

В роботі побудовано моделі спрямованих реляційних мереж відношення імплікації, які можуть бути використані для представлення знань в інтелектуальних системах паралельної дії. Створено реляційну мережу, що реалізує операцію імплікації. Здійснено тестування роботи реляційної мережі. Розглянуто приклад функціонування дуги реляційної мережі за допомогою лінійного логічного оператора. Описано метод побудови спрямованих схем реляційної мережі

Ключові слова: теорія інтелекту, алгебра предикатів, алгебропредикатні структури, спрямовані реляційні мережі, лінійні логічні оператори

В работе построены модели направленных реляционных сетей отношения импликации, которые могут быть использованы для представления знаний в интеллектуальных системах параллельного действия. Создана реляционная сеть, реализующая операцию импликации. Осуществлено тестирование работы реляционной сети. Рассмотрен пример функционирования дуги реляционной сети с помощью линейного логического оператора. Описан метод построения направленных схем реляционной сети

Ключевые слова: теория интеллекта, алгебра предикатов, алгебропредикатные структуры, направленные реляционные сети, линейные логические операторы

УДК 004.825

DOI: 10.15587/1729-4061.2014.30567

МОДЕЛИРОВАНИЕ ОТНОШЕНИЯ ИМПЛИКАЦИИ С ИСПОЛЬЗОВАНИЕМ НАПРАВЛЕННЫХ РЕЛЯЦИОННЫХ СЕТЕЙ

В. И. Булкин

Кандидат технических наук, доцент
Кафедра экономической теории и
информационных технологий
Макеевский
экономико-гуманитарный институт
ул. Островского, 16, г. Макеевка,
Украина, 86157
E-mail: bulkin01@mail.ru

1. Введение

Естественный интеллект имеет очень сложную структуру. В нем можно выделить уровни подсознания, сознания, логического мышления, алгоритмической деятельности. Логично было бы предположить, что и моделировать функции различных уровней интеллекта следует различными методами. Однако на практике современные исследователи, имея в своих руках единственное моделирующее устройство – компьютер последовательного действия фон неймановского типа, пытаются использовать его для моделирования функций и подсознания, и сознания, и логической, и алгоритмической деятельности интеллекта. Так, например, задачи распознавания образов на протяжении многих лет безуспешно пытаются решать, используя алгоритмический метод, реализуемый на компьютерах фон неймановского типа. В то же время эти задачи естественный интеллект решает в автоматическом режиме на уровне подсознания.

Причиной неудач при решении таких задач является, очевидно, то, что исследователи выбирают неадекватный метод моделирования и, соответственно, не совсем подходящее моделирующее устройство. Ведь принцип работы компьютеров фон неймановского типа абсолютно не совпадает с механизмом работы человеческого интеллекта. Наш интеллект представляет собой естественную нейронную сеть огромной сложности. Обработка данных в этой сети осуществляется параллельно. Память человека является ассоциативной и работает совсем не так как память компьютера. Поэтому функции интеллекта, которые реализуются на уровне подсознания, было бы целесообразно моде-

лировать с помощью искусственных нейронных сетей (ИНС), которые изначально создавались для решения задач распознавания образов [1]. Ассоциативную память также можно строить с использованием ИНС, в том числе с использованием ИНС Хопфилда [2] и Хэмминга [3].

Однако при создании искусственных нейронных сетей существует проблема, которая связана с тем, что заранее не известна структура сети, которая могла бы моделировать ту или иную функцию интеллекта. Поэтому архитектура сети, как правило, выбирается произвольным образом без учета того, какие функции она сможет моделировать. В дальнейшем вновь созданную ИНС пытаются обучить так, чтобы она могла воспроизводить те или иные функции интеллекта. Такой подход к созданию интеллектуальных систем называют структурно-функциональным [4].

Альтернативой этому подходу может служить функционально-структурный подход к созданию интеллектуальных систем. При использовании этого подхода моделируемая функция интеллекта вначале описывается системой логических уравнений, записанных на языке алгебры предикатов [5], а затем каждому уравнению ставится в соответствие алгебропредикатная структура (АП-структура), реализующая аппаратно данное уравнение. При использовании данного подхода автоматически формируется структура, моделирующая соответствующую функцию интеллекта. В свою очередь, любую АП-структуру можно представить в виде поля ассоциативной памяти, хранящей коды предметов универсума предметов, и логического блока, выполняющего логические операции над предикатами узнавания предметов [6].

Используя этот подход, можно воспроизводить такие функции интеллекта, как понимание текстов естественного языка, хранение и обработка знаний, логическое мышление. Функции интеллекта, связанные с реализацией алгоритмической деятельности человека можно возложить на универсальный компьютер последовательного действия. Следовательно, интеллектуальную систему можно представить в виде комплекса технических устройств, состоящего из искусственных нейронных сетей, моделирующих деятельность подсознания, алгебропредикатных структур, моделирующих логическое мышление и способность владеть языком, а также способность хранить и обрабатывать знания. Управление этими устройствами и решение задач алгоритмического характера может взять на себя универсальный компьютер фон неймановского типа.

2. Анализ литературных источников

Создание искусственного интеллекта невозможно без использования комплексного подхода к решению этой крупномасштабной задачи. Комплексный подход подразумевает использование не только алгоритмического метода, но и методов создания интеллектуальных систем параллельного действия. При обзоре периодических научных изданий можно отметить, что основное большинство работ по искусственному интеллекту опираются на IT технологии, использующие алгоритмический метод реализации математических моделей и компьютер последовательного действия как основное и практически единственное моделирующее устройство.

Так, например, в работе [7] представлены два новых алгоритма поиска, основанных на использовании новой функции оценки. В работе [8] представлен новый подход к визуальной аналогии с использованием фрактального алгоритма (Fractal Ravens algorithm). Работа [9] представляет собой всеобъемлющий обзор достижений в области свом-интеллекта (Swarm intelligence), который определяется как коллективное поведение децентрализованных и самоорганизующихся стай. Были разработаны несколько алгоритмов в зависимости от различных видов интеллектуального поведения пчелиных роев и применяются для решения реальных мировых проблем. В работе [10] предлагается новая техника избыточной дискретизации, которая получила название DBSMOTE. Экспериментальные результаты показывают, что данная техника имеет ряд преимуществ по сравнению с известными методами обработки несбалансированных множеств данных.

В статье [11] представлен новый эволюционный алгоритм под названием Динамический Алгоритм Раздельного Поиска (DPSA) для глобальных задач оптимизации с непрерывными переменными. DPSA является алгоритмом на основе стохастического поиска в природе, который, в основном, состоит из процесса инициализации и процесса эволюции. Экспериментальные результаты показывают, что DPSA имеет лучшую производительность, чем другие известные алгоритмы, в особенности для высоко размерных проблем оптимизации. DPSA является эффективным эволюционным алгоритмом решения масштабных задач глобальной оптимизации. В работе [12] представлен

обзор различных контролируемых нейронных сетей с онлайн возможностью обучения. Рассмотрен ряд ключевых нейронных сетевых архитектур, в том числе нейронные сети прямого распространения, рекуррентные нейронные сети, нечеткие нейронные сети и другие нейронные сети. Представлено резюме обзора, который охватывает различные сетевые архитектуры и их приложения.

В работе [13] дан обзор методов распределенного машинного обучения. Для автоматического машинного анализа больших объемов данных предлагается использовать метод крупномасштабного обучения. Для этого предлагается распределенное обучение, так как процесс обучения между несколькими рабочими станциями является естественным способом увеличения масштабов алгоритмов обучения. Приводится обзор методов распределенного обучения для работы с "очень большими" наборами данных. В работе [14] высказывается сомнение в том, что решение проблемы программного обеспечения занимает центральное место в ИИ.

Краткий выборочный обзор периодических научных публикаций в журналах по искусственному интеллекту говорит о том, что большинство исследователей не выходит за рамки тематики связанной с алгоритмическими методами, используемыми при создании систем искусственного интеллекта. В связи с этим возрастает актуальность работ, связанных с созданием интеллектуальных систем параллельного типа, поскольку системы искусственного интеллекта должны иметь в своем составе материальный носитель интеллекта в виде динамических сетевых мозгоподобных структур способных к обучению и самообучению. Создание направленных схем реляционных сетей является одним из первых шагов на этом пути.

3. Цель и задачи исследования

Целью данной работы является создание моделей направленных реляционных сетей, которые могут быть использованы для представления знаний в интеллектуальных системах параллельного действия.

Для достижения поставленной цели необходимо решить следующие задачи:

- построить модели направленных реляционных сетей отношения импликации;
- создать реляционную сеть, реализующую операцию импликации;
- осуществить потактовое тестирование работы реляционной сети;
- рассмотреть пример функционирования дуги реляционной сети с помощью линейного логического оператора;
- описать метод построения направленных схем реляционной сети.

В рамках теории интеллекта в процессе развития алгебры логики и многозначной логики была разработана алгебра предикатов. Последние исследования в области теории интеллекта связаны с моделированием реляционных сетей [15, 16]. Реляционная сеть представляет собой графическое изображение результата бинарной декомпозиции многоместного предиката. Реляционная сеть включает в свой состав

множество полюсов, связанных между собой дугами. Полюса реляционной сети соответствуют наборам, составленным из предметной переменной и области ее определения. Дуги реляционной сети соответствуют отношениям, заданным на множествах значений соответствующих предметных переменных. Построение реляционных сетей связано с решением проблемы создания интеллектуальных систем параллельного действия, которые могли бы работать в составе мозгоподобных ЭВМ [17].

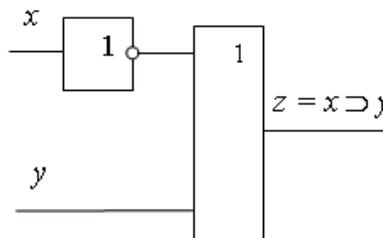


Рис. 1. Логическая схема, реализующая функцию импликации

4. Моделирование реляционной сети

В работе [18] рассматривается метод построения направленных схем реляционных сетей для отношения эквивалентности. Отношение эквивалентности задают булевым уравнением

$$z = x \sim y, \tag{1}$$

где x, y и z – это логические переменные, $x, y, z \in \Sigma = \{0, 1\}$.

Отношение эквивалентности играет важную роль при описании объектов различной природы на языке алгебры предикатов. Отношение эквивалентности используется для описания лингвистических объектов. Кроме того, отношение эквивалентности лежит в основе метода компараторной идентификации. На множестве логических значений $\Sigma = \{0, 1\}$ отношение эквивалентности совпадает с бинарной логической функцией эквивалентности. Логическая функция эквивалентности является основой построения одноразрядного компаратора, выполняющего операцию сравнения в каждом разряде ячейки ассоциативного запоминающего устройства [6].

Не менее важным отношением, используемым при описании интеллектуальных процессов в рамках теории интеллекта, является отношение импликации. Импликация (от лат. *implicare* – тесно связываю) – это логическая операция, которой в естественном языке соответствует связка «если..., то...», образующая из двух высказываний (предложений) A и B условное высказывание «Если A, то B». Импликацией часто называют и само условное высказывание, а также его формализованные аналоги – формулы логических исчислений, содержащие знак импликации (например, "⊃", "→") и имеющие вид $A \supset B$, где A и B – формулы исчисления. Операция импликации широко используется при описании лингвистических закономерностей на языке алгебры предикатов, а также для записи правил продукции при моделировании знаний.

Логическая функция импликации задается уравнением

$$z = x \supset y, \tag{2}$$

где x, y и z – это логические переменные, $x, y, z \in \Sigma = \{0, 1\}$.

Функция импликации может быть представлена следующей формулой:

$$z = x \supset y = \bar{x} \vee y, \tag{3}$$

Уравнению (3) соответствует логическая схема (рис. 1)

Построим реляционную сеть на основе уравнения (3). Метод построения реляционных сетей изложен в [16]. Построение реляционной сети состоит из нескольких этапов. На первом этапе выявляют предметные переменные x_1, x_2, \dots, x_n моделируемого объекта и отношения заданные на множествах их значений. Значения функции импликации можно представить в виде отношения P, заданного таблицей истинности (табл. 1).

Таблица 1

Отношение импликации

x	y	z
0	0	1
0	1	1
1	0	0
1	1	1

На втором этапе осуществляют запись выявленных отношений на языке алгебры предикатов. В результате получается предикат, соответствующий отношению P, которое представляет собой множество всех наборов предметов x_1, x_2, \dots, x_n , удовлетворяющих уравнению $P(x_1, x_2, \dots, x_n) = 1$. Предикат $P(x, y, z)$, соответствующий отношению импликации $P = \{(0, 0, 1), (0, 1, 1), (1, 0, 0), (1, 1, 1)\}$, на языке алгебры предикатов записывается в следующем виде:

$$P(x, y, z) = x^0 y^0 z^1 \vee x^0 y^1 z^1 \vee x^1 y^0 z^0 \vee x^1 y^1 z^1.$$

Затем производят бинарную декомпозицию предиката $P(x, y, z)$ вводя дополнительную переменную $t \in \{0, 1, 2, 3\}$. В результате получаем полную (развернутую) бинаризацию предиката $P(x, y, z)$:

$$R(x, y, z, t) = x^0 y^0 z^1 t^0 \vee x^0 y^1 z^1 t^1 \vee x^1 y^0 z^0 t^2 \vee x^1 y^1 z^1 t^3. \tag{4}$$

Предикату (4) соответствует отношение R, которое можно представить в виде табл. 2

Таблица 2

Отношение R

x	y	z	t
0	0	1	0
0	1	1	1
1	0	0	2
1	1	1	3

Если применить квантор существования по переменной t , то из предиката $R(x, y, z, t)$ мы вновь получим исходный предикат $P(x, y, z)$

$$\exists t R(x, y, z, t) = x^0 y^0 z^1 \vee x^0 y^1 z^1 \vee x^1 y^0 z^0 \vee x^1 y^1 z^1.$$

Исключая с помощью кванторов существования последовательно по одной переменной, сформируем бинарные предикаты:

$$P_1(x, t) = x^0 t^0 \vee x^0 t^1 \vee x^1 t^2 \vee x^1 t^3 = x^0 (t^0 \vee t^1) \vee x^1 (t^2 \vee t^3),$$

$$P_2(y, t) = y^0 t^0 \vee y^1 t^1 \vee y^0 t^2 \vee y^1 t^3 = y^0 (t^0 \vee t^2) \vee y^1 (t^1 \vee t^3),$$

$$P_3(z, t) = z^1 t^0 \vee z^1 t^1 \vee z^0 t^2 \vee z^1 t^3 = z^1 (t^0 \vee t^1 \vee t^3) \vee z^0 t^2.$$

Бинарные отношения P_1, P_2, P_3 , соответствующие предикатам $P_1(x, t), P_2(y, t), P_3(z, t)$ представлены в табл. 3.

Таблица 3

Отношения P_1, P_2, P_3

Отношение P_1		Отношение P_2		Отношение P_3	
x	t	y	t	z	t
0	0	0	0	1	0
0	1	1	1	1	1
1	2	0	2	0	2
1	3	1	3	1	3

Таким образом, на данном этапе построена математическая модель решения исходного булевого уравнения (2). Она представляет собой систему бинарных отношений P_1, P_2, P_3 , которые задаются соответствующими предикатами $P_1(x, t), P_2(y, t), P_3(z, t)$ и двудольными графами (рис. 2).

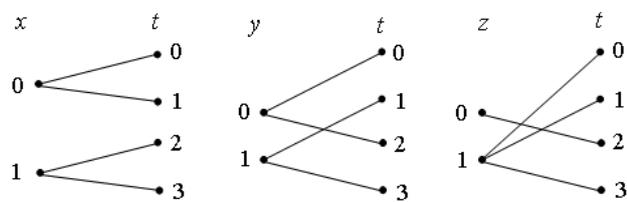


Рис. 2. Двудольные графы бинарных отношений реляционной сети

Если образовать конъюнкцию этих предикатов, то получим предикат модели (4)

$$R(x, y, z, t) = P_1(x, t) \wedge P_2(y, t) \wedge P_3(z, t) = x^0 y^0 z^1 t^0 \vee x^0 y^1 z^1 t^1 \vee x^1 y^0 z^0 t^2 \vee x^1 y^1 z^1 t^3. \quad (4)$$

Графическое изображение результата бинарной декомпозиции предиката $P(x, y, z)$ представляет собой реляционную сеть, которая показана на рис. 3.

На схеме реляционной сети показаны переменные x, y, z, t ($x, y, z \in \Sigma = \{0, 1\}, t \in T = \{0, 1, 2, 3\}$) и связывающие их отношения P_1, P_2, P_3 . Наборы $(x, \Sigma), (y, \Sigma), (z, \Sigma), (t, T)$, состоящие из переменных и множеств, на которых они заданы, рассматриваются как полюсы сети, а отношения P_1, P_2, P_3 – как ветви сети, соединяющие полюсы сети.

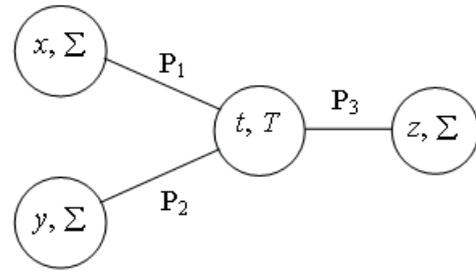


Рис. 3. Реляционная сеть для вычисления импликации

5. Тестирование модели реляционной сети

Функция $z = x \supset y$ вычисляется однозначно по набору значений переменных (x, y) . Рассмотрим работу сети для всех наборов значений переменных x, y .

1. $x=0, y=0$. Аналитическая связь между переменными сети запишется так: $x^0 y^0 z^1 = t^0$. На первом шаге заданные значения переменных x, y порождают соответствующие им множества значений переменной $t - \{0, 1\}$ и $\{0, 2\}$. На втором шаге вычисляется пересечение этих множеств $\{0, 1\} \cap \{0, 2\} = \{0\}$. В результате получаем значение дополнительной переменной $t=0$. Из отношения P_3 получаем значение переменной $z=1$. Результат работы реляционной сети на наборе $x=0, y=0$ представлен на рис. 4.

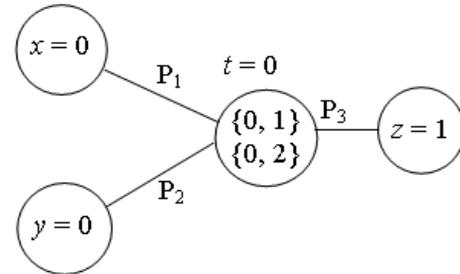


Рис. 4. Результат работы реляционной сети на наборе $x=0, y=0$

2. $x=0, y=1$. Аналитическая связь между переменными сети: $x^0 y^1 z^1 = t^1$. Результат работы сети на наборе $x=0, y=1$ представлен на рис. 5.

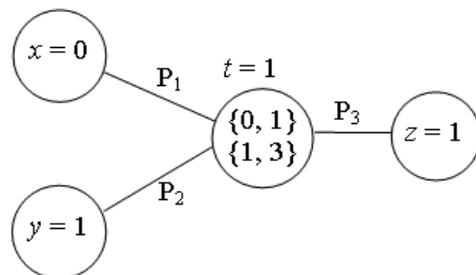


Рис. 5. Результат работы реляционной сети на наборе $x=0, y=1$

3. $x=1, y=0$. Аналитическая связь между переменными сети: $x^1 y^0 z^0 = t^2$. Результат работы сети на наборе $x=1, y=0$ представлен на рис. 6.

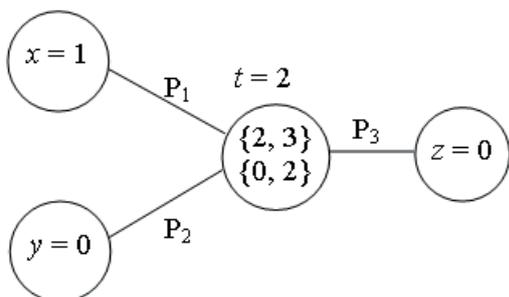


Рис. 6. Результат работы реляционной сети на наборе $x=1, y=0$

4. $x=1, y=1$. Аналитическая связь между переменными сети: $x^1y^1z^1=t^3$. Результат работы сети на наборе $x=1, y=1$ представлен на рис. 7.

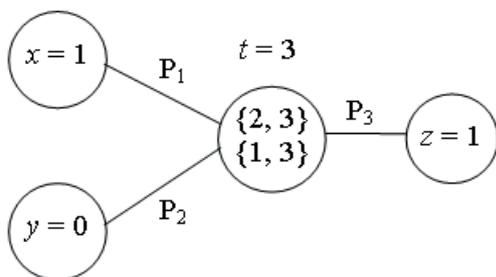


Рис. 7. Результат работы реляционной сети на наборе $x=1, y=1$

А теперь рассмотрим вычисление функции импликации в обратном направлении – от значений переменной z к значениям переменным x, y . Для того чтобы по значению переменной z определить значения переменных x, y , необходимо уточнить значение промежуточной переменной t .

1. $z=0$. Аналитическая связь между переменными сети: $z^0=x^1y^0t^2$. На первом шаге заданное значение переменной z порождает множество значений промежуточной переменной $t = \{2\}$. На втором шаге по значению промежуточной переменной $t=2$ однозначно определяем значения переменных $x=1, y=0$ из отношений P_1 и P_2 соответственно. Результат работы реляционной сети для значения переменной $z=0$ представлен на рис. 8.

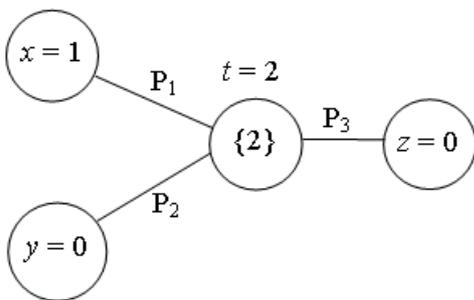


Рис. 8. Результат работы реляционной сети на наборе $x=1, y=0$

2. $z=1$. Аналитическая связь между переменными сети:

$$z^1 = x^0y^0t^0 \vee x^0y^1t^1 \vee x^1y^1t^3.$$

На первом шаге заданное значение переменной z порождает множество значений промежуточной переменной $t = \{0, 1, 3\}$. На следующем шаге однозначно определить значения переменных x, y невозможно. Только задавая последовательно значения промежуточной переменной t равные 0, 1, 3 можно получить соответствующие наборы значений переменных x, y равные (0, 0), (0, 1), (1, 1). На этом анализ работы реляционной сети в прямом и обратном направлениях можно считать завершенным.

В работе [19] приводится математическое описание принципов функционирования реляционной сети. Обработка знаний в ветвях реляционной сети осуществляется линейными логическими операторами. Каждая ветвь реляционной сети представляет собой двунаправленную дугу, которая описывается парой линейных логических операторов. С теоретической точки зрения реляционная сеть может рассматриваться как система взаимодействующих линейных логических операторов. Вопросам разработки теории линейных логических операторов посвящены работы [20, 21]. Линейными логическими операторами $L(A)=B$ и $L(B)=A'$ с ядром $F(x, y)$ называют преобразования

$$B(y) = \exists x \in M(F(x, y) \wedge A(x)), \tag{6}$$

$$A'(x) = \exists y \in N(F(x, y) \wedge B(y)). \tag{7}$$

Преобразование вида (6) – это образ множества $A \subseteq M$ относительно некоторого отображения $f(x)=y$, т.е. множество $B \subseteq N$, образованное из всех образов предметов, принадлежащих множеству A . Преобразование вида (7) представляет собой прообраз множества $B \subseteq N$ относительно некоторого отображения $f(x)=y$ т.е. множество $A' \subseteq M$, образованное из всех прообразов предметов, принадлежащих множеству B .

Для того чтобы построенная модель реляционной сети функционировала так, чтобы из нее можно было извлекать некоторые знания, необходимо решать систему предикатных уравнений

$$\begin{cases} P_1(x, t) = 1, \\ P_2(y, t) = 1, \\ P_3(z, t) = 1. \end{cases}$$

Рассмотрим пример функционирования модели реляционной сети для дуги, которая описывается предикатом $P_1(x, t) = x^0t^0 \vee x^0t^1 \vee x^1t^2 \vee x^1t^3$. Решим предикатное уравнение

$$P_1(x, t) = 1. \tag{8}$$

Решение этого предикатного уравнения можно свести к нахождению образа или прообраза некоторого множества относительно отображения $f(x)=t$ (рис. 9).

Рассмотрим примеры решения предикатного уравнения (8).

1. Дано $x=1$. Найти все возможные значения переменной t . Это значит, что необходимо найти образ одноэлементного множества $A=\{1\}$ относительно отображения $f(x)=t: A = \{1\} \xrightarrow{f} B = \{2, 3\}$ (рис. 10). Если

использовать формулу (6), то получим следующий результат:

$$B(t) = \exists x \in \{1\} (F(x, t) \wedge A(x)) = F(1, t) \wedge A(1) = F(1, t) \wedge 1 = 1^0 t^0 \vee 1^0 t^1 \vee 1^1 t^2 \vee 1^1 t^3 = t^2 \vee t^3.$$

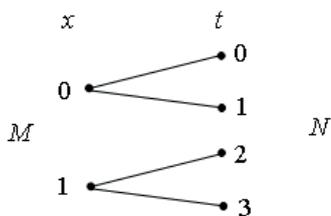


Рис. 9. Отображение $f(x)=t$

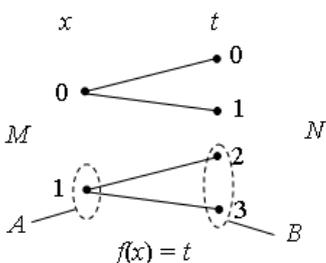


Рис. 10. Образ множества $A=\{1\} \xrightarrow{f} B=\{2, 3\}$

2. Решим теперь обратную задачу. Дано $t=3$. Найти всевозможные значения переменной x . Это значит, что для решения этой задачи необходимо найти прообраз одноэлементного множества $B=\{3\}$ относительно отображения $f(x)=t: B=\{3\} \xrightarrow{f} A'=\{1\}$ (рис. 11).

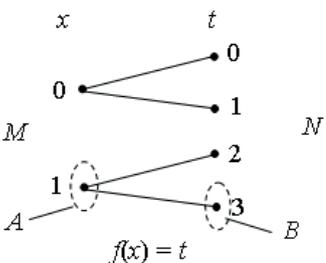


Рис. 11. Прообраз множества $B=\{3\} \xrightarrow{f} A'=\{1\}$

Подставляя в формулу (7) значение переменной $t=3$, получим следующий результат:

$$A'(x) = \exists t \in \{3\} (F(x, t) \wedge B(t)) = F(x, 3) \wedge B(3) = F(x, 3) \wedge 1 = x^0 t^0 \vee x^0 t^1 \vee x^1 t^2 \vee x^1 t^3 = x^1.$$

Аналогичным способом можно отыскивать остальные образы и прообразы подмножеств A и B множеств M и N .

6. Построение направленных схем реляционной сети

На следующем этапе разработаем метод построения направленных схем реляционной сети, моделирующей работу функции импликации. Запишем связь

переменных x и t в виде следующих предикатных уравнений:

$$x^0 y^0 = t^0, x^0 y^1 = t^1, x^1 y^0 = t^2, x^1 y^1 = t^3.$$

Построим на основе этих уравнений прямую направленную схему связей переменных x , y и t (рис. 12).

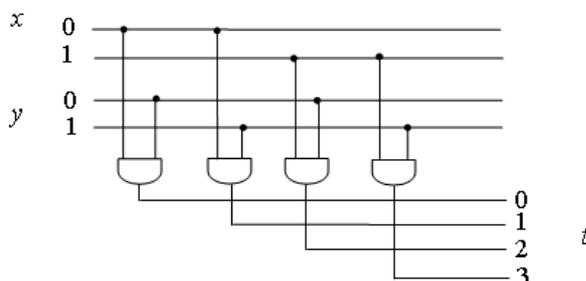


Рис. 12. Направленная схема связей переменных x, y и t (прямая)

Обратная связь переменных t и x, y может быть записана в виде следующих предикатных уравнений:

$$t^0 \vee t^1 = x^0, t^2 \vee t^3 = x^1, t^0 \vee t^2 = y^0, t^1 \vee t^3 = y^1.$$

На основе этих уравнений можно построить обратную направленную схему связей переменных t и x, y (рис. 13).

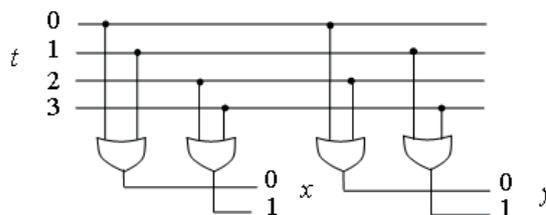


Рис. 13. Направленная схема связей переменных t и x, y (обратная)

Прямая связь между предметной переменной z и дополнительной переменной t записывается с помощью следующих предикатных уравнений:

$$z^1 = t^0, z^1 = t^1, z^0 = t^2, z^1 = t^3.$$

На основе этих уравнений построим прямую направленную схему связей переменной z и дополнительной переменной t (рис. 14).

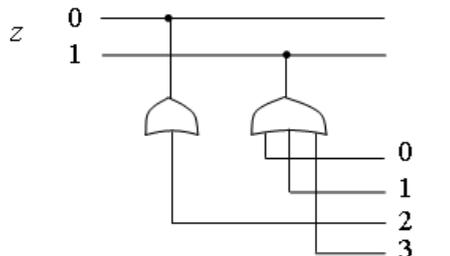


Рис. 14. Направленная схема связей переменных z и t (прямая)

Обратная связь между дополнительной переменной t и предметной переменной z может быть представлена следующими предикатными уравнениями:

$$t^0 \vee t^1 \vee t^3 = z^1, t^2 = z^0.$$

На основе этих уравнений строим обратную направленную схему связей переменных t и z (рис. 15).

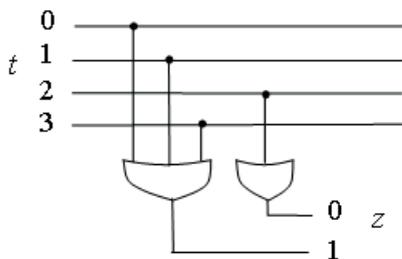


Рис. 15. Направленная схема связей переменных t и z (обратная)

Если объединить схемы, представленные на рис. 13 и 15, то получим направленную схему реляционной сети (прямую) (рис. 16).

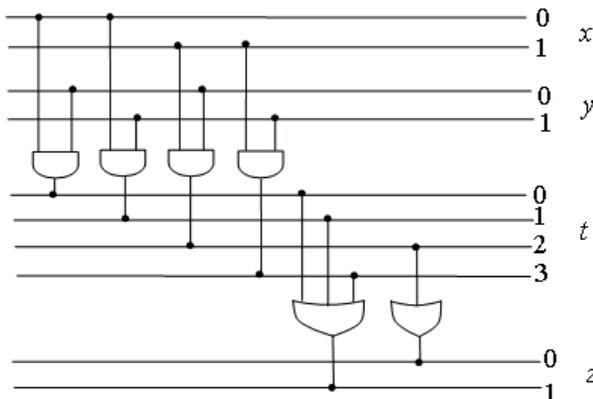


Рис. 16. Направленная схема реляционной сети (прямая)

Аналогичным образом, объединив схемы, показанные на рис. 14 и 16, получим направленную схему реляционной сети (обратную) (рис. 17).

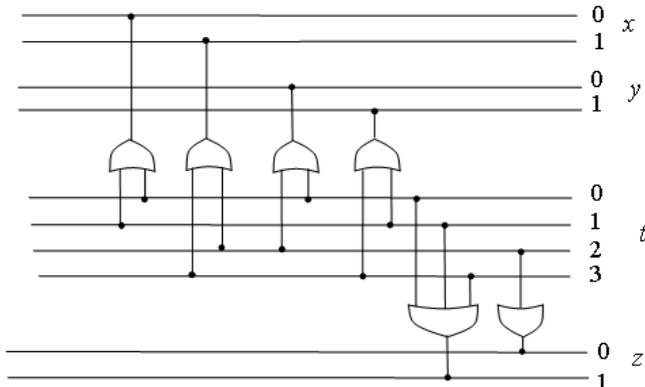


Рис. 17. Направленная схема реляционной сети (обратная)

Таким образом, в результате объединения направленных схем связей между отдельными переменными

реляционной сети, были получены прямая и обратная схемы реляционной, задающие связи между всеми переменными модели x, y, z и t .

7. Выводы

Таким образом, в работе было выполнено моделирование реляционной сети для очень важного при создании баз знаний отношения импликации. Реляционная сеть представляет собой графическое изображение результата бинарной декомпозиции предиката, задающего отношение импликации. На схеме реляционной сети отображаются переменные и связывающие их отношения.

Правильность функционирования реляционной сети была проверена в процессе ее потактового тестирования на различных наборах переменных в прямом и обратном направлениях. Для математического описания принципов функционирования реляционной сети были использованы линейные логические операторы. Реляционная сеть рассматривается как система взаимодействия линейных логических операторов. Рассмотрен пример функционирования модели реляционной сети для одной из дуг сети. Были найдены образы и прообразы множеств значений переменных относительно соответствующих отображений.

Используя метод построения направленных схем реляционной сети, были разработаны схемы, задающие связи между отдельными переменными в прямом и обратном направлениях, на основе которых были созданы объединенные схемы реляционной сети, определяющие связи между всеми переменными модели.

Построенные направленные реляционные сети можно использовать при создании баз знаний для построения правил продукции. Операция импликации используется для записи таких правил продукции на языке алгебры предикатов. Направленные схемы реляционной сети могут стать одной из важных составляющих параллельных баз знаний и логического вывода, которые могут быть реализованы аппаратно с помощью программируемых матриц FPGA [21].

Литература

1. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain [Text] / F. Rosenblatt // Psychological Review. – 1958. – Vol. 65, Issue 6. – P. 386–408. doi: 10.1037/h0042519
2. Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities [Text] / J. J. Hopfield // Proceedings of the National Academy of Science. – 1982. – Vol. 79, Issue 8. – P. 2554–2558. doi: 10.1073/pnas.79.8.2554
3. Lipman, R. An Introduction to Computing with Neural Nets [Text] / R. Lipman // IEEE Acoustic Speech and Signal Processing Magazine. – 1987. – Vol. 4, Issue 2. – P. 4–22. doi: 10.1109/massp.1987.1165576
4. Булкин, В. И. Математические модели знаний и их реализация с помощью алгебропредикатных структур [Текст]: монография / В. И. Булкин, Н. В. Шаронова. – Издатель: Дмитренко Л. Р., Донецк, 2010. – 304 с.

5. Шабанов-Кушнарченко, Ю. П. Теория интеллекта. Математические средства [Текст] / Ю. П. Шабанов-Кушнарченко. – Х.: Вища шк., 1984. – 144 с.
6. Булкин, В. И. Представление алгебропредикатных структур в виде ассоциативно-логических преобразователей [Текст] / В. И. Булкин // Искусственный интеллект. – 2012. – № 3. – С. 6–17.
7. Stern, R. Potential-based bounded-cost search and Anytime Non-Parametric [Text] / R. Stern, A. Felner, J. Berg, R. Puzis, R. Shah, K. Goldberg // Artificial Intelligence. – 2014. – Vol. 214. – P. 1–25. doi: 10.1016/j.artint.2014.05.002
8. McGreggor, K. Fractals and Ravens [Text] / K. McGreggor, M. Kunda, A. Goel // Artificial Intelligence. – 2014. – Vol. 215. – P. 1–23. doi: 10.1016/j.artint.2014.05.005
9. Karaboga, D. A comprehensive survey: artificial bee colony (ABC) algorithm and applications [Text] / D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga // Artificial Intelligence Review. – 2014. – Vol. 42, Issue 1. – P. 21–57. doi: 10.1007/s10462-012-9328-0
10. Bunkhumpornpat, C. DBSMOTE: Density-Based Synthetic Minority Over-sampling Technique [Text] / C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap // Applied Intelligence. – 2012. – Vol. 36, Issue 3. – P. 664–684. doi: 10.1007/s10489-011-0287-y
11. Sun, G. Dynamic partition search algorithm for global numerical optimization [Text] / G. Sun, R. Zhao // Applied Intelligence. – 2014. – Vol. 41, Issue 4. – P. 1108–1126. doi: 10.1007/s10489-014-0587-0
12. Jain, L. C. A review of online learning in supervised neural networks [Text] / L. C. Jain, M. Seera, C. P. Lim, P. Balasubramaniam // Neural Computing and Applications. – 2014. – Vol. 25, Issue 3–4. – P. 491–509. doi: 10.1007/s00521-013-1534-4
13. Peteiro-Barral, D. A survey of methods for distributed machine learning [Text] / D. Peteiro-Barral, B. Guijarro-Berdiñas // Progress in Artificial Intelligence. – 2013. – Vol. 2, Issue 1. – P. 1–11. doi: 10.1007/s13748-012-0035-5
14. Charlesworth A. The Comprehensibility Theorem and the Foundations of Artificial Intelligence [Text] / A. Charlesworth // Minds and Machines. – 2014. – Vol. 24. – Issue. 4, P. 439–476.
15. Лещинский, В. А. Модели бинарных логических сетей и их применение в искусственном интеллекте [Текст] : дис. ... канд. техн. наук / В. А. Лещинский. – Харьков, 2006. – 160 с.
16. Ефимова, И. А. О методе построения моделей бинарных логических сетей [Текст] / И. А. Ефимова, В. А. Лещинский // Восточно-Европейский журнал передовых технологий. – 2005. – № 4. – С. 121–124.
17. Бондаренко, М. Ф. О мозгоподобных ЭВМ [Текст] / М. Ф. Бондаренко, З. В. Дударь, И. А. Ефимова, В. А. Лещинский, С. Ю. Шабанов-Кушнарченко // Радиоэлектроника и информатика. – 2004. – № 2. – С. 89–105.
18. Лещинская, И. А. О методе построения направленных схем реляционных сетей на примере отношения эквивалентности [Текст] / И. А. Лещинская // Системи обробки інформації. – 2010. – № 1 (82). – С. 75–81.
19. Ротин, И. М. Линейные и билинейные логические операторы и их применение в автоматизированных информационных системах [Текст] : дис. ... канд. техн. наук / И. М. Ротин. – Харьков, 1994. – 163 с.
20. Вечирская, И. Д. Линейные логические преобразования и их применение в искусственном интеллекте [Текст] : дис. ... канд. техн. наук / И. Д. Вечирская. – Харьков, 2007. – 149 с.
21. Bondarenko, M. F. Logic networks application for computing process organization [Text] / M. F. Bondarenko, I. V. Hahanova // Радиоэлектроника и информатика. – 2003. – № 3. – С. 150–156.