

УДК 381.326

*КРИВУЛЯ Г.Ф., ШКИЛЬ А.С., КУЧЕРЕНКО Д.Е.*

## **АНАЛИЗ КОРРЕКТНОСТИ ПРОДУКЦИОННЫХ ПРАВИЛ В СИСТЕМАХ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА С ИСПОЛЬЗОВАНИЕМ КВАНТОВЫХ МОДЕЛЕЙ**

---

Рассматривается вопрос автоматизации проверки на корректность базы знаний производственных правил, которая лежит в основе экспертного анализа компьютерной системы. Экспертный анализ того или иного компонента системы основывается на экспертной оценке выбранных диагностических признаков с использованием правил и процедур нечеткой логики. Предлагается квантовая модель кодирования символов многозначного алфавита. Разрабатывается компьютерная программа, которая позволяет выполнить формальную проверку базы производственных правил на корректность и тем самым значительно улучшить качество системы нечеткого экспертного диагностирования в целом.

### **1. Введение**

На сегодняшней день интеллектуальные методы диагностики, основанные на знаниях, являются одними из самых распространенных. Это обусловлено тем, что в рамках классической технической диагностики, необходимо строить формальную модель объекта диагностирования (ОД). Возрастающая сложность компьютерных систем (КС) делает зачастую невозможным построение такой модели. В этом случае используется метод экспертных оценок с привлечением специалистов-экспертов, знания которых позволяют создать модель того или иного компонента компьютерной системы (аппаратного обеспечения, программного обеспечения или персонала).

Методы экспертных оценок базируются на гипотезе, что, используя знания одного (индивидуальные оценки диагностических признаков) или нескольких (коллективные оценки диагностических признаков) специалистов-экспертов, удастся создать модель будущего состояния ОД, близкую к реальной. Зачастую эти невербализуемые знания представляют наибольший интерес для проектируемой экспертной системы диагностирования (ЭСД). Именно они отражают тот профессиональный опыт специалиста, который он накопил в процессе своей профессиональной деятельности в области технической диагностики. Однако данный вид знаний является трудноформализуемым, так как существует у специалиста подсознательно. Именно этот факт и обусловил выбор нечеткой ЭСД для оценки технического состояния ОД.

### **2. Нечеткая экспертная система диагностирования**

Важнейшим компонентом ЭСД является система нечеткого вывода, в основе которой лежит база данных (БД) и база знаний (БЗ). БД содержит исходные количественные и качественные диагностические признаки ОД. База знаний ЭСД содержит факты (данные о диагностических параметрах ОД) и правила, использующие эти факты как основу для принятия решения о диагностировании. Таким образом, БЗ хранит долгосрочные данные, описывающие рассматриваемую область, а БД – промежуточные данные решаемой в данный момент задачи. Система нечеткого вывода, используя исходные данные из БД и знания из БЗ, обеспечивает решение задачи диагностирования КС. Каждому диагностическому признаку, независимо от характера ОД (компонента КС), ставится в соответствие лингвистическая переменная (ЛП), а оценка состояния объекта осуществляется путем нечеткого логического вывода. В качестве алгоритма нечеткого вывода предлагается

использовать алгоритм Мамдани [1], где вывод осуществляется по следующим этапам: фазификация (переход к нечеткости), агрегирование подусловий в нечетких правилах, активизация заключений нечетких правил, аккумулирование заключений нечетких правил и дефазификация (ликвидация нечеткости).

В зависимости от типа ОД и физического смысла диагностических признаков подсистема логического вывода будет использовать соответствующую базу знаний [2]. Эксперт, определив знания (данные и правила), характеризующие проблемную область, обеспечивает полноту и правильность введенных в ЭСД знаний.

В качестве модели представления знаний была выбрана продукционная модель, которая обладает рядом преимуществ: наглядность; модульность (отдельные продукционные правила могут быть добавлены в базу знаний, удалены или изменены независимо от других, что позволяет автоматизировать их проектирование); легкость внесения дополнений и изменений; простота логического вывода.

### **3. Продукционные правила (ПП) как способ представления знаний эксперта**

В настоящее время одной из самых распространенных моделей представления знаний является продукционная модель, в основе которой лежат продукционные правила вида:

$$\text{ЕСЛИ } A_1 \text{ И/ИЛИ } A_2 \text{ И/ИЛИ } \dots \text{ И/ИЛИ } A_n, \text{ ТО } B_i, \quad (1)$$

где  $A_i$  – высказывания (условия, антецедент), к которым применимы логические операции И/ИЛИ;  $B_i$  – высказывание (следствие, консеквент), которое становится истинным при истинности условия.

Синтез продукционных правил состоит из трех этапов:

- составление списка диагностических признаков (критериев оценивания), которые должны быть учтены: они станут лингвистическими переменными в базе нечетких правил;
- составление перечня качественных оценок, которые принимаются к рассмотрению и определяют, когда они являются истинными, а когда ложными: эти оценки станут лингвистическими термами базы правил;
- составление формулировок правил: они должны описывать влияние диагностических признаков ОД на результат диагноза.

Таким образом, синтез БЗ сводится к формализации словесных (вербальных) правил, подготовленных экспертами на этапе словесного описания задачи, в формате, удобном для чтения (1) и для дальнейшего анализа [3].

Несмотря на разный физический смысл, вкладываемый в понятие «диагностические признаки» объекта диагностирования, для всех трех компонентов КС (аппаратура, программное обеспечение и персонал) критерии экспертного оценивания имеют три уровня градации и соответственно термы входных лингвистических переменных будут иметь три значения: «низкий», «средний», «высокий». Термы выходной лингвистической переменной имеют пять значений: «очень низкий», «низкий», «средний», «достаточный», «высокий». ПП, как правило, составляются на основе представлений эксперта об объекте диагностирования и о влиянии значений отдельных входных параметров на результат диагностирования.

Несмотря на разнородность компонентов КС, как объекта диагностирования, и различные способы составления ПП, их база должна удовлетворять единым формальным требованиям корректности, не касающимся смыслового аспекта самих ПП, а именно, быть полной, минимальной, связной и непротиворечивой. Таким образом, целью данной работы является создание эффективного вычислительного алгоритма анализа ПП на корректность.

### **4. Кубическое представление продукционных правил**

Для перехода к анализу ПП на корректность необходимо привести их к форме представления функций алгебры логики, для чего следует выполнить преобразования входных лингвистических переменных, входящих в состав ПП, а точнее сделать термы этих переменных независимыми.

При этом на диагностические признаки и способы их описания накладывается ряд ограничений.

– Входные ЛП (оценки диагностических признаков) должны иметь одинаковое количество термов, а термы разных входных ЛП должны иметь одинаковые диапазоны.

– Продукционные правила должны быть полностью определены, т.е. любому сочетанию значений термов входных ЛП соответствует определенное значение термов выходной ЛП.

Рассмотрим ОД, техническое состояние которого в процессе экспертного диагностирования определяется четырьмя диагностическими признаками. Оценка диагностических признаков (ДП) осуществляется исходя из трехбалльной шкалы: «низкий» (Н), «средний» (С), «высокий» (В). Результат диагноза (РД) имеет пять уровней градации: «очень низкий» (ОН), «низкий» (Н), «средний» (С), «достаточный» (Д), «высокий» (В). Например, если число ДП равно четырем, то число входных ЛП  $n = 4$ , при этом число термов каждой переменной  $m = 3$ , а число термов выходной ЛП равно 5.

Для упрощения дальнейшего изложения предположим, что веса всех входных ЛП равны. Диапазоны изменения переменных и типы функций принадлежности не влияют на способ синтеза ПП, поэтому принципы их выбора в данной работе не рассматриваются. Представим фрагмент ПП в классической развернутой форме (1):

$$\begin{aligned}
 & \{\text{ДП}_1 = \text{Н}\} \text{ И } \{\text{ДП}_2 = \text{Н}\} \text{ И } \{\text{ДП}_4 = \text{С}\} \text{ ИЛИ} \\
 \text{ЕСЛИ} \quad & \{\text{ДП}_1 = \text{С}\} \text{ И } \{\text{ДП}_3 = \text{Н}\} \text{ И } \{\text{ДП}_4 = \text{Н}\} \text{ ИЛИ} \\
 & \{\text{ДП}_1 = \text{Н}\} \text{ И } \{\text{ДП}_2 = \text{Н}\} \text{ И } \{\text{ДП}_3 = \text{С}\}, \\
 \text{ТО} \quad & \text{РД} = \text{ОН}.
 \end{aligned} \tag{2}$$

Для формализации процесса синтеза и анализа базы ПП воспользуемся представлением конъюнкций термов ЛП в векторной форме, аналогично кубическому представлению логических функций в многозначном алфавите. Как правило, кубическое исчисление используется в табличных методах моделирования цифровых схем и построении тестов. Так, ПП из (2) в кубическом виде представляются следующим образом:

$$\text{РД}^{\text{ОН}} = \{\text{ННХС, СХНН, ННСХ}\}. \tag{3}$$

Каждой конъюнкции (строке) из (2) соответствует куб, ранг которого определяется количеством несущественных ЛП (координат, равных X). В данном контексте под X будем понимать следующее: ЛП может принять любое значение, так как изменение её значения не ведет к изменению значения выходной ЛП, т.е. она является несущественной.

Для формализации и последующей автоматизации работы с данной кубической формой представления ПП исходя из трехуровневой градации оценок входных ДП, введем многозначный алфавит кубического исчисления АЗ, состоящий из трех примитивов Н, С, В ( $m = 3$ ).

Данные символы должны принимать четкие значения, при этом оставаясь эквивалентами термов нечетких переменных. Для этого необходимо сделать некоторые преобразования над входными лингвистическими переменными, входящими в состав продукционных правил. Исходя из того, что этап формирования продукционных правил идет до этапа фазификации, то ни форма функции принадлежности, ни диапазоны значений термов не влияют на процесс синтеза и анализа продукционных правил. Таким образом, нечеткие значения термов ЛП заменяются их эквивалентами, что в целом не меняет формы представления продукционных правил.

Рассмотрим пример лингвистической переменной «диагностический признак», параметры термов которой представлены в табл. 1, а график гауссовой функции принадлежности – на рис. 1, а.

Таблица 1

Термы	Диапазоны		Ширина	M	$\sigma$
Н	0	40	40	20	10,25
С	30	70	40	50	10,25
В	60	100	40	80	10,25

Выполним преобразование функции принадлежности ЛП таким образом, чтобы термы были независимы друг от друга, т.е. чтобы границы диапазонов значений термов не пересекались. Таким образом, получим П-образную функцию принадлежности (рис. 1, б), где термы H, C и B не пересекаются. Подобные преобразования, выполненные над каждой входной лингвистической переменной, позволят перейти к представлению продукционных правил в кубической форме с использованием четких переменных в многозначном алфавите.

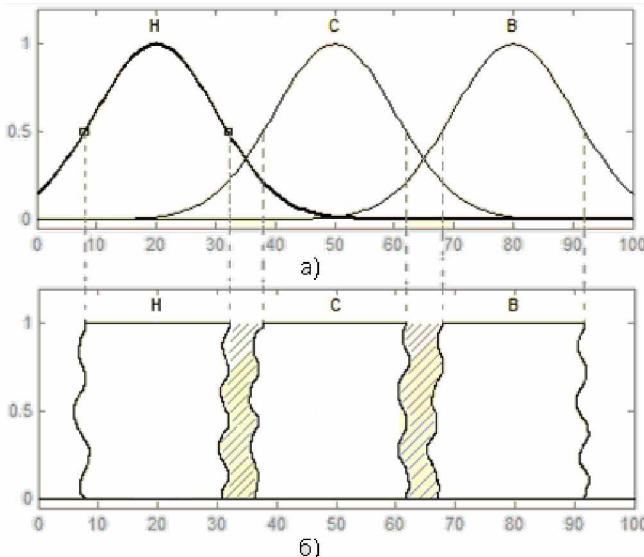


Рис. 1. График функции принадлежности в виде гауссовой кривой (а) и П-подобной функции принадлежности с размытыми границами (б)

### 5. Кубитное кодирование многозначного алфавита

Предложенный многозначный алфавит кубического исчисления А3, состоящий из  $m$  примитивов H, C, B ( $m = 3$ ), имеет  $2^m = 2^3 = 8$  символов:

$$A3 = \{H, C, B, X = \{H, C, B\}, K = \{H, C\}, L = \{C, B\}, M = \{H, B\}, \emptyset(U)\},$$

где  $X = H \cup C \cup B$  – универсум, а  $U$ , с одной стороны – символ алфавита, который служит для замыкания алфавита относительно теоретико-множественных операций, а с другой – результат операции пересечения и обозначает пустое множество  $\emptyset$ .

Для программной реализации операций с символами предложенного алфавита требуется закодировать каждый его символ таким образом, чтобы максимально упростить и ускорить процесс вычислительной работы с ними. В [4] предложена кубитная структура данных, которая является нечисленной формой описания объекта. Под кубитом здесь понимается двоичный вектор, содержащий  $n$ -битов, для задания булеана (множества всех подмножеств) состояний, состоящих из  $n$ -примитивов. С практической точки зрения (при программировании) кубит представляет собой способ кодирования  $2^m$  символов замкнутого алфавита (булеана), при котором  $m$  базовых примитивов кодируются унитарным кодом. Для приведенного примера при  $m = 3$  базовые символы алфавита кодируются: {H = 001, C = 010, B = 100}.

Кубиты символов алфавита получаются путем дизъюнкции (ИЛИ) между базовыми примитивами:

$$\begin{aligned} K &= \{H \vee C\} = 001 \vee 010 = 011, \\ L &= \{C \vee B\} = 010 \vee 100 = 110, \\ M &= \{H \vee B\} = 001 \vee 100 = 101, \\ X &= \{H \vee C \vee B\} = 001 \vee 010 \vee 100 = 111, \\ \emptyset(U) &= 000. \end{aligned}$$

Графической интерпретацией кубитного представления (булеана) является диаграмма Хассе, представленная на рис. 2.

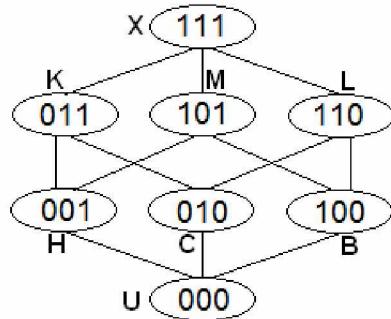


Рис. 2. Диаграмма Хассе для кубитного представления алфавита А3

Теоретико-множественные операции (пересечение, объединение, дополнение и алгебраическая сумма) над символами алфавита сводятся к соответствующим поразрядным логическим операциям над их кубитами (AND – пересечение, OR – объединение, NOT – дополнение и XOR – алгебраическая сумма (+)).

Например:

$$\begin{aligned}
 K &= H \cup C = 001 \vee 010 = 011, \\
 L &= B \cup C = 100 \vee 010 = 110, \\
 M &= H \cup B = 001 \vee 100 = 101, \\
 X &= H \cup C \cup B = 001 \vee 010 \vee 100 = 111, \\
 \bar{L} &= \overline{110} = 001 = H, \\
 K + X &= 011 \oplus 111 = 100 = B, \\
 C + C &= 010 \oplus 010 = 000 = U(\emptyset), \\
 L + M &= 110 \oplus 101 = 011 = K.
 \end{aligned}$$

Таким образом, каждое ПП можно представить как набор n-разрядных векторов (кубов соответствующего ранга), значение каждого разряда которых определяется многозначным алфавитом, задаваемым m примитивами. При этом каждая конъюнкция из (1) представима одним n-разрядным вектором. Понятие «вектор» в данном аспекте является синонимом понятия «куб». Ранг куба определяется количеством символов «X» в конъюнкции. Например, куб ННННН имеет 0-й ранг, а куб ННННХ – 1-й ранг.

В алфавите А3 введены теоретико-множественные операции пересечения, объединения, дополнения и операции логические (алгебраические), описанные в работе [3].

Рассмотрим примеры выполнения данных операций для двух четырехразрядных векторов (фрагмент ПП для четырех ДП) в кубитной кодировке:

1. Результат пересечения векторов  $KLMB \cap HKBM = HCBB$  приведен в табл. 2.

Таблица 2

K	L	M	B	$\cap$	H	K	B	M	=	H	C	B	B
0	1	1	1		0	0	1	1		0	0	1	1
1	1	0	0	$\wedge$	0	1	0	0	$=$	0	1	0	0
1	0	1	0		1	1	0	1		1	0	0	0

2. Результат объединения векторов  $KLMB \cup HKBM = KXMM$  приведен в табл. 3.

Таблица 3

K	L	M	B	∪	H	K	B	M	=	K	X	M	M
0	1	1	1		0	0	1	1		0	1	1	1
1	1	0	0	∨	0	1	0	0	=	1	1	0	0
1	0	1	0		1	1	0	1		1	1	1	1

3. Результат дополнения вектора  $\overline{KLMB} = BHCK$  приведен в табл. 4.

Таблица 4

~	(K	L	M	B)	=	B	H	C	K
	0	1	1	1	=	1	0	0	0
	1	1	0	0		0	0	1	1
	1	0	1	0		0	1	0	1

4. Результат алгебраической суммы векторов  $KLMB + HKBM = CMNH$  приведен в табл. 5.

Таблица 5

K	L	M	B	+	H	K	B	M	=	C	M	H	H
0	1	1	1		0	0	1	1		0	1	0	0
1	1	0	0	⊕	0	1	0	0	=	1	0	0	0
1	0	1	0		1	1	0	1		0	1	1	1

Рассмотрим примеры выполнения операций в символах многозначного алфавита.

Примеры пересечения векторов:

$$HHH \cap HHC = HHU = \emptyset,$$

$$BCH \cap BCK = BCH.$$

Частным случаем пересечения является поглощение:

$$BBH \cap BBK = BBH \quad (BBH \in BBK).$$

Примеры объединения векторов:

$$HHH \cup HHC = HHK,$$

$$BCH \cup BCK = BCK.$$

Частным случаем объединения является склеивание. Векторы могут склеиваться, если кодовое расстояние между ними равно 1. Кодовым расстоянием  $d$  между двумя векторами называется количество отличающихся координат (разрядов), т.е. количество символов, не равных  $U$  в наборе, полученном в результате алгебраической суммы двух векторов. Например, если  $A = HHHHB$  и  $B = HHHHC$ , то результатом их алгебраической суммы будет вектор  $UUUUL$ . Тогда кодовое расстояние между ними  $d=1$  (результат суммы дал только одно значение, не равное  $U$ ).

Тогда за счет универсума  $X = H \cup C \cup B$  получим:

$$HHH \cup HHC \cup HNB = HHX,$$

$$HHK \cup HNB = HHX,$$

$$HHL \cup HHH = HHX,$$

$$HNM \cup HNC = HHX.$$

Операция объединения в общем случае является корректной только при  $d=1$  (склеивание). Рассмотрим объединение двух векторов  $A1 = HKHM$  и  $A2 = HBLC$ . В результате алгебраической суммы получим вектор  $UXXX$ , что говорит о том, что кодовое расстояние между ними  $d=3$ . Результатом объединения этих векторов будет вектор  $HXXX$ . Однако этот результат противоречив, так как в результате  $HXXX$  содержится вектор  $HBBC$ , не принадлежащий ни  $A1$ , ни  $A2$ :

$$HBBC \in (A1 \cup A2), \quad \text{но} \quad \begin{cases} HBBC \notin B1, & \text{т.к. } HBBC \cap HKHM = \emptyset \\ HBBC \notin B2, & \text{т.к. } HBBC \cap HBLC = \emptyset \end{cases}.$$

Это показывает, что при кодовом расстоянии  $d > 1$  операция объединения может давать противоречивый результат.

Примеры дополнения векторов:

$$\begin{aligned} \sim HHH &= LLL, \\ \sim HCB &= LMK, \\ \sim BBX &= LLU. \end{aligned}$$

Примеры алгебраической суммы двух векторов:

$$\begin{aligned} HHHHB + HHHHC &= UUUUL, \\ HLBBC + BLBBC &= MUUUU. \end{aligned}$$

## 6. Анализ продукционных правил на корректность

Анализ ПП на корректность выполняется на основе разработанной вербальной модели, поэтому формализуемая модель в виде базы знаний должна удовлетворять определенным формальным требованиям, не касающимся смыслового аспекта правила [4, 5]. База знаний в целом должна быть корректной, а именно полной, непротиворечивой, связной и минимальной.

База знаний является полной, если для каждого входного вектора  $\forall X = (x_1, x_2, \dots, x_n) \in E_1 \times E_2 \times \dots \times E_n$  механизм нечеткого вывода может получить некоторое значение выходной переменной  $y \in E_y$ , т.е. существует как минимум одно правило, которое ставит в соответствие входному вектору  $X$  лингвистическое значение выходной переменной  $y$  (лингвистическая полнота).

База знаний является непротиворечивой (согласованной), если она не содержит несовместные правила – правила с одинаковыми лингвистическими условиями, но различными заключениями, например, такими, как правила П1 и П2:

$$\begin{aligned} P1: \quad \text{ЕСЛИ } \{x_1 = H\} \text{ И } \{x_2 = H\}, \quad \text{ТО } y = OH, \\ P2: \quad \text{ЕСЛИ } \{x_1 = H\} \text{ И } \{x_2 = H\}, \quad \text{ТО } y = H. \end{aligned}$$

Эти правила являются противоречивыми, так как одно и то же условие ( $H$ ) активизирует разные выходные термы ( $OH$  и  $H$ ).

База знаний является связной, если для любой пары смежных правил значение выходной ЛП тоже смежное, т.е. условия этих правил отличаются только одним из  $n$  подусловием и в этих подусловий используется одна и та же ЛП с различными значениями.

Например, приведенные правила:

$$\begin{aligned} P1: \quad \text{ЕСЛИ } \{x_1 = H\} \text{ И } \{x_2 = H\} \text{ И } \{x_3 = H\} \text{ И } \{x_4 = H\} \text{ И } \{x_5 = C\}, \quad \text{ТО } y = OH \\ P2: \quad \text{ЕСЛИ } \{x_1 = H\} \text{ И } \{x_2 = H\} \text{ И } \{x_3 = H\} \text{ И } \{x_4 = H\} \text{ И } \{x_5 = B\}, \quad \text{ТО } y = H \end{aligned}$$

являются смежными, так как термы  $H$  и  $C$  переменной  $x_5$  – смежные.

При создании базы знаний вступают в противоречие два требования: для увеличения точности модели необходимо увеличивать число правил в БЗ; для увеличения быстродействия и улучшения прозрачности модели, что напрямую связано с точностью

диагностирования компьютерных систем, необходимо уменьшать число правил. Минимальная БЗ – это база, из которой нельзя удалить ни одного из производственных правил, не нарушив тем самым её полноту.

## 7. Пример составления производственных для ЭСД

В качестве ОД рассмотрим корпоративную компьютерную сеть, которая представляет собой объединение 400 узлов, распределенных территориально в пределах города. Сеть состоит из трех сегментов, каждый из которых расположен в отдельном здании и характеризуется различными способами подключения узлов: по витой паре, по Wi-Fi технологии и по оптоволокну. Структура каждого сегмента - «звезда». В сегменте № 1 размещены: 250 компьютеров и ноутбуков; файлообменный сервер; DHCP, DNS, TFTP и почтовый серверы; сервер IP-телефонии; оптоволоконная линия 1000BaseSX к PC-router через медиаконвертер; PC - router, обслуживающий 1 линию Ethernet 1000Base-T; восемь 48 -портовых управляемых коммутаторов 1000Base-T; точка доступа (2,4 ГГц, до 300 Мбит/с), выступающая в роли моста и отвечающая за связь с сегментом № 2; 4 точки доступа (2,4 ГГц, до 300 Мбит/с). В сегменте № 2 размещены: 50 компьютеров, ноутбуков и бездисковых станций; файлообменный и терминальный сервер; DHCP, DNS, TFTP и почтовый серверы; два 32-портовых управляемых коммутатора 100Base-T; 4 точки доступа (2,4 ГГц, до 300 Мбит/с); точка доступа (5 ГГц, до 300 Мбит/с), выступающая в роли моста и отвечающая за связь с сегментом № 1. В сегменте № 3 размещены: 100 компьютеров, ноутбуков; DHCP (резервный), DNS (резервный) – сервера; сервер IP-телефонии; три 48-портовых управляемых коммутатора 1000Base-T; 2 точки доступа (2,4 ГГц, до 300 Мбит/с); оптоволоконная линия 1000BaseSX к PC-router через медиаконвертер; PC-router, обслуживающий 1 линию Ethernet 1000Base-T.

Анализируя топологию и назначение данной сети, администратор определил следующие «узкие места»:

- Точки доступа (2,4 ГГц, до 300 Мбит/с) в каждом сегменте «режут» максимальную скорость 1 Гбит/с до своей заявленной (теоретической) максимальной скорости (300 Мбит/с). А если учесть тот факт, что реальная пропускная способность точки доступа находится в границах от 40 до 150 Мбит в секунду (т.е. зависит от внешних факторов и от числа подключений), то данная точка доступа является «узким местом».
- Сетевые адAPTERы файлообменного сервера в сегменте № 1. Существует вероятность того, что к данному серверу будут обращаться одновременно все узлы сегмента (250 узлов).
- Прикладное программное обеспечение «1С: Предприятие» (сетевая версия), база данных которого расположена на файлообменном сервере сегмента 1 и работает с которым осуществляется с персональных компьютеров, является «узким местом» в силу специфики самой программы. В сегменте № 2 такое же программное обеспечение не является «узким местом» в связи с тем, что база данных находится на терминальном сервере и обращение к ней происходит как к локальной.
- Некорректные параметры в настройках коммутационного оборудования. Так, в силу человеческого фактора может быть ограничена пропускная способность оборудования для конкретного узла или сегмента.

Рассмотрим маршрут сети от персонального компьютера из сегмента № 2 (192.168.252.101) до DHCP/TFTP/DNS/mail сервера (10.0.0.16), принадлежащего сегменту № 1. Администратор выделил следующие диагностические признаки, которые позволяют обнаружить потенциальные «узкие места»:

1. Время реакции узла – определяется как интервал между возникновением запроса пользователя к какой-либо сетевой службе и получением ответа на него.
2. Коэффициент потери пакетов – процент потерянных запросов-ответов от данного узла.
3. Реальная пропускная способность – измеряется путем копирования файла большого размера (100 Мбит) с сервера.

Эти характеристики качества сети будут играть роль диагностических признаков (ДП) в процессе экспертного диагностирования. Эксперт предложил следующий набор

продукционных правил:  $k^{<\text{OH,H,C,D,B}>} = \{k_1^{\text{OH}}, k_1^{\text{H}}, k_2^{\text{H}}, k_1^{\text{C}}, k_2^{\text{C}}, k_1^{\text{D}}, k_2^{\text{D}}, k_1^{\text{B}}\}$ , где  $k_1^{\text{OH}} = \text{HHK}$ ,  $k_1^{\text{H}} = \text{HNB}$ ,  $k_2^{\text{H}} = \text{HCC}$ ,  $k_1^{\text{C}} = \text{HCB}$ ,  $k_2^{\text{C}} = \text{CCC}$ ,  $k_1^{\text{D}} = \text{HBB}$ ,  $k_2^{\text{D}} = \text{CCB}$ ,  $k_3^{\text{D}} = \text{BCB}$ ,  $k_1^{\text{B}} = \text{BBC}$  – компактные минимизированные формы кубов.

Выполним анализ предложенной базы ПП на корректность с использованием разработанной программы.

### 8. Программа анализа производственных правил на корректность

Предложенные процедуры проверки базы ПП на корректность реализованы в виде отдельной программы. Она написана на основе библиотеки QT (версия 4.8) – кроссплатформенного инструментария разработки ПО на языке программирования C++, позволяет абстрагироваться от вызовов низкоуровневых функций системы, предоставляя взамен удобный интерфейс классов для управления программой. С её помощью организовано взаимодействие с пользователем, работа с файлами и потоками. Разработанный программный продукт в полном объеме реализует алгоритм проверки ПП на корректность (ввод ЛП, ввод ПП в векторной форме, вывод ПП в развернутой форме, преобразование в кубическую форму, проверка на полноту, непротиворечивость, связность и минимальность). Интерфейс программного модуля интуитивно понятен и пригоден для использования не только экспертами, занимающимися созданием базы знаний, но и рядовыми пользователями ЭСД. Инсталляции программа не требует, записи в реестр не заносятся, ввиду автономности мастер установки отсутствует. Удаление программного продукта осуществляется путем удаления папки, содержащей приложение. Графический интерфейс отделен от ядра, что позволяет использовать ядро и в других проектах. Реализована возможность сохранения и загрузки проектов.

Алгоритм проверки базы ПП:

- Выбираем File a New Project. В появившемся окне вводим имена входных и выходных лингвистических переменных (рис. 3, а). Далее жмем Next. В появившемся окне вводим производственные правила в компактном виде. В скобках напротив каждого правила отображается количество правил, которые получатся в результате перехода от компактной формы представления ПП к классической развернутой форме (рис. 3, б).

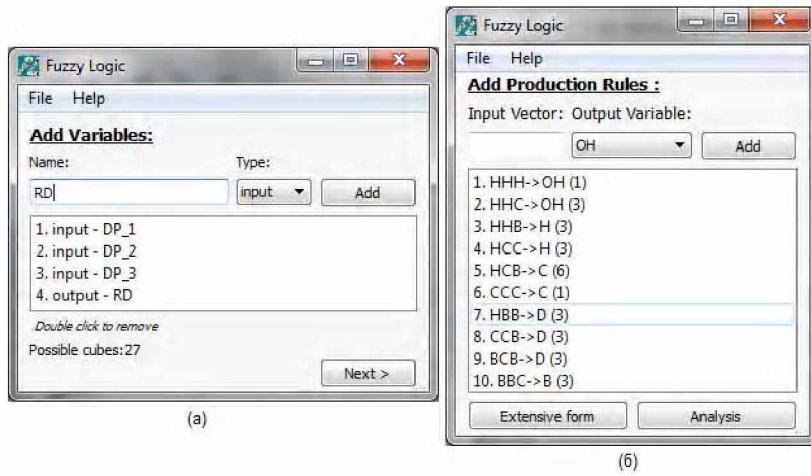


Рис. 3. Окно ввода ЛП (а) и окно ввода ПП (б)

Удалить то или иное правило можно путем двойного щелчка на нем.

- Нажав на кнопку "Extensive form" просматриваем все ПП, представленные в классической развернутой форме (рис. 4). Выделив и скопировав полученные правила, можно в дальнейшем использовать их в пакете Fuzzy logic Toolbox.

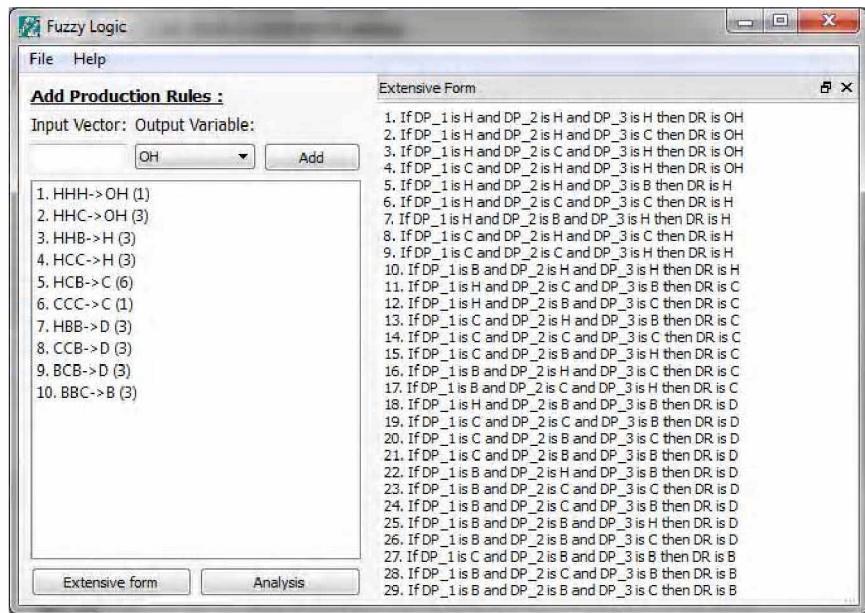


Рис. 4. Окно вывода ПП в развернутой форме

3. Нажав на кнопку “Analysis”, переходим к анализу ПП на корректность (рис. 5). Поочередное нажатие на кнопки “Completeness”, “Consistency”, “Coherence” и “Minimality” позволит посмотреть результат анализа на полноту, противоречивость, связность и минимальность соответственно. При этом в окне “Source rule set” будут отображаться ПП, представленные в минимизированной компактной форме.

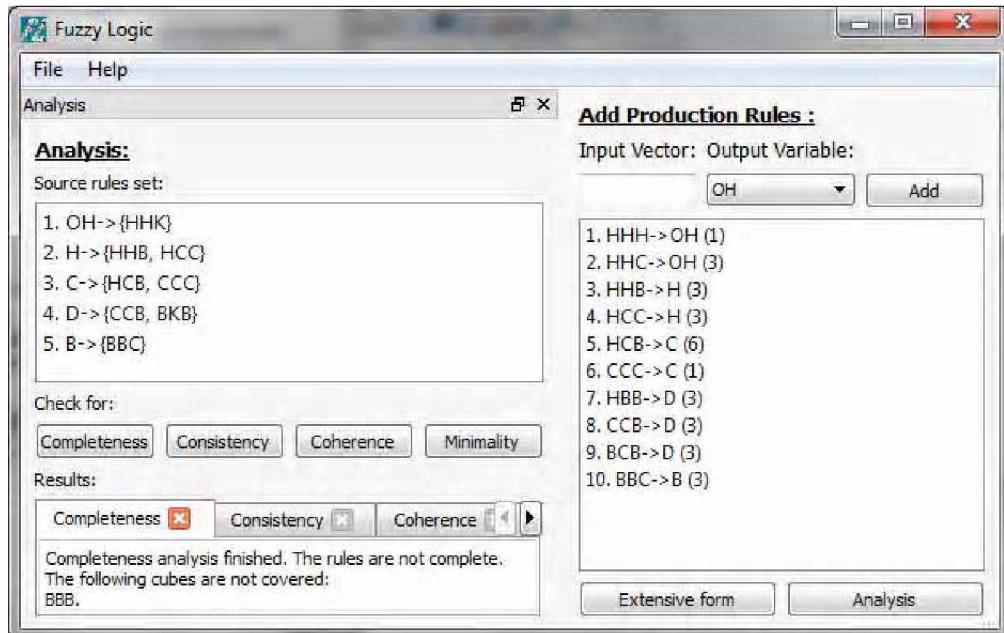


Рис. 5. Окно анализа ПП на корректность

Согласно результатам проверки программа показала, что предложенный набор ПП является неполным (рис. 6, а), противоречивым (рис. 6, б), не связанным (рис. 6, в) и минимальным (рис. 6, г). А точнее, анализ на полноту выявил, что в результате синтеза базы ПП правило *BBB* было упущено. Отрицательный результат данной проверки ведет к тому, что проверка на минимальность не требуется (в таком случае программа всегда будет считать данную БЗ минимальной). Анализ на непротиворечивость выявил, что в результате синтеза базы ПП одно и то же правило (*BBC*) принадлежит одновременно двум

разным выходным термам:  $BCB \cap BBC = BBC \Rightarrow k_3^D \cap \{k_1^B\} \neq \emptyset$ . Анализ на связность показал, что векторы  $BCB \in k_3^D$  и  $BBC \in k_1^B$  не являются смежными ( $BCB + BBC = UUU$ ), а следовательно, БЗ не является связной.

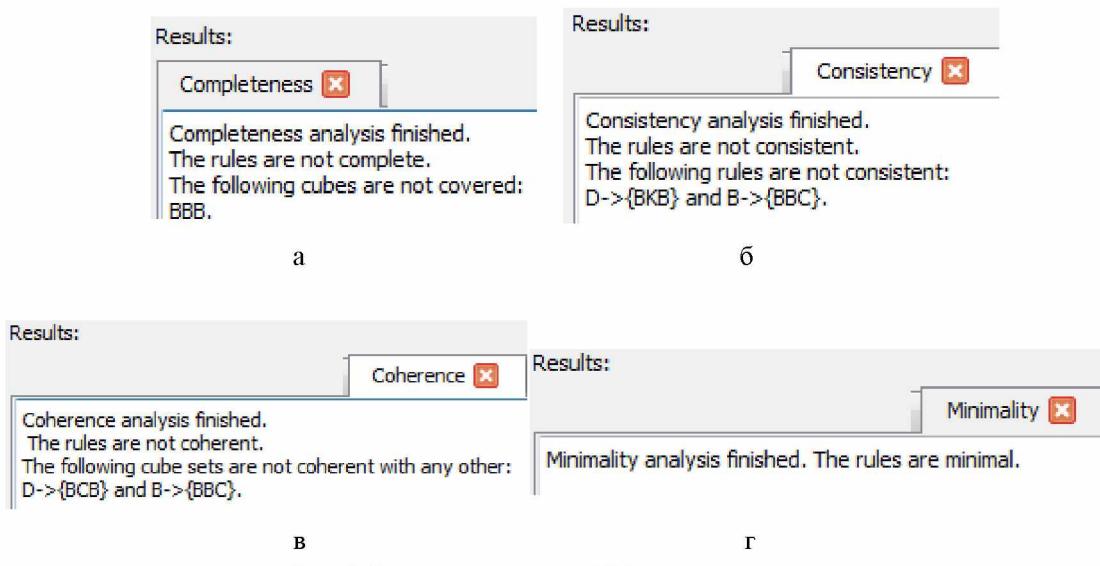


Рис. 6. Результаты анализа ПП на корректность:

- а – проверка на полноту;
- б – проверка на непротиворечивость;
- в – проверка на связность;
- г – проверка на минимальность

Обнаружив данные ошибки, целесообразно выполнить корректировку базы ПП:  $k_1^{OH} = HHK$ ,  $k_1^H = HNB$ ,  $k_2^H = HCC$ ,  $k_1^C = HCB$ ,  $k_2^C = CCC$ ,  $k_1^D = HBB$ ,  $k_2^D = CCB$ ,  $k_1^B = BBL$ . Повторная проверка показала, что данный набор ПП является корректным и пригоден для использования в нечеткой ЭСД. Эта программа позволяет существенно уменьшить количество времени, необходимое эксперту на подготовку корректной базы знаний, посредством уменьшения большого количества ручной работы.

#### 9. Экспертное оценивание качества корпоративной сети

Процесс оценивания качества сети можно рассматривать как диагностический эксперимент (ДЭ), в ходе которого на основании значений диагностических признаков определяется производительность маршрута сети. ДЭ состоит из определения значений основных ДП администратором сети и эталонных значений для данного класса программ – экспертом, проведения испытания и сравнения результатов с эталоном, а затем принятия решения о результате испытания.

В результате проведения диагностического эксперимента было установлено следующее.

Время реакции узла при обращении к серверу в среднем за сутки составило 31 мс. Нормируя полученное значение по формуле (1), получаем  $t_r' = 100 - 31 = 69$  баллам.

Коэффициент потерь при обращении к серверу в среднем за сутки составил 7 %. Нормируем полученное значение по формуле (2), получаем  $t_l' = 10 * 7 = 70$  баллов.

Реальная пропускная способность при передаче файла размером 100 Мбайт в среднем за сутки составила 30 Мбит/с. Нормируя полученное значение по формуле (3), получаем  $S' = 30 / 2,9999 \approx 10$  баллов.

Подытожим результаты анализа линии связи в сети передачи данных.

Каждый диагностический признак был оценен администратором следующим образом:  $D\pi_1 = 69$  баллов,  $D\pi_2 = 70$  баллов,  $D\pi_3 = 10$  баллов. В качестве алгоритма нечеткого вывода был выбран алгоритм Мамдани, где на этапе аккумуляции использовался метод граничной суммы и метод максимума, а в качестве метода дефазификации – метод

«центра тяжести». Для анализа результатов диагностирования применялась математическая система Matlab 7.5, а именно, специальный пакет нечеткого вывода Fuzzy Logic Toolbox (лицензия № 532868 на продукт MathworkAcademic). При использовании в качестве операции аккумуляции операции максимума система нечеткого вывода вычислила следующие результаты: «уровень качества» = 52,7 баллов, что с большей степенью ( $\mu_C(52,7) \approx 0,941$ ) принадлежит среднему уровню ( $C$ ) производительности линии связи. А при использовании операции граничной суммы «уровень качества» = 52,5 баллов, что также с большей степенью ( $\mu_C(52,5) \approx 0,949$ ) принадлежит среднему уровню ( $C$ ) производительности участка сети. На рис. 7 приведены нечеткие множества и результаты дефазификации для двух методов аккумуляции.



Рис. 7. Аккумуляция по методу “max” (а) и “sum”(б)

## 10. Выводы

При экспертном анализе состояния КС экспертные знания играют решающую роль. Для автоматизации процесса оценки предложено использовать ЭСД, в состав которой входит блок нечеткого вывода. В качестве базы знаний используются продукционные правила. Знания в виде правил легко формулируются и воспринимаются экспертами, однако процесс их создания и анализа крайне затруднен.

Для упрощения процесса анализа ПП предложена процедура преобразования развернутой формы их представления в кубическую форму. Для этого был предложен многозначный алфавит кубического исчисления на основе трех примитивов. Для вычислительной реализации операций с символами многозначного алфавита использована кубитная форма их кодирования, а также введены теоретико-множественные и логические операции в кубитной кодировке символов алфавита.

Практической реализацией предложенного алгоритма проверки ПП на корректность стала программа, позволяющая значительно сократить время подготовки базы ПП для нечеткой ЭСД.

**Список литературы:** 1. Круглов В.В. Нечеткая логика и искусственные нейронные сети / В.В. Круглов, М.И. Дли, Р.Ю. Голубов. М.: Физматлит, 2001. 221 с. 2. Кривуля Г.Ф. Нечеткая логика в экспертной оценке ИКТ–компетентностей / Г.Ф. Кривуля, А.С. Шкиль, Д.Е. Кучеренко, Е.В. Гаркуша // Вестник Херсонского государственного технического университета. Херсон: ХГТУ. 2011. №2 (41). С.13–22. 3. Шкиль А.С. Анализ продукционных правил в экспертных системах диагностирования/ Г.Ф.Кривуля, А.С. Шкиль, Д.Е. Кучеренко//Автоматика и вычислительная техника. Рига. 2013. №6. С.59-72. 4. Хаханов В.И. Квантовые модели вычислительных процессов/ В.И. Хаханов, Али А. Мурад, Е.И. Литвинова, О.А. Гузь, И.В. Хаханова // Радиоэлектроника и информатика. 2011. №3 (54). С. 35-40. 5. Каргин А.А. Введение в интеллектуальные машины. Книга 1. Интеллектуальные регуляторы / А.А. Каргин. Донецк: Норд-Пресс, ДонНУ, 2010. 526 с.