

УДК 519.712.1

МОДЕЛЬ ЛОГИЧЕСКОЙ СЕТИ С ПРЕДИКАТНЫМИ ОПЕРАЦИЯМИ

Рудометкина М.Н., Спицын В.Г.

ФГАОУ ВО «Национальный исследовательский Томский политехнический университет»,
Томск, e-mail: mn.rud@inbox.ru

В статье развивается теория логических сетей – алгебро-логические методы и модели описания информационных объектов и процессов. В качестве математической базы используется аппарат конечных предикатов. В результате анализа средств формального описания информационных процессов был выбран аппарат логических сетей, позволяющий строить полные модели. Этот аппарат был создан для моделирования статических объектов, поэтому при описании информационных процессов возникла необходимость в модификации модели логической сети, позволяющей учитывать динамику процессов. Предложена двуслойная структура логической сети. Первый слой представляет собой систему бинарных предикатов, а второй – систему предикатных операций. Представление гибкого процесса модифицированной логической сетью позволяет адаптировать модель к предметной области.

Ключевые слова: логическая сеть, алгебра предикатов, гибкий процесс, лог процесса, process mining

LOGICAL NETWORK MODEL WITH PREDICATE OPERATIONS

Rudometkina M.N., Spitsyn V.G.

National Research Tomsk Polytechnic University, Tomsk, e-mail: mn.rud@inbox.ru

The paper develops the theory of logical networks – algebraic-logical methods and models describing data objects and processes. As a mathematical framework used machine finite predicates. An analysis of the means of formal description of information processes was selected device logical networks, allowing to build complete models. This unit was created to simulate the static objects, so when describing information processes was necessary to modify the logical network model, to take account of the dynamics of processes. Proposed a two-layer structure of the logical network. The first layer is a system of binary predicate, and the second – a system predicate operations. Presentation flexible process modified the logical network to adapt the model to the problem domain.

Keywords: logical network, predicate algebra, flexible process, the log process, process mining

Для целей формализации конечных объектов и процессов, оперирующих символьной информацией, разработан универсальный аппарат на базе алгебры конечных предикатов [1]. Этот аппарат развивается около 50 лет и за это время доказал свою эффективность в области теории искусственного интеллекта [2].

Алгебра конечных предикатов является полной, т.е. ее средствами можно описывать любые конечные отношения. Однако информационные процессы – это не только конечные отношения, но и действия над ними. Аппаратом второй ступени является алгебра предикатных операций. Язык алгебры предикатных операций тоже универсален, но он используется для выражения действий над отношениями.

Теория конечных предикатов имеет некоторые результаты по декомпозиции предикатов, т.е. взаимно однозначного преобразования предикатов одной размерности в предикаты другой размерности. Преобразование системы предикатов, имеющих, вообще говоря, произвольную конечную размерность, в систему бинарных предикатов дает ряд преимуществ при моделировании информационных объектов и процессов. Основное из них – возможность перехода от последовательного алгоритмического

вычисления к параллельному, что важно при решении задач, требующих больших объемов вычислений. Графическое представление системы бинарных предикатов называется логической сетью.

Логическая сеть состоит из полюсов и ветвей, ветви соединяют полюсы. Каждому полюсу соответствует своя предметная переменная x_i с областью определения A_i ($i = 1, m$). Пара полюсов x_i и x_j , соединенных ветвью $K(x_i, x_j)$, реализуют линейный логический оператор первого рода.

Рассматриваемый аппарат предназначен для решения задач идентификации структур в области искусственного интеллекта [1, 2]. Однако традиционные логические сети не предназначены для обработки структур, отражающих выполнение процессов произвольной физической природы (процессы мышления, производства, поиска данных в сети интернет и т.п.), поскольку последние обязательно направлены во времени. Такие процессы могут быть представлены в виде последовательности событий, каждое из которых обладает временной меткой. Следует отметить, что важно не абсолютное значение, задаваемой временной меткой, а порядок событий (действий процесса).

В связи с изложенным возникает проблема предикатного описания процессов

произвольной физической природы на основе логических сетей, рассмотрению которой и посвящена данная работа.

1. Анализ работ в данной области

В работе [3] был рассмотрен способ построения и настройки модели гибкого процесса и детализированы особенности этапа адаптации модели. Модель гибкого процесса, адаптированной к предметной области, в общем виде представлена системой бинарных предикатов

$$M^P = \{R_j \mid \forall R_j \exists P_k, R_j \in M, P_k \in P, \\ k, j = \overline{1, n}\}, \quad (1)$$

где R_j – бинарные предикаты, а P_k – объекты предметной области.

В статье [4] предложена предикатная процессная модель на основе дерева процессов и аппарата конечных предикатов. Предложенный метод позволяет объединить базовые способы слияния логов и построения модели процесса средствами process mining. Были введены два типа узлов логической сети. Вершины первого типа $x \in V^*$ отражают конкретные действия (процедуры) процесса. С каждой вершиной связаны определенные действия процесса с помощью бинарных предикатов:

$$\forall x_i \in V^* \exists! x_j, R_k(x_i, x_j), i, j = \overline{1, n}, k = \overline{1, m}.$$

Соответственно, модель (1) дополняется следующим образом:

$$M^P = \{R_k \mid \forall R_k(x_i, x_j) \in M, x_i \in V^* i, j = \overline{1, n} \exists P_k \in P, k = \overline{1, m}\} \quad (2)$$

Вершина называется вершиной второго типа $x' \in V^{**}$, если из вершины x' графа логической сети выходит две дуги. Адаптация процессной модели к предметной области

достигается за счет применения известных операторов адаптации базовых элементов модели: последовательное выполнение, выбор, параллельное выполнение, цикл.

$$\rightarrow (R_k(x', x_i), R_l(x', x_j), r), \quad (3)$$

$$\text{XOR}(R_k(x', x_i), R_l(x', x_j)) = R_k(x', x_i) \oplus R_l(x', x_j) \quad (4)$$

или

$$\text{OR}(R_k(x', x_i), R_l(x', x_j)) = R_k(x', x_i) \vee R_l(x', x_j);$$

$$\text{AND}(R_k(x', x_i), R_l(x', x_j)) = R_k(x', x_i) \wedge R_l(x', x_j); \quad (5)$$

$$\Omega(R_k(x', x_i), R_l(x', x_j), t). \quad (6)$$

В приведенных работах рассмотрено общее представление графа процесса, однако вопросы моделирования таких процессов логической сетью требуют дальнейшего развития.

2. Постановка задачи

Задача данной работы заключается в построении модели логической сети с предикатными операциями вида (3)-(6), для формализации поведения, а также оценки процессов различной физической природы, которые характеризуются заданной последовательностью действий.

2.1 Модифицированная логическая сеть

Вводя в конструкцию логической сети операции типа (3)-(6), мы уходим от общепринятого определения логической сети,

как системы бинарных предикатов. В-первых, логическая сеть – это неориентированный граф, а полученная нами конструкция – ориентированный граф. Во-вторых, мы используем не только бинарные предикаты, но и тернарные. В-третьих, и это, по нашему мнению, основное отличие, в полученной конструкции используются предикатные операции, т.е. предикаты второго порядка [3]. Назовем такую сеть логической сетью с предикатными операциями.

Графически, за счет применения предикатных операций, логическая сеть из плоской становится объемной, рис. 1. Такая конструкция имеет физический смысл, т.к. отражает реальное разделение процедур процесса на элементарные базовые действия и операции над этими процедурами. Полученная модификация логической сети явля-

ется естественной адаптацией ее структуры для рассматриваемой предметной области – построения гибкой процессной модели.

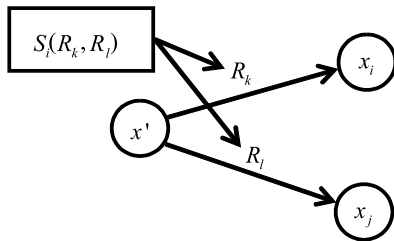


Рис. 1. Логическая сеть с предикатной операцией $S_i(R_k, R_l)$

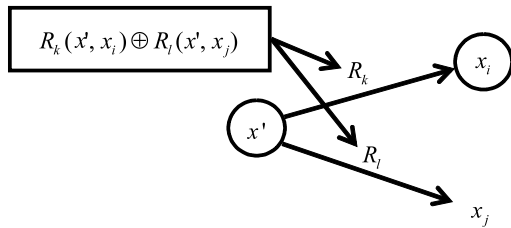


Рис. 2. Эксклюзивный выбор

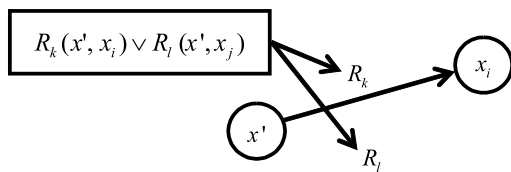


Рис. 3. Неэксклюзивный выбор

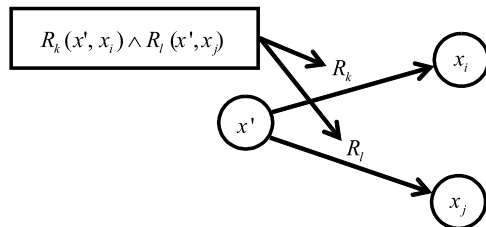


Рис. 4. Параллельное выполнение

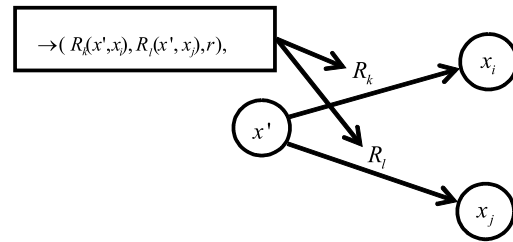


Рис. 5. Последовательное выполнение

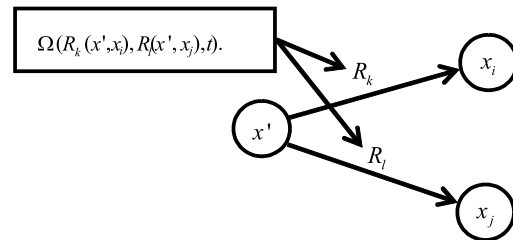


Рис. 6. Предикатное представление цикла

Иллюстрация предикатного представления эксклюзивного и неэксклюзивного выбора, а также параллельного выполнения приведена на рис. 2–6.

Для операторов последовательного выполнения и цикла важно задать порядок вершин (слева-направо) вне зависимости от их типа, поскольку порядок вершин задает порядок выполнения действий процесса:

В случае цикла, одна вершина соответствует началу цикла, содержащему условие повторения (обычно соответствует фрагменту с «do», «for»).

Правая вершина соответствует собственно операции (группе операций) цикла, после которых происходит переход к новому повторению.

В целом модель процесса представляется в виде иерархии предикатных операций вида (3)–(6) на верхних уровнях, а также бинарных предикатов на нижнем уровне:

$$M = \langle S_k, R_i \rangle,$$

где

$$R_k(x_i, x_j) \in M, k = \overline{1, m}, x_i \in (V^* \cup V^{**}), i, j = \overline{1, n}, \quad (7)$$

$$S_l(R_i, R_j) \in S, l = \overline{1, p}.$$

Исходя из общих представлений о процессе, полная модель процесса обязательно должна иметь одно начальное и одно конечное состояние, а также содержать пути достижения конечного состояния. Математически это можно описать структурой, имеющей:

- начальную переменную x_1 (вход модели);
- конечную переменную x_m (выход модели);

- систему бинарных предикатов $R_k(x_i, x_j)$, описывающих процедуры процесса;
- систему предикатных операций $S_l(R_i, R_j)$;
- матрицу $Z^R(R_k, t)$, определяющую последовательность вычисления бинарных предикатов $R_k(x_i, x_j)$, t – номер такта работы логической сети, число тактов определяется максимальной длиной цепочек процедур в логической сети, моделирующей процесс;

– матрицу $Z^S(S_p, t)$, определяющую, как должны применяться предикатные операции $S_l (R_p, R_j)$.

$$M = \langle S_k, R_i, Z^R, Z^S \rangle. \quad (8)$$

3. Пример адаптации модели модифицированной логической сети к предметной области

Рассмотрим пример структуры логической сети с предикатными операциями.

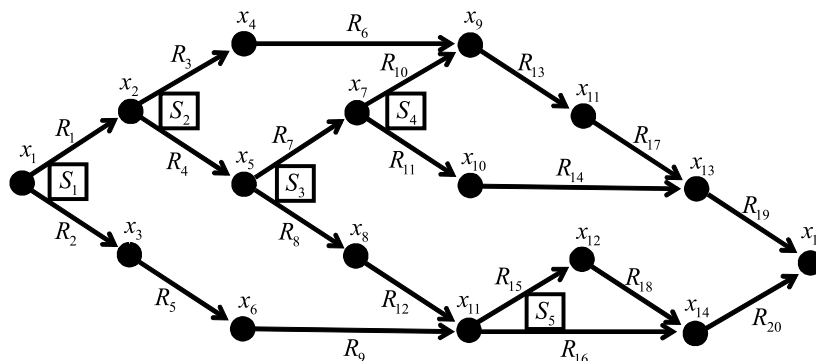


Рис. 7. Логическая сеть с предикатными операциями

Имеем логическую сеть с 15-ю событиями $v_i \in V, i = \overline{1,15}$, обозначенными переменными x_1, x_2, \dots, x_{15} , 20-ю действиями $e_i \in E, i = \overline{1,20}$, которые описывают попарные связи между событиями, обозначенными

предикатами $R_i, i = \overline{1,20}$ и 5-ю предикатными операциями $S_l, l = \overline{1,5}$.

Матрица $Z^R(R_k, t)$ для логической сети, изображенной на рис. 7, имеет следующий вид:

Таблица 3

Матрица последовательности вычисления бинарных предикатов R_k

| t \ k | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 1 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 1 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |

Как видно из табл. 3, некоторые действия могут выполняться в течение двух тактов ($k = 6, 9, 14, 16$). Если несколько действий выполняются на одном и том же такте, можно определить последовательность их выполнения, если это необходимо. Например, на 2-м такте вначале нужно выполнить 4-е действие, затем 3-е. Если последовательность выполнения действий в пределах одного такта не важна, то в матрице ставятся единицы, например, на 1-м такте действия 1 и 2 могут выполняться в любом порядке.

Матрица $Z^S(S_p, t)$ для логической сети, изображенной на рис. 7, имеет следующий вид:

Таблица 4
Матрица выполнения предикатных операций $S_l (R_i, R_j)$.

| t \ l | 1 | 2 | 3 | 4 | 5 |
|-------|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 |

Матрица выполнения предикатных операций имеет приоритет над матрицей

последовательности вычисления бинарных предикатов (в случае конфликта правил). В случае если предикатная операция цикла содержит несколько повторений, то они могут выполняться в течение одного такта логической сети.

При предлагаемом иерархическом предикатном представлении каждый предикат нижележащего уровня соответствует переменной предиката вышележащего уровня и, за исключением нижнего уровня, реализует соответствующую функциональность.

Иерархическое представление гибкого процесса позволяет адаптировать модель к предметной области простым отсечением

«избыточных» ветвей путем удаления либо преобразования связанных с вершинами предикатов. В результате последовательность действий процесса корректно упрощается. Адаптация модели к предметной области должна проводиться на основании дополнительных знаний о моделируемом процессе. Рассмотрим процесс адаптации модели на примере логической сети, изображенной на рис. 7. В качестве дополнительных знаний введем стоимость (вес) действий. Кроме того, необходимо учитывать особенности выполнения предикатных операций.

Зададим веса дуг графа логической сети.

Таблица 5

Матрица весов дуг графа логической сети

| | | | | | | | | | | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| Дуга | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Вес | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 2 | 2 | 1 | 2 | 1 | 3 | 1 | 1 | 2 | 2 | 4 | 2 | 3 |

Найдем наименьший путь на этом графе от x_1 к x_{15} . Рассмотрим все варианты таких путей. Через запятую указаны номера дуг графа:
 1, 3, 6, 13, 17, 19. Вес равен 12.
 1, 4, 7, 10, 13, 17, 19. Вес равен 12.
 1, 4, 7, 11, 14, 19. Вес равен 9.

1, 4, 8, 12, 15, 18, 20. Вес равен 14.
 1, 4, 8, 12, 16, 20. Вес равен 11.
 2, 5, 9, 15, 18, 20. Вес равен 12.
 2, 5, 9, 16, 20. Вес равен 9.
 Получили два равноценных варианта с весом пути, равным 9 – №3 и №7, рис. 8.

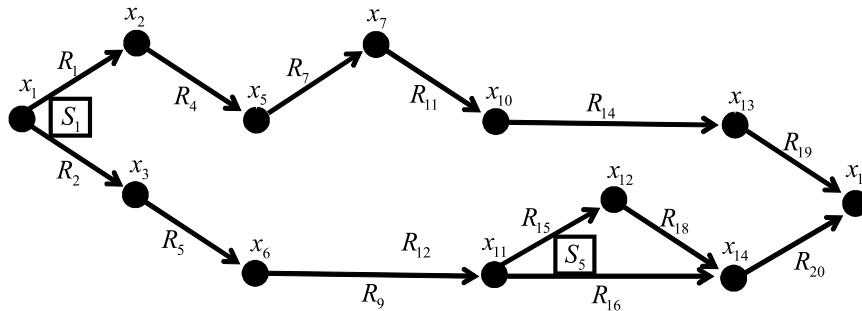


Рис. 8. Минимальные пути на графе логической сети без учета предикатных операций

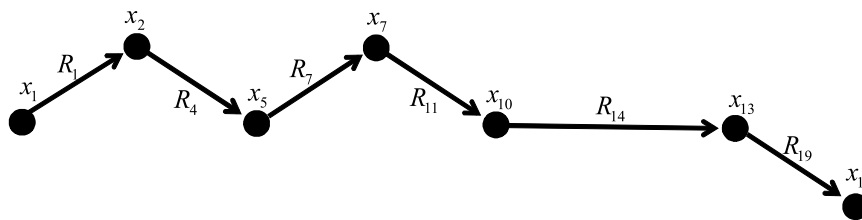


Рис. 9. Минимальный путь на графе логической сети с учетом предикатных операций

Теперь учтем влияние предикатных операций S_i . Зададим эти операции. Пусть

$$S_1 = R_1(x_1, x_2) \oplus R_2(x_1, x_3),$$

$$S_2 = R_3(x_2, x_4) \vee R_4(x_2, x_5),$$

$$S_3 = R_7(x_5, x_7) \vee R_8(x_5, x_8),$$

$$S_4 = R_{10}(x_7, x_9) \wedge R_{11}(x_7, x_{10}),$$

$$S_5 = \Omega(R_{15}(x_{10}, x_{12}), R_{16}(x_{10}, x_{14}), 3).$$

Предикатная операция S_1 определяет эксклюзивный выбор между R_1 и R_2 , на вес путей №3 и №7 это не влияет. А предикатная операция S_5 задает цикл из трех повторений,

что делает вес пути № 7 большим. Поэтому минимальный вес имеет путь № 3:

В результате адаптации модели получили из системы из 20 бинарных предикатов, плюс пяти тернарных предикатных операций систему из 6 бинарных предикатов.

Заключение

В статье предложена модель направленной логической сети с предикатными операциями (предикаты второго порядка) следующих типов: AND, OR, XOR, цикл, последовательность. В отличие от известной модели логической сети, представляющей собой систему бинарных предикатов, предложенная модификация имеет двуслойную структуру – к традиционной модели логической сети добавлен уровень предикатных операций. Разработанная модель обеспечивает возможность формализации

поведения, а также оценки процессов различной физической природы, которые характеризуются заданной последовательностью действий или событий, фиксирующих выполнение действий.

Область практического применения предлагаемой модели – интеллектуальный анализ процессов.

Список литературы

1. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Мозгоподобные структуры: Справочное пособие. Том первый. Под редакцией акад. НАН Украины И.В. Сергиенко. – К.: Наукова думка, 2011. – 460 с.
2. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Теория интеллекта. Учебник – Харьков: изд-во СМИТ, 2007 – 576 с.
3. Рудометкина М.Н. Формирование и адаптация модели гибкого процесса на основе анализа логов. Динамика сложных систем XXI-век № 4, Москва, 2014. – С 90.
4. Рудометкина М.Н. Предикатная модель гибкого процесса. Международный журнал прикладных и фундаментальных исследований №10, часть 2, Российской Академии Естествознания, Москва, 2014. – Принято к печати.