

УДК 004.89

АПЛИКАТИВНО-ФРЕЙМОВАЯ МОДЕЛЬ ПРЕДСТАВЛЕНИЯ МЕДИЦИНСКИХ ЗНАНИЙ

¹Лебедев Г.С., ²Лосев А.Ю.

¹ФГБУ «Центральный научно-исследовательский институт организации и информатизации здравоохранения», Москва, e-mail: geramail@rambler.ru;

²ФГУП «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «Стандартинформ»), Москва, e-mail: alexlos07@mail.ru

Созданный в 2015 году Институт развития Интернета определил в качестве одной из основных задач развития Интернета в медицине создание и применение интеллектуальных систем поддержки принятия решений. Основу систем такого класса составляет система управления базой знаний, в которой медицинские знания формализуются и накапливаются. Модель представления медицинских знаний должна адекватно соответствовать процессу принятия решения. Авторами предлагается такая модель, основанная на графе потоков данных, описанной средствами комбинаторной логики. Единицей хранения знаний в разработанной модели предлагается аппликативный фрейм, содержащий в себе входные атрибуты и целевой атрибут, связь между которыми задается функцией аппликации. Процедура вывода решения является редукцией дерева аппликативных фреймов. Приводятся формальные правила редукции.

Ключевые слова: интеллектуальная система, принятие решения, модель представления знаний, медицинская информационная система

APPLICATIVE-FRAME MODEL OF MEDICAL KNOWLEDGE REPRESENTATION

¹Lebedev G.S., ²Losev A.Yu.

¹Federal Research Institute for Health Organization and Informatics of Ministry of Health of the Russian Federation, Moscow, e-mail: geramail@rambler.ru;

²Russian Scientific and Technical Centre for Information on Standardization, Metrology and Conformity Assessment, Moscow, e-mail: alexlos07@mail.ru

Created in 2015, the Internet Development Institute has identified as one of the main objectives of development of the Internet in medicine the creation and application of intelligent decision support systems. The basis of this systems is knowledge base management system in which medical knowledge is formalized and accumulate. Medical knowledge representation model should adequately meet the decision-making process. The authors propose such a model, based on the graph data streams described by means of combinatorial logic. The unit of storage of knowledge in the developed model proposed applicative frame containing a input attributes and the target attribute, the relationship between them is given by application function. Conclusion procedure is a reduction applicative frames tree. They gives the formal rules of the reduction.

Keywords: intelligent system, decision-making, the knowledge representation model, medical information system

Проблемы создания прикладных интеллектуальных систем поддержки принятия врачебных решений (ИСППВР) и экспертных систем широко обсуждаются в литературе, достаточно полный обзор которой приведен в [8]. Главным достоинством любой системы является качество ее базы знаний. Поэтому главный вопрос, решаемый при создании ИСППВР – это представление медицинских знаний.

Для решения интеллектуальной задачи в информационной системе необходимо решить следующие основные задачи:

- разработать формальную схему процесса решения задачи, представляющую из себя модель знаний о решении задачи;
- разработать формализмы представления этой модели (этих знаний) в ИСППВР, т.е. модель представления знаний или логическую структуру базы знаний и формальные процедуры вывода решений;

- реализовать предложенную модель представления знаний и процедуры вывода решения.

Процесс решения задачи включает, как правило, два этапа: представление задачи и поиск решения. Успех решения задачи в значительной мере определяется формой ее представления. Формы и способы решения задачи зависят как от природы самой задачи, так и от человека, ее решающего.

Поиск удобной формы представления задачи является трудноформализуемым творческим процессом. Рассмотрим подход к представлению задачи выработки врачебного решения.

Представление задачи принятия врачебного решения

Описаны следующие наиболее употребительные способы представления задач:

- представление в пространстве состояний;

- представление путем сведения задачи к подзадачам;
- представление в виде теорем;
- комбинированное представление.

Представление задач в пространстве состояний заключается в описании всех состояний, или только начальных состояний и операторов, отображающих одни состояния в другие, и целевого состояния. Возможны различные формы описания состояний задачи (строки, векторы, матрицы, графы). Операторы могут быть заданы с помощью таблицы, связывающей каждое входное состояние с некоторым выходным.

Поиск решения в пространстве состояний заключается в нахождении последовательности операторов, отображающей начальное состояние в целевое. Решением задачи будет указанная последовательность операторов. Очевидно, что такое представление задачи задается тройкой (S, F, G) , где S – множество начальных состояний, F – множество операторов, отображающих одно состояние в другое, G – множество целевых состояний.

Представление, сводящее задачу к подзадачам, заключается в разбиении исходной задачи на множество подзадач, отдельное решение которых дает решение исходной задачи. Каждая из подзадач, в свою очередь, может быть также разбита на подзадачи и так до тех пор, пока разбиение будет возможно. Очевидно, что разбиение заканчивается при достижении подзадач, решение которых известно. Такие задачи называются элементарными.

Представление в виде теорем используется для задач, которые можно представить в виде теорем, подлежащих доказательству. В этом случае задаются посылки (множество известных истинных утверждений – аксиом), затем формулируется теорема, доказательство которой позволяет получить решение поставленной задачи.

Комбинированное представление в общем случае используется для эффективного решения отдельных задач путем комбинирования различных способов представления задач.

Рассмотрим специфику принятия врачебного решения. Формально модель принятия врачебного решения можно представить в виде

$$M = \langle X, R, A \rangle, \quad (1)$$

где X – множество исходных данных для принятия решения, т.е. текущих показателей состояния здоровья; R – множество всевозможных решений, принимаемых на различных этапах обобщения информации; A – множество операций обработки информации и принятия решения.

Процесс обработки информации осуществляется поэтапно, и на каждом этапе реализуются лишь элементарные операции по переработке информации, то есть модель вида (1) всегда представима в виде последовательности:

$$M = \langle X, R_1, a_1 \rangle \langle R_1, R_2, a_2 \rangle \dots \langle R_{i-1}, R_i, a_i \rangle, \quad (2)$$

где X – множество исходных данных; R_1, \dots, R_i – множества промежуточных решений (обобщенных показателей объекта управления); a_1, \dots, a_i – элементарные операции обработки (обобщения) информации; R_i – конечное решение;

$$R = \{R_1, \dots, R_i\};$$

$$A = \{a_1, \dots, a_i\};$$

$\langle R_{n-1}, R_n, a_n \rangle$ – элементарная n -операция обобщения информации ($n = 1, \dots, i$); i – количество элементарных операций обобщения информации для выработки решения (глубина поиска решения в традиционном понимании).

Отметим, что каждое множество промежуточных решений можно рассматривать как множество исходных данных для очередного этапа обобщения информации, т.е., с содержательной точки зрения, R является состоянием пациента, на фиксированном уровне обобщения информации, а элементарные операции определяют правила перехода от нижнего к верхнему уровню обобщения информации о состоянии здоровья.

Выражение (2) можно представить в виде совокупности функций обработки информации, каждая из которых отображает соответствующее подмножество множества исходных данных X в одно решение множества R . Каждая такая функция представима в виде

$$R_m = f_m(x_1, \dots, x_N), \quad (3)$$

где y_m – элемент множества Y , $m = 1, \dots, I$; I – количество элементарных операций обработки информации; N – количество исходных данных; f_m – функция, соответствующая элементарной операции обработки информации a_m .

В такой нотации процесс принятия решения можно представить в виде формулы

$$M = f_i(f_{i-1}(\dots f_1(x_1, \dots, x_N) \dots)). \quad (4)$$

Таким образом, модель процесса выработки врачебного решения (1) можно представить в виде графа функций обработки информации, где вершинами графа являются имена (описания функций), а дугами – направления переходов информации между функциями. Такую модель можно назвать аппликативной моделью предметной области (процесса выработки решения).

Построение аппликативно-фреймовой модели представления знаний

С содержательной точки зрения, аппликативную функцию обработки информации можно трактовать как одну из задач (подзадач), на которые можно декомпозировать исходную задачу (задачу выработки решения). В этой интерпретации можно считать, что задача выработки врачебного решения представима в виде разбиения исходной задачи на подзадачи. Задача описывается в виде графа, называемого графом редукции задачи. При этом вершинам будут соответствовать задачи, а дугам – операторы редукции задачи. Причем корню дерева соответствует исходная задача, вершинам 1-го уровня – задачи, непосредственно порожденные исходной задачей, вершинам 2-го уровня – задачи, порождаемые задачами 1-го уровня, и т.д. Кроме того, решение задачи соответствует значению функции обработки информации, а исходные данные задачи соответствуют параметрам функции обработки информации.

Если для решения вышестоящей задачи необходимо решение всех вытекающих из нее подзадач, то узлы графа, соответствующие этим подзадачам, называются связанными; в противном случае узлы называются несвязанными. Вершины графа, соответствующие задачам, которые могут быть решены, называются разрешимыми, не имеющим решения – неразрешимыми. Вершины, соответствующие элементарным задачам на самом нижнем уровне разбиения, называются заключительными. Следовательно, вершина, не являющаяся заключительной, будет разрешима, если разрешимы все ее дочерние связанные вершины или хотя бы одна из дочерних несвязанных вершин.

Поиск решения при сведении задачи к подзадачам опирается на граф редукции задачи, который является графом типа И-ИЛИ. Для описания формальных процедур поиска решения необходимо определить формализм неявного описания графа.

Традиционен подход к неявному представлению графа посредством описания операторов порождения подзадач [21]. Оператор преобразует исходное описание задачи в множество дочерних подзадач. Для конкретной задачи может существовать множество операторов преобразования. Применение каждого оператора порождает альтернативное множество подзадач, что определяет существование отношения ИЛИ на графе редукции.

В работе предлагается подход к представлению дерева подзадач в виде графа потока данных (ГПД).

Под ГПД [11, 21, 24, 28, 29, 31, 33, 34] понимается формализованное представление процесса обработки информации в целях решения задачи в виде графа, вершины которого есть функции обработки информации (правила обобщения), а дуги – потоки информации, проходящие через эти функции. Отметим основные черты, характеризующие ГПД:

1. ГПД наглядно и адекватно отображает процесс обработки информации при выводе решения.

2. ГПД отражает параллелизм и последовательность выполнения отдельных функций обработки информации.

3. Каждая функция иницируется, когда данные присутствуют на ее входе, и, следовательно, процесс обработки данных непрерывен, и ГПД отражает обработку данных в реальном масштабе времени.

Очевидно, что если вместо арифметических функций * и + использовать логиколингвистические функции над понятиями предметной области, можно построить ГПД решения интеллектуальной задачи.

Рассмотрим формальный аппарат, который может быть предложен для описания ГПД, отражающего процесс принятия решения. Очевидно, в качестве такого аппарата должна выступать алгоритмическая теория, как например нормальные алгоритмы Маркова, машина Поста, машина Тьюринга, λ -исчисление или теория комбинаторов. В [6] доказана эквивалентность этих систем.

Представляется наиболее удачным использовать аппарат теории комбинаторов [1, 25, 26]. Поскольку теория комбинаторов является производной от теории λ -исчисления, следовательно, она обладает преимуществами, присущими λ -исчислению, а также специфическими уникальными свойствами. В [5] убедительно доказаны преимущества теории комбинаторов и описаны практические результаты. Покажем основные из них:

1. Теория комбинаторов является алгоритмической системой и позволяет формально описать все вычислимые функции.

2. Теория комбинаторов – бестиповая теория, все переменные, константы и функции трактуются как символы.

3. Все комбинаторы выразимы в минимальном базисном наборе комбинаторов.

В приведенной литературе дано подробное описание теории комбинаторов, но для дальнейшего изложения введем некоторые определения.

Пусть дан алфавит, состоящий из скобок (,), символа $\langle \rightarrow \rangle$ и букв латинского алфавита: A, B, \dots, a, b, \dots .

Тогда определим объекты по индукции: если a – буква, то a – элементарный объект; каждый элементарный объект есть объект; если a и b – объекты, то (ab) – объект.

Примем соглашение, что объекты ab , $a(bc)$ и $abcd$ являются сокращениями для объектов (ab) , $(a(bc))$, $((ab)c)d$.

В данной нотации под комбинатором будем понимать объекты, с каждым из которых связана характеристическая функция изменения комбинаций переменных, к которым этот комбинатор применяется. Отметим, что комбинатор есть объект, в который не входит ни одной переменной. То есть, если комбинатор X последовательно применить к ряду объектов a, \dots, a_i , то возникающий новый объект $Xa\dots a_i$ конвертируется (преобразуется) в объект b , являющийся новой комбинацией объектов a, \dots, a_i . Это записывается в виде

$$Xa\dots a_i \rightarrow b.$$

Базисным набором комбинаторов называется совокупность комбинаторов, через которую выводимы все другие комбинаторы, но которые не выводимы друг из друга. Таким набором является следующий: $\{K, S\}$, где комбинаторы имеют следующие определения:

$$Kab \rightarrow a \text{ (канцелятор);}$$

$$Sabc \rightarrow ac(bc) \text{ (коннектор).}$$

Кроме того, полезными являются следующие комбинаторы:

$$Ia \rightarrow a \text{ (тождество);}$$

$$Babc \rightarrow a(bc) \text{ (композитор);}$$

$$Cabc \rightarrow acb \text{ (пермутатор);}$$

$$Wab \rightarrow abb \text{ (дубликатор).}$$

После введенных определений продемонстрируем представления ГПД в многомерных комбинаторах B_k^n, I_n^i, K_n [21], имеющие следующие комбинаторные определения:

$$(C1) B_k^n, fg_1 \dots g_k x_1 \dots x_n = f(g_1(x_1, \dots, x_n) \dots g_k(x_1, \dots, x_n));$$

$$(C2) I_n^i, x_1 \dots x_i \dots x_n = x_i;$$

$$(C3) K_n, xy_1 \dots y_n = x.$$

В нотации этих комбинаторов каждую функцию обработки информации (3) можно записать в виде

$$y = B_k^n \ fI_1 \dots I_k x_1 \dots x_n, \quad (5)$$

а всю модель (4) в виде

$$y = B_k^n \ fY_1 \dots Y_k x_1 \dots x_n, \quad (6)$$

где f – функция отображения корневой вершины; Y_1, \dots, Y_k – комбинаторные выраже-

ния нижестоящих узлов дерева; X_1, \dots, X_n – исходные данные самого нижнего уровня; k – количество нижестоящих узлов; n – количество исходных данных.

Таким образом, дерево подзадач записывается в виде одной комбинаторной строки, представление которой в информационной системе не вызывает особых сложностей. Редукция комбинаторного выражения осуществляется согласно правилам С1-С3, что требует для вывода решения использования всего трех процедур вывода, а также процедур, реализующих функции отображения. Достоинством данного подхода является компактность хранимых определений и возможность проведения параллельной редукции. Интересные пути реализации машин потоков данных и других перспективных архитектур изложены в [5, 21, 31].

Покажем связь между многомерными комбинаторами и введенным ранее базисным набором комбинаторов $\{K, S\}$ [25]:

$$I = I_1^1;$$

$$K = K_1;$$

$$B = B_1^1;$$

$$K_n = BKK_{n-1};$$

$$I_n^m = K_{m-1}I_{n-m+1}^1 = K_{m-1}K_{n-m};$$

$$B_n^m = F_m^{(n)} = BF_m F_m^{(n-1)},$$

где

$$I_n^1 = K_{n-1};$$

$$I_n^n = K_{n-1}I;$$

$$K_0 = I; \ F_m = B(BS_{m-1})B;$$

$$S_m = B(BS_{m-1})S.$$

Для представления знаний о принятии врачебных решений в БЗ ИСППВР необходимо разработать такую модель представления знаний (МПЗ), которая позволяет формально описать граф редукции задачи (дерево подзадач), записанное в комбинаторной нотации. Причем необходимо предусмотреть два основных условия:

1. МПЗ должна поддерживать построение сетевых структур для представления графовых моделей знаний.

2. МПЗ знаний должна обеспечивать вывод значения при редукции каждой из вершин графа.

В литературе широко освещены такие основные МПЗ, как фреймовые, продукционные, логические, их модификации и семантические сети [15, 22]. Очевидно,

требуемая модель должна совмещать свойства фреймовых моделей или семантических сетей для представления древовидных структур и продукционных или логических моделей для обеспечения вывода значений вершин дерева. В работе предлагается подход, объединяющий в себе свойства фреймовых и продукционных моделей, ориентированный на представление знаний о процессах принятия врачебных решений [10, 13, 17, 19, 23, 27, 30, 32].

Для представления конструкций вида (3) в виде, удобном для реализации в базе знаний, предлагается использовать формальную конструкцию – «аппликативный фрейм» (АФ), который можно задать в соответствии с традиционной нотацией в виде

$$N := (G, A_1, \dots, A_M, F, E), \quad (7)$$

где N – имя АФ; G – целевой атрибут АФ; A_1, \dots, A_L – аргументы АФ; F – функция отображения АФ (отображающая значения аргументов фрейма в значения целевого атрибута), т.е. АФ обеспечивает аппликацию функции отображения к аргументам фрейма; E – функция объяснения вывода решения; L – количество аргументов АФ.

Связь между конструкциями (3) и (7) очевидна. Целевой атрибут АФ соответствует значению функции обработки информации, аргументы АФ соответствуют ее аргументам, а функция отображения АФ соответствует ее описанию.

Аппликативной модели предметной области соответствует дерево АФ, вершины которого являются именами АФ, а ветви – аргументами фреймов и (или) целевыми атрибутами. Целевой атрибут и аргументы определяются в области своих допустимых значений. Функция отображения задается списком операторов отображения, каждый из которых состоит из двух основных частей: списка условий и действия. Условие описывается в виде предикатов на подмножествах аргументов АФ, а действие описывается в виде различных выполнимых операций над аргументами, операций обращения к подпрограммам и к базам данных.

Описанную МПЗ необходимо расширить средствами ведения диалога, метасредствами управления редукцией и средствами объяснения решений и реализовать в комплексе инструментальных средств построения интеллектуальных систем поддержки принятия решений.

Реализация модели представления знаний

Вывод решения для аппликативно-фреймовой МПЗ должен осуществляться на двух уровнях:

1. Редукция всего дерева АФ.
2. Редукция одного АФ.

Редукция дерева АФ осуществляется путем последовательной редукции его вершин. Редукция осуществляется рекурсивной процедурой RED, заданной формальными правилами следующего вида:

$$(R1) \text{ RED}[N_i] \rightarrow \text{RED}[(G_{Ni}, A_{1i}, \dots, A_{Mi}, F_i)].$$

$$(R2) \text{ RED}[(G_i, A_{1i}, \dots, A_{Mi}, F_i)] \rightarrow \text{EVAL}[F_i, \text{RED}(A_{1i}), \dots, \text{RED}(A_{Mi})].$$

$$(R3) \text{ RED}[A_i] \rightarrow \text{RED}[N_i], A_i = G_{Ni}$$

$$(R4) \text{ RED}[A_i] \rightarrow \#NR, A_i \#(G_{1i}, \dots, G_{Ni}),$$

где N_i – имя i -АФ; G_i – имя целевого атрибута i -АФ; A_{mi} – имя m -аргумента i -АФ; F_i – имя функции отображения i -АФ; $\#NR$ – специальное значение терминальной вершины; $i = 1, \dots, I$; $m_i = 1, \dots, M_i$; I – количество АФ, выделенных в предметной области; M_i – количество аргументов у i -АФ.

Правила (R1)–(R4) выполняют следующие действия:

(R1) – правило расширения; выполняется, если аргументом процедуры является имя АФ, и осуществляет поиск полной нотации АФ;

(R2) – правило редукции вершины дерева АФ, выполняется, если аргументом процедуры является конструкция полной нотации АФ, и осуществляет рекурсивное применение процедуры к аргументам АФ, с последующим применением функции EVAL, вычисляющей значение процедуры;

(R3) – правило перехода к следующему узлу дерева АФ; выполняется, если аргументом функции является имя аргумента АФ, при условии, что найден АФ, для которого этот аргумент является целевым атрибутом, и осуществляет рекурсивное применение процедуры к имени АФ, содержащему этот целевой атрибут;

(R4) – правило определения терминальной вершины, выполняется, если аргументом процедуры является имя аргумента АФ, причем этот аргумент не является целевым атрибутом никакого другого АФ, и присваивает процедуре специальное значение $\#NR$ (no references); в этом случае процесс редукции заканчивается.

Значением процедуры RED является либо значение процедуры EVAL, либо специальное значение $\#NR$.

Алгоритм вычисления значения процедуры редукции EVAL формально задается следующими правилами:

$$(E1) \text{ EVAL } [F_i, VG_{i1}, \dots, VG_{Mi}] \rightarrow \text{APPLAY}[F_i, VG_{i1}, \dots, VG_{Mi}].$$

$$(E2) \text{ EVAL}[F_i, VG_{i1}, \dots, VG_{im-1}, \#NR, VG_{im+2}, \dots, VG_{iM}] \rightarrow \text{APPLAY}[F_i, VG_{i1}, \dots, VG_{im-1}, A_m, VG_{im+2}, \dots, VG_{iM}],$$

где VG_i – значение целевого атрибута i -АФ, являющегося m -аргументом i -АФ; A_m – имя m -аргумента i -АФ; $\#NR$ – специальное значение терминальной вершины.

Правило (E1) определяет состав аргументов и осуществляет инициализацию процедуры APPLAY, если среди аргументов процедуры не обнаружено терминальных символов $\#NR$. Правило (E2) выполняется в противном случае и вместо терминального символа передает процедуре APPLAY значение соответствующего аргумента АФ.

Процедура APPLAY является традиционной процедурой интерпретаторов функциональных языков программирования и применяет функцию, являющуюся ее первым аргументом к последующим своим аргументам. Реализация функции APPLAY, которая и должна проводить непосредственное вычисление значения фрейма, представляет наибольшую сложность. Эта процедура фактически выполняет логический вывод целевого значения атрибута. Сложность этой процедуры определяется сложностью задания функции отображения.

Существует несколько способов задания механизма логического вывода, к которым можно отнести метод резолюций [19] и модифицированный метод резолюций для языка Пролог [2], обратный метод Маслова [12], метод секвенц-резолюций [9] и другие методы.

В экспертных системах большее распространение получили методы вывода решений на правилах [15]. В силу большей универсальности продукционных моделей функцию отображения наиболее удобно задать в виде совокупности правил продукций, где в левой части заданы условия на аргументах фрейма, а в правой элементарная функция вычисления целевого атрибута. В этой связи предложенная модель может быть классифицирована как фреймово-продукционная. Процедура APPLAY должна обеспечить выбор необходимого правила в соответствии с заложенным алгоритмом. Известны следующие основные алгоритмы [15]:

- выбор первого применимого правила;
- выбор по заранее заданному приоритету;
- выбор по вероятности;
- выбор наиболее применимого правила;

– выбор последнего примененного правила и т.д.

Редукция целевого АФ осуществляется формальной рекурсивной процедурой RED, заданной правилами R1–R4, значение которой есть значение целевого атрибута этого АФ. Начало редукции корневого АФ задается правилом R1.

Работа процедуры редукции заключается в определении значений аргументов АФ и применении его первого подходящего определяющего выражения из описания функции отображения этого АФ к значениям аргументов. Рекурсивность процедуры редукции позволяет говорить о редукции всего дерева АФ. Поэтому процесс вывода решения можно отождествить с процессом редукции дерева АФ.

В целях придания МПЗ свойств И-ИЛИ графа редукции в описание АФ необходимо ввести метаправила, которые инициализируются при начале редукции АФ. Каждое такое метаправило должно осуществлять альтернативный проход по дереву АФ. Этого можно достичь путем динамической модификации списка аргументов АФ: либо добавляя в него новые члены, либо исключая существующие.

Заключение

Предложенная МПЗ может лечь в основу построения системы знаний о выводе врачебного или управляющего решения. Реализация такой МПЗ может быть выполнена в различных системах программирования, хранение формализованных конструкций может быть осуществлено в различных системах управления базами данных.

Список литературы

1. Барендрегт Х. Лямбда-исчисление. Его синтаксис и семантика. – М.: Мир, 1985.
2. Борщев В.Б. Пролог – основные идеи и конструкции // Прикладная информатика. – Вып. 2 (11). – М.: Финансы и статистика, 1986.
3. Бураковский В.И., Бокерия Л.А., Газизова Д.Ш., Лещук В.А. и др. Компьютерная технология интенсивного лечения: контроль, анализ, диагностика, лечение, обучение. – М.: НИЦ ССХ РАМН, 1995.
4. Вагин В.Н. Дедукция и обобщение в системах принятия решений. – М.: Наука, 1988.

5. Вольфенгаген В.Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. – М.: JurInfoR Ltd, АО «Центр ЮрИнфоР», 2004 – 789 с.
6. Вольфенгаген В.Э., Яцук В.Я. Аппликативные вычислительные системы и концептуальный метод проектирования систем знаний. – Л.: МО СССР, 1987.
7. Ефимов Е.И. Решатели интеллектуальных задач. – М.: Наука, 1982.
8. Карпов О.Э., Клименко Г.С., Лебедев Г.С. Применение интеллектуальных систем в здравоохранении // Современные наукоемкие технологии. – 2016. – № 7. – С. 38–43.
9. Кирилов С.Г., Панков Ю.В. Комплекс программных средств ФЛОРИС для разработки экспертных систем // Интеллектуальное программное обеспечение ЭВМ: Тезисы докладов Всесоюзного научно-практического семинара. Ч. I. – Ростов-на-Дону – Терскол, 1990. – С. 50, 51.
10. Кобринский Б.А. Особенности медицинских интеллектуальных систем // Информационно-измерительные и управляющие системы. – 2013. – Т. 11. – № 5. – С. 58–64.
11. Лебедев Г.С., Мажирин И.В., Тронин Ю.Н., Яцук В.Я. Об одном подходе к реализации компонентов машины знаний средствами аппликативной компьютерной логики // Всесоюзная конференция по искусственному интеллекту: тезисы докладов. Т. 3. – Переславль-Залесский, 1988. – С. 314–317.
12. Маслов С.Ю. Обратный метод установления выводимости для логических исчислений // Труды математического института АН СССР. – 1968. – Т. 98. – С. 26, 27.
13. Назаренко Г.И., Осипов Г.С., Назаренко А.Г., Молдченков А.И. Интеллектуальные системы в клинической медицине. Синтез плана лечения на основе прецедентов // Информационные технологии и вычислительные системы. – 2010. – Т. 1. – С. 24–35.
14. Обработка нечеткой информации в системах принятия решений // А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьева и др. – М.: Радио и связь, 1989.
15. Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. – М.: Наука, 1987.
16. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. – М.: Радио и связь, 1989.
17. Постников В.М., Черненький В.М. Методы принятия решений в системах организационного управления: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 205 с.
18. Симанков В.С. Халафян А.А. Системный подход к разработке медицинских систем поддержки принятия решений // Известия высших учебных заведений. Северокавказский регион. Технические науки. – 2010. – № 1. – С. 29–36.
19. Робинсон Д. Машинно-ориентированная логика, основанная на методе резолюции // Киберн. сб. Нов. сер. № 7. – М.: Мир, 1970.
20. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. – М.: Наука, 1983.
21. Abdali S.K. An abstraction algorithm for combinatory logic // J. Symbol. Logic 41. – 1976. – P. 222–224.
22. gatonovic-Kustrin S.A., Beresford R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research // Journal of Pharmaceutical and Biomedical Analysis. – 2000. – № 22. – P. 717–727.
23. Barnett G.O., Cimino J.J., Hupp J.A., Hoffer E.P. DXplain – an evolving diagnostic decision-support system // JAMA. – 1987. – № 258. – P. 67–74.
24. Carkci M. Dataflow and Reactive Programming Systems: A Practical Guide. – CreateSpace Independent Publishing Platform, 2014. – 570 p. – ISBN 9781497422445.
25. Church R. The calculi of Lambda-conversion. – Princeton, 1941.
26. Curry H.B., Feys R., Craig W. Combinatory logic. Vol. 1. – North-Holland, Amsterdam, 1958.
27. Eta S. Berner, Ed.D. Clinical Decision Support Systems: State of the Art. // AHRQ Publication № 09-0069-EF June 2009. https://healthit.ahrq.gov/sites/default/files/docs/page/09-0069-EF_1.pdf.
28. Kent A. Dataflow languages // Encyclopedia of Library and Information Science: Volume 66 – Supplement 29 – Automated System for the Generation of Document Indexes to Volume Visualization. – Taylor & Francis, 2000. – P. 101. – 500 p. – ISBN 9780824720667.
29. Maurer P.M., Oldehoeft A.E. The use of combinators in translating a purely functional language to low-level data flow graphs // Comp. Lang. – 1983. – Vol. 8, № 1. – P. 27–45.
30. Osheroff J.A., Teich J.M., Middleton B.F. et al. A roadmap for national action on clinical decision support // J Am Med Inform Assoc. – 2007 Mar-Apr. – № 14(2). – P. 141–145. doi: 10.1197/jamia.M2334 Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2213467/>
31. Sharp J.A. Data Flow Computing: Theory and Practice. – Intellect, Limited, 1992. – 566 p. – ISBN 9780893919214.
32. Shortliffe E.H. Computer-Based Medical Consultations: MYCIN. Elsevier // North Holland. – New York NY, 1976.
33. Van-Roy P. and Haridi, S. Concepts, Techniques, and Models of Computer Programming. – Prentice-Hall, 2004. – 900 p. – ISBN 9780262220699.
34. Wesley M., Johnston J.R. Paul Hanna, Richard J. Millar. Advances in Dataflow Programming Languages. ACM Computing Surveys. – March 2004. – Vol. 36, № 1. – P. 1–34.

References

1. Barendregt H. Lambda-ischislenie. Ego sintaksis i semantika. M.: Mir, 1985.
2. Borshev V.B. Prolog osnovnye idei i konstrukcii // Prikladnaja informatika, Vypusk 2 (11). M.: Finansy i statistika, 1986.
3. Burakovskij V.I., Bokerija L.A., Gazizova D.Sh., Lishuk V.A. i dr. Kompjuternaja tehnologija intensivnogo lechenija: kontrol, analiz, diagnostika, lechenie, obuchenie. M.: NC SSH RAMN, 1995.
4. Vagin V.N. Dedukcija i obobshhenie v sistemah prijatija reshenij. M.: Nauka, 1988.
5. Volfengagen V.E. Metody i sredstva vychislenij s obektami. Applikativnye vychislitelnye sistemy. M.: JurInfoR Ltd, AO «Centr JurInfoR», 2004. 789 p.
6. Volfengagen V.E., Yacuk V.Ya. Applikativnye vychislitelnye sistemy i konceptualnyj metod proektirovanija sistem znaniy. L.: MO SSSR, 1987.
7. Efimov E.I. Reshateli intellektualnyh zadach. M.: Nauka, 1982.
8. Karpov O.E., Klimenko G.S., Lebedev G.S. Primenenie intellektualnyh sistem v zdavoohranenii // Sovremennye naukoemkie tehnologii. 2016. no. 7 (podpisano v pechat). pp. 38–43.
9. Kirilov S.G., Pankov Yu.V. Kompleks programmyh sredstv FLORIS dlja razrabotki jekspertnyh sistem // Intellektualnoe programmnoe obespechenie JeVM: Tezisy dokladov Vsesojuznogo nauchno-prakticheskogo seminar. Chast I. Rostov-na-Donu Terskol, 1990, pp. 50, 51.
10. Kobrinskij B.A. Osobennosti medicinskih intellektualnyh sistem // Informacionno-izmeritelnye i upravljajushhie sistemy. 2013. T. 11. no. 5. pp. 58–64.
11. Lebedev G.S., Mazhirin I.V., Tronin Yu.N., Yacuk V.Ya. Ob odnom podhode k realizacii komponentov mashiny znaniy sredstvami applikativnoj kompjuternoj logiki // Vsesojuznaja konferencija po iskusstvennomu intellektu. Tezisy dokladov. Tom 3. Pereslavl-Zaleskij, 1988, pp. 314–317.
12. Maslov S.Yu. Obratnyj metod ustanovlenija vyvodimosti dlja logicheskikh ischislenij // Trudy matematicheskogo instituta AN SSSR, T. 98, 1968, pp. 26, 27.

13. Nazarenko G.I., Osipov G.S., Nazarenko A.G., Molodchenkov A.I. *Intellektualnye sistemy v klinicheskoj medicine. Sintez plana lechenija na osnove precedentov // Informacionnye tehnologii i vychislitelnye sistemy. Tom.1, 2010, pp. 24–35.*
14. *Obработка nechetkoj informacii v sistemah prinjatija reshenij. // A.N. Borisov, A.V. Alekseev, G.V. Merkureva i dr. M.: Radio i svjaz, 1989.*
15. Popov E.V. *Jekspertnye sistemy: Reshenie neformalizovannyh zadach v dialoge s JeVM. M.: Nauka, 1987.*
16. Pospelov D.A. *Modelirovanie rassuzhdenij. Opyt analiza myslitelnyh aktov. M.: Radio i svjaz, 1989.*
17. Postnikov V.M., Chernenkij V.M. *Metody prinjatija reshenij v sistemah organizacionnogo upravlenija: ucheb. posobie. M.: Izd-vo MGTU im. N.E. Baumana, 2014. 205 p.*
18. Simankov V.S., Halafjan A.A. *Sistemnyj podhod k razrabotke medicinskih sistem podderzhki prinjatija reshenij // Izvestija vysshih uchebnyh zavedenij. Severokavkazskij region. Tehniceskie nauki. 2010. no. 1. pp. 29–36.*
19. Robinson D. *Mashinno-orientirovannaja logika, osnovannaja na metode rezoljucii // Kibern. sb. Nov. ser. no. 7. M.: Mir, 1970.*
20. Chen Ch., Li R. *Matematiceskaja logika i avtomaticheskoe dokazatelstvo teorem. M.: Nauka, 1983.*
21. Abdali S.K. *An abstraction algorithm for combinatory logic // J. Symbol. Logic 41, 1976, pp. 222–224.*
22. Agatonovic-Kustrin S., Beresford R. *Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. // Journal of Pharmaceutical and Biomedical Analysis. 22 (2000) 717–727.*
23. Barnett G.O., Cimino J.J., Hupp J.A., Hoffer E.P. *DX-plain an evolving diagnostic decision-support system. JAMA. 1987; 258: 67–74.*
24. Carkci M. *Dataflow and Reactive Programming Systems: A Practical Guide. CreateSpace Independent Publishing Platform, 2014. 570 p. ISBN 9781497422445.*
25. Church R. *The calculi of Lambda-conversion. Princeton, 1941.*
26. Curry H.B., Feys R., Craig W. *Combinatory logic. Vol.1. North-Holland, Amsterdam, 1958.*
27. Eta S. Berner, Ed.D. *Clinical Decision Support Systems: State of the Art. // AHRQ Publication no. 09-0069-EF June 2009. https://healthit.ahrq.gov/sites/default/files/docs/page/09-0069-EF_1.pdf*
28. Kent A. *Dataflow languages // Encyclopedia of Library and Information Science: Volume 66 Supplement 29 Automated System for the Generation of Document Indexes to Volume Visualization. Taylor & Francis, 2000. pp. 101. 500 p. ISBN 9780824720667.*
29. Maurer P.M., Oldehoeft A.E. *The use of combinators in translating a purely functional language to low-level data flow graphs // Comp.Lang. Vol. 8, no 1, 1983, pp. 27–45.*
30. Osheroff J.A., Teich J.M., Middleton B.F. et al. *A roadmap for national action on clinical decision support. J Am Med Inform Assoc. 2007 Mar-Apr; 14(2): 141–145. doi: 10.1197/jamia.M2334 Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2213467>.*
31. Sharp J.A. *Data Flow Computing: Theory and Practice. Intellect, Limited, 1992. 566 p. ISBN 9780893919214.*
32. Shortliffe E.H. *Computer-Based Medical Consultations: MYCIN. Elsevier/North Holland, New York NY, 1976.*
33. Van-Roy P., Haridi S. *Concepts, Techniques, and Models of Computer Programming. Prentice-Hall, 2004. 900 p. ISBN 9780262220699.*
34. Wesley M. Johnston, J.R. Paul Hanna, Richard J. Millar. *Advances in Dataflow Programming Languages. ACM Computing Surveys, Vol. 36, no. 1, March 2004, pp. 1–34.*