УДК 510.62

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

О МОДЕЛИРОВАНИИ СЛОВА И ПРЕОБРАЗОВАНИИ ЕГО ФОРМ

Формальный и приборный аппарат теории интеллекта [1, 2], а также результаты моделирования некоторых конечных функций и отношений [3—5] дают возможность приступить к моделированию процессов обработки слов. Под термином «слово» мы понимаем любые конечные последовательности, составленные из букв произвольного конечного алфавита. Длина слов и число букв алфавита в теории интеллекта ограничиваются значениями параметров n_0 и κ_0 универсальной алгебры конечных предикатов. Понятие «слово» охватывает любые слова, предложения и тексты естественного языка (например, русского), которыми оперирует человеческий интеллект. Процессы обработки слов лежат в основе всей интеллектуальной деятельности человека.

Рассмотрим следующий вопрос: математически описать средствами теории интеллекта отдельно взятое слово, например *мама*. Очевидно, искомое описание будет получено, если мы сформируем для данного слова соответствующее ему уравнение теории интеллекта. Сама запись слова *мама* не может служить его описанием, поскольку она не является уравнением, а другими средствами описания, кроме уравнений, в теории интеллекта мы не располагаем. Казалось бы, в качестве описания заданного слова можно использовать уравнение $X^{м}Y^{а}Z^{м}t^{а} = 1$ **(а)**, поскольку единственным его решением служит вектор (м, а, м, а).

Однако такое описание оказывается неполным. Действительно, для того чтобы иметь возможность записать решение уравнения (а) в виде вектора (м, а, м, а), нужно заранее знать, что x есть первая переменная, y — вторая, z — третья, t — четвертая. В заданном же уравнении (а) ввиду коммутативности операции конъюнкции информация о способе нумерации переменных отсутствует. Это уравнение можно записать в виде

$$t^{а}\, x^{м}\, y^{а}\, z^{м} = 1 \qquad \textbf{(б)},$$

при этом порядок расположения переменных будет уже иным.

Для полноты формального описания слова необходимо, чтобы соответствующее ему уравнение содержало в себе информацию о порядке следования букв в слове. Этого можно достичь, если рассматривать слово как функцию $x=f(s)$ зависимости буквы x от номера s ее места в слове. Условимся нумеровать буквы слова слева направо. Тогда слово *мама* может быть задано в виде функции

Соответствующее этому уравнение имеет вид

$$S^1X^{\text{м}}VS^2X^{\text{а}}VS^3X^{\text{м}}VS^4X^{\text{а}}=1.$$

В общем случае слово $X_1X_2\ \dots\ X_n$ запишем так:

$$s^1(x\approx x_1)\mathrm{V}\,s^2(x\approx x_2)\mathrm{V}\dots\mathrm{V}\,s^n(x\approx x_n)=1. \qquad (1)$$

Формулу (1) назовем уравнением слова $x_1x_2\ \dots\ x_n$. Последовательность букв $x_1x_2\ \dots\ x_n$ будем называть записью слова, рассматривая ее не как само слово, а лишь как его обозначение. Под словом же теперь будем понимать функцию $x=f(s)$, задаваемую уравнением (1), где $x_1,\ x_2\ \dots\ x_n$ — некоторые буквенные константы, играющие роль параметров уравнения.

Как физический объект запись слова может существовать в двух различных формах: параллельной (пространственной) и последовательной (временной). Примерами параллельного представления записи слова могут служить: запись слова в книге; двоичный код, хранящийся в регистре вычислительной машины. Примеры последовательного представления записи слова: звучащая речь; двоичный код, передаваемый по проводнику. Отправляясь от уравнения (1), построим устройства, преобразующие запись слова из параллельной формы в последовательную и обратно.

В составе этих устройств потребуются ячейки памяти для хранения буквы, которые мы построим по принципу триггера с раздельными входами при двоичном его кодировании. Отмеченная таблица переходов этого триггера представлена ниже. Здесь x — троичный входной сигнал триггера, значения которого представлены двухразрядными двоичными кодами $(x_2,\ x_1)$; $y_1,\ y_2$ — двоичные прямой и инверсный входные сигналы; u — двоичное состояние триггера; t — дискретное время.

Мы изложили обычную интерпретацию триггера со счетным входом. Теперь посмотрим на этот триггер с иной точки зрения с позиций теории интеллекта. Будем считать, что триггер оперирует не с двоичными кодами сигналов, а с буквами a_1, a_2, представленными своими узнаваниями. Входной сигнал по-прежнему будем обозначать буквой x, полагая, что x принимает значения a_1, a_2. Младший разряд x_1 двоичного кода прежнего входного сигнала рассматриваем как узнавание x^{a_1}, старший x_2 — как узнавание x^{a_2}. Коду 00 теперь соответствует отсутствие входного сигнала $(x^{a_1}=0,\ x^{a_2}=0)$, коду 01 — сигнал $x=a_1$ $(x^{a_1}=1,\ x^{a_2}=0)$, коду 10 — сигнал $x=a_2$ $(x^{a_1}=0,\ x^{a_2}=1)$. Выходной сигнал триггера обозначим буквой y, полагая $y^{a_1}=y_1$, $y^{a_2}=y_2$. Если $y_1=1$ и $y_2=0$, то $y=a_1$, если же $y_1=0$ и $y_2=1$, то $y=a_2$. Случаю, когда y не существует, соответствуют нулевые значения сигналов y_1, y_2. Состояние

$y_2(t-1)$	1	0
$y_1(t-1)$	0	1
$x(t)$	$u(t-1)$	
	0	1
0 0	0	1
0 1	0	0
1 0	1	1

триггера будем обозначать буквой a, полагая, что a принимает значения a_1, a_2. Принимаем $a^{a_1} = \overline{u}$, $a^{a_2} = u$, тогда $a = y$. Будем считать возможным случай, когда $a^{a_1} = 0$ и $a^{a_2} = 0$, полагая, что при этом ячейка не хранит в своей памяти никакой буквы (очищена от информации).

Связь между узнаваниями сигналов при принятой интерпретации триггера описывается диаграммами Вейча.

Согласно этим диаграммам, если сигнал x не существует, то триггер сохраняет прежнее состояние, т. е. $a(t)^{a_1} = a(t-1)^{a_1}$ и $a(t)^{a_2} = a(t-1)^{a_2}$. Если $x = a_1$, то триггер из любого состояния переходит в состояние $a(t) = a_1$, если же $x = a_2$, то $a(t) = a_2$. В случае, когда $x(t)$ или $a(t-1)$ многозначные, т.е. $x(t)^{a_1} = x(t)^{a_2} = 1$ или $a(t-1)^{a_1} = a(t-1)^{a_2} = 1$, состояние триггера $a(t)$ оставлено неопределенным (в соответствующих ячейках диаграммы проставлены прочерки).

$a(t-1)^{a_1}$, $a(t-1)^{a_2}$

		00	01	11	10
$x(t)^{a_2}$	00	0	0	—	1
	01	0	0	—	0
$x(t)^{a_1}$	11	—	—	—	—
	10	1	1	—	1

$a(t)^{a_1}$

$a(t-1)^{a_1}$, $a(t-1)^{a_2}$

		00	01	11	10
$x(t)^{a_2}$	00	0	1	—	0
	01	1	1	—	1
$x(t)^{a_1}$	11	—	—	—	—
	10	0	0	—	0

$a(t)^{a_2}$

$a(t-1)^{a_1}$, $a(t-1)^{a_2}$

		00	01	11	10
$x(t)^{a_2}$	00	0	0	1	1
	01	0	0	0	0
$x(t)^{a_1}$	11	1	1	1	1
	10	1	1	1	1

$a(t)^{a_1}$

$a(t-1)^{a_1}$, $a(t-1)^{a_2}$

		00	01	11	10
$x(t)^{a_2}$	00	0	1	1	0
	01	1	1	1	1
$x(t)^{a_1}$	11	1	1	1	1
	10	0	0	0	

$a(t)^{a_2}$

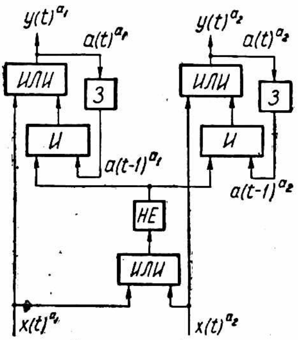
Ниже показан вариант доопределения диаграмм и склеивания их ячеек, обеспечивающий получение экономных форм для явной записи сигналов $a(t)^{a_1}$ и $a(t)^{a_2}$.

Переходя от диаграммы к формулам, имеем

$$x(t)^{a_1} \vee \overline{x(t)^{a_1} \vee x(t)^{a_2}}\, a(t-1)^{a_1} = a(t)^{a_1};$$

$x(t)^{a_2} \vee x(t)^{a_1} \vee x(t)^{a_2} a(t-1)^{a_2} = a(t)^{a_2}$ (2). Соответствующая этим формулам схема триггера представлена на рис. 1. По сравнению с обычной схемой триггера со счетным входом (рис. 2) эта схема выглядит неэкономной, однако она дает ключ к построению нужной нам ячейки памяти для хранения буквы произвольного конечного алфавита. Схема такой ячейки и ее условное обозначение представлены на рис. 3, *а*, *б*. Она отличается от схемы рис. 1. лишь тем, что переменные x, y, u принимают значения a_1, a_2, $\ldots$, a_k.

Назовем такую схему ячейкой буквенной памяти. Она функционирует следующим образом. При подаче в момент времени t



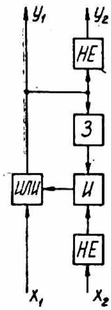


Рис. 1. Схема триггера, построенная по формулам (2)

Рис. 2. Триггер со счетным входом

буквы a на вход x ячейка в тот же момент повторяет эту букву на выходе y и хранит ее в своей памяти в течение следующего такта времени, вплоть до момента $t+1$. Если же на вход ячейки не поступает никакой буквы (все узнавания x^{a_1}, x^{a_2}, $\ldots$, x^{a_k} — нулевые), то на выходе формируется буква, хранимая ячейкой в предыдущем такте времени. Эта буква продолжает сохраняться ячейкой и в следующем такте времени. Ячейка буквенной памяти может быть описана в явном виде следующей системой уравнений:

$$x(t)^{a_i} \mathrm{V} x(t)^{a_1} \mathrm{V} x(t)^{a_2} \mathrm{V} \ldots \mathrm{V} x(t)^{a_k} a(t-1)^{a_i} = a(t)^{a_i}, \qquad (3)$$

где индекс i пробегает значения от I до K. При импульсном представлении сигналов схему ячейки буквенной памяти можно существенно упростить, заменив в ней все блоки разделения в верхнем ряду схемы (рис. 3) узлами. Такая замена допустима, поскольку на входы этих блоков никогда не поступает более одного импульса одновременно; кроме того, исключено образование ложных цепей.

Ячейку буквенной памяти можно также использовать для совершенно иной цели: запоминания множества, представленного в виде булева вектора. Если вектор содержит хотя бы один единичный компонент (непустое множество), то он запоминается схемой и хранится ею в течение одного последующего такта.

Если же все компоненты вектора нули (пустое множество), то схема продолжает помнить прежний вектор.

Теперь, когда ячейка памяти для хранения буквы построена, возвратимся к задаче разработки устройства, преобразующего запись слова из параллельной формы в последовательную. Схе-

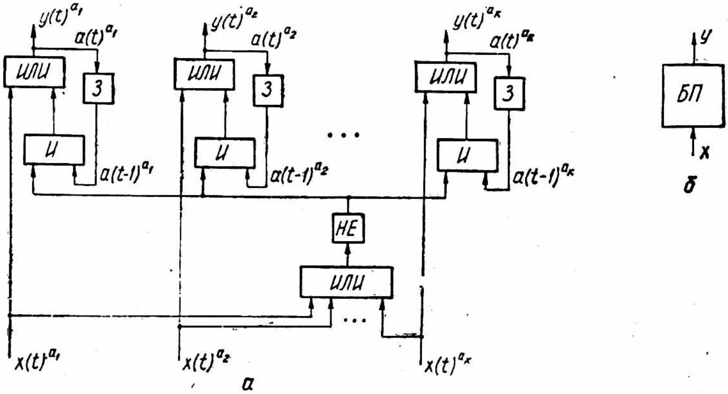
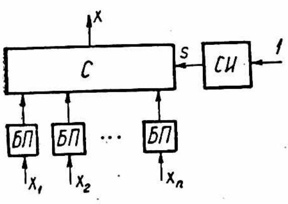


Рис. 3. Схема ячейки памяти для хранения буквы

ма устройства дана на рис. 4. В ней, кроме ячеек буквенной памяти использован селектор С и счетчик импульсов СИ. В начальный момент времени ($t=0$) в ячейки памяти устройства за-



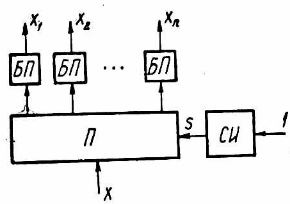


Рис. 4. Устройство, преобразующее запись слова из параллельной формы в последовательную

Рис. 5. Устройство, преобразующее запись слова из последовательной формы в параллельную

сылается запись $x_1 x_2 ... x_n$ слова в параллельной форме. На вход счетчика подаются сигналы от генератора тактовых импульсов. На выходе счетчик, работающий в данном случае в режиме датчика управляющих импульсов, в моменты времени $t=1, 2, ..., n$ формирует номера $S=1, 2, ..., n$ букв слова. В те же моменты времени на выходе селектора последовательно во времени появляются буквы $x=x_1, x_2, ..., x_n$ записи слова в последовательной форме. Уравнение селектора с точностью до обозначений совпадает с уравнением слова (1).

Нетрудно построить также устройство, осуществляющее обратное преобразование записи слова из последовательной формы в параллельную. В схеме устройства (рис. 5) используется счетчик импульсов ячейки буквенной памяти и переключатель П. На вход переключателя последовательно во времени ($t=1,2, ..., n$) поступают буквы $x_1, x_2, ..., x_n$ слова. В те же моменты со счетчика в переключатель поступают номера букв $s=1, 2, ..., n$. Переключатель распределяет буквы по своим выходам: букву x_1 в момент $t=1$ посылает на первый выход, букву X_2 в момент $t=2$ — на второй и т. д.

Список литературы: 1. *Шабанов-Кушнаренко Ю. П.* Об универсальной алгебре конечных предикатов. — АСУ и приборы автоматики, 1980, вып. 55, с. 69—74. 2. *Шабанов-Кушнаренко Ю. П.* О переключательных цепях теории интеллекта. — Проблемы бионики, 1980, вып. 25, с. 11—18. 3. *Шабанов-Кушнаренко Ю. П.* О моделировании алфавитных операторов средствами теории интеллекта. — Проблемы бионики, 1981, вып. 26, с. 3—10. 4. *Шабанов-Кушнаренко Ю. П.* О моделировании конечных множеств средствами теории интеллекта. — Математические методы анализа динамических систем, 1980, вып. 4, с. 76—80.

Поступила в редколлегию 01.10.80.