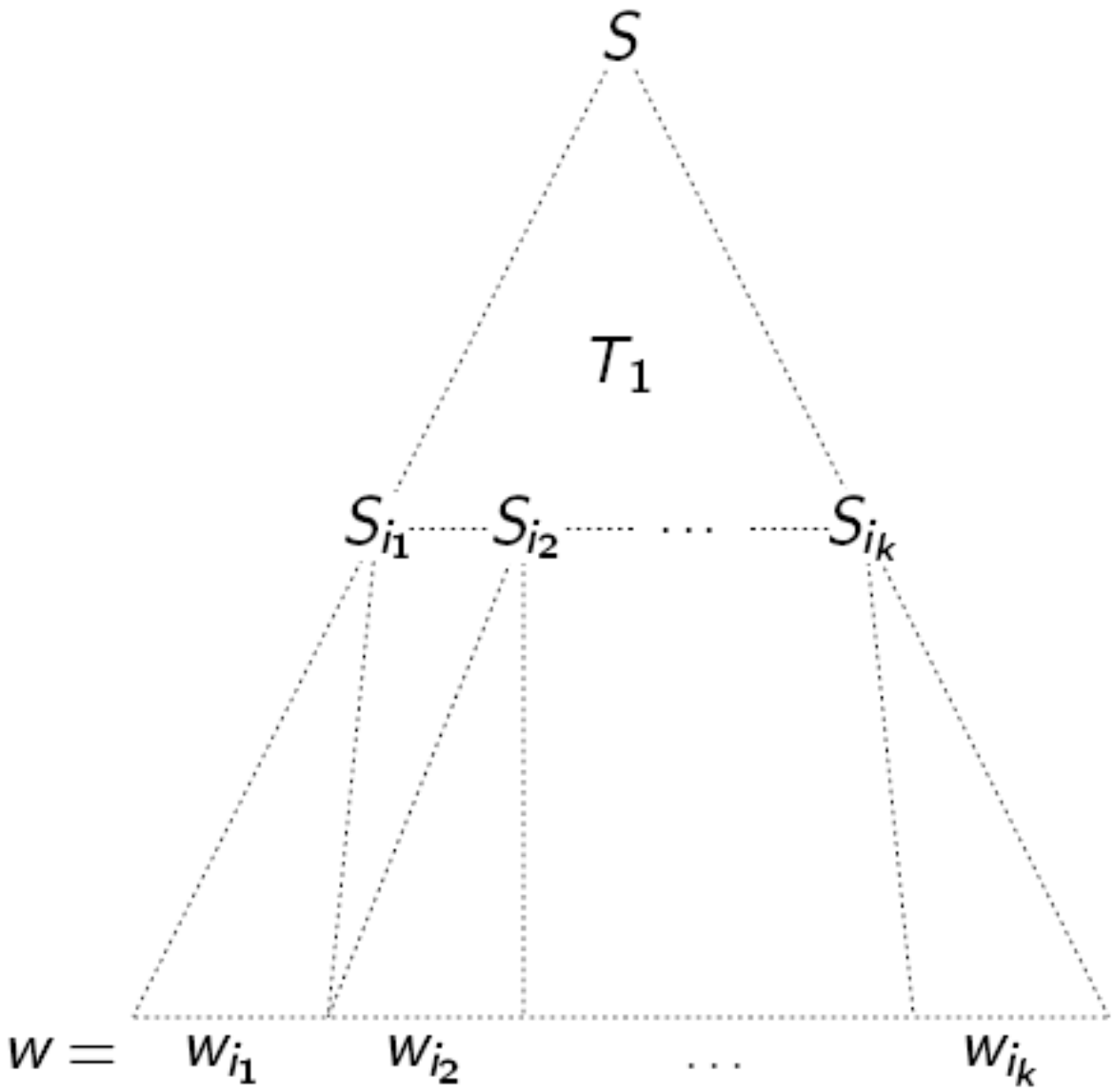
Теорема: продолжение доказательства

- Определим КС-грамматику $H = (\Delta,\ \Gamma \cup \Gamma_1 \cup \ldots \cup \Gamma_n,\ P',\ S)$,

$$P' = P_1 \cup \ldots \cup P_n \cup \{A \to [\beta]^{\mathrm{a}_1,\ldots,\mathrm{a}_n}_{S_1,\ldots,S_n} \mid (A \to \beta) \in P\},$$

  где $[\beta]^{\mathrm{a}_1,\ldots,\mathrm{a}_n}_{S_1,\ldots,S_n}$ — результат замены всех вхождений $\mathrm{a}_1, \ldots, \mathrm{a}_n$ в β на $S_1, \ldots, S_n$ соответственно. Докажем, что $\tau(L) = L(H)$.
- Пусть $w \in \tau(L)$. Тогда $w \in \tau(u)$ для некоторой цепочки $u = \mathrm{a}_{i_1} \ldots \mathrm{a}_{i_k} \in L$.
- $w \in \tau(u) = \tau(\mathrm{a}_{i_1}) \ldots \tau(\mathrm{a}_{i_k})$, значит, $w = w_{i_1} \ldots w_{i_k}$ для некоторых цепочек $w_{i_1} \in \tau(\mathrm{a}_{i_1}),\ \ldots,\ w_{i_k} \in \tau(\mathrm{a}_{i_k})$.
- Имеем $S \Rightarrow^*_G \mathrm{a}_{i_1} \ldots \mathrm{a}_{i_k},\ \ S_{i_1} \Rightarrow^*_{G_{i_1}} w_{i_1},\ \ \ldots,\ \ S_{i_k} \Rightarrow^*_{G_{i_k}} w_{i_k}$.
- Отсюда $S \Rightarrow^*_H S_{i_1} \ldots S_{i_k} \Rightarrow^*_H w_{i_1} \ldots w_{i_k} = w$, следовательно, $w \in L(H)$.
- Обратно, пусть $w \in L(H)$. Возьмём какое угодно дерево вывода T цепочки w в грамматике H.

Теорема: продолжение доказательства



Теорема о замкнутости класса КС-языков относительно подстановок: окончание доказательства

- По определению грамматики H существует такое стандартное поддерево T_1 дерева T, что все листья T_1 помечены нетерминалами $S_{i_1}, \ldots, S_{i_k}$.
- T_1 представляет вывод $S \Rightarrow_H^* S_{i_1} \ldots S_{i_k}$.
- Тогда $S \Rightarrow_G^* \mathrm{a}_{i_1} \ldots \mathrm{a}_{i_k} = u \in L$.
- Далее, $S \Rightarrow_H^* S_{i_1} \ldots S_{i_k} \Rightarrow_H^* w_{i_1} \ldots w_{i_k} = w$, где
  - $S_{i_1} \Rightarrow_{G_{i_1}}^* w_{i_1} \in L(G_{i_1}) = \tau(\mathrm{a}_{i_1})$,
  - $\ldots$,
  - $S_{i_k} \Rightarrow_{G_{i_k}}^* w_{i_k} \in L(G_{i_k}) = \tau(\mathrm{a}_{i_k})$.
- Таким образом, $w = w_{i_1} \ldots w_{i_k} \in \tau(\mathrm{a}_{i_1}) \ldots \tau(\mathrm{a}_{i_k}) = \tau(u) \subseteq \tau(L)$.

Следствия теоремы о замкнутости класса КС-языков относительно подстановок I

Уже доказанное нами

Предложение

Класс КС-языков замкнут относительно объединения, произведения и итерации.

можно получить как следствие теоремы о замкнутости класса КС-языков относительно подстановок:

- путь мы имеем КС-языки L_1 и L_2;
- определим подстановку τ так, что $\tau(\mathrm{a}_1) = L_1$ и $\tau(\mathrm{a}_2) = L_2$;
- КС-языки $\{\mathrm{a}_1, \mathrm{a}_2\}$, $\{\mathrm{a}_1\mathrm{a}_2\}$, $\{\mathrm{a}_1\}^*$ в результате действия подстановки τ дают языки $L_1 \cup L_2$, $L_1 L_2$, L_1^* соответственно.

Следствия теоремы о замкнутости класса КС-языков относительно подстановок II

Следствие

Класс КС-языков замкнут относительно перехода к гомоморфным образам.

Доказательство.

- Пусть L — КС-язык в алфавите Σ, $\phi : \Sigma^* \to \Delta^*$ — гомоморфизм.
- Тогда $\phi(L) = \tau(L)$, где τ — такая подстановка, что $\tau(\mathrm{a}) = \{\phi(\mathrm{a})\}$ для каждого символа $\mathrm{a} \in \Sigma$.
- По теореме о замкнутости класса КС-языков относительно подстановок язык $\phi(L) = \tau(L)$ контекстно-свободен.

□

Язык $\{\mathrm{a}^n\mathrm{b}^n\mathrm{c}^n \mid n \geqslant 1\}$ не является контекстно-свободным, поскольку его гомоморфный образ $\{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n \mid n \geqslant 1\}$ не является КС-языком (см. лекцию 4).

Следствия теоремы о замкнутости класса КС-языков относительно подстановок III

Следствие

Если L — КС-язык, то множество $\{|w| \mid w \in L\}$ является периодическим.

Доказательство.

- Пусть Σ — алфавит языка L, $\mathtt{b}$ — какой угодно символ.
- Определим гомоморфизм $\phi : \Sigma^* \to \{\mathtt{b}\}^*$ так, что $\phi(\mathtt{a}) = \mathtt{b}$ для любого символа $\mathtt{a} \in \Sigma$.
- Тогда по предыдущему следствию язык $\phi(L) \subseteq \{\mathtt{b}\}^*$ контекстно-свободен.
- Очевидно, $\{|w| \mid w \in L\} = \{n \geqslant 0 \mid \mathtt{b}^n \in \phi(L)\}$.
- В силу теоремы о языках в однобуквенных алфавитах и периодических множествах (см. лекцию 4) множество $\{|w| \mid w \in L\}$ является периодическим.

Предложение о незамкнутости класса КС-языков относительно пересечения и дополнения

Предложение

Класс КС-языков не замкнут относительно пересечения и дополнения.

Доказательство.

- Язык $L_1 = \{a^n b^n a^m \mid m, n \geqslant 1\}$ является контекстно-свободным, поскольку L_1 есть произведение КС-языков $\{a^n b^n \mid n \geqslant 1\}$ и $\{a\}^+$.
- Аналогично язык $L_2 = \{a^m b^n a^n \mid m, n \geqslant 1\}$ является контекстно-свободным.
- Однако $L_1 \cap L_2 = \{a^n b^n a^n \mid n \geqslant 1\}$ не является КС-языком (см. лекцию 4). Таким образом, класс КС-языков не замкнут относительно пересечения.
- Класс КС-языков замкнут относительно объединения и $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$, поэтому класс КС-языков не замкнут относительно дополнения.

Теорема о пересечении КС-языка с регулярным языком

Теорема

Пересечение КС-языка с регулярным языком является КС-языком.

Доказательство.

- Пусть K — КС-язык, порождаемый КС-грамматикой $G = (\Sigma, \Gamma, P, S)$, R — регулярный язык, распознаваемый ДКА $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. Покажем, что $K \cap R$ — КС-язык.
- В силу теоремы о неукорачивающей КС-грамматике, которую мы докажем на лекции 6, можно считать, что G не имеет правил вида $A \to \varepsilon$, кроме разве лишь $S \to \varepsilon$ (но в последнем случае S не входит в правую часть никакого правила).
- Если $\varepsilon \in K$ и мы докажем, что $(K \setminus \{\varepsilon\}) \cap R$ — КС-язык, то и $K \cap R$ будет КС-языком. Далее считаем, что $\varepsilon \notin K$ и G не имеет правил вида $A \to \varepsilon$.

Теорема о пересечении КС-языка с регулярным языком: продолжение доказательства

- Для каждого состояния $\mathtt{f} \in F$ определим ДКА $\mathcal{A}_\mathtt{f} = (Q, \Sigma, \delta, \mathtt{q}_0, \{\mathtt{f}\})$.
- Имеем $R = L(\mathcal{A}) = \bigcup\limits_{\mathtt{f}\in F} L(\mathcal{A}_\mathtt{f})$ и потому

$$K \cap R = K \cap \bigcup_{\mathtt{f}\in F} L(\mathcal{A}_\mathtt{f}) = \bigcup_{\mathtt{f}\in F} K \cap L(\mathcal{A}_\mathtt{f}).$$

- Таким образом, достаточно доказать, что $K \cap R$ — КС-язык, если F содержит единственное состояние. Положим $F = \{\mathtt{f}\}$.
- Для порождения языка $K \cap R$ определим КС-грамматику $H = (\Sigma, \Gamma', P', S')$, где $\Gamma' = Q \times (\Sigma \cup \Gamma) \times Q$, $S' = (\mathtt{q}_0, S, \mathtt{f})$, а множество правил P' строится так:
  (1) если $(A \to X_1 X_2 \dots X_k) \in P$, то для каждой упорядоченной $(k+1)$-ки состояний $\mathtt{q}, \mathtt{r}, \mathtt{p}_1, \dots, \mathtt{p}_{k-1} \in Q$ множество P' содержит правило $(\mathtt{q}, A, \mathtt{r}) \to (\mathtt{q}, X_1, \mathtt{p}_1)(\mathtt{p}_1, X_2, \mathtt{p}_2) \dots (\mathtt{p}_{k-1}, X_k, \mathtt{r})$;
  (2) если $\mathtt{a} \in \Sigma$ и $\delta(\mathtt{q}, \mathtt{a}) = \mathtt{r}$, то P' содержит правило $(\mathtt{q}, \mathtt{a}, \mathtt{r}) \to \mathtt{a}$.

Теорема о пересечении КС-языка с регулярным языком: окончание доказательства

- Правила грамматики H вида (2) порождают только листья дерева вывода, поэтому можно считать, что в выводе цепочки терминалов в грамматике H из S' сначала применяются правила вида (1), а затем — правила вида (2).
- Тогда такой вывод имеет вид

$$S' = (\mathrm{q}_0, S, \mathrm{f}) \Rightarrow_H^* (\mathrm{q}_0, \mathrm{a}_1, \mathrm{p}_1)(\mathrm{p}_1, \mathrm{a}_2, \mathrm{p}_2) \dots (\mathrm{p}_{k-1}, \mathrm{a}_k, \mathrm{f}) \Rightarrow_H^* \mathrm{a}_1\mathrm{a}_2 \dots \mathrm{a}_k = w.$$

- По определению грамматики H выводимость, отмеченная первой стрелкой $\Rightarrow_H^*$, имеет место тогда и только тогда, когда $S \Rightarrow_G^* w$, т. е. когда $w \in K$.
- Выводимость, отмеченная второй стрелкой $\Rightarrow_H^*$, имеет место тогда и только тогда, когда $\delta(\mathrm{q}_0, w) = \mathrm{f}$, т. е. когда $w \in R$.
- $S' \Rightarrow_H^* w$ тогда и только тогда, когда $w \in K \cap R$. Значит, $K \cap R = L(H)$ и $K \cap R$ — КС-язык.

План

Теорема о распознавании КС-языка НМПА

Теорема

Любой КС-язык распознаётся некоторым НМПА с единственным состоянием и единственной командой допуска $(\dashv, \nabla) \to \checkmark$.

Доказательство.

- Пусть $G = (\Sigma, \Gamma, P, S)$ — произвольная КС-грамматика.
- Определим НМПА $\mathcal{M}$ с
  - единственным состоянием (мы всюду опускаем это состояние),
  - входным алфавитом $\Sigma \cup \{\dashv\}$,
  - стековым алфавитом $\Sigma \cup \Gamma \cup \{\nabla\}$,
  - начальным содержимым стека S и
  - множеством команд δ.

  Множество команд δ строится так:

  (1) для каждой пары $a \in \Sigma$ и $B \in \Gamma$ и каждого правила $(B \to \gamma) \in P$ в δ есть команда $(a, B) \to (\gamma, \_)$;
  (2) для каждого $a \in \Sigma$ в δ есть команда $(a, a) \to (\varepsilon, \rightarrow)$;
  (3) в δ есть команда допуска $(\dashv, \nabla) \to \checkmark$.

Теорема о распознавании КС-языка НМПА: продолжение доказательства

- Покажем, что $L(G) = L(\mathcal{M})$.
- Пусть $w \in L(G)$. Тогда существует левосторонний вывод в G

$$S \Rightarrow u_1 B_1 \alpha_1 \Rightarrow u_1 u_2 B_2 \alpha_2 \Rightarrow \ldots \Rightarrow u_1 \ldots u_{n-1} B_{n-1} \alpha_{n-1} \Rightarrow u_1 \ldots u_n = w.$$

- Тогда автомат $\mathcal{M}$ может последовательно переходить в конфигурации

$$[w,\ S] = [u_1 u_2 \ldots u_n,\ S] \models [u_1 u_2 \ldots u_n,\ u_1 B_1 \alpha_1] \models^* [u_2 \ldots u_n,\ B_1 \alpha_1] \models^* \ldots \models^* [u_n,\ B_{n-1} \alpha_{n-1}] \models [u_n,\ u_n] \models^* [\varepsilon,\ \varepsilon].$$

- В конфигурации $[\varepsilon,\ \varepsilon]$ выполняется команда допуска. Таким образом, $w \in L(\mathcal{M})$.

Теорема о распознавании КС-языка НМПА: продолжение доказательства

- Обратно, пусть $w \in L(\mathcal{M})$. Тогда автомат $\mathcal{M}$ из конфигурации $[w,\ S]$ может перейти в конфигурацию $[\varepsilon,\ \varepsilon]$ за конечное число m шагов и в последней конфигурации выполнить команду допуска.
- Выберем $\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_m \in \Sigma \cup \{\varepsilon\}$ и $\gamma_1 = S,\ \gamma_2, \ldots, \gamma_m,\ \varepsilon = \gamma_{m+1} \in (\Sigma \cup \Gamma)^*$ так, что $w = \mathrm{a}_1\mathrm{a}_2 \ldots \mathrm{a}_m$ и

$$[w,\ S] = [\mathrm{a}_1\mathrm{a}_2 \ldots \mathrm{a}_m,\ \gamma_1] \models [\mathrm{a}_2 \ldots \mathrm{a}_m,\ \gamma_2] \models \ldots \models [\mathrm{a}_m,\ \gamma_m] \models [\varepsilon,\ \varepsilon].$$

- По построению команд автомата $\mathcal{M}$ имеем:
  - если на i-м шаге $(i = 1, \ldots, m)$ применяется команда вида (2), то $\gamma_i = \mathrm{a}_i\gamma_{i+1}$, $\mathrm{a}_i \in \Sigma$;
  - если на i-м шаге $(i = 1, \ldots, m)$ применяется команда вида (1), то $\mathrm{a}_i = \varepsilon$, $\gamma_i = B_i\beta_i$ для некоторых $B_i \in \Gamma$ и $\beta_i \in (\Sigma \cup \Gamma)^*$, а также $\gamma_{i+1} = \alpha_i\beta_i$ для некоторого $(B_i \to \alpha_i) \in P$.

Теорема о распознавании КС-языка НМПА: окончание доказательства

- Пусть команды вида (1) применялись только на шагах $1 = k_1 < k_2 < \ldots < k_n$.
- Тогда имеем следующий вывод цепочки w в грамматике G:

$$S = B_{k_1} \Rightarrow \alpha_{k_1} = a_1 \ldots a_{k_2-1} B_{k_2} \beta_{k_2} \Rightarrow$$
$$a_1 \ldots a_{k_2-1} \alpha_{k_2} \beta_{k_2} = a_1 \ldots a_{k_2-1} a_{k_2} \ldots a_{k_3-1} B_{k_3} \beta_{k_3} \Rightarrow \ldots \Rightarrow$$
$$a_1 \ldots a_{k_{n-1}-1} \alpha_{k_{n-1}} \beta_{k_{n-1}} = a_1 \ldots a_{k_{n-1}-1} a_{k_{n-1}} \ldots a_{k_n-1} B_{k_n} \beta_{k_n} \Rightarrow$$
$$a_1 \ldots a_{k_n-1} \alpha_{k_n} \beta_{k_n} = a_1 \ldots a_m = w.$$

- Итак, $w \in L(G)$.

Теорема о принадлежности языка, распознаваемого НМПА, классу КС-языков. Следствия

Теорема

Язык, распознаваемый НМПА, является контекстно-свободным.

Следствие

Клас языков, распознаваемых НМПА, совпадает с классом КС-языков.

Следствие

Клас языков, распознаваемых ДМПА, является собственным подклассом класса КС-языков.

План

Форма Бэкуса-Наура

- Синтаксис языков программирования традиционно описывается с помощью *форм Бэкуса-Наура* (*БНФ*) или их модификаций.
- БНФ является иной записью КС-грамматики.
- Обычно в БНФ
  - вместо знака «$\to$» в записи правил БНФ используется знак «::=» или знак «=»;
  - каждый "терминал" заключается в кавычки (при этом кавычки «"» в качестве терминала записываются два раза подряд в кавычках: «""""""»), и каждый нетерминал является начинающейся с заглавной буквы цепочкой из букв русского (английского) алфавита, десятичных цифр и пробела; или каждый ⟨нетерминал⟩ заключается в угловые скобки;
  - вместо обозначения пустой строки стоит она сама.

Расширенная БНФ

В правых частях правил *расширенной БНФ* (*РБНФ*) можно использовать следующие операции:

Операция	Значение
$[\beta]$	0 или 1 вхождение цепочки β
$\{\beta\}$	0 или более вхождений подряд цепочки β
$(\beta_1\|\ldots\|\beta_n)$	вхождение либо цепочки β_1, ..., либо цепочки β_n

Правило РБНФ	Правила [Р]БНФ без дополнительной операции
$A = \alpha[\beta]\gamma$	$A = \alpha\gamma\|\alpha\beta\gamma$
$A = \alpha\{\beta\}\gamma$	$A = \alpha B\gamma$, $B = \|\beta B$, где B — новый нетерминал
$A = \alpha(\beta_1\|\ldots\|\beta_n)\gamma$	$A = \alpha\beta_1\gamma\|\ldots\|\alpha\beta_n\gamma$

Фрагмент РБНФ языка программирования Паскаль

- Программа = Заголовок программы ";" Блок "."
- Оператор присваивания = (Переменная | Имя функции) ":=" Выражение
- Выражение = Простое выражение [Операция отношения Простое выражение]
- Простое выражение = [Знак] Терм {Аддитивная операция Терм}
- Терм = Фактор {Мультипликативная операция Фактор}
- Фактор = Константа без знака | Имя границы | Переменная | Конструктор множества | Обозначение функции | "not" Фактор | "(" Выражение ")"
- Операция отношения = "=" | "<>" | "<" | "<=" | ">" | ">=" | "in"
- Аддитивная операция = "+" | "-" | "or"

Язык программирования Паскаль не является КС-языком

- Пусть L_P — множество всех правильных программ языка программирования Паскаль,

  $L = \{\texttt{program p; var } u\texttt{:integer; begin } u\texttt{:=2 end.} \mid u \in \{\texttt{c}, \texttt{f}\}^+\}$,

  R — регулярный язык, определяемый регулярным выражением (где r^+ обозначает rr^*)

  $$\texttt{program p; var } (\texttt{c}|\texttt{f})^+\texttt{:integer; begin } (\texttt{c}|\texttt{f})^+\texttt{:=2 end.}$$

- Тогда $L = L_P \cap R$.
- Пусть h — такой гомоморфизм, что $h(\texttt{c}) = \texttt{c}$, $h(\texttt{f}) = \texttt{f}$, $h(\texttt{;}) = \texttt{;}$ и $h(X) = \varepsilon$ для всех прочих X.
- Если бы L_P был КС-языком, то и язык

  $$h(L) = \{\texttt{;}w\texttt{;}w \mid w \in \{\texttt{c}, \texttt{f}\}^+\}$$

  был бы КС-языком — противоречие.

Контекстные условия

- Не любая терминальная цепочка, выводимая из аксиомы в КС-грамматике (БНФ) языка программирования Паскаль, является программой, правильной относительно спецификации этого языка программирования.
- Спецификация Паскаля содержит некоторые контекстные условия, которые не выражаются КС-грамматиками.
  - Описание имён.
    `program p; var c:integer; begin f:=2 end.`
  - Совместимость типов.
    `program p; var b:boolean; begin b:=2 end.`
  - Соответствие (по числу и типам) фактических параметров процедур и функций формальным параметрам.
- Типичный компилятор проверяет контекстные условия на фазе семантического (или контекстного) анализа, следующей за (или проходящей одновременно с) фазой синтаксического анализа.

План

ε-правила и аннулирующие нетерминалы КС-грамматики

- Правило вывода вида $A \to \varepsilon$ называется *аннулирующим* (или *ε-правилом*).
- Нетерминал B КС-грамматики G называется *аннулирующим*, если $B \Rightarrow_G^* \varepsilon$.
- Множество всех аннулирующих нетерминалов КС-грамматики G обозначим через $Ann(G)$.
- Пример. Грамматика G_1:
  $S \to \mathtt{a}BC|AE,\ A \to \mathtt{b}CD|\varepsilon,\ B \to ACA,\ C \to \varepsilon,\ E \to CA,\ D \to \mathtt{b}ES|\mathtt{c}.$
  - $A, C \in Ann(G)$, поскольку $A \Rightarrow \varepsilon$, $C \Rightarrow \varepsilon$;
  - $B, E \in Ann(G)$, поскольку $B \Rightarrow ACA \Rightarrow^* \varepsilon$, $E \Rightarrow CA \Rightarrow^* \varepsilon$;
  - $S \in Ann(G)$, поскольку $S \Rightarrow AE \Rightarrow^* \varepsilon$;
  - $D \notin Ann(G)$.

Алгоритм ANN нахождения аннулирующих нетерминалов КС-грамматики

Вход. КС-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Множество $Ann(G)$ всех аннулирующих нетерминалов грамматики G.

1. $i := 1$; $Ann_0 := \emptyset$; $Ann_1 := \{A \in \Gamma \mid (A \to \varepsilon) \in P\}$
2. **пока** $(Ann_{i-1} \neq Ann_i)$ **повторять**
3. $\quad Ann_{i+1} := \{A \in \Gamma \mid \exists \alpha \in Ann_i^* \ ((A \to \alpha) \in P)\}$;
4. $\quad i := i + 1$;
5. $Ann(G) := Ann_i$

Пример работы этого алгоритма для грамматики G_1 с правилами $S \to \mathrm{a}BC|AE,\ A \to \mathrm{b}CD|\varepsilon,\ B \to ACA,\ C \to \varepsilon,\ E \to CA,\ D \to \mathrm{b}ES|\mathrm{c}$.

- $Ann_1 = \{A, C\}$;
- $Ann_2 = \{A, C, B, E\}$;
- $Ann_3 = \{A, C, B, E, S\} = Ann_4 = Ann(G_1)$.

Корректность алгоритма Ann

- Индукцией по i легко показать, что $Ann_{i-1} \subseteq Ann_i$ для любого $i \geqslant 1$.
- Алгоритм Ann завершается, выполнив тело цикла не более $|\Gamma|$ раз, поскольку иначе выполнялось бы

$$\emptyset = Ann_0 \subsetneq Ann_1 \subsetneq Ann_2 \subsetneq \ldots \subsetneq Ann_{|\Gamma|} \subsetneq Ann_{|\Gamma|+1} \subseteq \Gamma$$

  и в $Ann_{|\Gamma|+1}$ было бы более $|\Gamma|$ нетерминалов.
- Далее, индукцией по i покажем, что $A \in Ann_i$ влечёт $A \Rightarrow^* \varepsilon$. Поэтому все нетерминалы, возвращаемые алгоритмом, являются аннулирующими.
- Заметим, что если $Ann_{i-1} = Ann_i$, то $Ann_{i-1} = Ann_j$ для любого $j \geqslant i$.
- Наконец, индукцией по длине вывода докажем, что $A \Rightarrow^* \varepsilon$ влечёт $A \in Ann_i$ для некоторого i. Поэтому все аннулирующие нетерминалы возвращаются алгоритмом.

Литература

Основная литература

- Замятин А. П., Шур А. М. Языки, грамматики, распознаватели: Учебное пособие. Екатеринбург : Изд-во Урал. ун-та, 2007 (электронный вариант книги — на `http://elar.usu.ru`, поиск).
- Йенсен К., Вирт Н. Паскаль: руководство для пользователя. М.: Финансы и статистика, 1989.

Дополнительная литература

- Ахо А., Лам М., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструментарий. М.: ООО "И.Д. Вильямс", 2008.
- Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.
- Мартыненко Б. К. Языки и трансляции: Учеб. пособие. СПб.: Издательство С.-Петербургского университета, 2004 (электронный вариант книги — на `http://www.math.spbu.ru/user/mbk`).