Лекции по теории формальных языков

Лекция 9.
Синтаксический анализ для LL(1)-грамматик (окончание).
LL(k)-грамматики.
Синтаксический анализ методом рекурсивного спуска.
Общая схема восходящего синтаксического анализа

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

8 апреля 2011 г.

План

План

Вычисление $\textsc{select}(A \to \alpha)$ и $\textsc{first}(X)$

Пусть дана LL(1)-грамматика $G = (\Sigma, \Gamma, P, S)$.

$$\textsc{select}(A \to \alpha) = \begin{cases} \textsc{first}(\alpha), & \text{если } \varepsilon \notin \textsc{first}(\alpha), \\ (\textsc{first}(\alpha) \setminus \{\varepsilon\}) \cup \textsc{follow}(A), & \text{если } \varepsilon \in \textsc{first}(\alpha). \end{cases}$$

Для вычисления $\textsc{select}(A \to \alpha)$ достаточно уметь вычислять $\textsc{first}(\alpha)$ и $\textsc{follow}(A)$.

Как вычислить $\textsc{first}(X)$?

- $\textsc{first}(\mathsf{a}) = \{\mathsf{a}\}$.
- $\textsc{first}(A) = \bigcup\limits_{(A \to \alpha) \in P} \textsc{first}(\alpha)$.
- Пусть $L \subseteq_{\mathsf{t}} M$ означает, что $L \cap \Sigma \subseteq M$.
- Рассмотрим $\alpha = X_1 \ldots X_n$, $n > 0$.
  - $\textsc{first}(X_1) \subseteq_{\mathsf{t}} \textsc{first}(\alpha)$;
  - $\textsc{first}(X_2) \subseteq_{\mathsf{t}} \textsc{first}(\alpha)$, если $\varepsilon \in \textsc{first}(X_1)$;
  - $\textsc{first}(X_3) \subseteq_{\mathsf{t}} \textsc{first}(\alpha)$, если $\varepsilon \in \textsc{first}(X_1) \cap \textsc{first}(X_2)$; $\ldots$;
  - $\textsc{first}(X_n) \subseteq_{\mathsf{t}} \textsc{first}(\alpha)$, если $\varepsilon \in \textsc{first}(X_1) \cap \ldots \cap \textsc{first}(X_{n-1})$;
  - $\varepsilon \in \textsc{first}(\alpha)$, если $\varepsilon \in \textsc{first}(X_1) \cap \ldots \cap \textsc{first}(X_n)$.

Алгоритмы вычисления $\mathrm{FIRST}(A)$ и $\mathrm{FIRST}(\alpha)$

Вход. LL(1)-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Массив множеств $\mathrm{FIRST}(A)$ для каждого $A \in \Gamma$.

1. **для каждого** $a \in \Sigma$ $\mathrm{FIRST}(a) := \{a\}$;
2. **для каждого** $A \in \Gamma$
3. **если** $((A \to \varepsilon) \in P)$ $\mathrm{FIRST}(A) := \{\varepsilon\}$;
4. **иначе** $\mathrm{FIRST}(A) := \emptyset$;
5. **пока** все множества $\mathrm{FIRST}(A)$ не стабилизировались, **повторять**
6. **для каждого** $(A \to X_1 \dots X_n) \in P$, где $n > 0$, **выполнить**
7. $i := 1$;
8. $\mathrm{FIRST}(A) := \mathrm{FIRST}(A) \cup (\mathrm{FIRST}(X_i) \cap \Sigma)$;
9. **если** $(\varepsilon \in \mathrm{FIRST}(X_i))$
10. **если** $(i < n)$
11. $i := i + 1$; **перейти на** 8;
12. **иначе** $\mathrm{FIRST}(A) := \mathrm{FIRST}(A) \cup \{\varepsilon\}$

Вычисление $\mathrm{FIRST}(\alpha)$ для $\alpha = X_1 \dots X_n$ $(n > 0)$: инициализировать $\mathrm{FIRST}(\alpha) := \emptyset$ и выполнить строки 7–12, где A заменён на α.

План

Вычисление $\textsc{follow}(A)$

- $\dashv \in \textsc{follow}(S)$.
- Какие символы из $\Sigma \cup \{\dashv\}$ могут стоять непосредственно за нетерминалом B в (каких угодно, необязательно левых) выводах из аксиомы?
- Рассмотрим каждое вхождение B в правую часть каждого правила: $A \to \alpha B \beta$. Символ a может стоять непосредственно за B в указанных выводах в случаях:
  - $\beta \Rightarrow^* \mathrm{a}\beta'$, тогда $\mathrm{a} \in \textsc{first}(\beta)$;
  - $\beta \Rightarrow^* \varepsilon$, тогда $\mathrm{a} \in \textsc{follow}(A)$.

Алгоритм вычисления $\textsc{follow}(A)$

Вход. LL(1)-грамматика $G = (\Sigma, \Gamma, P, S)$.
Выход. Массив множеств $\textsc{follow}(A)$ для каждого $A \in \Gamma$.

1. **для каждого** $A \in \Gamma$ $\quad \textsc{follow}(A) := \emptyset$;
2. $\textsc{follow}(S) := \{\dashv\}$;
3. **пока** все множества $\textsc{follow}(A)$ не стабилизировались, **повторять**
4. $\quad$ **для каждого** $(A \to X_1 \dots X_n) \in P$, где $n > 0$, **выполнить**
5. $\quad\quad i := n$; $\quad ann := \texttt{true}$;
6. $\quad\quad$ **если** $(X_i \in \Gamma)$
7. $\quad\quad\quad \textsc{follow}(X_i) := \textsc{follow}(X_i) \cup (\textsc{first}(X_{i+1} \dots X_n) \cap \Sigma)$;
8. $\quad\quad\quad$ **если** (ann) $\quad // \; X_{i+1} \dots X_n \Rightarrow^* \varepsilon$
9. $\quad\quad\quad\quad \textsc{follow}(X_i) := \textsc{follow}(X_i) \cup \textsc{follow}(A)$;
10. $\quad\quad$ **если** $(i > 1)$
11. $\quad\quad\quad$ **если** $(ann \wedge \varepsilon \notin \textsc{first}(X_i))$ $\quad ann := \texttt{false}$;
12. $\quad\quad\quad i := i - 1$; **перейти на** 6

Пример вычисления FIRST(A) и FOLLOW(A)

Грамматика GA_3:

(1) $E \to TE'$, (2) $E' \to +TE'$, (3) $T \to FT'$, (4) $T' \to *FT'$,
(5) $F \to (E)$, (6) $F \to \mathtt{x}$, (7) $E' \to \varepsilon$, (8) $T' \to \varepsilon$.

Построение FIRST	Построение FOLLOW
(8) $\varepsilon \mapsto T'$	$\dashv \mapsto E$
(7) $\varepsilon \mapsto E'$	(1) $\dashv \mapsto E'$
(6) $\mathtt{x} \mapsto F$	(1) $+ \mapsto T$
(5) $(\mapsto F$	(1) $\dashv \mapsto T$
(4) $* \mapsto T'$	(3) $+, \dashv \mapsto T'$
(3) $\mathtt{x}, (\mapsto T$	(3) $* \mapsto F$
(2) $+ \mapsto E'$	(3) $+, \dashv \mapsto F$
(1) $\mathtt{x}, (\mapsto E$	(5) $) \mapsto E$
	(1) $) \mapsto E'$
	(1) $) \mapsto T$
	(3) $) \mapsto T'$
	(3) $) \mapsto F$

Нтрм.	FIRST	FOLLOW
E	$(, \mathtt{x}$	$), \dashv$
E'	$+, \varepsilon$	$), \dashv$
T	$(, \mathtt{x}$	$+,), \dashv$
T'	$*, \varepsilon$	$+,), \dashv$
F	$(, \mathtt{x}$	$+, *,), \dashv$

План

Ошибки, обнаруживаемые при синтаксическом анализе

- Хороший синтаксический анализатор, кроме сообщения о найденной во входной цепочке ошибке, должен восстановить своё состояние («исправить ошибку») и продолжить анализ.
- МПА обнаруживает ошибку, когда у него нет команды с левой частью (входной символ, верхний символ стека).
- Возможные синтаксические ошибки в цепочке, которая должна быть арифметическим выражением, задаваемым грамматикой GA_3:
  - отсутствует операнд;
  - отсутствует оператор;
  - правая скобка без парной ей левой скобки;
  - левая скобка без парной ей правой скобки.

Исправление ошибок: пример

	$\mathtt{x}$	$+$	$*$	$($	$)$	$\dashv$
E	TE'			TE'		
E'		$+TE'$			ε	ε
T	FT'			FT'		
T'		ε	$*FT'$		ε	ε
F	$\mathtt{x}$			(E)		

Позиция указателя	Содержимое стека	Исправление
$\diamond\ )(\mathtt{x}+\mathtt{x})(*\mathtt{x}+\mathtt{x}\dashv$	$E\nabla$	пропустить)
$)(\mathtt{x}+\mathtt{x})\diamond(*\mathtt{x}+\mathtt{x}\dashv$	$T'E'\nabla$	вставить $*$
$)(\mathtt{x}+\mathtt{x})(\diamond*\mathtt{x}+\mathtt{x}\dashv$	$E)T'E'\nabla$	пропустить $*$
$)(\mathtt{x}+\mathtt{x})(*\mathtt{x}+\mathtt{x}\diamond\dashv$	$)T'E'\nabla$	вставить)

Позиция указателя	Содержимое стека	Исправление
$(\mathtt{x}+\mathtt{x})\diamond\ )(*\mathtt{x}+\mathtt{x}\dashv$	$T'E'\nabla$	
$(\mathtt{x}+\mathtt{x})\diamond\ )(*\mathtt{x}+\mathtt{x}\dashv$	$E'\nabla$	
$(\mathtt{x}+\mathtt{x})\diamond\ )(*\mathtt{x}+\mathtt{x}\dashv$	∇	пропустить) ?

Общая стратегия восстановления после ошибок: режим паники

- Пропуск входных символов до тех пор, пока не встретится символ, (предположительно) позволяющий продолжить анализ.
- Если в момент обнаружения ошибки на вершине стека находится нетерминал A, то можно пропустить все входные символы до первого из $\textsc{follow}(A)$ и снять A с вершины стека.
- Также можно пропустить все входные символы до первого из $\textsc{first}(A)$ и продолжить анализ, оставив A на вершине стека.
- Если при обнаружения ошибки на вершине стека находится терминал, то можно вставить этот терминал во входную цепочку.
- Замечание. Ошибочный входной символ может повредить стек до обнаружения ошибки.

План

LL(k)-грамматики: определения

- Пусть $G = (\Sigma, \Gamma, P, S)$ — КС-грамматика, $\alpha \in (\Sigma \cup \Gamma)^*$. Тогда

  $$\mathrm{FIRST}_k(\alpha) = \{u \in \Sigma^* \mid (\exists \beta\, (\alpha \Rightarrow^* u\beta) \wedge |u| = k) \vee (\alpha \Rightarrow^* u \wedge |u| < k)\}.$$

- Пусть k — целое неотрицательное число. КС-грамматика G с аксиомой S называется *LL(k)-грамматикой*, если из существования двух (левых) выводов
  - $S \Rightarrow^* wA\alpha \Rightarrow w\beta\alpha \Rightarrow^* wx$ и
  - $S \Rightarrow^* wA\alpha \Rightarrow w\gamma\alpha \Rightarrow^* wy$,

  для которых $\mathrm{FIRST}_k(x) = \mathrm{FIRST}_k(y)$, следует, что $\beta = \gamma$.
- Грамматика называется *LL-грамматикой*, если она является LL(k)-грамматикой для некоторого целого неотрицательного числа k.
- Язык называется *LL(k)-языком*, если существует LL(k)-грамматика, порождающая этот язык.
- Язык называется *LL-языком*, если он является LL(k)-языком для некоторого k.

LL(k)-грамматики: факты и примеры

- Класс всех LL(k)-грамматик $\subsetneq$ класс всех LL($k+1$)-грамматик.
- Грамматика $G_1 = \{S \to \mathrm{a}^k\mathrm{b} | \mathrm{a}^k\mathrm{c}\}$ не является LL(k)-грамматикой, но является LL($k+1$)-грамматикой.
- Грамматика

$$G_2 = \{S \to A|B,\ A \to \mathrm{a}A\mathrm{b}|0,\ B \to \mathrm{a}B\mathrm{bb}|1\}$$

  не является LL-грамматикой:
  - $S \Rightarrow^* S \Rightarrow \beta = A \Rightarrow^* x = \mathrm{a}^k 0 \mathrm{b}^k$,
  - $S \Rightarrow^* S \Rightarrow \gamma = B \Rightarrow^* y = \mathrm{a}^k 1 \mathrm{b}^{2k}$,

  $\mathrm{FIRST}_k(x) = \mathrm{FIRST}_k(y)$, но $\beta \neq \gamma$.
- Класс всех LL(k)-языков $\subsetneq$ класс всех LL($k+1$)-языков.
- $L(G_2)$ не является LL-языком.

План

Метод рекурсивного спуска

- Применяется для синтаксического анализа цепочек языка, порождаемого нелеворекурсивной грамматикой.
- Программа синтаксического анализатора состоит из набора процедур, по одной для каждого нетерминала грамматики.
- Процедура $A()$ для нетерминала A распознаёт во входной цепочке подцепочку, выводимую из A.

процедура $A()$ {
  выбрать A-правило $A \to X_1 \dots X_k$;
  for $(i = 1, \dots, k)$ {
    if (X_i — нетерминал)
      вызвать процедуру $X_i()$;
    else if (X_i = текущий входной символ)
      перейти к следующему входному символу;
    else возврат в точку выбора правила или обработка ошибки;
  }
}

Выбор A-правила в процедуре $A()$

- Если имеется более одного A-правила, то в общем случае требуется возврат и выбор другого A-правила (и повторное чтение части входной цепочки!).
- Для LL(1)-грамматики возвраты не требуются: выбирается такое A-правило $A \to \alpha$, что текущий входной символ принадлежит $\textsc{select}(A \to \alpha)$.

План

Восходящий анализ. Соглашения

- Синтаксический анализ называется *восходящим анализом* (*анализом снизу вверх*), если дерево вывода анализируемой цепочки строится снизу (от листьев) вверх. При этом цепочка терминалов «сворачивается» в аксиому грамматики и строится (от конца к началу) правый вывод этой цепочки.
- Далее считаем все рассматриваемые КС-грамматики приведёнными, неукорачивающими и однозначными, выводы — правыми, а формы — r-формами, если не будет оговорено иное.

Основа r-формы

- Пусть дан (правый) вывод в грамматике с аксиомой S:

$$S = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n = w \ .$$

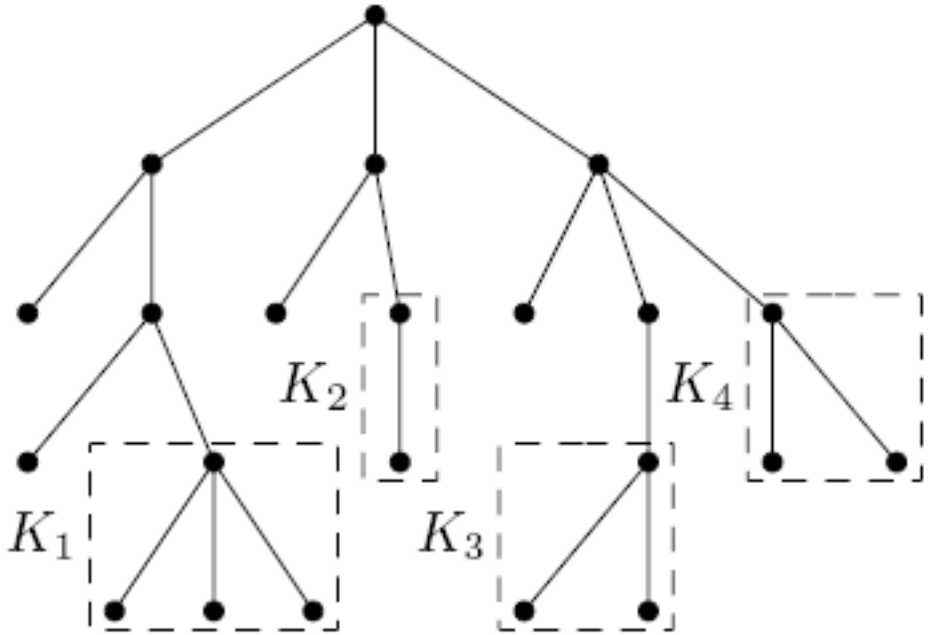
  Тогда для каждого $k = 1, \ldots, n$ *основой* r-формы α_k называют такое вхождение цепочки β в цепочку α_k, что $\alpha_{k-1} = \gamma_1 A \gamma_2$ и $\alpha_k = \gamma_1 \beta \gamma_2$ для некоторых γ_1, γ_2 и A.
- Часто об основе формы α_k говорят как о цепочке β, подразумевая вхождение цепочки β в цепочку α_k.
- Основа r-формы определяется единственным образом в силу однозначности грамматики.
- Пример. Грамматика

$$G_1 = \{S \to \mathtt{a}F S \mathtt{d} | \mathtt{c},\ F \to F\mathtt{b} | \mathtt{b}\}.$$

  Если цепочка $\mathtt{a}F\mathtt{bcd}$ является r-формой, то её основа — $F\mathtt{b}$.
  $S \Rightarrow \mathtt{a}FS\mathtt{d} \Rightarrow \mathtt{a}F\mathtt{cd} \Rightarrow \mathtt{a}F\mathtt{bcd} \Rightarrow \mathtt{abbcd}$

Куст дерева

Поддерево K дерева T назовём *кустом*, если K состоит из некоторого узла дерева T и всех сыновей этого узла, причём все эти сыновья являются листьями дерева T.



Куст дерева вывода и основа формы

- Пусть (правый) вывод

$$S = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n = w$$

  в грамматике (с аксиомой S) представлен деревом вывода T; T_i, T_{i+1} — стандартные поддеревья дерева T, представляющие выводы форм α_i и α_{i+1} соответственно.
- Тогда T_i получается из T_{i+1} удалением всех листьев самого левого куста.
- На листьях самого левого куста стандартного поддерева дерева вывода написана основа r-формы, вывод которой представлен этим поддеревом.

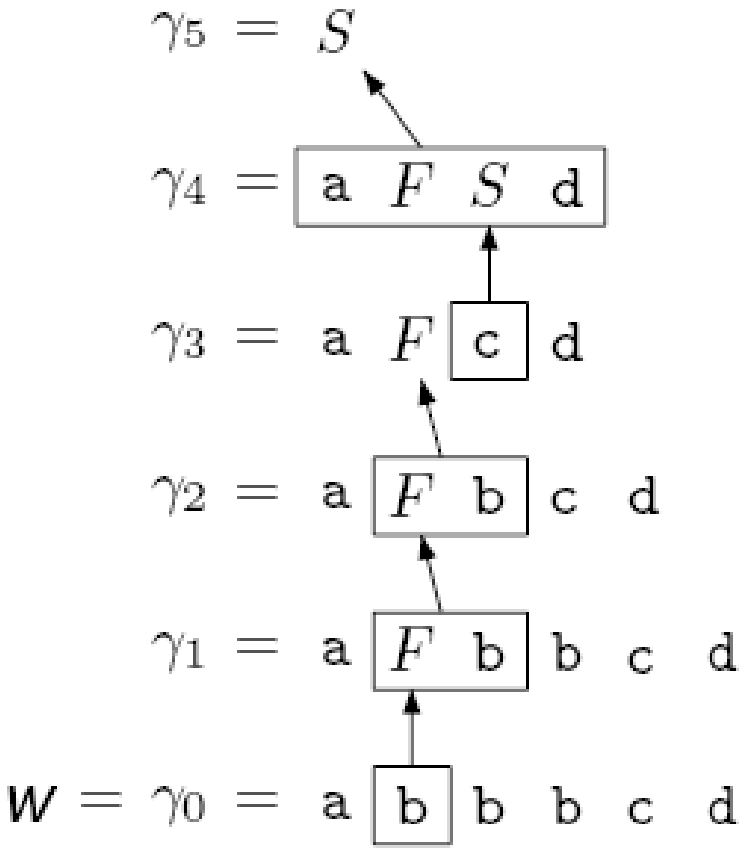
Схема восходящего анализа

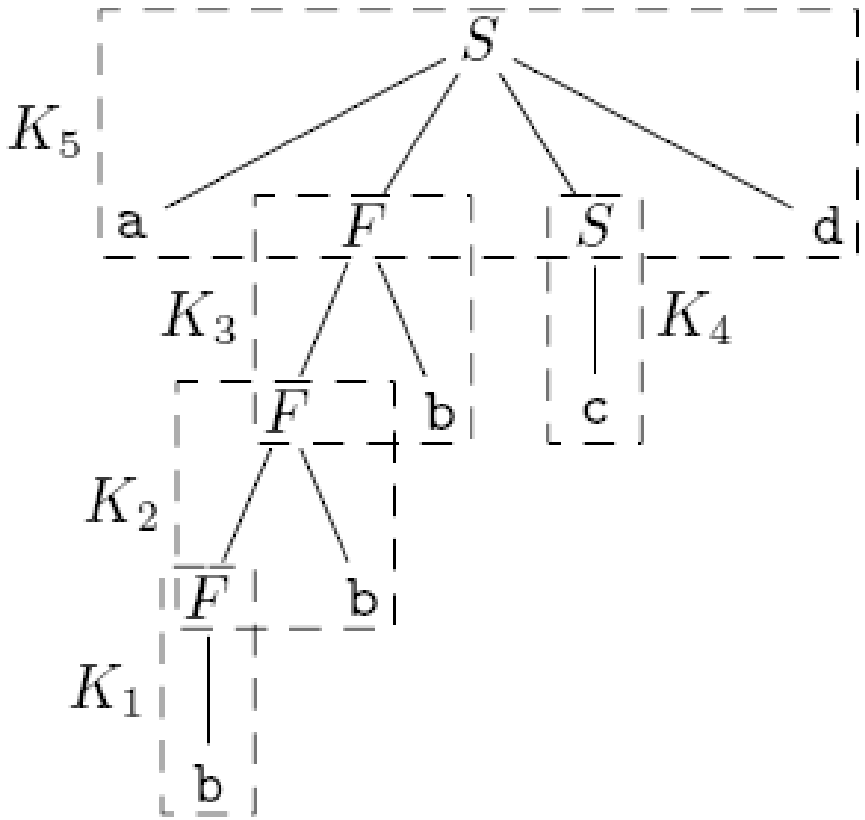
- Предполагаем, что заданная цепочка терминалов $w = \gamma_0$ выводима из аксиомы S.
- Находим предполагаемую основу β цепочки γ_0.
- Если в грамматике нет правила вида $A \to \beta$, то заключаем, что γ_0 невыводима.
- Иначе выберем правило вида $A \to \beta$ и произведём *свёртку по этому правилу*, т. е. заменим в цепочке γ_0 её основу на A. Обозначим полученную в результате этой замены цепочку через γ_1.
- Описанный процесс повторяем для цепочки γ_1 и т. д. (Конкретный алгоритм анализа может предписывать возврат в точки выбора правил для испытания других свёрток.)
- Если получили цепочку $\gamma_n = S$, то имеем (правый) вывод цепочки w:

$$S = \gamma_n \Rightarrow \gamma_{n-1} \Rightarrow \ldots \Rightarrow \gamma_1 \Rightarrow \gamma_0 = w \ .$$

Пример восходящего анализа

Грамматика $G_1 = \{S \to \mathtt{a}F S\mathtt{d} | \mathtt{c},\ F \to F\mathtt{b} | \mathtt{b}\}$.





$S \Rightarrow \mathtt{a}F S\mathtt{d} \Rightarrow \mathtt{a}F\mathtt{cd} \Rightarrow \mathtt{a}F\mathtt{bcd} \Rightarrow \mathtt{a}F\mathtt{bbcd} \Rightarrow \mathtt{abbbcd} = w$

Реализация восходящего анализа при помощи стека (алгоритм типа «перенос-свёртка»)

- Вначале стек пуст.
- Входная цепочка посимвольно переносится в стек до тех пор, пока наверху стека не окажется некоторая основа (правая часть некоторого правила $A \to \beta$).
- Затем в стеке производится свёртка по правилу $A \to \beta$.
- Описанный процесс повторяется, пока вся входная цепочка не перенесена в стек и не сделаны все возможные свёртки после завершения этого переноса.
- Если в стеке находится только аксиома грамматики, то входная цепочка допускается.
- Почему каждая основа окажется наверху стека?
- Соглашение: при записи содержимого стека пишем символ дна ∇ слева, а символ на вершине стека — справа и считаем символ на вершине стека последним символом цепочки, представляющей содержимое стека.

Грамматика $G_1 = \{S \to \mathtt{a}F S\mathtt{d} | \mathtt{c},\ F \to F\mathtt{b} | \mathtt{b}\}$, цепочка $w = \mathtt{abbbcd}$.

Такт	Содержимое стека	Позиция указателя
1	∇	$\diamond\ \mathtt{abbbcd}\ \dashv$
2	$\nabla\mathtt{a}$	$\mathtt{a}\diamond\mathtt{bbbcd}\ \dashv$
3	$\nabla\mathtt{ab}$	$\mathtt{ab}\diamond\mathtt{bbcd}\ \dashv$
4	$\nabla\mathtt{a}F$	$\mathtt{ab}\diamond\mathtt{bbcd}\ \dashv$
5	$\nabla\mathtt{a}F\mathtt{b}$	$\mathtt{abb}\diamond\mathtt{bcd}\ \dashv$
6	$\nabla\mathtt{a}F$	$\mathtt{abb}\diamond\mathtt{bcd}\ \dashv$
7	$\nabla\mathtt{a}F\mathtt{b}$	$\mathtt{abbb}\diamond\mathtt{cd}\ \dashv$
8	$\nabla\mathtt{a}F$	$\mathtt{abbb}\diamond\mathtt{cd}\ \dashv$
9	$\nabla\mathtt{a}F\mathtt{c}$	$\mathtt{abbbc}\diamond\mathtt{d}\ \dashv$
10	$\nabla\mathtt{a}FS$	$\mathtt{abbbc}\diamond\mathtt{d}\ \dashv$
11	$\nabla\mathtt{a}FS\mathtt{d}$	$\mathtt{abbbcd}\diamond\ \dashv$
12	∇S	$\mathtt{abbbcd}\diamond\ \dashv$

Конкатенация содержимого стека и необработанного суффикса цепочки w является r-формой в выводе цепочки w:

$S \Rightarrow \mathtt{a}FS\mathtt{d} \Rightarrow \mathtt{a}F\mathtt{cd} \Rightarrow \mathtt{a}F\mathtt{bcd} \Rightarrow \mathtt{a}F\mathtt{bbcd} \Rightarrow \mathtt{abbbcd} = w$.

Предложение о восходящем анализе при помощи стека

Предложение

Пусть восходящий анализ производится при помощи стека. Тогда верхний символ стека либо входит в основу текущей r-формы, либо входит в текущую r-форму левее этой основы.

Доказательство.

- Индукция по порядковому номеру свёртки.
- База индукции верна: перед первой свёрткой основа находится в стеке и её последний символ находится на вершине стека.
- Индукционный переход. Рассмотрим два идущих подряд применения правил в правом выводе:
  (1) $S \Rightarrow^* \alpha Aw \Rightarrow \alpha\underline{\underline{\beta Bv}}w \Rightarrow \alpha\beta\underline{\gamma}vw$
  или
  (2) $S \Rightarrow^* \alpha BuAw \Rightarrow \alpha Bu\underline{\underline{v}}w \Rightarrow \alpha\underline{\gamma}uvw$.

Предложение о восходящем анализе при помощи стека: окончание доказательства

- В случае (1)

$$S \Rightarrow^* \alpha A w \Rightarrow \alpha \underline{\underline{\beta B v}} w \Rightarrow \alpha \beta \underline{\gamma} v w$$

  сразу после свёртки по правилу $B \to \gamma$ в стеке будет $\alpha\beta B$; верхний символ стека B принадлежит очередной основе $\beta B v$.

- В случае (2)

$$S \Rightarrow^* \alpha B u A w \Rightarrow \alpha B u \underline{\underline{v}} w \Rightarrow \alpha \underline{\gamma} u v w$$

  сразу после свёртки по правилу $B \to \gamma$ в стеке будет αB; верхний символ стека B находится левее очередной основы v.

Два типа алгоритмов восходящего анализа

Мы изучим два типа алгоритмов восходящего анализа.

1. Алгоритмы, основанные на *отношениях предшествования*.
   Такой алгоритм находит основу, анализируя пары соседних символов в текущей цепочке.
2. Алгоритмы *LR-анализа*.
   Для выделения основы используется информация о символах, расположенных до предполагаемой правой границы основы, и информация о k символах, идущих после правой границы (число k заранее фиксировано).

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).

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`).