информатика

Мартыненко Борис Константинович

УДК 21.7.12

КАСКАДНЫЙ МЕТОД РЕГУЛЯРИЗАЦИИ КС-ГРАММАТИК

Аннотация

Описывается метод регуляризации приведённых КС-грамматик без самовставлений, основанный на топологической сортировке нетерминалов грамматики по отношению зависимости, определяемой правилами грамматики.

Ключевые слова: КС-грамматика, регулярное выражение, отношение зависимости, топологическая сортировка нетерминалов, эквивалентное преобразование грамматики.

Определим отношение R зависимости между нетерминалами КС-грамматики следующим образом.

Определение 1. Пусть $G = (V_N, V_T, P, S)$ – КС-грамматика. Будем считать, что нетерминал $A \in V_N$ зависит от $B \in V_N$, если существует правило вида $A \to \alpha B \beta \in P$, где $\alpha, \beta \in V^*$, $V = V_N \cup V_T$. Этот факт будем записывать $(A, B) \in R$, а множество всех таких пар R будем называть *отношением зависимости между нетерминалами* КС-грамматики. Другими словами, $R \subseteq V_N \times V_N$. Когда $A = B$ говорят, что нетерминал A *самозависим.*

Определение 2. Нетерминал $A \in V_N$ назовём *абсолютно независимым*, если не существует никакого $B \in V_N$, чтобы пара $(A, B) \in R$.

Другими словами, абсолютно независимые нетерминалы определяются правилами, в правых частях которых нет ни одного нетерминала.

Наша ближайшая задача разделить всё множество нетерминалов V_N данной приведённой КС-грамматики без самовставлений на такие непересекающиеся подмножества $s_0, s_1, s_2, \ldots, s_m$, $m \le n$, где n – число нетерминалов грамматики, которые обладали бы ниже следующими свойствами.

1. Все нетерминалы $A \in s_0$ абсолютно независимы. Другими словами, регулярные выражения для таких нетерминалов *априори* определены A-правилами, то есть правилами с нетерминалом A в левой части.

2. Нетерминалы любого множества s_l, $1 \le l \le m$, *независимы*, то есть для любой пары нетерминалов $A, B \in s_l$ выполняется $(A, B) \notin R$.

3. Нетерминалы множества s_l, $1 \le l \le m$, *непосредственно* «вычислимы» по регулярным значениям нетерминалов из множеств s_k, $0 \le k \le l - 1$. Другими словами, если $A \in s_l$, то в правой части A-правила все вхождения нетерминалов замещаются регулярными значениями нетерминалов из s_k, $0 \le k \le l - 1$, уже вычисленных к этому моменту.

Метод разбиения множества нетерминалов грамматики на подмножества независимых нетерминалов $s_0, s_1, s_2, \ldots, s_m$ с такими свойствами – это топологическая сортировка множества V_N в отношении R.

Будем говорить, что нетерминалы из множества s_l относятся к уровню l. Очевидно, что на максимальном уровне (m) всегда располагается начальный нетерминал грамматики S и только он, если он *не встречается* в правых частях правил. Это всегда можно гарантировать предварительно.

На уровне 0 находятся нетерминалы, регулярные значения которых уже определены. Их можно назвать *опорными*. Далее, параллельно продвигаясь по уровням, «вычисляются» регулярные значения всех других нетерминалов путём замещения, как указано выше в свойстве 3. В результате такого «каскадного» процесса в последнюю очередь мы получим регулярное выражение для начального нетерминала грамматики S на уровне m. Именно оно и есть искомый результат эквивалентных преобразований исходной КС-грамматики.

Остаётся описать метод топологической сортировки множества нетерминалов грамматики в отношении зависимости R и определить порядок получения регулярных вы-

Пример 1. Грамматика чисел в языке Алгол 68.

Дана КС-грамматика $G = (V_N, V_T, P, S)$, где

V_T = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.', '\' , 'e', '+' ; '-'} = {d, , '.', '\' , 'e', '+' ; '–'}.

$V_N = \{A_{15\{\text{number}\}}, A_{1\{\text{plusminus}\}}, A_{2\{\text{plusminus option}\}}, A_{3\{\text{power of ten}\}}, A_{4\{\text{times ten to the power choice}\}},$
$A_{5\{\text{exponent part}\}}, A_{6\{\text{stagnant part}\}}, A_{7\{\text{floating point numeral}\}}, A_{8\{\text{fractional part}\}}, A_{9\{\text{integral part}\}},$
$A_{10\{\text{integral part option}\}}, A_{11\{\text{variable point numeral}\}}, A_{12\{\text{digit cypher}\}}, A_{13\{\text{digit cypher sequence}\}},$
$A_{14\{\text{fixed point numeral}\}}\}$, $S = A_{15}$,

$P = \{A_{15}\colon A_{14} ; A_{11} ; A_7.$
$A_{14}\colon A_{13}.$
$A_{13}\colon A_{12} ; A_{13}, A_{12}.$
A_{12}: '0'; '1'; '2'; '3'; '4'; '5'; '6'; '7'; '8'; '9'.
$A_{11}\colon A_{10}, A_8.$
$A_{10}\colon A_9 ; \varepsilon.$
$A_9\colon A_{14}.$
A_8: '.' , A_{14}.
$A_7\colon A_6, A_5.$
$A_6\colon A_{14} ; A_{11}.$
$A_5\colon A_4, A_3.$
A_4: '\' ; 'e'.
$A_3\colon A_2, A_{14}.$
$A_2\colon A_1 ; \varepsilon.$
A_1: '+' ; '–'.}

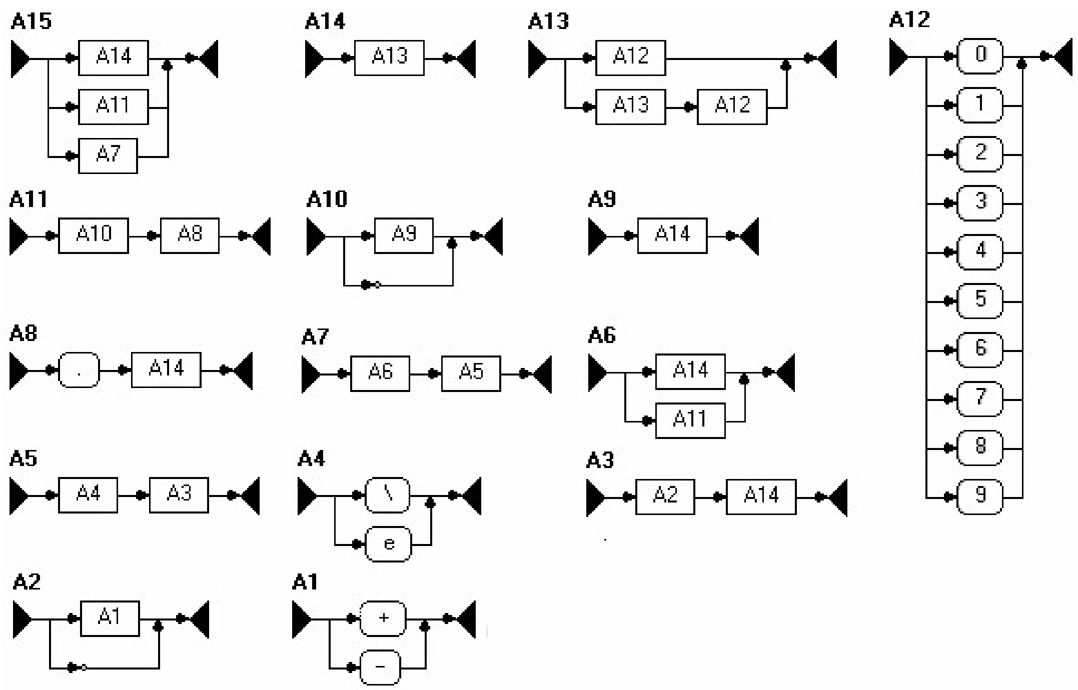

Рис. 1. Синтаксическая граф-схема чисел в языке программирования Алгол 68

ражений для нетерминалов с учётом результата этой сортировки. Лево- и/или правосторонние рекурсии в преобразуемых правилах, в случае их обнаружения, исключаются по образцам $A\colon A\alpha;\,\beta \equiv A\colon (\beta), (\alpha)^*$ или $A\colon \alpha A;\,\beta \equiv A\colon (\alpha)^*(\beta)$ соответственно (см. [2]).

Для иллюстрации будем использовать грамматику чисел Алгола 68 [1], приведённую в [2], с двойственной трактовкой правил, как регулярных формул с операциями объединения и конкатенации [3].

В процессе преобразований правил грамматик могут появляться и другие регулярные операции: два вида замыкания, *рефлексивно-транзитивное замыкание*, обозначаемое «звёздочкой» Клини, и *транзитивное замыкание*, обозначаемое «плюсом» Клини.

Напомним, что все альтернативы для каждого нетерминала собраны в один «пакет» и разделены металингвистическом символом «;», а конкатенируемые символы правых частей – символом «,». Конец пакета отмечен «.».

На рис. 1 эта грамматика представлена в виде многокомпонентной синтаксической граф-схемы, аналога синтаксических диаграмм Н. Вирта.

Отношение зависимости нетерминалов R легко построить непосредственно по правилам грамматики или, глядя на рис. 1, в виде множества пар $R \subseteq V_N \times V_N$.

Получаем $R = \{(A_2, A_1), (A_3, A_2), (A_3, A_{14}), (A_5, A_3), (A_5, A_4), (A_6, A_{11}), (A_7, A_5), (A_7, A_6), (A_8, A_{14}), (A_9, A_{14}), (A_{10}, A_9), (A_{11}, A_8), (A_{11}, A_{10}), (A_{13}, A_{12}), (A_{13}, A_{13}), (A_{14}, A_{13}), (A_{15}, A_7), (A_{15}, A_{11}), (\mathrm{A}_{15}, A_{14})\}$.

Это отношение зависимости R для наглядности можно представить и в виде *матрицы зависимости* (табл. 1). Строки и столбцы этой матрицы «оцифрованы» нетерминальными символами грамматики $(A_1, A_2, ..., A_{15})$. Непустой элемент матрицы, размещающийся i-й строке и j-м столбце, помеченный символом Т (true), означает, что нетерминал A_i зависит от нетерминала A_j (как, например, A_2 зависит от A_1; A_3 зависит от A_2 и A_{14} и т. д.). Пустые строки матрицы (как, например, 1-я, 4-я, 12-я) означают *абсолютную*, то есть *полную независимость* нетерминалов A_1, A_4, A_{12}, которым эти строки соответствуют, от всех других нетерминалов грамматики. Элемент (A_{13}, A_{13}) = Т

Табл. 1. Зависимость между нетерминалами

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_1															
A_2	Т														
A_3		Т												Т	
A_4															
A_5			Т	Т											
A_6											Т				
A_7					Т	Т									
A_8														Т	
A_9														Т	
A_{10}									Т						
A_{11}								Т		Т					
A_{12}															
A_{13}												Т	Т		
A_{14}													Т		
A_{15}							Т				Т			Т	

означает *самозависимость* нетерминала A_{13}, то есть нетерминал A_{13} – (лево-) рекурсивен, поскольку входная грамматика без самовставлений.

Отношение зависимости R можно изобразить и в виде *графа зависимости*, как на рис. 2. Узлы этого графа помечены нетерминалами грамматики, а ориентированные дуги представляют зависимость нетерминалов. Если нетерминал A зависит от B, то от узла, помеченного символом A, проводится дуга к узлу, помеченному символом B.

Читатель вправе выбирать эти или другие представления данных при обдумывании решения поставленной проблемы.

1. МЕТОД ТОПОЛОГИЧЕСКОЙ СОРТИРОВКИ НЕТЕРМИНАЛОВ

Опишем в абстрактных терминах алгоритм топологической сортировки нетерминалов КС-грамматики, исходя из построенного отношения зависимости R, а иллюстрации – в более зримых представлениях.

Цель алгоритма – расположить множество нетерминалов грамматики по уровням таким образом, чтобы все нетерминалы одного уровня l $(1 \le l \le m)$ зависели только от нетерминалов уровня $k < l$ в соответствии с отношением R. Уровень $l = 0$ отводится абсолютно независимым нетерминалам, то есть тем, альтернативы которых представляются терминальными цепочками.

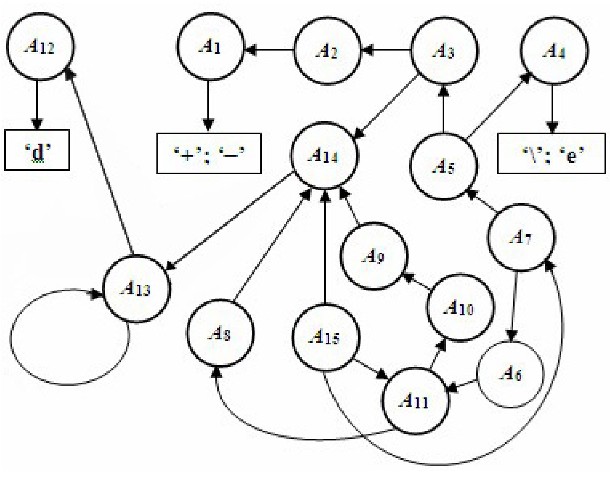

Рис. 2. Граф зависимости нетерминалов КС-грамматики

Алгоритм 1. Топологическая сортировка нетерминалов грамматики

Вход: $G = (V_N, V_T, P, S)$ – приведённая КС-грамматика без самовставлений.

$R \subseteq V_N \times V_N$ – отношение зависимости нетерминалов.

Выход: $N = \{s_0, s_1, s_2, \dots, s_m\}$,

где $s_k \subseteq V_N^+$ – непустое подмножество нетерминалов данной грамматики уровня k, $0 \le k \le m$.

Шаг 0. Расположение на уровне $l = 0$ всех *абсолютно независимых* нетерминалов:
$l = 0; s_0 = \{A \mid \forall\ (A, B \in V_N): (\sim\exists\,(\alpha, \beta \in V^*): A \to \alpha B \beta \in P)\};$

Шаг 1. Построение множества нетерминалов следующего уровня:
$l = l + 1;\ s_l = \{A \mid \forall\ (A \in V_N): (\exists\ B \in V_N): (B \in s_{l-1})\ \&\ (A, B) \in R\ \&\ (A \ne B)\};$

Шаг 2. Исключение из всех подмножеств s_k, где $0 \le k \le l - 1$, нетерминалов, содержащихся в подмножестве s_l:

for $\forall\ (A \in s_l)$:
do for k **from** 0 **to** $l - 1$
  do if $A \in s_k$ **then** $s_k := s_k \setminus \{A\}$ **od**
do;

Шаг 3. Продолжать ли процесс топологической сортировки нетерминалов:

if $s_l \ne \varnothing$ **then goto** Шаг 1 **else goto** Шаг 4;

Шаг 4. Процесс топологической сортировки нетерминалов закончен:

$m = l - 1$; {Максимальный уровень нетерминалов}

Результат: $N = \{s_0, s_1, s_2, \dots, s_m\}$.

В применении к нашему примеру 1 результат получается такой:
$s_0 = \{A_1, A_4, A_{12}\}; s_1 = \{A_2, A_{13}\}; s_2 = \{A_{14}\}; s_3 = \{A_3, A_8, A_9\}; s_4 = \{A_5, A_{10}\}; s_5 = \{A_{11}\}; s_6 = \{A_6\}; s_7 = \{A_7\}; s_8 = \{A_{15}\}$.

Максимальный номер уровня в рассматриваемом примере $m = 8$. На нём всегда располагается начальный нетерминал грамматики $(S = A_{15})$.

Заметим, что условие $A \ne B$, используемое на шаге 1 алгоритма 1, существенно. Оно предотвращает бесконечный рост уровней.

2. КАСКАДНОЕ ВЫЧИСЛЕНИЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Используя результат топологической сортировки нетерминалов и правила исходной КС-грамматики, будем строить регулярные выражения для всех нетерминалов, исходя из следующих соображений.

Вычисление регулярных значений нетерминалов, представленных регулярными выражениями, опирается на *априорные* значения, порождаемые правилами грамматики, альтернативы которых представлены терминалами грамматики. Такие априорные значения всегда существуют, поскольку грамматика приведённая. Все они относятся к нетерминалам уровня 0.

В рассматриваемом примере (см. табл. 2) это нетерминалы A_1, A_4 и A_{12} и соответствующие регулярные выражения для них $\boldsymbol{V_1}$ = ('+'; '–'), $\boldsymbol{V_4}$ = ('\' ; 'e') и $\boldsymbol{V_{12}}$ = (d).

Будем использовать массив V[1.. n], где n – число нетерминалов грамматики для хранения правых частей регуляризованных правил, используя для доступа к его элементам номер нетерминала. Так, в нашем примере V[1] = $\boldsymbol{V_1}$, V[4] = $\boldsymbol{V_4}$, V[12] = $\boldsymbol{V_{12}}$.

Задав регулярные значения для всех нетерминалов уровня 0, мы инициализировали процесс построения регулярных значений нетерминалов уровня l, $1 \le l \le m$, по значениям нетерминалов предыдущих уровней. При этом мы можем быть уверены, благодаря выполненной топологической сортировке нетерминалов, что в массиве V уже существуют регулярные значения нетерминалов, необходимые для вычисления значений нетерминалов уровня l путём замещения вхождений нетерминалов в альтернативы исходных правил КС-грамматики соответствующими регулярными выражениями из массива V.

Обращаясь к примеру, по матрице зависимости (табл. 1) определяем, что от регулярного значения нетерминала A_1, равного $\boldsymbol{V_1}$, зависит значение нетерминала A_2 и только оно, и, стало быть, замещая в правиле для A_2 вхождение A_1 значением $\boldsymbol{V_1}$, получаем регулярное выражение $\boldsymbol{V_2}$ = (('+' ; '–'), ε) в качестве правой части нового правила для нетерминала A_2. В массиве V оно сохраняется под индексом 2: V[2] = $\boldsymbol{V_2}$.

Аналогичные рассуждения приводят к заключению, что регулярное значение $\boldsymbol{V_4}$ = ('\' ; 'e') должно передаваться в правую часть правила для A_5, а $\boldsymbol{V_{12}}$ = (d) – в A_{13}. В результате подстановок получаем регулярные выражения $\boldsymbol{V_5}$ = (('\' ; 'e'), (('+' ; '–'), ε), (d)$^+$) и $\boldsymbol{V_{13}}$ = ((d), (d)*) в качестве правых частей новых правил для нетерминалов A_5 и A_{13} уровня 1. Они также фиксируются в массиве V как элементы с индексами 5 и 13: V[5] = $\boldsymbol{V_5}$, V[13] = $\boldsymbol{V_{13}}$.

Затем мы переходим к вычислению значений нетерминалов 2-го уровня. В нашем примере это A_{14}. После подстановки в исходном правиле для A_{14} вместо A_{13} значения $\boldsymbol{V_{13}}$ из V[13], получаем V_{14} = (((d), (d)*)), которое фиксируется в массиве V : V[14] = $\boldsymbol{V_{14}}$.

Подобным же образом вычисляются регулярные значения нетерминалов следующих уровней вплоть до максимального, на котором находится всегда один – начальный – нетерминал грамматики. Его регуляр-

Табл. 2. Распределение нетерминалов по уровням

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
8															**T**
7							T								
6						T									
5											T				
4					T					T					
3			T					T	T						
2														T	
1		T											T		
0	T			T								T			

ное значение и есть цель преобразований исходной КС-грамматики.

Напомним, что в правилах исходной грамматики подстановки регулярных значений производятся после устранения рекурсии, если она имеет место в данном правиле.

Ход преобразований исходной КС-грамматики примера 1 показан в табл. 3.

В ней представлены исходные правила, относящиеся к нетерминалам последовательных уровней 0, 1, 2, ..., 8, и соответствующие им регулярные выражения. При этом за первыми представлениями этих значений, получаемых автоматически, следуют регулярные выражения, получаемые за счёт искусного (неавтоматического) применения эквивалентных тождеств.

Итак, регуляризованная КС-грамматика, эквивалентная исходной, есть RBNF-грамматика (см. рис. 3), или приведённая к «красивому» виду (см. рис. 4).

Описанный способ получения регулярных выражений на основе топологической сортировки нетерминалов по отношению зависимости между ними естественно назвать каскадным[1] методом регуляризации грамматики. Далее приводятся две синтаксические граф-схемы, соответствующие этим грамматикам (см. рис. 5 и рис. 6).

Заметим, что конечный автомат, построенный по грамматике G' имеет 16 состояний, конечный автомат, построенный по грамматике G'', имеет 9 состояний, а минимальные автоматы после оптимизации со-

G' = ({A_{15}}, {d, , '.', '\' , 'e', '+' ; '–'} ,
{A_{15}: (((d), (d)*) ; (((((d), (d)*)) ; ε), '.', ((d), (d)*)) ; (((d), (d)*), ((((d), (d)*)) ; ε),'.',
((d), (d)*)) , (('\' ; 'ε'), ((((('+' ; “-') , ε), ε)), ((d), (d)*)))).}, A_{15})

Рис. 3.

G'' = ({A_{15}}, {d, , '.', '\' , 'e', '+' ; '–'}, {A_{15}: (d$^+$; d*, '.', d$^+$), [('\' ; 'e'), ['+' ; '–'], d$^+$].}, A_{15})

Рис. 4.

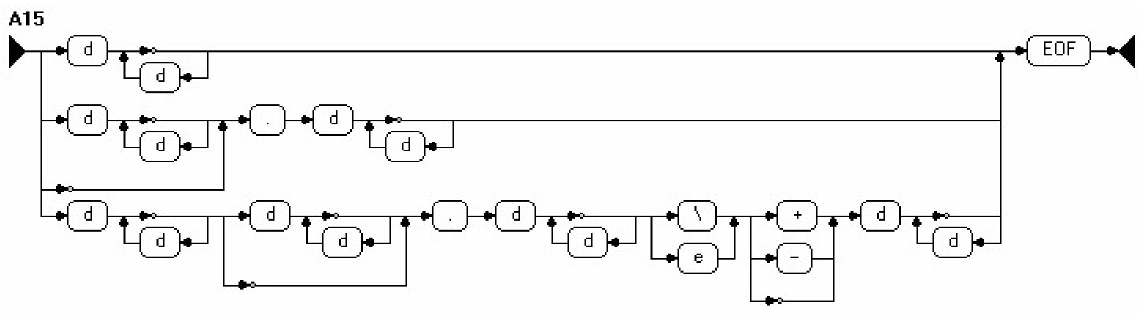

Рис. 5. Синтаксическая граф-схема чисел Алгола 68 – грамматика G'

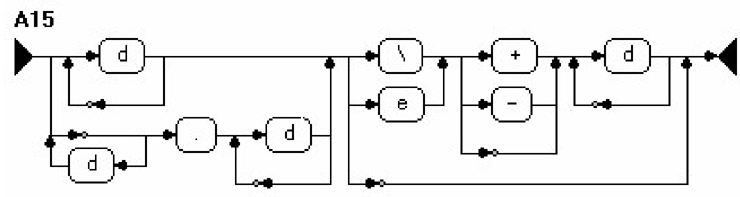

Рис. 6. Синтаксическая граф-схема чисел Алгола 68 – грамматика G''

[1] Метафора, естественная для автора, жителя Петергофа, столицы фонтанов и водяных каскадов.

Табл. 3. Порядок вычислений регулярных значений нетерминалов

Уровень	**ПРАВИЛА**	**РЕГУЛЯРНЫЕ ЗНАЧЕНИЯ**
0	A_1: '+' ; '–'. A_4: '\' ; 'e'. A_{12}: '0'; '1'; '2'; '3'; '4'; '5'; '6'; '7'; '8'; '9'.	$\boldsymbol{V_1}$ = ('+' ; '–') $\boldsymbol{V_4}$ = ('\' ; 'e') $\boldsymbol{V_{12}}$ = (d) Опорные значения
Подстановка значений $\boldsymbol{V_1}$, $\boldsymbol{V_{12}}$ в правые части правил для нетерминалов уровня 1		
1	A_2: A_1 ; ε. $\boldsymbol{A_{13}}$: A_{12} ; $\boldsymbol{A_{13}}$, A_{12} ≡ A_{12}, $(A_{12})^*$.	$\boldsymbol{V_2}$ = (('+' ; '–') , ε) ≡ (['+' ; '–']) $\boldsymbol{V_{13}}$ = ((d), (d)*) ≡ d$^+$ Устранение левой рекурсии
Подстановка значения $\boldsymbol{V_{13}}$ в правую часть правила для нетерминала уровня 2		
2	A_{14}: A_{13}.	$\boldsymbol{V_{14}}$ = ((d), (d)*) ≡ d$^+$
Подстановка значений $\boldsymbol{V_2}$, $\boldsymbol{V_{14}}$ в правые части правил для нетерминалов уровня 3		
3	A_3: A_2 , A_{14}. A_8: '.' , A_{14}. = '.' , ((d), (d)*). A_9: A_{14}.	$\boldsymbol{V_3}$ = ((('+' ; '–'), ε), ((d), (d)*)) ≡ ['+' ; '–'], d$^+$ $\boldsymbol{V_8}$ = '.', ((d), (d)*) ≡ '.', d$^+$ $\boldsymbol{V_9}$ = ((d), (d)*) ≡ d$^+$
Подстановка значений $\boldsymbol{V_3}$, $\boldsymbol{V_9}$ в правые части правил для нетерминалов уровня 4		
4	A_5: A_4 , A_3. A_{10}: A_9 ; ε.	$\boldsymbol{V_5}$ = ('\' ; 'e'), (((('+' ; '–'), ε)), ((d), (d)*)) ≡ ≡ ('\' ; 'e'), ['+' ; '–'], (d)$^+$ $\boldsymbol{V_{10}}$ = ((((d), (d)*)) ; ε) ≡ d*
Подстановка значений $\boldsymbol{V_8}$, $\boldsymbol{V_{10}}$ в правую часть правила для в нетерминала уровня 5		
5	A_{11}: A_{10} , A_8.	$\boldsymbol{V_{11}}$ = (((((d), (d)*)) ; ε), '.', ((d), (d)*)) ≡ ≡ d*, '.', d$^+$
Подстановка значений $\boldsymbol{V_{11}}$, $\boldsymbol{V_{14}}$ в правую часть правила для нетерминала уровня 6		
6	A_6: A_{14} ; A_{11}.	$\boldsymbol{V_6}$ = ((d), (d)*), ((((d), (d)*)) ; ε), '.', ((d), (d)*) ≡ ≡ d$^+$; d*, '. ', d$^+$
Подстановка значений $\boldsymbol{V_5}$, $\boldsymbol{V_6}$ в правую часть правила для нетерминала уровня 7		
7	A_7: A_6 , A_5.	$\boldsymbol{V_7}$ = (((d), (d)*), ((((d), (d)*)) ; ε), '.', ((d), (d)*)), (('\' ; 'e'), (((('+' ; '–'), ε), ((d), (d)*)))) ≡ ≡ (d$^+$; d*, '.', d$^+$), ('\' ; 'e'), ['+' ; '–'], d$^+$
Подстановка значений $\boldsymbol{V_7}$, $\boldsymbol{V_{11}}$, $\boldsymbol{V_{14}}$ в правую часть правила для нетерминала уровня 8		
8	A_{15}: A_{14} ; A_{11} ; A_7.	$\boldsymbol{V_{15}}$ = ((d), (d)*) ; (((((d), (d)*)) ; ε), '.', ((d), (d)*)) ; (((d), (d)*), ((((d), (d)*)) ; ε), '.', ((d), (d)*)) , (('\' ; 'e'), ((((('+' ; '–'), ε), ε)), ((d), (d)*)))) ≡ ≡ d$^+$; d*, '.', d$^+$; (d$^+$; d*, '.', d$^+$), ('\' ; 'e'), ['+' ; '–'], d$^+$ ≡ ≡ (d$^+$; d*, '.', d$^+$), [('\' ; 'e'), ['+' ; '–'], d$^+$]

впадают и имеют 7 состояний (см. табл. 1 в [2]). Это значит, что в SYNTAX-технологии нет нужды заботиться о «красоте» грамматики – любая даёт минимальный конечный автомат. Однако получение «красивой» грамматики автоматически является трудной проблемой, достойной темой дипломной работы для ИТ-студентов.

Замечание. Необходимым и достаточным условием применимости метода каскадной регуляризации КС-грамматики является отсутствие циклов в графе зависимости её нетерминалов.

Циклы, если они существуют, обнаруживаются по ходу алгоритма 1 сами собой, когда ещё не все нетерминалы распределены по уровням, а на шаге 2 в результате исключения нетерминалов текущего верхнего уровня, уже распределённых по всем предыдущим уровням, не остаётся ни одного.

Если угодно, можно использовать алгоритм 1 специально как логическую функцию, которая отвечает на вопрос: существует ли в данном графе цикл или нет. Её значение *истина*, если процесс распределения вершин графа по уровням, заканчивается успешно (все вершины графа распределены), и *ложь*, когда он прерывается на шаге 2.

ЗАКЛЮЧЕНИЕ

В статье приведён алгоритм топологической сортировки по отношению зависимости нетерминалов КС-грамматики. Причём некоторые особенности именно этого отношения явно присутствуют в описании алгоритма сортировки. Однако не трудно его переписать в терминах, независимых от конкретных свойств отношения, по которому производится топологическая сортировка. Эта задача оставляется в качестве упражнения читателю, как и сравнение с алгоритмом на графах, описанному в [5].

Литература

1. Пересмотренное сообщение об Алголе 68. Под редакцией *А. ван Вейнгаарден, Б. Майу, Дж. Пек, К. Костер* и др. М., 1979.

2. *Мартыненко Б.К.* Регулярные языки и КС-грамматики // Компьютерные инструменты в образовании. 2012, № 1. С. 14–20.

3. *Мартыненко Б.К.* Синтаксически управляемая обработка данных. 2-издание, исправленное и дополненное. Ленинград: СПбГУ, 2004.

4. *Мартыненко Б.К.* Языки и трансляции. Ленинград: СПбГУ, 2004.

5. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы. Построение и анализ. МЦНМО, Москва, 1999.

Abstract

The method for the regularization of reduced non-self-embedding context-free grammars, based on the topological sorting of nonterminals of the grammar according the relation of the dependence defined by the grammar rules, is described.

Keywords: context-free grammar, regular expression, dependence relation, topological sorting of nonterminals, equivalent conversion of grammar.

Мартыненко Борис Константинович, доктор физико-математических наук, профессор кафедры информатики математико-механического факультета СПбГУ,

mbk@ctinet.ru