УДК 62.506.2

М. Ф. БОНДАРЕНКО, канд. техн. наук, *Ю. В. ЛОПУХИН*, *А. Ф. ОСЫКА*, *Н. К. СВИНАРЬ*

АЛГОРИТМ ФОРМАЛЬНОГО СПРЯЖЕНИЯ ГЛАГОЛОВ

Цель настоящей работы — построение алгоритмов спряжения глаголов в прошедшем времени. Из грамматики русского языка известно, что в прошедшем времени глагол употребляется в следующих видах числа и рода:

1) единственное число мужского рода;
2) единственное число женского рода;
3) единственное число среднего рода;
4) множественное число.

В работе приводится алгоритм получения форм глагола прошедшего времени из инфинитива. Для описания работы алгоритма введем некоторые определения.

Цепочкой назовем любую конечную упорядоченную последовательность элементов множества A, которое в дальнейшем будем называть алфавитом, а его элементы — литерами. Под длиной цепочки будем понимать количество входящих в нее литер.

Разбиением цепочки d будем называть такую цепочку $d' = t|r$, из которой может быть получена цепочка d путем удаления символа границы «|» и пустых интервалов.

Две цепочки будут тождественно равны, если количество литер в них равно и литеры, стоящие на одинаковых местах, совпадают.

Введем операцию *вложения* V цепочки d_1 в цепочку d_2. Результат такой операции является логическим значением и равен true, если существует разбиение $d_2 = t|d_1$, и false в противном случае. Цепочку t условимся называть остатком (он может быть и пустым). Отношение вложения V обладает следующими свойствами:

1. Рефлексивность $d \mathrm{V} d$ (d вкладывается в d).

2. Антисимметричность $\mathrm{V} \cap \mathrm{V}^{-1} \subseteq E$ (соотношения $d \mathrm{V} f$ и $f \mathrm{V} d$ выполняются одновременно только тогда, когда $d = f$).

3. Транзитивность $\mathrm{V}^2 \subseteq \mathrm{V}$ (если $d \mathrm{V} f$ и $f \mathrm{V} g$, то выполнено и $d \mathrm{V} g$. Отсюда следует по индукции: если $d \mathrm{V} f_1$, $f_1 \mathrm{V} f_2$, ... , $f_{n-1} \mathrm{V} g$, то $d \mathrm{V} g$).

Введенные обозначения и определения несколько отличаются от принятых в литературе по прикладной лингвистике, но полностью соответствуют методу анализа, применяемому в алгоритме.

Работа алгоритма состоит из трех этапов:

1) получение глагола прошедшего времени единственного числа мужского рода;

2) получение глаголов прошедшего времени единственного числа женского и среднего рода и множественного числа;

3) грамматическая корректировка полученных глагольных форм.

На каждом этапе на вход подается входная цепочка, представляющая собой инфинитив глагола русского языка, а на выходе получаем выходную цепочку. Входная и выходная цепочки слагаются из литер алфавита A, состоящего из букв русского алфавита и некоторого вспомогательного символа «—», заменяемого в дальнейшем на одну из букв русского алфавита.

Каждый из этапов работы алгоритма связан с проверкой на последовательное вложение цепочек из конечного словаря S_i в подаваемую на вход цепочку (индекс i указывает номер словаря и этапа, на котором он применяется).

Опишем подробнее структуру словарей, используемых в работе алгоритма. Словарь S_1 состоит из пар характерных окончаний инфинитивов глаголов и глагольных форм прошедшего времени мужского рода. Термин окончания отличен от грамматического; под ним подразумевается некоторая подцепочка, находящаяся в конце слова. Словарь S_2 состоит из пар окончаний глаголов прошедшего времени единственного числа мужского рода и некоторого обобщенного окончания глагольных форм прошедшего времени единственного числа женского и среднего рода и множественного числа, из которого может быть получено любое из перечисленных окончаний путем замены символа «—» в конце выходной цепочки на буквы *а*, *о*, *и* соответственно. Словарь S_3 содержит пары окончаний тех глаголов прошедшего времени, которые правильно не могут быть получены из инфинитива обычным путем. Появление словаря S_3 связано с нерегулярностью спряжения русских глагольных словоформ. Словари S_1, S_2, S_3 приведены в приложении.

Отметим особенность построения словарей. В словаре S_i последовательность пар цепочек (s'_{ij}, s''_{ij}) расположена в порядке убывания длин цепочек s'_{ij}. Это позволяет избежать неправильных замен, так как могут найтись две цепочки s'_{ij_1} и s'_{ij_2} такие, что цепочка s_{ij_2} окажется частью цепочки s'_{ij_1}.

Работу каждого этапа алгоритма можно представить в виде процедуры (рис. 1), параметрами которой являются входная v и выходная w цепочки и словарь $S_i = \{s'_{ij}, s''_{ij}\}$ $(i = 1, 2, 3;\ j = 1, 2, \ldots, n_i)$. Блок-схема алгоритма представлена на рис. 2.

Работу i-го этапа алгоритма можно представить следующим образом:

1. $j := 1$ (выбираем первую пару окончаний).
2. Производим операцию вложения цепочки s'_{ij} в цепочку v.
3. Если результат операции true, то перейти к п. 4, иначе к п. 6.
4. Формируем выходную цепочку w, состоящую из остатка t входной цепочки v и приписанной справа цепочки s''_{ij}.

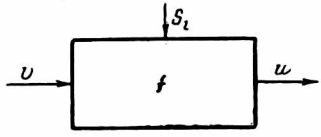

Рис. 1.

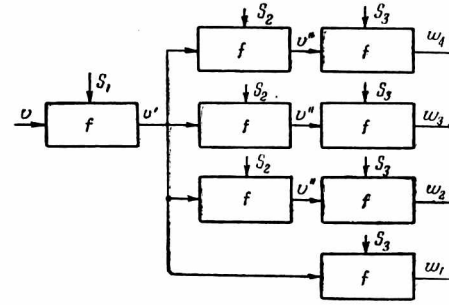

Рис. 2.

5. Выход из блока.
6. $j := j + 1$ (выбираем следующую пару окончаний словаря).
7. Если $j \leqslant n_i$, то перейти к п. 2, иначе к п. 5.

Приведенный алгоритм был реализован на ЭВМ «Урал-14Д» с использованием алгоритмического языка АЛГОЛ-ЦЭМИ. Выбор именно этого языка объясняется наличием ряда расширений (по сравнению с эталонным языком АЛГОЛ-60), позволяющих производить обработку текстовой информации.

Перечислим основные из них.

1. Введены текстовые величины и действия над ними.
2. Значение текстовых величин — последовательность литер. Литерой является буква, цифра, знак операции, разделители и некоторые специальные знаки. Каждая литера в строке занимает отдельную позицию. Позиции считаются занумерованными слева направо. Количество литер в текстовом значении называется длиной. Допускаются пустые строки, не содержащие ни одной литеры. В качестве изображения текстовой константы используется строка литер, ограниченная кавычками для строк, имеющих вид: « ' » (верхняя левая дуга) и « ' » (верхняя правая дуга).
3. В число описаний языка входит текстовый описатель, представляющий собой символ text, за которым может следовать взятое в круглые скобки целое без знака. Целое без знака не должно превышать 128.

Словарь S_1

j	s'_{1j}	s''_{1j}
1	торгнуть	торг
2	клизнуть	клиз
3	дерзнуть	дерзнул
4	жолкнуть	жолк
5	молкнуть	молк
6	меркнуть	мерк
7	горкнуть	горк
8	брякнуть	брякнул
9	звякнуть	звякнул
10	хрипнуть	хрип
11	-терпнуть	-терп
12	креснуть	крес
13	глохнуть	глох
14	дряхнуть	дрях
15	рубнуть	рубнул
16	лебнуть	лебнул
17	бегнуть	бег
18	юзгнуть	юзг
19	тигнуть	тиг
20	ергнуть	ерг
21	чезнуть	чез
22	лекнуть	лек
23	никнуть	ник
24	мокнуть	мок
25	выкнуть	вык
26	репнуть	реп
27	липнуть	лип
28	сипнуть	сип
29	-гаснуть	-гас
30	виснуть	вис
31	киснуть	кис
32	пахнуть	пах
33	чахнуть	чах
34	тихнуть	тих
35	сохнуть	сох
36	-бухнуть	-бух
37	пухнуть	пух
38	стынуть	стыл
39	скрести	скреб
40	рзнуть	рз
41	узнуть	уз
42	язнуть	яз
43	якнуть	як
44	сереть	серел
45	беречь	берег
46	облечь	облек
47	грести	греб
48	бнуть	б
49	ереть	ер
50	влечь	влек
51	бречь	брег
52	расти	рос
53	нести	нес
54	печь	пек
55	речь	рек
56	сечь	сек
57	течь	тек
58	лочь	лок
59	идти	тел
60	есть	ел
61	юсти	юл
62	йти	шел
63	зти	з
64	сти	с
65	сть	л
66	зть	з
67	ти	л
68	ть	л
69	чь	р

Словарь S_2

j	s''_{2j}	s''_{2j}
1	л	л-
2	б	бл-
3	г	гл-
4	з	зл-
5	к	кл-
6	п	пл-
7	р	рл-
8	с	сл-
9	х	хл-

Словарь S_3

j	s'_{3j}	s''_{3j}
1	терпнул-	терпл-
2	толокл-	толкл-
3	бжегл-	божгл-
4	джегл-	дожгл-
5	зжегл-	зожгл-
6	сжегл-	сожгл-
7	тжегл-	тожгл-
8	жегл-	жгл-
9	бчел-	бочл-
10	дчел-	дочл-
11	зчел-	зочл-
12	счел-	сочл-
13	тчел-	точл-
14	чел-	чл-
15	тел-	тл-

Переменные, элементы массива, идентификаторы процедур или формальные параметры, которым описанием или спецификацией задан тип text (<целое без знака>), могут принимать лишь текстовые значения, длина которых не превышает величины целого без знака. Отсутствие спецификации длины означает максимальную длину, равную 128.

4. Имеются операции соединения текстовых значений и выделения части текстового значения. Результатом соединения двух текстовых значений является новая строка, составленная последовательно из литер первого и второго текстовых значений. Операция соединения текстовых значений представляется знаком «|» (вертикальная черта). Выделение части первичного текстового выражения, каким является строка, переменная, указатель функции или текстовое выражение, взятое в круглые скобки, осуществляется с помощью выделителя, следующего за этим первичным текстовым выражением и имеющего вид from < первичное индексное выражение > thru <индексное выражение>. Действие выделителя состоит в выделении подпоследовательности литер из значения предшествующего ему текстового выражения: выделяются литеры, начиная с занимающей позицию под номером, определяемым значением первого индексного выражения выделителя и кончая литерой, занимающей позицию под номером, определяемым значением второго индексного выражения включительно.

Одно из выражений выделителя вместе с предшествующим ему символом from или thru может быть опущено. Пустое начало выделителя эквивалентно началу выделителя from 1. Пустой конец выделителя эквивалентен концу выделителя thru n, где n — длина значения текстового выражения.

5. Текстовые значения могут быть операндами операции отношения равенства и неравенства.

Приведенный выше алгоритм работы блоков реализован в виде следующей процедуры:

```
procedure f (v, w, S, n); value n;
text (20) v, w; text (20) array (2) S; integer n;
begin
  integer i, k, l;
    for k := 1 step 1 until n do
    begin
      l := if S [k, 1] thru 1 = '—' then 2 else 1;
      for i := l step 1 until 20 do
      if (v from i) = (S [k, 1] from l thru 20 — i + l) then
      begin
        w := (v thru i — 1) | S [k, 2];
        go to exit
      end
    end;
exit : end f.
```

Формальными параметрами процедуры являются: v — входная цепочка символов; w — выходная цепочка символов; S — идентификатор словаря, представляющего собой двумерный текстовый массив; n — размерность словаря (количество входящих в него пар цепочек).

ЛИТЕРАТУРА

1. Грамматика русского языка. Т. 1. Под ред. В. В. Виноградова. М., Изд-во АН СССР, 1960. 720 с.
2. Транслятор АЛГОЛ-ЦЭМИ для ЭВМ «Урал-14». Инструкция. М., 1971. 24 с. Авт.: К. С. Кузьмин, М. Р. Левинсон, И. В. Максимова, А. В. Юнисова.