

Розроблено метод формального представлення базових сутностей парадигми граничних узагальнень: елементарних тестів, орграфів їх доменів і системопатернів на мові алгебри предикатів. Формалізовано правила перерахунку між доменами різного рівня загальності, з використанням математичного апарату алгебри предикатів. Отримані моделі реалізовано за допомогою алгебропредикатних структур, які представлені у вигляді асоціативно-логічних перетворювачів

Ключові слова: парадигма граничних узагальнень, орграфи доменів, системопатерни, алгебра предикатів, алгебропредикатні структури

Разработан метод формального представления базовых сущностей парадигмы предельных обобщений: элементарных тестов, орграфов их доменов и системопаттернов на языке алгебры предикатов. Формализованы правила пересчета между доменами разного уровня общности, с использованием математического аппарата алгебры предикатов. Полученные модели реализованы с помощью алгебропредикатных структур, которые представлены в виде ассоциативно-логических преобразователей

Ключевые слова: парадигма предельных обобщений, орграфы доменов, системопаттерны, алгебра предикатов, алгебропредикатные структуры

ПРЕДСТАВЛЕНИЕ БАЗОВЫХ СУЩНОСТЕЙ ПАРАДИГМЫ ПРЕДЕЛЬНЫХ ОБОБЩЕНИЙ С ПОМОЩЬЮ АЛГЕБРОПРЕДИКАТНЫХ СТРУКТУР

В. И. Булкин

Кандидат технических наук, доцент
Кафедра прикладной математики
и информационных технологий

Макеевский экономико-гуманитарный институт
ул. Островского, 16, г. Макеевка, Украина, 86157

E-mail: bulkin01@mail.ru

Ю. А. Прокопчук

Кандидат физико-математических наук, доцент
Отдел системного анализа и проблем управления
Институт технической механики НАНУ и ГКАУ

ул. Лешко-Попеля, 15,

г. Днепропетровск, Украина, 49005

E-mail: itk3@ukr.net

1. Введение

На современном этапе развития вычислительной техники возникли проблемы, связанные с использованием последовательных алгоритмов обработки данных и исключительно двоичного кодирования. Эти проблемы требуют решения задач создания вычислительных средств с использованием новых архитектур и новых принципов вычислений, использования пространственного и временного параллелизма, создания параллельных машин баз знаний и логического вывода, использования недвоичных методов кодирования. Попытки внедрить принципы параллелизма на основе неймановских процессоров не привели к значительному увеличению быстродействия и созданию высокопроизводительных систем искусственного интеллекта [1].

Одним из новых подходов к организации вычислительных сред является построение графодинамических ассоциативных машин на основе парадигмы предельных обобщений (ППО) [2, 3] и алгебропредикатных структур (АПС) [4]. ППО – это основанный на когнитивном подходе комплекс методов критической самоорганизации и рефлексии знаний, выявление

предпараметров и параметров порядка эволюции сложных систем, процессов и ситуаций, а также использования ключевых параметров для управления эволюцией [2]. Предлагаемая в рамках ППО логика реализации базируется на автоматическом переводе предельных когнитивных структур в функциональные системы, например в среды радикалов. Радикалы (системопаттерны, семантические указатели) реализуют автоматизмы среды. АПС показывает один из возможных путей «материализации» радикала на различных масштабных уровнях.

Таким образом, ППО определяет логику формирования предельных структур знаний и функциональных систем на основе перцептивного опыта, а АПС описывает один из вариантов «материализации» как первичных (генетических) структур, так и функциональных систем. Подобный симбиоз позволяет замкнуть весь цикл проектирования вычислительных сред определенного класса на указанных выше принципах. В данной работе рассматриваются в основном вопросы «материализации» первичных (базовых, генетических) структур графодинамических ассоциативных машин класса «ППО-АПС», таких как орграфы доменов и системопаттерны.

2. Анализ литературных источников

Процесс формирования автономной или гибридной, адаптивной, рефлексивной, управляющей вычислительной среды условно можно разделить на два этапа. Первый этап (назовем его – когнитивный) заключается в накоплении опыта и самоорганизации структур знаний [2, 5, 7], а второй этап реализует логику исполнения, т.е. закрепляет выявленные структуры знаний в каких-либо материальных структурах, включая «генетический» материал [8 – 10]. Наиболее сложной как в методологическом плане, так и для практической реализации (применительно к моделям знаний) является концепция самоорганизующейся критичности (self-organized criticality) [7]. ППО в полном объеме реализует данную концепцию.

Вычислительная нейронная среда SPAUN (модель мозга) на основе семантических указателей предложена в [8]. Фрактальные аспекты биологических вычислений рассматриваются в [10]. Функциональная система в рамках ППО является аттрактором мультифрактального эволюционного вычислительного процесса. Альтернативой гомеостазису функциональной системы (среды радикалов) в рамках ППО является гомеостатическая нейросеть [11].

Графодинамические системы с сетевым управлением в математически однородном поле компьютерной информации рассматриваются в [6], в мультиагентном варианте в [12]. Однако ППО позволяет реализовать действительно математически однородное поле компьютерной информации в гибридной или агентной среде в полном соответствии с концепцией трансдисциплинарности.

Одним из перспективных способов формализации функционирования произвольной интеллектуальной системы является использование семантических сетей, с помощью которых могут быть представлены как знания системы (база знаний системы), так и алгоритмы интеллектуальной обработки информации, используемые данной системой. Семантическая сеть по своей сути является некоторой графовой структурой, элементы которой наделяются дополнительным смыслом [13]. На практике при создании прикладной интеллектуальной системы, как правило, используется программная модель семантической сети, реализованная на универсальной вычислительной системе (персональном компьютере), что, очевидно, существенно снижает общую стоимость системы и нередко является подходящим решением [13]. Однако использование универсальных процессоров имеет ряд ограничений по масштабируемости и не подходит для реализации сложных вычислительных систем и сред. Для таких систем, предъявляющих особо высокие требования к производительности либо максимально возможному объему знаний, могут быть использованы спецпроцессоры (суперкомпьютеры) [14], однако это зачастую невозможно из-за больших размеров и большой стоимости таких систем (например, для реализации бортовых советующих систем в антропоцентрических объектах). Другим хорошо известным решением является использование универсальных параллельных систем, таких как GPU [15], кластеров, облачных вычислений [16], что позволяет достичь гораздо большей производительности по сравнению с обычными универсальными процессорами при сохранении

относительно низкой стоимости всей системы в целом. Одним из недостатков такого рода систем является тот факт, что изначально они создавались для решения совершенно другого класса задач и в случае интеллектуальных систем они не всегда могут быть использованы с максимальной эффективностью. Для удовлетворения противоречивым требованиям предлагается создавать проблемно-ориентированные процессоры [17]. Для них ключевой проблемой является нахождение баланса между высокой специализацией и шириной охвата потенциальных задач.

Предлагаемый авторами настоящей работы подход, также заключается в разработке архитектуры проблемно-ориентированного процессора, предназначенного для решения задач семантической обработки на основе ППО. ППО гарантирует определенную унификацию, т.е. ориентирует процессор на решение достаточно широкого класса схожих задач. Другими словами, ППО играет роль верхнего уровня абстракции над системой команд проблемно-ориентированного процессора, который жестко ориентирован на конкретную предметную область или решаемую задачу. Реализация такого процессора обеспечит достижение лучшей эффективности в некоторых технических приложениях по сравнению с универсальными решениями (универсальными процессорами и универсальными параллельными процессорами) при сохранении низкой общей стоимости (заданной рентабельности). Фактически, процессор архитектуры «ППО-АПС» может рассматриваться как сложная масштабируемая динамическая ассоциативная память.

3. Цель и задачи исследования

В работе [2] вводятся понятия элементарного теста, ориентированного графа (орграфа) набросков и динамического системопаттерна (или просто системопаттерна). К орграфам набросков относятся и орграфы доменов элементарного теста. С помощью множества элементарных тестов описываются произвольные ситуации действительности, а с помощью системопаттернов в ППО описываются произвольные преобразования. Целью исследования является демонстрация того, что перечисленные выше базовые сущности ППО можно формально представить на языке алгебры предикатов [18]. Алгебра предикатов – это первый уровень формального языка для разработки систем искусственного интеллекта на основе АПС. На втором уровне находится алгебра предикатных операций. На языке алгебры предикатов выражаются отношения, а на языке алгебры предикатных операций – действия над отношениями.

Задачами исследования являются:

- создание моделей орграфа доменов и системопаттерна на языке алгебры предикатов;
- построение алгебропредикатных структур в виде ассоциативно-логических преобразователей.

4. Модели базовых сущностей парадигмы предельных обобщений на языке алгебры предикатов

Элементарный тест обозначается греческой буквой τ и записывается в виде: «тест = значение» или

«тест? значение» что равносильно записи «параметр = значение». Конкретный результат теста τ обозначают через τ . Результаты тестов выбираются из разных доменов. Домен – это множество значений теста со связями наследования. Для записи того факта, что результаты τ_i теста τ выбираются из домена T ($T = \{\tau_1, \tau_2, \dots, \tau_n\}$), используют нотацию τ/T . Примеры нотаций: a/A – теста, значения которого выбираются из домена A ; $\{b/V\}$ – множество тестов b_1, b_2, \dots, b_n , значения каждого из которых выбираются из доменов V_1, V_2, \dots, V_n соответственно; $\{c\}/C$ – множество однотипных тестов c_1, c_2, \dots, c_m значения каждого из которых выбираются из одного и того же домена C ; $\{a/A\}$ – множество значений группы тестов.

Пример группы тестов:

$\{b/V\} = \{\text{ФИО}/C_1; \text{Пол}/C_2; \text{Возраст}/N_1; \text{АДс}/N_2; \text{АДд}/N_3; \text{Жалобы}/C_3\}$, где «ФИО», «Пол», «Возраст», «АДс», «АДд», «Жалобы» – это имена тестов (АДс, АДд – артериальное давление систолическое и диастолическое), а $C_1, C_2, N_1, N_2, N_3, C_3$ – соответствующие домены этих тестов.

Пример значений группы тестов:

$\{b/V\} = \{\text{ФИО}/C_1? \text{Петров И.И.}, \text{Пол}/C_2? \text{М}, \text{Возраст}/N_1? \text{35}, \text{АДс}/N_2? \text{140}, \text{АДд}/N_3? \text{87}, \text{Жалобы}/C_3? \text{На головную боль, на боли в суставах}\}$. Здесь «Иванов И.И.», «М», «35», «140», «87», «На головную боль», «на боли в суставах» – это конкретные значения соответствующих тестов. Заметим, что тест «Жалобы» имеет два значения: «На головную боль», «на боли в суставах».

Формальная модель системопаттерна представляется в виде:

$$f/\mu: \{a/A\}, e/E \rightarrow \{b/V\}, \mu \in \{\mu\}f, \quad (1)$$

где $\{a/A\}$ – входные тесты; $\{b/V\}$ – выходные тесты; e/E – требуемая структурная энергия, ресурсы; μ – механизм реализации.

Отношения описывают свойства предметов и явлений и связи между ними. При этом отношения интерпретируются как мысли интеллекта, а действия над отношениями – как сам процесс мышления [18-5]. Отношения задаются над предметным пространством. Предметным пространством U^m называют декартову степень m универсума предметов U . Входные сигналы, предъявляемые испытуемому в экспериментах теории T , называют предметами теории T . Непустое множество U всевозможных предметов теории T называют универсумом предметов теории T ($U = \{a_1, a_2, \dots, a_m\}$). Предметные переменные x_1, x_2, \dots, x_m теории T содержательно интерпретируются как места предметного пространства U^m . Здесь m – число предметных переменных. Непустое множество V всевозможных переменных теории T называют универсумом переменных теории T ($V = \{x_1, x_2, \dots, x_m\}$). Если справедливо утверждение $x_1 = a_1, x_2 = a_2, \dots, x_m = a_m$, то говорят о наличии набора (a_1, a_2, \dots, a_m) предметов a_1, a_2, \dots, a_m , находящихся на местах x_1, x_2, \dots, x_m . Если предмет a находится на месте x , то говорят, что место x находится в состоянии a . Внешний (объективный) мир и внутренний (субъективный) мир человека устроены единообразно. Всюду можно обнаружить места, которые находятся в определенных состояниях. С течением времени состояния этих мест могут изменяться. Каждое место пред-

метного пространства в любой момент времени может находиться только в одном состоянии.

Любое подмножество P предметного пространства U^m называют m -местным отношением, заданным над этим пространством. Для записи отношений используют множества наборов предметов, графы, таблицы, графики. Необходимо иметь математические средства для записи отношений в виде формул. С помощью формул в математике возможно представление только функций. Отношения – это более общее понятие, чем функции. Поэтому непосредственно представить их в формульном виде не представляется возможным. Однако, если произвольному отношению P_A поставить в соответствие высказывание A , которое принимает значение истина, если оно соответствует отношению P_A , и – ложь, если не соответствует, то мы приходим к понятию предиката. Предикатом, заданным на декартовом произведении $A_1 \times A_2 \times \dots \times A_m$ множеств A_1, A_2, \dots, A_m называется любая функция $P(x_1, x_2, \dots, x_m) = \xi$, отображающая декартово произведение $A_1 \times A_2 \times \dots \times A_m$ множеств A_1, A_2, \dots, A_m в множество $\Sigma = \{1, 0\}$. Переменная $\xi \in \{1, 0\}$, которая является значением предиката P , называется булевой переменной. Предикат $P(x_1, x_2, \dots, x_m)$ в отличие от соответствующего ему отношения P , является функцией, поэтому появляется возможность выразить его в виде формулы. Для разработки способа записи предикатов в виде формул будем использовать базисные предикаты и базисные операции, связывающие их между собой. В качестве базисных предикатов будем использовать предикаты 0 и 1, а также специальные предикаты узнавания предмета a по переменной x_i ($i = 1, m, a \in A_i$), которые записываются в виде x_i^a . Предикат x_i^a «узнает» произвольно выбранный из множества A_i предмет x_i сравнивая его с предметом a . Если $x_i = a$, то предикат x_i^a сигнализирует об этом, принимая значение $x_i^a = a^a = 1$. В противном случае, если $x_i = b$ ($b \neq a$), то предикат принимает значение $x_i^a = b^a = 0$. Таким образом, получаем следующее определение базисного предиката:

$$x_i^a = \begin{cases} 1, & \text{если } x_i = a \\ 0, & \text{если } x_i \neq a \end{cases} \quad (2)$$

Символ a , который стоит в показателе записи x_i^a , называют характеристикой предиката узнавания предмета. В качестве базисных операций, связывающих между собой базисные предикаты, используют операции конъюнкции, дизъюнкции и отрицания предикатов. В результате получаем алгебру предикатов [18]. Для предикатов узнавания предметов справедливы следующие законы:

- закон отрицания $\overline{x_i^a} = \bigvee_{\substack{b \in A_i \\ b \neq a}} x_i^b$;
- закон ложности $x_i^a x_i^b = 0$, если $b \neq a$;
- закон истинности $\bigvee_{a \in A_i} x_i^a$.

Все эти законы справедливы для любых $i = \overline{1, m}$; $x_i \in A_i$; $a, b \in A_i$.

Любой элементарный тест формально может быть представлен в виде предиката узнавания предмета (1). Используя обозначения, принятые в работе [2],

формула (1) может быть представлена в следующем виде:

$$x_i^{\tau} = \begin{cases} 1, & \text{если } x_i = \tau \\ 0, & \text{если } x_i \neq \tau \end{cases}, \quad (3)$$

где x_i – предметная переменная, соответствующая тесту τ_i , τ – конкретный результат теста τ_i .

На языке алгебры предикатов пример значений группы тестов {b/V}, приведенный выше запишем в виде следующего предикатного уравнения:

$$x_1^{\Pi} x_2^M x_3^{34} x_4^{145} x_5^{87} (x_6^B x_6^C). \quad (4)$$

Здесь предметная переменная x_1 соответствует тесту ФИО, символ «П» соответствует значению данного теста – «Петров И. И.». Аналогично предметные переменные x_2, x_3, x_4, x_5, x_6 соответствуют тестам «Пол», «Возраст», «АДс», «АДд», «Жалобы», а символы «М», «34», «145», «87», «Б», «С» соответствуют значениям каждого из этих тестов. Корнями уравнения (2) являются значения предметных переменных $x_1 = \Pi, x_2 = M, x_3 = 34, x_4 = 145, x_5 = 87, x_6 = B, x_6 = C$, которые совпадают со значениями соответствующих тестов. В свою очередь домены тестов также могут быть записаны на языке алгебры предикатов. Например, домену S_2 теста Пол, который представляет собой множество $S_2 = \{M, Ж\}$ соответствует предикатное уравнение

$$x_2^M \vee x_2^Ж = 1. \quad (5)$$

Очевидно, что множество корней уравнения (5) совпадает с множеством S_2 . Аналогично на языке алгебры предикатов могут быть представлены домены любых элементарных тестов.

Используя разные домены, можно управлять общностью (масштабом) результатов одного и того же теста. Между доменами разного уровня общности могут быть заданы однозначные правила пересчета. Правила пересчета из одного домена в другой позволяют построить ориентированный граф доменов $G(\tau) \equiv \{T \rightarrow T'\}_{\tau}$. Наиболее общие правила пересчета значений одних тестов в значения других тестов содержатся в базе знаний.

Примеры доменов теста $\tau = \text{Возраст}$:

$V1 = [0; 100]$;

$V2 = \{\text{юный, молодой, средних лет, пожилой, старческий}\}$;

$V3 = \{\text{молодой, средних лет, пожилой}\}$.

На языке алгебры предикатов домену $V1$ соответствует предикатное уравнение

$$x_1^0 \vee x_1^1 \vee x_1^2 \vee \dots \vee x_1^{100} = 1,$$

домену $V2$ соответствует предикатное уравнение

$$x_2^0 \vee x_2^M \vee x_2^{c1} \vee x_2^n \vee x_2^c = 1,$$

домену $V3$ соответствует предикатное уравнение

$$x_3^M \vee x_3^{c1} \vee x_3^n = 1.$$

Приведем примеры результатов тестов и уравнения алгебры предикатов, которые им соответствуют:

Возраст/ $V1$? 87 $\rightarrow x_1^{87} = 1, x_1 = 87$; Возраст/ $V2$? «пожилой» $\rightarrow x_2^n = 1, x_2 = \langle \text{п} \rangle$ – пожилой; Возраст/ $V3$? «пожилой» $\rightarrow x_3^n = 1, x_3 = \langle \text{п} \rangle$ – пожилой.

Отметим, что Возраст/ $V2$? «пожилой» \neq Возраст/ $V3$? «пожилой», а также $x_2^n \neq x_3^n$. Это говорит о том, что значения тестов Возраст/ $V2$ и Возраст/ $V3$ и, соответственно, значения предметных переменных x_2 и x_3 берутся из разных доменов. Зададим правила пересчета между доменами $V1, V2$ и $V3$ следующим образом:

$V1.[0; 14] \rightarrow V2.\{\text{юный}\}$; $V1.[15; 33] \rightarrow V2.\{\text{молодой}\}$; $V1.[34; 55] \rightarrow V2.\{\text{средних лет}\}$; $V1.[56; 70] \rightarrow V2.\{\text{пожилой}\}$; $V1.[71; 100] \rightarrow V2.\{\text{старческий}\}$; $V2.\{\text{юный, молодой}\} \rightarrow V3.\{\text{молодой}\}$; $V2.\{\text{средних лет}\} \rightarrow V3.\{\text{средних лет}\}$; $V2.\{\text{пожилой; старческий}\} \rightarrow V3.\{\text{пожилой}\}$.

Данным правилам будет соответствовать следующая система импликативных уравнений:

$$x_1^0 \vee x_1^1 \vee \dots \vee x_1^{14} \supset x_2^{\text{ю}}, \quad (6)$$

$$x_1^{15} \vee x_1^{16} \vee \dots \vee x_1^{33} \supset x_2^{\text{м}}, \quad (7)$$

$$x_1^{34} \vee x_1^{35} \vee \dots \vee x_1^{55} \supset x_2^{c1}, \quad (8)$$

$$x_1^{56} \vee x_1^{57} \vee \dots \vee x_1^{70} \supset x_2^n, \quad (9)$$

$$x_1^{71} \vee x_1^{72} \vee \dots \vee x_1^{100} \supset x_2^c, \quad (10)$$

$$x_2^{\text{ю}} \vee x_2^{\text{м}} \supset x_3^{\text{м}}; \quad (11)$$

$$x_2^{c1} \supset x_3^{c1}, \quad (12)$$

$$x_2^n \vee x_2^c \supset x_3^n. \quad (13)$$

Правила пересчета $V1 \rightarrow V2 \rightarrow V3$ задают орграф доменов для теста ‘Возраст’ $G_1(\text{Возраст})$. Используя орграф доменов можно автоматически вычислять значения теста для всех доменов по одному самому точному значению. Например: Возраст/ $V1$? 5 \rightarrow Возраст/ $V2$? «юный» \rightarrow Возраст/ $V3$? «молодой»

Возраст/ $V1$? 76 \rightarrow Возраст/ $V2$? «старческий» \rightarrow Возраст/ $V3$? «пожилой».

На языке алгебры предикатов эти вычисления осуществляются путем подстановки значений предметных переменных в соответствующие предикатные уравнения. Значение теста Возраст/ $V1$? 5 означает, что предметная переменная $x_1 = 5$. Подставляя это значение в уравнение (6), получаем:

$$5^0 \vee 5^1 \vee \dots \vee 5^{14} \supset x_2^{\text{ю}}; \quad 0 \vee 0 \vee \dots \vee 0 \supset x_2^{\text{ю}};$$

$$1 \supset x_2^{\text{ю}}; \quad x_2^{\text{ю}} = 1; \quad x_2 = \langle \text{ю} \rangle - \text{юный}.$$

Подставляя значение $x_2 = \langle \text{ю} \rangle$ – юный в уравнение (11), получаем:

$$\text{Ю}^{\text{ю}} \vee \text{Ю}^{\text{м}} \supset x_3^{\text{м}}; \quad 1 \vee 0 \supset x_3^{\text{м}}; \quad 1 \supset x_3^{\text{м}}; \quad x_3^{\text{м}} = 1; \quad x_3 = \langle \text{м} \rangle - \text{молодой}.$$

Аналогично вычисляем значение теста ‘Возраст’ для всех доменов, используя точное значение теста Возраст/ $V1$?76.

Подставляем значение предметной переменной $x_1 = 76$ в уравнение (10) и получаем

$$76^{71} \vee 76^{72} \vee \dots \vee 76^{76} \vee \dots \vee 76^{100} \supset x_2^c;$$

$$0 \vee 0 \vee \dots \vee 1 \vee \dots \vee 0 \supset x_2^c; \quad 1 \supset x_2^c;$$

$$x_2^c = 1; \quad x_2 = \langle \text{с} \rangle - \text{старческий}.$$

Подставляя значение $x_2 = \langle \text{с} \rangle$ – старческий в уравнение (13), получаем: $c^n \vee c^c \supset x_3^n; \quad 0 \vee 1 \supset x_3^n; \quad 1 \supset x_3^n; \quad x_3^n = 1; \quad x_3 = \langle \text{п} \rangle$ – пожилой.

Импликативные уравнения (6) – (13) задают правила пересчета значений доменов и формируют базу знаний, включающую в свой состав факты в виде множества значений тестов и продукционные правила пересчета значений тестов для различных доменов. Правила пересчета значений доменов В1, В2 и В3, представленные в виде системы импликативных уравнений (6) – (13), можно реализовать в виде некоторого алгоритма. На основе этого алгоритма может быть составлена программа, и результат решения задачи может быть получен программным путем. Однако программным путем можно реализовать только функциональные отношения. Отношения же произвольного типа можно реализовать в виде алгебропредикатных структур (АП-структур), которые, в свою очередь, можно представить в виде ассоциативно-логических преобразователей.

5. Алгебропредикатные структуры базовых сущностей

Представим уравнения (6) – (13), задающие правила пересчета значений доменов, в виде АП-структуры, показанной на рис. 1.

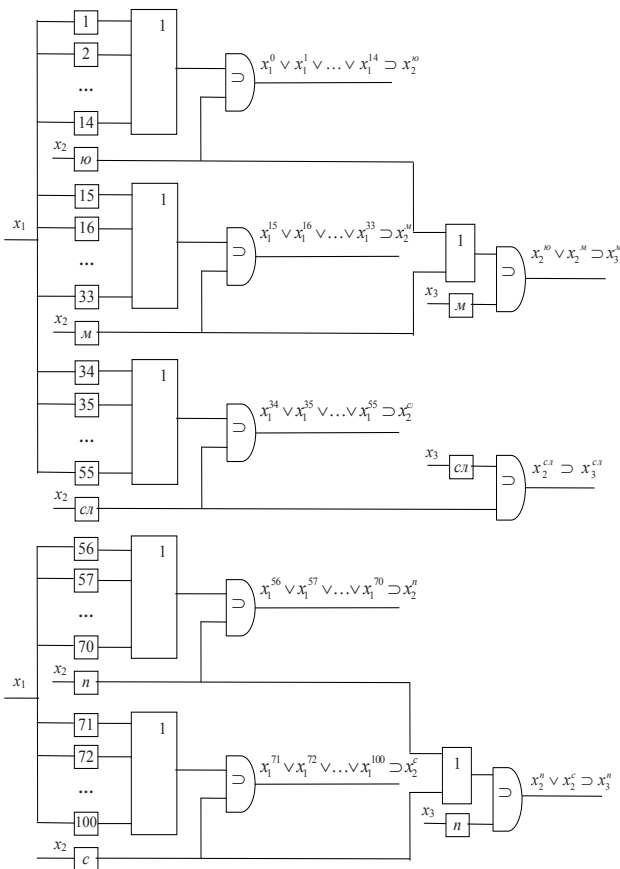


Рис. 1. АП-структура, задающая правила пересчета значений доменов

Рассмотрим в качестве примера работу АП-структуры, реализующей вычисление значения теста для всех доменов по одному самому точному значению $x_1 = 5$. Схема работает следующим образом. Если на ее вход подать сигнал $x_1 = 5$, то элемент узнавания предмета сформирует сигнал $x_1^5 = 1$ и на выходе сформиру-

ется сигнал $x_2^0 = 1$, а, следовательно, будет выработан сигнал $x_2 = \text{«ю»}$ – юный. Сигнал $x_2^0 = 1$ поступает на вход схемы разделения и на выходе будет сформирован сигнал $x_3^m = 1$, а, следовательно, будет выработан сигнал $x_3 = \text{«м»}$ – молодой. Аналогично можно вычислить значение теста ‘Возраст’ для всех доменов, используя точное значение теста Возраст/В1?76.

Следует отметить, что получение вывода с помощью АП-структуры осуществляется всего за один такт работы цепи, а алгоритмическим путем – за несколько шагов подстановки. Это говорит о преимуществе использования АП-структур перед алгоритмическими методами решения задач логического вывода. Это преимущество заключается, во-первых, в том, что резко уменьшается время решения задачи, а во-вторых, для решения задачи нет необходимости использовать сложное аппаратное и программное обеспечение универсальных компьютеров последовательного действия.

Любой объект, явление или процесс полностью описывается множеством элементарных тестов и системопаттернов на их основе, которые формируют информационную структуру. Описания структур могут содержать онтологические соглашения. К онтологическим соглашениям относятся все виды определений, включая формулы. Примеры онтологических соглашений клинической медицины:

«Заключение d имеет место тогда и только тогда, когда множество результатов тестов $\{\tau/T\}$ удовлетворяет фиксированному множеству предикатов $\{P\}$ », $d \Leftrightarrow \{P(\{\tau/T\}) = \text{True}\}$;

$d = \text{«Воспаление среднего уха»} \Leftrightarrow \{\text{Боль в ухе? Есть; Шум в ухе? Есть; Ухудшение слуха? Есть; Гиперемия и выпуклость барабанной перепонки? Есть}\}$. Данное соглашение можно представить в виде системопаттерна $f/\mu: \{\tau/T\} \rightarrow d/D$. Для того чтобы этот системопаттерн представить на языке алгебры предикатов, введем предметные переменные, соответствующие элементарным тестам, входящим в состав системопаттерна, и зададим области их определения.

Первому элементарному тесту соответствует предметная переменная z_1 – «Боль в ухе» и имеет значения «б» – боль в ухе есть, «н» – боли в ухе нет. Область определения этой переменной задается предикатным уравнением $z_1^b \vee z_1^n = 1$. Второму элементарному тесту поставим в соответствие переменную z_2 – «Шум в ухе», которая принимает значения «ш» – шум в ухе есть, «н» – шума в ухе нет. Уравнение, задающее область определения этой переменной, имеет вид: $z_2^ш \vee z_2^n = 1$. Предметная переменная z_3 – «Ухудшение слуха» соответствует третьему тесту и имеет значения «у» – ухудшение слуха есть, «н» – ухудшения слуха нет. Область определения этой переменной запишется в виде уравнения алгебры предикатов $z_3^у \vee z_3^n = 1$. Четвертому тесту соответствует переменная z_4 – «Гиперемия и выпуклость барабанной перепонки» и имеет значения «г» – гиперемия и выпуклость барабанной перепонки есть, «н» – гиперемии и выпуклости барабанной перепонки нет. Область определения этой переменной задается уравнением $z_4^г \vee z_4^n = 1$. Все выше перечисленные переменные являются бинарными также как и значения элементарных тестов, которым они соответствуют. Заключение d поставим в соответствие переменную u, которая может принимать значения «в» – воспаление

среднего уха есть и «н» – воспаления среднего уха нет. Область определения этой переменной задается уравнением $y^b \vee y^n = 1$. Теперь после введения предметных переменных и задания областей их определения можно формально представить системопаттерн на языке алгебры предикатов в виде следующего импликативного уравнения: $x_1^b x_2^u x_3^y x_4^z \supset y^b$.

Полученное импликативное уравнение представим в виде АП-структуры, показанной на рис. 2.

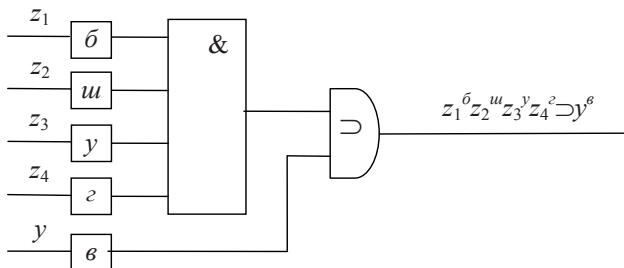


Рис. 2. АП-структура, задающая значения системопаттерна

Любую АП-структуру можно представить в виде двух блоков – блока узнавания предметов и блока логических операций. Блок узнавания предметов включает в свой состав множество элементов узнавания предметов, каждый из которых является аппаратной реализацией базисного предиката узнавания предмета (1).

В работе [4] показано, что алгоритм узнавания предмета, а также формальная запись базисного предиката узнавания предмета, представляют собой не что иное, как алгоритм и формальную запись работы ячейки ассоциативной памяти. Поэтому любой элемент узнавания предмета (рис. 3) может быть представлен в виде ячейки ассоциативной памяти, в которой записан код предмета а.

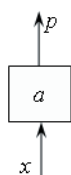


Рис. 3. Элемент узнавания буквы (предмета) а

Если на вход ячейки ассоциативного запоминающего устройства (АЗУ) подать зашифрованное имя предмета а, то на выходе этой ячейки появится сигнал 1 в случае, если в этой ячейке записан код имени данного предмета.

Если на вход ячейки подается код любого другого предмета, например, предмета b ($b \neq a$), то на выходе ячейки будет выработан сигнал 0.

Исходя из этого, элемент узнавания буквы (предмета) должен включать в свой состав шифратор, кодирующий имена предметов, принадлежащих универсуму предметов, и ячейку ассоциативной памяти, в которой записан двоичный код имени предмета а. Логические схемы ячейки АЗУ распознают код этого предмета (рис. 4).

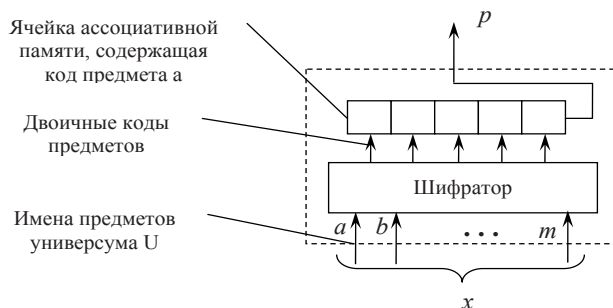


Рис. 4. Структурная схема элемента узнавания буквы (предмета) а

Например, элемент узнавания предмета, реализующий базисный предикат x_2^o , показанный на рис. 1, может быть представлен в виде структуры, включающей в свой состав шифратор и ячейку ассоциативной памяти (рис. 5).

Аналогичным образом можно представить элементы узнавания значений тестов «м» – молодой, «сл» – средних лет, «п» – пожилой, «с» – старческий. Таким образом, любую АП-структуру можно представить в виде поля ассоциативной памяти, хранящей коды предметов универсума и логического поля, выполняющего логические операции над предикатами узнавания предметов. В нашем случае речь идет о предикатах узнавания результатов тестов.

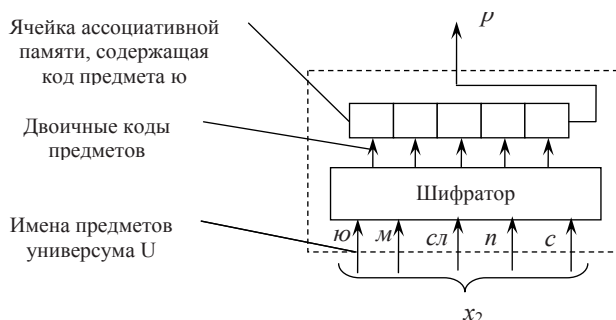


Рис. 5. Структура элемента узнавания буквы (предмета) «ю» – юный

Построенные структуры получили название ассоциативно-логических преобразователей.

6. Выводы

Алгебропредикатные структуры, представленные в виде ассоциативно-логических преобразователей, очень напоминают те мозгоподобные структуры, о которых писал в свое время В. М. Глушков [19]. Для построения ассоциативной памяти в составе АП-структур могут быть использованы нейронные сети в том числе – сети Хопфилда и Хэмминга, что делает ассоциативно-логические преобразователи схожими со структурами человеческого мозга. Полученные АП-структуры можно реализовать аппаратно с помощью программируемых логических интегральных схем (ПЛИС), и использовать при создании интеллектуальных систем параллельного действия, работающих в масштабе реального времени.

В будущем такие системы могут стать основой создания материального носителя интеллекта в виде мозгоподобной ЭВМ или «электронной нервной системы» (Digital Nervous System) [20].

Литература

1. Бондаренко, М. Ф. Основы теории багатозначних структур і кодування в системах штучного інтелекту [Текст] / М. Ф. Бондаренко, З. Д. Коноплянко, Г. Г. Четвериков. – Х. : Фактор-Друк, 2003. – 336 с.
2. Прокопчук, Ю. А. Принцип предельных обобщений: методология, задачи, приложения [Текст]: монография / Ю. А. Прокопчук. – Днепропетровск: Ин-т технической механики НАНУ и НКАУ, 2012. – 384 с.
3. Прокопчук, Ю. А. Модели когнитивных архитектур и процессов на основе парадигмы предельных обобщений [Текст] / Ю. А. Прокопчук // Кибернетика и вычисл. техника. – 2013. – Вып. 171. – С. 37–51.
4. Булкин, В. И. Представление алгебропредикатных структур в виде ассоциативно-логических преобразователей [Текст] / В. И. Булкин // Искусственный интеллект. – № 3. – С. 6–17.
5. Samsonovich, A. V.: Toward a Unified Catalog of Implemented Cognitive Architectures (Review) [Text] / A. V. Samsonovich, K. R. Johansdottir, A. Chella, B. Goertzel // Biologically Inspired Cognitive Architectures 2010: Proc. 1st Annual Meeting of BICA Society, Frontiers in Artificial Intelligence and Applications – 2010. – Vol. 221. – P. 195–244
6. Затуливетер, Ю. С. Графодинамические системы с сетцентрическим управлением в математически однородном поле компьютерной информации [Текст] / Ю. С. Затуливетер, Е. А. Фищенко // Управление большими системами. – 2010. – № 30–1. – С. 567–604.
7. Малинецкий, Г. Г. Теория самоорганизации. На пороге IV парадигмы [Текст] / Г. Г. Малинецкий // Компьютерные исследования и моделирование. – 2013. – Т. 5, № 3. – С. 315–366.
8. Rasmussen, D. A Neural model of rule generation in inductive reasoning [Text] / D. Rasmussen, C. Eliasmith // Topics in Cognitive Science. – 2011 – Vol. 3 (1). – P. 140–153.
9. Edelman, G. M. Biology of consciousness [Text] / G. M. Edelman, J. A. Gally, B. J. Baars // Front. Psychology. – 2011. – P. 2–4.
10. Dixon, J. A. Multifractal Dynamics in the Emergence of Cognitive Structure [Text] / J. A. Dixon, J. G. Holden, D. Mirman, D. G. Stephen // Topics in Cognitive Science. – 2012. – Vol. 4. – P. 51–62.
11. Бутенко, Д. В. Гомеостатическая нейросеть [Текст] / Д. В. Бутенко, Е. В. Албегов, Л. Н. Бутенко // Нейрокомпьютеры: разработка, применение. – 2013. – № 2. – С. 45–53.
12. Иващенко, А. В. Мультиагентные технологии для разработки сетцентрических систем управления [Текст] / А. В. Иващенко, О. В. Карсаев и др. // Известия Южного федерального университета. Технические науки. – 2011. – Т. 116, № 3. – С. 11–23.
13. Байрак, С. А. Параллельные процессоры для построения интеллектуальных систем [Текст]: матер. II междунар. научн.-техн. конф. / С. А. Байрак, Д. Н. Одинец и др. // Открытые семантические технологии проектирования интеллектуальных систем. – Минск: БГУИР, 2012. – С. 135–140.
14. Каляев, И. А. Высокопроизводительные реконфигурируемые вычислительные системы нового поколения [Текст] / И. А. Каляев, А. И. Дордопуло и др. // Вычислительные методы и программирование: новые вычислительные технологии. – 2011. – Т. 12, №2. – С. 82–89.
15. Brodtkorb, A. R. Graphics processing unit (GPU) programming strategies and trends in GPU computing [Text] / A. R. Brodtkorb, T. R. Hagen, M. L. Saetra // Journal of Parallel and Distributed Computing. – 2013. – Vol. 73, Issue 1. – P. 4–13.
16. Бухановский, А. В. Перспективная технология «облачных» вычислений второго поколения [Текст] / А. В. Бухановский, В. Н. Васильев и др. // Изв. вузов. Приборостроение. – 2011. – Т. 54, № 10. – С. 7–15.
17. Вереник, Н. Л. Разработка проблемно-ориентированных процессоров семантической обработки информации [Текст] / Н. Л. Вереник, Е. Н. Сейткулов, М. М. Татур // Электроника инфо. – 2012. – № 8. – С. 95–98.
18. Бондаренко, М. Ф. Мозгоподобные структуры [Текст]: Справ. пос.; Т. 1. / М. Ф. Бондаренко, Ю. П. Шабанов-Кушнаренко; под ред. И. В. Сергиенко. – К.: Наукова думка, 2011. – 460 с.
19. Глушков, В. М. Основные архитектурные принципы повышения производительности ЭВМ [Текст]: избр. тр.; Т. 2 / В. М. Глушков. – К.: Наукова думка, 1990. – 267 с.
20. Gates, B. Business at the Speed of Thought: Using a Digital Nervous System (Penguin Joint Venture Readers) [Text] / B. Gates. – Pearson Education Limited, 2001. – 112 p.