ISSN 0555-2656.

ПРОБЛЕМЫ БИОНИКИ

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ УССР
ХАРЬКОВСКИЙ ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ

ПРОБЛЕМЫ БИОНИКИ

ВЫПУСК 23

Республиканский
межведомственный
научно-технический
сборник

Основан в 1968 г.

ХАРЬКОВ
ИЗДАТЕЛЬСТВО ПРИ ХАРЬКОВСКОМ
ГОСУДАРСТВЕННОМ УНИВЕРСИТЕТЕ
ИЗДАТЕЛЬСКОГО ОБЪЕДИНЕНИЯ «ВИЩА ШКОЛА»
1979

32.818
П78

УДК 007:573.6

Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник.— Харьков: Вища школа. Изд-во при Харьк. ун-те, 1979.— 124+4 с.

В сборнике представлены результаты бионических исследований процессов переработки информации в сенсорных каналах и нейронных структурах. Рассмотрены некоторые аспекты интеллектуальной деятельности, средства формального описания этой деятельности и способы ее воспроизведения в искусственных системах отображения и управления. Затронуты вопросы формализации и экспликации понятий естественного языка, изложен подход к математическому моделированию процессов обработки словесной информации. Разрабатывается математическое и информационное обеспечение, пригодное для моделирования и машинной имитации психических функций человека.

Для научных работников и специалистов, занимающихся бионическими исследованиями с привлечением средств и методов кибернетики и вычислительной техники.

Списки лит. в конце статей.

Адрес редакционной коллегии:

310218, Харьков-218, пр. Ленина, 14. институт радиоэлектроники тел. 40-96-45

Редакция естественнонаучной литературы

П $\frac{30501-044}{М226(04)-79}$ 478—79 1502000000

УДК 62.506.2

Ю. П. ШАБАНОВ-КУШНАРЕНКО, д-р техн. наук

МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ КОНЕЧНЫХ ЛОГИЧЕСКИХ ОБЪЕКТОВ

Средствами алгебры конечных предикатов [1—4] здесь описываются простейшие отношения, задаваемые на конечных множествах. Введем конечный алфавит E, содержащий все интересующие нас знаки. Алфавит E принимаем в качестве алфавита используемой нами в дальнейшем алгебры конечных предикатов. Пусть $a_1, a_2, \ldots, a_p$ — буквы алфавита E; p — число всех букв алфавита E. Отношение равенства $x = y$ произвольных букв x, y алфавита E может быть записано на языке алгебры конечных предикатов следующим образом:

$$x = y \underset{d}{=} x^{a_1}y^{a_1} \vee x^{a_2}y^{a_2} \vee \ldots \vee x^{a_p}y^{a_p}. \qquad (1)$$

Знак $\underset{d}{=}$ означает тождество формул по определению. Тождество (1) вводит *предикат равенства букв* $x = y$, обращающийся в единицу, когда буквы x и y совпадают, и обращающийся в нуль — в противном случае. Отношение неравенства $x \neq y$ букв x, y алфавита E может быть описано в виде

$$x \neq y \underset{d}{=} \overline{x = y}. \qquad (2)$$

Тождество (2) вводит *предикат неравенства букв* $x \neq y$, обращающийся в единицу, когда буквы x и y не совпадают, и обращающийся в нуль, когда они совпадают. Заметим, что равенство $x = \sigma$, где σ — фиксированная буква алфавита E, формально может быть записано, согласно выражению (1), в виде x^σ, а неравенство $x \neq \sigma$, согласно формуле (2), — в виде $\overline{x^\sigma}$.

Пусть $A = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ — некоторое фиксированное подмножество алфавита E. *Отношение* $x \in \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ *принадлежности* произвольной буквы x алфавита E множеству A может быть представлено в виде

$$x \in \{\sigma_1, \sigma_2, \ldots, \sigma_k\} \underset{d}{=} x^{\sigma_1} \vee x^{\sigma_2} \vee \ldots \vee x^{\sigma_k}. \qquad (3)$$

Принадлежность элемента x пустому множеству $\varnothing$ записываем в виде предиката, тождественно равного нулю:

$$x \in \varnothing \underset{d}{=} 0. \qquad (4)$$

Тождествами (3) и (4) мы вводим *предикат принадлежности* элемента x множеству A, обращающийся в единицу, если $x \in A$, и обращающийся в нуль, если $x \mathrel{\overline{\in}} A$. *Предикат принадлежности* $x \mathrel{\overline{\in}} A$ элемента x множеству A вводим тождеством

$$x \mathrel{\overline{\in}} A \underset{d}{\equiv} \overline{x \in A}. \tag{5}$$

Пусть $P(x)$ — некоторый фиксированный конечный предикат, заданный на множестве E. Введем для этого предиката *квантор общности* $\forall x P(x)$ и *квантор существования* $\exists x P(x)$, задавая их следующими формулами алгебры конечных предикатов:

$$\forall x P(x) \underset{d}{\equiv} P(a_1) \wedge P(a_2) \wedge \ldots \wedge P(a_p); \tag{6}$$

$$\exists x P(x) \underset{d}{\equiv} P(a_1) \vee P(a_2) \vee \ldots \vee P(a_p). \tag{7}$$

Квантор общности равен единице, когда $P(x) = 1$ для всех x из E, в противном случае он равен нулю. Квантор существования равен нулю, когда $P(x) = 0$ для всех x из E, в противном случае он равен единице.

Пусть A и B — фиксированные подмножества алфавита E. Выражение $A = B$, обозначающее *равенство множеств* A и B, может быть представлено в виде такой формулы алгебры конечных предикатов:

$$A = B \underset{d}{\equiv} \forall x (x \in A \sim x \in B). \tag{8}$$

Выражение $A \neq B$, обозначающее *неравенство множеств* A и B, вводим с помощью предиката

$$A \neq B \underset{d}{\equiv} \overline{A = B}. \tag{9}$$

Выражение $A \subseteq B$, обозначающее *включение множества* A в множество B, вводим с помощью предиката

$$A \subseteq B \underset{d}{\equiv} \forall x (x \in A \supset x \in B). \tag{10}$$

Дополнение $\tilde{A}$ *множества* A, а также *объединение* $A \cup B$, *пересечение* $A \cap B$, *разность* $A \setminus B$ и *симметрическая разность* $A \dot{-} B$ *множеств* A и B вводим следующими тождествами:

$$x \in \tilde{A} \underset{d}{\equiv} x \mathrel{\overline{\in}} A; \tag{11}$$

$$x \in A \cup B \underset{d}{\equiv} x \in A \vee x \in B; \tag{12}$$

$$x \in A \cap B \underset{d}{\equiv} x \in A \wedge x \in B; \tag{13}$$

$$x \in A \setminus B \underset{d}{\equiv} x \in A \wedge x \in \tilde{B}; \tag{14}$$

$$x \in A \dot{-} B \underset{d}{\equiv} x \in A \setminus B \vee x \in B \setminus A. \tag{15}$$

Рассмотрим примеры логических задач, решаемых формально с помощью введенных определений.

Пример 1. Доказать, что $A \cap B \subseteq A$. Имеется в виду, что A и B — некоторые фиксированные подмножества алфавита E.

Решение. $A \cap B \subseteq A \equiv \forall x\,(x \in A \cap B \supset x \in A) \equiv \forall x\,(x \in A \wedge x \in B \supset x \in A) \equiv \forall x\,(\overline{x \in A \wedge x \in B} \vee x \in A) \equiv \forall x\,(\overline{x \in A} \vee \overline{x \in B} \vee x \in A) \equiv 1$ ($\equiv$ означает тождественное равенство формул алгебры конечных предикатов).

Пример 2. Доказать, что $A \setminus B = A \dot{-} (A \cap B)$.

Решение. $A \setminus B = A \dot{-} (A \cap B) \equiv \forall x\,(x \in A \setminus B \sim x \in A \dot{-} (A \cap B)) \equiv \forall x\,(x \in A \setminus B \sim x \in A \setminus (A \cap B) \vee x \in (A \cap B) \setminus A) \equiv \forall x\,(x \in A \setminus B \sim x \in A \wedge \overline{x \in A \cap B} \vee x \in A \cap B \wedge \overline{x \in A}) \equiv \forall x\,(x \in A \wedge \overline{x \in B} \sim x \in A \wedge (\overline{x \in A} \vee \overline{x \in B}) \vee x \in A \wedge x \in B \wedge \overline{x \in A}) \equiv \forall x\,(x \in A \wedge \overline{x \in B} \sim x \in A \wedge \overline{x \in B}) \equiv 1.$

Пример 3. Задана система уравнений $A \cap X = B$; $A \cup X = C$, где X — неизвестное подмножество алфавита A. Известно, что $A = \{a_1, a_2\}$, $B = \{a_1\}$, $C = \{a_1, a_2, a_3\}$. Найти X.

Решение. Полагаем $x \in X \equiv \alpha_1 x^{a_1} \vee \alpha_2 x^{a_2} \vee \ldots \vee \alpha_p x^{a_p}$. Здесь α_1, $\alpha_2, \ldots, \alpha_p$ — некоторые булевы константы, значения которых определяют состав множества X. Когда $\alpha_i = 1$, то элемент a_i принадлежит множеству X, когда же $\alpha_i = 0$, то элемент a_i множеству X не принадлежит. Имеем $(A \cap X = B)\,(A \cup X = C) \equiv \forall x\,(x \in A \wedge x \in X \sim x \in B)\,(x \in A \vee x \in X \sim x \in C) \equiv \forall x\,\left((x^{a_1} \vee x^{a_2})\left(\alpha_1 x^{a_1} \vee \alpha_2 x^{a_2} \vee \ldots \vee \alpha_p x^{a_p}\right) \sim x^{a_1}\right) \wedge \left(x^{a_1} \vee x^{a_2} \vee \alpha_1 x^{a_1} \vee \alpha_2 x^{a_2} \vee \ldots \vee \alpha_p x^{a_p} \sim x^{a_1} \vee x^{a_2} \vee x^{a_3}\right) \equiv (\alpha_1 \sim 1)\,(\alpha_2 \sim 0)\,(\alpha_3 \sim 1)\,(\alpha_4 \sim 0) \ldots (\alpha_p \sim 0) \equiv \alpha_1 \bar{\alpha}_2 \alpha_3 \bar{\alpha}_4 \ldots \bar{\alpha}_p = 1.$ Таким образом, $\alpha_1 = \alpha_3 = 1$, $\alpha_2 = \alpha_4 = \ldots = \alpha_p = 0$. Искомое множество имеет вид $X = \{a_1, a_2\}$.

Перейдем к математическому описанию отношений. Пусть $x_1, x_2, \ldots, x_n$ — переменные алгебры конечных предикатов, n — число всех переменных. Фиксированное *отношение* Q на n-й декартовой степени E^n алфавита E будем записывать в виде индивидуального предиката $P(x_1, x_2, \ldots, x_n)$, обращающегося в единицу, если слово $(x_1, x_2, \ldots, x_n)$ принадлежит отношению Q, и обращающегося в нуль в противном случае.

$$\text{Полагаем } (x_1, x_2, \ldots, x_n) \in Q \underset{d}{\equiv} P(x_1, x_2, \ldots, x_n), \tag{16}$$

$$\text{где } P(x_1, x_2, \ldots, x_n) = \begin{cases} 1, & \text{если } (x_1, x_2, \ldots, x_n) \in Q; \\ 0, & \text{если } (x_1, x_2, \ldots, x_n) \overline{\in} Q. \end{cases} \tag{17}$$

Область изменения D_i значений переменной $x_i\,(1 \leqslant i \leqslant n)$ отношения Q может быть записана в виде

$$x_i \in D_i \underset{d}{\equiv} \exists x_1 \exists x_2 \ldots \exists x_{i-1} \exists x_{i+1} \ldots \exists x_n\,((x_1, x_2, \ldots, x_n) \in Q). \tag{18}$$

Дополнение $\bar{Q}$ отношения Q, а также объединение $Q \cup R$, пересечение $Q \cap R$, разность $Q \setminus R$ и симметрическая разность $Q \dot{-} R$

отношений Q и R на языке алгебры конечных предикатов могу[т] быть представлены в виде

$$\xi \in \bar{Q} \underset{d}{\equiv} \overline{\xi \in Q}; \tag{19}$$

$$\xi \in Q \cup R \underset{d}{\equiv} \xi \in Q \vee \xi \in R; \tag{20}$$

$$\xi \in Q \cap R \underset{d}{\equiv} \xi \in Q \wedge \xi \in R; \tag{21}$$

$$\xi \in Q \setminus R \underset{d}{\equiv} \xi \in Q \wedge \xi \in \bar{R}; \tag{22}$$

$$\xi \in Q \dot{-} R \underset{d}{\equiv} \xi \in Q \setminus R \vee \xi \in R \setminus Q. \tag{23}$$

Здесь $\xi = (x_1, x_2, \ldots, x_n)$.

Пусть $A_1, A_2, \ldots, A_s$ — фиксированные подмножества алф[а]вита E; $A_1 \times A_2 \times \ldots \times A_s$ — декартово произведение этих по[д]множеств; $\xi = (x_1, x_2, \ldots, x_s)$ — произвольное слово из E^s. Пол[а]гаем

$$\xi \in A_1 \times A_2 \times \ldots \times A_s \underset{d}{\equiv} x_1 \in A_1 \wedge x_2 \in A_2 \wedge \ldots \wedge x_s \in A_s. \tag{24}$$

Пусть $\xi S \eta$ — бинарное отношение, заданное на декартово[м] произведении $E^k \times E^l$, элементами которого служат пары (ξ, η) слов $\xi = (x_1, x_2, \ldots, x_n)$, $\eta = (y_1, y_2, \ldots, y_l)$ длины k и l. Н[а] языке алгебры конечных предикатов отношение S может быт[ь] записано в виде предиката

$$\xi S \eta \underset{d}{\equiv} (\xi, \eta) \in S \underset{d}{\equiv} H(x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_l), \tag{25}$$

определяемого следующим образом:

$$H(x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_l) = \begin{cases} 1, & \text{если } \xi S \eta; \\ 0, & \text{если } \xi \tilde{S} \eta. \end{cases} \tag{26}$$

Пусть S_1 и S_2 — бинарные отношения, заданные соответс[т]венно на $E^k \times E^l$ и $E^l \times E^m$. Произведение этих отношен[ий] $S_1 \bigcirc S_2$, определяемое на $E^k \times E^m$, запишем в виде

$$\xi S_1 \bigcirc S_2 \zeta \underset{d}{\equiv} \exists \eta (\xi S_1 \eta \wedge \eta S_2 \zeta). \tag{27}$$

Здесь $\xi = (x_1, x_2, \ldots, x_k)$; $\eta = (y_1, y_2, \ldots, y_l)$; $\zeta = (z_1, z_2, \ldots, z_m)$; кроме того, с целью сокращения записи, принято $\exists \eta = \exists y_1 \exists y_2 \ldots \exists y_l$.

Условие функциональности отношения S на $E^k \times E^l$ запи[]шется в виде следующего уравнения алгебры конечных пред[и]катов:

$$\forall \xi \forall \eta \forall \zeta (\xi S \eta \wedge \xi S \zeta \supset \eta = \zeta). \tag{28}$$

Здесь принято $\forall \xi = \forall x_1 \forall x_2 \ldots \forall x_k$; $\forall \eta = \forall y_1 \forall y_2 \ldots \forall y_l$; $\forall \zeta = \forall z_1 \forall z_2 \ldots \forall z_l$.

Равенство $\eta = \zeta$ слов η и ζ, фигурирующее в формуле (28), может быть записано на языке алгебры конечных предикатов:

$$\eta = \zeta \underset{d}{\equiv} (y_1 = z_1)(y_2 = z_2) \ldots (y_l = z_l). \tag{29}$$

Пусть S и T — фиксированные отношения на $E^k \times E^l$. Выражение $S \subseteq T$, обозначающее включение отношения S в отношение T, вводим с помощью предиката:

$$S \subseteq T \underset{d}{\equiv} \forall\xi\forall\eta\,(\xi S\eta \supset \xi T\eta). \tag{30}$$

Выражение $S = T$, обозначающее равенство отношений S и T, запишем в виде

$$S = T \underset{d}{\equiv} S \subseteq T \wedge T \subseteq S. \tag{31}$$

Неравенство отношений $S \neq T$ представляем так:

$$S \neq T \underset{d}{\equiv} \overline{S = T}. \tag{32}$$

Пусть S — некоторое отношение на $E^k \times E^l$. Отношение S^{-1} на $E^l \times E^k$, обратное к отношению S, запишем в виде

$$\xi S^{-1}\eta \underset{d}{\equiv} \eta S\xi. \tag{33}$$

Примем введенные формальные определения для решения конкретных примеров логических задач.

Пример 4. Даны множества A, B, C, D, являющиеся подмножествами алфавита E. Доказать, что $(A \times B) \cup (C \times D) \subseteq (A \cup C) \times (B \cup D)$.

Решение. $(A \times B) \cup (C \times D) \subseteq (A \cup C) \times (B \cup D) \equiv \forall x\forall y\,(x \in A \wedge y \in B \vee x \in C \wedge y \in D \supset (x \in A \vee x \in C) \wedge (y \in B \vee y \in D)) \equiv \forall x\forall y\,(\overline{x \in A \wedge y \in B} \wedge \overline{x \in C \wedge y \in D} \vee x \in A \wedge y \in B \vee x \in C \wedge y \in D \vee x \in C \wedge y \in B \vee x \in A \wedge y \in D) \equiv \forall x\forall y\,(x \in A \wedge y \in B \vee \overline{x \in C \wedge y \in D} \vee x \in C \wedge y \in D \vee x \in C \wedge y \in B \vee x \in A \wedge y \in D) \equiv \forall x\forall y\,(x \in A \wedge y \in B \vee 1 \vee x \in C \wedge y \in B \vee x \in A \wedge y \in D) \equiv 1.$

Пример 5. Для каких бинарных отношений S, заданных на E^2, справедливо равенство $S^{-1} = \tilde{S}$?

Решение. $S^{-1} = \tilde{S} \equiv \forall x\forall y\,(ySx \sim xSy) \equiv \bigwedge_{(\sigma_1,\,\sigma_2)} (ySx \sim \overline{xSy}) \equiv (a_1Sa_1 \sim \overline{a_1Sa_1})(a_2Sa_1 \sim \overline{a_1Sa_2}) \ldots (a_pSa_p \sim \overline{a_pSa_p}) \equiv 0 \wedge (a_2Sa_1 \sim \overline{a_1Sa_2}) \wedge \ldots \wedge 0 \equiv 0 = 1.$

Получили противоречие. Поэтому не существует ни одного отношения S, удовлетворяющего заданному равенсгву.

Пример 6. Доказать, что $R \bigcirc (S_1 \cup S_2) = R \bigcirc S_1 \cup R \bigcirc S_2$. Имеется в виду, что R задано на $E^k \times E^l$, а S_1, S_2 — на $E^l \times E^m$.

Решение. $R \bigcirc (S_1 \cup S_2) = R \bigcirc S_1 \cup R \bigcirc S_2 \equiv \forall\xi\forall\zeta\,(\exists\eta\,(\xi R\eta \wedge (\eta S_1\zeta \vee \eta S_2\zeta)) \sim \exists\eta\,(\xi R\eta \wedge \eta S_1\zeta) \vee \exists\eta\,(\xi R\eta \wedge \eta S_2\zeta))) \equiv \forall\xi\forall\zeta\,(\exists\eta\,(\xi R\eta\,(\eta S_1\zeta \vee \eta S_2\zeta)) \sim \exists\eta\,(\xi R\eta \wedge \eta S_1\zeta \vee \xi R\eta \wedge \eta S_2\zeta)) \equiv \forall\xi\forall\zeta\,(\exists\eta\,(\xi R\eta\,(\eta S_1\zeta \vee \eta S_2\zeta)) \sim \exists\eta\,(\xi R\eta\,(\eta S_1\zeta \vee \eta S_2\zeta))) \equiv 1.$

Перейдем к математическому описанию средствами алгебры конечных предикатов функций k-значной логики. Произвольная фиксированная функция k-значной логики $y = f(x_1, x_2, \ldots, x_n)$, где $x_1, x_2, \ldots, x_n, y \in \{0, 1, \ldots, k-1\}$, может быть математически описана на языке алгебры конечных предикатов в виде предиката $F(x_1, x_2, \ldots, x_n, y)$

$$y = f(x_1, x_2, \ldots, x_n) \underset{d}{\equiv} F(x_1, x_2, \ldots, x_n, y), \tag{34}$$

определяемого следующим образом:

$$F(x_1, x_2, \ldots, x_n, y) = \begin{cases} 1, \text{ если } y = f(x_1, x_2, \ldots, x_n); \\ 0, \text{ если } y \neq f(x_1, x_2, \ldots, x_n). \end{cases} \tag{35}$$

Область $\{0, 1, \ldots, k-1\}$ изменения переменных $x_1, x_2, \ldots, x_n, y$ k-значной логики может быть задана системой уравнений

$$x_i^0 \vee x_i^1 \vee \ldots \vee x_i^{k-1} = 1, \ (1 \leqslant i \leqslant n); \tag{36}$$

$$y^0 \vee y^1 \vee \ldots \vee y^{k-1} = 1.$$

В k-значной логике вводят семейство *характеристических функций* $y = x_*^\sigma$ $(\sigma = 0, 1, \ldots, k-1)$, определяемых следующим образом:

$$x_*^\sigma = \begin{cases} k-1, \text{ если } x = \sigma; \\ \quad 0, \text{ если } x \neq \sigma. \end{cases}$$

На языке алгебры конечных предикатов характеристическая функция запишется так:

$$y = x_*^\sigma \underset{d}{\equiv} (x^\sigma y^{k-1} \vee \overline{x^\sigma} y^0)(x^0 \vee x^1 \vee \ldots \vee x^{k-1}). \tag{37}$$

Запишем с помощью зависимости (37) *булеву функцию отрицания* $y = \bar{x}$. В этом частном случае многозначной логики $k = 2$; $\sigma = 0$. Имеем $y = \bar{x} \equiv (x^0 y^1 \vee \overline{x^0} y^0)(x^0 \vee x^1)$. Таким образом,

$$y = \bar{x} \underset{d}{\equiv} x^0 y^1 \vee x^1 y^0. \tag{38}$$

Отношение равенства $x = y$ *в k-значной логике* представим в следующем виде:

$$x = y \underset{d}{\equiv} x^0 y^0 \vee x^1 y^1 \vee \ldots \vee x^{k-1} y^{k-1}. \tag{39}$$

Отношение «больше или равно» в k-значной логике выразим так:

$$x \geqslant y \underset{d}{\equiv} x^0 y^0 \vee x^1 (y^0 \vee y^1) \vee \ldots \vee x^{k-1}(y^0 \vee y^1 \vee \ldots \vee y^{k-1}). \tag{40}$$

Дизъюнкцию k-значной логики $z = x \vee y$ определим следующим образом:

$$z = x \vee y \underset{d}{\equiv} (x \geqslant y \supset z = x) \wedge (y \geqslant x \supset z = y). \tag{41}$$

Аналогично определяем *конъюнкцию k-значной логики*:

$$z = x \wedge y \underset{d}{\equiv} (x \geqslant y \supset z = y) \wedge (y \geqslant x \supset z = x). \tag{42}$$

Для полноты определения конъюнкции и дизъюнкции зависимости (41) и (42) должны быть дополнены уравнениями

$$x^0 \vee x^1 \vee \ldots \vee x^{k-1} = 1; \quad y_з^0 \vee y^1 \vee \ldots \vee x^{k-1} = 1;$$
$$z^0 \vee z^1 \vee \ldots \vee z^{k-1} = 1, \qquad (43)$$

ограничивающими область изменения переменных x, y, z значениями $0, 1, \ldots, k-1$.

Запишем с помощью введенных зависимостей булевы функции *дизъюнкции и конъюнкции*, полагая $k=2$: $z = x \vee y \equiv (x \geqslant y \supset z = x)(y \geqslant x \supset z = y) \equiv ((x^0y^0 \vee x^1(y^0 \vee y^1)) \supset (z^0x^0 \vee z^1x^1 \vee \ldots \vee z^px^p))((y^0x^0 \vee y^1(x^0 \vee x^1)) \supset (z^0y^0 \vee z^1y^1 \vee \ldots \vee z^py^p))(x^0 \vee x^1)(y^0 \vee y^1)(z^0 \vee z^1)$. Аналогично $z = x \wedge y \equiv ((x^0y^0 \vee x^1(y^0 \vee y^1)) \supset (z^0y^0 \vee z^1y^1 \vee \ldots \vee z^py^p)) \wedge ((y^0x^0 \vee y^1(x^0 \vee x^1)) \supset (z^0x^0 \vee z^1x^1 \vee \ldots \vee z^px^p))(x^0 \vee x^1)(y^0 \vee y^1)(z^0 \vee z^1)$. Окончательно

$$z = x \vee y \underset{d}{\equiv} x^0y^0z^0 \vee x^1y^0z^1 \vee x^0y^1z^1 \vee x^1y^1z^1; \qquad (44)$$

$$z = x \wedge y \underset{d}{\equiv} x^0y^0z^0 \vee x^1y^0z^0 \vee x^0y^1z^0 \vee x^1y^1z^1. \qquad (45)$$

Функцию циклического отрицания $y = \bar{x}_*$ в k-значной логике введем тождеством

$$y = \bar{x}_* \underset{d}{\equiv} x^0y^1 \vee x^1y^2 \vee \ldots \vee x^{k-2}y^{k-1} \vee x^{k-1}y^0. \qquad (46)$$

В частном случае при $k=2$ по этому определению получаем булеву функцию отрицания $y = \bar{x} \equiv x^0y^1 \vee x^1y^0$.

Суперпозицию $z = f_1(f_2(x_1, x_2, \ldots, x_m), y_2, \ldots, y_n)$ *функций* f_1 и f_2 k-значной логики определим как произведение двух отношений $z = f_1(t, y_2, \ldots, y_n)$ и $t = f_2(x_1, x_2, \ldots, x_m)$:

$$y = f_1(f_2(x_1, x_2, \ldots, x_m), y_2, \ldots, y_n) \underset{d}{\equiv} \exists t(y = f_1(t, y_2, \ldots, y_n) \wedge$$
$$\wedge\, t = f_2(x_1, x_2, \ldots, x_m)). \qquad (47)$$

Запишем, к примеру, описанным способом булеву функцию $z = x \vee \bar{y}$. Имеем $z = x \vee t$, $t = \bar{y}$. Следовательно, $z = x \vee \bar{y} \equiv \exists t((z = x \vee t) \wedge (t = \bar{y})) \equiv (z = x \vee 0)(0 = \bar{y}) \vee (z = x \vee 1)(1 = \bar{y}) \equiv x^0y^1z^0 \vee (x^1 \vee y^0)z^1$.

Список литературы: 1. *Шабанов-Кушнаренко Ю. П.* О теории интеллекта. — Проблемы бионики. Харьков, 1977, вып. 22, с. 15—22. 2. *Шабанов-Кушнаренко Ю. П.* Об алгебре конечных предикатов. — АСУ и приборы автоматики. Харьков, 1977, вып. 52, с. 21—28. 3. *Шабанов-Кушнаренко Ю. П.* Об алгебре предикатов с отрицанием. — АСУ и приборы автоматики. Харьков, 1977, вып. 52, с. 42—49. 4. *Шабанов-Кушнаренко Ю. П.* Об уравнениях теории интеллекта. — АСУ и приборы автоматики. Харьков, 1977, вып. 53, с. 63—70.

УДК 007:612.843.72

Б. К. ЛОПАТЧЕНКО, канд. техн. наук

АКСИОМАТИЧЕСКОЕ ВВЕДЕНИЕ МЕТРИКИ В БИНОКУЛЯРНОМ ЗРИТЕЛЬНОМ ПРОСТРАНСТВЕ. *СООБЩЕНИЕ 1*

Восприятие человеком трехмерного физического пространства во многом обеспечивается бинокулярным механизмом. Субъективный образ пространства, полученный только за счет бинокулярного восприятия, существенно отличается от пространства стимулов [1].

Для изучения процессов зрительного восприятия используем метод «черного ящика». Условия психофизического эксперимента предполагают выполнение ряда требований. Испытуемый рассматривается как некоторый преобразователь информации — «черный ящик», входными сигналами которого являются зрительные картины. Зрительные образы, возникшие в результате действия механизмов восприятия, недоступны непосредственному измерению. Для получения информации о законах восприятия входные сигналы подбирались таким образом, чтобы испытуемый работал в режиме нуль-органа, сравнивая образы. Выходными сигналами «черного ящика» служат двоичные ответы «да» или «нет», свидетельствующие о равенстве или неравенстве зрительных образов. Таким образом множество входных сигналов (зрительных картин) $x_1, x_2, \ldots, x_n \in X$ отображается на множество выходных $y \in Y$, $Y = \{0, 1\}$.

Испытуемый может работать в следующих режимах:

1. На вход поступают пары сигналов (x_1, x_2) из множества X. Испытуемый реализует функцию

$$y = f(x_1, x_2), \tag{1}$$

которую можно представить в виде

$$f(x_1, x_2) = L(\gamma F(x_1), \gamma F(x_2)), \tag{2}$$

где L — характеристическая функция диагонали квадрата

$$L(u, v) = \begin{cases} 1, \text{ если } u = v; \\ 0, \text{ если } u \neq v, \end{cases}$$

γ — некоторая взаимно однозначная функция. При исследовании алгоритмов зрительного восприятия под входными сигналами x_1, x_2 следует понимать элементы физического пространства зрения, под выходными — ответы испытуемого «да» или «нет», свидетельствующие о равенстве или неравенстве зрительных ощущений $F(x_1)$ и $F(x_2)$. Оператор F и есть искомый оператор преобразования физических стимулов и субъективный зрительный образ. Для представления функции f в виде (2) она должна удовлетворять следующим необходимым и достаточным условиям 1) $f(x_1, x_1) = 1$

(рефлексивность); 2) если $f(x_1, x_2)=1$, то $f(x_2, x_1)=1$ (симметричность); 3) если $f(x_1, x_2)=1$ и $f(x_2, x_3)=1$, то $f(x_1, x_3)=1$ (транзитивность).

2. На вход поступают тройки сигналов (x_1, x_2, x_3). Испытуемый реализует функцию

$$y=f(x_1, x_2, x_3)=L(g(F(x_1), F(x_2)), F(x_3)), \qquad (3)$$

где $g(u, v)=(u+v)/2$.

Эксперимент, описываемый формулой (3), заключается в том, что испытуемому предъявляют три стимула: x_1, x_2, x_3. Если один из стимулов (x_3) воспринимается субъективно средним между остальными двумя, испытуемый должен ответить «да», в противном случае — «нет». Необходимо отметить, что субъективно средний стимул, как правило, не является таковым в физическом пространстве. Нахождение физического стимула x_3, воспринимаемого наблюдателем субъективно средним между стимулами x_1 и x_2, принято называть операцией субъективного равноделения $x_3=x_1 \bigcirc x_2$, где $\bigcirc$ — операция внутреннего равноделения.

Нахождение точки x_4, для которой x_1 видится средней между точками x_2 и x_4 (или точки x_5, для которой x_2 воспринимается средней между x_1 и x_5), называется операцией внешнего равноделения: $x_4=x_1 * x_2$; $x_5=x_2 * x_1$, где $*$ — операция внешнего равноделения.

Применение операции субъективного равноделения основано на выполнении следующих условий [2]: 4) для любых точек x_1, $x_2 \in R^n$ уравнение $x_1 \bigcirc x=x_2$ однозначно разрешимо; 5) для любых точек x_1, $x_2 \in R^n$ справедливо $x_1 \bigcirc x_2=x_2 \bigcirc x_1$; 6) $x_1 \bigcirc x_1 = x_1$; 7) операция равноделения непрерывна на пространстве R^n. Для нахождения алгоритма преобразования физического пространства в воспринимаемое прежде всего необходимо выяснить метрический характер воспринимаемого пространства.

Экспериментальные данные свидетельствуют о том, что для воспринимаемого бинокулярного пространства характерна геометрия Лобачевского [1]. Строгое решение задачи о виде метрики пространства сводится к проверке системы аксиом, из которых вытекает тот или иной вид пространства.

Известна аксиоматика Гильберта, состоящая для трехмерного пространства из 20 аксиом [3]. Но постулирование в ней таких элементов, как прямая, отрезок, плоскость затрудняет ее применение при экспериментальном исследовании. Формирование этих элементов в воспринимаемом пространстве должно быть согласовано с реальной методикой экспериментальной проверки. С этой целью построена аксиоматика, базирующаяся на операциях равноделения и сравнения расстояний.

Рассмотрим конкретное содержание пространства входных сигналов X, отображений f и F. Элементами пространства входных сигналов являются светящиеся точечные источники, предъявляемые испытуемому в затемненном помещении. В эксперименте испытуемый при предъявлении двух пар точек (a, b) и (c, d) должен отвечать «да», если видимые расстояния между точками в парах равны, и «нет» — в противном случае. Входными сигналами x_1 и x_2 «черного ящика» в этом случае будут пары точек (a, b) и (c, d), выходными — ответы испытуемого «да» или «нет». Функция f описывает поведение испытуемого в рассматриваемом эксперименте, отображая множество входных сигналов $X \times X$ на множество выходных $Y = \{0, 1\}$:

$$y = f(a, b, c, d). \tag{4}$$

Психофизические эксперименты свидетельствуют о том, что выполняются следующие аксиомы:

Аксиома 1. Одинаковые пары точек воспринимаются испытуемым равными $f(a, b, a, b) = 1$ (свойство рефлексивности).

Аксиома 2. Порядок предъявления одинаковых пар точек не влияет на результат их восприятия. Если $f(a, b, c, d) = 1$, то $f(c, d, a, b) = 1$ (свойство симметричности).

Аксиома 3. Если пары точек (a, b) и (c, d) воспринимаются равными и субъективно равны пары точек (c, d) и (l, m), то пара (a, b) видится равной паре (l, m), т. е. если $f(a, b, c, d) = 1$ и $f(c, d, l, m) = 1$, то $f(a, b, l, m) = 1$ (свойство транзитивности).

Выполнение этих свойств в эксперименте означает, что функцию $f(a, b, c, d)$ можно представить в виде

$$f(a, b, c, d) = L(\rho(F(a), F(b)), \rho(F(c), F(d))), \tag{5}$$

где ρ — расстояние в метрике воспринимаемого пространства; F — отображение физического пространства R^3 в воспринимаемое V^3.

При предъявлении наблюдателю пары светящихся точек a и b он с уверенностью вызывает точку c, которая кажется ему лежащей на равном и кратчайшем расстоянии между точками a и b. Таким образом в зрительном пространстве определено трехместное отношение $f(a, c, b)$, справедливое в том и только в том случае, если точка c является средней между a и b в указанном выше смысле. Функцию f можно представить в виде

$$f(a, b, c) = L(g(F(a), F(b)), F(c)), \tag{6}$$

где g — операция равноделения в воспринимаемом пространстве V^3.

Аксиомы 1—3, а также аксиомы, сформулированные на основе условий 4—7, вошли составной частью в построенную аксиоматику. В остальных аксиомах в качестве элементов зрительного пространства используются более сложные объекты: «прямая», «угол», «отрезок», «плоскость». Построение этих элементов из светящихся точек производится с помощью операции субъективного равноделения.

Список литературы: 1. *Лопатченко Б. К., Шульгин И. В.* Экспериментальные исследования бинокулярного восприятия пространства. — Проблемы бионики. Харьков, 1979, вып. 22, с. 15—18. 2. *Майстровская Л. М., Шабанов-Кушнаренко Ю. П.* О многомерных шкалах равноделения. Матер. II. Всесоюз. конф. «Биологическая и медицинская кибернетика», ч. 3. М. —Л., 1974, с. 135—136. 3. *Гильберт Д.* Основания геометрии. М.— Л., Гостехиздат, 1948. 280 с.

УДК 007:612.843.72

Б. К. ЛОПАТЧЕНКО, канд. техн. наук

АКСИОМАТИЧЕСКОЕ ВВЕДЕНИЕ МЕТРИКИ В БИНОКУЛЯРНОМ ЗРИТЕЛЬНОМ ПРОСТРАНСТВЕ. *СООБЩЕНИЕ 2*

Для удобства математического описания свойств бинокулярного пространства введем некоторые определения.

Определение 1. Множество $M \subset R^3$ называется завершенным, если 1) M содержит не менее двух точек; 2) $u, v \in M$, то $u \bigcirc v \in M$; 3) $u, v \in M$, то $u * v \in M$; $v * u \in M$; 4) множество M — замкнутое.

Определение 2. Множество M называется «прямой», если оно завершенное и никакая его собственная часть не является завершенным множеством.

Определение 3. Множество M называется «плоскостью», если оно завершенное, не является «прямой» и не совпадает с R^3.

На основании условий 4—7 (см. сообщение 1) сформулируем следующие аксиомы:

Аксиома 1. Для любых точек $a, b \in R^3$ уравнение $a \bigcirc x = b$ однозначно разрешимо.

Аксиома 2. Для любых точек $a, b \in R^3$ справедливо равенство $a \bigcirc b = b \bigcirc a$.

Аксиома 3. Операция равноделения обладает свойством $a \bigcirc a = a$.

Аксиома 4. Операции внутреннего ($\bigcirc$) и внешнего ($*$) равноделения непрерывны на пространстве R^3.

С помощью операций равноделения построим некоторую конструкцию Г, которая может быть использована в качестве «прямой» и «плоскости».

Пусть $\{a, b, \ldots, l\}$ — конечное непустое множество точек в пространстве R^3. Положим

$$S_0(a, b, \ldots, l) = \{a, b, \ldots, l\}. \qquad (1)$$

Построим по индукции последовательность множеств $S_k(a, b, \ldots, l)$.

Пусть множество $S_k(a, b, \ldots, l)$ определено. Положим

$$S_{k+1}(a, b, \ldots, l) = \Big(\bigcup_{u,\, v \in S_k} (u \bigcirc v)\Big) \cup \Big(\bigcup_{u,\, v \in S_k} (u * v)\Big) \cup \Big(\bigcup_{u,\, v \in S_k} (v * u)\Big). \qquad (2)$$

Таким образом, по индукции определена последовательность множеств $S_0(a, b, \ldots, l)$, $S_1(a, b, \ldots, l)$, $S_2(a, b, \ldots, l)$, ..., $S_k(a,$

$b, \ldots, l), \ldots$ Для двух точек эта последовательность будет иметь вид

$$S_0(a, b) = \{a, b\};$$

$$S_1(a, b) = \{a, b, a \bigcirc b, a * b, b * a\};$$

$$S_2(a, b) = \{a, b, a \bigcirc b, a * b, b * a, a \bigcirc (a \bigcirc b), b \bigcirc (a \bigcirc b), a \bigcirc (b * a), b \bigcirc (a * b), b * (a * b), a * (b * a), (a \bigcirc b) * (a * b), (a \bigcirc b) * (b * a)\},$$

. . . .

Из аксиомы 3 следует, что $S_0(a, b, \ldots, l) \subset S_1(a, b, \ldots, l) \subset \ldots \subset S_k(a, b, \ldots, l) \subset \ldots$ Очевидно (из аксиомы 2), что $S_k(a, b, \ldots, l)$ не зависит от порядка точек $a, b, \ldots, l$.

Определение 4. Положим $S(a, b, \ldots, l) = \bigcup\limits_{k=0}^{\infty} S_k(a, b, \ldots, l)$.

Определение 5. $\Gamma(a, b, \ldots, l) = \overline{S}(a, b, \ldots, l)$. Очевидно, $\Gamma(a, b, \ldots, l)$ не зависит от порядка точек $a, b, \ldots, l$. Так как $S_k(a, b, \ldots, l)$ не зависит от порядка точек, следовательно, и S не зависит от порядка и при замыкании это свойство не меняется. Множество $\Gamma(a, b, \ldots, l)$ непусто (оно содержит, например, точки $a, b, \ldots, l$).

Утверждение 1. Если $u, v \in \Gamma(a, b, \ldots, l)$, то $u \bigcirc v \in \Gamma(a, b, \ldots, l)$, $u * v \in \Gamma(a, b, \ldots, l)$.

Доказательство. Из определения 5 следует, что найдутся такие точки $u_k, v_k \in S_k(a, b, \ldots, l)$, что $\lim\limits_{k \to \infty} u_k = u$, $\lim\limits_{k \to \infty} v_k = v$. Тогда $u_k \bigcirc v_k \in S_{k+1}(a, b, \ldots, l)$ и $u_k * v_k \in S_k(a, b, \ldots, l)$, следовательно, $u_k \bigcirc v_k, u_k * v_k \in \Gamma(a, b, \ldots, l)$. Но в силу аксиомы 4 $\lim\limits_{k \to \infty} u_k \bigcirc v_k = u \bigcirc v$, $\lim\limits_{k \to \infty} u_k * v_k = u * v$ и $\Gamma(a, b, \ldots, l)$ — замкнуто. Следовательно, $u \bigcirc v, u * v \in \Gamma(a, b, \ldots, l)$.

Следствие 1. Если набор точек $e_1, e_2, \ldots, e_s \in \Gamma(a, b, \ldots, l)$, то $\Gamma(e_1, e_2, \ldots, e_s) \subset \Gamma(a, b, \ldots, l)$.

Утверждение 2. Если среди точек $a, b, \ldots, l$ есть хотя бы две различных, то множество $\Gamma(a, b, \ldots, l)$ — завершенное.

Доказательство. Свойства 1,4 (см. определение 1) вытекают из определения 5, а свойства 2, 3 — из утверждения 1.

Утверждение 3. Если M — завершенное множество и $a, b, \ldots, l \in M$, то $\Gamma(a, b, \ldots, l) \subset M$.

Доказательство. Точки $a, b, \ldots, l$ образуют некоторое множество $S_0(a, b, \ldots, l) \subset M$. Так как по условию множество M — завершенное, то в силу свойств 2 и 3 определения 1 можно построить последовательность вложенных множеств $S_0(a, b, \ldots, l) \subset S_1(a, b, \ldots, l) \subset \ldots \subset S_k(a, b, \ldots, l) \subset S_{k+1}(a, b, \ldots, l) \subset \ldots \subset M$, где элемент $S_{k+1}(a, b, \ldots, l)$ определяется формулой (2).

Очевидно, для объединения множеств $S_k(a, b, \ldots, l)$ выполняется соотношение $S(a, b, \ldots, l) \subset M$. При замыкании это соотношение не изменится, т. е. $\overline{S}(a, b, \ldots, l) \subset \overline{M}$, но $\overline{M} = M$

(в силу свойств замыкания), а $\bar{S}(a, b, \ldots, l) = \Gamma(a, b, \ldots, l)$ по определению 5, следовательно, $\Gamma(a, b, \ldots, l) \subset M$.

Аксиома 5. Если $c, d \in \Gamma(a, b)$ и $c = d$, то $a, b \in \Gamma(c, d)$.

Утверждение 4. Если $a \neq b$, то множество $\Gamma(a, b)$ есть «прямая».

Доказательство. Множество $\Gamma(a, b)$ — завершенное (согласно утверждению 2). Пусть множество $\Gamma(a, b)$ имеет собственную часть

$$M \subset \Gamma(a, b), \tag{3}$$

где M — завершенное. Так как M — завершенное множество, то из определения 1 (свойство 1) следует, что существует $c, d \in M$ и $c \neq d$. Точки $c, d \in \Gamma(a, b)$, так как M — его часть, следовательно, $a, b \in \Gamma(c, d)$ (из аксиомы 5). Множество $\Gamma(c, d)$ — завершенное (утверждение 2). Теперь на основании утверждения 3 могут быть записаны следующие соотношения:

$$\Gamma(a, b) \subset \Gamma(c, d); \tag{4}$$

$$\Gamma(c, d) \subset M. \tag{5}$$

Соотношения (3)—(5) могут иметь место только если $\Gamma(a, b) = M$, т. е. множество $\Gamma(a, b)$ — завершенное, и его завершенная собственная часть совпадает со всем множеством, следовательно, по определению 2 $\Gamma(a, b)$ — «прямая», что и требовалось доказать.

Следствие 2. Через любые две точки проходит «прямая».

Утверждение 5. Если M — «прямая», $a, b \in M$ и $a \neq b$, то $M = \Gamma(a, b)$.

Доказательство. Так как M — завершенное множество (по определению 2), то по утверждению 3 имеем $\Gamma(a, b) \subset M$. Если $\Gamma(a, b) \neq M$, то собственная часть множества M является завершенным множеством, что противоречит определению 2. Следовательно, $M = \Gamma(a, b)$.

Следствие 3. Через две различные точки проходит не более одной «прямой».

Следствие 4. На каждой «прямой» лежит по крайней мере две точки.

Утверждение 6. Если $a \neq b$ и $c \in \Gamma(a, b)$, то $a \overline{\in} \Gamma(a, b)$.

Доказательство. Пусть $a \in \Gamma(c, b)$. Тогда в силу аксиомы выполняется условие: если $a, d \in \Gamma(a, b)$ и $a \neq d$, то $b, c \in \Gamma(a, b)$. В нашем случае $a, b \in \Gamma(c, b)$ и $a = b$, следовательно, $c \in \Gamma(a, b)$, что противоречит нашему предположению.

Следствие 5. Пусть a, b, c — различные точки. Тогда эквивалентны следующие три условия:

$$c \overline{\in} \Gamma(a, b); \quad a \overline{\in} \Gamma(b, c); \quad b \overline{\in} \Gamma(a, c). \tag{6}$$

Определение 6. Пусть a, b, c — различные точки. Если справедливо хотя бы одно из условий (6), будем говорить, что точки не лежат на одной «прямой».

Следствие 6. Пусть a, b, c — различные точки. Тогда эквивалентны следующие три условия:

$$c \in \Gamma(a, b); \quad a \in \Gamma(b, c); \quad b \in \Gamma(a, c). \tag{7}$$

Определение 7. Пусть a, b, c — различные точки. Если справедливо одно из условий (7), будем говорить, что точки a, b, c лежат на одной прямой.

Определения 6 и 7 корректны в силу того, что в нашем случае «прямые» не зависят от порядка точек. Это следует из способа построения конструкции Γ.

Аксиома 6. Конструктивное множество Γ, построенное по трем точкам, не заполняет все пространство R^3, т. е. $\Gamma(a, b, c) \neq R^3$.

Утверждение 7. Если a, b, c — различные точки, не лежащие на одной «прямой», то $\Gamma(a, b, c)$ — «плоскость».

Доказательство. Из утверждения 2 следует, что $\Gamma(a, b, c)$ — завершенное множество. $\Gamma(a, b, c)$ — не «прямая», так как по условию теоремы a, b, c не лежат на одной «прямой».

Из аксиомы 3 следует, что $\Gamma(a, b, c) \neq R^3$, следовательно, по определению 3 множество $\Gamma(a, b, c)$ — «плоскость».

Следствие 7. Через любые три точки проходит «плоскость».

Аксиома 7. Если a, b, c — различные точки и $d \notin \Gamma(a, b, c)$, то $\Gamma(a, b, c, d) = R^3$.

Утверждение 8. Если M — «плоскость», a, b, $c \in M$ и различны, то $M = \Gamma(a, b, c)$.

Доказательство. M — завершенное множество (по определению 3). Тогда из утверждения 3 следует

$$\Gamma(a, b, c) \subset M. \tag{8}$$

Пусть $M \neq \Gamma(a, b, c)$. Тогда из соотношения (8) заключаем, что существует точка $d \in M$, $d \notin \Gamma(a, b, c)$, и из аксиомы 7 имеем

$$\Gamma(a, b, c, d) = R^3. \tag{9}$$

Но a, b, c, $d \in M$, где M — завершенное, следовательно, из утверждения 3

$$\Gamma(a, b, c, d) \subset M. \tag{10}$$

Из выражений (9) и (10) имеем $M = R^3$, что противоречит определению 3.

Следствие 8. Через три различные точки проходит ровно одна «плоскость».

Следствие 9. Если a, $b \in M$, где M — «плоскость», то $\Gamma(a, b) \subset M$, т. е. вся прямая принадлежит «плоскости».

Следствие 10. Существует по крайней мере четыре точки, не лежащие в одной «плоскости».

Аксиома 8. Если две «плоскости» имеют общую точку, то они имеют, по крайней мере, еще одну общую точку.

Построим некоторое конструктивное множество Δ, которое будет использовано нами в качестве такого элемента зрительного пространства, как «отрезок». Пусть $a \neq b$ — некоторые фиксированные точки. Положим

$$T_0(a, b) = \{a, b\}. \tag{11}$$

Построим по индукции последовательность множеств T_k. Пусть множество $T_k(a, b)$ определено. Положим

$$T_{k+1}(a, b) = \bigcup_{u, v \in T_k} (u \bigcirc v). \tag{12}$$

Тогда образуется следующая последовательность множеств: $T_0(a, b) = \{a, b\}$; $T_1(a, b) = \{a, b, a \bigcirc b\}$; $T_2(a, b) = \{a, b, a \bigcirc b, a \bigcirc \bigcirc (a \bigcirc b), b \bigcirc (a \bigcirc b)\}$. . . . .
В силу аксиомы 3 эти множества вложены друг в друга: $T_0(a, b) \subset T_1(a, b) \subset T_2(a, b) \subset \ldots \subset T_k(a, b) \subset \ldots$

Определение 8. Положим $T(a, b) = \left(\bigcup\limits_{k=0}^{\infty} T_k(a, b)\right) \diagdown T_0(a, b)$.

Определение 9. Положим $\Delta(a, b) = \overline{T}(a, b) \diagdown T_0(a, b)$.

Таким образом, мы ввели некоторый открытый отрезок. По аксиоме 3 $T_1(a, b) = \{a\} = \{b\}$. Учитывая выражение (12), имеем $T_k = \{a\} = \{b\}$. По определению 8 $T(a, b) = T_0(a, b)$ и замыкание множества $T(a, b)$ также равно $T_0(a, b)$. По определению 9 $\Delta(a, b) = \varnothing$.

Справедливо и обратное утверждение: если $\Delta(a, b) = \varnothing$, то $a = b$. Действительно, из определения 9 вытекает, что $\overline{T}(a, в) \diagdown \diagdown T_0(a, b) = \varnothing$. Такое соотношение имеет место либо если $\overline{T}(a, b) = \varnothing$ и $T_0(a, b) = \varnothing$, либо если $\overline{T}(a, b) = T_0(a, b)$, но $T_0(a, b) \neq \varnothing$, так как в соответствии с выражением (11) содержит хотя бы одну точку, следовательно, справедливо $\overline{T}(a, b) = T_0(a, b)$. То есть разность $\bigcup\limits_{k=0}^{\infty} T_k(a, b) \diagdown T_0(a, b)$ равна $T_0(a, b)$. Это справедливо только тогда, когда $a = b$. Очевидно,

$$\Delta(a, b) \subset \Gamma(a, b), \tag{13}$$

так как из равенств (2) и (12) $T_k(a, b) \subset S_k(a, b)$. Учитывая определения 4 и 8, $T(a, b) \subset S(a, b)$, а в соответствии с определениями 5 и 9 $\Delta(a, b) \subset \Gamma(a, b)$. Очевидно,

$$\Delta(a, b) = \Delta(b, a) \tag{14}$$

в силу аксиомы 2.

Определение 10. Если $c \in \Delta(a, b)$, будем говорить, что точка c лежит между точками a и b или (что одно и то же) точка c лежит между точками b и a.

Очевидно, если $c \in \Delta(a, b)$, то a, b, c — различные точки (так как отрезок открытый) одной «прямой» [из свойства (13)].

Утверждение 9. Для любых точек a и c существует точка $b \in \Gamma(a, c)$ такая, что $c \in \Delta(a, b)$.

Доказательство. Положим $b = a * c$. Согласно утверждению 1 $b \in \Gamma(a, c)$ и из способа построения $\Delta(a, b)$ (формула (12) определения 8, 9) следует, что $a \bigcirc b \in \Delta(a, b)$.

Утверждение 10. Если $c \in \Delta(a, b)$, то $T(a, c) \subset \Delta(a, b)$.

Доказательство проведем индукцией по k. Пусть $c \subset \Delta(a, b)$, следовательно, по определению 9 существует последовательность точек $c_n \in T_{k(n)}(a, b)$ такая, что $\lim\limits_{n \to \infty} c_n = c$. Тогда $a \bigcirc c_n \in T_{k(n)+1}(a, b)$ и поскольку $a \bigcirc c_n \neq a$ и $a \bigcirc c_n \neq b$, $a \bigcirc c_n \in T_{k(n)+1}(a, b) \setminus T_0(a, b)$. Далее из способа построения $\Delta(a, b)$ вытекает соотношение $T_{k(n)+1}(a, b) \setminus T_0(a, b) \subset T(a, b) \subset \overline{T}(a, b) = \Delta(a, b) \cup T_0(a, b)$. Из аксиомы 4 следует $a \bigcirc c \in \overline{T}(a, b) = \Delta(a, b) \cup T_0(a, b)$. Так как $a \bigcirc c \neq a$ и $a \bigcirc c \neq b$ (поскольку иначе было бы $c = a * b$, что противоречит условию $c \in \Delta(a, b)$), то $a \bigcirc c \in \Delta(a, b)$. $T_0(a, c) \subset \subset \Delta(a, b) \cup \{a\}$, пусть и $T_k(a, c) \subset \Delta(a, b) \cup \{a\}$. Положим $z = T_{k+1}(a, c)$, тогда $z = u \bigcirc v$, где $u, v \in T_k(a, c)$, но $T_k(a, c) \subset \subset \Delta(a, b) \cup \{a\}$, следовательно, существуют точки u_n, v_n такие, что $\lim\limits_{n \to \infty} u_n = u$ и $\lim\limits_{n \to \infty} v_n = v$; $u_n, v_n \in T_{k(n)}(a, b)$, тогда $u_n \bigcirc v_n \in \in T_{k(n)+1}(a, b) \subset \Delta(a, b)$. Из аксиомы 4 следует, что $u \bigcirc v \in \Delta(a, b) \cup T_0(a, b)$ и $z \in \Delta(a, b)$;

$$\bigcup_{k=0}^{\infty} T_k(a, c) \subset \Delta(a, b) \cup \{a\}.$$

Из определения 8 вытекает $T(a, c) \subset \Delta(a, b) \setminus \{c\} \subset \Delta(a, b)$. Утверждение доказано.

Следствие 11. Если $c \in \Delta(a, b)$, то $\Delta(a, c) \subset \Delta(a, b)$.

Утверждение 11. Если $c \in \Delta(a, b)$ и $a \in \Delta(c, b)$, то $\Delta(a, b) = \Delta(c, b)$.

Доказательство. Если $c \in \Delta(a, b)$, из следствия 11 имеем $\Delta(c, b) \subset \Delta(a, b)$. Если $a \in \Delta(c, b)$, из следствия 11 имеем $\Delta(a, b) \subset \subset \Delta(c, b)$, откуда $\Delta(a, b) = \Delta(c, b)$.

Аксиома 9. Если $\Delta(a, b) = \Delta(c, b)$, то $a = c$.

Следствие 12. Среди любых трех точек существует не более одной, лежащей между двумя другими.

Действительно, если $c \in \Delta(a, b)$ и $a \in \Delta(c, b)$, то в силу утверждения 11 и аксиомы 9 $a = c$, но это противоречит тому, что $c \in \Delta(a, b)$.

Аксиома 10. Пусть точки a, b, c не лежат на одной «прямой» и Γ — некоторая «прямая» $\Gamma \subset \Gamma(a, b, c)$, причем $a, b, c \in \Gamma$. Тогда, если $d \in \Delta(a, b)$, то либо $e \in \Delta(a, c)$, либо $e \in \Delta(b, c)$, где $d, e \in \Gamma$.

Определение 11. Будем говорить, что если $\Delta(a, b) \sim \Delta(c, d)$, то существует преобразование, при котором $\Delta(a, b) = \Delta(c, d)$.

Поскольку точки a, b однозначно определяют «отрезок» $\Delta(a, b)$, то, используя отношения эквивалентности (аксиомы 1—3) 1 для «отрезков», можно вывести следующие следствия из свойств «отрезков»:

Следствие 13. Если $a, b \in \Gamma(a, b)$, $a' \in \Gamma(a', b')$, то существует точка b' (или $b'' = a * b'$) такая, что $\Delta(a, b) \sim \Delta(a', b')$.

Следствие 14. Если $\Delta(a, b) \sim \Delta(e, d)$ и $\Delta(e, m) \sim \Delta(c, d)$, то $\Delta(a, b) \sim \Delta(e, m)$.

Следствие 15. Пусть $\Delta(a, b) \subset \Gamma(a, b)$, $\Delta(b, c) \subset \Gamma(a, b)$ и $\Delta(a, b) \cap \Delta(b, c) = \varnothing$. Пусть также $\Delta(a', b') \subset \Gamma(a', b')$, $\Delta(b', c') \subset \subset \Gamma(a', b')$ и $\Delta(a', b') \cap \Delta(b', c') = \varnothing$, тогда если $\Delta(a, b) \sim \Delta(a', b')$ и $\Delta(b, c) \sim \Delta(b', c')$, то $\Delta(a, c) \sim \Delta(a', c')$.

Для введения в пространство «углов» необходимо определить понятие «полупрямой». Пусть $a \neq b$ — некоторые фиксированные точки. Примем

$$P_0(a, b) = \{a, b\}. \tag{15}$$

Построим по индукции последовательность множеств $P_k(a, b)$. Пусть определено $P_k(a, b)$. Положим

$$P_{k+1}(a, b) = \Big(\bigcup_{u, v \in P_k} (u \bigcirc v)\Big) \cup \Big(\bigcup_{u, v \in P_k} (u * v)\Big). \tag{16}$$

Образуется следующая последовательность множеств:

$$P_0(a, b) = \{a, b\};$$

$$P_1(a, b) = \{a, b, a \bigcirc b, a * b\};$$

$$P_2(a, b) = \{a, b, a \bigcirc b, a * b, a \bigcirc (a \bigcirc b), b \bigcirc (a \bigcirc b);$$
$$a \bigcirc (a * b), b \bigcirc (a * b), b * (a * b)\};$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$

Из аксиомы 3 следует $P_0(a, b) \subset P_1(a, b) \subset \ldots \subset P_k(a, b) \subset \ldots$ Положим

$$P(a, b) = \bigcup_{k=0}^{\infty} P_k(a, b). \tag{17}$$

Определим «полупрямую» следующим образом:

$$\gamma(a, b) = \overline{P}(a, b). \tag{18}$$

Определение 12. «Углом» назовем множество $\psi(b, a, c) = \gamma(a, b) \cup \gamma(a, c)$.

Аксиома 11. Пусть $\psi(b, a, c) \subset \Gamma(b, a, c)$ и $\Gamma(a, b) \subset \Gamma(a', b')$. Тогда существует «полупрямая» $\gamma(a', c')$ такая, что $\psi(b, a, c) \sim \sim \psi(b', a', c')$, а все внутренние точки плоскости, ограниченные «полупрямыми» «угла» $\psi(b', a', c')$ лежат по одну сторону от «прямой» $\Gamma(a', b')$.

Аксиома 12. Пусть a, b, c не лежат на одной «прямой» и a', b', c' не лежат на одной «прямой». Если при этом $\Delta(a, b) \sim \sim \Delta(a', b')$, $\Delta(a, c) \sim \Delta(a', c')$ и $\psi(b, a, c) \sim \psi(b', a', c')$, то $\psi(a, b, c) \sim \psi(a', b', c')$ и $\psi(a, c, b) \sim \psi(a', b', c')$.

Аксиома 13 (Архимеда). Пусть $\Delta(a, b)$ и $\Delta(c, d)$ — произвольные «отрезки». Тогда на «прямой» $\Gamma(a, b)$ существует ряд точек $a_1, a_2, \ldots, a_n$, расположенных так, что $a_1 \in \Delta(a, a_2)$, $a_2 \in \Delta(a_1, a_3)$ и т. д., причем $\Delta(a, a_1) \sim \Delta(c, d)$, $\Delta(a_1, a_2) \sim \Delta(c, d) \ldots \Delta(a_{n-1}a_n) \sim \sim \Delta(c, d)$ и $b \in \Delta(a, a_n)$.

Аксиома 14 (Кантора). Пусть на «прямой» $\Gamma(a, b)$ дана бесконечная последовательность отрезков $\Delta(a_1, b_1) \supset \Delta(a_2, b_2) \supset \ldots \supset \Delta(a_n, b_n) \supset \ldots$ Пусть для любого отрезка $\Delta(c, d)$ найдется номер n, для которого $\Delta(a_n, b_n) \subset \Delta(a_k, b_k) \sim \Delta(c, d)$. Тогда существует точка $x \in \Gamma(a, b)$, лежащая внутри всех «отрезков», то есть $x \in \Delta(a_n, b_n)$ при $n \to \infty$.

Введенная система аксиом порождает абсолютную геометрию. Добавление к ней некоторой аксиомы параллельности приводит либо к евклидову пространству, либо к пространству Лобачевского. В качестве указанной аксиомы используем свойство средней линии треугольника. В используемых нами операциях она будет иметь следующий вид:

Аксиома 15. Пусть a, b, c — различные точки, не лежащие на одной «прямой». Тогда найдется точка $l \in \Delta(a, a \bigcirc c)$ (либо $l' \in \Delta(c, a \bigcirc c)$) такая, что $\Delta(a \bigcirc b, b \bigcirc c) \sim \Delta(a, l)$ (либо $\Delta(a \bigcirc b, b \bigcirc c) \sim \Delta(a, l')$).

Изложенная аксиоматика удовлетворяет требованиям непротиворечивости и полноты, поскольку по ходу ее введения в качестве следствий получены все аксиомы Гильберта. Экспериментальная проверка аксиом сводится к построению в физическом пространстве «прямых», «плоскостей», «углов» и «отрезков» и оценке точности выполнения над ними отношений, предусмотренных аксиомами. Анализ психофизических экспериментов по проверке аксиом позволяет сделать вывод о том, что метрика воспринимаемого бинокулярного зрительного пространства есть метрика Лобачевского.

УДК 681.33:612.822:62—506

С. Н. ГРИНЧЕНКО, канд. техн. наук,
С. Л. ЗАГУСКИН, канд. биол. наук

МОДЕЛЬ ОПТИМИЗИРУЮЩЕГОСЯ НЕЙРОНА

Биологические системы управления обладают рядом достоинств, которые не реализованы пока в технике: поддержание с высокой точностью большого количества параметров при большой помехоустойчивости, высокая надежность и полифункциональность при небольшой избыточности и т. д. Причины этого, очевидно, следует искать в особенностях способов и механизмов управления в биологических системах.

В ряде работ [см., напр., 1; 2] показано, что наибольшей бионичностью в этом смысле обладают поисковые системы, которые, как и в случае управления биологическими объектами, требуют априорной информации о величине и знаке отклонения управляемых переменных объекта. Становится очевидной аналогия между пробными движениями в поисковых системах и эндогенными колебаниями метаболизма и микроструктуры клеток полезным использованием в них броуновского движения, а на более высоких уровнях — фоновой «спонтанной» активностью, физиологическим тремором и т. д.

Гомеостатическая функция биосистем от организма до клетки направлена на минимизацию отношений со средой, что определяет специфический вид целевой функции соответствующих биосистем. Иными словами, основная задача биосистем с их сложным и, в основном, случайным характером внешних воздействий является поисковой уже по самой своей постановке [2]. Наконец, громадное количество слабых связей между элементами биосистем, определяемое кооперативным взаимодействием макромолекул, взаимосвязями митохондрий через уровень АТФ, АДФ, субстратов окисления и т. д., взаимодействием рибосом через изменение концентрации аминокислот, клеток — посредством общих источников метаболизма, креаторных и гуморальных связей и т. п., является еще одним косвенным подтверждением вывода о поисковом характере систем управления биологических объектов.

В работах [3; 4] нами обосновано применение алгоритмов поисковой оптимизации для управления энергетикой нейрона и высказана гипотеза о роли оптимизации энергетики нейрона в выработке значения его функционального выхода. В соответствии с этой гипотезой согласование и направленность всех рабочих процессов в клетке, вызванных изменением режима функциональной активности нейрона, обеспечивается минимизацией энергетических затрат на регуляцию. Цель нашего исследования — разработка модели «оптимизирующегося» нейрона, учитывающей как энергетические, так и функциональные характеристики поведения нейрона на основе конкретных экспериментальных результатов функционально-топохимических исследований одиночной нервной клетки и выяснения структурно-метаболических механизмов управления его функциональной активностью.

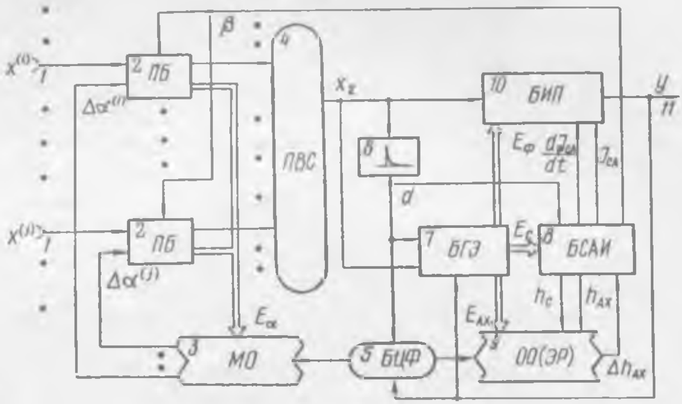
Особенности работы основных подсистем такой модели нейрона описаны [см. 5—8]; принципиальные отличия ее от существующих модельных представлений нейрона следующие:

а) нейрон рассматривается как *активный элемент*, содержащий поисковые оптимизационные механизмы (работающие по алгоритмам *случайного* поиска), с помощью которых он изменяет свою внутреннюю структуру и параметры, достигая экстремума некоторой целевой функции;

б) целевая функция структурной и параметрической адаптации нейрона имеет *энергетический характер*;

в) энергия, вырабатываемая в определенных блоках модели, в соответствии с некоторыми переменными приоритетами распределяется на *потоки*, которые и обеспечивают функционирование соответствующих блоков модели; при этом величины таких потоков энергии являются важнейшими характеристиками, определяющими параметры, а в некоторых случаях — и внутреннюю структуру таких блоков;

г) память в указанной модели рассматривается как изменени в пространственно-временной последовательности факторов: «вход ное воздействие» + «структурно-метаболические процессы в тигро иде» → «чувствительность локальных участков синаптической мем браны» = «веса межнейронных связей», причем в понятие изменени чувствительности синаптических мембран включается как сравни тельно медленная модификация их проводимости под действие мембранотропных белков, так и относительно быстрая модифика ция — при изменении внутриклеточных градиентов концентраци кальция; важной характеристикой следовых процессов в синапса являются также пластические и энергетические процессы, свя занные с функциональной активностью нейрона.



Агрегированная блок-схема модели оптимизирующегося нейрона.

На рисунке показана функциональная схема модели в укру ненной, агрегированной форме. Модель содержит n входов (*1*) n постсинаптических блоков ПБ (*2*), многоканальный оптимизато МО (*3*), блок пространственно-временного суммирования ПВС (*4* блок вычисления целевой функции оптимизации БЦФ (*5*), бло формирования уровня дисбаланса энергии БДЭ (*7*), блок вычисле ния сомааксонного индекса САИ (*8*), одноканальный оптимизато (экстремальный регулятор) ЭР (*9*), блок информационного преобра зования БИП (*10*) и выход (*11*).

Постсинаптический блок *2* представляет собой функционал ный преобразователь, реализующий зависимости: $x^{(i)}_{эфф} = \omega^{(i)}\Gamma_1 \times [\Phi(\omega^{(i)})]$; $E^{(i)}_\alpha = k_1\omega^{(i)} + \Gamma_1 [-\Phi(\omega^{(i)})]$, где $x^{(i)}_{эфф}$ — эффектив ное значение i-го входного воздействия; $\omega^{(i)} = x^{(i)}\Gamma_1 [k_2\beta + \theta(\Delta\alpha^{(i)})]$; $x^{(i)}$ — i-е входное воздействие; Γ_1 — оператор выпря ления (первый интеграл функции Хэвисайда); k_j — коэффициент β — параметр коэффициентов чувствительности синапсов; θ — опе ратор интегрирования по времени; $\Delta\alpha^{(i)}$ — приращение коэфф циента чувствительности i-го синапсов; Φ — оператор интегральн дифференцирующего преобразования; $E_{\alpha}^{(i)}$ — i-я составляющая потока энергии, затрачиваемого на функционирование многоканального оптимизатора.

Блок *4* пространственно-временного суммирования реализует функцию $x_{\Sigma} = \sqrt{\sum_{i=1}^{N} \nu^{(i)} x_{\text{эфф}}^{(i)}}$, где x_{Σ} — суммарное входное воздействие; N — число синапсов (постсинаптических блоков); $\nu^{(i)}$ — весовые коэффициенты.

Блок *5* вычисления целевой функции оптимизации осуществляет преобразование $Q = F(d) + F(y)$, где Q — целевая функция оптимизации энергетики нейрона; F — оператор инерционности; $d = \frac{dx_{\Sigma}}{dt}$ — уровень дисбаланса энергии; y — функциональный выход нейрона (мгновенная частота импульсации).

Блок *6* формирования уровня дисбаланса энергии представляет собой дифференцирующее звено.

Блок *7* генерации энергии осуществляет функциональные преобразования: $E_{\Phi} = \Gamma_1(-\varphi)$; $E_{C} = \Gamma_1(-k_3 d) + \Gamma_1(\varphi)$; $E_{AX} = \Gamma_1(\varphi)$, где E_{Φ} — энергия обеспечения функционального выхода нейрона; $\varphi_{(s+1)} = F^2[\pi(-\varphi_{(s)}) + |x_{\Sigma} - F(y)|]$; s — дискретное время; π — оператор пропорционально-интегрального преобразования; E_{C} — энергия изменения размеров сомы нейрона; E_{AX} — энергия изменения размеров аксонного холмика.

Блок *8* вычисления сомааксонного индекса представляет собой функциональный преобразователь, реализующий зависимости $I_{CA} = \psi - k_4 \Delta h_{AX} - k_5$; $\frac{dI_{CA}}{dt}$ — производная по времени; $h_{AX} = \Delta h_{AX} + k_6$; $h_{C} = \psi + k_7$; $\beta = k_8{}^{\varepsilon}S(d) + k_9$, где I_{CA} — сомааксонный индекс; $\psi = k_{10} E_{C} \operatorname{sign}(d)$; sign — оператор релейного преобразования; Δh_{AX} — приращение размера аксонного холмика; h_{AX} — размер аксонного холмика; h_{C} — размер сомы нейрона; ${}^{\varepsilon}S$ — оператор релейного преобразования с зоной нечувствительности шириной ε.

Блок *10* информационного преобразования реализует зависимость $y = k_{11} E_{\Phi} + (x_{\Sigma} - k_{12})\left[1 + k_{13} I_{CA} \operatorname{sign}\left(\frac{dI_{CA}}{dt}\right)\right]$.

Принцип работы модели оптимизирующего нейрона состоит в следующем. Через i-й вход *1* на i-й постсинаптический блок *2* поступает входное воздействие $x^{(i)}$; i-й ПБ, имея параметрами соответствующее приращение коэффициента чувствительности синапса $\Delta\alpha^{(i)}$ и общий для всех синапсов параметр коэффициента чувствительности β, вырабатывает эффективное значение информационного выхода $x_{\text{эфф}}^{(i)}$, а также энергетический выход $E_{\alpha}^{(i)}$. Величины $\Delta\alpha^{(i)}$ и β отражают внутриклеточный градиент кальция, определяющий натриевую проводимость; оба этих параметра зависят от агрегации тигроида и соответственно от аккумуляции в нем кальция, но β отражает общую направленность процесса в связи с изменением размеров сомы, а $\Delta\alpha^{(i)}$ — локальные процессы, связанные с нарушением энергетического равновесия в постсинаптических участках.

Информационные выходы $x_{эфф}^{(i)}$ постсинаптических блоков 2 суммируются в блоке 4 пространственно-временного суммирования, давая значение суммарного генераторного потенциала сомы нейрона x_{Σ}. Вид функции суммирования позволяет учесть геометрические константы и удаленность каждого конкретного синапса от триггерной зоны.

Энергетические выходы $E_{\alpha}^{(i)}$ постсинаптических блоков 2 определяют интенсивность работы многоканального оптимизатора 3 (т. е. величину его рабочего шага), изменяющего поисковым образом свой выход $\overrightarrow{\Delta\alpha}$ так, чтобы минимизировать целевую функцию Q:

$$\overrightarrow{\Delta\alpha}_{(s+1)} = \overrightarrow{\Delta\alpha}_{(s)} - D_{1(s)}\left(E_{\alpha(s)}\right)\hat{\nabla} Q\left(\overrightarrow{\Delta\alpha}_{(s)}\right), \qquad (1)$$

где D — коэффициент итерации: $\hat{\nabla}$ — оценка градиента. Итеративный процесс (1) реализуется с помощью использования алгоритмов случайного поиска.

Разные значения вектора $\overrightarrow{\Delta\alpha}$ соответствуют различным топохимическим мозаикам агрегации тигроида, концентрационного градиента кальция и гетерогенности энергетических процессов постсинаптических зон (активность цитохромоксидазы синаптосомальных митохондрий, активность ПТФ-азы постсинаптической мембраны и т. д.), что было зарегистрировано в эксперименте.

Введение блока вычисления целевой функции основано также на экспериментальных фактах о сопряжении входных информационных потоков (с помощью регуляции чувствительности входа при изменении агрегации тигроида) и выходных информационных потоков (с помощью агрегации митохондрий при изменении сома-аксонного индекса) через общий механизм перераспределения кальция между внутриклеточными компартментами. Высвобождение кальция из тигроида и аккумуляция его в митохондриях, а также обратный процесс, влияя на степень агрегации тигроида и митохондрий, обеспечивают сопряжение уровня энергетики и трофики нейрона в соответствии с уровнем его функциональной активности на основе общего критерия минимизации регуляторной составляющей энергетических затрат Q.

Одним из аргументов функции Q является степень нарушения равновесия энергетических процессов в соме нейрона d, которая оценивается в блоке 6 формирования уровня дисбаланса энергии, другим аргументом является величина функционального выхода нейрона y. Обе эти величины характеризуют степень нарушения равновесия энергетических процессов в клетке.

Биологическим основанием введения блока БДЭ является свойство АТФ-азы и форма кривых синтеза и расхода АТФ клетки в зависимости от размера энергетического заряда (соотношения АТФ/АДФ/АМФ).

В блоке *7* генерации энергии вырабатываются три потока энергии: энергия изменения размеров сомы $E_{\text{С}}$, энергия изменения размеров аксонного холмика $E_{\text{АХ}}$ и энергия обеспечения функционального выхода нейрона $E_{\text{Ф}}$. Эти потоки энергии используются в качестве дополнительных настроечных параметров блоков, в которых потребляется соответствующая энергия. Так, в блоке *8* вычисления сомааксонного индекса энергия $E_{\text{С}}$ определяет интенсивность изменений размеров сомы $h_{\text{С}}$ и, как следствие,— сомааксонного индекса $I_{\text{СА}}$. В одноканальном оптимизаторе *9* энергия $E_{\text{АХ}}$ задает размер его рабочего шага. В блоке *10* информационного преобразования энергия $E_{\text{Ф}}$ является параметром функции вычисления мгновенной частоты импульсной активности нейрона (его информационной функции).

Блок *8* вычисления сомааксонного индекса вырабатывает значения величин, связанных с геометрическими параметрами клетки: $h_{\text{С}}$ и $h_{\text{АХ}}$ — соответственно диаметры сомы и аксонного холмика, сомааксонный индекс $I_{\text{СА}}$, определяющий декремент распространения генераторного потенциала к триггерной зоне нейрона, и его производная $\frac{dI_{\text{СА}}}{dt}$, параметр коэффициентов чувствительности постсинаптических блоков β. Декремент генераторного потенциала, как известно, определяет изменение частоты импульсной активности нейрона y.

Одноканальный оптимизатор *9* вырабатывает значение приращения диаметра аксонного холмика, также имея своей целью минимум функции Q:

$$\Delta h_{\text{АХ}(s+1)} = \Delta h_{\text{АХ}(s)} - D_{2(s)}\left(E_{\text{АХ}(s)},\ h_{\text{С}(s)},\ h_{\text{АХ}(s)}\right) \frac{\partial Q\left(\Delta h_{\text{АХ}(s)}\right)}{\partial t}.$$

Переменные $h_{\text{С}}$ и $h_{\text{АХ}}$ используются в алгоритме экспериментального регулирования одноканальной оптимизации для ограничения размера его рабочего шага. Морфологическим выражением механизма экстремального регулирования являются колебательные изменения агрегации митохондрий, сопровождающиеся активацией либо контактным угнетением их окислительного метаболизма. Таким образом, оптимизация энергетики нейрона связывается с отысканием оптимальных геометрических параметров сомы и аксонного холмика, наиболее соответствующих текущему уровню функциональной активности нейрона. В эксперименте эту связь демонстрируют увеличение активности цитохромоксидазы и образование барьера из митохондрий на границе с аксонным холмиком при увеличении диаметра последнего при затормаживании нейрона [9].

Блок *10* информационного преобразования на основе учета важнейших внутренних переменных модели x_{Σ}, $E_{\text{Ф}}$, $I_{\text{СА}}$ и $\frac{dI_{\text{СА}}}{dt}$

вырабатывает значение ее функционального выхода. Эта величина, таким образом, очень существенно зависит от переменных, характеризующих энергетику и внутреннюю структуру нейрона: от размера генераторного потенциала x_Σ, в котором отражается работа многоканального оптимизатора; от E_Φ, характеризующей уровень функциональной составляющей энергетических затрат; от структурно-геометрических параметров нейрона I_{CA} и $\frac{dI_{CA}}{dt}$. Таким образом обеспечивается участие процесса оптимизации энергетики в процессе переработки нервной клеткой полезной информации. Прямое влияние функции на энергетику происходит за счет дислокации ионов, а обратная связь обеспечивается путем изменения агрегации тигроида и митохондрий с выходом через Δa, β и I_{CA} на x_Σ и y.

Представленная модель была реализована в виде АЛГОЛ-программы для ЭЦВМ «БЭСМ-4» и исследована в основных режимах ее работы. Эксперименты показали адекватность изменения основных переменных модели наблюдаемым в нейроцитохимических и нейрофизиологических опытах особенностям поведения их морфологических прототипов и, кроме того, дали возможность уточнить внутреннюю структуру некоторых ее блоков.

Дальнейший анализ имеющихся функционально-топохимических данных позволяет осуществить последовательное расширение и детализацию модели нейрона: введение подсистем синтеза белка, неспецифических входных воздействий, регуляции проницаемости для предшественников энергетического и пластического обмена, параметрической регуляции со стороны циклического АМФ, аксоплазматического тока. По нашему мнению: а) только такое детальное отражение множества свойств нейрона в его модели делает ее системой *достаточно сложной* для адекватного воспроизведения поведения нервной клетки и взаимодействия нейронов в сети; б) описанная разработка может явиться начальным этапом синтеза высокоэффективных устройств, алгоритмов и систем искусственного интеллекта, управляющих систем роботов и т. п., так как она качественно изменяет основные характеристики, а следовательно, и *возможности нейробионического подхода* к такому синтезу, уже начинающему оправдывать себя на практике.

Список литературы: 1. *Растригин Л. А.* Случайный поиск в системах целенаправленного поведения.— В кн.: Проблемы адаптивного управления. Ростов н/Д., 1974, с. 104—119. 2. *Фицнер Л. Н.* Биологические поисковые системы. М., Наука, 1977. 136 с. 3. *Гринченко С. Н.*, *Загускин С. Л.*, *Загускина Л. Д.* Модель оптимизации энергетики нейрона.— Проблемы бионики. Харьков, 1975, вып. 15, с. 71—80. 4. *Гринченко С. Н.*, *Загускин С. Л.* Моделирование функционально-топохимических механизмов нейрона с использованием алгоритмов поисковой оптимизации.— В кн.: Математическая теория биологических процессов. Калининград, 1976, с. 383—386. 5. *Гринченко С. Н.*, *Загускин С. Л.* Устройство для моделирования адаптивного нейрона. А. с. 553635 (СССР).— Открытия. Изобрет. Пром. образцы. Товарные знаки, 1977, № 13, с. 190. 6. *Гринченко С. Н.*, *Загускин С. Л.* Устройство для моделирования нейрона. А. с. 553636 (СССР).— Открытия. Изобрет. Пром. образцы. Товарные знаки, 1977, № 13, с. 190. 7. *Гринченко С. Н., Загускин С. Л.* Устройство для моделирования адаптивного нейрона. А. с. 561198 (СССР).— Открытия. Изобрет. Пром. образцы. Товарные знаки, 1977, № 21, с. 166. 8. *Гринченко С. Н., Загускин С. Л.* Устройство для моделирования адаптивного нейрона. А. с. 565306 (СССР).— Открытия. Изобрет. Пром. образцы. Товарные знаки, 1977, № 26, с. 104—105. 9. *Загускин С. Л., Загускина Л. Д.* Динамика цитохромоксидазы при торможении рецепторного нейрона рака.— В кн.: Матер. XVI науч. конф. физиологов Юга РСФСР. Орджоникидзе, 1967, с. 141—142.

УДК 62.506.2

С. А. УСЕНКО

К ВОПРОСУ О ДИСКРЕТНОСТИ СЛУХОВОЙ ИНФОРМАЦИИ

В настоящее время в области распознавания, синтеза и передачи звукового сигнала на расстояние возник ряд технических задач, связанных с созданием эффективных устройств кодирования речевого сигнала. Одним из важнейших источников совершенствования этих устройств является изучение органа слуха человека, представляющего собой естественный образец ввода информации и существенно превосходящий имеющиеся на сегодняшний день технического устройства, связанные с переработкой акустической информации.

Задачей данной работы является попытка показать на основе психоакустических экспериментов, что количество информации, извлекаемое слуховым аппаратом человека из звуковой осциллограммы, является конечным. Эта конечная информация, представленная в дискретном виде, может быть введена в цифровые устройства вычислительной машины и обработана на дискретном уровне.

Обращаясь к исследованиям в области психофизики, мы находим подтверждение того положения, что информация, воспринимаемая органами чувств, может быть задана в дискретном виде. Это положение вытекает прежде всего из того факта, что любой сенсорный анализатор можно рассматривать как неидеальный прибор, воспринимающий и преобразующий непрерывную информацию и не реагирующий на слишком малые изменения характеристик преобразуемой информации.

В работе [1] на примере зрительного анализатора приводятся виды ограничений, присущие любому приемнику информации и позволяющие рассматривать эту информацию как конечную. К таким ограничениям автор относит разрешающую способность прибора, его ограниченную чувствительность, а также ограничение полосы пропускания. Рассмотрим последний вид ограничений подробнее, поскольку он имеет прямое отношение к вопросам, поставленным в настоящей статье.

В силу известной теоремы Котельникова [2] ограничение полосы пропускания эквивалентно тому, что при передаче информации вместо обычного непрерывного времени вводится условное дискретное время, соседние моменты которого отличаются друг от друга на весьма малый отрезок времени. В качестве такого отрезка времени выбирается максимальный отрезок, в течение которого рассматриваемый прибор оказывается неспособным различить изменения величины несущей информации. Поскольку теорема Котельникова применима исключительно к линейным системам, а процесс преобразования информации слуховым аппаратом человека носит нелинейный характер, мы вынуждены искать новые доказательства факта конечности акустической информации. Таким доказательством может служить обобщенный закон Тальбота для слуховых ощущений, описанный в работе [3], который позволяет рассматривать орган слуха как своеобразный фильтр, ограничивающий объем информации, которая поступает из внешнего мира в мозг человека. Заметим, что закон Тальбота свидетельствует о том, что некоторые существенно различные входные сигналы преобразуются органом слуха в одинаковые выходные сигналы. Указанные опыты [см. 3], очевидно, не исчерпывают всего класса сигналов, удовлетворяющих этому закону.

Для экспериментального доказательства дискретности слуховой информации необходимо предъявить испытуемому произвольную осциллограмму звукового сигнала $x(t)$ и поставить ей в соответствие дискретный сигнал $x_n(t)$, выбранный таким образом, чтобы

$$\int_{t_1}^{t_2} x(t)\,dt = \int_{t_1}^{t_2} x_n(t)\,dt. \qquad (1)$$

Согласно закону Тальбота ощущения, формируемые этими сигналами, должны совпадать. Поскольку $x_n(t)$ в равенстве (1) может быть задан неоднозначно, возникает вопрос, какой вид дискретного сигнала необходимо использовать для достижения целей эксперимента. С одной стороны, этот сигнал должен удовлетворять равенству (1), с другой — целесообразно представить его серией стандартных импульсов, следующих с частотой, превышающей критическую частоту слития прерывистого звука в непрерывный. Как показывают эксперименты [3], $f_{кр} \geqslant 30$ кГц. Информация, содержащаяся в сигнале и сформированная серией стандартных импульсов, может быть легко представлена двоичным кодом. Очевидно, что такой дискретный сигнал $x_n(t)$ можно получить путем частотно-импульсной модуляции сигнала $x(t)$.

Обращаясь к вопросам генерации импульсов в нейронных сетях, уместно заметить, что именно такой способ кодирования имеет место в слуховом анализаторе человека [4]. Возбуждение нейрона происходит при условии, что интенсивность воздействующего на нейрон раздражителя превышает некоторую величину — нижний порог чувствительности, или абсолютный порог. По мере увеличения интенсивности раздражителя возрастает количество или частота импульсов, генерацией которых нейрон откликается на данное раздражение.

Опишем аналитически такой вид преобразования. Допустим, на вход слухового аппарата поступает какой-то акустический стимул. В предположении дискретности выходного сигнала слухового аппарата это преобразование будет иметь вид, показанный на временной диаграмме (рис. 1), т. е. на выходе слухового аппарата появляется стандартный сигнал, например какой-либо импульс, когда

$$\int_0^t f(\tau)\, d\tau = E, \tag{2}$$

где E — энергия порога срабатывания.

Рис. 1. Временная диаграмма работы преобразователя: *а* — непрерывный входной сигнал; *б* — преобразованный дискретный сигнал.

Таким образом, информация в выходном сигнале будет переноситься, в основном, интервалом времени между импульсами T_n.

Найдем зависимость $T_n(t)$ от $f(t)$, т. е. функцию преобразования

$$\int_t^{t+T_n} f(\tau)\, d\tau = \Phi[t + T_n(t)] - \Phi(t) = E, \quad \text{где} \quad \Phi[t + T_n(t)] = \Phi(u).$$

Разложим $\Phi(u)$ в ряд Тейлора окрестности точки t:

$$\Phi[t + T_n(t)] = \Phi(t) + \Phi'(t)\frac{T_n(t)}{1} + \Phi''(t)\frac{[T_n(t)]}{2!} = \Phi(t) + f(t). \tag{3}$$

Поскольку минимальная частота следования импульсов не превышает 30 кГц, можно допустить, что в промежутке между кодовыми импульсами осциллограмма кривой сигнала $f(t)$ изменяется линейно во времени. Это дает возможность пренебречь значениями высших производных в разложении (3).

В первом приближении $\Phi[t + T_n(t)] = \Phi(t) + f(t)\, T_n(t)$, тогда

$$E = f(t)\, T_n(t), \quad \text{откуда} \quad T_n(t) = \frac{E}{|f(t)|}. \tag{4}$$

Во втором приближении

$$E = f(t)\,T_n(t) + f'(t)\,\frac{T_n^2(t)}{2}. \qquad (5)$$

Величина порога срабатывания для гармонического сигнала

$$E_{\min} = \int_0^t A_{\min}\sin\omega t = \frac{A_{\min}}{\omega}(1-\cos\omega t), \quad E_{\min} = \frac{2A_{\min}(\omega_{\max})}{\omega_{\max}},$$

т. е. минимальный порог определяется порогом слышимости на верхней граничной частоте и этой частотой.

Резюмируя сказанное, запишем функцию преобразования первого приближения:

1) положение во времени n-го импульса $t_n = \sum_1^n T_i$;

2) интервал времени от n до $n+1$ импульса $T_{n,\,n+1} = \frac{E}{|f(t_n)|}$;

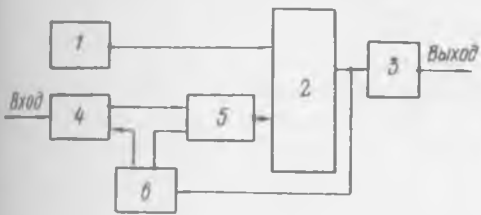
3) порог срабатывания связан с амплитудно-частотной характеристикой сигнала соотношением $E = 2A_{\min}(\omega_{\max})/\omega_{\max}$, где $A_{\min}$ — порог слышимости.

Мы получили аналитическое описание дискретного сигнала, который согласно закону Тальбота должен восприниматься человеком как аналоговый сигнал. Эксперименты по проверке этого утверждения проводились на специальном устройстве, реализующем зависимость (4). Блок-схема устройства изображена на рис. 2.

Схема работает следующим образом. Речевой сигнал в аналоговой форме поступает на вход накопительного элемента *4*. Накопительный элемент представляет собой интегрирующий усилитель, постоянная времени которого выбирается больше максимального интервала времени между импульсами. Значение сигнала на выходе накопительного элемента пропорционально площади осциллограммы звукового сигнала для каждого момента времени. Выход накопительного элемента подключен на вход триггера Шмидта *5*. В исходном состоянии на выходе триггера Шмидта установлен нулевой уровень. При достижении величины входного сигнала, равной порогу срабатывания, триггер Шмидта устанавливается в единичное состояние. Генератор тактовых импульсов *1* формирует серию стандартных прямоугольных импульсов финансированной частоты, необходимых для формирования двоичного кода. На выходе вентиля *2* будет установлен сигнал высокого уровня только в том случае, когда высокий уровень установится на обоих входах вентиля. Импульс с выхода вентиля подается на вход схемы гашения, которая обеспечивает быстрый разряд накопительного элемента и устанавливает на выходе триггера Шмидта нулевой уровень выходного сигнала. Далее процесс повторяется. Таким образом с выхода вентиля на вход формирователя поступает преобразованный дискретный сигнал. Схема формирователя *3* обеспечивает получение выходных импульсов стандартной длительности.

Как видно из работы схемы, описанная экспериментальная установка — это преобразователь аналог-код интегрирующего типа, который может быть использован для ввода речевого сигнала в ЭВМ. В экспериментах по восприятию дискретного сигнала участвовало три человека. Акустический сигнал, записанный на магнитофон, поступал на преобразователь аналог-код. Дискретный сигнал с выхода преобразователя подавался на телефон.

Экспериментатор имел возможность при помощи специального тумблера предъявлять испытуемому поочередно дискретный и аналоговый сигналы. В первой серии экспериментов на вход преобразователя подавались сигналы синусоидального вида. Осциллографом, подключенным к наушникам, отмечался факт преобразования синусоиды в стандартные импульсы. Уровень аналогового



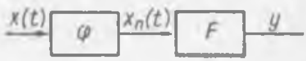


Рис. 2. Функциональная схема преобразователя.

Рис. 3. Блок-схема преобразования слуховой информации.

сигнала равнялся 50 дб. При этом частота следования импульсов составляла 40 кГц. Во всех опытах было отмечено тождество звучания дискретного и аналогового сигналов, т. е. испытуемые воспринимали дискретный сигнал как чистый тон. Во второй серии экспериментов в качестве входного сигнала использовали дикторскую речь и музыкальные произведения.

Эксперименты показали, что непрерывный и дискретный сигналы по слуховому ощущению неотличимы; это полностью удовлетворяет закону Тальбота.

По своему способу построения дискретный сигнал содержит конечное количество информации. Отсюда можно сделать вывод, что информация, содержащаяся в слуховом ощущении, также конечна и может быть представлена в дискретном виде для ввода в ЭВМ с целью обработки на цифровом уровне. Описанный способ дискретизации позволяет не только ввести информацию в ЭВМ, но также и вывести ее из машины в виде акустического сигнала без предварительного сглаживания. Выведенная таким способом информация может быть воспринята ухом как ощущение, неотличимое от ощущения, формируемого непрерывной осциллограммой.

Таким образом, вывод о дискретности информации, содержащейся в слуховом ощущении, может послужить базой для разработки устройств анализа и синтеза речи. Вторым важным моментом, вытекающим из положения о дискретности слухового восприятия, является возможность представить преобразованный сигнал в слуховой системе как суперпозицию двух преобразований: 1. Аналого-цифровое преобразование по предложенному нами рецепту. 2. Чисто дискретное преобразование (рис. 3).

Данный вывод не зависит от наших анатомо-физиологических данных, касающихся вопросов импульсной генерации в нейронных сетях, а логически следует из того факта, что дискретный сигнал, формируемый по нашему рецепту, абсолютно не различим на слух от непрерывного сигнала. Обращаясь с этим выводом к техническим задачам, мы получаем возможность вырабатывать рекомендации по моделированию процессов переработки информации слуховым анализатором человека. Согласно схеме (рис. 3) следует сначала производить аналого-цифровое преобразование φ, а уже затем — все дальнейшие преобразования F, требуемые по слуховым моделям. Последние преобразования информации проводятся на дискретном уровне с помощью ЭВМ.

Данные электро-физиологических исследований слухового аппарата позволяют предположить, что такое последовательное соединение преобразователей действительно используется слуховой системой.

Если представление функции в виде суперпозиции двух функций невозможно, значит, невозможна и пробная реализация этих функций в виде последовательного соединения двух блоков. Однако это не исключает возможности приборной реализации двух функций в виде последовательного соединения двух блоков. Выводы структурного характера из функционального результата рискованы и относиться к ним следует осторожно, поскольку они имеют чисто гипотетический характер.

Список литературы: 1. *Глушков В. М.* Введение в кибернетику. Киев, Изд-во АН УССР, 1968. 323 с. 2. *Котельников В. А.* О пропускной способности эфира и проволоки в электросвязи.— В кн.: Материалы к 1-му Всесоюзному съезду по вопросам технической реконструкции дела связи и развития слаботочной промышленности. М., Изд-во Ред. упр. связи РККА, 1933, с. 111—124. 3. Эффект сглаживания в слухе.— Проблемы бионики. Харьков, 1977, вып. 19, с. 31—37. Авт.: *А. Я. Абрамов, С. А. Усенко, Ю. П. Шабанов-Кушнаренко.* 4. *Лабутин В. К., Молчанов А. П.* Слух и анализ сигналов. М., 1967, с. 79.

УДК 62.506.2

Л. Н. БОНДАРЕВА, А. И. КОВАЛЕВ, Л. С. КРИВОШЕИНА,
В. А. ЛОВИЦКИЙ, В. А. ЛОГИНОВ

ФОРМИРОВАНИЕ «ЗНАНИЙ» УНИВЕРСАЛЬНОГО РЕШАТЕЛЯ ЗАДАЧ С ПОМОЩЬЮ СИСТЕМЫ ФОРМИРОВАНИЯ ПОНЯТИЙ. *СООБЩЕНИЕ 1*

При построении систем с искусственным интеллектом важнейшее значение приобретает проблема формирования, хранения, обобщения и использования знаний. Ее решение связано с построением универсальной системы формирования понятий. Согласно работе [1, с. 12] определим понятие следующим образом.

Понятие — обобщенная информация о множестве объектов, представленных наборами значений признаков, заданных как в явном, так и в неявном виде [2], которая: а) отображает характерные для этого множества логические отношения между отдельными значениями признаков; б) является достаточной для различения с помощью некоторого правила распознавания объектов, принадлежащих множеству, от объектов, не принадлежащих ему.

Формирование знаний у систем с искусственным интеллектом должно осуществляться универсальной системой формирования понятий (УСФП), которая, по нашему мнению, будет включать в себя следующие системы формирования понятий (СФП):

1. Систему формирования индуктивных понятий, как с независимым, так и с зависимым выбором [1—6].
2. СФП, работающую с объектами, обобщенная информация о которых задается путем перечисления самих объектов. Например, понятие «русская гласная буква» формируется путем перечисления букв: А, Е, ..., Я.
3. СФП, обобщающую объекты по аналогии [7; 8, с. 127].
4. СФП, анализирующую объекты, которые заданы путем описания [9]. В этом случае объект может, например, задаваться определением его назначения.
5. СФП, осуществляющая формирование понятия путем переноса. Так, например, ребенок, однажды выделив цвет, форму, величину в предмете, начинает выделять эти признаки и в других сходных предметах при сходных обстоятельствах [10, с. 72].

Очевидно, что УСФП должна обеспечить формирование сходных, смежных и контрастных понятий. В данной работе не будет рассматриваться ни принцип формирования перечисленных понятий, ни конструкции указанных СФП. Нас в первую очередь интересует практический аспект использования УСФП для формирования знаний системы. Поэтому на примере универсального решателя арифметических задач на движение (УРАЗ) рассмотрим систему знаний УРАЗ, формирование которой и должна осуществлять УСФП.

В чем же состоит универсальность такого решателя задач? Предположим, что необходимо построить на базе ЭЦВМ систему, способную решать задачи определенного класса. Можно разработать алгоритм решения каждой задачи из данного класса, записать их на выбранном алгоритмическом языке и составленные таким образом программы решения каждой задачи вложить в ЭЦВМ. После этого достаточно будет указать номер задачи, подлежащей решению, как ЭЦВМ мгновенно найдет соответствующую программу и решит эту задачу. В то же время ЭЦВМ будет бессильна, если предложить ей решить задачу, аналогичную рассмотренным ранее, но для которой не была заранее составлена программа. Простейший пример. ЭЦВМ «научили» находить скорость движения тела, если известны путь и время, а затем ей предложили найти путь по скорости движения тела и времени, затраченному телом на прохождение этого пути. Вычислительная машина не сможет решить ее по той причине, что она не формирует алгоритм решения задачи, а просто слепо использует алгоритм, вложенный в нее исследователем.

Таким образом, универсальность предлагаемого решателя задач состоит в том, что он не использует заранее составленные алгоритмы, а строит их сам, соотнося с условиями заданных задач.

Разработке подобных решателей задач посвящен ряд работ как советских, так и зарубежных авторов [8—9; 11—14]. Несмотря на кажущееся единство указанных исследований, фактически каждое из них преследует различные цели. Это связано с тем, что проблема построения универсального решателя задач представляет собой комплексную проблему, направленную на изучение и реализацию следующих задач: 1) грамматический анализ условия задачи, заданного на естественном языке, с целью его формализации, т. е. представления условия в виде системы формальных отношений, допускающих простые преобразования; 2) построение семантического базиса (СБ) решателя, представляющего знания системы о понятиях, используемых при решении того или иного класса задач, причем СБ должен быть расширяющимся, т. е. необходимо предусмотреть возможность включения в СБ новых понятий; 3) разработка алгоритма построения дерева цели и подцелей; 4) построение системы преобразования и решения алгоритмических уравнений (для универсальных решателей арифметических задач).

Большое внимание [11, 13] уделяется решению первой из перечисленных задач. При построении систем, способных решать арифметические задачи, центральным был вопрос «понимания» системами словесного условия арифметических задач. Практически разработанные системы могли решать задачи простейшего типа. Например: «У одного человека M атомов, у второго — N атомов. Сколько атомов у них вместе?» [13, с. 157].

Вторая и третья задачи рассматриваются авторами в комплексе [8; 9; 12], причем каждый из авторов представляет «знания» системы и алгоритм достижения цели по-разному. Анализ особенностей этих систем не входит в задачу данной статьи.

Система преобразования и решения алгебраических выражений (задача 4), по сути, должна входить в математическое обеспечение современных ЭЦВМ. Разработанные системы, как правило, ориентированы на конкретные вычислительные машины [15], а не на универсальные алгоритмические языки типа ФОРТРАН и ПЛ-1.

Рассмотрим основные принципы построения УРАЗ на конкретной арифметической задаче на движение: «Катер, скорость которого в стоячей воде 15 км/ч, отправился от речного причала вниз по течению реки и, пройдя 36 км, догнал плот, отправленный от того же причала за 10 часов до отправления катера. Найти скорость течения реки».

Для упрощения решения проблемы «понимания» УРАЗ условия задачи, представленного на естественном языке, был введен специализированный русский язык (в рамках данной статьи строгое описание этого языка рассматриваться не будет). Применив введенный язык, пользователь представляет условие задачи в следующем виде: 1. Катер двигался по течению реки. 2. Катер прошел путь ='36' км, где ='N' используется для обозначения числовых констант. 3. Скорость катера в стоячей воде ='15' км/ч. 4. Катер догнал плот. 5. Катер вышел через ='10' ч после плота. 6. Найти скорость течения реки.

Для «понимания» такого условия задачи УРАЗ должна обладать определенными знаниями, но что она должна знать, УРАЗ не знает и в этом ей должен помочь человек, определив начальную организацию структуры знаний и функцию их пополнения и коррекции с помощью УСФП. Значения УРАЗ будут представлены семантическим базисом или библиотекой формализованных понятий (БФП). Формирование БФП осуществляется УСФП, или путем перечисления понятий, или путем объяснения УРАЗ их значения в режиме диалога. Пусть путем перечисления понятий была сформирована следующая БФП:

Имена понятий	Семантика
Скорость объекта I	$VI = SI/TI$
Путь объекта I	SI
Объект I двигался по течению реки	$VPI = VI + VT$
Путь объекта I по течению	$SPI = VPI * TPI$
Объект I вышел за объектом J из того же пункта и догнал объект J	$SI = SJ$ $TSI = TSJ$
Время прихода объекта I	$TSI = TBI + TI$
Время выхода объекта I	TBT
Объект I вышел через X часов после объекта J	$TBI = TBJ + X$, если TBJ неизвестно, то $TBJ = 0$, а $TBI = X$
Время движения объекта I по течению	$TPI = TSI - TBI$
Скорость течения	VT
Скорость плота	$VPL = VT$

В БФП имена понятий представлены пирамидальной структурой [16; 17], а семантика — И/ИЛИ-структурой [12]. Эти структуры связаны между собой общими элементами.

«Понимание» задачи УРАЗом сводится к замене словесного описания задачи формальным. Суть этого преобразования связана с формированием четырех массивов: ДАНО, ЗНАЧЕНИЯ, УСЛОВИЕ, ЦЕЛЬ. В массив ДАНО заносятся сокращенные обозначения объектов и обозначения их характеристик, численные значения которых заданы в условии задачи. В массив ЗНАЧЕНИЯ заносятся числовые константы и устанавливаются указатели к соответствующим элементам массива ДАНО. В массив УСЛОВИЕ записываются формальные описания ситуаций условия задачи, полученные с помощью БФП. В массив ЦЕЛЬ заносится то, что нужно найти или доказать в задаче.

Для рассматриваемой задачи содержимое этих массивов будет следующим: ДАНО: K; PL; VK; SPK; X. ЗНАЧЕНИЯ: —; —; 15; 36; 10. УСЛОВИЕ: $VPK=VK+VT$; $SPK=VPK*TPK$; $SK=SPL$; $TSK=TSPL$; $SK=SPK$; $TBK=TBPL+X$. ЦЕЛЬ: VT.

«Понимание» условия задачи УРАЗом осуществляется в двух режимах. Работая в первом режиме, УРАЗ «старается» самостоятельно «понять» описание ситуации. Так, прочитав описание ситуации «Катер двигался по течению реки», УРАЗ, работая с пирамидальной структурой БФП, находит имя понятия («Объект I двигался по течению реки») и делает вывод, что «Объект I» — это «катер». На основании данного вывода, во-первых, формируется обозначение объекта «K» и вводится в массив ДАНО, во-вторых, определяется правило замены: $I\rightarrow K$ (символ $\rightarrow$ читается как «I заменяется K»),— с помощью которого выражение $VPI=VI+VT$ преобразуется в $VPK=VK+VT$, а затем преобразованная семантическая интерпретация анализируемой ситуации заносится в массив УСЛОВИЕ.

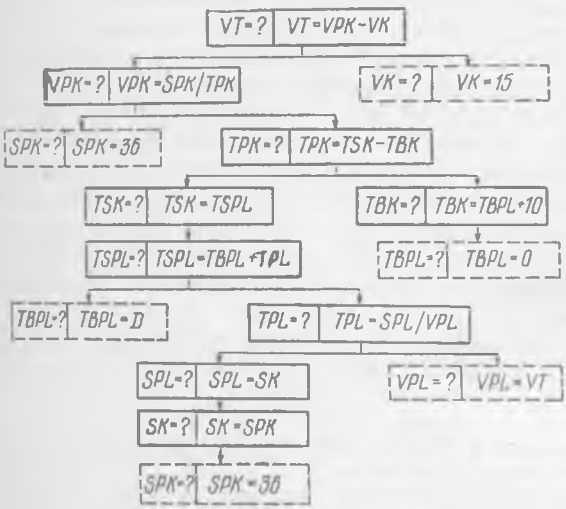
Переход ко второму режиму работы (режиму диалога) происходит в том случае, когда одной и той же ситуации соответствуют в равной степени несколько (не менее двух) имен понятий или когда УРАЗ «не уверен», что найденное имя понятия соответствует рассматриваемой ситуации. Например, при анализе ситуации «Катер догнал плот» УРАЗ находит в БФП имя понятия: «Объект I вышел за объектом J из того же пункта и догнал объект J». Зная, что «объект I» — это «катер», УРАЗ делает вывод, что «объект J» — это «плот». Тогда найденное имя понятия будет преобразовано следующим образом: «Катер вышел за плотом из того же пункта и догнал плот». Поскольку УРАЗ не располагает информацией о том, откуда вышли «катер» и «плот», он задает пользователю вопрос: «Катер вышел за плотом из того же пункта?» При положительном ответе УРАЗ полностью идентифицирует данное имя понятия и, осуществив соответствующие замены ($I\rightarrow K$, $J\rightarrow PL$), заносит в массив УСЛОВИЕ выражения $SK=SPL$ и $TSK=TSPL$.

Более совершенный УРАЗ, перед тем как задать вопрос пользователю, просматривает описания всех ситуаций до конца, чтобы найти ответ на поставленный самому себе вопрос, и только в случае неудачи обращаемся за помощью к пользователю.

К режиму диалога УРАЗ прибегает при построении дерева цели и подцелей в случае, когда какая-либо величина оказывается неопределенной (так, тот факт, что $SK=SPK$, сообщается системе пользователем). Для простоты данное утверждение заранее включено в массив УСЛОВИЕ.

Заметим, что семантическая интерпретация не для всех имен понятий БФП задана в явном виде. Например, для определения семантики понятия, имя которого «Путь объекта I», программы, обслуживающие БФП, найдут, что для этого нужно преобразовать формулу $VI = SI/TI$. С этой целью к работе будет подключен блок алгебраических преобразований (БАП), в результате работы которого ST будет определено как $VI * TI$.

После анализа словесного описания всех ситуаций, составляющих условие задачи, и окончания формирования содержимого массивов — ДАНО, ЗНАЧЕНИЯ, УСЛОВИЕ, ЦЕЛЬ — УРАЗ переходит к поиску решения задачи. В основу поиска положен известный метод, идея которого состоит в «размышлении в обратном направлении от задачи, которую предстоит решить, с тем чтобы выделить подзадачи, подподзадачи и т. д., пока, наконец,



первоначальная задача не будет сведена к набору тривиальных элементарных задач» [12, с. 91]. В данном случае под элементарной задачей будем понимать выражение вида $a = S$, где a — переменная; S — арифметическое выражение (например, определенное по правилам ФОРТРАН), у которого S или не содержит неизвестных величин, или неизвестная величина в S представляет собой цель. Кроме того, вводится понятие условно элементарной задачи. Данная задача будет иметь место, если в S будет входить одно или несколько неизвестных, не являющихся целью, для разрешения которых не могут быть заданы подзадачи.

Реализация метода редукции задачи, или сведения задачи к совокупности подзадач приводит к построению И/ИЛИ-графа [12]. Начальная вершина такого графа соответствует цели исходной задачи. Цель процесса поиска решения, осуществляемого на И/ИЛИ-графе,— показать, что начальная вершина разрешима. Определение разрешимости и неразрешимости вершин в И/ИЛИ-графе приведено в работе [12]. На рисунке показан И/ИЛИ-граф решения данной задачи. В двойных рамках указаны элемента ные задачи. Если при построении И/ИЛИ-графа какая-либо по задача не может быть сформулирована на основании содержимо массива УСЛОВИЕ, то УРАЗ обращается к БФП.

После доказательства разрешимости или неразрешимости н чальной вершины построение И/ИЛИ-графа считается закончен ным. В первом случае УРАЗ, используя у к а з а т е л и в о з в р а т определяет результирующее выражение для цели: 1) $TPL=36/VT$ 2) $TPK=36/VT-10$; 3) $VPK=36/(36/VT-10)$; 4) $VT=36/(36/VT-$ $-10)-15$.

Практически УРАЗ вначале получает результирующее выр жение в общем виде, а затем блок алгебраических преобразов ний приводит результирующее выражение к виду XVT^2+ $+XVKVT-VKSPK=0$. Если заданы численные значения пер менных, блок подстановки осуществляет соответствующие замен и производит возможные сокращения: $VT^2+15VT-54=0$, наконец, блок решения алгебраических уравнений получает р зультат $VT=3$ км/ч. Специальный блок унификации (БУ) след за соблюдением соответствия между единицами измерения зада ных численных значений переменных. Так, если скорость задан в км/ч, а расстояние — в м, то БУ преобразует последнюю вел чину таким образом, чтобы она тоже измерялась в км. Есл конкретные значения переменных не заданы, уравнение решаетс в общем виде.

УСФП реализована на языках ФОРТРАН и АССЕМБЛЕ и ориентирована на ЕС ЭВМ, а УРАЗ полностью описан на язы ФОРТРАН-4.

Список литературы: 1. *Гладун В. П.* Эвристический поиск в сложных сред Киев, Наукова думка, 1977. 166 с. 2. *Ловицкий В. А.* Система формирован понятий для объектов, заданных неявным набором признаков.— Пробле бионики. Харьков, 1977, вып. 18, с. 73—81. 3. *Бонгард М. М.* Пробле узнавания. М., Наука, 1967. 320 с. 4. *Бенерджи Р.* Теория решения задач. Мир, 1972. 224 с. 5. *Хант Э., Марин Дж., Стоун Ф.* Моделирование проце формирования понятий на вычислительной машине. М., Мир, 1970. 301 6. *Погосян Э. М.* К теории автоматического синтеза понятий.— Семиоти и информатика. М., 1977, вып. 8, с. 125—152. 7. *Рейтман У.* Познание и мы ление. М., Мир, 1968. 400 с. 8. *Слейгл Дж.* Искусственный интеллект. Мир, 1973. 319 с. 9. *Тыугу Э., Унт М.* Эксперименты с решателем вычис тельных задач.— В кн.: Труды IV Международной объединенной конферен по искусственному интеллекту, кн. 3. М., Изд-во АН СССР, 1976, с. 3.17 3.182. 10. *Катаева А. А., Ким С. Г.* Восприятие и обобщение величи умственно отсталых дошкольников.— Дефектология, 1976, № 6, с. 72— 11. *Kazuya S., Masao J., Hideo J.* A. system solving arithmetic problems somputer.— Scientific and Engineering Review Doshisha University, 1973, v. No 4, p. 175-193. 12. *Нильсон Н.* Искусственный интеллект. М., Мир, 19 301 с. 13. *Мальковский М. Г.* Программа APRIL, решающая арифметичес задачи в словесной формулировке.— Алгоритмы и алгоритмические языки. 1973, вып. 6, с. 113—159. 14. *Кац Б. Г.* Арифметические задачи на движ ние.— Автоматика и телемеханика, 1972, № 2, с. 109—112. 15. *Аксельрод И.* *Белоус Л. Ф.* Входной язык системы автоматического программирован «Сириус». Харьков, Изд-во ХГУ, 1969. 68 с. 16. *Khalil T. M., Lovitsky V. A.* Structure of memory in concept formation. — In: IEEE Conference on System

Man and Cybernetics. Boston, Mass, 1973, p. 346-347. 17. *Khalil T. M., Lovitsky V. A.* A process of concept formation.— In: Proceeding of the Human Factors Society Annual Meeting. Washington, D. C., oct. 1973, p. 509-512.

УДК 007:573.6

М. В. ВОЛЧЕНКО

ВОПРОСНО-ОТВЕТНАЯ СИСТЕМА: ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ, ПОИСК ОТВЕТА

При построении вопросно-ответной системы возникают проблемы, связанные с анализом естественного языка. Это прежде всего проблема представления информации, состоящая в выборе средств для максимально точной передачи смысла текста, данного на естественном языке. В настоящее время интенсивно разрабатывается представление информации в виде семантических сетей [1]. Сходный с указанным способ применяется в ситуационном управлении [2]. При таком подходе создание вопросно-ответной системы предполагает разработку формализованного языка, который стал бы «посредником» между естественным языком и представлением информации в виде семантической сети. По структуре этот язык-посредник должен быть приближенным к естественному, что обеспечит простоту перевода и большую точность передачи смысла переводимого текста. Предлагаемый язык вопросно-ответной системы (язык Q) строился в рамках разработки ориентированного на семантические сети алгоритмического языка АЯКС [3]. Возможно установление однозначного соответствия между языком Q и семантической сетью.

Еще одна существенная проблема связана с наличием различных по характеру и сложности вопросов. Алгоритмы поиска ответа, основанные на применении различных процедур доказательства, разработаны лишь для упрощенного фрагмента естественного языка, для перевода которого достаточен язык логики предикатов 1-го порядка [4—6]. При этом, как правило, рассматривались лишь некоторые, произвольно взятые виды вопросов, выбор которых был обусловлен их простотой. В данной работе задача распространения метода доказательств на любые виды вопросов решается на основании их классификации. С помощью языка Q описан алгоритм поиска ответа на любой вопрос. В качестве иллюстраций берутся различные по простоте предложения естественного языка, принадлежащие к различным областям знания. Такой выбор обусловлен желанием показать независимость языка Q от предметной области.

1. **Представление информации.** Язык Q является расширением языка прикладного функционального исчисления. Алфавит включает следующие элементы: x, x_1, x_2, ... — индивидные переменные; a, a_1, a_2, ... — индивидные константы; $\sim$, &, $\vee$, $\supset$, $\equiv$,

$\forall$, $\exists$ — логические константы — отрицание, конъюнкция, дизъюнкция, импликация, эквивалентность, квантор общности, квантор существования; $\in$ — отношение принадлежности; функции: $p(x)$ — сопоставляет x-у множество его признаков; $r(x_1, \ldots, x_n)$ — сопоставляет n-ке множество n-местных отношений между ее элементами; $c(x)$ — сопоставляет x-у множество его условий (причин); $s(x)$ — сопоставляет x-у множество следствий из него; операторы: (?) — оператор вопроса I типа; (?x) — оператор вопроса II типа. Два оператора соответствуют двум основным типам вопросов, различаемым в логике: вопросам I типа — подтверждающим (требующим ответа «да» или «нет») и вопросам II типа — восполняющим (начинающимся с вопросительного слова или словосочетания) [7].

Существенной особенностью языка Q является введение единственного предиката (отношения принадлежности), определенного на функциях, несущих основную смысловую нагрузку. Использование такой схемы снимает проблемы, связанные с переходом к более высоким, чем первый, порядкам [6, с. 306—308]. В языке Q нет кванторов по предикатным переменным, что не снижает его выразительных возможностей, поскольку второпорядковые конструкции могут быть построены с помощью функций. Например, выражения «все отношения между a_1 и a_2» и «некоторые свойства a» представляются как $\forall x\,[x \in r(a_1, a_2)]$ и $\exists x\,[x \in p(a)]$. Очевидно, что в данной формализации не требуется вводить различные сорта переменных, а также накладывать какие-либо ограничения на сложность представляемых выражений естественного языка.

Отдельно следует отметить свойства функций $c(x)$ и $s(x)$, используемых для представления условных предложений. Эти функции взаимозаменимы, т. е. $\forall x_1\,\forall x_2\,[x_1 \in c(x_2) \equiv x_2 \in s(x_1)]$. Фактически они всегда служат для выражения конструкций более высокого, чем второй, порядка. В этом случае представляемое предложение является сложным и не только указывает на связь и индивиды, но и включает их дополнительное описание. Такие выражения представляются с помощью импликации, причем «дополнительная» информация записывается слева от знака импликации. Например, предложение «Медь электропроводна, потому что она является металлом» принимает вид [электропр. $\in p$ (медь) & металл $\in p$ (медь)] $\supset$ металл $\in c$ (электропр.). Эта запись читается следующим образом: «Если электропроводность есть признак меди и металл есть признак меди, то металл есть причина электропроводности».

Для вопросов принимается аналогичная запись. Например, вопросы «Верно ли, что медь тяжелее железа?» и «Почему ртуть жидкая?» запишутся соответственно как (?) [тяжелее $\in p$ (медь, железо)]; (?x) [жидкая $\in p$ (ртуть) $\supset x \in c$ (жидкая)].

На следующем этапе формализации необходимо определить понятия элементарной подформулы (ЭП), правильно построенной формулы (ППФ) и особой разновидности ППФ — правильно поставленного вопроса (ППВ). ЭП есть подформула, содержащая отношение принадлежности и не содержащая никаких логических констант, кроме, возможно, отрицания. ППФ — формула, не содержащая свободных переменных. Элементарная ППФ — это формула, представляющая простое суждение.

(I). Пусть A есть ЭП, не содержащая переменных, тогда A — элементарная ППФ.

(II). Пусть A есть ЭП и $x_1, \ldots, x_n$ — все, входящие в нее переменные, тогда $\Join x_1 \ldots \Join x_n A$ — элементарная ППФ, где $\Join$ — квантор общности или существования.

(III). Пусть A есть ППФ, а B — элементарная ППФ вида (I) или (II), тогда $A \supset B$ — элементарная ППФ.

(IV). Пусть $t_1, \ldots, t_n$ есть ЭП, а x — единственная свободная переменная, входящая во все $t_1, \ldots, t_n$, тогда $\exists x [t_1 \& \ldots \& t_n]$ и $S_x^a \exists x [t_1 \& \ldots \& t_n] |$ — элементарные ППФ, где $S|$ — знак подстановки.

(V). Пусть A, B — ППФ, тогда $\sim A$, $A \& B$, $A \vee B$, $A \supset B$, $A \equiv B$ — ППФ.

(VI). Пусть A есть ППФ, тогда $(?) A$ — ППВ I типа.

(VII). Пусть $x_1, \ldots, x_n$ — все свободные переменные формулы A, тогда $(?x_1 \& \ldots \& ?x_n) A$ и $(?x_1 \vee \ldots \vee ?x_n) A$ — ППВ II типа.

Определение ППФ дополняется классификацией вопросов.

Классификация простых вопросов. I тип. Вопросы вида $(?) [A \supset B]$, где $A \supset B$ — ППФ и B содержит только одну ЭП:

1. Безусловные вопросы (A, возможно, пусто): 1.1. B содержит $p(x)$; 1.2. B содержит $r(x_1, \ldots, x_n)$.

2. Условные вопросы (A не пусто): 2.1. B содержит $c(x)$; 2.2. B содержит $s(x)$.

II тип. Вопросы вида $(?x) [A \supset B]$, где x — единственная свободная переменная формулы $A \supset B$, а x входит в элементарную подформулу B. Дальнейшее деление на безусловные и условные вопросы такое же, как и в I типе.

Классификация сложных вопросов. I тип. 1. Соединительные вопросы — $(?) [A \supset B \& C]$; 2. Разделительные вопросы — $(?) [A \supset B \vee C]$, где A, B, C — ППФ.

II тип. 1. Вопросы с одним оператором: 1.1. Соединительные — $(?x) [A \supset B \& C]$; 1.2. Разделительные — $(?x) [A \supset B \vee C]$, где x — единственная свободная переменная формулы, стоящей под оператором вопроса, x входит в B и x входит в C; 2. Вопросы с несколькими операторами: 2.1. Соединительные — $(?x_1 \& \ldots \& ?x_n) A$; 2.2. Разделительные — $(?x_1 \vee \ldots \vee ?x_n) A$, где $x_1, \ldots, x_n$ — все свободные переменные формулы A.

Примечание 1. Приведенные формулы являются схемами, т. е. вместо B, C можно подставлять сложные выражения, получая, например, такие вопросы, как $(?) [A \supset B_1 \& \ldots \& B_n]$ или $(?x) [A \supset B_1 \& (B_2 \vee B_3)]$.

Примечание 2. Деление сложных вопросов на безусловные и условные осуществляется так же, как и в случае простых вопросов. Сложные условные вопросы могут быть условно-соединительными или условно-разделительными.

Рассмотрим для иллюстрации несколько примеров записи сложных вопросов:

— Верно ли, что Эдгар По родился в Америке и писал стихи? (?) [родился $\in r$ (Э. По, Америка) & писал $\in r$ (Э. По, стихи)];

— Кто изобрел телескоп и радио? $(?x)$ [изобрел $\in r$ (x, телескоп) & изобрел $\in r$ (x, радио)];

— Почему железо электропроводно и тяжелее воды? $(?x)$ {[электропр. $\in p$ (железо) & тяжелее $\in r$ (железо, вода)] $\supset$ [$x \in c$ (электропр.) & $x \in c$ (тяжелее)]};

— Когда и где родился Чехов? $(?x_1 \,\&\, ?x_2)$ [родился $\in r$ (Чехов, x_1) & & родился $\in r$ (Чехов, x_2)].

Примечание 3. В случае вопросов с несколькими операторами индексы обозначают различные вопросительные слова. Например, переменные в вопросе «Когда и где родился Чехов?» фактически должны быть представлены как «$x_{\text{когда}}$» и «$x_{\text{где}}$».

Аксиоматизация языка Q предполагает введение нескольких систем аксиом, описывающих различные по характеру элементы алфавита: АО — обычные аксиомы для логических констант [8, с. 26—27, 155—156]; АФ—аксиомы для функциональных констант; А? — аксиомы для вопросных операторов, описывающие сведение сложных вопросов к простым. Это сведение состоит в удалении знаков конъюнкции и дизъюнкции из области действия вопросных операторов. Каждому указанному в классификации виду сложного вопроса соответствует особая аксиома.

Поскольку ППФ всегда замкнута, вводятся правила замены связанных переменных: замена осуществляется для всех вхождений переменной в область действия данного квантора. При этом вводятся два правила — переименование и подстановка, которые соответственно имеют вид $S^{x_i}_{x_j} \Join x_j A \mid \underline{\overline{\circ}} \Join x_i A$ и $S^{a}_{x} \Join x A \mid \underline{\overline{\circ}} A$, где $S|$ — знак подстановки; $\Join$ — квантор общности или существования; $\underline{\overline{\circ}}$ — знак графического равенства.

2. Поиск ответа. При постановке вопроса предполагается, что значение терминов, в которых он сформулирован, является вполне определенным и известно тому, кто будет отвечать на вопрос. Кроме того, предполагается истинной «дополнительная» информация о связях между этими терминами. Данное базовое знание может быть выявлено и записано в виде некоторого утверждения, называемого *предпосылкой вопроса*. Рассмотрим, например, вопрос «Почему Земля вращается?» $(?x)$ [вращается $\in p$ (Земля) $\supset x \in c$ (вращается)]. Предпосылка этого вопроса имеет вид $\exists x$ [$x \in p$ (Земля)] & $\exists x$ [$x \in p$ (вращается)] & вращ. $\in p$ (Земля). В общем случае предпосылка имеет вид конъюнкции $\exists x\,[x \in p\,(a_1)]\ \&\ \ldots\ \&\ \exists x\,[x \in p \times (a_n)]\ \&\ t_1\ \&\ \ldots\ \&\ t_k$, где $a_1, \ldots, a_n$ — все, используемые в формулировке вопроса, индивидные константы, а $t_1, \ldots, t_k$ — не содержащие переменных ЭП, указывающие на наличие определенных связей между $a_1, \ldots, a_n$. Если предпосылка оказалась ложным утверждением, вопрос называется *некорректным*.

Областью поиска ответа (ОП) предлагается называть формулу, описывающую логическую структуру ответа или альтернативу возможных структур. ОП простого вопроса I типа — $(?)\,A$ — имеет вид $A \vee \sim A$, т. е. представляет собой альтернативу утвердительного и отрицательного ответов. ОП простого вопроса II типа — $(?x)\,A$ — имеет вид $\exists x' A$ и представляет собой матрицу суждения-ответа. При этом отмеченная переменная связана квантором существования, заменившим вопросный оператор. Например, ОП вопроса «Кто изобрел радио?» имеет вид $\exists x_{\text{кто}}$ [изобрел $\in r$ ($x_{\text{кто}}$, радио)].

Формулы предпосылки и ОП будут использованы при описании алгоритма поиска ответа. При этом возможны следующие виды ответов на простой вопрос I типа [$(?)\,A$]: 1) A; 2) $\sim A$; 3) вопрос некорректен (неизвестное значение таких-то терминов или неверны такие-то связи). Для простого вопроса II типа — $(?x)\,A$ — также имеется три варианта ответов: 1) $a_1, \ldots, a_n$ — константы (— а), появляющиеся в результате удаления квантора существования, заменившего вопросный оператор; 2) $\sim \exists x' A$ — отсутствует связь, на которую указывает вопросный оператор (в естественном языке такими являются ответы вида «никто», «нигде» и т. п.); 3) вопрос некорректен.

В процессе поиска ответа система, отвечающая на вопрос, должна решить следующие задачи:

I. Проанализировать структуру вопроса и, если он является сложным, свести его к простым с помощью системы аксиом (А?) и по отдельности рассматривать полученные простые вопросы.

II. Если это необходимо, свести встречающиеся в формулировке вопроса производные термины к базовым с помощью соответствующей системы определений.

III. Определить корректность вопроса и, если имеет место некорректность, указать, какой термин неопределен или какая связь неверна. Это осуществляется через выявление предпосылки вопроса.

IV. Используемая для поиска ответа процедура доказательства основана на допущении отрицательного ответа и последующем поиске контрпримера. Очевидно, что она применима к вопросам I типа, ОП которых имеет вид $A \vee \sim A$. В случае вопросов II типа требуется найти способ их сведения к такому виду, который позволил бы применить тот же метод.

К *процедуре доказательства* предъявляется требование найти для некоторого допущения контрпример или подтверждение либо указать на недостаточность базовой информации. В случае *I типа* таким допущением является отрицательный ответ, а утвердительный ответ играет роль контрпримера. С вопросом *II типа* предлагается поступить следующим образом: 1. Для вопроса $(?x)\,A$ строится ОП II типа — $\exists x' A$. 2. Ставится вопрос I типа об истинности полученной формулы — $(?)\,\exists x' A$. 3. Строится формула —

ОП I типа — $\exists x'A \vee \sim \exists x'A$. 4. Берется отрицательное допущение — $\sim \exists x'A$.

Проблемы, связанные со сколемизацией квантора существования [9, с. 448], здесь не возникают, поскольку, во-первых, сколемизируется только один квантор, заменивший вопросный оператор, и, во-вторых, этот квантор всегда стоит впереди всех остальных кванторов, если таковые имеются.

В том случае, когда входящие в простой вопрос *производные термины* заменяются их определениями, вопрос может превратиться в сложный. В этом случае процесс поиска ответа усложняется. Рассмотрим, например, вопрос из геометрии, относящийся к данным на чертеже прямым A и B («Верно ли, что A и B параллельны?»):

1. В языке Q получаем (?) [параллельны $\in r(A, B)$].

2. Допустим, имеется следующее определение параллельности: $\forall x_1 \forall x_2$ {паралл. $\in r(x_1, x_2) \equiv \exists x_3$ [плоскость $\in p(x_3)$ & принадлежит $\in r(x_1, x_3)$ & принад. $\in r(x_2, x_3)$] & $\sim \exists x_4$ [точка $\in p(x_4)$ & принад. $\in r(x_4, x_1)$ & прин. $\in r(x_4, x_2)$]]}.

3. С помощью правила подстановки строится частный случай определения: {паралл. $\in r(A, B) \equiv \exists x_3$ [плоскость $\in p(x_3)$ & прин. $\in r(A, x_3)$ & прин. $\in r(B, x_3)$] & $\sim \exists x_4$ [точка $\in p(x_4)$ & прин. $\in r(x_4, A)$ & прин. $\in r(x_4, B)$]]}.

4. В формулировке вопроса производный термин заменяется его определением.

Одновременно с анализом терминов начинается *проверка корректности* вопроса. Значение термина может оказаться неизвестным, т. е. он не принадлежит к базовым и не был определен как производный. В таком случае система должна указать этот термин. В частности, это может быть сделано в виде «встречного» вопроса (?x) [$x \in p(a)$], где a — неизвестный термин. В результате проверяются конъюнктивные члены формулы-предпосылки, имеющие вид $\exists x[x \in p(a)]$. На следующем этапе проверяются остальные члены предпосылки, если они имеются. Например, в случае вопроса «Почему железо легче воды?», предпосылка которого имеет вид $\exists x[x \in p$ (железо) & $\exists x[x \in p$ (вода)] & $\exists x[x \in p$ (легче)] & легче $\in r$(железо, вода), система должна выдать ответ: ВОПРОС НЕКОРРЕКТЕН: $\sim$ [легче $\in r$ (железо, вода)].

Если *в результате замены производного термина простой вопрос превратился в сложный,* он сводится к определенной последовательности формул. Рассмотрим вопрос *I типа* «Верно ли, что A и B параллельны?»

1. Данный вопрос разлагается на следующие: а) (?) $\exists x_3$ [плоскость $\in p(x_3)$ & прин. $\in r(A, x_3)$ & прин. $\in r(B, x_3)$]; б) (?) $\sim \exists x_4$ [точка $\in p(x_4)$ & прин. $\in r(x_4, A)$ & прин. $\in r(x_4, B)$].

2. Для каждого из полученных вопросов строятся формулы ОП.

3. Берется список отрицательных допущений: а) $\sim \exists x_3$ [плоскость $\in p(x_3)$ & ...]; б) $\sim$ {$\sim \exists x_4$ [точка $\in p(x_4)$ & ...]}.

4. В случае *а* ответ будет утвердительным, если найден контрпример, и отрицательным, если найдено подтверждение. В случае *б* имеет место обратное соотношение, причем сколемизируется формула, взятая без внешнего знака отрицания. Если в полученном списке ответов имеются только утверждения, то окончательный ответ также является утвердительным. В противном случае окончательный ответ является отрицательным. В принципе можно потребовать, чтобы система указала, где именно был получен промежуточный отрицательный ответ, т. е. «объяснила» причину окончательного отрицания.

Рассмотрим вопрос *II типа* — «Что параллельно A?» (?x) [параллельно $\in r(A,\ x)$].

1. Строится ОП II типа — $\exists x'$ [параллельно $\in r(A,\ x')$].
2. Ставится вопрос I типа — (?) $\exists x'$ [параллельно $\in r(A,\ x')$].
3. Аналогично уже рассмотренному примеру берется определение производного термина «параллельно».
4. Строится частный случай определения.
5. В результате замены термина определением вопрос принимает вид (?) $\exists x'$ {$\exists x_3$ [плоскость $\in p(x_3)$ & ...] & $\sim \exists x_4$ [точка $\in p(x_4)$ & ...]}.
6. Ставится вопрос к каждой подформуле, начинающейся с квантора существования, отличного от $\exists x'$.
7. Рассматриваются вопросы к подформулам, перед которыми квантор существования стоит без отрицания. При этом из рассмотрения исключаются ЭП, содержащие отмеченную переменную. В данном примере на этом шаге берется вопрос (?) $\exists x_3$ [плоскость $\in p(x_3)$ & принад. $\in r(A,\ x_3)$].
8. Для каждого полученного на шаге 7 вопроса строится формула ОП и берется отрицательное допущение. Если найдено подтверждение, система указывает на некорректность вопроса. Так, в рассматриваемом примере на геометрическом чертеже может быть не указана соответствующая плоскость, что сообщается в ответе «ВОПРОС НЕКОРРЕКТЕН»: $\sim \exists x_3$ [плоскость $\in p(x_3)$ & прин. $\in r(A,\ x_3)$]. Если найден контрпример, производится подстановка в исходную формулировку вопроса.
9. Пусть в данном примере на чертеже указано, что прямая A лежит в плоскости M. В результате подстановки получаем вопрос (?) $\exists x'$ {[плоскость $\in p(M)$ & прин. $\in r(A,\ M)$ & прин. $\in r(x',\ M)$ & $\sim \exists x_4$ [точка $\in p(x_4)$ & прин. $\in r(x_4,\ A)$ & прин. $\in r(x_4,\ B)$]]}.
10. Далее рассматривается подформула, стоящая под квантором существования с отрицанием, причем исключаются ЭП, содержащие отмеченную переменную. В данном примере на этом шаге рассматривается выражение $\sim \exists x_4$ [точка $\in p(x_4)$ & принад. $\in r(x_4,\ A)$]. Найденные контрпримеры ограничивают область значения отмеченной переменной (множество данных на чертеже прямых), исключая из рассмотрения все линии, имеющие точки пересечения с A.

11. Ставится вопрос к ЭП, исключенным на шаге 7, причем учитывается подстановка, произведенная на шаге 9: (?) $\exists x'$ [принадлежит $\in r(x', M)$], где областью значения отмеченной переменной является множество, выделенное на шаге 10.

12. Строится формула ОП.

13. Берется отрицательное допущение.

14. Ответом на исходный вопрос является найденный контрпример (контрпримеры). Если было найдено подтверждение, например на шаге 10 были исключены все изображенные на чертеже прямые, ответ имеет вид $\sim \exists x'$ [параллельно $\in r(A, x')$], т. е. берется отрицание формулы ОП, построенной на шаге 1.

Для приведения более содержательного примера работы алгоритма необходимо расширить язык Q на какую-либо конкретную область, что означает: а) введение базовых терминов, которые будут играть роль индивидных констант; б) возможно, также введение дополнительных функций; в) соответственно п. *а* и *б* должна быть расширена система аксиом (АФ); г) использование производных терминов предполагает введение системы определений.

В результате расширения можно рассмотреть также такие задачи, как поиск ответа, неявно содержащегося в базовом знании, или возможности формализации фрагмента естественного языка, описывающего некоторую предметную область.

Список литературы: 1. *Попов Э. В., Фирдман Г. Р.* Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. М., Наука, 1976. 456 с. 2. *Клыков Ю. И.* Ситуационное управление большими системами. М., Энергия, 1974. 136 с. 3. *Н. Нильсон.* Искусственный интеллект. М., Мир, 1973. 270 с. 4. *Дж. Слэйгл.* Искусственный интеллект. М., Мир, 1973. 320 с. 5. *К. Грин.* Доказательство теорем с использованием правила резолюции как основа для построения вопросно-ответной системы. Приложение к [5], с. 275—309. 6. *M. Bell.* Questioning. — The Philosophical Quarterly. (Indian), 1975, v. 25, № 100, p. 193-211. 7. *Клини С. К.* Математическая логика. М., Мир, 1973. 480 с. 8. *Минц Г. Е.* Функциональная форма. Теорема Эрбрана для непредваренных формул. Приложение 2 к [8], с. 448—450.

УДК 681.327.12 0888

Ю. А. ВАСИЛЕНКО, В. И. РОБОТИШИН, Г. Я. ШЕВЧЕНКО

АЛГОРИТМ ОБРАБОТКИ ДИСКРЕТНЫХ НАБОРОВ ДЛЯ ОБУЧАЮЩИХ ВЫБОРОК БОЛЬШОГО ОБЪЕМА

Алгоритм распознавания дискретных наборов, основанный на представлении обучающей выборки (ОВ) в виде графической структуры — распознающего дерева (РД), имеет ряд преимуществ перед алгоритмами, связанными с РД [1; 2].

Основные преимущества алгоритма* построения РД. 1. Алгоритм построения РД не требует одновременного ввода в ЭВМ

* Близок к алгоритму в работе [5].

всех данных ОВ: возможен ввод данных повекторно и даже отдельных значений признаков при условии, что после значения признака P_i вводится значение признака P_{i+1} (значения признаков P_i, P_{i+1} принадлежит одному и тому же набору). После ввода поступающий вектор используется для построения РД и в дальнейшем в памяти ЭВМ не хранится. Такой ввод информации в ЭВМ и специфика алгоритма значительно экономят память машины и позволяют решать задачи распознавания образов (РО) и ОВ большой мощности.

2. При решении задач РО приведенным алгоритмом число ячеек памяти ЭВМ, используемых для хранения РД, не превышает 2^{n+1}, где n — количество признаков в наборах ОВ.

3. Алгоритм обеспечивает безошибочную классификацию ОВ.

4. Работа алгоритма не зависит от количества образов ОВ. В частности, алгоритм можно применять в случае наличия в ОВ только одного образа. Отсутствие такой возможности в некоторых случаях является отрицательным фактом [3; 4].

5. Алгоритм включает в себя простой подалгоритм дообучения и устранения обнаруженных ошибок при распознавании.

6. Число ошибок, допускаемое РД на фиксированной экзаменационной выборке (ЭВ), не возрастает, если увеличивается объем ОВ. Это требование позволяет определить сходимость алгоритма [4].

Описание алгоритма. Пусть ОВ задана в виде матрицы

$$\begin{array}{l} x_{11}, \ldots, x_{1n}; \\ \vdots \\ x_{m1}, \ldots, x_{mn}; \\ x_{m+1,1}, \ldots, x_{m+1,n}; \\ \vdots \\ x_{k1}, \ldots, x_{kn}; \end{array} \qquad (1)$$

где в общем случае $x_{ij} \in \{0,1, \ldots, k_i-1\}$, $\{x_{ij}\}$ — значение j-го признака i-го набора ОВ, $m, n, k \in R$ $(i = 1, \ldots, k;\ j = 1, \ldots, n)$, m строк ОВ характеризует класс A_1, а остальные — A_2.

Для упрощения изложения считаем, что $x_{ij} \in \{0, 1\}$. По формуле (1) строим РД. В первую вершину РД ставим признак P_1, и от него строим полный путь в РД, соответствующий набору $x_{11}, \ldots, x_{1n}$, т. е. из вершины с признаком P_j выходит стрелка, которая входит в вершину с признаком P_{j+1} (номер стрелки зависит от значения, которое принимает x_{1j}) у ребра, выходящего из конечной вершины, ставим значение $f_R(x_{11}, \ldots, x_{1n})$*.

В следующем наборе $\tau_2 = \langle x_{21}, \ldots, x_{2n} \rangle$ при $\tau_1 \neq \tau_2$ РД изменяем так. При $x_{2j} \neq x_{1j}$ из вершины с признаком x_{1j} проводим вторую стрелку, которая соответствует значению x_{2j} (для удобства

---

* $f_R(x_{i1}, \ldots, x_{in}) = m \leftrightarrow x_{i1}, \ldots, x_{in}$ из образа с номером m $(m \in \{0, 1\})$.

стрелки с номером 0 располагают с левой стороны, а с 1 — с правой), и в конце стрелки ставим вершину с признаком P_{j+1}. Из этой вершины проводим стрелку, соответствующую значению признака $x_{2,j+1}$ и т. д. В конечную вершину ставим $f_R(\tau_2)$ (рис. 1).

Аналогично достраиваем РД для всех других наборов. Если на одном пути РД встречаются два или несколько одинаковых наборов из разных классов, то количество значений функций $f_R(\tau\Delta)$ $(\Delta \in \{0, 1, \ldots, n\})$, соответствующее этим наборам, фиксируем в конечной вершине и окончательно записываем то значение функции $f_R(\tau_\Delta)$, для которого число наборов будет максимальным.

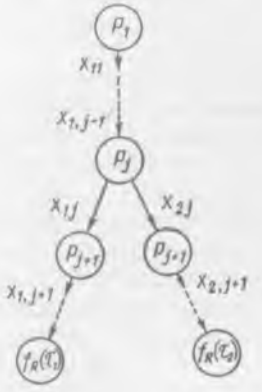


Рис. 1.

Если число различных значений функции $f_R(\tau_\gamma)$ $(\gamma \in \{0, \ldots, n\})$ одинаково в некоторой конечной вершине, то преимущество отдаем образу, мощность которого меньше. После этого построение РД по ОВ считаем законченным.

Задача распознавания состоит в том, чтобы РД классифицировало наборы из ЭВ, не входящие в ОВ (наборы из ОВ РД распознает без ошибок).

После построения РД по данному алгоритму не из всех вершин выходят две стрелки. Если из вершины выходит одна стрелка, то ее вместе со стрелкой удаляем из РД. Минимизированное таким образом РД обозначим через $\overline{\text{РД}}$.

Отметим следующие свойства РД: 1) Сложность* $\overline{\text{РД}}$ меньше, чем сложность РД (при минимизации РД происходит отсев наименее важной информации); 2) $\overline{\text{РД}}$, так же как и РД, безошибочно классифицирует объекты из ОВ и производит экстраполяцию наборов из ЭВ, которые не встречались в ОВ.

Каждому набору Δ из ЭВ соответствует $f_R(\Delta)$ конечной вершины на РД, которое определяет принадлежность данного набора к тому или иному классу.

Различные алгоритмы дают разные результаты на данной ОВ: одни и те же объекты они относят к разным образам. Причиной этого является то, что задачи РО не имеют единственного решения и каждый из алгоритмов дает свое решение. Задача о существовании единственного решения рассматривалась, например, в работе [5].

---

* Под сложностью понимаем количество вершин в РД ($\overline{\text{РД}}$).

Рассмотрим количество решений задачи РО на простом примере. В ОВ, представленной в табл. 1, задано 5 наборов. Значение $f_R(P_1, P_2, P_3)$ указывает на принадлежность наборов из ОВ к тому или иному классу. Требуется найти функцию $\tilde{f}_R \times \times (P_1, P_2, P_3)$, аппроксимирующую $f_R(P_1, P_2, P_3)$. Очевидно, что существует 8 таких различных функций. Если $P_j \in \{0, 1\}$, $j = 1, 2, \ldots, n$ (n — количество признаков; m — количество наборов ОВ), то количество различных решений относительно не входящих в ОВ наборов равно $L = 2^{2^n - m}$. Если $P_j \in \{0, 1, \ldots, k-1\}$, то $L = K^{k^n - m}$.

Таблица 1

Номер набора	Признак			
	P_1	P_2	P_3	$f_R(P_1, P_2, P_3)$
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	?
7	1	1	0	?
8	1	1	1	?

Заметим, что если задано M различных наборов в ОВ, то количество ошибок не может быть более, чем $2^n - (M - 1)$. Таким образом, по информации только об ОВ нельзя сделать вывод о том, какой алгоритм дает лучший результат для данной задачи распознавания. В связи с этим невозможно классифицировать алгоритм РО по «узким областям применимости» [3]. Учитывая изложенное, каждый алгоритм должен включать в себя алгоритм, который устраняет обнаруженные ошибки, алгоритм должен быть гибким.

Приводим простой и мало изменяющий РД алгоритм дообучения и исправления ошибок. Пусть имеем некоторое $\overline{\text{РД}}$, построенное по ОВ из наборов по n признаков. Обнаружив, что на наборе $\tau_i = \langle x_{i1}, \ldots, x_{in} \rangle$* происходит ошибка, РД измеряем следующим образом. Выписываем признаки, которых нет в РД на пути $\langle x_{i1}, \ldots, x_{in} \rangle$, но они присутствуют в поступившем наборе. В конечную вершину, находящуюся на пути $\langle x_{i1}, \ldots, x_{in} \rangle$, ставим один из этих признаков и достраиваем РД так, чтобы ветвь из этих признаков соответствовала пути $\langle x_{i1}, \ldots, x_{in} \rangle$, в конечную вершину которого ставим номер класса, соответствующий набору $\langle x_{i1}, \ldots, x_{in} \rangle$.

Для вершин РД с одной стрелкой на пути $\langle x_{i1}, \ldots, x_{in} \rangle$ добавляем недостающие стрелки, в конце их записываем то значение $f_R(\tau)$, которое было в конечной вершине до обнаружения ошибки.

При таком исправлении ошибок количество вершин РД несколько увеличивается, но наборы из ОВ распознаются безошибочно, а на наборе $(x_{i1}, \ldots, x_{in})$ ошибок не происходит.

---

* Его может не быть в ОВ.

Рассмотрим пример применения описанного алгоритма. ОВ задана табл. 2, а ЭВ — в табл. 3. Построим по ОВ РД и проверим его работу (включая подалгоритм дообучения и исправления ошибок) на ЭВ.

Таблица 2

№	P_1	P_2	P_3	$f_R(P_1, P_2, P_3)$
1	0	0	0	1
2	0	0	1	0
3	0	1	1	0
4	1	0	1	1
5	1	1	0	0

Таблица 3

№	P_1	P_2	P_3	$f_R(P_1, P_2, P_3)$
1	0	0	1	0
2	0	1	1	0
3	0	1	0	1
4	0	1	0	1
5	1	0	0	1
6	1	1	1	0
7	1	0	0	1
8	1	1	1	0

Этапы построения РД по ОВ показаны на рис. 2, *а — д*. Минимизированное РД имеет вид, показанный на рис. 2, *е*.

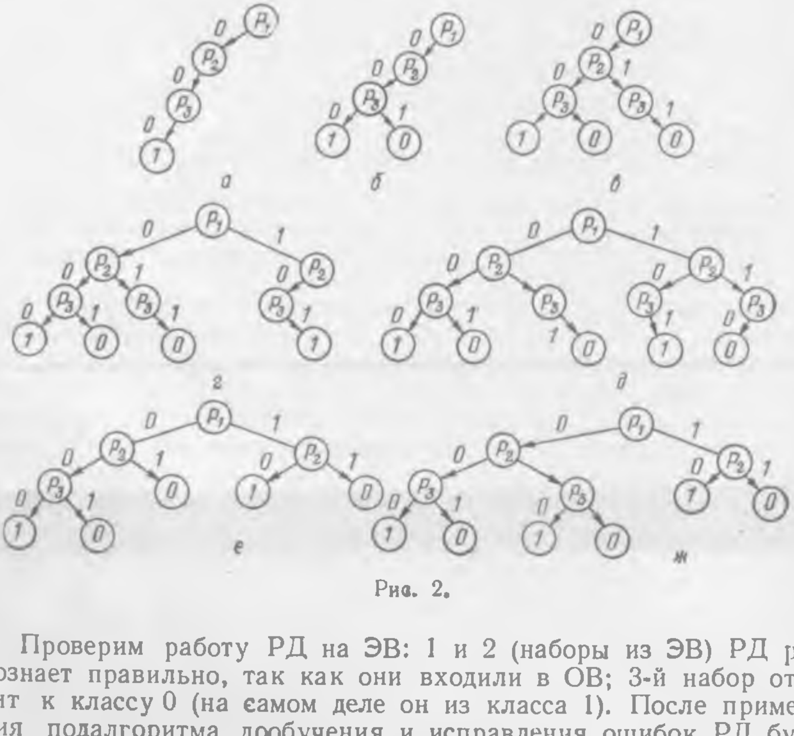


Рис. 2.

Проверим работу РД на ЭВ: 1 и 2 (наборы из ЭВ) РД распознает правильно, так как они входили в ОВ; 3-й набор относит к классу 0 (на самом деле он из класса 1). После применения подалгоритма дообучения и исправления ошибок РД будет иметь вид, показанный на рис. 2, *ж*, обозначим его через РД

4-й набор совпадает с 3-м, и РД отнесет его к классу 1. Таким образом, произошло дообучение на 3-м наборе из ЭВ. Остальные наборы ЭВ РД распознает без ошибок.

Результаты применения описанного выше алгоритма и сравнение его эффективности с другими алгоритмами на 3-х реальных задачах по геологии, геохимии и геофизики [3] отражены в табл. 4. Для каждого из 14 алгоритмов (по каждой из 3-х задач) приведено количество правильных ответов (в %), полученных на ЭВ.

Таблица 4

Алгоритм	Задача		
	1	2	3
Линейная решающая функция «Одуванчик»	64	64	55
Нелинейная решающая функция «Эдельвейс»	76	68	55
Кора-3	32	68	65
Тест-2	72	64	70
Энтропия-1	72	64	70
Энтропия-2	56	56	25
Энтропия-3	56	76	15
Голотип	44	52	50
Потенциальная функция	44	60	50
Геолог-1	72	36	55
Юг	76	68	55
Алгоритм построения РД с пошаговой оценкой важности	56	68	—
Полное РД	72	70	—
Вышеописанный алгоритм	72	80	65

Список литературы: 1. *Пряницкий А. М., Василенко Ю. А.* Об одном алгоритме обучения распознаванию дискретных сигналов.— Вопросы радиоэлектроники. Серия АСУ, 1975, вып. 6, с. 53—64. 2. Последовательные методы принятия решений при помощи графов. Матер. IV симпозиума по кибернетике, ч. 1. Тбилиси, 1972. 3. Геология и математика. Новосибирск, Наука, 1970. 4. Геология и математика. Новосибирск, Наука, 1966. 5. *Хабаров В. В., Косарев Ю. Г.* Об эффективности автоматического кодирования и исправления ошибок при подготовке данных.— Вычислительные системы. Новосибирск, 1975, вып. 62, с. 25—28.

УДК 681.327.12 0888

Ю. А. ВАСИЛЕНКО, А. М. ПРЯНИЦКИЙ, Г. Я. ШЕВЧЕНКО

ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ РАСПОЗНАЮЩИХ ДЕРЕВЬЕВ С УЧЕТОМ ВАЖНОСТИ ПРИЗНАКОВ

1. Постановка задачи распознавания образов. Пусть имеется разбиение R множества G и некоторая распознающая система S (человек, ЭВМ или произвольный автомат). Задача распознавания — систему S «научить» вычислять функцию $f_R(x)$, где $f_R(x)$ — функция, которая зависит от признаков $x_1, x_2, \ldots, x_n$, входящих в обучающую выборку (ОВ) и принимающих на этой ОВ некоторые значения. Обучить систему S распознаванию образов разбиения R можно двумя путями: первый — функция f_R заранее задается каким-то образом системе S, второй — система S последовательно «обучается» распознаванию образов разбиения R. Во втором случае процесс «обучения» удобно представлять с помощью графической конструкции типа дерева. Назовем такую конструкцию распознающим деревом (РД). Более подробно этот вопрос рассмотрен в работе [1].

Для эффективного распознавания объектов, заданных в виде дискретных наборов, необходимо выявить наиболее важные (в смысле информативности) признаки в ОВ.

Оценка важности признака $W(P_i)$ в общем случае по отношению к функции $f_R(x)$, задающей разбиение ОВ, определяется так:

$$W(P_i) = \sum_{j=0}^{k_i-1} (b_j/h)\, \rho_j; \quad \rho_j = \max_{0 < m < k-1} \frac{q_j^m}{b_j}, \tag{1}$$

где P_i — i-й признак, $P_i \in \{0, 1, \ldots, k_i - 1\}$; b_j $(j = 0, 1, \ldots, k_i - 1)$ — количество наборов ОВ, в которых $P_i = j$; q_j^m — количество наборов ОВ, принадлежащих классу U_m $(U_m \in G)$ $(m = 0$ $1, \ldots, k - 1)$ при $P_i = j$; h — количество всех наборов ОВ.

В работе [2] предложен алгоритм распознавания объектов заданных в виде дискретных наборов, причем при построении РД по ОВ на каждом шаге выбора новой ветви РД оценивается важность признаков по формуле (1) и на каждом шаге выбирается признак, имеющий максимальное значение W. Представляет интерес создание на основе этого алгоритма специализированного устройства (СУ) для распознавания дискретных наборов, работающего в реальном масштабе времени.

В данной работе предлагается структурная схема и алгоритм функционирования такого СУ, реализующего алгоритм построения РД [2] непосредственно по ОВ, которая имеет допустимый* вид дискретных наборов и классификацию поступающих наборов признаков $x_1, \ldots, x_n$ при экзамене (эксплуатации) в соответствии с построенным на этапе обучения решающим правилом, которым в данном случае является РД. Чтобы упростить изложение, рассмотрим бинарный случай, когда каждый признак x_i $(i = 1, 2, \ldots, n)$ и функция разбиения $f_R(x)$ могут принимать лишь два значения: 0 или 1.

2. Структурная схема СУ. Обрабатывая ОВ по алгоритму, приведенному в работе [2], на каждом шаге (под шагом алгоритма понимается вычисление некоторой вершины РД) построения РД вычисляются функционалы $W(x_i)$ $(i \in \{1, \ldots, n\})$ и выбирается $W(x_l) = \max(W(x_1), \ldots, W(x_n))$ $(1 \leqslant l \leqslant n)$, соответствующий признак x_l помещается в очередную неконечную вершину РД. Она считается неконечной, так как относительно ее необходимо производить дальнейшие шаги до тех пор, пока не получим одно из значений функции $f_R(x)$. Тогда вычисленная вершина считается конечной и в нее заносится найденное значение функции $f_R(x)$. Таким образом, обрабатывая ОВ, мы на каждом шаге

* $H_i \cap H_j = \varnothing$ $(i \neq j)$, $H_i \cup H_j \in G$.

вычисляем вершину с одним из признаков $x_i \, (i \in \{1, 2, \ldots, n\})$ либо вершину с некоторым значением функции $f_R(x)$. После обучения получаем некоторое РД, в неконечных вершинах которого находятся признаки $x_i \, (i \in \{1, 2, \ldots, n\})$, а в конечных — значения функции $f_R(x)$.

СУ должно включать в себя устройства, моделирующие ребро РД и вершину РД. Назовем их соответственно моделью ребра (МР) и моделью вершины (МВ). С помощью числа МР и МВ, зависящего от сложности РД, можно промоделировать любой путь в данном РД.

Очевидно, СУ должно работать в двух режимах — обучения и классификации (экзамена) — и включать в себя следующие основные блоки: запоминающее устройство (ЗУ1) для хранения ОВ; арифметическое устройство (АУ) для вычисления функционалов W_i и выбора в каждом шаге $\max(W_1, \ldots, W_i) \, (i \in \{1, \ldots, n\})$ или вычисления значения функции $f_R(x)$; число МР и МВ (такое, чтобы можно было промоделировать любой путь в данном РД); запоминающие устройства ЗУ2 для хранения в определенном порядке вычисленных вершин РД и ЗУ3 для хранения адресов МВ и МР, с помощью которых происходит дальнейшее построение РД; блок проверки БП окончания обучения (построения) РД; устройство управления (УУ), которое по информации из АУ, ЗУ1, ЗУ2, ЗУ3 и БП управляло бы ходом построения РД; вспомогательные устройства (кодирующее устройство, устройство приведения ОВ к допустимому виду, устройство ввода — вывода информации и т. д.). В режиме классификации (экзамена) добавляют: устройства обработки входной информации и индикации результатов классификации (экзамена).

В процессе построения РД МВ, соответствующая вычисленной на некотором шаге неконечной вершине, настраивается на пропуск всех значений признака $x_i \, (i \in \{1, 2, \ldots, n\})$, признанного наиболее важным на данном шаге, а для продолжения обучения в данном направлении соответствующая МР настраивается последовательно на пропуск j-го значения $(j \in \{0, 1\})$ признака x_i.

МВ, соответствующие пути, по которому происходит обучение в данный момент, настроены на пропуск всех значений тех признаков, которые были вычислены в предыдущие моменты времени (шаги алгоритма), а МР, осуществляющие связь между МВ данного пути, настраиваются на пропуск одного из $j \, (j \in \{0, 1\})$ значений (этих же признаков) в зависимости от пути, по которому происходит обучение РД в данный момент.

Исходя из изложенного, можно предложить следующую структурную схему СУ для распознавания объектов, заданных в виде дискретных наборов (рис. 1, где

1 — ЗУ1 для хранения ОВ; *2* — блоки МВ и МР; *3* — АУ; *4* — ЗУ2 и ЗУ3; *5* — счетчик для подсчета количества вычисленных неконечных вершин (СЧ1); *6* — счетчик для подсчета считываемых из ЗУ2 неконечных вершин (СЧ2); *7* — схема сравнения (СхС); *8* — устройство управления (УУ); КУ — кодирующее устройство; S_1 — сигнал «продолжить построение РД»; S_2 — сигнал «окончить построение РД»; $\Rightarrow$ — информационный сигнал; $\rightarrow$ — управляющий сигнал).

3. Алгоритм функционирования СУ в режиме обучения. Для более полного представления о работе СУ кроме структурной схемы необходимо иметь формализованное описание совокупности предписаний, ведущих к правильному выполнению процесса

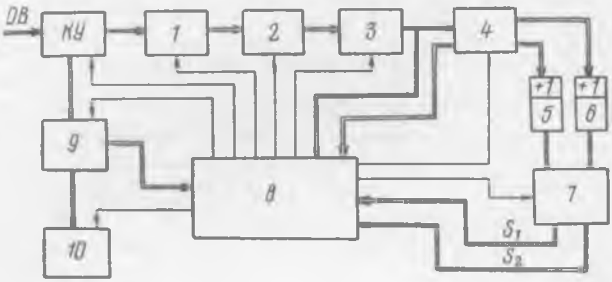


Рис. 1.

построения РД. Такой совокупностью предписаний является алгоритм функционирования (АФ) СУ, который запишем в виде операторной схемы

$$B \underset{8}{\downarrow} A_1A_{11}P_9 \overset{5}{\uparrow} P_1 \overset{3}{\uparrow} \underset{5}{\downarrow} A_2A_3A_4A_5P_{11} \overset{6}{\uparrow} P_{10} \overset{1}{\uparrow} \underset{6}{\downarrow} A_{10} \underset{1}{\downarrow} A_6A_0$$

$$P_6 \overset{9}{\uparrow} A_{12}P_2 \overset{2}{\uparrow} A_8 \underset{2}{\downarrow} P_8 \overset{4}{\uparrow} A_9 \underset{9.4}{\downarrow} A_{13}P_3 \overset{3}{\uparrow} A_7P_4 \overset{1}{\uparrow} \underset{3}{\downarrow} A_{14}P_5 \overset{7}{\uparrow} P_7 \overset{8}{\uparrow} \underset{7}{\downarrow} S,$$

где функциональными операторами являются следующие: B — первоначальный пуск, считывание информации из ЗУ1, вычисление первой неконечной вершины РД (считаем, что первая вычисленная вершина всегда неконечная), занесение $+1$ в СЧ1, индекса признака в ЗУ2, адреса МВ$_1$ в ЗУ3; A_0 — оператор записи в очередную свободную ячейку ЗУ2 очередной вычисленной вершины (конечной или неконечной); A_1 — оператор считывания очередной ячейки ЗУ2; A_2 — оператор считывания очередной ячейки ЗУ3; A_3 — оператор считывания и дешифрации адреса пути, записанного в ячейке ЗУ3; A_4 — оператор настройки на пропуск значений признака x_s, записанного в считываемой ячейке ЗУ2, в МВ$_s$ $(1 \leqslant s \leqslant n)$; A_5 — оператор настройки на пропуск нулевых значений признака x_s в МР$_s$; A_6 — оператор считывания информации из ЗУ1 и вычисления очередной вершины РД; A_7 — оператор настройки на пропуск единичных значений признака x_s в МР$_s$; A_8 — оператор формирования и записи в очередную свободную ячейку ЗУ3 адреса МВ, соответствующей неконечной вершине РД, вычисленной относительно нулевого значения последнего считанного из ЗУ2 признака; A_9 — оператор формирования и записи в очередную свободную ячейку ЗУ3 адреса МВ, соответствующей неконечной вершине РД, вычисленной относительно единичного значения последнего считанного из ЗУ2 признака; A_{10} — оператор настройки МР, соответствующих проходимому в данный момент обучения пути в состояния, соответствующие данному пути; A_{11} — оператор записи $+1$ в СЧ1; A_{12} — оператор записи $+1$ в СЧ2; A_{13} — оператор записи $+1$ в СЧ3 (счетчик 3 предназначен для вычисления числа состояний МР, соответствующей той неконечной вершине РД, относительно которой идет дальнейшее построение РД в данный момент); A_{14} — оператор разрешения сравнения содержимого СЧ1 и СЧ2.

В качестве логических используются следующие операторы:

$$P_1 = \begin{cases} 1, & \text{если в считываемой ячейке ЗУ2 находится признак;} \\ 0, & \text{если в считываемой ячейке ЗУ2 находится значение функции } f_R(x); \end{cases}$$

$$P_2 = \begin{cases} 1, & \text{если в СЧ3 записан 0;} \\ 0, & \text{если в СЧ3 записана 1;} \end{cases} \qquad P_3 = \begin{cases} 1, & \text{если в СЧ3 записана 1;} \\ 0, & \text{если в СЧ3 записана 2;} \end{cases}$$

$$P_4 = \begin{cases} 1, & \text{если МР, соответствующая той неконечной вершине РД, относительно которой идет построение РД в данный момент, настроена на пропуск нулевых значений признака } x_l; \\ 0, & \text{если МР настроена на пропуск единичных значений признака } x_l \ (1 < t < n); \end{cases}$$

$$P_5 = \begin{cases} 1, & \text{если } b_{СЧ1} \neq b_{СЧ2} \ (b_{СЧ1} \text{ — содержимое СЧ1; } b_{СЧ2} \text{ — содержимое СЧ2);} \\ 0, & \text{если } b_{СЧ1} = b_{СЧ2}; \end{cases}$$

$$P_6 = \begin{cases} 1, & \text{если вычислена неконечная вершина;} \\ 0, & \text{если вычислена конечная вершина;} \end{cases}$$

$$P_7 = \begin{cases} 1, & \text{если оператор } P_5 \text{ в состоянии 0;} \\ 0, & \text{если оператор в состоянии 1;} \end{cases}$$

$$P_8 = \begin{cases} 1, & \text{если оператор } P_2 \text{ в состоянии 0;} \\ 0, & \text{если оператор в состоянии 1;} \end{cases}$$

$$P_9,\ P_{10} = \begin{cases} 1, & \text{если хотя бы один раз выполнялся оператор } A_{14}; \\ 0, & \text{если ни разу не выполнялся оператор } A_{14}; \end{cases}$$

$$P_{11} = \begin{cases} 1, & \text{если ни разу не выполнялся оператор } A_{14}; \\ 0, & \text{если хотя бы один раз выполнялся оператор } A_{14}; \end{cases}$$

Согласно структурной схеме на рис. 1 и АФ, записанному в виде операторной схемы, СУ в режиме обучения работает следующим образом.

После сигнала «пуск» ОВ считывается из ЗУ1, поступает в АУ, где вычисляется неконечная вершина, и соответствующий индекс признака $x_k (1 \leqslant k \leqslant n)$ заносится в первую ячейку ЗУ2. В первую ячейку ЗУ3 заносится адрес МВ, в СЧ1 — $+1$. После этого из первой ячейки ЗУ2 считывается записанный там индекс признака x_k, по адресу в первой ячейке ЗУ3 отыскивается МВ, которая настраивается на пропуск всех значений признака $x_k (k \in \{1, 2, \ldots, n\})$, $МР_1$ настраивается на пропуск j-го $(j \in \{0, 1\})$ значения признака x_k, вычисленная вершина (конечная или неконечная) заносится в очередную свободную ячейку ЗУ2. Если в ЗУ2 заносится индекс признака (т. е. неконечная вершина), то в соответствующую ячейку ЗУ3 заносится адрес МВ предыдущих неконечных вершин, связанных с вычисленной на данном этапе (шаге) неконечной вершиной.

В СЧ3 заносится $+1$, $P_3 = 1$ и $МВ_1$ настраивается на пропуск единичных значений признака x_k. Вновь вычисляется вершина РД (конечная или неконечная), соответствующая информация заносится в ЗУ2 и, если необходимо, в ЗУ3.

После вычисления очередных двух вершин следующего яруса РД относительно 0 и 1-го значений одной из неконечных вершин предыдущего яруса в СЧ3 будет записано число $+2$, что является условием для проверки окончания обучения РД.

При $b_{\text{СЧ1}} = b_{\text{СЧ2}}$ в момент проверки схема сравнения выдает сигнал на прекращение обучения, при $b_{\text{СЧ1}} \neq b_{\text{СЧ2}}$ процесс обучения продолжается. Из ЗУ2 считывается очередная заполненная ячейка. Если в ней записан признак, относительно каждого из его j значений ($j \in \{0, 1\}$) проводятся шаги алгоритма, а соответствующие результаты заносятся в СЧ1, СЧ2, СЧ3, в ЗУ2 и, если необходимо, в ЗУ3.

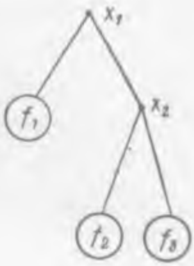


Рис. 2.

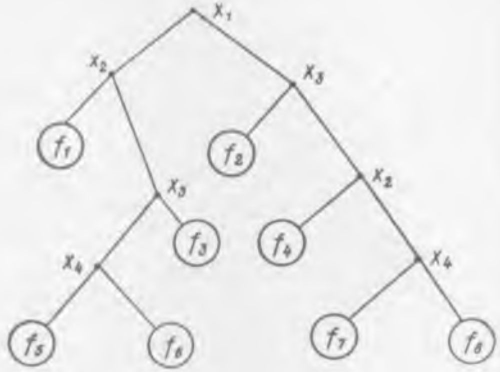


Рис. 3.

Если же в считываемой ячейке ЗУ2 находится какое-либо значение функции $f_R(x)$, то ее пропускают, считывая следующую ячейку ЗУ2 и т. д. до тех пор, пока не будет считана ячейка с признаком.

Процесс обучения происходит до тех пор, пока устройство сравнения не выдаст сигнал о прекращении. Такой сигнал обязательно поступит, так как доказана [см. 2] сходимость за конечное число шагов алгоритма построения РД.

П р и м е р. Пусть в результате обучения получим РД (рис. 2). *При этом выполнятся следующие* операторы: $BA_1A_{11}P_9A_2A_3A_4A_5$ $P_{11}P_{10}A_6A_0P_6A_{13}P_3A_7P_4A_6A_0P_6A_{12}P_2P_8P_9A_{14}P_3$ (сброс СЧ3 в 0), $A_{14}(b_{\text{СЧ1}} \neq b_{\text{СЧ2}})$ $P_5P_7A_1A_{14}P_9P_4A_{14}(b_{\text{СЧ1}} \neq b_{\text{СЧ2}})$ $P_5P_7A_1A_{11}P_9P_4A_2$ $A_3A_4A_5P_{11}P_{10}A_{10}A_6A_9P_6A_{13}P_3A_7P_4A_6A_0P_6A_{13}P_3$ (сброс СЧ3 в 0), $A_{14}(b_{\text{СЧ1}} = b_{\text{СЧ2}})$ P_5S. Обучение закончено.

4. Алгоритм функционирования СУ в режиме экзамена. В ЗУ2 после обучения записаны в упорядоченной последовательности индексы $j(P)$ признаков, находящихся в неконечных вершинах и значения функции $f_R(x)$, находящиеся в конечных вершинах. Такая запись эквивалентна записи некоторой перестановки n целых чисел $\sigma_n = (i_1, i_2, \ldots, i_k, \ldots, i_n)$; $i_k \in \{j(P), f_R(x)\}$. В общем случае $j(P) \in \{1, 2, \ldots, n\}$; $f_R(x) \in \{0, 1, \ldots, k-1\}$, но мы рассматриваем бинарный случай, поэтому $f_R(x) \in \{0, 1\}$, где i_k — число, стоящее на k-м месте в перестановке σ_n.

Таким образом, перестановка σ_n (с учетом правила записи в ней вершин РД) эквивалентна некоторому РД, но в РД наглядно видны связи между вершинами, в отличие от σ_n.

Для того, чтобы в σ_n определить номера вершин следующего яруса, связанных с данной неконечной вершиной единичными путями (единичный путь — путь между двумя смежными вершинами), поступаем следующим образом. Вначале находим числа

$$N_k^j = ki - (k - C_k^j) \ (j = 1, 2, \ldots, k); \ C_k^{j+1} = C_k^j + 1, \ C_k^1 = 2, \quad (2)$$

где k — значность РД; i — номер неконечной вершины, относительно которой вычисляем номера смежных с ней вершин, расположенных на следующем ярусе. Определив при заданном k числа $C_k^j (j = 1, 2, \ldots, k)$ и подставив их в выражение (2), получим числа $N_k^j (j = 1, 2, \ldots, k)$. Номера вершин следующего яруса РД, соединенных единичным путем с заданной неконечной вершиной с номером i, находим подстановкой вместо i номера интересующей неконечной вершины РД. Например, для $k = 2$ имеем

$$C_2^1 = 2; \ C_2^2 = 3; \ N_2^1 = 2i; \ N_2^2 = 2i + 1. \quad (3)$$

Подставив в уравнение (3) какой-либо номер i неконечной вершины и вычислив N_2^1, мы определим номер левой, смежной с ней вершины следующего яруса относительно вершины с номером i, а вычислив N_2^2, получим номер правой вершины следующего яруса относительно той же вершины с номером i.

При экзамене на вход СУ поступает набор значений признаков $x_1 \ldots x_n$; обозначим его через $\varphi_n = < \xi_1 \ \xi_2 \ldots \xi_n >$ и будем считать, что φ_n всегда упорядочена, т. е. на первом месте всегда стоит признак x_1, на k-м месте — признак x_k.

Тогда с учетом изложенного при поступлении на вход блока 9 какого-либо экзаменационного набора φ_n СУ реализует алгоритм вычисления значения функции $f_k(x)$, который запишем в виде операторной схемы $A_1 A_2 \underset{3}{\downarrow} A_3 P_1 \overset{1}{\uparrow} A_4 A_5 \underset{4}{\downarrow} A_6 P_3 \overset{4}{\uparrow} A_7 A_8 P_2 \overset{2}{\uparrow} A_9 \omega \overset{3}{\uparrow} \underset{2}{\downarrow} A_{10} \omega \overset{3}{\uparrow} \underset{1}{\downarrow} S$, где функциональные операторы A_i выполняют следующие операции: A_1 — запись в буферную память блока 9 значения входного набора φ_n; A_2 — присвоение величине $k(\sigma_n)$ значения, равного 1 ($k(\sigma_n)$ — порядковый номер i_k в σ_n); A_3 — считывание в σ_n числа, находящегося в ячейке с номером, равным $k(\sigma_n)$; A_4 — определение численного значения $i_k(\sigma_n)$; A_5 — определение порядкового номера $k'(\sigma_n)$ (индекса признака) в σ_n (с учетом лишь индексов признаков); A_6 — определение общего порядкового номера $i_k(\sigma_n)$ в σ_n (считывание начинается одновременно с началом работы оператора A_5 и заканчивается при остановке выполнения оператора A_5); A_7 — определение в φ_n ячейки с номером, равным $i_k(\sigma_n)$; A_8 — считывание из φ_n значения признака из ячейки с номером, равным $i_k(\sigma_n)$; A_9 — вычисление величины $k(\sigma_n) = 2 \times \times k'(\sigma_n) + 1$; A_{10} — вычисление величины $k(\sigma_n) = 2 \times k'(\sigma_n)$.

Логические операторы выполняют следующие операции:

$$P_1 = \begin{cases} 1, \text{ если считанное число из ячейки в } \sigma_n \text{ — индекс признака;} \\ 0, \text{ если считанное число из ячейки в } \sigma_n \text{ — значение функции} \\ f_R(x); \end{cases}$$

$$P_2 = \begin{cases} 1, \text{ если значение считываемого признака в } \varphi_n \text{ равно } 1; \\ 0, \text{ если значение считываемого признака в } \varphi_n \text{ равно } 0; \end{cases}$$

$\omega = \mathrm{const} = 0;$

$$P_3 = \begin{cases} 1, \text{ если } k(\sigma_n) = l(\sigma_n); \\ 0, \quad \text{»} \quad k(\sigma_n) \neq l(\sigma_n); \end{cases}$$

$l(\sigma_n)$ — число, полученное в результате выполнения оператора A_6.

Пример. Пусть в результате обучения получили РД (рис. 3), тогда σ_n запишется так: $\sigma_n = < 123f_1 3f_2 24f_3 f_4 4f_5 f_6 f_7 f_8 >$. Пусть на вход блока *9* на экзамене поступает набор $\varphi_n = (0101)$. Он будет отнесен к значению функции f_6. При вычислении значения функции f_6 выполнятся следующие операторы: $A_1A_2A_3P_1A_4A_5A_6P_3A_7A_8P_2A_{10}\omega A_3P_1A_4A_5A_6P_3A_7A_8P_2A_9\omega A_3P_1A_4A_6A_5P_3A_6A_5P_3A_6A_5P_3A_7A_8P_2A_{10}\omega A_3P_1A_4A_6A_5P_3A_6A_5P_3A_6A_5P_3A_6A_5P_3A_7A_8P_2A_9\omega A_3P_1$(считывается значение функции f_6) S.

Список литературы: 1. *Василенко Ю. А., Пряницкий А. М.* Об оценке информационных признаков при распознавании дискретных наборов. — Управляющие системы и машины, 1972, № 2, с. 38—40. 2. *Пряницкий А. М., Василенко Ю.* А. Общие аспекты обучения распознаванию образов. — Отбор и передача информации. Киев, 1975, вып. 44, с. 14—25.

УДК 007:573.6

А. И. КАЛЮЖНЫЙ

О ПОСТРОЕНИИ АНАЛИТИЧЕСКИХ ОПИСАНИЙ СЛОЖНЫХ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ В ЗАДАЧАХ РАСПОЗНАВАНИЯ ГЕОМЕТРИЧЕСКИХ ОБРАЗОВ

Решение на ЭВМ задач распознавания геометрических образов тесно связано с разработкой методов и алгоритмов аналитических описаний сложных геометрических объектов [12]. Рассмотрим вопросы использования аппарата k-логики для аналитического описания широкого, с прикладной точки зрения, класса геометрических объектов. Предлагается полная система функций, доказывается ее простота. Решением является алгебраическая конструкция в виде произведения отношений [5], которая позволяет строить аналитические представления функций, принимающих заданные отношения на заданных множествах. Теоретико-множественная символика принята согласно [7].

Постановка задачи. Под отношением $f: A^n \to R$ понимаем задание соответствия элементам множества A^n (где A^n обозначает декартову степень) элементов множества R. Произведением отношений $f_1: A \to B$ и $f_2: B \to C$ будем называть такое отношение

$f_3: A \to C$, для которого справедливо $\forall a \in A \to \exists x_i \in B \to c \in C$. Функциями или отображениями будем считать такие отношения $f: A \to B$, в которых каждому $a \in A$ ставится в соответствие единственный элемент $b \in B$. Формулы — все отношения, места в которых замещены символами элементов, переменными, формулами. Совпадение значений двух формул обозначим так:

$$f_i(\alpha_1, \ldots, \alpha_n) = f_j(\alpha_1, \ldots, \alpha_n); \; \alpha_i \in A.$$

Если $\forall x \in A$ справедливо $f_i(x \in A) = f_j(x \in A)$, говорим, что формулы $f_i(A)$ и $f_j(A)$ коэкстенсивны на множестве A, и обозначаем $f_i(A) \equiv f_j(A)$. Систему S непустых подмножеств множества A условимся называть разбиением множества A, если каждый элемент A принадлежит одному и только одному подмножеству из системы S. Подмножества из S называются смежными классами разбиения S. Совокупность всех смежных классов множества A по эквивалентности σ обозначается A/σ и называется фактор-множеством от A по σ. Однозначное отображение $A \to A/\sigma$, при котором каждый элемент $a \in A$ переходит в содержащий его смежный класс $[a]$, называется каноническим отображением A на A/σ.

Пусть на множестве A, где A является подмножеством евклидового пространства R_n при $n = 2, 3$, задано множество сложных геометрических объектов $D = \{D_1, \ldots, D_p\}$, образованных системой опорных элементов $\psi = \{\psi_1, \ldots, \psi_n\}$ и обладающих свойством $D_i \cap D_j = \varnothing$ при $i \neq j$, причем каждый опорный элемент ψ_i образует каноническое отображение A на A/ψ_i, т. е. для каждого ψ_i существуют такие смежные классы $B_1^i, \ldots, B_k^i \subset A$, что $B_j^i \cap B_e^i = \varnothing$ при $j \neq e$ и $\bigcup\limits_{j=1}^{k} B_j^i = A$. В задаче требуется построить отношение $\alpha: A \to A_1$ в виде конечной логико-арифметической или арифметической формулы, удобной для реализации на ЭВМ, с тем свойством, чтобы на заданных множествах D_i полученное решение α было коэкстенсивно априорно заданному отношению f_i, т. е. $\alpha(D_i) \equiv f_i(D_i)$; $i = 1, \ldots, p$. Отношение будем искать в виде $\alpha \equiv \alpha_1\alpha_2\alpha_4\alpha_5$. В связи с выбором и построением неопределенных отношений α_1, α_2, α_3, α_4, α_5 рассмотрим вопросы анализа алгебраических систем, которые необходимы для решения поставленной задачи.

k-Системы. Рассмотрим алгебраическую систему, объекты которой представляют собой множество $E = \{e_1, \ldots, e_m, \ldots\}$, а множество отношений образуют множество $F = \{f_1, \ldots, f_p, \ldots\}$. Множества, образованные отношениями f_i над множеством E, образуют соответственно множества C_i, т. е. $f_i: E^x \to C$; $C = \{c_1, \ldots, c_p, \ldots\}$, x — некоторая декартова степень.

Определение 1. *Отношение f_i называется инвариантным, если $C_i \in E$, а отношение f_i определено $\forall x \in E$. Если все f_i являются инвариантными, то множество отношений F назовем инвариантным.*

Определение 2. *Алгебраическая система, определенная на множестве $E = \{e_1, \ldots, e_m, \ldots\}$ со множеством инвариантных отношений $F = \{f_1, \ldots, f_p, \ldots\}$, называется k-системой.*

Очевидно, что k-системы являются, согласно работе [5], алгебрами с полностью определенными отношениями.

Определение 3. *Если множество $E = \{e_1, \ldots, e_k\}$ конечно, такая система называется k-логикой.*

k-логика, у которой множество $E = \{0, 1, \ldots, k-1\}$, является k-значной логикой.

Очевидно, что k-логика изоморфна k-значной логике. Исследования k-логики в настоящей работе связаны с возможностью решения поставленной задачи более простым (с вычислительной точки зрения) методом, чем при использовании аппарата k-значной логики. Однако аппарат k-логики может быть успешно применен и в других случаях.

Сформулируем для k-логик некоторые результаты. Результаты, полученные в k-значной логике [3; 4] без учета специфики E, справедливы и в k-логике и будут приведены без доказательства. Мощность классов R_n ($n = 0, \ldots, n$) отношений порядка n является фиксированной. Для класса R_n справедливо утверждение

$$\left|R_n\right| = k^{k^n}, \text{ где } k = |E|.$$

Мощность формул k-логики с конечной системой отношений равна χ. В целях дальнейшего анализа k-логик рассмотрим некоторые отношения и их свойства.

k-Отношения. Определим в k-логике некоторые элементарные функции:

1. Все отношения порядка 0 образуют множество функций R_0, обладающие таким свойством:

$$f_i \equiv e_i; \quad e_i \in E; \quad i = 1, \ldots, k.$$

2. Бинарное отношение, обладающее свойством $f(x, y) = \max(x, y)$, является k-значной дизъюнкцией и обозначается $f_v(x, y) \equiv x \vee y$.

3. Бинарное отношение, обладающее свойством $f(x, y) = \min(x, y)$, является k-значной конъюнкцией и обозначается $f_\wedge(x, y) = x \wedge y$.

4. Бинарное отношение, обладающее свойством

$$f(x, y) \equiv \begin{cases} \max(e_1, \ldots, e_k) \text{ при } x = y; \\ \min(e_1, \ldots, e_k) \text{ при } x \neq y, \end{cases}$$

является характеристической функцией и обозначается $I_x(y)$.

Очевидно, что для n-арной и k-значной дизъюнкции в k-логике справедливо

$$\bigvee_{i=1}^{n} (x_i) \equiv \max(x_1, \ldots, x_n).$$

Аналогично для n-арной k-значной конъюнкции справедливо

$$\bigwedge_{i=1}^{n} (x_i) \equiv \min(x_1, \ldots, x_n).$$

Определение 4. *Отношение порядка $k+1$ в k-логике, обладающее свойством*

$$f(x, y_1, \ldots, y_k) = \begin{cases} y_1 & \text{при } x = e_1; \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ y_k & \text{при } x = e_k, \end{cases}$$

называется k-функцией.

Опираясь на свойство k-функции, переменную x будем называть управляющим параметром, а переменные $y_1, y_2, \ldots, y_k$ — значениями функции в точках управляющего параметра и символически обозначать

$$K(x, y_1, \ldots, y_k) \equiv \xrightarrow[y_1 - y_2 - \ldots - y_k]{x}.$$

Теорема 1. *k-Функция к-логики коэкстенсивна полиному Лагранжа $(k-1)$-й степени относительно управляющего параметра при $x, y_1, y_2, \ldots, y_k \in E$:*

$$K(x, y_1, \ldots, y_k) \equiv L_{k-1}(x) \equiv$$

$$\equiv \sum_{i=1}^{n} y_i \frac{(x-e_1)\ \ldots\ (x-e_{i-1})\ (x-e_{i+1})\ \ldots\ (x-e_k)}{(e_i-e_1)\ \ldots\ (e_i-e_{i-1})\ (e_i-e_{i+1})\ \ldots\ (e_i-e_k)}.$$

Доказательство основывается на свойстве полинома Лагранжа принимать заданные значения в заданных точках.

Следствие. *Полином Лагранжа при $x, y_1, \ldots, y_k \in E$ может рассматриваться как отношение порядка $(k+1)$ в k-логике.*

Теорема 2. *k-Функция коэкстенсивна формуле $F_1(x, y_1, \ldots, y_k)$:*

$$K(x, y_1, \ldots, y_k) \equiv F_1(x, y_1, \ldots, y_k) \equiv \bigvee_{i=1}^{k} (I_{e_i}(x) \wedge y_i).$$

Теорема 3. *Характеристическая функция коэкстенсивна формуле $F_2(x, y)$:*

$$I_x(y) \equiv F_2(x, y) \equiv$$

$$\equiv \xrightarrow[\xrightarrow[f_+ - f_- - \ldots - f_-]{y} - \xrightarrow[f_- - f_+ - \ldots - f_-]{y} - \ldots - \xrightarrow[f_- - f_- - \ldots - f_+]{y}]{x},$$

где $f_+ = \max(e_1, \ldots, e_k)$, $f_- = \min(e_1, \ldots, e_k)$.

Теорема 4. *k-Значная дизъюнкция коэкстенсивна формуле $F_3(x, y)$:*

$$f_\vee(x, y) \equiv F_3(x, y) \equiv$$

$$\equiv \overrightarrow{\overrightarrow{\frac{y}{f_1 - f_2 - \ldots - f_k}} - \overrightarrow{\frac{y}{f_2 - f_2 - \ldots - f_k}} - \ldots - \overrightarrow{\frac{y}{f_k - f_k - \ldots - f_k}}}^{\,x},$$

где f_i — *отношения порядка* 0.

Теорема 5. *k-Значная конъюнкция коэкстенсивна формуле* $F_4(x, y)$:

$$f_\wedge(x, y) \equiv F_4(x, y) \equiv$$

$$\equiv \overrightarrow{\overrightarrow{\frac{y}{f_1 - f_1 - \ldots - f_1}} - \overrightarrow{\frac{y}{f_1 - f_2 - \ldots - f_2}} - \ldots - \overrightarrow{\frac{y}{f_1 - f_2 - \ldots - f_k}}}^{\,x},$$

где f_i — *отношения порядка* 0.

Доказательство теорем 2, 3, 4, 5 осуществляется путем прямой проверки.

Полные системы функций и базисы k-логик. Согласно работе [3] множество отношений $B = \{f_1, \ldots, f_k, \ldots\}$ называется полным в k-логике, если любое отношение f^* данной k-логики может быть представлено в виде формулы конечного порядка над множеством B.

В k-логике, как и в k-значной логике [4], справедливо утверждение: если в k-логике существуют два множества отношений B_1 и B_2, причем множество B_1 полное в k-логике, а для всех отношений $f_i \in B^1$ существуют коэкстенсивные формулы конечного порядка во множестве B_2, т. е. $f_i(B_1) \equiv f_i(B_2)$, то множество отношений B_2 является полным в k-логике.

Теорема 6. *Для любой функции* $f(x_1, \ldots, x_n)$ *k-логики справедливо следующее тождество*:

$$f(x_1, \ldots, x_n) \equiv \bigvee_{\substack{\text{по всем} \\ (\sigma_1, \ldots, \sigma_n)}} [I_{\sigma_1}(x_1) \wedge I_{\sigma_2}(x_2) \wedge \ldots \wedge I_{\sigma_n}(x_n) \wedge$$
$$\wedge f(\sigma_1, \ldots, \sigma_n)].$$

Следствие 1.

$$f(x_1, \ldots, x_n) \equiv \bigvee_{i=1}^{k} [I_{e_i}(x_i) \wedge f(x_1, \ldots, x_{i-1}, e_i, x_{i+1}, \ldots, x_n)].$$

Следствие 2. *Множество отношений* $B = \{f_1, \ldots, f_k,$ $I_1(x), \ldots, I_k(x), x \wedge y, x \vee y\}$, *является полным в k-логике (аналог полного множества Россера и Тьюкетта* [3]).

Теорема 7. *Множество отношений* $K = \{f_1, \ldots, f_k, K(x,$ $y_1, \ldots, y_k)\}$ *является полным в k-логике.*

Доказательство. В силу следствия 1 теоремы 6 имеем

$$f(x_1, \ldots, x_n) \equiv \bigvee_{i=1}^{k} [I_{e_i}(x_i) \wedge f(x_1, \ldots, e_i, \ldots, x_n)].$$

Заменяя все отношения коэкстенсивными, согласно теоремам 3, 4, 5, и применяя к отношениям $f(x_1, \ldots, e_i, \ldots, x_n)$ аналогичную процедуру, приходим к утверждению теоремы.

Понимая под базисом такое полное множество B, для которого не существует другого полного множества B_1, удовлетворяющего условиям $B_1 \subset B$ и $B_1 \neq B$, сформулируем теорему.

Теорема 8. *Множество $K = \{f_1, \ldots, f_k, K(x, y_1, \ldots, y_k)\}$ является базисом k-логики.*

Идея доказательства состоит в том, что для каждого подмножества $K_1 \subset K$ находится отношение f^*, для которого не существует выражения в виде формулы конечного порядка над множеством K_1. В силу громоздкости доказательство опускается.

Определение 5. *Базис, образованный множеством отношений $K = \{f_1, \ldots, f_k, K(x, y_1, \ldots, y_k)\}$, называется k-базисом.*

Понимая под наследственной системой отношений такой базис, в некоторых отношениях которого произведено отождествление ряда переменных, следуя работе [8], введем понятие простого базиса. Пусть B_1 является наследственной системой базиса B. Если множество отношений наследственной системы не является полным в k-логике, базис B называют простым. В работах [9; 11] указано на конечность числа простых базисов и дана верхняя оценка.

Теорема 9. *k-Базис является простым.*

Доказательство. Согласно теореме 3 имеем

$$I_x(y) \equiv$$

$$\equiv \xrightarrow[\xrightarrow[f_+ - f_- - \ldots - f_-]{y} - \xrightarrow[f_- - f_+ - \ldots - f_-]{y} - \ldots - \xrightarrow[f_- - f_- - \ldots - f_+]{y}]{x}.$$

Так как любое отождествление переменных не позволит построить формулу, коэкстенсивную характеристической, наследственная система не является полной, а k-базис является простым.

Следует отметить, что аналогичный результат в булевой алгебре был получен ранее [10].

Теорема 10. *Число различных представлений отношения в k-базисе не превосходит $n!\ k^{(n-2)(n-1)/2}$.*

Доказательство. На первом шаге построения формулы существует n возможностей представления, на втором — $(n-1)k, \ldots$ на i-м — $k^{i-1}(n-i+1)$. Количество шагов — $(n-1)$. Следовательно, общее число представлений может быть выражено так:

$$n(n-1)k \ldots (n-i+1)k^{i-1} \ldots 2k^{n-2} = n!\ k^{(n-2)(n-1)/2}.$$

В связи с множеством представлений возникает проблема выбора конкретного решения, возникает проблема минимизации, если в качестве критерия минимальности считать число вхождений в формулу отношений порядка 0.

Алгоритм построения решения. Согласно постановке задачи, на множество A действуют одновременно отношения ψ_i, образуя канонические отображения A на A/ψ_i. Под отношением α_1 будем понимать отношение вида

$$\alpha_1 : A \to \begin{cases} A/\psi_1; \\ \cdots \\ \cdots \\ A/\psi_n. \end{cases}$$

Согласно теории R-функций, каждый опорный элемент ψ_i сопровождается некоторым отношением r_i. Тогда под отношением α_2 будем понимать отношение вида

$$\alpha_2 : B_j^i \to r_i(B_j^i) \subset R, \quad i = 1, \ldots, n.$$

Следовательно,

$$\alpha_1\alpha_2 : A \to \begin{cases} r_1(A/\psi_1); \\ \cdots\cdots \\ \cdots\cdots \\ r_n(A/\psi_n). \end{cases}$$

Будем рассматривать такие геометрические объекты, опорные элементы которых имеют характеристические функции α_3:

$$\forall x \in B_j^i \to r_i(B_j^i) \to \exists e \in E,$$

где E — множество признаков и $|E|$ — конечна. Тогда $\alpha_3 : r_i \times \times (B_j^i) \to E$ и $\alpha_1\alpha_2\alpha_3 : A \to E^n$. Выбрав удобные с вычислительной точки зрения отношения, мы, естественно, приходим к использованию k-логик с произвольным основанием.

В качестве отношения α_4 рассмотрим некоторую формулу в k-логике, которая позволила бы поставить в соответствие каждому элементу множества A, обладающему n-характеристиками из множества признаков E, элемент множества D:

$$\alpha_4 : E^n \to D; \quad \alpha_1\alpha_2\alpha_3\alpha_4 : A \to D.$$

Под отношением α_5 будем понимать задание соответствия каждому D_i априорно заданного отношения, согласно условию задачи:

$$\alpha_5 \equiv f : D \to A_1; \ \alpha_1\alpha_2\alpha_3\alpha_4\alpha_5 : A \to A_1,$$

что и требовалось получить. Данная конструкция является полезной при решении ряда практических задач распознавания геометрических образов — построения многозначных предикатов для сложных геометрических областей, построения «склеивающих» функций и т. д.

Пример 1. Пусть необходимо построить предикатное уравнение, т. е. уравнение, принимающее значения ε_1, ε_2, ε_3, ε_4, ε_5 на каждом множестве M_1, M_2, M_3, M_4, M_5 соответственно, где

$$M_1 = \{(x, y)/(x^2 + y^2 - R^2 > 0)\};$$
$$M_2 = \{(x, y)/(x^2 + y^2 - R^2 = 0)\};$$

$$M_3 = \{(x,\ y)/(x^2 - a^2 > 0) \cap (x^2 + y^2 - R^2 < 0)\};$$
$$M_4 = \{(x,\ y)/(x^2 - a^2 = 0) \cap (x^2 + y^2 - R^2 < 0)\};$$
$$M_5 = \{(x,\ y)/(x^2 - a^2 < 0) \cap (x^2 + y^2 - R^2 < 0)\}.$$

В качестве опорных элементов выберем

$$\psi_1 : A \to \begin{cases} B_1^1 = \{(x,\ y)/(R^2 - x^2 - y^2 < 0)\} \\ B_2^1 = \{(x,\ y)/(R^2 - x^2 - y^2 = 0)\} \\ B_3^1 = \{(x,\ y)/(R^2 - x^2 - y^2 > 0)\} \end{cases}$$

$$\text{и } \psi_2 : A \to \begin{cases} B_1^2 = \{(x,\ y)/(a^2 - x^2 < 0)\}; \\ B_2^2 = \{(x,\ y)/(a^2 - x^2 = 0)\}; \\ B_3^2 = \{(x,\ y)/(a^2 - x^2 > 0)\}. \end{cases}$$

Следовательно, отношение α_1 определим так:

$$\alpha_1 : A \to \begin{cases} A/\psi_1; \\ A/\psi_2. \end{cases}$$

Для построения отношения α_2 определим сопровождающие опорный элемент функции:

$$r_1 \equiv (a^2 - x^2); \quad r_2 \equiv (R^2 - x^2 - y^2).$$

Тогда для α_2 имеет место

$$\alpha_1\alpha_2 : A \to \begin{cases} r_1\,(A/\psi_1); \\ r_2\,(A/\psi_2). \end{cases}$$

Отношение α_3 выберем в виде

$$\alpha_3 \equiv \operatorname{sign} : r_i(A) \to E, \quad E = \{-1,\ 0,\ 1\}, \quad i = 1,\ 2.$$

Функцию sign определим следующим образом:

$$\operatorname{sign} : x \to \begin{cases} -1, & \text{если } x < 0; \\ 0, & \text{»} \quad x = 0; \\ 1, & \text{»} \quad x > 0. \end{cases}$$

Отношение α_4, согласно условию примера, запишем так:

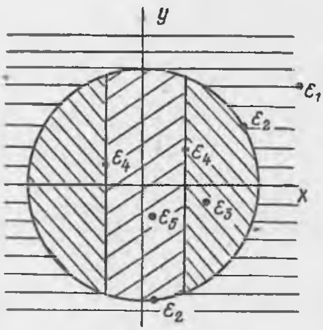
$$\alpha_4 \equiv \frac{\operatorname{sign}(R^2 - x^2 - y^2)}{M_1 - M_2 - \dfrac{\operatorname{sign}(a^2 - x^2)}{M_3 - M_4 - M_5}\to}\to.$$

Под отношением α_5 будем понимать $\alpha_5 : M_i \to \varepsilon_i$, $i = 1, \ldots, 5$. Следовательно, решение можно представить в виде

$$f(x,\ y) \equiv \frac{\operatorname{sign}(R^2 - x^2 - y^2)}{\varepsilon_1 - \varepsilon_2 - \dfrac{\operatorname{sign}(a^2 - x^2)}{\varepsilon_3 - \varepsilon_4 - \varepsilon_5}\to}\to.$$

Используя теорему 1, запишем решение следующим образом:

$$f(x, y) \equiv \varepsilon_1 \frac{\operatorname{sign}(R^2 - x^2 - y^2)\,[1 - \operatorname{sign}(R^2 - x^2 - y^2)]}{2} +$$
$$+ \varepsilon_2 [1 - \operatorname{sign}^2 (R^2 - x^2 - y^2)] +$$
$$+ \frac{\operatorname{sign}(R^2 - x^2 - y^2)\,[1 + \operatorname{sign}(R^2 - x^2 - y^2)]}{2} \times$$
$$\times \left\{ \varepsilon_3 \frac{\operatorname{sign}(a^2 - x^2)\,[1 - \operatorname{sign}(a^2 - x^2)]}{2} + \right.$$
$$\left. + \varepsilon_4 [1 - \operatorname{sign}^2 (a^2 - x^2)] + \varepsilon_5 \frac{\operatorname{sign}(a^2 - x^2)\,[1 + \operatorname{sign}(a^2 - x^2)]}{2} \right\}.$$



Результаты исследования представлены на рисунке.

Список литературы: 1. *Рвачев В. Л.* Элементы дискретного анализа и теории R-функций/Харьк. политехн. ин-т. Харьков, 1972. 352 с. 2. *Рвачев В. Л.* Методы алгебры логики в математической физике. Киев, Наукова думка, 1974. 334 с. 3. *Яблонский С. В.* Функциональные построения в k-значной логике.— Труды МИАН СССР, 1958, т. 51, с. 5—142. 4. *Яблонский С. В. Лупанов О. Б.* Дискретная математика и математические вопросы кибернетики, т. I. М., Наука, 1974. 147 с. 5. *Мальцев А. И.* Алгебраические системы. М., Наука, 1970. 288 с. 6. *Робинсон А.* Введение в теорию моделей и математику алгебры. М., Наука, 1967. 279 с. 7. *Куратовский К., Мостовский А.* Теория множеств. М., Мир, 1970. 207 с. 8. *Новиков П. С.* Элементы математической логики. М., Физматгиз, 1959. 76 с. 9. *Яблонский С. В.* О суперпозициях функций в P_k.— Проблемы кибернетики. М., 1963, вып. 9, с. 12—15. 10. *Шестопал Г. А.* О числе простых базисов булевых функций.— ДАН СССР, 1961, т. 140, № 2, с. 141—149. 11. *Алексеев В. Б.* О числе простых базисов в k-значной логике.— Дискретный анализ. Новосибирск, 1971, вып. 19, с. 27—36. 12. *Сироджа И. Б.* Алгоритм распознавания геометрических образов.— Изв. АН СССР. Техническая кибернетика, 1967, т. 5, с. 136—144.

УДК 681.3.06

С. К. КОЛУБАЙ

ПРОГРАММИРОВАНИЕ НА МК-ЯЗЫКЕ ТИПА $\langle i, k, j, l \rangle$

В работе [1] предложен асинхронный внутренний язык вычислительной системы, названный МК-языком типа $\langle i, k, j, l \rangle$, а в работе [2] — параллельная вычислительная система для обработки МК-программ программ на МК-языке этого типа, включающая в себя специальным образом сконструированную активную память с логикой, названную процессоро-памятью (Пр-П). Рассмотрим вопросы программирования на МК-языке типа $\langle i, k,$

j, l> и исполнение МК-программ в параллельной вычислительной системе (ПВС).

Процесс составления и выполнения МК-программы существенно отличается от составления программ на традиционных языках — АЛГОЛ-60 [3], ФОРТРАН [4], КОБОЛ [5], ПЛ/1 [6], Ассемблер [7], АЛМО [8] и др. Программирование на МК-языке ближе к программированию на таких языках, как k-язык [9], *DFL* [10], язык таблиц решений [11] и др.

Программирование и выполнение МК-программ типа $<i, k, j, l>$ нами осуществляется на примерах построения МК-программ для двух типов операторных схем: линейной и разветвляющейся без циклов. Задание операторных схем производим в графовой форме, причем ввод и вывод указываем соответственно буквами В и Ы, за которыми в круглых скобках перечислены вводимые (выводимые) переменные. Термины и обозначения те же, что и в работах [1; 2].

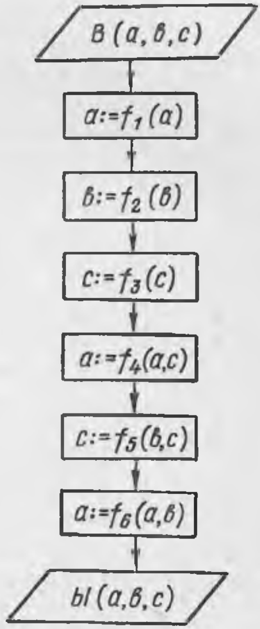


Рис. 1.

МК-программа линейной операторной схемы. МК-программа, соответствующая линейной операторной схеме, изображенной на рис. 1, имеет вид

1) $B(a, b, c)$; 5) $a := f_4(a, c)(2, 4)$;

2) $a := f_1(a)(1)$; 6) $c := f_5(b, c)(3, 4)$;

3) $b := f_2(b)(1)$; 7) $a := f_6(a, b)(5, 3)$;

4) $c := f_3(c)(1)$; 8) Ы $(a, b, c)(7, 6)$.

При записи данной МК-программы приняты упрощенные обозначения: вместо инструкции типа i) $k(0, l)$ и i) $k(j, 0)$ записывали i) $k(l)$ и i) $k(j)$ соответственно, так как для обработки их в ячейках процессоро-памяти безразлично, какой из номеров j или l равен нулю; вместо инструкций типа i) $k(0, 0)$ записывали i) k. В МК-программе, соответствующей линейной операторной схеме, номера j и l инструкций задают связи между инструкциями по данным, т. е. указывают номера инструкций, в которых происходит вычисление используемых данных. При выполнении этой МК-программы на ПВС [2] в начальный момент времени готова к выполнению единственная инструкция — с номером 1. После ее выполнения произойдет обнуление номеров у инструкций с номерами 2, 3, 4, а ячейки Пр-П, содержащие данные инструкции, перейдут в состояние «готова». При наличии хотя бы одной ячейки Пр-П в состоянии «готова» центральное устройство ПВС выдает на блок Пр-П сигнал чтения Ч, и команды инструкции с номерами 2, 3, 4 будут переданы на процессоры. Порядок завершения работы команд инструкций отвечает порядку поступления их номера на процессоро-память. Если первой завершит работу команда из инструкции с номером 3, будет иметь место обнуление номера 3 в инструкциях с номерами 6 и 7. Однако ни одна из ячеек Пр-П, хранящая инструкции с номерами 6 и 7, не перейдет в состояние «готова».

Предположим теперь, что следующей выполнится команда инструкции с номером 4. В этом случае произойдет обнуление номера 4 в инструкциях с номерами 5 и 6, и в результате среди ячеек Пр-П, занимаемых МК-программой, появится ячейка, находящаяся в состоянии «готова». Эта ячейка содержит инструкцию с номером 6. Команда инструкции с номером 6 передается на процессор, начиная с этого момента времени на процессорах одновременно выполняются команды инструкции с номерами 2 и 6.

Таблица 1

Время t	№ инструкции							
	1	2	3	4	5	6	7	8
0	C_4	C_2	C_2	C_3	C_1	C_1	C_1	C_1
1	C_5	C_2	C_2	C_3	C_1	C_1	C_1	C_1
2	C_0	C_4	C_4	C_4	C_1	C_1	C_1	C_1
3	C_0	C_5	C_5	C_5	C_1	C_1	C_1	C_1
4	C_0	C_5	C_0	C_5	C_1	C_2	C_3	C_1
5	C_0	C_5	C_0	C_0	C_3	C_4	C_3	C_1
6	C_0	C_5	C_0	C_0	C_3	C_5	C_3	C_1
7	C_0	C_0	C_0	C_0	C_4	C_5	C_3	C_1
8	C_0	C_0	C_0	C_0	C_5	C_5	C_3	C_1
9	C_0	C_0	C_0	C_0	C_0	C_5	C_4	C_1
10	C_0	C_0	C_0	C_0	C_0	C_5	C_5	C_1
11	C_0	C_0	C_0	C_0	C_0	C_0	C_5	C_3
12	C_0	C_0	C_0	C_0	C_0	C_0	C_0	C_4
13	C_0	C_0	C_0	C_0	C_0	C_0	C_0	C_5
14	C_0	C_0	C_0	C_0	C_0	C_0	C_0	C_0

Пусть в дальнейшем вычислительный процесс протекает так: сначала выполняется команда инструкции с номером 2, после чего ячейка Пр-П, содержащая инструкцию с номером 5, переходит в состояние «готова», команда этой инструкции передается на свободный процессор.

Предположим, что выполняющиеся команды инструкций с номерами 5 и 6 завершат работу в следующем порядке: сначала выполнится команда инструкции с номером 5, а затем — с номером 6. В этот момент времени произойдет передача на процессор команды инструкции с номером 7, а после завершения работы команд инструкций с номерами 6 и 7 ячейка Пр-П, хранящая команду инструкции 8, перейдет в состояние «готова». Осуществится вывод, работа МК-программы на этом завершится. В табл. 1 отражен процесс изменения состояния ячеек Пр-П, содержащих в начальный момент времени инструкции МК-программы (обозначения — принятые при построении графа переходов ячейки Пр-П [2]).

В качестве начального состояния ячеек, имеющих инструкции с номерами 2, 3 и 4, были произвольно выбраны состояния C_2, C_2 и C_3 соответственно. Из табл. 1 видна динамика происходящих в блоке Пр-П процессов, а также максимальная параллельность при выполнении этой МК-программы. При $t = 3$ количество одновременно выполняемых команд максимально и равно трем. Примечательно, что после завершения выполнения МК-программы линейной операторной схемы все ячейки Пр-П, которые в начальный момент времени занимала МК-программа, перешли в состояние «свободна», т. е. в процессе выполнения МК-программа как бы самоуничтожилась.

Выполнение данной МК-программы возможно и на однопроцессорной ВС, однако наиболее рационально с точки зрения загрузки процессоров инструкциями и времени решения выполнение МК-программы на двухпроцессорной ПВС. Обеспечение же максимального параллелизма при выполнении этой МК-программы достигается на трехпроцессорной ПВС.

Составление МК-программы для линейной операторной схемы несложно, так как необходимо следить только за номерами тех инструкций, в которых вычисляются данные, использованные в данной инструкции. Рассмотренная ранее МК-программа для линейной операторной схемы позволяет осуществить любую перестановку ее инструкций. Например, ее можно записать в следующем виде:

5) $a := f_4(a, c)(2, 4)$;	4) $c := f_3(c)(1)$;
3) $b := f_2(b)(1)$;	1) В (a, b, c);
8) Ы $(a, b, c)(7, 6)$;	7) $a := f_6(a, b)(5, 3)$;
2) $a := f_1(a)(1)$;	6) $c := f_5(b, c)(3, 4)$.

При этом МК-программа становится вообще менее наглядной, однако ее выполнение будет происходить точно так же, как и предыдущей, в этом смысле приведенные записи МК-программы эквивалентны. Точно так же эквивалентна им обеим следующая МК-программа:

1) В (a, b, c);	5) $b := f_2(b)(1)$;
2) $a := f_1(a)(1)$;	6) $a := f_6(a, b)(4, 5)$;
3) $c := f_3(c)(1)$;	7) $c := f_5(b, c)(5, 3)$;
4) $a := f_4(a, c)(2, 3)$;	8) Ы $(a, b, c)(6, 7)$.

Можно сделать вывод о том, что при записи МК-программы для решения данной задачи единственным требованием является правильное указание при помощи номеров j и l связей между инструкциями МК-программы по данным.

МК-программа разветвляющейся операторной схемы. Составление МК-программ для разветвляющихся операторных схем покажем на примерах [12].

Пример 1. МК-программа для операторной схемы, показанной на рис. 2, *а*, имеет следующий вид:

1) a_1;

2) $a_2 : 3, 4 (1)$;

5) $a_3 (3)$;

5) $a_4(4)$;

6) $a_5(5)$.

Здесь в МК-программе инструкция под номером 2 вырабатывает номер 3, если условие, заданное в операторе a_2, выполняется, или номер 4 — в противном случае. Две инструкции в МК-программе имеют одинаковый номер 5, потому что выполнение оператора a_5 возможно лишь после того, как выполнится один из двух операторов a_3 или a_4, находящихся в разных ветвях операторной схемы. Путем присвоения одинакового номера инструкциям разных ветвей операторной схемы и задания этого номера в инструкции с оператором, следующим за разветвлением, в МК-программе указываются информационно-управляющие связи для случая разветвляющихся процессов. Заметим, что присваивать номера инструкциям с операторами a_2 и a_5 необязательно, так как в МК-программе они не используются для указания управляющих или

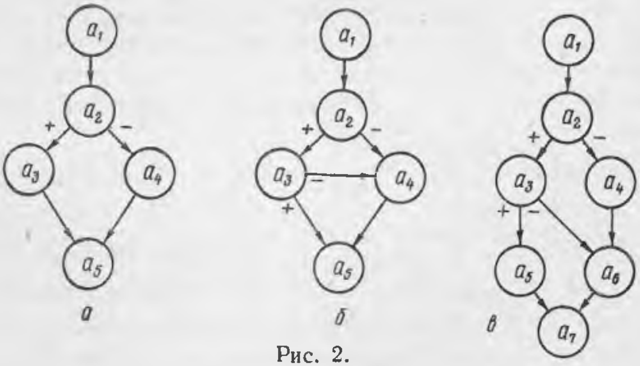


Рис. 2.

информационных связей. Данную МК-программу можно записать в одном из следующих видов:

1) a_1;		1) a_1;
2) $a_2: 2,\ 3(1)$;		$a_2: 2,\ 3(1)$;
4) $a_3(2)$;	или	4) $a_3(2)$;
4) $a_4(3)$;		4) $a_4(3)$;
5) $a_5(4)$		$a_5(4)$.

С целью сократить записи МК-программ иногда будем применять строчную форму записи, несмотря на некоторую потерю наглядности. При строчной форме МК-программа задается как множество инструкций $\{i_1;\ i_2;\ \ldots;\ i_n\}$, в котором в качестве разделителя используется символ (;) — точка с запятой. При этом инструкции МК-программы нагляднее записывать так: $i,\ k(j,\ l)$. Например, предыдущая МК-программа в строчной форме записи имеет вид

$$\{1.\ a_1;\ a_2: 2{,}3(1);\ 4.\ a_3(2);\ 4.\ a_4(3);\ a_5(4)\}.$$

Пример 2. МК-программа для операторной схемы, показанной на рис. 2, *б*, может иметь такой вид:

$$\{1.\ a_1;\ a_2: 2,\ 3(1);\ a_3: 4,\ 3(2);\ 4.\ a_4(3);\ a_5(4)\}.$$

Здесь для перевода в состояние готовности инструкции, содержащей оператор a_5, использован один и тот же номер 3 в инструкциях с операторами a_2 и a_3 в качестве номера, вырабатываемого инструкцией управления по значению «ложь» (—) логической функции.

Перевод в сос ояние готовности инструкции с оператором a_5 в случае, если значение оператора a_3 есть «истина» (+) или закончил выполнение оператор a_4, осуществляется использованием одного и того же номера 4 в инструкции с оператором a_3 и в качестве номера инструкции с оператором a_4.

Пример 3. МК-программа для операторной схемы, приведенной на рис. 2, *в*, имеет вид

$$\{1.\ a_1;\ a_2: 2,\ 3(1);\ a_3: 4,\ 5(2);\ 5.\ a_4(3);\ 6.\ a_5(4);\ 6.\ a_6(5);\ a_7(6)\}.$$

В этой МК-программе использован номер 5 для объединения ветвей, сходящихся на операторе a_6, и номер 6 для объединения ветвей, сходящихся на операторе a_7.

Пример 4. Рассмотрим теперь МК-программу для более сложной операторной схемы, показанной на рис. 3, при составлении которой использованы описанные приемы, а также применена инструкция преобразования номеров $i.\ P(j,\ l)$. МК-программа для этого случая имеет вид

$$\{1.\ B(a,\ b,\ c);\ L(a): 2,\ 3(1);\ 4.\ d:= f_1(a,\ b)(2);\ L(b):\ 5,\ 6(2);$$
$$7.\ d:=f_5(d)(5,\ 4);\ 8.\ c:=f_6(c)(5);$$
$$Ы(c,\ d)(7,\ 8);\ 7.\ d:=f_3(d)(6,\ 4);$$
$$8.\ c:=f_4(c)(6);\ 4.\ d:=f_2(b,\ c)(3);$$
$$6.\ P(4,\ 3)\}.$$

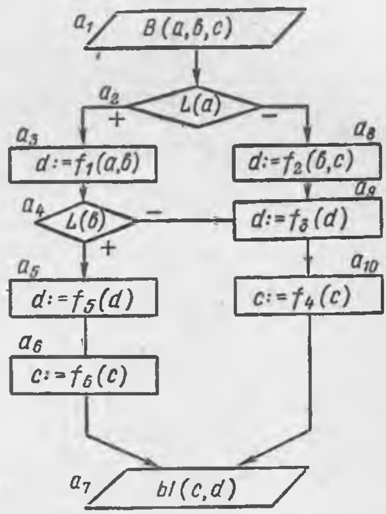


Рис. 3.

Инструкция преобразования номеров 6. $P(4,\ 3)$ вырабатывает при наличии номеров 4 и 3 номер 6, который переводит в состояние готовности инструкции, вычисляющие значения функций f_3 и f_4.

Данная МК-программа сохранила присущий исходной задаче, для которой была написана операторная схема (рис. 2, *г*), параллелизм и асинхронность. Например, вычисление операторов a_3 и a_4, a_5 и a_6, a_9 и a_{10} может быть осуществлено параллельно и в произвольном порядке. Для случая $L(a)$ — «истина», а $L(b)$ — «ложь» возможны следующие вычислительные процессы:

1. $a_1,\ a_2,\ a_3,\ a_4,\ a_9,\ a_{10},\ a_7;$
2. $a_1,\ a_2,\ a_4,\ a_3,\ a_9,\ a_{10},\ a_7;$
3. $a_1,\ a_2,\ a_3,\ a_4,\ a_{10},\ a_9\ a_7;$
4. $a_1,\ a_2,\ a_4,\ a_3,\ a_{10},\ a_9,\ a_7;$
5. $a_1,\ a_2,\ a_4,\ a_{10},\ a_3,\ a_9,\ a_7;$
6. $a_1,\ a_2,\ \begin{matrix}a_3\\ a_4\end{matrix},\ \begin{matrix}a_9\\ a_{10}\end{matrix},\ a_7;$
7. $a_1,\ a_2,\ a_4,\ \begin{matrix}a_3\\ a_{10}\end{matrix},\ a_9,\ a_7.$

Рассмотрим теперь, как меняется состояние ячеек памяти в процессе выполнения МК-программы для разветвляющейся операторной схемы. Для конкретности предположим, что в процессе выполнения был получен седьмой вычислительный процесс:

$$a_1,\ a_2,\ a_4,\ \begin{matrix} a_3 \\ a_{10} \end{matrix},\ a_9,\ a_7.$$

Перепишем МК-программу, присваивая произвольные номера тем инструкциям, у которых они не используются в процессе выполнения, а также определяя конкретно значения номеров j, l всех инструкций:

$$\{1.\ \text{В}(a,\ b,\ c)(0,\ 0);\ 2.\ L(a):2,\ 3(1,\ 0);\ 4.\ d:=f_1(a,\ b)(0,\ 2);$$
$$5.\ L(b):5,\ 6(2,\ 0);\ 7.\ d:=f_5(d)(5,\ 4);\ 8.\ c:=f_6(c)(5,\ 0);$$
$$9.\ \text{Ы}(c,\ d)(7,\ 8);\ 7.\ d:=f_3(d)(6,\ 4);\ 8.\ c:=f_4(c)(6,\ 0);$$
$$4.\ d:=f_2(b,\ c)(3,\ 0);\ 6.\ P(4,\ 3)\}.$$

Таблица 2

Время t	Ячейки Пр-П										
	$Я_1$	$Я_2$	$Я_3$	$Я_4$	$Я_5$	$Я_6$	$Я_7$	$Я_8$	$Я_9$	$Я_{10}$	$Я_{11}$
0	C_4	C_3	C_2	C_3	C_1	C_3	C_1	C_1	C_3	C_3	C_1
1	C_5	C_3	C_2	C_3	C_1	C_3	C_1	C_1	C_3	C_3	C_1
2	C_0	C_4	C_2	C_3	C_1	C_3	C_1	C_1	C_3	C_3	C_1
3	C_0	C_5	C_2	C_3	C_1	C_3	C_1	C_1	C_3	C_3	C_1
4	C_0	C_0	C_4	C_4	C_1	C_3	C_1	C_1	C_3	C_3	C_1
5	C_0	C_0	C_4	C_5	C_1	C_3	C_1	C_1	C_3	C_3	C_1
6	C_0	C_0	C_4	C_5	C_1	C_3	C_1	C_2	C_4	C_3	C_0
7	C_0	C_0	C_5	C_5	C_1	C_3	C_1	C_2	C_5	C_3	C_0
8	C_0	C_0	C_0	C_5	C_3	C_0	C_3	C_4	C_0	C_0	C_0
9	C_0	C_0	C_0	C_5	C_3	C_0	C_3	C_5	C_0	C_0	C_0
10	C_0	C_0	C_0	C_5	C_0	C_0	C_4	C_0	C_0	C_0	C_0
11	C_0	C_0	C_0	C_5	C_0	C_0	C_5	C_0	C_0	C_0	C_0
12	C_0	C_0	C_0	C_5	C_0	C_0	C_0	C_0	C_0	C_0	C_0

Для размещения данной программы в процессоро-памяти необходимо 11 ячеек. Обозначим эти ячейки через $Я_1$, $Я_2$, ..., $Я_{11}$ и будем считать, что инструкции заняли ячейки в том порядке, в каком они записаны, т. е. инструкция 1. $\text{В}(a,\ b,\ c)\times\times(0,\ 0)$ заняла ячейку $Я_1$, ..., инструкция 6. $P(4,\ 3)$ заняла ячейку $Я_{11}$. Изменение состояний ячеек Пр-П показано в табл. 2.

Как видно, в момент времени $t=6$ ячейка $Я_4$ не перешла в состояние «свободна» из состояния «выполняюсь», хотя инструкция выполнилась. Это произошло потому, что после выполнения инструкции управления на Пр-П поступает один из номеров, определяющих истинность или ложность логической функции, а не номер инструкции. Однако таких ситуаций при выполнении МК-программы допускать нельзя, так как концом выполнения ее считается отсутствие ячеек в состояниях «готова» и «выполняюсь». Поэтому применяют следующий прием при написании МК-программы. Каждая инструкция управления записывается тремя инструкциями вида $\{i.\ L:m,\ n(j,\ l);\ i.\ P(m);\ i.\ P(n)\}$. Инструкции преобразования номеров m и n в номер i обеспечивают перевод ячейки, занимаемой инструкцией управления, в состояние «свободна». Так, для предыдущего случая инструкция 5. $L(b): 5, 6\,(2, 0)$ заменяется тремя инструкциями следующего вида:

$$\{10.\ L(b): 5, 6\,(2, 0);\ 10.\ P(5, 0);\ 10.\ P(6, 0)\}.$$

Пусть эти три инструкции расположены соответственно в ячейках $Я_4$, $Я_{12}$ и $Я_{13}$, тогда в процессе выполнения МК-программы они будут принимать состояния, показанные в табл. 3.

Как видно, ячейка $Я_4$ перешла в момент $t = 8$ в состояние «свободна» одновременно с ячейкой $Я_{13}$, при этом ячейка $Я_{12}$ осталась в состоянии «неготова». Возможны и другие приемы перевода ячейки Пр-П, занимаемой инструкцией управления, в состояние «свободна» из состояния «выполняюсь». Например, путем передачи на Пр-П не только номера, определяющего значение логической функции, но и номера самой инструкции. Иначе говоря, инструкция управления записывается в виде $i.\ L: m, n\,(j, l)$, причем $i \neq m$ и $i \neq n$, и после выполнения ее на Пр-П передается пара номеров: (i, m) либо (i, n).

Таблица 3

Ячейки Пр-П	Время t										
	0	1	2	3	4	5	6	7	8	9	10
$Я_4$	C_3	C_3	C_3	C_3	C_4	C_5	C_5	C_5	C_0	C_0	C_0
$Я_{12}$	C_3	C_3	C_3	C_3	C_3	C_3	C_3	C_3	C_3	C_3	C_3
$Я_{13}$	C_3	C_3	C_3	C_3	C_3	C_3	C_4	C_5	C_0	C_0	C_0

После завершения работы МК-программы для разветвляющейся операторной схемы возможно, что некоторые ячейки, занимаемые МК-программой, останутся в состоянии «неготова», как, например, $Я_{12}$ (табл. 3). При хранении нескольких МК-программ в процессоро-памяти одновременно такие ячейки, оставшиеся после выполнения в состоянии «неготова», могут сильно «засорить» Пр-П. Поэтому предусматриваются средства очистки Пр-П, т. е. средства перевода ячеек Пр-П из состояния «неготова» в состояние «свободна». В качестве такого средства предлагается прежде всего использовать одинаковые номера для инструкций разных ветвей разветвляющегося процесса.

Так, в табл. 2 показано, что ячейки $Я_5$ и $Я_6$ перешли из состояния «неготова» в состояние «свободна» после выполнения инструкций, хранящихся в ячейках $Я_8$ и $Я_9$, которые имели одинаковые номера с инструкциями ячеек $Я_5$ и $Я_6$. Однако этого может оказаться недостаточно для очистки Пр-П, как в случае ячейки $Я_{12}$, поэтому необходимы какие-то общие приемы централизованного управления. В случае однопрограммной работы таким приемом может быть общая очистка Пр-П после завершения работы каждой программы. В случае же двух или более программ, выполняющихся одновременно и с постоянной подкачкой новых программ, можно применить следующий прием. Все номера инструкций, используемые в каждой из программ, имеют свои диапазоны изменения, определяемые при загрузке МК-программы в Пр-П. В конце выполнения определенной программы осуществляется выдача всех номеров выполнившейся программы на канал номеров Пр-П. Это приведет к полной очистке всей области Пр-П, ранее занятой данной МК-программой, при этом ни одна ячейка, занятая другими программами, не будет затронута.

Таким образом, рассмотрение методов программирования в МК-языке типа $<i, k, j, l>$ для линейных и разветвляющихся операторных схем позволяет заключить, что принципиальные трудности написания и выполнения МК-программ отсутствуют.

Список литературы: 1. *Колубай С. К.* Об одном подходе к построению внутреннего языка вычислительной системы.— Проблемы бионики. Харьков, 1979, вып. 22, с. 33—37. 2. *Колубай С. К.* Параллельная вычислительная система для МК-программ.— Проблемы бионики [см. статью в настоящем выпуске]. 3. *Халилов А. И., Ющенко А. А.* АЛГОЛ-60. Киев, Вища школа, 1975. 352 с. 4. *Первин Ю. А.* Основы ФОРТРАНА. М., Наука, 1972. 216 с. 5. *Коддингтон Л.* Ускоренный курс КОБОЛа. М., Мир, 1974. 272 с. 6. *Скотт Р., Сондак Н.* ПЛ/1 для программистов. М., Статистика, 1977. 224 с. 7. *Гурова Л. И.* Основы программирования. М., Статистика, 1976. 232 с. 8. *Богданов В. В., Ермаков Е. А., Маклаков А. В.* Программирование на языке АЛМО. М., Статистика, 1976. 118 с. 9. *Шахбазян К. В., Лебединский М. М.* Функциональный алгоритмический язык.— Труды мат. ин-та АН СССР им. В. А. Стеклова, 1968, т. 96, с. 16—47. 10. *Rumbaugh J. E.* A parallel asynchronous computer architecture for data flow programs.— Proj. MAC Techn. Rept. Massachussetts, 1975, p. 1-271. 11. *Хамби Э.* Программирование таблиц решений. М., Мир, 1974. 70 с. 12. *Котов В. Е.* Теория параллельного программирования.— Кибернетика, 1974, № 2, с. 1—18.

УДК 681.323

Г. Ф. ДЮБКО канд. техн. наук, *Ю. С. ЗАМАЛЕЕВ*

АВТОМАТИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ПРОИЗВОЛЬНЫХ ФУНКЦИЙ В СИСТЕМУ УРАВНЕНИЙ ШЕННОНА

К проблеме автоматизации программирования для различны[х] классов вычислительных машин можно подойти как к моделиро[ванию] интеллектуальных функций человека при преобразовани[и] информации из одной формы в другую. При этом осуществляетс[я] перевод с языков программирования высокого уровня на машин[но-]ориентированный язык. Необходимо отметить, что они прим[и]тивны и ни в коей мере не конкурируют с естественными языкам[и].

В предлагаемой работе моделируется деятельность челове[ка] по преобразованию математических зависимостей, записанных [на] языке высокого уровня, в систему уравнений Шеннона (СУШ) — некоторую форму машинно-ориентированного языка. Как показа[ла] практика, уровень использования вычислительной техники, [в] частности интегрирующих машин, в большой мере зависит [от] удобства программирования, от степени близости языка програ[м]мирования к языку пользователя. При программировании д[ля] интегрирующих машин необходимо все решаемые задачи представлять в форме СУШ. Эта работа трудоемка, целесообразно автоматизировать ее так, чтобы по записи задачи на обычном математическом языке автоматически получать СУШ. Известные методы преобразования различных математических уравнений в СУШ базируются на СУШ, полученных от простых функций, образующих данное уравнение. В самом общем виде система уравнений Шеннона может быть записана следующим образом [1]:

$$\left.\begin{aligned} dz_\gamma &= \sum_{\alpha=0}^{n}\sum_{\beta=0}^{n} a_{\gamma\alpha\beta} z_\alpha dz_\beta, \\ dz_0 = 0,\ dz_1 &= dx,\ z_\gamma(x_0) = z_{\gamma 0}\ (\gamma = 2, 3, \ldots, n) \end{aligned}\right\} \tag{1}$$

или

$$\left.\begin{aligned} dz_k &= \sum_{s=0}^{N}\sum_{r=0}^{N} A_{ksr} z_s dz_r, \\ dz_0 = 0,\ dz_1 &= dx,\ z_k(x_0) = z_{k0},\ k = 2, 3, \ldots, N, \end{aligned}\right\} \tag{2}$$

где в системе уравнений (1) $a_{\gamma\alpha\beta}$ — любые постоянные коэффициенты, а в системе уравнений (2) A_{ksr} — постоянные коэффициенты, равные 0 или 1.

Процесс преобразования функциональных зависимостей в СУШ состоит в расчленении сложных функций на простейшие, преобразовании последних в СУШ и объединении всех элементарных СУШ в общую СУШ [1]. В соответствии с методами преобразования элементарных функций (т. е. функций, которые не являются суперпозициями никаких других функций) в СУШ, описанными в работе [1], построены СУШ элементарных функций, которые приводятся в табл. 1.

Таблица 1

Вид функции	$z = 1/x$	$z = \sin x$
СУШ	$dz = -z_1 dx$ $dz_1 = 2z dz$	$dz = z_1 dx$ $dz_1 = -z dx$
Вид функции	$z = \cos x$	$z = \sqrt{x}$
СУШ	$dz = -z_1 dx$ $dz_1 = z dx$	$dz = \frac{1}{2} z_1 dx$ $dz_1 = -z_2 dz$ $dz_2 = 2z_1 dz_1$
Вид функции	$z = \log_a x$	$z = z_1/z_2$

СУШ	$dz = (\log_a l)\, z_1 dx$ $dz_1 = -z_2 dx$ $dz_2 = 2z_1 dz_1$	$dz = z_1 dz_3 + z_3 dz_1$ $dz_3 = -z_4 dz_2$ $dz_4 = 2z_3 dz_3$
Вид функции	$z = \operatorname{tg} x$	$z = \arcsin x$
СУШ	$dz = z_1 dx$ $dz_1 = -z_2 dz_3$ $dz_2 = 2z_1 dz_1$ $dz_3 = 2z_4 dz_5$ $dz_4 = dz_5$ $dz_5 = -z_6 dx$ $dz_6 = z_5 dx$	$dz = z_1 dx$ $dz_1 = -z_3 dz_2$ $dz_2 = z_5 dz_4/2$ $dz_3 = 2z_1 dz_1$ $dz_4 = dz_7 - dz_8$ $dz_5 = -z_6 dz_2$ $dz_6 = 2z_5 dz_5$ $dz_7 = 0$ $dz_8 = 2z_9 dx$ $dz_9 = dx$ $z_7(0) = 1$
Вид функции	$z = x^n$ (n — целое)	$z = a^x$
СУШ	$dz = nz_1 dx$ $dz_1 = (n-1) z_2 dx$ . . . . $dz_{n-2} = 2z_{n-1} dx$ $dz_{n-1} = dx$	$dz = (\ln a)\, z dx$
Вид функции	$z = x$	$z = \text{const}$
СУШ	$dz = dx$	$dz = 0$ $z(0) = \text{const}$
Вид функции	$z = z_1 + z_2$	$z = z_1 z_2$
СУШ	$dz = dz_1 + dz_2$	$dz = z_1 dz_2 + z_2 dz_1$

Количество элементарных функций, которыми мы пользуемся на практике, невелико, поэтому нетрудно составить СУШ для всех элементарных функций и свести их в одну таблицу (аналогичную табл. 1), которую затем можно использовать для получения СУШ любой сложной функции. Ниже приводится описание алгоритма преобразования произвольной функции, заданной аналитически, в форму Шеннона. Для этого алгоритма табл. 1 преобразована в табл. 2.

Таблица 2

Вид функции	$z_j = 1/< b_n >$	$z_j = \sin < b_n >$
СУШ	$dz_j = -z_{j+1} d < b_n >$ $dz_{j+1} = 2z_j dz_j$	$dz_j = z_{j+1} d < b_n >$ $dz_{j+1} = -z_j d < b_n >$
Вид функции	$z_j = \cos < b_n >$	$z_j = \sqrt{< b_n >}$
СУШ	$dz_j = -z_{j+1} d < b_n >$ $dz_{j+1} = z_j d < b_n >$	$dz_j = \frac{1}{2} z_{j+1}\, d < b_n >$ $dz_{j+1} = -z_{j+2} dz_j$ $dz_{j+2} = 2z_{j+1} dz_{j+1}$
Вид функции	$z_j = \log_d < b_n >$	$z_j = < b_{n1} > / < b_{n2} >$
СУШ	$dz_j = (\log_d 1)\, z_{j+1} d < b_n >$ $dz_{j+1} = -z_{j+2} d < b_n >$ $dz_{j+2} = 2z_{j+1} dz_{j+1}$	$dz_j = < b_{n1} > dz_{j+3} +$ $+ z_{j+3} d < b_{n1} >$ $dz_{j+3} = -z_{j+4} d < b_{n2} >$ $dz_{j+4} = 2z_{j+3} dz_{j+3}$
Вид функции	$z_j = \lg < b_n >$	$z_j = \arcsin < b_n >$
СУШ	$dz_j = z_{j+1} d > b_n >$ $dz_{j+1} = -z_{j+2} dz_{j+3}$ $dz_{j+2} = 2z_{j+1} dz_{j+1}$ $dz_{j+3} = 2z_{j+4} dz_{j+5}$ $dz_{j+4} = dz_{j+5}$ $dz_{j+5} = -z_{j+6} d < b_n >$ $dz_{j+6} = z_{j+5} d < b_n >$	$dz_j = z_{j+1} d < b_n >$ $dz_{j+1} = -z_{j+3} dz_{j+2}$ $dz_{j+2} = \frac{1}{2} z_{j+5} dz_{j+4}$ $dz_{j+3} = 2z_{j+1} dz_{j+1}$ $dz_{j+4} = dz_{j+7} - dz_{j+8}$ $dz_{j+5} = -z_{j+6} dz_{j+2}$ $dz_{j+6} = 2z_{j+5} dz_{j+5}$ $dz_{j+7} = 0$ $dz_{j+8} = 2z_{j+9} d < b_n >$ $dz_{j+9} = d < b_n >$ $z_{j+7}(0) = 1$
Вид функции	$z_j = < b_n > \uparrow n$	$z_j = a \uparrow < b_n >$

Продолжение табл. 2

СУШ	$dz_j = nz_{j+1}d < b_n >$ $dz_{j+1}=(n-1)z_{j+2}d<b_n>$ $\cdots\cdots\cdots$ $dz_{j+n-2}=2z_{j+n-1}d<b_n>$ $dz_{j+n-1} = d < b_n >$	$dz_j = (\ln a)\, z_j d < b_n >$
Вид функции	$z_j = < b_n >$	$z_j = \mathrm{const}$
СУШ	$dz_j = d < b_n >$	$dz_j = 0$ $z_j\,(0) = \mathrm{const}$
Вид функции	$z_j = <b_{n1}> + < b_{n2} >$	$z_j = <b_{n1}> \times <b_{n2}>$
СУШ	$dz_j = d < b_{n1} > +$ $+ d < b_{n2} >$	$dz_j = <b_{n1}> d < b_{n2}> +$ $+ <b_{n2}> d < b_{n1} >$

Здесь независимая переменная x заменена символом $<b_n>$ — входная переменная,— а каждой функции z_j присвоен текущий номер $j+\nu$, ($\nu = 0, 1, 2, \ldots$).

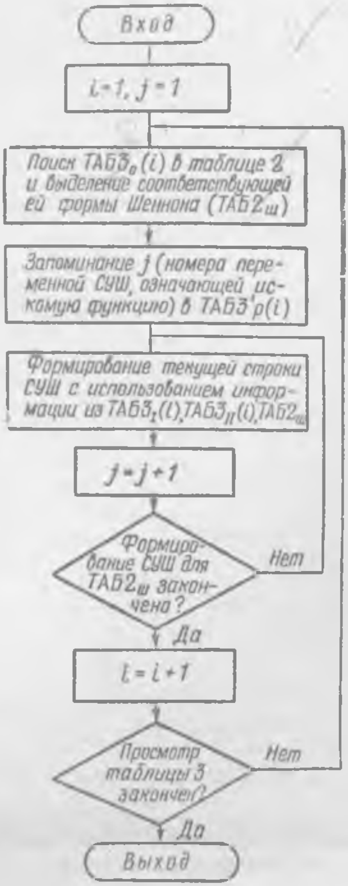
Табл. 2 задает элементарные блоки в виде соответствующих элементарным функциям СУШ и является базой данных алгоритма преобразования. Другой базой данных алгоритма преобразования служит декомпозированная запись исходной сложной функции. В качестве декомпозированной записи можно избрать польскую инверсную запись, двоичное дерево, триады или тетрады [2] и вообще любую запись, которая за один просмотр ее слева направо позволяет выделить все операции и операнды, а также порядок выполнения операций. Эта запись — промежуточная между исходным языком высокого уровня и конечным (СУШ). Получить декомпозированную запись из исходной можно путем синтаксического анализа и семантической обработки в рамках грамматики, описывающей исходный язык [3]. Получение декомпозированной записи в данной работе не рассматривается.

В качестве промежуточной формы — декомпозированной записи — изберем тетрады, которые каждую функцию представляют в виде трехадресной машинной команды. Первой компонентой этой команды является вид функции, далее — I и II операнды, а четвертой компонентой является имя, значением которого есть переменная СУШ, обозначающая функцию, указанную в первой компоненте. Последовательность команд задает последовательность элементарных функций в сложной функции. Все команды можно объединить в табл. 3.

Таким образом, блок-схема алгоритма преобразования тетрад в СУШ, представленная на рисунке, имеет базами данных табл. 2

Таблица 3

№ п/п	Операция	I операнд	II операнд	Результат
1	sin	x	—	R_1
2	arc sin	x	—	R_2
3	$\times$	R_1	R_2	R_1
4	$\log_2$	x	—	R_2
5	/	R_1	R_2	R_1
6	cos	x	—	R_2
7	$\sqrt{}$	x	—	R_2
8	tg	x	—	R_3
9	$\uparrow$	R_3	2	R_3
10	+	R_2	R_3	R_2
11	/	R_1	R_2	R_1



Блок-схема алгоритма преобразования произвольной функции в форму Шеннона.

и 3. Проиллюстрируем работу алгоритма на примере. Пусть задана функция $\sin(x) * \arc\sin(x)/\log_2(x)/(\sqrt{\cos x} + \mathrm{tg}^2 x$, а ее декомпозированный вид представлен в табл. 3. В блок-схему алгоритма на рис. 1 введены условные обозначения ТАБ3$_0$(i), ТАБ3$_I$(i), ТАБ3$_{II}$(i), которые обозначают соответственно первую компоненту (вид функции), вторую и третью компоненты (I и II операнды) i-й строки табл. 3; ТАБ3$'_p$(i) — имя (R_i), которое содержится в четвертой компоненте (результате) i-й строки табл. 3; ТАБ2$_{III}$ — СУШ из табл. 2. Тогда общая система уравнений Шеннона, полученная согласно табл. 2 и 3 по алгоритму, приведенному на рисунке, имеет вид

1. $dz_1 = dz_2 dx;$
2. $dz_2 = -z_1 dx;$ } sin x
3. $dz_3 = z_4 dx;$
4. $dz_4 = -z_6 dz_5;$
5. $dz_5 = \frac{1}{2} z_8 dz_7;$
6. $dz_6 = 2z_4 dz_4;$
7. $dz_7 = dz_{11} - dz_{12}$ } arc sin x
8. $dz_8 = -z_{10} dz_5;$
9. $dz_9 = 2z_{11} dz_{11};$
10. $dz_{10} = 0;$
11. $dz_{11} = 2z_{12} dx;$
12. $dz_{12} = dx;$
13. $z_{10}(0) = 1;$
14. $dz_{13} = z_1 dz_3 + z_3 dz_1,$ } sin (x)× ×arc sin (x)

$$\left.\begin{aligned}&15.\ dz_{14} = (\log_2 1)\times \\ &\quad \times z_{15}dx; \\ &16.\ dz_{15} = -z_{16}dx; \\ &17.\ dz_{16} = 2z_{15}dz_{15};\end{aligned}\right\} \log_2 x$$

$$\left.\begin{aligned}&18.\ dz_{17} = z_{14}dz_{18} + \\ &\quad + z_{18}dz_{14}; \\ &19.\ dz_{18} = -z_{19}dz_{15}; \\ &20.\ dz_{19} = 2z_{18}dz_{18};\end{aligned}\right\} \begin{aligned}&(\sin(x)\times \\ &\times \arcsin(x))/ \\ &/\log_2(x) = A\end{aligned}$$

$$\left.\begin{aligned}&21.\ dz_{20} = -z_{21}dx; \\ &22.\ dz_{21} = z_{20}dx;\end{aligned}\right\} \cos x$$

$$\left.\begin{aligned}&23.\ dz_{22} = \frac{1}{2}z_{23}dz_{20}; \\ &24.\ dz_{23} = -z_{24}dz_{22}; \\ &25.\ dz_{24} = 2z_{23}dz_{23};\end{aligned}\right\} \sqrt{\cos x}$$

$$\left.\begin{aligned}&26.\ dz_{25} = z_{26}dx; \\ &27.\ dz_{26} = -z_{27}dz_{28}; \\ &28.\ dz_{27} = 2z_{28}dz_{26}; \\ &29.\ dz_{28} = 2z_{29}dz_{30}; \\ &30.\ dz_{29} = dz_{30}; \\ &31.\ dz_{30} = -z_{31}dx; \\ &32.\ dz_{31} = z_{30}dx;\end{aligned}\right\} \operatorname{tg} x$$

$$\left.\begin{aligned}&33.\ dz_{32} = 2z_{33}dz_{25}; \\ &34.\ dz_{33} = dz_{25};\end{aligned}\right\} \operatorname{tg}^2 x$$

$$\left.\begin{aligned}&35.\ dz_{34} = dz_{22} + \\ &\quad + dz_{32};\end{aligned}\right\} \begin{aligned}&\sqrt{\cos x} + \\ &+ \operatorname{tg}^2 x = B\end{aligned}$$

$$\left.\begin{aligned}&36.\ dz_{35} = z_{16}dz_{36} + \\ &\quad + z_{36}dz_{16}; \\ &37.\ dz_{36} = -z_{37}dz_{34}; \\ &38.\ dz_{37} = 2z_{36}dz_{36}.\end{aligned}\right\} A/B$$

В результате автоматического преобразования произвольных функций, записанных на языке программирования высокого уровня в СУШ, повышается уровень использования таких средств вычислительной техники, как цифровые интегрирующие машины и структуры.

Список литературы: 1. *Каляев А. В.* Теория цифровых интегрирующих машин и структур. М., Сов. радио, 1970. 471 с. 2. *Грис Д.* Конструирование компиляторов для цифровых вычислительных машин. М., Мир, 1975. 544 с. 3. *Вайнгартен Ф.* Трансляция языков программирования. М., Мир, 1977. 190 с.

УДК 681.142.3

Г. Г. ЧЕТВЕРИКОВ, А. Г. ЖИРОВ

ПРОГРАММНЫЙ СПОСОБ МАСШТАБИРОВАНИЯ ИНТЕГРИРУЮЩИХ СТРУКТУР

Для повышения эффективности использования однородных цифровых интегрирующих структур (ОЦИС) необходимы высокоразвитые средства математического обеспечения, позволяющие полностью автоматизировать процесс программирования.

Полная автоматическая подготовка программ для ОЦИС возможна при помощи ЦВМ. Идея использования ЦВМ в качестве автомата подготовки программ для вычислительных структур аналогового типа впервые была высказана Олинджером еще в 1960 г. и нашла свое воплощение в системе ANATRAN [1]. Согласно работе [2] при решении задач математического моделирования некоторых функций человеческого зрения возникает проблема масштабирования. Рассмотрим программный способ реализации процедуры масштабирования ОЦИС, к которому привели следующие рассуждения.

Вычислительный процесс на ОЦИС для каждой i-й задачи осуществляется, как правило, в рамках одной из основных стратегий:

I. С заданной погрешностью и в заданное время $t_{iз}$ $\delta_i \leqslant \delta_{iз}$; $t_i \leqslant t_{iз}$, где $i = \overline{1,\ n}$.

II. Погрешность решения должна быть минимальной при заданном времени решения: $\min \delta_i$; $t_i \leqslant t_{iз}$.

III. Время решения должно быть минимальным, а погрешность решения не более заданной: $\min t_i$; $\delta_i \leqslant \delta_{iз}$.

Ниже приведены в качестве примера соотношения времени решения задачи и точности представления функции для ЦИМ с быстродействием 50 итераций в секунду при $n = 20$, где n — количество разрядов в Y-регистре интегратора, для решения задачи в пределах 90 единиц независимой переменной [3]:

δ	2	3	4	5	6
T	0,66	4	60	960	3840

(δ — точность представления функции в десятичных знаках; T — машинное время решения задачи, мин).

Определим шаг интегрирования и масштаб приращений независимой переменной, исходя из требования представления функции с точностью в три десятичных знака. Для этого вес каждого приращения функции должен быть не более 10^{-3}, или для двоичной системы счисления $-2^{-10} < 10^{-3}$, а масштаб приращений соответственно $m_{dy} =$ (он связан с масштабом данной величины через основание системы счисления $M = 2^m$). Из соотношения $m_{dy} = m_{dt} + 3$ (полученного предварительно) определяем масштаб поступления независимой переменной $m_{dt} = 7$. Шаг интегрирования составляет величину $\Delta\tau = 2^{-7}$. При таком шаге интегрирования и оговоренных выше условиях необходимо $2^{-7} \cdot 90$ итераций или 4 мин машинного времени. Аналогичным образом получены и остальные соотношения.

При невозможности удовлетворить требованиям заказчика в рамках заданных стратегий на данной вычислительной машине следует обратиться к вычислительной машине с другим быстродействием и числом разрядов в Y-регистре интеграторов.

Основными шагами процедуры масштабирования ОЦИС программным способом можно назвать следующие:

1. Описание решающей структуры на языке внутренней интерпретации ОЦИС получаем в результате трансляции исходной программы, представленной на входном языке описания объекта моделирования. В качестве внутреннего языка в данном случае используется язык, описанный в работе [4].

2. Формальное описание решающей структуры помещается в таблицу, которая используется в процессе расчета масштабных коэффициентов, а также при построении и коммутации схемы решения.

3. Определяются соотношения масштабов отдельных интеграторов и их зависимость от масштаба независимой переменной, выбранной в рамках одной из стратегий решений (I—III).

Здесь же, определив масштабы входов и выходов интегратора (m_{dx}, m_{dy} и m_{ds}), находим количество используемых разрядов n'.

Одновременно начальные значения содержимого Y-регистров и постоянные множители переводятся в двоичную систему и записываются с учетом количества используемых разрядов.

Заметим, что наибольшие значения переменных ориентировочно определяются из следующего условия: $a^{-m_y-1} < y_{\max} < a^{-m_y}$, где a — основание системы счисления.

Известно [3], что каждому цифровому интегратору соответствуют масштабные соотношения

$$m_{dx} + m_y - m_{ds} = 0; \; n' = m_{dy} - m_y. \quad (1)$$

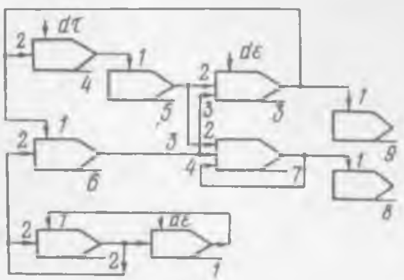


Рис. 1. Схема решения уравнений Бесселя.

Они положены в основу расчета масштабов при программном способе масштабирования. Реализацию данного способа рассмотрим на примере решения уравнения Бесселя для функции нулевого порядка [3]. Схема решения, приведенная на рис. 1, описывается в результате трансляции на языке внутренней интеритерации в таблице с помощью последовательности указателей. Указатель главного элемента называет элемент, выход которого должен соединяться со входом (входами) других элементов. Эти элементы задаются при помощи указателя вспомогательного элемента.

	1	2	3	4 5
1	0,1———	2—1◡◡		
2	0,2———	1—2◡◡	2—2——	6—2——
3	0,3———	6—1◡◡	4—2——	9—1——
4	0,4———	5—1◡◡		
5	0,5———	3—2◡◡	7—2——	
6	0,6———	7—3◡◡	3—3——	
7	0,7———	7—4◡◡	8—1——	
8	1,			
. . .				.
i	0, i———	l—j——	. . .	
. . .				

Формальное описание решающей структуры, приведенное в таблице, расшифровывается следующим образом.

1-я строка — 0, признак главного элемента, с выхода которого начинается коммутация схемы для данного описания (заметим, что главным элементом может быть выбран произвольно любой элемент в схеме решения из числа тех, на которые поступает независимая переменная). Здесь в качестве главного элемента (для первой строки) выбран интегратор, номер которого 1, поэтому запись первой строки следует читать: выход главного элемента соединить со входом 2 интегратора, номер которого 2.

2-я строка — 0,2, указатель главного элемента, с выхода которого продолжается коммутация схемы решения, т. е. запись 2-й строки: выход интегратора 2 соединить со входом 2 интегратора 1, со входом 2 интегратора 2 и со входом 2 интегратора 6 и т. д.

8-я строка — 1, конец (схема замкнута).

В общем случае i-я строка формального описания имеет вид 0, i □□□ l □ j □□ ... Из приведенного пояснения видна простота составления и анализа формального описания схем решения (см. также таблицу).

Анализ формального описания, приведенного в таблице, показывает, что первый столбец его содержит имя-код $MDS(i)$ на выходах всех интеграторов, последующие столбцы — имя-коды $MDX(l)$ или $MDY(l)$ всех входов интеграторов (за исключением входов по независимой переменной). При этом каждая строка показывает связь имени-кода выхода данного интегратора с именами-кодами входов его или других интеграторов. Перед анализом первой и каждой последующей строки формального описания производится формирование стандартной записи $MDS(i) = MY(i) + MDX(i)$. Далее формируются стандартные записи $MDX(l) = MDS(i)$ или $MDY(l) = MDS(i)$, вид которых определяется значением номера входа j рассматриваемого интегратора, а количество записей — числом столбцов в формальном описании, следующих за первым. Алгоритм анализа формального описания интегрирующих структур для составления соотношений масштабов отдельных интеграторов машинным способом приведен на рис. 2.

В работе используется программа расчета масштабов уравнения Бесселя для функции нулевого порядка, которая может быть условно представлена тремя частями. Первая часть — это область задания всех исходных значений: а) масштабов всех интеграторов ($MY(i)$); б) масштаба поступления независимой переменной (MDT) и указание тех входов интеграторов ($MDX(i)$), на которые она поступает. Вторая часть составляется согласно алгоритму анализа формального описания интегрирующих структур. Третья часть — вывод на печать операторами *WRITE* или *PRINT* результатов счета программы.

Результаты, полученные программой, совпадают с результатами, приведенными в работе [3] при решении данной задачи ручным способом. Таким образом, на примере уравнения Бесселя вполне очевидна воспроизводимость результата расчета масштабов в цифровых интегрирующих структурах программным способом.

Выводы. Предложенный способ позволяет производить решение задачи в рамках одной из стратегий решения I—III. Метод делает возможным автоматизировать практически весь процесс масштабирования. Очевидна простота и удобство данного способа масштабирования. При организации предложенного

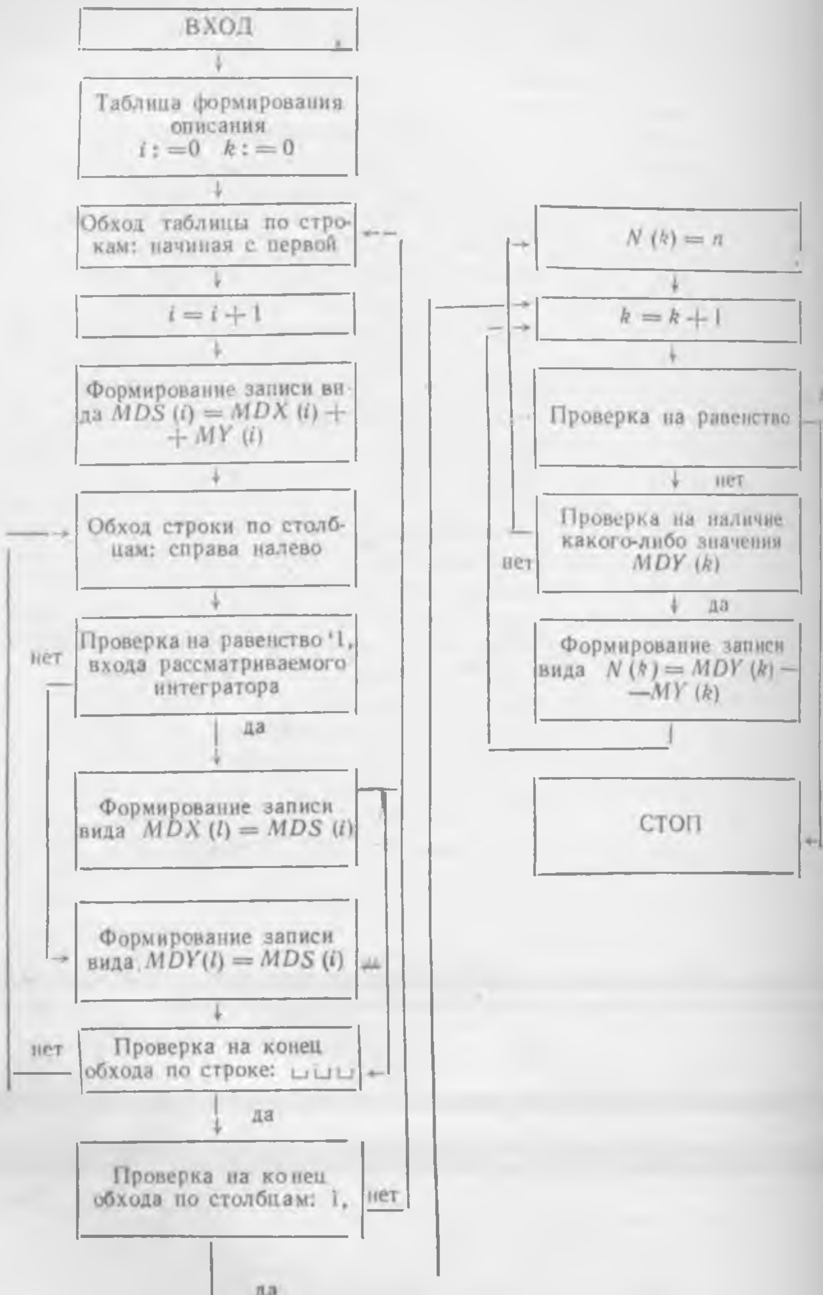


Рис. 2. Блок-схема алгоритма анализа формального описания.

способа вычислений не требуется дополнительная аппаратура. Программа унифицирована таким образом, что задание всех исходных данных не требует набивки большого числа перфокарт (так, для $K < 99$ необходимо не более трех перфокарт, где K — число интеграторов в схеме решения); при этом затраты машинного времени сравнительно невелики (для рассмотренного примера — 96 сек).

В дальнейшем предполагается рассмотреть программный способ масштабирования, для ОЦИС в составе неоднородной вычислительной системы (НВС), а также для более сложных случаев масштабирования (например, когда неизвестны максимальные значения переменных).

Список литературы: 1. *Ohlinger L.* Anotran — first step in greeding the diagnalog.— Proceedings of the Western Joint Computer Conference, 1960, v. 17, p. 11-19. 2. *Шабанов-Кушнаренко Ю. П.* Математическое моделирование некоторых функций человеческого зрения. Автореф. дис. на соиск. учен. степени д-ра техн. наук. Киев, 1965. 45 с. 3. *Неслуховский К. С.* Цифровые дифференциальные анализаторы. М., Машиностроение, 1968. 261 с. 4. *Мурашко А. Г., Сенченко Н. И., Терентьев М. Ф.* Об одном способе формального описания структурных схем на АВМ. — Аналоговая и аналого-цифровая вычислительная техника. М., 1973, вып. 5, с. 253.

УДК 681.3.01.02

С. К. КОЛУБАЙ

ПАРАЛЛЕЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА ДЛЯ МК-ПРОГРАММ

В работе [1] предложен подход к построению внутреннего языка вычислительной системы, функционирующей в соответствии с принципом расщепления программ, который является моделью организации вычислительного процесса человеком. Предложенный язык назван МК-языком, а программы на МК-языке — соответственно МК-программами.

Рассмотрим структуру параллельной вычислительной системы для обработки МК-программ. Ее основной особенностью является применение специальным образом построенной активной памяти с логикой, названной процессоро-памятью. В работе будем пользоваться терминологией и обозначениями, принятыми в статье [1].

Сформулируем требования к вычислительной системе, обрабатывающей МК-программы: 1. Вычислительная система должна обладать возможностями параллельного выполнения инструкций. 2. Время включения готовой к выполнению инструкции должно быть значительно меньше времени выполнения самой инструкции. 3. Время, затрачиваемое на слежение за состоянием множества выполняющихся инструкций, должно быть соизмеримо с временем выполнения отдельной инструкции. 4. Время, затрачиваемое на определение и корректировку множества готовых к выполнению инструкций в процессе выполнения МК-программы, должно быть значительно меньше времени выполнения любой инструкции.

Требования 1—3 в той или иной степени могут быть удовлетворены, например в классической мультипроцессорной вычислительной системе с общим полем памяти [2]. Требованию 4, однако, в рамках традиционных структур МВС удовлетворить невозможно принципиально. Это связано с тем, что для определения множества готовых инструкций и корректировки его в процессе выполнения необходимо одновременно следить за поведением всех инструкций МК-программ, а для решения этой задачи требуется специальная управляющая программа, время работы которой будет весьма значительно. Выходом из этого положения, на наш взгляд, является аппаратный подход к решению перечисленных выше задач, введение специального блока, названного

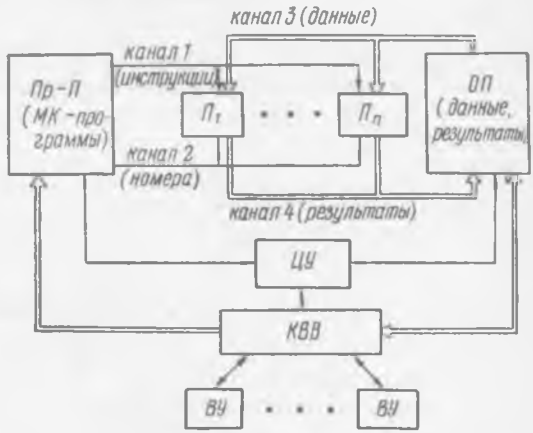


Рис. 1. Параллельная вычислительная система.

нами процессоро-памятью. В отличие от классической оперативной памяти на блок процессоро-памяти возлагаются также задачи по первичной обработке МК-программ.

На рис. 1 приведена структура параллельной вычислительной системы (ПВС) для обработки МК-программ. Она состоит из процессоров $П_1$, $П_2$, ..., $П_n$, оперативной памяти (ОП), процессоро-памяти (Пр-П,) центрального устройства (ЦУ), каналов ввода — вывода (КВВ) и некоторого набора внешних устройств (ВУ). В структуре ПВС реализован принцип разделения памяти на две части: оперативную память классического типа для хранения данных и результатов и процессоро-память (Пр-П) для хранения и обработки МК-программ. Данная ПВС построена как МВС с общим для всех процессоров ($П_1$, $П_2$, ..., $П_n$) полем оперативной памяти и полем процессоро-памяти. Не исключается возможность иметь для каждого из процессоров свою небольшую сверхоперативную память, в которой можно организовывать небольшую очередь инструкций с данными на обработку и очередь результатов вычислений на запись в оперативную память. Кроме того, такая память могла бы служить буфером у процессора и позволила бы значительно упростить задачи, возлагаемые на ЦУ по синхронизации работы отдельных блоков ПВС. Назначение каналов ввода — вывода и внешних устройств стандартное — обеспечение связи ПВС с внешним миром, т. е. ввод данных и программ и вывод результатов.

Рассмотрим одну из возможных моделей организации вычислительного процесса в ПВС и сформулируем задачи, решаемые блоком процессоро-памяти (Пр-П). Пусть в некоторый начальный момент времени в Пр-П введена МК-программа. Блок Пр-П определит множество готовых к выполнению инструкций и, получив от ЦУ разрешение на выполнение, начнет выбирать по некоторому правилу отдельные инструкции на включение. Таким правилом может быть, например, условие равновероятного выбора из множества готовых инструкций или включение инструкций с меньшими (большими) номерами раньше. Для упрощения полагаем, что в ПВС реализуется одномерный вычислительный процесс, а блок Пр-П включает готовые к выполнению инструкции до тех пор, пока их множество не станет пусто. Заметим, однако, что эти ограничения не обязательны.

Таким образом, блок Пр-П, реализуя одномерный вычислительный процесс, с некоторого момента времени начнет включать готовые к выполнению инструкции. Включение инструкции на выполнение состоит в выдаче в канал 1 команды и номера инструкции, причем пусть не указывается, на какой конкретно процессор направляется инструкция. Заметим, что в зависимости от способа организации связи процессоров с блоками Пр-П и ОП (рис. 1) возможны несколько различных структур ПВС. Они будут рассмотрены после описания принципа построения и функционирования блока Пр-П. В соответствии с анализом основных задач, решаемых блоком Пр-П, не является существенным, в какой конкретной структурной организации ПВС работает блок Пр-П.

Пусть инструкция, выданная в канал 1, обязательно принята каким-либо из процессоров, например $П_j$. Процессор $П_j$, приняв инструкцию, обращается к блоку ОП за данными и, получив их, приступает к выполнению операции, предписанной командой инструкции. После выполнения операции, вычислив результат, процессор $П_j$ снова обращается к блоку ОП, но уже для записи результата. Осуществив запись результата в ОП, процессор $П_j$ выдает на блок Пр-П номер i инструкции, которую он выполнил. С этого момента процессор $П_j$ опять готов к приему новой инструкции. Следует считать, что время выдачи в канал 1 готовых к выполнению инструкций значительно меньше, чем время их выполнения, поэтому описанный процесс будет протекать на многих процессорах ПВС одновременно. Число одновременно занятых процессоров будет определяться количеством готовых к выполнению инструкций в МК-программе. После выполнения инструкций каждый из процессоров сначала осуществляет запись результата в ОП, а затем выдачу на блок Пр-П номера выполненной инструкции. Из рассмотренной схемы организации вычислительного процесса в рамках структуры ПВС следуют основные функции процессоро-памяти: 1. Хранение МК-программы. 2. Определение для каждого момента времени множества готовых к выполнению инструкций с учетом поступающих от процессоров номеров выполненных инструкций (задача обнуления номеров j, l инструкций). 3. Выдача в канал 1 номеров и команд готовых к выполнению инструкций.

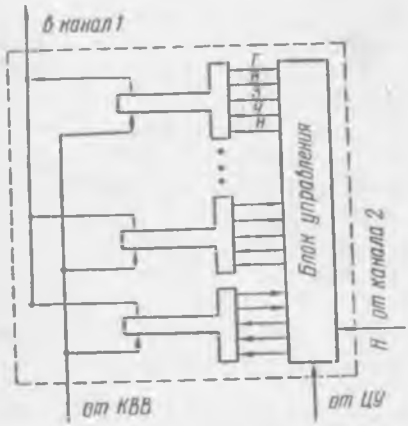


Рис. 2. Блок процессоро-памяти: *а)* функциональная структура; *б)* схематическое изображение ячейки.

Функциональная структура одной из возможных реализаций блока процессоро-памяти, выполняющего перечисленные выше функции, и схематическое изображение ячейки Пр-П приведены на рис. 2.

Блок процессоро-памяти состоит из ячеек процессоро-памяти и блока управления (БУ). Ячейка Пр-П осуществляет хранение одной инструкции МК-программы. Основные функции ячейки состоят в следующем: ячейка Пр-П вырабатывает на БУ два сигнала состояния ячейки — сигнал готовности (Г) и сигнал выпол-

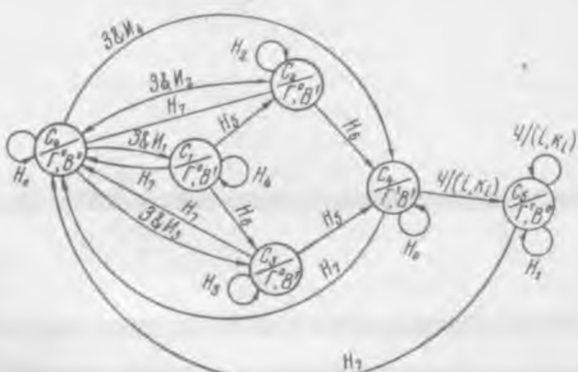


Рис. 3. Граф переходов ячейки Пр-П.

нения (В), осуществляет прием новой инструкции от КВВ на хранение по сигналу запись (З), выдает в канал 1 номер и команду инструкции по сигналу «чтение» (Ч), корректирует состояния номеров j, l в зависимости от номера, поступающего по шине (Н) канала 2. Анализ функционирования ячейки Пр-П произведем с помощью графа (рис. 3), рассматривая ячейку Пр-П как конечный автомат с памятью. При построении графа ячейки Пр-П приняты обозначения, указанные в табл. 1—4.

Таблица 1

Состояние ячейки Пр-П	
Наименование	Обозначение
Свободна	C_0
Неготова 1	C_1
Неготова 2	C_2
Неготова 3	C_3
Готова	C_4
Выполняюсь	C_5

Таблица 2

Входные сигналы от КВВ	
Вид инструкции	Обозначение
$i)\ k_i(j,\ l)$	$И_1$
$i)\ k_i(0,\ l)$	$И_2$
$i)\ k_i(j,\ 0)$	$И_3$
$i)\ k_i(0,\ 0)$	$И_4$

Таблица 3

Входные сигналы по шине Н	
Значение сигнала	Обозначение
Любой, но не равен *	$Н_0$
Любой, но не равен i	$Н_1$
$Н_1$ и не равен l	$Н_2$
$Н_1$ и не равен j	$Н_3$
$Н_2$ и $Н_3$	$Н_4$
Равен j	$Н_5$
Равен l	$Н_6$
Равен i	$Н_7$

Таблица 4

Выходные сигналы Г и В	
Значение сигнала	Обозначение
$Г = 0$	$Г^0$
$Г = 1$	$Г^1$
$В = 0$	$В^0$
$В = 1$	$В^1$

Предположим, что в начальный момент времени ячейка Пр-П находится в состоянии «свободна» (C_0). При поступлении на ячейку инструкции от КВВ совместно с сигналом «запись» (З), ячейка может перейти в одно из четырех состояний в зависимости от вида инструкции $(i)\ k_i(j,\ l)$. Вид инструкции $И_1$, $И_2$, $И_3$ и $И_4$ определяется, как это видно из табл. 2, состоянием номеров j, l.

Пусть инструкция, поступившая на ячейку Пр-П, имеет вид $И_1$. В этом случае ячейка перейдет в состояние «неготова 1» (C_1) и будет оставаться в нем, пока по шине Н не поступит сигнал $Н_5$ или $Н_6$. При поступлении сигнала $Н_5$ инструкция примет вид $И_2$, а ячейка перейдет в состояние «неготова 2» (C_2). Из этого состояния ячейка может перейти в состояние «готова» (C_4) по сигналу $Н_6$, а инструкция примет вид $И_4$. После того как ячейка Пр-П перешла в состояние «готова», команда хранимой в ней инструкции может выполняться. Для этого ее необходимо выдать в канал 1 на процессоры, что осуществляется по сигналу «чтение» (Ч) от блока управления. По этому сигналу в канал 1 выдается номер (i) и команда инструкции (k_i), ячейка Пр-П переходит в состояние «выполняюсь» (C_5), а инструкция сохраняет вид $И_4$. Из состояния «выполняюсь» ячейка Пр-П может перейти только в состояние «свободна» при поступлении по шине Н сигнала $Н_7$.

При этом ячейка Пр-П опять может принять новую инструкцию от КВВ. Блок управления Пр-П синхронизирует работу ячеек. Путем анализа поступающих на него от каждой из ячеек сигналов готовности и выполняемости в соответствии с поступающими на него от ЦУ сигналами блок управления выдает на каждую ячейку сигналы записи и чтения. Например, реализуя одномерный вычислительный процесс, блок управления выдает последовательно на все ячейки Пр-П, где хранятся готовые к выполнению инструкции, сигналы чтения. Кроме этого, блок управления осуществляет одновременную передачу на все ячейки Пр-П значений номеров выполненных инструкций, поступающих по каналу 2.

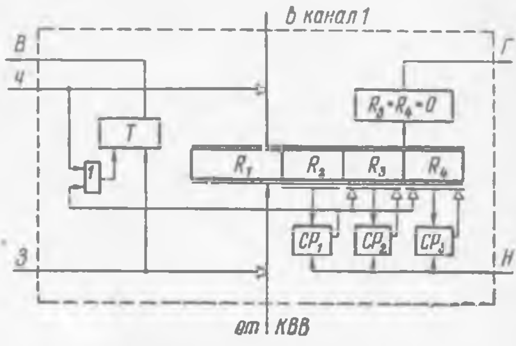


Рис. 4. Операционная схема ячейки Пр-П.

Таким образом, описанный подход приводит к построению блока процессоро-памяти, полностью удовлетворяющего сформулированным ранее требованиям.

Рассмотрим одну из возможных реализаций ячейки Пр-П. На рис. 4 показана одна из возможных операционных схем ячейки Пр-П, состоящая из регистров R_1, R_2, R_3 и R_4, триггера T, схем сравнения CP_1, CP_2 и CP_3, осведомительной схемы $R_3 = R_4 = 0$ и управляемых шин. Регистр R_1 предназначен для хранения команды инструкции k_i, регистр R_2 — для хранения номера инструкции i, а регистры R_3 и R_4 — для хранения номеров j, l. Осведомительная схема $R_3 = R_4 = 0$ формирует сигнал $\Gamma = 1$ только тогда, когда содержимое регистров R_3 и R_4 равно нулю. Схемы сравнения CP_1, CP_2 и CP_3 однотипны, каждая из них вырабатывает на выходе установочный сигнал в том и только в том случае, если совпадают значения номеров, поступающих по шине Н, с номерами, хранящимися в регистре R_1 для схемы CP_1 и, соответственно, в регистрах R_2 и R_5 для схем CP_2 и CP_3. Установочные сигналы, вырабатываемые схемами CP_2 и CP_3, устанавливают в нулевое состояние, соответственно, регистры R_3 и R_4. Установочный сигнал со схемы CP_1 заносит в каждый из регистров R_3 и R_4 некоторый не используемый в МК-программе номер, который обозначен (*) «звездочкой» (см. табл. 3). Триггер T ячейки Пр-П служит для формирования сигнала В выполнения. В начальный момент времени ячейка Пр-П содержит в регистрах R_3 и R_4 значение*, и триггер T находится в нулевом состоянии. Ячейка Пр-П находится в состоянии «свободна» и формирует сигналы: В = 0; Г = 0. При этом поступление по шине Н любого возможного номера $Н_0$ не изменяет ее состояния.

Пусть по шине от КВВ поступит инструкция $И_1$ (табл. 2) и сигнал записи 3 от блока управления. В этом случае ячейка Пр-П перейдет в состояние «неготова 1», а инструкция $И_1$ запомнится на регистрах ячейки. Триггер T установится в единичное состояние, и на его выходе будет сформирован сигнал В = 1. Сигнал Г при этом останется равным нулю, так как содержимым регистров R_3 и R_4 являются некоторые номера, не равные нулю. Из состояния «неготова 1» ячейка Пр-П может переходить в состояние «неготова 2» или «неготова 3» в зависимости от того, какой номер ($Н_5$ или $Н_6$) раньше поступит по шине Н. После поступления номеров $Н_5$ и $Н_6$, значения которых совпадают с содержимым регистров R_3 и R_4, последние устанавливаются схемами CP_2 и CP_3 в нулевое состояние. На выходе осведомительной схемы формируется сигнал Г = 1, а ячейка Пр-П переходит в состояние «готова».

Поступление сигнала чтения Ч сопровождается выдачей в канал 1 содержимого регистров R_1 и R_2, где хранятся команда и номер инструкции. В этом случае ячейка Пр-П переходит в состояние «выполняюсь». Из любого состояния $C_1, C_2, \ldots, C_5$ ячейка Пр-П может быть возвращена в состояние «свободна» при поступлении по шине Н номера $Н_7$, совпадающего с содержимым регистра R_2.

Рассмотренная операционная схема ячейки Пр-П позволяет сделать вывод о возможности технической реализации блока процессоро-памяти. Существенное значение при этом имеет регулярность структуры блока Пр-П, что позволяет представить блок процессоро-памяти в виде большой интегральной схемы (например, размером $10^4 \div 10^5$ ячеек).

Блок управления Пр-П может быть собран в виде отдельного субблока в микроинтегральном исполнении. Рассмотрев принципы функционирования ПВС и блока Пр-П, отметим основные задачи, решаемые блоком управления, — формирование сигналов чтения и сигналов записи, учет количества свободных ячеек, передачу на ячейки Пр-П номеров выполненных инструкций.

Схемная реализация узлов блока управления, решающих перечисленные выше задачи, хорошо известна из техники дискретного счета. Существующие принципы построения устройств управления в сочетании с принципами микропрограммного и аппаратного управления позволяют осуществить надежную организацию процессов чтения и записи, а также учета свободных ячеек блока процессоро-памяти. При организации записи в любую свободную ячейку и чтения из любой готовой ячейки блок Пр-П не будет содержать такого классического узла как дешифратор адреса ячейки.

Таким образом, рассмотренная специальная память на конечных автоматах, названная процессоро-памятью, позволяет совместно с МК-языком, соответствующим выбранной структуре ячейки Пр-П, организовать вычислительный процесс по принципу расщепления программ.

Список литературы: 1. *Колубай С. К.* Об одном подходе к построению внутреннего языка вычислительной системы. — Проблемы бионики. Харьков, 1979, вып. 22, с. 31—37. 2. Мультипроцессорные системы и параллельные вычисления/Под. ред. Ф. Г. Энслоу. М., Мир, 1976. 383 с.

УДК 62.506.2

Ю. П. ШАБАНОВ-КУШНАРЕНКО, д-р техн. наук

МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ УНАРНЫХ КОНЕЧНЫХ ПРЕДИКАТОВ ВЫСШИХ ПОРЯДКОВ

В статье средствами алгебры предикатов [1—5] описываются конечные предикаты второго и более высоких порядков. Пусть E — конечное множество (алфавит), содержащее в своем составе буквы $a_1, a_2, \ldots, a_p$; p — число всех букв алфавита E. Введем переменную X, называемую *переменным множеством*, областью изменения которой служит система всех подмножеств алфавита E. Каждому подмножеству A алфавита E можно поставить во взаимно однозначное соответствие булев вектор $(\alpha_1, \alpha_2, \ldots, \alpha_p)$ с помощью следующего правила: если буква a_i алфавита E принадлежит множеству A, то полагаем $\alpha_i = 1$, если не принадлежит, то полагаем $\alpha_i = 0$.

Формально векторную запись множества A определяем так:

$$A = (\alpha_1, \alpha_2, \ldots, \alpha_p), \tag{1}$$

где

$$\alpha_i = a_i \in A, \ (1 \leqslant i \leqslant p). \tag{2}$$

Иначе говоря,

$$A = (a_1 \in A, \ a_2 \in A, \ldots, a_p \in A). \tag{3}$$

Переменная X может быть записана в векторном виде $X = (\xi_1, \xi_2, \ldots, \xi_p)$, где $\xi_1, \xi_2, \ldots, \xi_p$ — булевы переменные. Подставляя вместо этих переменных булевы константы 0 или 1, получаем булев вектор, соответствующий тому или иному конкретному подмножеству алфавита E.

Переменное множество $X = (\xi_1, \xi_2, \ldots, \xi_p)$ может быть представлено средствами алгебры конечных предикатов в виде формулы

$$x \in X \underset{d}{\equiv} (\xi_1^1 x^{a_1} \vee \xi_2^1 x^{a_2} \vee \ldots \vee \xi_p^1 x^{a_p})(\xi_1^0 \vee \xi_1^1)(\xi_2^0 \vee \xi_2^1) \ldots (\xi_p^0 \vee \xi_p^1). \tag{4}$$

Здесь $\xi_1, \xi_2, \ldots, \xi_p$ — буквенные переменные, область изменения которых ограничена значениями 0 и 1. Выражение (4) можно рассматривать как запись переменного одноместного предиката первого порядка, зависящего от переменной x. Для формулы (4) будем использовать также сокращенную форму записи:

$$x \in X \underset{d}{\equiv} \xi_1 x^{a_1} \vee \xi_2 x^{a_2} \vee \ldots \vee \xi_p x^{a_p}, \tag{5}$$

условно рассматривая символы $\xi_1, \xi_2, \ldots, \xi_p$ как булевы переменные. Подставляя в формулу (5) конкретные значения булевых переменных $\xi_1 = \alpha_1, \xi_2 = \alpha_2, \ldots, \xi_p = \alpha_p$, получаем формальную запись фиксированного множества $A = (\alpha_1, \alpha_2, \ldots, \alpha_p)$:

$$x \in A \underset{d}{\equiv} \alpha_1 x^{a_1} \vee \alpha_2 x^{a_2} \vee \ldots \vee \alpha_p x^{a_p}. \tag{6}$$

Например, пусть векторная запись множества A имеет вид $A = (0, 1, 0, 1, 1, 0, 0, \ldots, 0)$, тогда $\alpha_1 = \alpha_3 = \alpha_6 = \alpha_7 = \ldots = \alpha_p = 0$, $\alpha_2 = \alpha_4 = \alpha_5 = 1$, следовательно, $x \in A \equiv 0 \cdot x^{a_1} \vee 1 \times x^{a_2} \vee 0 \cdot x^{a_3} \vee 1 \cdot x^{a_4} \vee 1 \cdot x^{a_5} \vee 0 \cdot x^{a_6} \vee 0 \cdot x^{a_7} \vee \ldots \vee 0 \cdot x^{a_p} \equiv x^{a_2} \vee x^{a_4} \vee x^{a_5}$. Аналогично может быть совершен обратный переход от формульного представления множества букв к его векторному представлению.

Дополнение, объединение, пересечение, разность и симметрическая разность множеств $A = (\alpha_1, \alpha_2, \ldots, \alpha_p)$ и $B = (\beta_1, \beta_2, \ldots, \beta_p)$ могут быть записаны в векторной форме:

$$\overline{A} = (\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_p); \tag{7}$$

$$A \cup B = (\alpha_1 \vee \beta_1, \alpha_2 \vee \beta_2, \ldots, \alpha_p \vee \beta_p); \tag{8}$$

$$A \cap B = (\alpha_1 \wedge \beta_1, \alpha_2 \wedge \beta_2, \ldots, \alpha_p \wedge \beta_p); \tag{9}$$

$$A \setminus B = (\alpha_1 \overline{\beta}_1, \alpha_2 \overline{\beta}_2, \ldots, \alpha_p \overline{\beta}_p); \tag{10}$$

$$A \dot{-} B = (\alpha_1 \oplus \beta_1, \alpha_2 \oplus \beta_2, \ldots, \alpha_p \oplus \beta_p). \tag{11}$$

Отношение равенства и отношение включения множеств букв $X = (\xi_1, \xi_2, \ldots, \xi_p)$ и $Y = (\eta_1, \eta_2, \ldots, \eta_p)$ могут быть записаны в виде формул алгебры конечных предикатов:

$$X = Y \underset{d}{\equiv} (\xi_1^1 \sim \eta_1^1)(\xi_2^1 \sim \eta_2^1) \ldots (\xi_p^1 = \eta_p^1); \tag{12}$$

$$X \subseteq Y \underset{d}{\equiv} (\xi_1^1 \supset \eta_1^1)(\xi_2^1 \supset \eta_2^1) \ldots (\xi_p^1 \supset \eta_p^1). \tag{13}$$

Здесь символы ξ_i и η_i $(1 \leqslant i \leqslant p)$ обозначают буквенные переменные. Если же эти символы условно рассматривать как булевы переменные, то зависимости (12) и (13) могут быть представлены в более компактном виде:

$$(\xi_1, \xi_2, \ldots, \xi_p) = (\eta_1, \eta_2, \ldots, \eta_p) \underset{d}{\equiv} \bigwedge_{i=1}^{p} (\xi_i \sim \eta_i); \tag{14}$$

$$(\xi_1, \xi_2, \ldots, \xi_p) \subseteq (\eta_1, \eta_2, \ldots, \eta_p) \underset{d}{\equiv} \bigvee_{i=1}^{p} (\xi_i \supset \eta_i). \tag{15}$$

Рассмотрим примеры практического применения введенных зависимостей.

Пример 1. Доказать тождество $\widetilde{A \cap B} = \tilde{A} \cup \tilde{B}$, где A и B — произвольные множества букв.

Решение. Записываем множества A и B в векторной форме $A = (\alpha_1, \alpha_2, \ldots, \alpha_p)$, $B = (\beta_1, \beta_2, \ldots, \beta_p)$. Тогда

$$\widetilde{A \cap B} = \tilde{A} \cup \tilde{B} \equiv (\overline{\alpha_1\beta_1}, \overline{\alpha_2 b_2}, \ldots, \overline{\alpha_p\beta_p}) = (\bar{\alpha}_1 \vee \bar{\beta}_1,$$
$$\bar{\alpha}_2 \vee \bar{\beta}_2, \ldots, \bar{\alpha}_p \vee \bar{\beta}_p) \equiv (\overline{\alpha_1\beta_1} \sim \bar{\alpha}_1 \vee \bar{\beta}_1)(\overline{\alpha_2\beta_2} \sim \bar{\alpha}_2 \vee$$
$$\vee \bar{\beta}_2) \ldots (\overline{\alpha_p\beta_p} \sim \bar{\alpha}_p \vee \bar{\beta}_p) \equiv 1.$$

Пример 2. Существуют ли такие множества A, B и C, что $A \cap B \neq \varnothing$, $A \cap C = \varnothing$, $(A \cap B) \setminus C = \varnothing$?

Решение. Согласно условию $A \cap B \neq \varnothing \wedge A \cap C = \varnothing \wedge$ $\wedge (A \cap B) \setminus C = \varnothing$. Полагаем $A = (\alpha_1, \alpha_2, \ldots, \alpha_p)$, $B = (\beta_1,$ $\beta_2, \ldots, \beta_p)$, $C = (\gamma_1, \gamma_2, \ldots, \gamma_p)$. Имеем

$$\overline{\alpha_1\beta_1 = 0 \wedge \alpha_2\beta_2 = 0 \wedge \ldots \wedge \alpha_p\beta_p = 0} \wedge \alpha_1\gamma_1 = 0 \wedge \alpha_2\gamma_2 =$$
$$= 0 \wedge \ldots \wedge \alpha_p\gamma_p = 0 \wedge \alpha_1\beta_1\bar{\gamma}_1 \wedge \alpha_2\beta_2\bar{\gamma}_2 \wedge \ldots \wedge \alpha_p\beta_p\bar{\gamma}_p \equiv$$
$$\equiv (\alpha_1\beta_1 \vee \alpha_2\beta_2 \vee \ldots \vee \alpha_p\beta_p)(\bar{\alpha}_1 \vee \bar{\gamma}_1)(\bar{\alpha}_2 \vee \bar{\gamma}_2) \ldots (\bar{\alpha}_p \vee \bar{\gamma}_p) \times$$
$$\times (\bar{\alpha}_1 \vee \bar{\beta}_1 \vee \gamma_1)(\bar{\alpha}_2 \vee \bar{\beta}_2 \vee \gamma_2) \ldots (\bar{\alpha}_p \vee \bar{\beta}_p \vee \gamma_p) \equiv 0 = 1.$$

Получили противоречие. Следовательно, множества A, B, C с требуемыми свойствами не существуют.

Пример 3. Доказать, что $A \cup B \subseteq C \leftrightarrow A \subseteq C$ и $B \subseteq C$.

Решение. $A \cup B \subseteq C \sim A \subseteq C \wedge B \subseteq C \equiv (\alpha_1\beta_1 \supset \gamma_1)(\alpha_2\beta_2 \supset \gamma_2) \ldots$
$$\ldots (\alpha_p\beta_p \supset \gamma_p) \sim (\alpha_1 \supset \gamma_1)(\alpha_2 \supset \gamma_2) \ldots (\alpha_p \supset \gamma_p)(\beta_1 \supset \gamma_1)(\beta_2 \supset \gamma_2) \ldots$$
$$\ldots (\beta_p \supset \gamma_p) \equiv (\alpha_1\beta_1 \supset \gamma_1)(\alpha_2\beta_2 \supset \gamma_2) \ldots (\alpha_p\beta_p \supset \gamma_p) \sim (\alpha_1\beta_1 \supset \gamma_1)(\alpha_2\beta_2 \supset$$
$$\supset \gamma_2) \ldots (\alpha_p\beta_p \supset \gamma_p) \equiv 1.$$

Прежде чем переходить к следующему примеру, рассмотрим метод решения уравнений алгебры логики с буквенными параметрами. Пусть задано уравнение алгебры логики

$$f(x, c_1, c_2, \ldots, c_n) = 1. \tag{16}$$

Требуется решить это уравнение относительно пе о-однозначное
и выразить ее явно через параметры $c_1, c_2, \ldots, c_n$, чего правила:
вид функции $x = \varphi(c_1, c_2, \ldots, c_n)$ такой, что $f(\varphi(c_1, c_2$ 1, если не
$c_1, c_2, \ldots, c_n) = 1$. Предположим, что функция φ су определяем
тогда $x = \varphi(c_1, c_2, \ldots, c_n) \equiv x \sim \varphi(c_1, c_2, \ldots, c_n) =$
$c_2, \ldots, c_n) \vee x\varphi(c_1, c_2, \ldots, c_n) = 1$. С другой стороны, сс (26)
первому следствию теоремы о разложении, имеем $\bar{x}f(0, c_1,$
$c_n) \vee xf(1, c_1, c_2, \ldots, c_n) = 1$. В силу единственности разлож

$$x = \varphi(c_1, c_2, \ldots, c_n) = \bar{f}(0, c_1, c_2, \ldots, c_n); \qquad (27)$$

$$x = \varphi(c_1, c_2, \ldots, c_n) = f(1, c_1, c_2, \ldots, c_n). \qquad (28)$$

Область изменения значений параметров $c_1, c_2, \ldots, c_n$, для которой существует единственное решение $x = \varphi(c_1, c_2, \ldots, c_n)$, определяется уравнением

$$f(0, c_1, c_2, \ldots, c_n) \oplus f(1, c_1, c_2, \ldots, c_n) = 1. \qquad (19)$$

Пример 4. Задана система уравнений

$$A \setminus X = B; \ X \setminus A = C, \qquad (а)$$

где A, B, C — известные, а X — неизвестное множество. При каких A, B, C система (а) имеет единственное решение?

Решение. Пусть множества A, B, C, X имеют в своей векторной записи компоненты α_i, β_i, γ_i, ξ_i $(1 \leqslant i \leqslant p)$. Тогда условие примера может быть переписано в виде системы уравнений алгебры логики:

$$(\alpha_i \bar{\xi}_i \sim \beta_i)(\xi_i \bar{\alpha}_i \sim \gamma_i) = 1. \qquad (б)$$

Находим область изменения параметров α_i β_i γ_i, в которой существует единственное решение для ξ_i: $(\alpha_i \cdot \bar{0} \sim \beta_i)(0 \cdot \bar{\alpha}_i \sim \gamma_i) \oplus (\alpha_i \cdot \bar{1} \sim \beta_i)(1 \cdot \bar{\alpha}_i \sim$ $\sim \gamma_i) = 1$. После преобразований $(\bar{\beta}_i \vee \alpha_i) \wedge (\bar{\alpha}_i \vee \bar{\gamma}_i)$ или $\beta_i \supset \alpha_i = 1$ и $\alpha_i \gamma_i = 0$. Следовательно, система (а) имеет единственное решение, когда $B \subseteq A$ и $A \cap C = \varnothing$.

Пример 5. Для области изменения параметров A, B, C, найденной в предыдущем примере, найти решение системы (а).

Решение. По формуле (17) находим решение уравнения (б):

$$\xi_i = \overline{(\alpha_i \cdot \bar{0} \sim \beta_i)(0 \cdot \bar{\alpha}_i \sim \gamma_i)} = \alpha_i \bar{\beta}_i \vee \bar{\alpha}_i \beta_i \vee \gamma_i.$$

Полученное решение может быть записано в более экономном виде $\xi_i = \alpha_i \bar{\beta}_i \vee \gamma_i$, — если учесть, что $\beta_i \supset \alpha_i = 1$, т. е. $\bar{\alpha}_i \beta_i = 0$. Таким образом решение системы (а) имеет вид $X = (A \setminus B) \cap C$.

Рассмотрим теперь метод решения уравнений алгебры конечных предикатов с буквенными параметрами. Пусть задано уравнение алгебры конечных предикатов

$$P(x, c_1, c_2, \ldots, c_m) = 1. \qquad (20)$$

Компоненты ξ_i вектора множества $X = (\xi_1, \xi_2, \ldots, \xi_m)$ всех решений уравнения (20) для переменной x могут быть определены по формуле

$$\xi_i = P(a_i, c_1, c_2, \ldots, c_m) \quad (1 \leqslant i \leqslant p). \qquad (21)$$

Пример 6. Решить уравнение $(x^{a_1} \vee x^{a_2})(a_1^{c_1}x^{a_2} \vee x^{c_2}) = 1$.

Решение. Находим компоненты вектора множества всех решений X переменной x:

$$\xi_1 = (a_1^{a_1} \vee a_1^{a_2})(a_1^{c_1}a_1^{a_2} \vee a_1^{c_2}) = a_1^{c_2}; \ \xi_2 = (a_2^{a_1} \vee a_2^{a_2})(a_1^{c_1}a_2^{a_2} \vee a_2^{c_2}) =$$
$$= a_1^{c_1} \vee a_2^{c_2}; \ \xi_3 = \ldots = \xi_p = 0.$$

$X = (a_1^{c_2},\ a_1^{c_1} \vee a_2^{c_2}, 0, \ldots, 0)$. Пусть, к примеру, $c_1 = c_2 = a_1$. Тогда
$$X = (a_1^{a_1},\ a_1^{a_1} \vee a_2^{a_1}, 0, \ldots, 0) = (1, 1, 0, \ldots, 0) = \{a_1, a_2\}.$$

Рассмотрим одноместные предикаты второго порядка. Пусть $X = (\xi_1, \xi_2, \ldots, \xi_p)$ — переменное множеств букв, задаваемое переменным одноместным предикатом первого порядка (4). Предикат второго порядка $P(X)$, аргументом которого является переменное множество X, представляет собой булеву функцию f от p переменных $\xi_1, \xi_2, \ldots, \xi_p$:

$$P(X) = f(\xi_1, \xi_2, \ldots, \xi_p). \tag{22}$$

Каждое уравнение алгебры логики

$$f(\xi_1, \xi_2, \ldots, \xi_p) = 1, \tag{23}$$

где $\xi_1, \xi_2, \ldots, \xi_p$ — буквенные переменные, ограниченные булевыми значениями 0, 1;

$$\xi_i^0 \vee \xi_i^1 = 1 \quad (1 \leqslant i \leqslant p), \tag{24}$$

совместно с уравнением

$$\xi_1^1 x^{a_1} \vee \xi_2^1 x^{a_2} \vee \ \ldots \ \vee \xi_p^1 x^{a_p} = 1 \tag{25}$$

задает некоторую систему Σ подмножеств X алфавита E ($X \in \Sigma$, $x \in X$), являющуюся результатом решения 1 системы уравнений (23), (24) относительно переменных $\xi_1, \xi_2, \ldots, \xi_p$ и 2) уравнения (25) относительно переменной x при найденных значениях $\xi_1, \xi_2, \ldots, \xi_p$.

Пример 7. Дано: $E = \{a_1, a_2, a_3\}$, $p = 3$. Определить систему Σ множеств $X = (\xi_1, \xi_2, \xi_3)$, компоненты ξ_1, ξ_2, ξ_3 векторов которых заданы уравнением $P(X) = \overline{\xi_2^1\xi_1^1 \vee \xi_3^0\xi_1^0}$.

Решение. Уравнения (24) в данном случае запишутся в виде $\xi_1^0 \vee \xi_1^1 = 1$; $\xi_2^0 \vee \xi_2^1 = 1$; $\xi_3^0 \vee \xi_3^1 = 1$. Решая их совместно с заданным в условии уравнением, находим $\Sigma = \{(1, 0, \xi_3), (0, \xi_2, 1)\}$, где ξ_2, ξ_3 пробегают значения 0, 1. Таким образом, вектор (ξ_1, ξ_2, ξ_2) принимает четыре значения из множества (0, 0, 0), (1, 0, 1), (0, 0, 1), (0, 1, 1). Решая при найденных значениях ξ_1, ξ_2, ξ_3 уравнение (25), которое в данном случае принимает вид $\xi_1^1 x^{a_1} \vee \xi_2^1 x^{a_2} \vee \xi_3^1 x^{a_3} = 1$, окончательно получим $x \in X$; $X \in \Sigma$; $\Sigma = \{\{a_1\}, \{a_1, a_3\}, \{a_3\}, \{a_2, a_3\}\}$.

Введем теперь переменную S, называемую переменной системой множеств, областью изменений которой служит множество всех систем всех подмножеств алфавита E. С каждым подмножеством $A = (\alpha_1, \alpha_2, \ldots, \alpha_p)$ алфавита E свяжем некоторый номер J, соответствующий двоичному коду $\alpha_1\alpha_2 \ldots \alpha_p$. Каждой системе подмножеств Σ алфавита E поставим во взаимно-однозначное соответствие булев вектор $(\sigma_1, \sigma_2, \ldots, \sigma_{2^p})$ с помощью следующего правила: если подмножество A_j принадлежит системе Σ, полагаем $\sigma_j = 1$, если не принадлежит — $\sigma_j = 0$. Формально векторную запись системы Σ определяем следующим образом:

$$\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{2^p}), \tag{26}$$

где

$$\sigma_j = A_j \in \Sigma \quad (1 \leqslant i \leqslant 2^p). \tag{27}$$

Иначе говоря, $\Sigma = (A_1 \in \Sigma, A_2 \in \Sigma, \ldots, A_{2^p} \in \Sigma)$. (28)

Переменная S может быть записана в векторном виде $S = (s_1, s_2, \ldots, s_{2^p})$, где $s_1, s_2, \ldots, s_{2^p}$ — булевы переменные. Подставляя вместо этих переменных булевы константы 0 или 1, получаем булев вектор, соответствующий той или иной конкретной системе подмножеств алфавита E.

Переменная система множеств $S = (s_1, s_2, \ldots, s_{2^p})$ может быть представлена средствами алгебры конечных предикатов в виде формулы

$$X \in S \underset{d}{\equiv} \Big(\bigvee_{(\alpha_1, \alpha_2, \ldots, \alpha_p)} s_j^1 \xi_1^{\alpha_1} \xi_2^{\alpha_2} \ldots \xi_p^{\alpha_p}\Big) (s_1^0 \vee s_1^1)(s_2^0 \vee s_2^1) \ldots (s_{2^p}^0 \vee s_{2^p}^1). \tag{29}$$

Здесь $X = (\xi_1, \xi_2, \ldots, \xi_p)$ — переменное множество; $s_1, s_2, \ldots, s_{2^p}$ — буквенные переменные, область изменения которых ограничена значениями 0 и 1. Выражение (29) можно рассматривать как запись переменного одноместного предиката второго порядка, зависящего от переменной X. Для формулы (29) будем также использовать сокращенную форму записи:

$$X \in S \underset{d}{\equiv} \bigvee_{(\alpha_1, \alpha_2, \ldots, \alpha_p)} s_j \xi_1^{\alpha_1} \xi_2^{\alpha_2} \ldots \xi_p^{\alpha_p}, \tag{30}$$

условно рассматривая символы $s_1, s_2, \ldots, s_{2^p}$ как булевы переменные. Заменяя в выражениях (29) и (30) переменные s_j булевыми константами, получаем формальную запись фиксированной системы подмножеств алфавита E. Дополнение, объединение, пересечение, разность и симметрическая разность систем множеств записываются на языке алгебры конечных предикатов так же, как представляются те же операции для множеств букв. Аналогично вводятся отношения равенства и включения систем множеств.

Рассмотрим одноместные предикаты третьего порядка. Пусть $S = (s_1, s_2, \ldots, s_{2^p})$ — переменная система множеств букв, задаваемая переменным одноместным предикатом второго порядка (29). Предикат третьего порядка $Q(S)$, аргументом которого является переменная система множеств S, представляет собой булеву функцию g от 2^p переменных $s_1, s_2, \ldots, s_{2^p}$:

$$Q(S) = g(s_1, s_2, \ldots, s_{2^p}). \tag{31}$$

Каждое уравнение алгебры логики

$$g(s_1, s_2, \ldots, s_{2^p}) = 1, \tag{32}$$

где $s_1, s_2, \ldots, s_{2^p}$ — буквенные переменные, ограниченные булевыми значениями 0, 1,

$$s_j^0 \vee s_j^1 = 1 \ (1 \leqslant j \leqslant 2^p), \tag{33}$$

совместно с уравнениями (30), (24) и (25) задает некоторое множество систем подмножеств алфавита E.

Рассуждая аналогичным образом, можно описать средствами алгебры конечных предикатов одноместные предикаты произвольного конечного порядка.

Список литературы: 1. *Шабанов-Кушнаренко Ю. П.* О теории интеллекта. — Проблемы бионики. Харьков, 1977, вып. 22, с. 15—22. 2. *Шабанов-Кушнаренко Ю. П.* Об алгебре конечных предикатов. — АСУ и приборы автоматики. Харьков, 1977, вып. 52, с. 21—28. 3. *Шабанов-Кушнаренко Ю. П.* Об алгебре предикатов с отрицанием.— АСУ и приборы автоматики. Харьков, 1977, вып. 52, с. 42—49. 4. *Шабанов-Кушнаренко Ю. П.* Об уравнениях теории интеллекта.— АСУ и приборы автоматики. Харьков, 1977, вып. 53, с. 63—70. 5. *Шабанов-Кушнаренко Ю. П.* Математическое описание конечных логических объектов.— Проблемы бионики. Харьков, 1977, вып. 23, с. 11—18.

УДК 519.766.2

М. Ф. БОНДАРЕНКО, канд. техн. наук, *В. М. БОНДАРЕВ*

О МАТЕМАТИЧЕСКОМ ОПИСАНИИ СЛОВОИЗМЕНЕНИЯ СУЩЕСТВИТЕЛЬНЫХ. *СООБЩЕНИЕ 1*

Описание любого фрагмента морфологической системы языка можно дать в виде предиката $L(X, Y, Z)$, где предметной областью X является множество слов (словарных форм), предметной областью Y — множество словоформ, а Z — множество значений грамматических категорий, характеризующих словоформу Y [1]. Предикат $L(X, Y, Z)$ принимает значение «истина», если значения переменных не являются взаимоисключающими с точки зрения нормы языка. В противном случае значение предиката — «ложь». Обозначим «истину» и «ложь» 1 и 0 соответственно, а сам предикат назовем морфологической функцией [1].

Если для морфологической функции существует аналитическое выражение, то разнообразные задачи морфологической обработки, в том числе синтез, анализ, нормализацию, можно представить уравнениями типа $L(X, Y, Z) = 1$, где, в зависимости от конкретной задачи, значения тех или иных переменных известны. Решение всякой задачи сведется при этом к поиску корней соответствующего уравнения.

Цель настоящей работы — предложить возможный способ описания предиката $L(X, Y, Z)$ и проиллюстрировать этот способ формализацией небольшого фрагмента морфологии русского языка.

В традиционном описании морфологии важную роль играют такие понятия, как основа слова, его окончание, тип склонения, чередование в основе и т. п. Всю совокупность таких понятий условно назовем морфологической характеристикой и обозначим Г. Формально Г — это набор переменных, каждая из которых принимает конечное число значений.

Из практики языка следует, что всякому совместимому набору значений переменных X, Y, Z можно поставить в соответствие хотя бы один совместимый с ним набор значений переменных Г. Поэтому можно задать предикат $L'(X, Y, Z, \Gamma)$, принимающий значение 1, когда все переменные X, Y, Z и Г совместимы, и 0 — в противном случае. Поскольку решение уравнений есть поиск таких значений неизвестных величин, которые были бы совместимы со значениями величин известных, ясно, что, приняв заданными некоторые переменные из X, Y, Z и решив уравнение $L'(X, Y, Z, \Gamma) = 1$, отыщем значения неизвестных переменных из набора X, Y, Z, так как если бы решали уравнение $L(X, Y, Z) = 1$. Другими словами, имея аналитическое выражение для морфологической функции L', можно решить все те задачи, которые решили бы, имея выражение для L.

Целесообразно формализовать предикат L', а не L, так как в этом случае можно полнее использовать существующие описания морфологии, например грамматику русского языка.

Заранее оговорим, что выражение для L' будем искать в виде конъюнкции предикатов L_1, L_2, ..., L_k от тех же переменных: X, Z, Г. Предпосылку к такому решению видим в том, что морфология обычно описывается набором параграфов или правил, совместное выполнение которых обеспечивает правильность грамматической обработки.

Процесс формализации морфологии представляется нам в виде накопления формул, выражающих грамматические правила, записанные в форме предикатов L_1, L_2, ..., L_k. Критерием качества математического описания может быть статистический эксперимент, показывающий, насколько удовлетворяют норме языка решения морфологических задач, полученные с помощью модели вида $L_1 \wedge L_2 \wedge \ldots \wedge L_k$.

Хотя формально всякий предикат L_i ($i = 1, 2, \ldots, k$) зависит от всех своих переменных, его фактическими аргументами является лишь часть их, для каждого L_i — своя. Остальные переменные можно считать связанными кванторами общности, которые в записи формул опускаем.

Чтобы придать содержательность нашим построениям, рассмотрим конкретный пример грамматического описания, а именно, описание склонения существительных среднего рода, оканчивающихся на *-е*, таких как *море, поле, солнце.* Пример этот достаточно прост и может быть разобран в рамках настоящей статьи, в то же время он отражает многие особенности словоизменения существительных. Заранее условимся, что слова и словоформы будут записаны без знака ударения. Исключим из рассмотрения существительные, склоняющиеся по типу прилагательных, например *животное*, а также те слова с дефисом, у которых склоняются обе части.

Поскольку нас интересует письменная форма языка, X и Y удобно представлять в виде наборов переменных $X = \langle x_1, x_2, \ldots, x_n \rangle$, $Y = \langle y_1, y_2, \ldots, y_n \rangle$. Каждая переменная из этих наборов соответствует отдельной позиции в записи слова или словоформы, значением переменной является буква или другой символ, стоящий в этой позиции. Число n должно быть выбрано с таким расчетом, чтобы вместить слово или словоформу максимальной длины.

Будем считать, что переменная Z также представляет собой набор $\langle z_1, z_2 \rangle$, где z_1 — грамматическая категория числа, а z_2 — категория падежа. Понятно, что при описании другого фрагмента морфологии этот набор может быть иным, в частности более полным. В качестве морфологической характеристики выберем набор переменных $\Gamma = \langle \alpha, \beta, \gamma, \eta, \omega \rangle$. Здесь α — основа слова; β — чередование в основе словоформы; γ — тип склонения; η — основа словоформы; ω — окончание словоформы.

Так как у нас имеется фиксированное число переменных $x_1, x_2, \ldots, x_n$, а длина слов различна, создается противоречие, разрешить которое можно, дополняя каждое слово особыми символами до стандартной длины. Будем использовать для этой цели символ пробела $\sqcup$, а само слово размещать в крайних левых позициях. Так, если $X =$ *поле*, $n = 6$, то $x_1 =$ п, $x_2 =$ о, $x_3 =$ л, $x_4 =$ е, $x_5 = \sqcup$, $x_6 = \sqcup$. То же касается и переменных y_i ($i = 1, 2, \ldots, n$).

Таким образом, область определения каждой из упомянутых переменных состоит из букв русского алфавита, дефиса и знака пробела — всего 35 символов. Это дает нам основание записать предикаты

$$x_i^{\text{а}} \vee x_i^{\text{б}} \vee \ \ldots \ \vee x_i^{\text{я}} \vee x_i^{-} \vee x_i^{\sqcup} \quad (i = 1, 2, \ldots, n); \qquad (1)$$

$$y_i^{\text{а}} \vee y_i^{\text{б}} \vee \ \ldots \ \vee y_i^{\text{я}} \vee y_i^{-} \vee y_i^{\sqcup} \quad (i = 1, 2, \ldots, n), \qquad (2)$$

которые с помощью символики, принятой в работах [1; 2], выражают тот факт, что значением переменных x_i и y_i могут быть лишь определенные символы.

Области определения переменных z_1 и z_2 состоят соответственно из двух значений числа {е, м} и шести значений падежа {и, р, д, в, т, п};

$$z_1^{\text{е}} \vee z_1^{\text{м}}; \qquad (3)$$

$$z_2^{\text{и}} \vee z_2^{\text{р}} \vee z_2^{\text{д}} \vee z_2^{\text{в}} \vee z_2^{\text{т}} \vee z_2^{\text{п}}. \qquad (4)$$

В нашем конкретном примере под основой слова будем понимать часть слова, оставшуюся после удаления последней буквы *е*, если слово относится к разряду склоняемых, и слово целиком, если оно не склоняется. Поскольку α — внутренняя переменная, т. е. не может быть ни входной ни выходной при решении задач синтеза, анализа, нормализации, в целях упрощения описания обозначим всякую основу просто числом, так называемым номером основы, при этом разные основы будут иметь различные номера. Областью определения переменной α будем считать множество чисел $\{1, 2, \ldots, t\}$, где t — общее число основ рассматриваемого класса слов,

$$\alpha^1 \vee \alpha^2 \vee \ldots \vee \alpha^t. \tag{5}$$

То же самое можно сказать о переменной ω — окончании словоформы. В отличие от α число ее значений вполне обозримо и составляет 18 элементов:

$$\omega^1 \vee \omega^2 \vee \ldots \vee \omega^{18}. \tag{6}$$

Раскрывая смысл переменной β, заметим, что в рамках рассматриваемого примера имеют место лишь двуступенчатые чередования [3], например: *поленце — поленец*, *ущелье — ущелий*. Отсюда следует, что область определения переменной β можно ограничить двумя символами $\{1, 2\}$, где 1 будет означать первую ступень чередования, а 2 — вторую,

$$\beta^1 \vee \beta^2. \tag{7}$$

Переменная γ представляет тип склонения существительного, который будем понимать так же, как в работе [4].

Оттуда же будем черпать всю необходимую лингвистическую информацию. Хотя в [4] отмечается 9 возможных типов склонения существительных, для нашего примера актуальны лишь 6 из них: 0 — несклоняемые существительные; 2 — существительные стандартного мягкого склонения; 4 — существительные с основой на шипящую; 5 — существительные с основой на *ц*; 6 — с основой на гласную (кроме *и*), *ь* или *й*; 7 — с основой на букву *и*. Множество 0, 2, 4, 5, 6, 7 будем считать областью изменения переменной γ:

$$\gamma^0 \vee \gamma^2 \vee \gamma^4 \vee \gamma^5 \vee \gamma^6 \vee \gamma^7. \tag{8}$$

Основой словоформы η будем считать часть словоформы без окончания, дополненную пробелами до стандартной длины. В отличие от основы слова η представим не числом, а набором переменных $(\eta_1, \eta_2, \ldots, \eta_n)$ подобно тому, как представлены слово X и форма Y. Область изменения переменных y_i $(i = 1, 2, \ldots, n)$ такая же, как переменных x_i и y_i,

$$\eta_i^{\text{а}} \vee \eta_i^{\text{б}} \vee \ldots \vee \eta_i^{\text{я}} \vee \eta_i^{-} \vee \eta_i^{\square} \quad (i = 1, 2, \ldots, n). \tag{9}$$

Центральное место в описании склонения занимает предикат, связывающий грамматические категории числа и падежа с типом склонения и падежными окончаниями. Обозначим его $L_1(\gamma, z_1, z_2, \omega)$ или просто L_1. В грамматике принято отдельно описывать словоизменение существительных каждого типа склонения. Можно передать это так: если слово имеет нулевой тип склонения, то связь падежей, чисел и окончаний описывается предикатом $L_1(0, z_1, z_2, \omega)$, если второй тип склонения — предикатом $L_1(2, z_1, z_2, \omega)$ и т. д.,— перебирая все необходимые типы склонения. Переведем эту фразу на язык формального исчисления:

$$L_1 = \bigwedge_{i=0,\,2,\,4,\,5,\,6,\,7} \gamma^i \supset L_1(i, z_1, z_2, \omega), \tag{10}$$

где $\wedge$ означает логическое произведение по всем значениям переменной i, перечисленным под ним.

Известно, что описание каждого типа склонения складывается из описаний склонения в единственном и во множественном числе, т. е. в i-м типе склонения в единственном числе окончания и падежи связаны предикатом $L_1(i, e, z_2, \omega)$, а во множественном числе — предикатом $L_1(i, м, z_2, \omega)$:

$$L_1(i, z_1, z_2, \omega) = \left(z_1^{\text{е}} \subset L_1(i, \text{е}, z_2, \omega)\right)\left(z_1^{\text{м}} \supset \right.$$
$$\left. \supset L_1(i, \text{м}, z_2, \omega)\right). \tag{11}$$

Всего имеется 6 выражений вида (11), для каждого типа склонения свое. В записи формулы (11) опущен знак операции конъюнкции, который должен стоять между двумя сомножителями. Будем опускать его везде, где это не вызовет недоразумения.

Рассмотрим склонение единственного числа существительных стандартного (второго) типа. Известно, что если такие существительные стоят в именительном, винительном и предложном падеже, то окончание их *-е*, в родительном *-я*, в дательном *-ю*, в творительном *-ем*:

$$L_1(2, \text{е}, z_2, \omega) = \left(z_2^{\text{и}} \vee z_2^{\text{в}} \vee z_2^{\text{п}} \supset \omega^{\text{е}}\right)\left(z_2^{\text{р}} \supset \right.$$
$$\left. \supset \omega^{\text{я}}\right)\left(z_2^{\text{д}} \supset \omega^{\text{ю}}\right) z_2^{\text{т}} \supset \omega^{\text{ем}}\big). \tag{12}$$

Хотя мы условились всякое окончание обозначать числом, здесь и ниже будем пользоваться его буквенной записью, чтобы не лишать изложение наглядности. Эту запись можно толковать как код числа в 33-ичной системе счисления.

Склонение слов стандартного типа во множественном числе определяется следующим правилом. Если падеж именительный, то окончание слова *-я*; если родительный, то в случае падения ударения на основу слово оканчивается на *-ь*, при ударении на окончании — на *-ей*; если падеж дательный, то окончание *-ям*; если падеж винительный, то окончание неодушевленных существительных такое же, как окончание существительных в именительном падеже, а одушевленных — такое же, как в родительном; если падеж творительный, то окончание *-ями*; если предложный, *-ях*. Рассмотрим отдельно ту часть правила, которая гласит: если падеж родительный, то в случае падения ударения на основу слово оканчивается на *-ь*, при ударении на окончании — на *-ей*.

У нас нет средств адекватно формализовать ее смысл, так как нет переменной, которая каким-то образом учитывала бы место ударения. Поэтому заменим эту часть правила ее неточным аналогом: если падеж родительный, то окончание словоформы *-ь* или *-ей*, что, конечно, огрубляет описание в целом. То же касается фрагмента правила, описывающего окончания винительного падежа. Так как одушевленность или неодушевленность также не учитываются, заменим этот фрагмент следующим: если падеж винительный, то окончание совпадает с окончанием именительного или родительного падежа. После этих замен правило в целом выразится следующим предикатом:

$$L_1(2,\ \text{м},\ z_2,\ \omega) = \left(z_2^{\text{и}} \supset \omega^{\text{я}}\right)\left(z_2^{\text{р}} \supset \omega^{\text{ь}} \vee \omega^{\text{ей}}\right)\left(z_2^{\text{д}} \supset \omega^{\text{ям}}\right) \wedge$$
$$\wedge \left(z_2^{\text{в}} \supset \omega^{\text{я}} \vee \omega^{\text{ь}} \vee \omega^{\text{ей}}\right)\left(z_2^{\text{т}} \supset \omega^{\text{ями}}\right)\left(z_2^{\text{р}} \supset \omega^{\text{ях}}\right). \tag{13}$$

Рассмотрим нулевой тип склонения, который значительно отличается от прочих типов. Известно, что слова нулевого типа имеют пустое окончание во всех падежах и числах, т. е.

$$L_1(0,\ z_1,\ z_2,\ \omega) = \omega^{\sqcup}, \tag{14}$$

где $\sqcup$ в данном случае символизирует пустое окончание Для остальных типов склонения приведем соответствующие формулы:

$$L_1(4,\ \text{е},\ z_2,\ \omega) = \left(z_2^{\text{и}} \vee z_2^{\text{в}} \vee z_2^{\text{п}} \supset \omega^{\text{е}}\right)\left(z_2^{\text{р}} \supset \omega^{\text{а}}\right) \wedge$$
$$\wedge \left(z_2^{\text{д}} \supset \omega^{\text{у}}\right)\left(z_2^{\text{т}} \supset \omega^{\text{ем}}\right); \tag{15}$$

$$L_1(4,\ \text{м},\ z_2,\ \omega) = \left(z_2^{\text{и}} \supset \omega^{\text{а}}\right)\left(z_2^{\text{р}} \supset \omega^{\sqcup}\right)\left(z_2^{\text{д}} \supset \omega^{\text{ам}}\right) \wedge$$
$$\wedge \left(z_2^{\text{в}} \supset \omega^{\text{а}} \vee \omega^{\sqcup}\right)\left(z_2^{\text{т}} \supset \omega^{\text{ами}}\right)\left(z_2^{\text{п}} \supset \omega^{\text{ах}}\right); \tag{16}$$

$$L_1(5,\ \text{е},\ z_2,\ \omega) = L_1(4,\ \text{е},\ z_2,\ \omega); \tag{17}$$

$$L_1(5,\ \text{м},\ z_2,\ \omega) = L_1(4,\ \text{м},\ z_2,\ \omega); \tag{18}$$

$$L_1(6,\ \text{е},\ z_2,\ \omega) = L_1(2,\ \text{е},\ z_2,\ \omega); \tag{19}$$

$$L_1(6,\ \text{м},\ z_2,\ \omega) = \left(z_2^{\text{и}} \supset \omega^{\text{я}}\right)\left(z_2^{\text{р}} \supset \omega^{\text{й}}\right)\left(z_2^{\text{д}} \supset \omega^{\text{ям}}\right) \wedge$$
$$\wedge \left(z_2^{\text{в}} \supset \omega^{\text{я}} \vee \omega^{\text{й}}\right)\left(z_2^{\text{т}} \supset \omega^{\text{ями}}\right)\left(z_2^{\text{п}} \supset \omega^{\text{ях}}\right); \tag{20}$$

$$L_1(7,\ \text{е},\ z_2,\ \omega) = \left(z_2^{\text{и}} \vee z_2^{\text{в}} \supset \omega^{\text{е}}\right)\left(z_2^{\text{р}} \supset \omega^{\text{я}}\right)\left(z_2^{\text{д}} \supset \omega^{\text{ю}}\right) \wedge$$
$$\wedge \left(z_2^{\text{т}} \supset \omega^{\text{ем}}\right)\left(z_2^{\text{п}} \supset \omega^{\text{и}}\right); \tag{21}$$

$$L_1(7,\ \text{м},\ z_2,\ \omega) = \left(z_2^{\text{и}} \supset \omega^{\text{я}}\right)\left(z_2^{\text{р}} \supset \omega^{\text{й}} \vee \omega^{\text{ев}}\right)\left(z_2^{\text{д}} \supset \omega^{\text{ям}}\right) \wedge$$
$$\wedge \left(z_2^{\text{в}} \supset \omega^{\text{я}} \vee \omega^{\text{й}}\right)\left(z_2^{\text{т}} \supset \omega^{\text{ями}}\right)\left(z_2^{\text{п}} \supset \omega^{\text{ях}}\right). \tag{22}$$

Важную роль в математической модели играют предикаты, связывающие слово X с морфологическими характеристиками α и γ. В нашем примере каждой из основ соответствует слово, и притом только одно:

$$L_2(X, \alpha) = \bigwedge_{i=1}^{i} \left(\alpha \sim x_1^{a_{i1}} x_2^{a_{i2}} \ \ldots \ x_n^{a_{in}}\right). \tag{23}$$

Здесь буквы a_{i1}, a_{i2}, ..., a_{in} составляют слово с основой i.

Распределение слов по типам склонения можно описать следующим образом. Если предпоследняя буква слова *л* или *р*, то слово 2-го типа склонения; *ж*, *ч* или *щ*, то 4-го; если *ц*, то 5-го; если *ь*, то 6-го; если *и*, то 7-го. Для формализации этого правила воспользуемся промежуточной переменной L_{31}, которой обозначим предпоследнюю букву слова:

$$L_{31} = (\mu^{\text{л}} \vee \mu^{\text{р}} \supset \gamma^2)(\mu^{\text{ж}} \vee \mu^{\text{ч}} \vee \mu^{\text{щ}} \supset \gamma^4)(\mu^{\text{ц}} \supset \gamma^5)(\mu^{\text{ь}} \supset \gamma^6)(\mu^{\text{и}} \supset \gamma^7). \tag{24}$$

К этому следует добавить, что на предпоследнем месте у склоняемых существительных среднего рода, кончающихся на *-е*, возможна лишь одна из букв: *л*, *р*, *ж*, *ч*, *щ*, *ц*, *ь*, *и*:

$$L_{32} = \mu^{\text{л}} \vee \mu^{\text{р}} \vee \mu^{\text{ж}} \vee \mu^{\text{ч}} \vee \mu^{\text{щ}} \vee \mu^{\text{ц}} \vee \mu^{\text{ь}} \vee \mu^{\text{и}}. \tag{25}$$

Само же понятие «предпоследняя буква» задается предикатом

$$L_{33} = (\forall i \in \{1, 2, \ldots, n-2\})\,[x_{i+1}^{\text{е}} x_{i+2}^{\sqcup} \supset (\mu = x_i)]. \tag{26}$$

Формула $\mu = x_i$ является сокращенной записью выражения

$$\mu^{\text{а}} x_i^{\text{а}} \vee \mu^{\text{б}} x_i^{\text{б}} \vee \ \ldots \ \vee \mu^{\text{я}} x_i^{\text{я}} \vee \mu^{-} x_i^{-} \vee \mu^{\sqcup} x_i^{\sqcup}.$$

Правило, которое выражается конъюнкцией предикатов L_{31}, L_{32}, L_{33}, справедливо только для склоняемых существительных. Тип склоняемых несклоняемых существительных — нулевой, он не зависит от того, какая буква стоит на предпоследнем месте:

$$L_{34} = \gamma^0. \tag{27}$$

Все несклоняемые существительные полностью определяются перечнем своих основ — M_0. При этом общее правило определения типа склонения выглядит так:

$$L_3(X, \alpha, \gamma) = (\overline{\alpha \in M_0} \supset L_{31} L_{32} L_{33})(\alpha \in M_0 \supset L_{34}). \tag{28}$$

Выражение $\alpha \in M_0$ представляет собой сокращенную запись предиката $\alpha^{m_1} \vee \alpha^{m_2} \vee \ \ldots \ \vee \alpha^{m_k}$, где $\{m_1, m_2, \ldots, m_k\} = M_0$.

Список литературы: 1. *Шабанов-Кушнаренко Ю. П.* Применение метода нуль-органа в лингвистике.— Проблемы бионики. Харьков, 1978, вып. 21, с. 3—15. 2. *Шабанов-Кушнаренко Ю. П.* О теории интеллекта.— Проблемы бионики. Харьков, 1979, вып. 22. с. 3—13. 3. *Зализняк А. А.* Русское именное словоизменение. М., Наука, 1967. 370 с. 4. *Зализняк А. А.* Грамматический словарь русского языка. М., Русский язык, 1977. 879 с.

УДК 519.766.2

М. Ф. БОНДАРЕНКО, канд. техн. наук, *В. М. БОНДАРЕВ*

О МАТЕМАТИЧЕСКОМ ОПИСАНИИ СЛОВОИЗМЕНЕНИЯ СУЩЕСТВИТЕЛЬНЫХ. *СООБЩЕНИЕ 2*

Необходимой частью математического описания словоизменения существительных являются предикаты, связывающие словоформу с морфологическими характеристиками. Представляется полезным уметь выделить буквы основы словоформы и указать ее окончание.

Рассмотрим следующее высказывание. Если окончание словоформы -*я*, то существует такой номер переменной k, что $y_i = \eta_i (i < k)$, $y_k = я$, $\eta_k = \sqcup$, а все переменные y_i и $\eta_i (i > k)$ равны пробелу,

$$L_{4,я} = \omega^{я} \supset (\exists\, k \in \{1,\ 2,\ \ldots,\ n-1\}) \left[\bigwedge_{i=1}^{k-1} (y_i = \eta_i)\, \overline{\eta_{k-1}^{\sqcup}} \wedge \right.$$

$$\left. \wedge\, y_k^{я} \eta_k^{\sqcup} \bigwedge_{i=k+1}^{n} y_i^{\sqcup} \eta_i^{\sqcup} \right]. \tag{29}$$

Запись $(\exists\, k \in \{1,\ 2,\ \ldots,\ n\})\, P(k)$ означает $P(1) \vee P(2) \vee\ \ldots\ \vee P(n)$.

Трехбуквенному окончанию -*ями* соответствует предикат:

$$L_{4,\,ями} = \omega^{ями} \supset (\exists\, k \in \{1,\ 2,\ \ldots,\ n-3\}) \left[\bigwedge_{i=1}^{k-1} (y_i = \eta_i)\, \overline{\eta_{k-1}^{\sqcup}} \wedge \right.$$

$$\left. \wedge\, y_k^{я} y_{k+1}^{м} y_{k+2}^{и} \eta_k^{\sqcup} \eta_{k+1}^{\sqcup} \eta_{k+2}^{\sqcup} \bigwedge_{i=k+3}^{n} y_i^{\sqcup} \eta_i^{\sqcup} \right]. \tag{30}$$

Обобщим выражения типа (29) и (30):

$$L_{4,\,i} = \omega^{i} \supset (\exists\, k \in \{1,\ 2,\ \ldots,\ n-3\}) \left[\bigwedge_{i=1}^{k-1} (y_i = \eta_i)\, \overline{\eta_{k-1}^{\sqcup}} \wedge \right.$$

$$\left. \wedge\, y_k^{a_1} y_{k+1}^{a_2} y_{k+2}^{a_3} \eta_k^{\sqcup} \eta_{k+1}^{\sqcup} \eta_{k+2}^{\sqcup} \bigwedge_{i=k+3}^{n} y_i^{\sqcup} \eta_i^{\sqcup} \right]. \tag{31}$$

Каждому значению переменной i соответствует один столбец табл. 1.

Таблица 1

ω	е	я	а	ю	у	и	й	ь	ем	ей	ев	ям	ам	ях	ах	ями	ами	$\sqcup$
a_1	е	я	а	ю	у	и	й	ь	е	е	е	я	а	я	а	я	а	$\sqcup$
a_2	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	м	й	в	м	м	х	х	м	м	$\sqcup$
a_3	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	и	и	$\sqcup$

Если окончание словоформы пустое, выражение (31) можно упростить:

$$L_{4,\sqcup} = \omega^{\sqcup} \supset \bigwedge_{i=1}^{n} (y_i = \eta_i). \qquad (32)$$

Общий предикат, учитывающий связь переменных y, ω и η, будет логическим произведением всевозможных предикатов $L_{4,i}$:

$$L_4(y,\ \omega,\ \eta) = \bigwedge_{i=1}^{18} L_{4,i}. \qquad (33)$$

Изучим предикаты, описывающие чередование в основах словоформ. Рассмотрим связь между словом X, основой словоформы η, типом склонения γ и ступенью чередования β. Если в словоформе имеет место первая ступень чередования, то все буквы слова X и основы η, совпадают, начиная с 1-й и кончая некоторой k-й буквой, причем эта буква — последняя в основе, не являющаяся пробелом. Если тип склонения нулевой, то $(k+1)$-я буква слова — $\sqcup$, если не нулевой, то $(k+1)$-я буква слова — $е$. Все остальные буквы основы и слова — пробелы:

$$L_{5,1} = \beta^1 \supset (\exists k \in \{1,\ 2,\ \ldots,\ n-2\}) \Big[\bigwedge_{i=1}^{k} (\eta_i = x_i)\, \overline{\eta_k^{\sqcup}} \wedge$$

$$\wedge\, \eta_{k+1}^{\sqcup} \left(\gamma^0 x_{k+1}^{\sqcup} \vee \overline{\gamma^0} x_{k+1}^{\text{е}}\right) \bigwedge_{i=k+2}^{n} x_i^{\sqcup} \eta_i^{\sqcup} \Big]. \qquad (34)$$

Таблица 2

x_k	x_{k+1}	x_{k+2}	η_k	η_{k+1}	η_{k+2}
ь	е	⊔	и	⊔	⊔
в	ц	е	в	е	ц
д	ц	е	д	е	ц
н	ц	е	н	е	ц
р	ц	е	р	е	ц
т	ц	е	т	е	ц
ь	ц	е	е	ц	⊔

Если ступень чередования вторая, то соответствие между буквами слова и основы нагляднее всего выразится табл. 2.

Все буквы слова и основы словоформы с 1-й по $(k-1)$-ю попарно одинаковы, а все символы, следующие за $(k+2)$-м, являются пробелами. После вынесения за скобки общих элементов предикат, описывающий таблицу и замечания к ней, будет следующим:

$$L_{5,2} = \beta^2 \supset (\exists k \in \{1,\ 2,\ \ldots,\ n-3\}) \Big[\Big(\bigwedge_{i=1}^{k-1} x_i = \eta_i \bigwedge_{i=k+3}^{n} x_i^{\sqcup} \eta_i^{\sqcup} \Big) \wedge$$

$$\wedge \big(x_k^{\text{ь}} x_{k+1}^{\text{е}} x_{k+2}^{\sqcup} \eta_k^{\text{и}} \eta_{k+1}^{\sqcup} \eta_{k+2}^{\sqcup} \vee x_{k+1}^{\text{ц}} x_{k+2}^{\text{е}} \big(\eta_{k+1}^{\text{е}} \eta_{k+2}^{\text{ц}} \big(x_k^{\text{в}} \eta_k^{\text{в}} \vee x_k^{\text{д}} \eta_k^{\text{д}} \vee$$

$$\vee\, x_k^{\text{н}} \eta_k^{\text{н}} \vee x_k^{\text{р}} \eta_k^{\text{р}} \vee x_k^{\text{т}} \eta_k^{\text{т}} \big) \big) \vee x_k^{\text{ь}} \eta_k^{\text{е}} \eta_{k+1}^{\text{ц}} \eta_{k+2}^{\sqcup} \big) \Big]. \qquad (35)$$

Произведение $L_{5,1} \wedge L_{5,2}$ даст предикат $L_5(\beta,\ X,\ \eta,\ \gamma)$:

$$L_5(\beta,\ X,\ \eta,\ \gamma) = L_{5,1} \wedge L_{5,2}. \qquad (36)$$

Последним звеном нашей модели будет описание зависимости между типом склонения γ, окончанием словоформы ω, чередованием β, основой α и словом X. Если тип склонения 6-й, а окончание словоформы -й или же тип склонения 5-й, окончание пустое, а третьей от конца буквой слова X является *н*, *р*, *т* или *ь* (слово *солнце* — исключение), то в словоформе должна иметь место вторая ступень чередования (в противном случае ступень чередования первая):

$$L_6(\alpha, X, \omega, \gamma, \beta) = L_{6,1} L_{6,2} \sim \beta^2; \tag{37}$$

$$L_{6,1} = \gamma^6 \omega^{\text{я}} \vee \gamma^5 \omega^{\sqcup} (\mu_1^{\text{в}} \vee \mu_1^{\text{д}} \vee \mu_1^{\text{н}} \vee \mu_1^{\text{р}} \vee \mu_1^{\text{т}} \vee \mu_1^{\text{ь}}) \bar{\alpha}^{c}. \tag{38}$$

Здесь показатель c в выражении $\bar{\alpha}^c$ означает номер основы слова *солнце*, μ_1 — третья от конца буква слова X:

$$L_{6,2} = (\forall i \in \{4, 5, \ldots, n\}) [x_{i-1}^{\text{е}} x_i^{\sqcup} \supset (\mu_1 = x_{i-3})]. \tag{39}$$

Построенную модель фрагмента морфологии русского языка укрупненно можно представить в виде системы предикатов

$$L_0(X, Y, Z, \Gamma);\ L_1(Z, \gamma, \omega);\ L_2(X, \alpha);\ L_3(X, \alpha, \gamma);\ L_4, (\eta, \omega, Y);$$
$$L_5(X, \beta, \eta);\ L_6(\gamma, \omega, \alpha, X, \beta).$$

Под предикатом $L_0(X, Y, Z, \Gamma)$ подразумевается конъюнкция предикатов (1—9).

При решении задач морфологической обработки не обязательно представлять систему (40) в виде единого уравнения $L_0 \wedge L_1 \wedge \ldots \wedge L_6 = 1$. Можно отдельно искать корни каждого из уравнений $L_i = 1$ $(i = 0, 1, \ldots, 6)$ и в качестве ответа задачи брать те из них, которые удовлетворяют всем уравнениям системы. При этом, решая очередное уравнение, целесообразно опираться на сведения о корнях уравнений, решенных ранее.

Усилия, затраченные на отыскание корней системы существенно зависят от того, в каком порядке решаются уравнения $L_i = 1$. По-видимому, для каждого типа морфологических задач существует некоторый оптимальный порядок. На наш взгляд, для задачи синтеза он таков: L_0, L_2, L_3, L_1, L_6, L_5, L_4.

Приведем пример морфологического синтеза. Заданы словарная форма *поле* ($x_1 = п$, $x_2 = о$, $x_3 = л$, $x_4 = е$, $x_5 = x_6 = x_7 = \sqcup$), число искомой словоформы — единственное ($z_1 = \text{е}$), падеж — творительный ($z_2 = \text{т}$). Требуется определить словоформу, т. е. найти значение переменной Y. В разбираемых примерах ограничимся величиной $n = 7$. Разделим весь процесс отыскания переменной Y на этапы, каждый из которых будем связывать с решением одного из уравнений $L_i = 1$. Для ясности начало описания очередного этапа обозначим символом соответствующего предиката.

L_0. В результате «решения» уравнения $L_0 = 1$ убеждаемся, что исходные значения X и Z не выходят за пределы их областей определения.

L_2. Значение $X = \langle п, о, л, е, \sqcup, \sqcup, \sqcup, \sqcup \rangle$ обращает в 0 все конъюнкции $x_1^{a_{i1}} x_2^{a_{i2}}, \ldots, x_n^{a_{in}}$, кроме одной — $x_1^{п} x_2^{о} x_3^{л} x_4^{е} \times$ $\times\, x_5^{\sqcup} x_6^{\sqcup} x_7^{\sqcup}$. Поэтому существует лишь одно значение α, которое удовлетворяет уравнению $L_2(X, \alpha) = 1$, а именно то, которое обращает в 1 эквивалентность $\alpha^m \sim x_1^{п} x_2^{о} x_3^{л} x_4^{е} x_5^{\sqcup} x_6^{\sqcup} x_7^{\sqcup}$. Поскольку предикат (23) записан в общем виде, будем считать, что $\alpha = m$, где m — условный номер основы слова *поле*.

L_3. Так как слово *поле* не относится к числу несклоняемых, M_0 не может содержать элемент m (из-за того, что M_0 не задано явно, мы лишены возможности проверить это непосредственно). Значит, справедливо утверждение $\overline{\alpha \in M_0}$, что влечет истинность всех сомножителей произведения $L_{31} L_{32} L_{33}$. Определим значение переменной μ из (26); $x_{i+1}^{е} x_{i+2}^{\sqcup} = 1$ только, если $i = 3$, значит, $\mu = x_3 = л$. Это согласуется с (25) и в силу (24) дает единственно возможное значение для типа склонения: $\gamma = 2$.

L_1. Так как $\gamma = 2$, из (10) заключаем, что $L(2, z_1, z_2, \omega) = 1$. Из условия $z_1 = м$ и (11) получаем, что $L(2, м, z_2, \omega) = 1$. Наконец, из выражения для $L(2, м, z_2, \omega)$ (13) и условия $z_2 = т$ имеем $\omega =$ *-ями*.

L_6. Значение переменной $\gamma = 2$ обращает в 0 предикат (38). Следовательно, из (37) имеем $0 \sim \beta^2$, что в сочетании с (7) дает $\beta = 1$.

L_5. Выражение (36) говорит о том, что корни системы должны обратить в 1 предикаты $L_{5,1}$ и $L_{5,2}$. Из того, что $\beta = 1$ и $L_{5,1} = 1$, однозначно следуют равенства $\eta_1 = x_1 = п$; $\eta_2 = x_2 = о$; $\eta_3 = x_3 = л$; $\eta_4 = \eta_5 = \eta_6 = \eta_7 = \sqcup$.

L_4. Из (33) и (30) получаем искомое значение переменной $Y = \langle y_1, y_2, \ldots, y_n \rangle$: $y_1 = \eta_1 = п$; $y_2 = \eta_2 = о$; $y_3 = \eta_3 = л$; $y_4 = я$; $y_5 = м$; $y_6 = и$; $y_7 = \sqcup$.

Рассмотрим пример морфологического анализа. Дана словоформа *донец* ($y_1 = д$, $y_2 = о$, $y_3 = н$, $y_4 = е$, $y_5 = ц$, $y_6 = \sqcup$, $y_7 = \sqcup$). Требуется отыскать значение числа и падежа, соответствующие заданной словоформе. Порядок решения уравнений в задаче анализа выберем следующим: L_0, L_4, L_5, L_2, L_3, L_6, L_1.

L_0. Согласно (2), заданные значения переменных y_i $(i = 1, 2, \ldots, 7)$ содержатся в их области определения.

L_4. Словоформа *донец* такова, что обращает в 0 выражения, стоящие в предикатах $L_{4,i}$ после знака импликации. Лишь в предикате $L_{4,\sqcup}$ (32) при условии $\eta_i = y_i$ $(i = 1, 2, \ldots, 7)$ это выражение может быть равным 1, что допускает значение $\omega = \sqcup$. Это единственно возможное решение, при котором истинны все предикаты $L_{4,i}$ $(i = 1, 2, \ldots, 18)$, а вместе с ними и L_4 (33).

L_5. Анализируя предикаты, составляющие L_5 (34)—(36), убеждаемся, что уравнению $L_5 = 1$ удовлетворяют два набора значений неизвестных X и β: $\langle\langle д, о, н, е, ц, \sqcup\rangle, 1\rangle$ и $\langle\langle д, о, н, ц, е, \sqcup, \sqcup\rangle, 2\rangle$.

L_2. Решая уравнение $L_2 = 1$ (23), приходим к выводу, что $X = \langle д, о, н, е, ц, е, \sqcup\rangle$ не является его корнем, так как наборы $\langle a_{i1}, a_{i2}, \ldots, a_{in}\rangle$ представляют собой русские слова, дополненные пробелами до стандартной длины, и среди них не может быть слова *донеце*. Определяем номер основы слова *донце* (условно принимаем $\alpha = d$).

L_3. Сопоставляя d и M_0, видим, что $d \in M_0$, значит, истинен предикат $\alpha \overline{\in M_0}$, поскольку $\alpha = d$. Из (28) заключаем, что $L_{3,1}L_{3,2}L_{3,3} = 1$, далее, согласно (26) $\mu = ц$. Это удовлетворяет (25) и в соответствии с (24) дает $\gamma = 5$.

L_6. Подстановка ранее найденных значений $\alpha = d$, $X = <д, о, н, ц, е, \sqcup, \sqcup>$, $\omega = \sqcup$, $\gamma = 5$, $\beta = 2$ в уравнения $L_{6,1} = 1$; $L_{6,2} = 1$; $L_6 = 1$ (37)—(39) не приводит к противоречию, что позволяет нам перейти к следующему этапу анализа.

L_1. Из (10) и $\gamma = 5$ делаем заключение об истинности предиката $L_1(5, z_1, z_2, \omega)$. Анализируя (17) и (18), замечаем, что L_1(5, е, z_2, ω) ложен при имеющемся значении $\omega = \sqcup$, а L_1(5, м, z_2, ω) $= 1$ в случае $z_2 =$ р или $z_2 =$ в. Из (11) в этих же случаях получаем: $z_1 =$ м.

Итак, решив задачу морфологического анализа, мы установили, что число словоформы *донец* — множественное, падеж — родительный или винительный. Это не должно вызывать удивления, если вспомнить, что построенная модель не учитывает категории одушевленности. Полученный ответ допускает оба возможных значения этой категории. Попутно выяснили, что $X = <д, о, н, ц, е, \sqcup, \sqcup>$, т. е. решили задачу нормализации.

Может показаться, что решение некоторых уравнений, например $L_2 = 1$, излишне при морфологическом анализе. Следующий пример доказывает обратное. Пусть требуется произвести морфологический анализ ошибочно заданной словоформы *маре*.

L_4. Уравнению $L_4 = 1$ удовлетворяют две пары значений: $\omega = \sqcup$, $\eta = <м, а, р, е, \sqcup, \sqcup, \sqcup>$ и $\omega = е$, $\eta = <м, а, р, \sqcup, \sqcup, \sqcup, \sqcup>$.

L_5. Возможными корнями уравнения $L_5 = 1$ будут: $X = <м, а, р, е, \sqcup, \sqcup, \sqcup>$, $X = <м, а, р, е, е, \sqcup, \sqcup>$, $X = <м, а, р, \sqcup, \sqcup, \sqcup, \sqcup>$.

L_2. Ни одно из буквосочетаний *маре*, *марее*, *мар* не является словом русского языка, поэтому ни одно из соответствующих значений переменной X не обратит в тождества одновременно уравнения $L_2 = 1$ (23) и (5). Следовательно, не существует решения системы L' в целом, что свидетельствует о противоречивости исходных данных.

Если требуется по заданной словоформе найти ее словарную форму, т. е. произвести нормализацию, необходимо последовательно решить уравнения: $L_0 = 1$; $L_4 = 1$; $L_5 = 1$; $L_2 = 1$. Например, задана словоформа *чудищ* ($y_1 = ч$, $y_2 = у$, $y_3 = д$, $y_4 = и$, $y_5 = щ$, $y_6 = y_7 = \sqcup$).

L_0. С помощью (2) проверяем, принадлежат ли значения переменных y_i ($i = 1, 2, \ldots, 7$) их области определения.

L_4. Из (33), (32) и (6) заключаем, что $\omega = \sqcup$, $\eta = <ч, у д, и, щ, \sqcup, \sqcup>$.

L_5. Уравнение $L_5 = 1$ дает два допустимых значения переменной X: $\langle ч, у, д, и, щ, е, \sqcup \rangle$ и $\langle ч, у, д, и, щ, \sqcup, \sqcup \rangle$.

L_2. Из этих значений уравнению $L_2 = 1$ удовлетворяет лишь первое, т. е. искомая словарная форма — *чудище*.

В заключение отметим, что построенная нами модель данного фрагмента морфологии не является единственно возможной. Состав предикатов этой модели может быть другим, или же они могут выражаться иными формулами.

УДК 62.506.2: 15: 612.821.3

А. П. ИЛЬИНСКИЙ, Л. С. ИЛЬИНСКАЯ, А. В. КОРОП

НЕКОТОРЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА. *СООБЩЕНИЕ 1*

Исследованиям интеллектуальной деятельности человека уделяют внимание представители разных областей знаний. Предпринимается ряд попыток построить алгоритмические модели некоторых видов интеллектуальной деятельности.

В связи с этим все большее распространение получает бионический подход, основанный на кибернетических аспектах изучения процесса принятия решения реальными людьми. Однако выяснение того, как человек решает даже простейшую задачу, наталкивается на огромные трудности. Непосредственное наблюдение за процессом решения и самоотчет после решения мало эффективны, так как многие важные моменты после окончания решения обычно забываются.

В психологических исследованиях решений человека широко распространен метод мышления вслух [5]. Однако вербализация существенно отвлекает испытуемого от размышлений. Н. Бор и Л. Бриллюэн образно называли возникающие «возмущения» процесса мышления «соотношением неопределенностей» в психологии: чем точнее думающий пытается выяснить, как он думает, тем труднее продолжать ему думать в этом же направлении [4]. Кроме того, в результате изучения научного, технического и других видов творчества установлено, что в продуктивное мышление включены процессы, осуществление которых не осознается [1]. Субъективно момент осознания решения, полученного интуитивно, переживается как внезапное озарение — «инсайт» [13]. В отличие от дискурсивного мышления об этой бессознательной части своих размышлений испытуемый ничего не знает и, естественно, рассказать не может.

Особое суггестивное состояние психики человека (ОССП). Делались попытки исследовать бессознательную составляющую мышления при помощи гипноза [3]. Однако из-за того, что в гипнозе снижается интеллектуальная продуктивность, такой заманчивый подход распространения не получил: Действительно, сомнамбулической фазе гипноза испытуемые обычно имеют вид находящихся в просоночном состоянии: неподвижный, устремленный вдаль взгляд с редкими миганиями век, «мутные глаза», движения задержаны, однообразная мимика, речь монотонна и несколько заглушена [12]. Происходит снижение корковой возбудимости, торможение второй сигнальной системы, наиболее глубокая диссоциация корковой деятельности в целом, включая афферентные и эфферентные звенья. Сенсорная хроноксия увеличивается в 1,5—2 раза. Умственная работоспособность резко падает, например, количество сложений пар чисел уменьшается в 2—4 раза, а число ошибок при этом увеличивается в 2—3 раза [8].

Проведенные В. Н. Пушкиным исследования влияния гипнотического сна на решение творческих проблем (т. е. задач, область поиска которых неопределена) показали:

а) задачи этого класса можно решать только в состоянии постгипнотического внушения;

б) интенсифицирующие творческое мышление формулы постгипнотических внушений срабатывали только в отношении тех задач, которые решались испытуемыми до гипноза, но воспринимались ими как новые в силу внушенной амнезии. При решении действительно новых задач этого же класса не была зарегистрирована разница результатов по сравнению с испытуемыми контрольной группы, которые бодрствовали [1]. В состоянии гипноза новые идеи не появляются, а постгипнотические внушения активизируют разработку лишь тех идей, которые, хотя бы неосознанно, возникали до гипноза.

Иные результаты получил В. Л. Райков. Испытуемым в гипнозе он внушал «гипнорепродукцию в другую активную личность». В этом состоянии они более продуктивно решали шахматные этюды, занимались художественным и иными видами творчества. В. Л. Райков отмечал, что полученное состояние «...отличается от классического гипноза самого глубокого уровня по схеме Е. С. Каткова... Не наблюдалось заметного торможения второй сигнальной системы, и речевая продукция иногда была даже более активной, чем в состоянии бодрствования. Не наблюдалось строго изолированного контакта испытуемых только с врачом. В пределах заданного образа гипнорепродукции испытуемые могли беседовать с любым человеком и даже завязывать эти беседы по собственной инициативе. Также не отмечено указанной Е. С. Катковым «мутности глаз» у загипнотизированных. Глаза в гипнозе имели обычное выражение, свойственное бодрствующему человеку... Важным доказательством глубины состояния являлась спонтанная невнушенная тотальная амнезия на весь период гипнотического сеанса... ЭЭГ регистрация не зафиксировала какого-либо отклонения от состояния бодрствования, хотя у испытуемых наблюдалось глубокое изменение самосознания и галлюцинаторная интерпретация собственной личности, а также осуществлялись внушения отрицательных галлюцинаций...» [10].

Из приведенного следует, что «гипнорепродукция в другую личность» — это методический прием получения нового состояния, отличающегося от классического гипноза. К сожалению, в последующих работах В. Л. Райкова не упоминается о более глубоких исследованиях этого нового психофизиологического состояния [9]. Подобное состояние исследовано нами ранее [6; 7].

А. А. Востриков совместно с В. В. Ивановым наблюдали новое, отличающееся от гипноза состояние, названное «энфазой» (и «самоэнфазой» — в случае самоуправления). Основное внимание было уделено лечебному и спортивному применению «энфазы» и «самоэнфазы». Обсуждение с А. А. Востриковым наших результатов привело к выводам об идентичности «энфазы» полученному нами особому психофизическому состоянию. По предложению А. А. Вострикова для названия этого состояния введена аббревиатура ОССП (особое суггестивное состояние психики). ОССП объединяет положительные стороны как гипноза, так и бодрствования:

1. Если в гипнозе повышение внушаемости происходит на фоне сужения сознания в результате значительного торможения коры, то в ОССП испытуемого не усыпляют, и повышенная внушаемость (или самовнушаемость) возникает на фоне бодрого статуса коры, по-видимому за счет облегчения системообразования доминант. Известно, что отличием доминантного очага является повышенная работоспособность охваченных им мозговых структур. И действительно, в ОССП наблюдается особенная обостренность сознания и эффективность деятельности (интеллектуальной или физической). Испытуемые, находящиеся в ОССП, малоутомляемы, имеют пониженную потребность в естественном сне. Не было отмечено утомления у испытуемых, занимавшихся в ОССП своей обычной деятельностью (подготовкой к экзаменам, спортивными тренировками, работой) и лишенных при этом сна на срок до пяти суток.

2. Как и в гипнозе, в ОССП возможно суггестивное управление рядом функций организма: мышечным тонусом вплоть до каталепсии, получение аналгезии, гиперестезии и т. п.; вниманием — получение высокой его концентрации и сосредоточенности, управление его объемом (с одинаково высоким вниманием воспринимается большое количество объектов), его распределяемостью (испытуемые различают несколько объектов, которые были объединены в одно целое); индивидуальным темпом времени синхронно с каким-либо внешним процессом, ускорение его или замедление, вплоть до полной «остановки», получения состояния «вневременья» или отсутствия только прошлого или только будущего.

3. Как в наивысшей (по Е. С. Каткову) стадии гипноза, т. е. как у сомнамбулы, у испытуемого в ОССП реализуются

любые положительные и отрицательные галлюцинации, которые, в отличие от гипноза, образуют единую систему с той частью объективной действительности, которая соответствующим внушением оставлена внутри «круга внимания». Например, стол с книгами и записями для создания комфортности можно было суггестивно поместить на лесную поляну или в читальный зал. В таких случаях испытуемый полностью переставал воспринимать все, что не относилось к его занятиям, вплоть до имевшей место в отдельных экспериментах полной потери раппорта, восстановить который удавалось только положив записку внутрь «круга внимания» испытуемого.

4. В гипнозе в состоянии торможения (сна) находится почти вся кора. Сознательная воля испытуемого заменяется волей и диктатом гипнотизера. В отличие от этого в ОССП испытуемый, как при бодрствовании, имеет инициативу, активность и критическое восприятие. В пределах системы внушений он четко ориентируется в окружающей среде и имеет неограниченный раппорт, а также ясное, даже более ясное, чем при бодрствовании, сознание (в отличие от суженного, характерного для гипноза).

5. Как и в гипнозе, в ОССП возможно полное управление памятью. Например, студента суггестивно переносили в прошлое. Он повторно слышал голос лектора, видел доску и дописывал в конспект пропущенные куски лекций. «Возврат в прошлое» использовался в учебном процессе (невидимые «шпаргалки»), для реализации некоторых из описанных ниже методик и др. Амнезия либо входит как органическая часть в ту или иную методику, либо может быть получена по потребности. Спонтанной амнезии, как у В. Л. Райкова [10], ОССП не дает.

Разработаны две группы методик получения ОССП:

1. Перевод испытуемого в ОССП из сомнамбулического состояния. В отличие от «гипнорепродукции» В. Л. Райкова во время сеансов с применением нашей методики испытуемые остаются сами собой, сохраняя свое имя и личность [6; 7]. Один из очень эффективных приемов перевода сомнамбулы в ОССП — стимулирование подробных воспоминаний о ближайшем прошлом вплоть до момента гипнотизации, о самом процессе гипнотизации и о всех дальнейших внушенных событиях. Активность и ясность ума испытуемого восстанавливается одновременно с восстановлением естественной цепочки воспоминаний. Кроме того, иногда удается получить ОССП прямым внушением сомнамбуле активности и ясности ума при полном отсутствии сонливости.

2. Разработаны методики обучения погружению в ОССП из состояния бодрствования. Для этого у испытуемого вызывают галлюцинацию при открытых глазах, например на фоне движущейся части его тела, либо во время движения самого испытуемого на фоне галлюцинаторной стенки. После получения ОССП у испытуемого вырабатывается умение «проваливаться» в него

мгновенно по кодовому слову или иному ключу, минуя различные стадии шарма, гипотаксии и усыпления. Следующий этап — обучение применению ОССП. Для самостоятельного использования ОССП в учебе, спорте и работе мы предоставляли испытуемым возможность самоуправления [6]. При этом в целях защиты от посторонних суггестивных воздействий формулировалось внушение, что вход и выход из ОССП может быть осуществлен только самим испытуемым по коду. В последующих сообщениях будут описаны методики, основанные на ОССП, с самоуправлением (аутосуггестией).

Ниже описываются гетеросуггестивные методики, основанные на применении ОССП.

Методика «речь». Традиционная психологическая методика вербальных самоотчетов 5, как известно, создает скованность, напряженность, мешает думать. ОССП позволяет вырабатывать психологический автоматизм, запускаемый по кодовому слову «речь». Испытуемый подробно рассказывает о том, что он думает, совершенно не осознавая своего рассказа.

Самоотчет получается без возмущения хода размышления. Для проверки чистоты этой методики во время экспериментов и после их окончания у испытуемых при помощи других методов выясняли, говорили ли они (по их мнению) во время решения задач и не мешало ли им что-либо думать. Ответы всегда были отрицательными.

Методика «маркировка». Существенный недостаток традиционной методики самоответов — дефицит времени. Испытуемые обычно жалуются, что не успевают вербализовать мысли и особенно образы, которые сменяются в сознании значительно быстрее, чем их можно охарактеризовать. Методика «маркировка» была создана как один из способов преодоления этой трудности. Здесь испытуемый не пытается более или менее полно выговаривать содержимое своего сознания, а лишь «нарекает» (снабжает краткими метками-названиями) возникающие у него образы и связки мыслей. Делает он это тоже автоматически вербально или графически, совершенно не замечая своих высказываний, надписей или зарисовок. Затем, используя ОССП, по каждой надписи, зарисовке или высказыванию, получают «возврат в прошлое». Теперь испытуемый имеет достаточно времени, чтобы более полно вербализовать мысли и образы, которые возникали во время эксперимента в процессе принятия решения.

Уровень проникновения в бессознательное методик «речь» и «маркировка» зависит от многих факторов. Эксперименты показали, что наиболее существенным из них, по-видимому, является осознанная актуальность решения. Так, если испытуемый решает какую-то конкретную задачу, упомянутые методики выявляют только дискурсивную информацию. Но если у испытуемого загружено только внимание, например он занимается чем-либо легким — читает, слушает, смотрит, эти методики фиксируют информацию о параллельных размышлениях, которые испытуемым не осознаются.

Методика «двухканальный вывод». Приведенные методики, если и допускали возможность выявления некоторых бессознательных эвристик, то лишь как альтернативу к осознаваемым. Наконец, не удавалось установить синхронизм между дискурсивным и неосознаваемым мышлением. Можно, однако, предоставить самому испытуемому провести такую дихотомию. Эта идея, предполагающая, что известно, где проходит граница между сознательным и неосознаваемым, не столь невероятна, как это кажется на первый взгляд. Действительно, йоги демонстрируют умение управлять многими неосознаваемыми физиологическими функциями организма, в том числе перистальтикой кишечника, тканевым метаболизмом и т. п. Ряд работ А. С. Ромена демонстрирует возможность произвольного управления вегетативными функциями, формулой крови, проводимостью точке акупунктуры, мышечным тонусом (вплоть до каталепсии) и др. [11].

Возможность управления приводит, в свою очередь, к мысли о существовании в мозгу некой «управляющей структуры», которая «знает», куда направлять информацию, адресованную неосознаваемой сфере. А отсюда вытекает предположение о том, что эта «управляющая структура» также «знает» адрес для вывода неосознаваемой информации. На основании этих соображений были поставлены эксперименты, удача которых превзошла ожидания.

Во время экспериментов по методике, названной методикой «двухканального вывода», испытуемый подробно выговаривал вслух содержимое своего бессознательного и одновременно зарисовками или надписями маркировал содержимое сознания. Высказывания и зарисовки испытуемым не осознавались и не мешали естественному ходу его мыслей. Каналы вывода информации о бессознательном и о сознании можно менять местами. Из многочисленных экспериментов с использованием этой методики приводим следующие:

Опыт 1. Режим «говорит бессознательное, рисует сознание». Тема размышлений не задавалась. Минут через двадцать после окончания опыта, находясь уже в состоянии бодрствования, испытуемый вдруг («к слову») произнес в тех же интонациях одну из фраз, зафиксированных по каналу бессознательного. Он утверждал, что эта мысль пришла ему в голову только сейчас и раньше никогда он такого не думал и не говорил.

Опыт 2. Режим «говорит сознание, рисует бессознательное». Испытуемый решал задачу № 21 [2, с. 60] и высказывал свои соображения по ходу решения. Рисунки с этой задачей связаны не были, более того, два из них явно относились к задаче № 23 [2, с. 61], решать которую испытуемому не поручали.

Опыт 3. Испытуемому в обычном состоянии бодрствования были предъявлены зарисовки, сделанные им в ходе предыдущего опыта. Типичная реакция — удивление: «Да, видимо, рисовал я, но что это может означать, я не знаю». Затем в ОССП, используя рисунки как маркировку потока бессознательного мышления, получили «возврат в прошлое» по каждому из предъявленных рисунков. В ходе такого восстановления и осознания прошлого потока бессознательного мышления испытуемый сообщил ответы к задачам № 23 и № 24 [2, с. 61], т. е. оказалось, что одновременно с дискурсивным решением задачи № 21 испытуемый, сам того не зная, решил еще две задачи, расположенные на соседней странице разворота книги,— задачи, решать которые ему никто не поручал.

Таким образом, обнаруженное особое суггестивное состояние психики человека, отличающееся от сна, гипноза и бодрствования обостренностью сознания, вселяет надежду на появление некоторых новых возможностей выявления и фиксации эвристик, свойственных как дискурсивным, так и неосознаваемым решениям человека.

Список литературы: 1. *Ангушев Г. И., Пушкин В. Н., Фетисов В. М.* Неосознаваемый компонент продуктивного мышления.— В кн.: I международен симпозиум по проблеми на сугестологията. Варна, 5—10 юни 1971, Основни доклади и резюмета. София, 1971, с. 100—101. 2. *Айзенк Г. Ю.* Проверьте свои способности. М., Мир, 1972. 176 с. 3. *Бинэ А.* Психология умозаключений на основании экспериментальных исследований посредством гипнотизма. М., 1889. 270 с. 4. *Бриллюэн Л.* Научная неопределенность и информация. М., Мир, 1966. 271 с. 5. *Дункер К.* Психология продуктивного (творческого) мышления.— В кн.: Психология мышления. М., Мир, 1965, с. 86—234. 6. *Карев А. В., Ильинская Л. С., Ильинский А. П.* Использование гипно- и аутосуггестии для программирования мыслительной деятельности в научном творчестве и обучении.— В кн.: Биокибернетика, моделирование биосистем, бионика. Матер. IV Респ. науч. конференции. Киев, 1970, с. 119. 7. *Карев А. В., Ильинская Л. С., Ильинский А. П.* О некоторых способах программирования интеллектуальной деятельности.— Проблемы бионики. Харьков, 1973, вып. 10, с. 139—144. 8. *Майоров Ф. П.* О физиологических характеристиках сомнамбулистической фазы гипноза.— Физиол. журн. СССР, 1950. т. 36, № 6, с. 649—652. 9. Психологические исследования творческой деятельности. М., Наука, 1975. 252 с. 10. *Райков В. Л.* Влияние глубокого гипноза на резервные возможности памяти и регистрация уровня гипнотического состояния с помощью ЭЭГ.— Психологические исследования в практике врачебно-трудовой экспертизы. Труды ЦИЭТИН МСО РСФСР. М., 1969, вып. 1, с. 128—136. 11. Опыт комплексного исследования изменений, возникающих под влиянием активного самовнушения.— В кн.: Психическая саморегуляция. Алма-Ата, Изд-во Минвуза КазССР, 1973, с. 8—12. Авт.: *А. С. Ромен* и др. 12. *Суслова М. М.* Исследование работоспособности больших полушарий в сомнамбулической фазе гипноза.— Труды ин-та физиологии им. И. А. Павлова. М., 1952, т. 1, с. 296—315. 13. *Werthejmer M.* Experimentalle Studien über das Schen von Bewegung.— Z. Psychol., 1912, Bd 61, S. 161—265.

УДК 62.506.2

В. Н. ЧУДАКОВ

К ВОПРОСУ ОБ ОБРАЗОВАНИИ ФОРМ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА В ОНТОГЕНЕЗЕ

Согласно работам [1; 2] информационные процессы нервной системы представим двумя группами структур мозга: Q — квантовыми и C — классическими. Нормировка сигналов на входах ЦНС этих структур неодинакова, в силу чего разными способами перерабатывается одна и та же входная информация: в дискретном или аналоговом (волновом) коде. Q и C структуры мозга характеризуют все множество типов элементарных базовых структур — детекторов сигналов: ординарных, вероятностных, информационных, волновых [2].

В работе [1] на основании соотношений Q и C структур в организации мозговой деятельности было выделено три основных класса интеллекта — шизо, цикло и эпилепто. Каждый из этих классов связан с одним из феноменов интеллектуальной деятельности. Шизо характеризует рационально-рассудочную деятельность, цикло — эмоционально-чувственную, эпилепто — контрольно-управляющую деятельность, связанную с функцией осознания.

Классы интеллекта передаются при рождении генетической наследственной информацией и характеризуют конституциональную нейроморфологическую организацию мозга человека. В зависимости от условий социальной среды, а также потенциального резерва нейронных структур проявляются и реализуются для каждого класса структуры мозга, соответствующие разным областям интеллектуальной деятельности. Всю интеллектуальную деятельность мозга можно разбить на следующие области. Сенсорика — определяет первичную переработку информации: оценку, измерение, увеличение размерности волн-сигналов и построение образов. Эмоции и потребности определяют окраску информации, генерирование существенно новой информации комбинированием и объединением образов, формированием новых потребностей и форм чувств. Интеллектолика характеризуется анализом и синтезом информации в понятийном знако-символьном коде языка. Аппарат сознания осуществляет фильтрацию, контроль и принятие решений. Моторика характеризует навыки, исполнительные механизмы интеллектуальной деятельности и средств ее выражения. Число потенциалов возможных структур мозга, могущих участвовать в разных областях деятельности, конечно, поэтому участие структур мозга в разных областях интеллектуальной деятельности неравномерно. Можно установить связь нейрофизиологической организации мозга или классов интеллекта и областей интеллектуальной деятельности мозга через коэффициент подавляемости (непроявляемости) тех или иных Q и C структур мозга K (таблица). В силу того что он является функцией параметров социальной среды, то, управляя этими параметрами, можно стимулировать образования самых разнообразных форм интеллектуальной деятельности. Коэффициент подавляемости K лежит в пределах от 0 до $+1$ и представляет собой функцию параметров внешней среды $\varkappa$, α, θ, σ: $K = f(\varkappa, \alpha, \theta, \sigma)$. Параметр $\varkappa$ характеризует уровень развития социальной среды, α — образовательный ценз оператора, θ — характеристика способов и приемов интеллектуальной деятельности в данной среде, σ — характеристика установок, норм и стереотипов в социальной среде. Таким образом, зная классы интеллекта и регулируя параметры внешней среды, мы имеем возможность согласовать мозговые структуры интеллекта с заданной формой интеллектуальной деятельности головного мозга человека, например оператора. В качестве примера приведем формирование интеллектуальной деятельности спортсменов: у шахматистов наиболее развита интеллектолика, у гимнастов и фигуристов, сенсорика и моторика, у спортсменов групповых видов спорта — аппарат осознания и психомоторики [4].

Зависимость нейропсихологической организации структур мозга человека от областей интеллектуальной деятельности

Классы интеллекта	Сенсорика	Эмоциональная окраска и потребности	Интеллектолика	Принятие решений: Аппарат бессознательного	Принятие решений: Аппарат сознания	Психомоторика
	K_1	K_2	K_3	K_4	K_4^*	K_5
Контрольно-управляющий (эпилепто-класс интеллекта)	min	средний	средний	средний	min	средний
Рационально-рассудочный (шизо-класс интеллекта)	max	max	min	max	средний	max
Эмоционально-чувственный (цикло-класс интеллекта)	min	min	max	min	max	min

Основываясь на связи нейрофизиологической организации структур мозга или классов интеллекта и областей интеллектуальной деятельности, можно обоснованно рекомендовать для формирования специфических типов операторской деятельности определенный класс интеллекта человека [1]. Например, для формирования интеллектуальной деятельности оператора-технолога лучше всего подойдет контрольно-управляющий класс интеллекта с минимальными коэффициентами подавляемости K_1, K_4, K_5. Для этого типа оператора в крайнем случае может подойти рационально-рассудочный класс интеллекта с минимальными и средними коэффициентами K_3 и K_4. Для формирования оператора-наблюдателя и контроля лучше всего подойдет эмоционально-рассудочный и контрольно-управляющий класс интеллекта с соответствующими минимальными коэффициентами подавления. Для оператора-исследователя самым лучшим будет рационально-рассудочный класс интеллекта человека, для оператора манипулятора — эмоционально-чувствительный.

Варьируя параметры социальной среды коэффициента подавляемости структур головного мозга Q и C, можно добиться значительного развития специфических областей интеллектуальной деятельности для данного типа операторов.

Список литературы: 1. *Чудаков В. Н.* Классификация нормальных и аномальных форм интеллектуальной деятельности на основании квантово-волновой теории когерентной модели мозга.— Проблемы бионики. Харьков, 1977, вып. 18, с. 46—61. 2. *Чавчанидзе В. В.* К квантово-волновой теории когерентной модели мозга.— Бионика. Киев, 1973, с. 102—112. 3. *Чудаков В. Н.* Классификация нормальных и аномальных форм интеллектуальной деятельности на основании квантово-волновой теории когерентной модели мозга. Сообщ. 2.— Проблемы бионики. Харьков, вып. 19, 1977, с. 137—150. 4. *Шварц В. В.* О наследовании спортивной одаренности.— В кн.: Матер. сентябрьского симпозиума. Соотношение биологического и социального в человеке. М., Наука, 1975, с. 430—440. 5. *Гордеева Н. Д., Деришвили В. М., Зинченко В.* Микро-структурный анализ исполнительной деятельности. М., Изд-во ВНИИТЭ, 1975. 230 с.

СОДЕРЖАНИЕ

РЕФЕРАТЫ

УДК 62.506.2

Математическое описание конечных логических объектов. Шабанов-Кушнаренко Ю. П.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 3—9.

Средствами алгебры конечных предикатов описываются некоторые теоретико-множественные понятия, отношения и операции, определенные на конечных множествах. С помощью этого же аппарата представляются функции К-значной логики: дизъюнкция, конъюнкция, характеристические функции. Полученное описание используется для формального решения ряда логических задач. Материал статьи поясняется примерами.
Список лит: 4 назв.

УДК 007 : 612.843.72

Аксиоматическое введение метрики в бинокулярном зрительном пространстве. ***Сообщение 1.*** Лопатченко Б. К.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 10—13.

Рассмотрен один из подходов к построению математической модели бинокулярного восприятия с использованием метода «черного ящика». Основное внимание сосредоточено на исследовании метрических свойств бинокулярного зрительного пространства.
Список лит.: 3 назв.

УДК 007 : 612.843.72

Аксиоматическое введение метрики в бинокулярном зрительном пространстве. ***Сообщение 2.*** Лопатченко Б. К.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 13—20.

Для исследования метрических свойств бинокулярного зрительного пространства вводится ряд аксиом, проверяемых в психофизическом эксперименте. Из предложенной аксиоматики выводятся аксиомы Гильберта для пространства Лобачевского.
Одна ссылка на лит.: в подстроч. примеч.

УДК 681.33 : 612.822 : 62—506

Модель оптимизирующегося нейрона. Гринченко С. Н., Загускин С. Л.— Проблемы бионики, вып. 23. Респ. межвед. науч-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 20—27.

На основе анализа структурно-метаболических механизмов управления информационными процессами в нервной клетке предлагается модель, адекватно отражающая свойства регуляции чувствительности к входным воздействиям, минимизацию регуляторной составляющей энергетических затрат при смене режимов функционирования нейрона, взаимосвязь изменений геометрических соотношений сомы и аксонного холмика с величиной декремента генераторного потенциала и частотой импульсной активности. Активный характер поведения модели обеспечивается введением блоков одно- и многоканального экстремального регулирования (поисковой оптимизации).
Ил. 1. Список лит.: 9 назв.

УДК 62.506.2.

К вопросу о дискретности слуховой информации. Усенко С. А.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 27—32.

Высказывается предположение о том, что информация, извлекаемая слуховым аппаратом человека из звуковой осциллограммы, конечна. Проводятся психоакустические эксперименты по восприятию человеком дискретной речи,

подтверждающие это предположение. Вырабатываются технические рекомендации по созданию систем синтеза и анализа речи.
Ил. 3. Список лит.: 4 назв.

УДК 62.506.2

Формирование «знаний» универсального решателя задач с помощью системы формирования понятий. *Сообщение 1.* Бондарева Л. Н., Ковалев А. И., Кривошеина Л. С., Ловицкий В. А., Логинов В. А.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 32—39.

Рассматривается система, осуществляющая преобразование условия арифметической задачи с естественного языка на формальный. Для выполнения преобразования система пользуется «знаниями», сформированными с помощью универсальной диалоговой системы формирования понятий.
Табл. 1. Ил. 1. Список лит.: 17 назв.

УДК 519.95

Вопросно-ответная система: представление информации, поиск ответа. Волченко М. В.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 39—46.

Рассматриваются проблемы построения вопросно-ответной системы. Разрабатывается способ представления информации, ориентированный на естественный язык. Дается классификация вопросов. Строится язык, позволяющий применить метод доказательств для поиска ответа. Описан алгоритм поиска, основанный на сведении любого вопроса к стандартному виду.
Список лит.: 9 назв.

УДК 681.327.120888

Алгоритм обработки дискретных наборов для обучающих выборок большого объема. Василенко Ю. А., Роботишин В. И., Шевченко Г. Я.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 46—51.

В данной работе предлагается алгоритм распознавания дискретных наборов, отличительной особенностью которого является построение решающего правила без занесения обучающей выборки в память ЭВМ.
Ил. 2. Табл. 3. Список лит.: 5 назв.

УДК 681.327.120888

Техническая реализация распознающих деревьев с учетом важности признаков. Василенко Ю. А., Пряницкий А. М., Шевченко Г. Я.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 51—58.

Описывается структурная схема и алгоритм функционирования специализированного устройства для распознавания дискретных наборов в виде технической реализации решающего правила, которым является распознающее дерево, и проведения классификации (экзамена) в соответствии с построенным решающим правилом.
Ил. 3. Список лит.: 2 назв.

УДК 007 : 573.6

О построении аналитических описаний сложных геометрических объектов в задачах распознавания геометрических образов. Калюжный А. И.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 58—66.

Рассматриваются вопросы использования аппарата К-логики, позволяющие получать аналитические описания для широкого, с прикладной точки зрения, класса геометрических объектов при решении на ЭВМ задач распознавания

геометрических образов. Предлагается полная система функций, доказывается ее простота. Решением является алгебраическая конструкция в виде произведения функций, которая позволяет строить аналитические описания сложных геометрических объектов, обладающая свойством принимать априорно заданные значения на заданных множествах. Приводится пример.
Ил. 1. Список лит.: 12 назв.

УДК 681.3.06

Программирование на МК-языке типа $<i, k, j, l>$. Колубай С. К.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 66—74.

Работа посвящена исследованию вопросов программирования на МК-языке типа $<i, k, j, l>$ и анализу процесса выполнения МК-программ. Показывается, что программирование на МК-языке типа $<i, k, j, l>$ существенно отличается от программирования на традиционных языках прежде всего тем, что не требуется упорядочивать команды программ в соответствии с порядком их (команд) исполнения. Анализ процесса исполнения МК-программ демонстрирует присущие им возможности по асинхронной и параллельной организации вычислений.
Табл. 3. Ил. 3. Список лит.: 12 назв.

УДК 681.323

Автоматическое преобразование произвольных функций в систему уравнений Шеннона. Дюбко Г. Ф., Замалеев Ю. С.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 74—80.

Рассматривается моделирование деятельности человека по преобразованию математических зависимостей, записанных на языке высокого уровня, в систему уравнений Шеннона (СУШ) — некоторую форму машинно-ориентированного языка. Приводится построение СУШ элементарных функций. Предлагается алгоритм преобразования произвольной функции, заданной аналитически, в форму Шеннона. Рассматривается на примере работа алгоритма.
Табл. 3. Ил. 1. Список лит.: 3 назв.

УДК 681.142.3.

Программный способ масштабирования интегрирующих структур. Четвериков Г. Г., Жиров А. Г.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 80—85.

Предлагается способ, позволяющий автоматизировать практически весь процесс масштабирования интегрирующих структур. Приводится алгоритм анализа формального описания решающих структур для составления соотношений масштабов отдельных агрегатов машинным способом и машинная реализация процесса расчета масштабных коэффициентов.
Табл. 2. Ил. 2. Список лит.: назв. 4.

УДК 681.3.01.02

Параллельная вычислительная система для МК-программ. Колубай С. К.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник, Харьков, издательское объединение «Вища школа», 1979, с. 85—92.

Предлагается структура параллельной вычислительной системы для обработки МК-программ, в которой реализован принцип разделения памяти на пассивную и активную. В пассивной памяти классического типа хранятся данные и результаты. В активной памяти, построенной специальным образом, хранятся обрабатываемые программы (МК-программы). Активная память названа процессоро-памятью. Обсуждаются возможности реализации ячейки процессоро-памяти и всего блока в целом.
Табл. 4. Ил. 4. Список лит.: 2 назв.

УДК 62.506.2

Математическое описание унарных конечных предикатов высших порядков. Шабанов-Кушнаренко Ю. П.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 92—98.

Средствами алгебры конечных предикатов описываются некоторые отношения и унарные предикаты порядков второго и выше. Рассматриваются примеры практического использования данного описания для решений и доказательства утверждений алгебры логики и алгебры множеств.
Список лит.: 5 назв.

УДК 519.766.2

О математическом описании словоизменения существительных. ***Сообщение 1.*** Бондаренко М. Ф., Бондарев В. М.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 98—104.

Предлагается способ формального описания морфологии русского языка, который заключается в накоплении алгебраических формул, выражающих правила грамматики. Приводится пример формализации фрагмента морфологической системы имен существительных.
Список лит.: 4 назв.

УДК 519.766.2

О математическом описании словоизменения существительных. ***Сообщение 2.*** Бондаренко М. Ф., Бондарев В. М.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 105—110.

Средствами алгебры конечных предикатов моделируется словоизменение русских существительных определенного вида. Приводятся примеры решения морфологических задач с помощью построенной модели. Статья служит продолжением работы с тем же названием (сообщение 1) в настоящем сборнике.
Табл. 2. Список лит.: 4 назв.

УДК 62.506.2 : 15 : 612.821.3

Некоторые методы исследования интеллектуальной деятельности человека. ***Сообщение 1.*** Ильинский А. П., Ильинская Л. С., Короп А. В.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 110—116.

Рассматривается применение особого суггестивного состояния психики человека в целях более полного извлечения эвристик при помощи вербальных и графических самоотчетов.
Список лит.: 13 назв.

УДК 62.506.2

К вопросу об образовании форм интеллектуальной деятельности человека в онтогенезе. Чудаков В. Н.— Проблемы бионики, вып. 23. Респ. межвед. науч.-техн. сборник. Харьков, издательское объединение «Вища школа», 1979, с. 117—119.

Рассмотрено соотношение нейропсихической организации мозга (классы интеллекта) с развитием и формированием областей интеллектуальной деятельности. В зависимости от деятельности развиваются разные структуры мозга человека. Выделены параметры внешней среды, влияющие на развитие разных структур мозга. Учитывая нейропсихологическую организацию мозга и варьируя параметры социальной среды в реализации структур мозга, можно добиться значительного развития специфических областей интеллектуальной деятельности для данного типа оператора.
Табл. 1. Список лит.: 5 назв.