

УДК 510.644 + 004.41

## МЕТОД ОБРАБОТКИ НЕПОЛНЫХ ДАННЫХ НА ОСНОВЕ ТРЕХЗНАЧНОЙ ЛОГИКИ\*

М. Н. Филиппов

*Институт автоматизации и электрометрии СО РАН,  
630090, г. Новосибирск, просп. Академика Коптюга, 1  
E-mail: michael@idisys.iae.nsk.su*

Представлен метод обработки данных, основанный на использовании варианта трехзначной логики, позволяющий учитывать недостоверную и неполную информацию, а также пример практического применения предложенного метода при разработке системы управления движением поездов Новосибирского метрополитена.

*Ключевые слова:* трехзначная (троичная, тернарная) логика, недостоверная и неполная информация, система управления, Новосибирский метрополитен.

**Введение.** При разработке систем управления сложными объектами зачастую возникает необходимость обеспечить предсказуемо безопасное поведение в условиях поступающей неполной или недостоверной информации. В первую очередь это относится к распределенным системам с большим количеством входных датчиков, связь с которыми иногда теряется, а сами датчики выходят из строя. При обработке подобных ситуаций система, очевидно, должна различным образом реагировать на достоверную информацию (например, «напряжение ниже 220 В», «напряжение больше или равно 220 В») и ситуацию «отсутствует связь с датчиком напряжения».

Если с технической стороны диагностику поступающей информации можно организовать многими хорошо известными и отработанными способами (такими как установка дублирующих датчиков и проверка того, что разница в их показаниях не превышает определенного предела, смещение нуля измеряемого тока или напряжения для определения обрыва провода и т. п.), то для корректной обработки необходимо использование специальных математических методов, позволяющих наряду с точными данными оперировать неполной или недостоверной информацией.

В качестве примера рассмотрим логические условия, необходимые гипотетической программе-ассистенту водителя при анализе упрощенной задачи «разрешен ли проезд на основании поступающей информации о текущем состоянии светофора и возможных препятствиях на пути»:

— «проезд разрешен», если «путь свободен» ( $F = \text{True}$ , F от англ. Free) и «горит разрешающий сигнал светофора» ( $G = \text{True}$ , G от англ. Green), при этом вся информация является достоверной ( $RF = RG = \text{True}$ , R от англ. Reliable):

$$\text{if } (RF \text{ and } F \text{ and } RG \text{ and } G) \text{ then ...;}$$

— «проезд не разрешен», если «не свободен путь» ( $F = \text{False}$ ), и эта информация является достоверной ( $RF = \text{True}$ ), или «не горит разрешающий сигнал» ( $G = \text{False}$ ), и эта информация является достоверной ( $RG = \text{True}$ ):

$$\text{if } ((RF \text{ and not } F) \text{ or } (RG \text{ and not } G)) \text{ then ...;}$$

---

\*Работа выполнена при частичной поддержке программы ОЭММПУ «Проблемы управления и безопасности энергетики и технических систем» (проект № 2.4 «Развитие программно-аппаратных систем и средств мониторинга, управления и поддержки принятия решений в энергетике»).

Таблица 1

## Возможные состояния (двоичный пример)

RF	F	RG	G	Решение
1	1	1	1	Р
1	1	1	0	З
1	1	0	1	Н
1	1	0	0	Н
1	0	1	1	З
1	0	1	0	З
1	0	0	1	З
1	0	0	0	З
0	1	1	1	Н
0	1	1	0	З
0	1	0	1	Н
0	1	0	0	Н
0	0	1	1	Н
0	0	1	0	З
0	0	0	1	Н
0	0	0	0	Н

— «недостаточно информации для принятия решения» — поведение по умолчанию, например остановиться и информировать водителя о невозможности автоматического управления в сложившейся ситуации:

if ((not RF and not RG) or (not RF and G) or (not RG and F)) then ...;

В этом случае таблица возможных состояний (табл. 1) насчитывает 16 комбинаций значений входных параметров (для вариантов решения приняты следующие обозначения: Р — «разрешено», З — «запрещено», Н — «недостаточно информации»).

Очевидно, что размер таблицы и сложность программирования логических условий отчасти объясняются избыточностью описания, поскольку из четырех комбинаций значений, возможных для каждой пары битов «достоверность/информация» (в данном примере RF/F и RG/G), логически независимыми являются только три варианта, а комбинации «недостоверно/истина» и «недостоверно/ложь» должны обрабатываться системой одинаково.

Таким образом, если использовать естественное в подобных случаях кодирование возможных значений переменных и результатов вычисления логических функций, включающее три варианта: «истина» (варианты обозначений:  $t, 1, P$ ), «ложь» ( $f, 0, N$ ) и дополнительное истинностное значение «неизвестно» или «не определено» ( $u, \frac{1}{2}, U$ ), которое характеризует отсутствие информации о точном значении, а также разработать адекватный математический аппарат, можно не только вдвое сократить количество аргументов в данном примере (отпадет необходимость в RF и RG), но и переписать логические условия, значительно их упростив (табл. 2).

Благодаря уменьшению сложности программного обеспечения, такой подход позволит добиться общего повышения надежности, что особенно актуально при создании ав-

Таблица 2

## Возможные состояния (троичный пример)

if_ternary (F and G)		and	$t$	$f$	$u$
then_true	«проезд разрешен»;	$t$	$t$	$f$	$u$
then_false	«проезд не разрешен»;	$f$	$f$	$f$	$f$
then_unknown	«недостаточно информации»;	$u$	$u$	$f$	$u$

томатизированных систем управления, насчитывающих большое количество различных внешних датчиков, а поступающая информация обрабатывается логическими условиями, включающими десятки (порой и сотни) аргументов, для которых крайне затруднительно вручную выписать все варианты, учитывающие возможную недостоверность данных. Дополнительно это даст механизм автоматического переключения на работу в безопасном режиме только тех подзадач, для корректного выполнения которых в настоящий момент недостаточно информации, а также позволит локализовать некоторые типичные ошибки программирования и спецификаций аппаратной части, например возможные комбинации и диапазоны значений входных параметров, пропущенные в процедурах обработки данных и используемой технической документации.

**Постановка задачи.** Целью данной работы является создание полезного для практического применения (в частности, для программирования автоматизированных систем управления) математического аппарата троичной (трехзначной, тернарной) логики, представляющего собой расширение классической двузначной логики дополнительным истинностным значением ( $u$ ), которое интерпретируется как отсутствие в данный момент информации о точном значении переменной или нарушение одного из заданных ограничений значения переменной во время вычисления выражения, а также разработка программных средств, содержащих необходимые процедуры преобразования внешних данных к внутреннему представлению и их дальнейшей обработки в соответствии с этим математическим аппаратом.

Например, потеря связи с датчиком, предоставляющим информацию о напряжении питания  $V_1$ , приводит к изменению атрибутов соответствующего  $V_1$  объекта, и в дальнейшем результат сравнения  $V_1 < 210$  принимает значение не «истина» или «ложь», а «неизвестно» (причина — «отсутствие информации»). Аналогично полученные входные значения  $R_1 = 0$  и  $R_2 = 1$ , соответствующие выходам параллельно соединенных реле (контролирующая цепь), не могут быть различными в корректно функционирующей системе («заданное ограничение» —  $R_1 = R_2$ ), в результате чего обе троичные переменные также принимают значение «неизвестно» (причина — «нарушение ограничения»).

Дальнейшая обработка преобразованных к троичному представлению входных данных производится в соответствии с логическими выражениями, записанными с помощью стандартного набора операторов (not, and, or, == и т. д.), которые заданы на расширенном (истинностным значением «неизвестно») множестве возможных значений аргументов и результата таким образом, что сохраняют при этом классический результат вычисления, если все аргументы определены («истина» или «ложь»).

**История вопроса.** Обращаясь к истории возникновения трехзначных логик, следует отметить, что принцип двузначности (бивалентности), согласно которому любое высказывание является или только истинным или только ложным, подвергался сомнению уже в античности. Аристотель в девятой главе трактата «Об истолковании» поднимает проблему «будущей случайности» (верно ли, что относительно завтрашнего морского сражения истинно или ложно утверждение «завтра морское сражение произойдет» или отрицание

Таблица 3

## Импликация У. Оккама

$p$	$q$	$p \rightarrow q$
$f$	$f$	$t$
$f$	$u$	$t$
$f$	$t$	$t$
$u$	$f$	$u$
$u$	$u$	$u$
$u$	$t$	$t$
$t$	$f$	$f$
$t$	$u$	$u$
$t$	$t$	$t$

«завтра морское сражение не произойдет?»), оказавшуюся в дальнейшем чрезвычайно продуктивной для появления нескольких вариантов многозначных логик [1].

Крупнейший философ и логик средневековья У. Оккам посвятил этой проблеме специальный трактат, в котором рассматриваются три значения истинности («истинно», «ложно» и «неопределенно»), используемые для построения логических выводов, в том числе и при анализе конструкций вида «если  $p$ , то  $q$ » (его таблица истинности импликации приведена по результатам исследования Ф. Бенера с использованием принятых в данной статье обозначений (табл. 3)). Некоторые исследователи считают именно У. Оккама предшественником трехзначной логики [2].

Следует также упомянуть введенную Д. Скотом классификацию утверждений, включавшую помимо истинных и ложных также и «неопределенные», в частности высказывания относительно будущих случайных событий. В начале XX века шотландский философ Х. МакКолл изучал названные им «переменными» высказывания, которые иногда истинны, а иногда ложны, при этом истинностные значения он связывал с исчислением вероятностей. Американский философ и логик Ч. Пирс уже в 1905 г. использовал истинностные таблицы для трехзначной логики, обосновывая третье истинностное значение как промежуточное между определенной истиной и определенной ложью. Элементы многозначной логики содержатся и в работах русского математика Н. А. Васильева по «воображаемой логике» 1910–1914 гг. (более подробно история вопроса приведена в [1, 2]).

Наконец, в 1920 г. появляется первая законченная система трехзначной логики, разработанная выдающимся польским логиком и философом Я. Лукасевичем (1878–1956 гг.), предложившим собственное решение аристотелевской проблемы истинностного статуса высказываний о будущих случайных событиях. Суть идеи состоит в том, что альтернатива «завтра произойдет морское сражение или завтра не произойдет морское сражение» уже сегодня истинна, но ни высказывание «завтра будет морское сражение», ни высказывание «завтра не будет морское сражение» сегодня истинными не являются. Для формального описания подобных утверждений вводится третье истинностное значение, которое в отличие от 1 («истина») и 0 («ложь») обозначается  $\frac{1}{2}$  и интерпретируется как «возможность». Оставляя классические значения для импликации ( $\rightarrow$ ) и отрицания ( $\neg$ ), когда аргументы принимают значения из множества  $\{1, 0\}$ , Лукасевич следующим образом доопределяет

Таблица 4

## Операторы Я. Лукасевича

$x$	$y$	$\neg x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \equiv y$
0	0	1	0	0	1	1
0	$1/2$		0	$1/2$	1	$1/2$
0	1		0	1	1	0
$1/2$	0	$1/2$	0	$1/2$	$1/2$	$1/2$
$1/2$	$1/2$		$1/2$	$1/2$	1	1
$1/2$	1		$1/2$	1	1	$1/2$
1	0	0	0	1	0	0
1	$1/2$		$1/2$	1	$1/2$	$1/2$
1	1		1	1	1	1

логические связи [3]:

$$(1 \rightarrow 1/2) = (1/2 \rightarrow 0) = 1/2,$$

$$(0 \rightarrow 1/2) = (1/2 \rightarrow 1/2) = (1/2 \rightarrow 1) = 1,$$

$$\neg 1/2 = 1/2.$$

Посредством исходных связей также определяются следующие операторы:

$$x \vee y = (x \rightarrow y) \rightarrow y \quad (\text{дизъюнкция}),$$

$$x \wedge y = \neg(\neg x \vee \neg y) \quad (\text{конъюнкция}),$$

$$x \equiv y = (x \rightarrow y) \wedge (y \rightarrow x) \quad (\text{эквивалентность}).$$

Переписав результаты в виде таблицы истинности основных операторов (табл. 4) и проанализировав ее, можно обнаружить, что предложенный Я. Лукасевичем вариант трехзначной логики (далее  $\mathcal{L}_3$ ) не позволяет непосредственно использовать введенное дополнительное истинностное значение ( $1/2$ ) в качестве индикатора «значение неизвестно» и обеспечить тем самым корректную обработку ситуаций типа «отсутствие информации по одному или нескольким аргументам». В первую очередь это относится к операторам импликации ( $\rightarrow$ ) и эквивалентности ( $\equiv$ ). Например, компьютерная программа, использующая оператор  $\equiv$  из  $\mathcal{L}_3$  для сравнения входных данных, значения которых не определены, будет ошибочно полагать, что они всегда равны:  $x \equiv y$  — «истина» при  $x = y = 1/2$ . Аналогичный пример несложно привести и для конструкции вида «если ..., то ...» ( $\rightarrow$  из  $\mathcal{L}_3$ ). Причина этого заключается в том, что Я. Лукасевич интерпретировал введенное третье истинностное значение не как полностью независимое или неизвестное состояние, а как «возможность» — промежуточное значение между «истиной» и «ложью», отчего, хотя с формальной точки зрения  $\mathcal{L}_3$  и выглядит безупречной, возникают определенные трудности с ее интуитивной интерпретацией даже по отношению к простым логическим операторам, например конъюнкции: «возможно, что завтра произойдет сражение» ( $1/2$ ) и «возможно, что завтра оно не произойдет» ( $1/2$ ); согласно  $\mathcal{L}_3$  также «возможно» ( $1/2 \wedge 1/2 = 1/2$ ), хотя интуитивно понятно, что сражение не может произойти и не произойти одновременно.

Работа Я. Лукасевича положила начало появлению целой серии новых трехзначных логик, отличающихся как начальными предпосылками, так и таблицами истинности. Сре-

Таблица 5

Операторы С. Клини

$x$	$\neg x$	$\vee$	$t$	$f$	$u$	$\wedge$	$t$	$f$	$u$	$\rightarrow$	$t$	$f$	$u$	$\equiv$	$t$	$f$	$u$
$t$	$f$	$t$	$t$	$t$	$t$	$t$	$t$	$f$	$u$	$t$	$t$	$f$	$u$	$t$	$t$	$f$	$u$
$f$	$t$	$f$	$t$	$f$	$u$	$f$	$f$	$f$	$f$	$f$	$t$	$t$	$t$	$f$	$f$	$t$	$u$
$u$	$u$	$u$	$t$	$u$	$u$	$u$	$u$	$f$	$u$	$u$	$t$	$u$	$u$	$u$	$u$	$u$	$u$

ди наиболее значимых можно отметить логику Д. А. Бочвара [4], построенную с целью разрешения парадоксов классической математики и интерпретирующую третье истинностное значение как «бессмысленность»; предложенные для интерпретации законов квантовой механики логики П. Детуш-Феврие и Г. Рейхенбаха; интуиционистские логики А. Гейтинга и Б. Собочинского; паранепротиворечивые логики Ф. Асеньо, Л. И. Розоноэра и А. Сетте, а также многие другие [1, 5, 6], подробное рассмотрение применимости которых к поставленной задаче корректной обработки неполных данных выходит за рамки данной работы. Остановимся лишь на трехзначной логике С. Клини (далее  $\mathcal{K}_3$ ), в основе которой лежит теория частично рекурсивных функций с аргументами, представляющими собой математические утверждения, которые хотя и являются истинными ( $t$ ) или ложными ( $f$ ), но недоказуемы и неопровержимы, при этом третье истинностное значение ( $u$ ) интерпретируется как «не определено» или «неизвестно». Таблицы истинности логических операторов (табл. 5), в свою очередь, определяются как самые сильные из возможных регулярных расширений классических двузначных таблиц: столбец (строка) содержит  $t$  в строке (столбце) для  $u$  только при условии, что этот столбец (строка) состоит целиком из  $t$ ; аналогично для  $f$  [7].

**Метод решения.** Таблицы истинности логических операторов (табл. 6), формально совпадающие с  $\mathcal{K}_3$  и построенные на основе схожих принципов, получены в лаборатории нечетких технологий Института автоматики и электрометрии СО РАН при разработке методов повышения надежности программного обеспечения. Приведем два наиболее важных принципа построения: во-первых, это сводимость к классической двузначной логике в случае, когда все аргументы функции принимают значения только «истина» ( $t$ ) или «ложь» ( $f$ ):

$$F_{\text{ternary}}(x_1, x_2, \dots) \equiv F_{\text{binary}}(x_1, x_2, \dots), \quad \text{если все } x_i \in \{t, f\};$$

во-вторых, если аргумент функции «неизвестен» ( $u$ ) и при его изменении как на «истину», так и на «ложь» значение функции не изменяется ( $C$ ), то результат вычисления принимается равным данной константе:

$$\text{если } F(\dots, t, \dots) \equiv F(\dots, f, \dots) \equiv C, \quad \text{то } F(\dots, u, \dots) \equiv C,$$

в противном случае результат вычисления также «неизвестен»:

$$F(\dots, u, \dots) \equiv u.$$

Полученный математический аппарат был использован при создании SCADA-системы (Supervisory Control and Data Acquisition), предназначенной для разработки автоматизированных систем управления повышенной надежности [8]. Данная программная среда (SC2 framework) предоставляет разработчику стандартный набор функций, позволяющий записывать выражения в привычном виде при операциях как с логическими (not, and, or, == и т. д.), так и с численными (<, <=, ==, !=, >, >= и т. п.) переменными, существенно облегчая тем самым обработку неполной и недостоверной информации в соответствии с описанными выше принципами.

Таблица 6

## Основные логические операторы

$x$	$y$	$\neg x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \equiv y$
$f$	$f$	$t$	$f$	$f$	$t$	$t$
$f$	$u$		$f$	$u$	$t$	$u$
$f$	$t$		$f$	$t$	$t$	$f$
$u$	$f$	$u$	$f$	$u$	$u$	$u$
$u$	$u$		$u$	$u$	$u$	$u$
$u$	$t$		$u$	$t$	$t$	$u$
$t$	$f$	$f$	$f$	$t$	$f$	$f$
$t$	$u$		$u$	$t$	$u$	$u$
$t$	$t$		$t$	$t$	$t$	$t$

Модуль графического интерфейса пользователя SC2 также содержит встроенные средства отображения дополнительного состояния «отсутствие информации об объекте». Например, размещенный на экране индикатор  $\text{alarm} = ((V < 215) \text{ or } (V > 225))$  помимо описанных программистом красного (True) и зеленого (False) цветов также может становиться бирюзовым (цвет Undefined состояния по умолчанию, возможно переопределение) в случае, если значение  $V$  (и, следовательно,  $\text{alarm}$ ) помечено как «неизвестное»: т. е. нет связи с соответствующим модулем ввода—вывода или полученное значение выходит за установленные разработчиком пределы ( $200 < V < 250$ ).

Работа над SCADA-системой началась в 2004 г. в рамках проекта по созданию автоматизированной системы диспетчерского управления движением поездов Новосибирского метрополитена. К настоящему моменту созданное программное обеспечение, одной из отличительных особенностей которого является использование трехзначной логики во всех операциях контроля действий оператора и анализа состояния оборудования, успешно эксплуатируется на станциях «Березовая роща», «Площадь Гарина-Михайловского», «Заельцовская» и «Красный проспект» [9].

**Заключение.** Предложен метод обработки данных на основе варианта трехзначной логики, представляющего собой расширение классической бинарной логики дополнительным истинностным значением, которое интерпретируется как отсутствие в данный момент информации о точном значении переменной или нарушение одного из заданных ограничений значения переменной во время вычисления выражения. Преимуществом метода является то, что он позволяет наряду с точными данными оперировать неполной и недостоверной информацией, существенно упрощая при этом запись логических выражений.

Полученный математический аппарат использован при создании SCADA-системы, предназначенной для разработки автоматизированных систем управления повышенной надежности. Программное обеспечение, созданное на ее основе, в настоящее время используется в системе управления движением поездов Новосибирского метрополитена.

Практическое применение метода, основанного на использовании описанного варианта трехзначной логики, дало возможность существенно упростить разработку программного обеспечения и повысить надежность всей системы управления, а также обеспечить ее корректное поведение в условиях поступающей неполной и недостоверной информации.

## СПИСОК ЛИТЕРАТУРЫ

1. Карпенко А. С. Многозначные логики. Сер. Логика и компьютер. Вып. 4. М.: Наука, 1997. 223 с.
2. Стяжкин Н. И. Формирование математической логики. М.: Наука, 1967. 508 с.

3. **Карпенко А. С.** Логика Лукасевича и простые числа. М.: Наука, 2000. 319 с.
4. **Бочвар Д. А.** Об одном трехзначном исчислении и его применении к анализу парадоксов классического расширенного функционального исчисления // Мат. сб. 1938. **4(46)**, № 2. С. 287–308.
5. **Гетманова А. Д.** Логика. М.: ИКФ Омега-Л; Высш. шк., 2002. 416 с.
6. **Галиев Ш. И.** Математическая логика и теория алгоритмов. Казань: Изд-во КГТУ им. А. Н. Туполева, 2002. 334 с.
7. **Клини С. К.** Введение в метаматематику: Пер. с англ. М.: Изд-во иностр. лит., 1957. 524 с.
8. **Белоконь С. А., Филиппов М. Н.** Метод построения многоплатформенной открытой модульной SCADA-системы // Вест. НГУ. Сер. Физика. 2008. **3**, вып. 1. С. 115–125.
9. **Белоконь С. А., Васильев В. В., Золотухин Ю. Н. и др.** Автоматизированная система диспетчерского управления движением поездов в Новосибирском метрополитене // Матер. конф. IV Всерос. школы-семинара молодых ученых «Проблемы управления и информационные технологии». Казань: Изд-во Казан. гос. техн. ун-та, 2008. С. 28–31.

*Поступила в редакцию 23 июля 2009 г.*

---