

## МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОПРЕДЕЛЕНИЯ ГРАММАТИЧЕСКОЙ КАТЕГОРИИ РОДА ГЛАГОЛОВ РУССКОГО ЯЗЫКА

*Ю. П. Шабанов-Кушнаренко, М. Ф. Бондаренко,  
Е. А. Соловьева*

Человек способен определять грамматическую категорию рода глаголов русского языка на основании правил морфологии [1, 2]. Если предложить испытуемому какую-либо форму глагола, то он сможет определить, входит ли эта форма в грамматическую категорию рода, а если входит, то какое именно частное грамматическое значение рода она имеет.

Целью данной работы являлось составление математической модели способности человека определять род глагола. Эта модель получена в виде алгоритма, блок-схема которого приведена на рис. 1.

Множество входных сигналов  $X = \{x_1, \dots, x_n\}$  составляют все глаголы русского языка, входящие в словарь [3], и их формы, представляющие собой одно слово. Множество выходных сигналов  $Y = \{y_1, \dots, y_m\}$  в данном случае состоит из четырех элементов. Каждому из выходных сигналов  $y_1, y_2, y_3$  соответствует частное грамматическое значение рода — мужского ( $y_1$ ), женского ( $y_2$ ), среднего ( $y_3$ ), а выходному сигналу  $y_4$  — признак отсутствия рода у глагольной формы. Подавая на вход  $B$  алгоритма глагольную форму  $x_i$ , в результате работы алгоритма на его выходе  $\mathcal{G}$  получаем сигнал-признак  $y_i$ :  $y_1$ , если глагол стоит в мужском роде,  $y_2$  — в женском,  $y_3$  — в среднем,  $y_4$  — если глагольная форма не входит в грамматическую категорию рода.

Блок-схема алгоритма включает в себя элементарные блоки (распознаватели и операторы) и составные (блоки 1, 2), которые, в свою очередь, состоят из элементарных. Распознаватель  $\Phi_1$

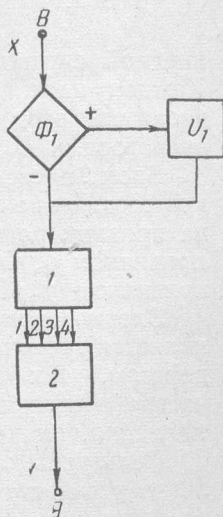


Рис. 1.

проверяет, удовлетворяет ли слово, поданное на его вход, какому-либо условию. В случае выполнения этого условия слово выходит из распознавателя по стрелке, отмеченной знаком плюс, при невыполнении — по стрелке со знаком минус. Распознаватель  $\Phi_1$  проверяет две последние буквы слова на «ся» или «сь». Блок  $U_k$ , который каким-то образом изменяет поступающий на его вход сигнал, т. е. выполняет над этим сигналом определенную операцию, является оператором. Оператор  $U_1$  отбрасывает две последние буквы поступающего на его вход слова.

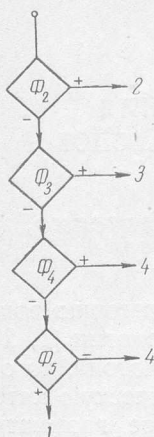


Рис. 2.

Блок 1 является основным блоком алгоритма. Он классифицирует слово (т. е. в данном случае направляет его на один из четырех пронумерованных выходов), основываясь на том, имеет ли оно род, а если имеет, то в зависимости от того, какой именно. Блок 2 формирует выходной сигнал. Рассмотрим подробнее составные блоки 1 (рис. 2) и 2 (рис. 3).

Входящие в состав блока 1 распознаватели выполняют проверку следующих условий:  $\Phi_2$  проверяет конец слова на «ла»,  $\Phi_3$  — на «ло»,  $\Phi_4$  — на совпадение со словом «ляг», которое является исключением, а  $\Phi_5$  — на одну из букв «л», «с», «з», «п», «б», «г», «к», «р», «х». Блок 1 не изменяет

входной сигнал, так как состоит только из распознавателей. Каждая из цифр 1, 2, 3 или 4 на выходе любого блока означает, что этот выход должен быть соединен соответственно с операторами  $U_2$ ,  $U_3$ ,  $U_4$  или  $U_5$  (они входят в состав блока 2).

Блок 2 представляет собой параллельное соединение операторов замены слова на признак рода  $y_i$ :  $U_2$  заменяет слово признаком  $y_1$ ,  $U_3$  — признаком  $y_2$ ,  $U_4$  — признаком  $y_3$ ,  $U_5$  — признаком  $y_4$ .

Входным при решении данной задачи является русский алфавит, расширенный дефисом, так как последний встречается в глаголах повелительного наклонения, оканчивающихся на «-ка».

Пользуясь классификацией, предложенной в работе [4], можно легко установить, что для нашей конкретной задачи характерно разделение глагольных форм на два типа: а) классифицирующиеся на основании формальных признаков; б) классифицирующиеся при условии известности их лексических значений. Небольшое количество глагольных форм типа б), а также отсутствие глагольных форм типа в) (классифицирующихся только при наличии контекста) в данной задаче определяют сравнительную легкость ее решения. Не случайно что словарь исключений состоит из одного слова, т. е. фактически отсутствует.

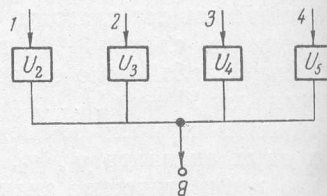


Рис. 3.

На выбранном множестве входных слов алгоритм работает безошибочно. Если же расширить это множество (словами или псевдословами), то погрешность работы алгоритма все равно будет ничтожно мала.

Составленный алгоритм реализован на ЭЦВМ «Минск-22». Полученная программа отлажена на различных массивах глагольных форм. Предложенный алгоритм может найти применение при решении различных задач классификации и представить самостоятельный интерес.

#### ЛИТЕРАТУРА

1. Грамматика современного русского литературного языка. М., «Наука», 1970. 767 с.
2. Грамматика русского языка, т. I. М., изд-во АН СССР, 1960. 719 с.
3. Орфографический словарь русского языка, изд. 11-е, М., «Сов. энциклопедия», 1971. 520 с.
4. Бондаренко М. Ф., Соловьева Е. А. Методы решения задач морфологической и субморфологической классификации. Сб. «Проблемы бионики», вып. 10. Изд-во Харьковск. ун-та, 1973, с. 145—149.

### НОРМАЛИЗАЦИЯ ИЗОБРАЖЕНИЙ ПРИ ИЗМЕНЕНИИ ЯРКОСТИ И КОНТРАСТНОСТИ

*Е. П. Путятин, Б. К. Лопатченко, В. Б. Левиков,  
Б. Я. Сердюченко*

Распознавание человеком зрительных картин обусловлено не абсолютным распределением лучистой яркости зрительной картины, а относительными различиями яркостей отдельных точек картины. В частности, пропорциональное изменение яркости зрительной картины и ее контрастность не влияют в широких пределах на результат узнавания.

Для зрительных картин, заданных в виде функций распределения яркости в поле зрения  $B(x, y)$ , эти изменения соответствуют преобразованию

$$B(x, y) = kB_0(x, y) + c, \quad (1)$$

где  $B_0(x, y)$  — изображение с эталонной яркостью и контрастностью;

$k, c$  — произвольные постоянные.

Нормализовать такие картины — значит построить некоторый оператор  $F(B)$ , действующий из пространства изображений снова в пространство изображений, и такой, что для каждого изображения из класса эквивалентности, образованного соотношением (1), будет выполняться тождество

$$F[B(x, y)] = B_0(x, y). \quad (2)$$