УДК 004.82 DOI 10.31429/vestnik-19-4-9-19

Структуры памяти и синтез знаний в сложно организованных интеллектуальных системах

К. И. Костенко ✉

Кубанский государственный университет, ул. Ставропольская, 149, Краснодар, 350040, Россия

✉ Костенко Константин Иванович; ORCID 0000-0002-9851-2455; e-mail: kostenko@kubsu.ru

Аннотация. Для построения моделей интеллектуальных систем используются принципы кибернетики. Основой реализации этих принципов являются математические описания концепций формализмов знаний, архитектуры интеллектуальных систем и управления интеллектуальными системами. Описания составляют базовую модель, расширения и детализации которой образуют структуру моделей разного уровня. Память интеллектуальных систем является одним из элементов модели. Она связана с форматами структур знаний, формирующихся в потоках процессов синтеза знаний. Общие структуры памяти реализуются как бесконечные насыщенные двоичные деревья, разделенные на подобласти, используемые в качестве доменов операций над знаниями. Комбинации прямых сумм и произведений доменов задают описания структур памяти. Особенностью доменов операций является возможность описания, основанного на регулярных выражениях. Определена топологическая структура регулярных областей памяти. Приведены схемы использования операций топологии при моделировании процессов синтеза знаний.

Ключевые слова: память интеллектуальной системы, регулярная область памяти, топология областей памяти, представление знаний, синтез знаний, операция над знаниями.

Финансирование. Работа выполнена при поддержке РФФИ и администрации Краснодарского края (проект № 19-41-230008) и при поддержке РФФИ (проект № 20-01-00289).

Цитирование: Костенко К. И. Структуры памяти и синтез знаний в сложно организованных интеллектуальных системах // Экологический вестник научных центров Черноморского экономического сотрудничества. 2022. Т. 19, № 4. С. 9–19. DOI 10.31429/vestnik-19-4-9-19

Поступила 22 ноября 2022 г. После доработки 24 ноября 2022 г. Принято 27 ноября 2022 г. Публикация 30 ноября 2022 г.

Автор заявляет об отсутствии конфликта интересов.

The Knowledge Synthesis in Memory Structures Complexly Organized Intelligent Systems

K. I. Kostenko ✉

Kuban State University, Stavropolskaya str., 149, Krasnodar, 350040, Russia

✉ Konstantin I. Kostenko; ORCID 0000-0002-9851-2455; e-mail: kostenko@kubsu.ru

Abstract. Cybernetic principles considered as the basis for the development of intelligent systems' formal models. These principles implementation founded on mathematical descriptions for general properties of knowledge formalisms, intelligent system architecture and intelligent systems control. Descriptions make up the basic model, the extensions and refinements of which form the structure of models at different levels. General memory structures modeled as infinite saturated binary trees, divided into subareas used as knowledge operation domains. Combinations of subareas direct sums and direct products represent memory structures within models descriptions. Considered knowledge operation domains set allow describing its elements by regular expressions. The topological structure of memory areas introduced for regular memory areas. The schemes of topology operations applications for knowledge synthesis processes are given.

Keywords: intelligent system memory, regular memory area, memory areas topology, knowledge representation, knowledge synthesis, knowledge domains.

Funding. The work was supported by the Russian Foundation for Basic Research and the administration of the Krasnodar Territory (project no. 19-41-230008) and with the support of the Russian Foundation for Basic Research (project no. 20-01-00289).

Cite as: Kostenko K.I. The knowledge synthesis in memory structures complexly organized intelligent systems. *Ecological Bulletin of Research Centers of the Black Sea Economic Cooperation*, 2022, vol. 19, no. 4, pp. 9–19. DOI 10.31429/vestnik-19-4-9-19

Received 22 November 2022. Revised 24 November 2022. Accepted 27 November 2022. Published 30 November 2022.

The author declare no competing interests.

Введение

Математическую модель интеллектуальной системы (ИС) образуют понятия и свойства, которые задают общие параметры и свойства интеллектуальности. Они являются многопредметными и исследуются в разных областях знаний. Это позволяет построить базовую формальную модель с универсальными возможностями моделирования и исследования свойств интеллекта. Модель использует уточнения фундаментальных понятий философии, когнитивной психологии, лингвистики, теории систем [1–3]. Модель применяет математический аппарат как основу для формализации понятий в других предметных областях [4–6]. Это позволяет рассматривать ИС как элементы класса систем, определяемых множествами формальных описаний, получаемых из описаний общих параметров и свойств интеллекта, изучаемыми в различных областях знаний. Базовую модель ИС составляют формальные описания параметров и свойств формализмов представления знаний, структур потоков процессов знаний и элементов управления жизненными циклами ИС [4, 7].

Для формализации элементов базовой модели ИС используются концепции и порождающие принципы, принятые в лингвистике и когнитивной психологии [1, 2]. Кибернетические инварианты для управления знаниями на основе принципов системной инженерии и кибернетики [8]. Математическая модель ИС основана на формализмах представления знаний [6]. Базовая модель преобразуется в модели прикладных ИС операциями кастомизации расширения и детализации моделей.

Память является одним из базовых понятий для моделирования интеллекта. Кратковременная память и долговременная (семантическая) память, разделенные на подклассы сознательной и бессознательной памяти, образуют первые уровни в рамках принятых классификаций памяти [9]. Фундаментальное понятие познания, реализуемое как модель формирования содержания памяти человека, по своим свойствам подобно структурам знаний в формализмах семантических иерархий [6]. Эта аналогия не решает проблему тождества естественного (сильного) и искусственного (слабого) интеллекта [9]. Она позволяет открывать новые возможности искусственного интеллекта.

Структуру памяти образуют отдельные подобласти как домены для синтеза знаний операциями над знаниями. Отдельные области соответствуют классам знаний с близкими структурными и семантическими свойствами.

Множество операций над доменами включает прямую сумму доменов и прямое произведение доменов. Эти операции позволяют определять домены со сложной структурой памяти на основе заранее выбранных регулярных областей памяти. Описания структур памяти определяются далее как комбинации доменов и операций, применяемых в ходе реализации процессов синтеза знаний в памяти ИС. Процессы используют иерархию классов специальных операций (морфизмов) знаний. Этими операциями моделируются этапы процессов, формирующих и использующих семантические структуры в памяти ИС.

1. Формализмы представления знаний

Понятие формализма знаний (KRF) введено как основа моделирования содержания предметных областей [4]. Формализмы позволяют исследовать структурные свойства, преобразования и сравнения знаний. Общее определение формализма соответствует четвёрке $\Im = (M, D_M, \circ, \prec)$. Здесь M и D_M обозначают алгоритмически перечислимые множества знаний и фрагментов знаний. Также $M \subseteq D_M$ и множество M разрешимо для D_M. Множество M содержит «*пустое знание*», которое обозначается как Λ. Остальные элементы четвёрки это композиции $\circ : D_M \times D_M \to D_M$ и разрешимое отношение вложения фрагментов знания $\prec \subseteq D_M \times D_M$.

Операция $\circ$ определяет алгебраические структуры элементов D_M как композиций элементов этого множества. Эти структуры представляются бинарными деревьями, внутренние вершины которых размечены символом $\circ$, а листья — элементами D_M. Двоичные последовательности удобны в качестве вершин в алгебраических структурах знаний. Символ I далее обозначает множество всех таких последовательностей. Следующее правило определяет последовательности, используемые в качестве вершин деревьев: корень дерева соответствует пустой бинарной последовательности и, если внутренняя вершина дерева соответствует $\alpha \in I$, то следующие левый (правый) потомки этой вершины — это $\alpha 0$ ($\alpha 1$).

1.1. Формализмы семантических иерархий

Формализмы семантических иерархий (SHF) — это подкласс множества формализмов знаний. Знания в таких формализмах называются конфигурациями. Эти формализмы используются для моделирования структур знаний, использующих семантические отношения между фрагментами таких структур. Пусть M — алгоритмически перечислимое множество, элементы которого называются конфигурациями, и $\Lambda \in M$, R — алгоритмически перечислимое множество разрешимых отношений на M. Для пар элементов R разрешимо свойство вложения множеств. Также заданы алгоритмически вычислимое отображение разложения конфигураций $\varepsilon : M \to M \times M$, называемое (связывания конфигураций $\psi : M \to R$). Конфигурация $z \in M$ называется элементарной, если $\varepsilon(z) = (\Lambda, \Lambda)$. Также $\varepsilon(\Lambda) = (\Lambda, \Lambda)$. Если $\varepsilon(z) = (z_1, z_2)$, $(z_1, z_2) \neq (\Lambda, \Lambda)$, и $\psi(z) = r$, то $(z_1, z_2) \in r$. В последнем случае z состоит из конфигураций z_1 и z_2. Отношение r связывает z_1 и z_2 в z.

Отображения ε и ψ определяют структуры конфигураций (CSR) как бинарные деревья с размеченными вершинами. Следующие правила определяют уникальное такое дерево $\Sigma(z)$ для структурного представления конфигурации z. Всякая элементарная конфигурация представляются одновершинным деревом. Неэлементарную конфигурацию $z \in M$, для которой $\varepsilon(z) = (z_1, z_2)$, представляет дерево, корень которого размечен как $\psi(z)$ с левым и правым поддеревьями корня, представляющими конфигурации z_1 и z_2.

Частным случаем формализмов семантических иерархий являются формализмы с конечной глубиной структур конфигураций. Параметрами структур являются множества вершин ($D(z)$) и множества листьев ($O(z)$) представления конфигураций z. Глубина $z \in M$ определяется как максимум длин путей, выходящих из корня, и обозначается как $d(z)$. Если $\alpha \in D(z)$, то выражение $[z]_\alpha$ ($(z)_\alpha$) используется для обозначения разметки этой вершины (семантической иерархии, представленной поддеревом CSR с корнем α). В формализмах семантических иерархий каждой иерархии соответствует единственная структура. Это позволяет не различать конфигурации и структуры конфигураций.

1.2. Домены конфигураций

Доменами отдельных операций над знаниями в формализмах семантических иерархий являются специальные классы конфигураций (структур конфигураций). Разнообразие таких классов включает все конфигурации $\Sigma = \{\Sigma(z) \,|\, z \in M\}$, семантические иерархии фиксированной глубины — Σ_k ($\Sigma_k = \{\Sigma(z) \,|\, z \in M \ \& \ d(z) = k\}$), где $k \in \{0, 1, \ldots\}$. Классы Σ_0 ($k = 0$) и Σ_1 ($k = 1$) образуют элементарные и простые знания. Примерами классов являются также последовательности (серии) и окрестности конфигураций [6].

Универсальная система классов операций над знаниями конструируется уточнением функциональных элементов разных областей математики как морфизмов знаний [6]. Структурные преобразования знаний моделируются операциями конфигураций в конфигурации ($Ins : \Sigma \times \Sigma \times I \to \Sigma$), удаление подконфигураций ($Del : \Sigma \times I \to \Sigma$), замена подконфигурации конфигурацией ($Exch : \Sigma \times \Sigma \times I \to \Sigma$), перестановка подконфигураций ($Perm : \Sigma \times I \times I \to \Sigma$) и извлечение конфигурации из конфигурации с помощью отображений трассирования ($Extr : \Sigma \times TR \to \Sigma$) [5,6]. В последнем выражении TR обозначает множество отображений трассирования на I. Эти отображения изотонно отображают I в I. Трассирования сохраняют порядок вершин для обхода дерева по алгоритму поиска в глубину. Инъективные трассирования определяют класс растяжений. Если $f \in TR$ и для $z \in M$ имеет место равенство $f(z, \xi) = z'$, то это означает, что ξ устанавливается соответствие вершин семантической иерархии z' вершинам семантической иерархии z [6]. Трассирования важны для моделирования сравнений знаний в *KRF*. Процессы синтеза знаний обычно заканчиваются извлечением знаний из синтезированных структур с помощью подходящих трассирований.

2. Синтез знаний в памяти ИС

Общие свойства структур памяти компонентов ИС адаптируются к процессам синтеза знаний в компонентах. Структура компонентов моделируется понятием измерения знаний [5]. Измерения относятся к параметрам знаний, определяемым конечными множествами. Значения измерений позволяют группировать знания с одинаковыми свойствами. Упорядоченные наборы значений отдельных измерений задают компоненты ИС. Множество всех таких наборов образует решетку компонентов ИС [4, 5].

Память является атрибутом компонента ИС. Она применяется для моделирования хранения и процессов обработки знаний. Каждой компоненте ИС соответствует формализм представления знаний. Этот формализм адаптирован к значениям измерений знаний для компонента. Общая структура памяти определяется бесконечным насыщенным бинарным деревом. Разметки вершин поддеревьев дерева задают структурные представления знаний в памяти.

2.1. Структуры памяти процессов синтеза знаний

Домены операций образуют алгоритмически перечислимый класс $\mathcal{D}$ перечислимых множеств конфигураций. Заданные ранее примеры доменов являются элементами $\mathcal{D}$. Конструирование доменов выполняют специальные операции. Эти операции позволяют определять домены с желаемыми свойствами как композиции доменов. Такими операциями являются прямая сумма и прямое произведение доменов ($\oplus$ и $\otimes$) [4]. Если $B_1, B_2 \in \mathcal{D}$, то $B_1 \oplus B_2 = \{z_1 \oplus z_2 \mid z_1 \in B_1 \,\&$ $\& \; z_2 \in B_2\}$.

Схема применения этой операции к $z_1 \in B_1 \;\&\; z_2 \in B_2$ приведена на рис. 1. Символы α, β, γ, λ, σ, δ, используемые на следующих рисунках, обозначают вершины памяти. Другие обозначения означают знания (фрагменты знаний), размечающие вершины. Корневая вершина суммы $z_1 \oplus z_2$ допускает любую возможную связь между конфигурациями z_1 и z_2. Множество возможных отношений для пар конфигураций является параметром операции прямой суммы. Стандартный случай этой операции соответствует использованию пустого отношения, которое выполняется для любых двух конфигураций.

Если $B_1, B_2 \in \mathcal{D}$, то

$$B_1 \otimes B_2 = \big\{z \mid \exists\, z_1 \in B_1(\forall\, \alpha \in D(z_1)\backslash O(z_1)\big(([z]_\alpha = [z_1]_\alpha) \;\& \\ \& \;\forall\, \alpha \in O(z_1)([z]_\alpha = [z_1]_\alpha \vee (z)_\alpha \in B_2)\big)\big\}.$$

Прямое произведение доменов состоит из конфигураций, составленных из элементов B_1 с заменой некоторых листьев в их структурах элементами B_2. Пример применения этой операции приведён на рис. 2.

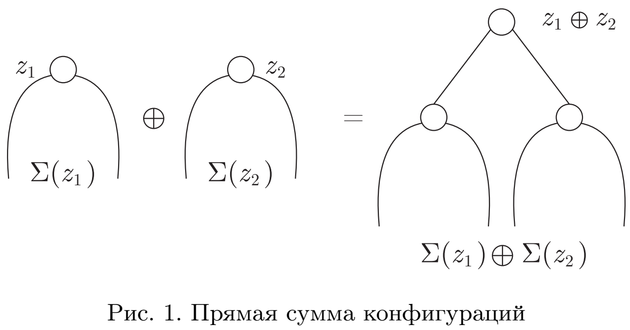


Рис. 1. Прямая сумма конфигураций

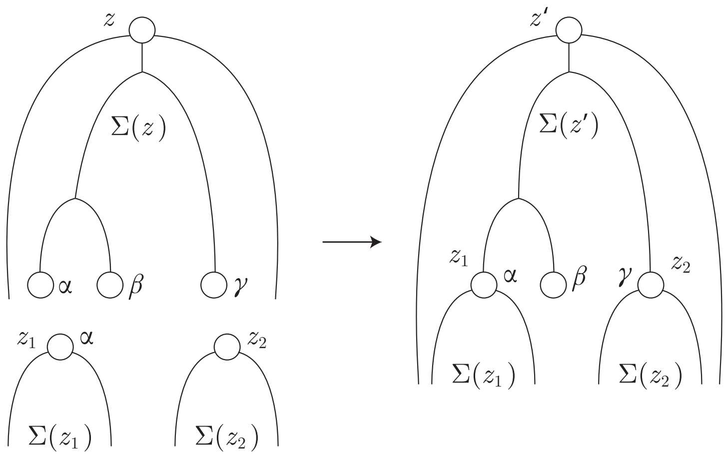


Рис. 2. Прямое произведение доменов

Конфигурация $z' \in B_1 \otimes B_2$ построена из $z \in B_1$, и $z_2 \in B_2$. Вершины α, β и γ являются листьями $\Sigma(z)$. Тогда $(z)_\alpha$ и $(z)_\gamma$ заменяются на $\Sigma(z_1)$ и $\Sigma(z_2)$. Свойства вершины β не изменяются. Реализации операции прямого произведения доменов используют дополнительные предикаты (P_1 и P_2) для выбора листьев и конфигураций для вставки. Конструкция $(\otimes, P_1, P_2)$ относится к языку синтеза знаний высокого уровня.

Комбинации доменов позволяют определять форматы областей памяти как структуры доменов. Это позволяет разделить память компонента на подобласти. Эти подобласти соответствуют структурам памяти, в которых реализуются операции и процессы обработки знаний в компонентах ИС.

Композиции доменов позволяют определить домены последовательностей (серий) элементарных и простых знаний $S \otimes M_0$ и $S \otimes M_1$. Последний класс используется для множеств простых знаний, составляющих содержание (онтологии) предметных областей. Подобные формулы памяти определяют последовательность окрестностей элементарных знаний разной глубины: $S \otimes O$, $(S \otimes O) \otimes O$, $\ldots$, $(\ldots(S \otimes O)\ldots) \otimes O$.

2.2. Потоки процессов обработки знаний

Другими элементами структуры ИС являются потоки знаний между компонентами и процессы обработки знаний внутри компонент. Диаграммы потоков знаний представляются в формате нагруженных ориентированных графов с вершинами, представляющими компоненты ИС, и ребрами, представляющими схемы передачи знаний между компонентами [4, 5]. Параметры потоков включают условия передачи знаний между компонентами ИС, и преобразования трансформации знаний из одного формализма в другой, а также форматы областей памяти для хранения передаваемых знаний. Пример потоков знаний в двумерной решетке компонентов ИС показан на рис. 3. Система компонентов ИС формируется для двух измерений с тремя возможными значениями в каждом измерении. Значения A (поверхностный), B (алгоритмический), C (когнитивный) и 3 (полностью структурированный), 2 (частично структурированный), 1 (неструктурированный) используются здесь для измерений абстрактности знаний и степени декомпозиции знаний. Четыре потока знаний с номерами 1–4 показаны на рис. 3.

Поток 1 реализует общую схему решения задач в предметной области. Этот поток проходит через компоненты, в которых исходное представление задачи декомпозируется и трансформируется во внутреннее представление (A1 – A2 – A3 – B3). Решение задач реализовано на алгоритмическом уровне (B3 – B2 – B1). Решение задачи начинается в B3, синтезируется в B2 и извлекается в компоненте B1. Последний этап потока знаний реализует трансляцию решения в формат предметной области A1. Другие потоки знаний (2–4) реализуют синтез метода решения задач (2), построение шаблонов решения задач (3) и запрос дополнительных данных при недостатке начальных данных (4).

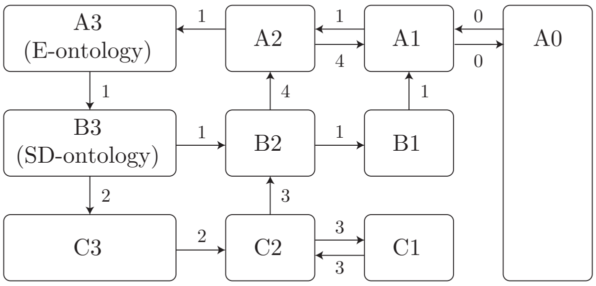


Рис. 3. Потоки знаний в ИС

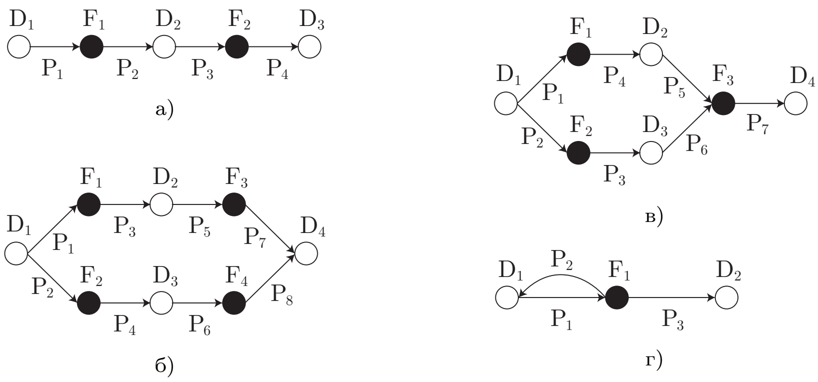


Рис. 4. Диаграммы процессов обработки знаний

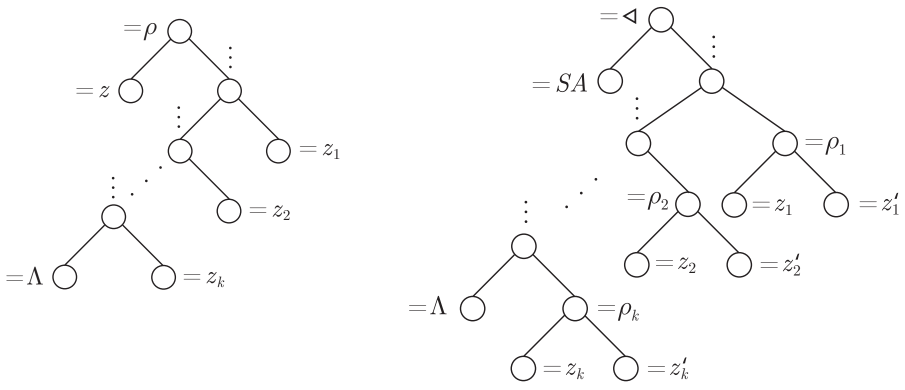
Процессы обработки знаний представляются специальными диаграммами. Диаграмма представляет собой размеченный ориентированный двудольный граф. Вершины диаграммы размечают классы операций и домены операций. Ребра диаграмм размечают формулами условий прохождения рёбер. Набор типов диаграмм включает диаграммы для последовательных, параллельных, вариантных, циклических и иерархических структур процессов [9]. Примеры диаграмм приведены на рис. 4. Индексированные символы F_i, D_i и P_i обозначают классы операций, домены операций и предикаты условий.

Случай а) на рис. 4 показывает последовательную диаграмму. Случаи б), в) и г) иллюстрируют другие три типа диаграмм обработки знаний. Эти диаграммы аналогичны диаграммам морфизмов в теории категорий и допускают формальное исследование и различные приложения [9].

3. Регулярные области памяти

Структура памяти компонента ИС соответствует бесконечному бинарному дереву с левыми и правыми потомками у каждой вершины. Вершины дерева представлены элементами I. Последовательность $\alpha \in I$ определяет путь, который начинается в корне дерева и заканчивается в вершине $\alpha \in I$. Элементы последовательности 0 и 1 управляют выбором левых или правых потомков внутренних вершин пути. Знания хранятся в виде их структурированных представлений, размещённых в вершинах некоторого поддерева памяти. Если $\alpha \in I$, то $I_\alpha = \{\beta \mid \beta = \alpha\gamma \ \& \ \gamma \in I\}$.

Рассмотрим класс структур памяти, описываемых регулярными подмножествами I. Такие подмножества определяются регулярными выражениями в алфавите $\{0, 1\}$ (Пустая строка λ является регулярным выражением. Символ x, где $x \in \{0, 1\}$ является регулярным выражением. Если E_1 и E_2 являются регулярными выражениями, то строки $E_1 \cup E_2$, $E_1 \cdot E_2$ и E_1^* являются регулярными выражениями). Каждое выражение определяет множество вершин, используемых для хранения структур конфигураций в памяти.



а) б)

Рис. 5. Структура окрестностей знаний и серий простых знаний

Если E — регулярное выражение, то оно определяет $U(E) \subseteq I$, называемое регулярным множеством. Элементы этого множества используются в качестве листьев в CSR конфигураций. Множество конфигураций, определяемое E, обозначенным как $M(E)$. Чтобы определить это множество, нам понадобится понятие замыкания $B \subseteq I$: $[B] = \{\alpha \mid \exists\, \beta \in B(\alpha \subseteq \beta)\}$. Выражение $\alpha \subseteq \beta$ означает, что последовательность β начинается с последовательности α. Следующее выражение определяет $M(E)$

$$z \in M(E) \Leftrightarrow (z = \Lambda) \vee \forall\, \alpha \in O(z)\backslash U(E)\big([z]_\alpha = \Lambda \;\&\; \alpha = \beta\sigma \;\&\; |\,\sigma\,| = 1 \;\&\; \beta \in [O(z) \cap U(E)]\big).$$

Здесь определена структура конфигурации. Листья структуры принадлежат множеству $O(z)\cap$ $\cap\, U(E)$ или находятся на минимальном расстоянии от $[O(z) \cap U(E)]$. В таком случае листья размечены элементарным знанием Λ.

Для любого $z \in M(E)$ множество $U(E)$ определяет множество вершин, содержащее листья CSR z, допускающие маркировку непустыми конфигурациями. Следующие соотношения представляют свойства конфигураций из $M(E)$:

1) $\forall\, z \in M(E)(\alpha \in O(z) \;\&\; \alpha \notin U(E) \;\&\; I_\alpha \cap U(E) \neq \varnothing \to [z]_\alpha = \Lambda)$;

2) $\forall\, z \in M(E)(\alpha \in D(z) \;\&\; I_\alpha \cap U(E) = \varnothing \to [z]_\alpha = \Lambda)$;

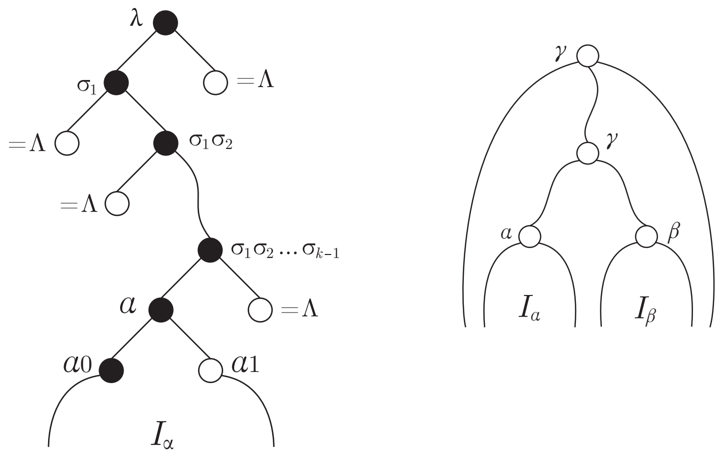
3) $\forall\, \alpha \in [U(E)]\, \exists\, z \in M(E)(\alpha \in D(z))$.

Первое свойство относится к случаю, когда лист CSR конфигурации не принадлежит $U(E)$. Второе свойство означает, что для любого $z \in M(E)$ в $D(z)$ нет вершин, лежащих ниже вершин $U(E)$. Если $M(E) \in \mathcal{D}$, то последнее свойство означает, что $[U(E)]$ это минимальное подмножество I, достаточное для представления всех элементов $M(E)$.

На рис. 5 показаны два примера регулярных областей памяти для операций со знаниями. Общая структура окрестности глубины 1 элементарного знания z в отношении ρ представлена на рис. 5а. Вершина 10^*, отмеченная пустым знанием, соответствует завершению последовательности $z_1, \ldots, z_k$ — соседей z, для которых $z\rho z_i$, $i = 1, \ldots, k$. Область памяти знаний, представляющих все такие окрестности, задаёт выражение $[0 \cup 10^*1]$. Конфигурация, определяемая как последовательность простых знаний, приведена на рис. 5б. Разметка вершины как $=SA$ обозначает предметную область. Корень дерева отмечен отношением агрегирования содержания предметной области в формате последовательности простых знаний $z_i\rho_i z_i'$, $i = 1, \ldots, k$. Регулярное выражение $0 \cup 10^*1(0 \cup 1)$ определяет множество листьев в деревьях, которые представляют эти знания.

Пусть E это регулярное выражение и $\alpha \in I$. Определим множество $D(\alpha, E) = \{\beta \mid \exists\, \gamma \in$ $\in [U(E)](\beta = \alpha\gamma)\}$. Множество $D(\alpha, E)$ является регулярным и называется регулярной областью памяти. Если $|\,\alpha\,| > 0$ и $\alpha = \sigma_1 \ldots \sigma_k$ $(\sigma_1, \ldots, \sigma_k \in \{0, 1\})$, то множество $M(D(\alpha, E))$ состоит из конфигураций, которые выглядят так, как показано на рис. 6а.

Если $z \in M(D(\alpha, E))$, то путь, который начинается в λ и заканчивается в α составляют вершины из $D(z)\backslash O(z)$. Соседи этих вершин, лежащие в стороне, отмечены пустыми знания-



а) б)

Рис. 6. Регулярные области памяти

ми (Λ). Если пустое знание означает неизвестный (неопределенный) объект или объект поиска, то I_α содержит текущее значение значения знания, представленное частью z. Тогда можно рассматривать часть z без этих неопределенных элементов. Это позволяет идентифицировать регулярные множества знаний $M(D(\alpha, E))$ и $M(E)$ при выполнении дополнительного условия $\forall\, \beta \in I(\beta \subset \alpha \to [z]_\alpha = \perp)$. Символ $\perp$ в последнем выражении означает пустое отношение на M. Это отношение связывает произвольные пары знаний минимальной зависимостью.

4. Топология регулярных областей памяти

Символ $\mathcal{E}$ далее обозначает множество регулярных выражений. Множество $[D(\alpha, E)]$, где $\alpha \in I$ и $E \in \mathcal{E}$, связано с открытым подмножеством множества I.

Пусть $\mathcal{B} = \{B \,|\, E \in \mathcal{E} \;\&\; \exists\, \alpha \in I(B = [U(E)] \cap I_\alpha\}$. Легко доказать, что множества $\varnothing$ и I принадлежат $\mathcal{B}$. Следовательно, $\mathcal{B}$ определяет топологическое пространство $\mathcal{T}_I = (I, X_I)$ с множеством X_I, образованным произвольными объединениями элементов $\mathcal{B}$. Это топологическое пространство удовлетворяет аксиоме отделимости Хаусдорфа.

Если $\alpha, \beta \in I$, $\alpha \not\subset \beta$ и $\beta \not\subset \alpha$ (см. рис. 6б), то $D(\alpha, E_1) \cap D(\beta, E_2) = \varnothing$. Пусть $\alpha = \sigma_1 \ldots \sigma_m$, $\beta = \sigma_1 \ldots \sigma_m \sigma_{m+1} \ldots \sigma_n$. Следующие элементы $\mathcal{B}$ не пересекаются:

$$B_\alpha = [U(\sigma_1 \ldots \sigma_m \overline{\sigma_{m+1}}(\lambda \cup (0{,}1)*)] \cap I_\alpha.$$

$$B_\beta = [U(\sigma_1 \ldots \sigma_m \sigma_{m+1} \ldots \sigma_n(\lambda \cup (0{,}1)*)] \cap I_\beta.$$

Для B_α и B_β выполняются условия: $\alpha \in B_\alpha$ и $\beta \in B_\beta$. Это означает, что $\mathcal{T}_I = (I, X_I)$ это пространство Хаусдорфа.

Топология $\mathcal{T}_I$ позволяет определить ассоциированное топологическое пространство $\mathcal{T}_M = (M, X_M)$. Базой этого пространства является множество $\{M(D(\alpha, E)) \,|\, \alpha \in I \;\&\; E \in \mathcal{B}\}$. $\mathcal{T}_M$ не является пространством Хаусдорфа.

Пусть область памяти используется несколькими операциями для размещения результатов операций. Последовательное выполнение операций обработки знаний изменяет структуры семантических иерархий, размещаемых в этой области. В описании этой области необходимо использовать объединения и пересечения регулярных областей памяти. Рассмотрим примеры композиций памяти в процессах синтеза знаний. Они приведены на рис. 7.

На рисунке приведены два примера описания областей памяти для синтезированных структур знаний. Случай а) (рис. 7а) относится к замене данной $z \in M(E)$ листа $\alpha \in U(E)$ структурой для $z_1 \in M(E_1)$. Пусть E_2 — регулярное выражение для множества $U(E) \backslash \{\alpha\}$, тогда $E_2 \cup \alpha E_1$ — регулярное выражение, область памяти для конфигурации $Exch(z, z_1, \alpha)$. Описание этой области имеет вид $D(\lambda, E_2) \cup D(\alpha, E_1)$, где $D(\lambda, E_2), D(\alpha, E_1) \in X_I$.

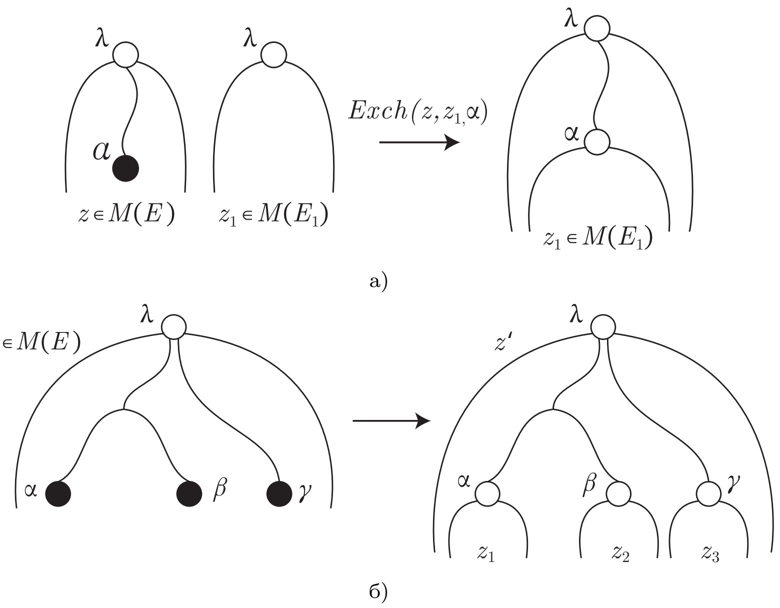


Рис. 7. Композиции областей памяти для доменов операций

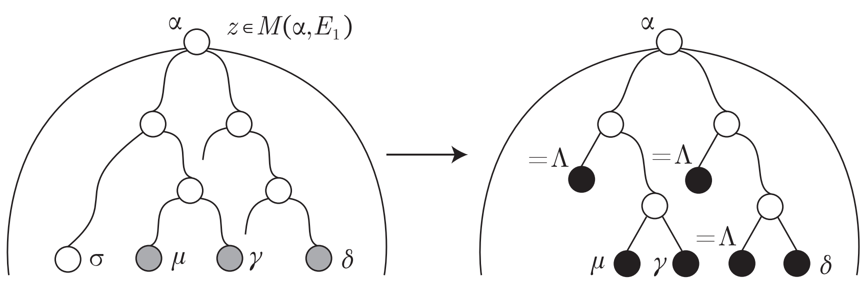


Рис. 8. Преобразование семантической иерархии для пересечения доменов

Случай б) (рис. 7б) представляет ситуацию, когда конфигурация синтезируется с использованием прямого произведения. Вершины α, β, $\gamma \in U(E)$ являются листьями в CSR $z \in M(E)$. Эта конфигурация трансформируется в новую путем замены вершин α, β и γ на $z_1, z_2, z_3 \in B$. Если B представляется $E_1 \in \mathcal{E}$, то выражение $E \cdot E_1$ соответствует области памяти для z'.

Рассмотрим применение операции пересечения открытых множеств X_I при синтезе знаний. Пусть $D(\alpha, E_1), D(\beta, E_2)$ — регулярные множества и $z \in M(\alpha, E_1)$. Тогда пересечение $D(\alpha, E_1) \cap D(\beta, E_2)$ определяет множество, состоящее из общих элементов $D(\alpha, E_1)$ и $D(\beta, E_2)$. Пересечение $D(\alpha, E_1) \cap D(\beta, E_2)$ связано с преобразованием z в новую конфигурацию.

Пример такого преобразования для $\alpha = \beta$ показан на рис. 8.

Конфигурация z удовлетворяет условиям:

$$\sigma, \mu, \lambda, \delta \in O(z), \mu, \lambda, \delta \in D(\alpha, E_1) \cap D(\alpha, E_2), \quad \sigma \notin D(\alpha, E_1) \cap D(\alpha, E_2).$$

Следующие правила задают спецификацию $D(\alpha, E_1) \cap D(\beta, E_2)$:

1) Если $\alpha \subseteq \beta$ ($\beta = \alpha\kappa$), то $D(\alpha, E_1) \cap D(\beta, E_2) = \{\kappa\omega \mid \kappa\omega \in D(\alpha, E_1) \ \& \ \omega \in D(\beta, E_2)\}$.
2) Если $\alpha \not\subset \beta$ & $\beta \not\subset \alpha$, то $D(\alpha, E_1) \cap D(\beta, E_2) = \varnothing$.
3) Если $\beta \subset \alpha$ ($\alpha = \beta\kappa$), то $D(\alpha, E_1) \cap D(\beta, E_2) = \{\kappa\omega \mid \kappa\omega \in D(\beta, E_2) \ \& \ \omega \in D(\alpha, E_1)\}$.

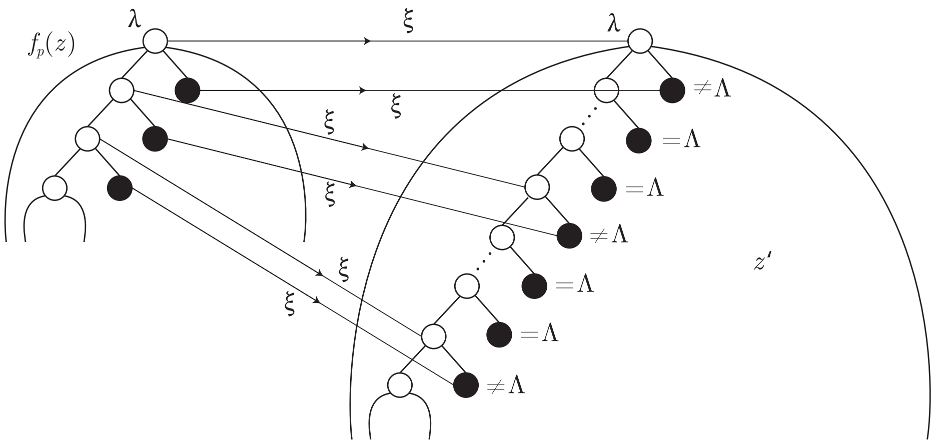


Рис. 9. Сжатие конфигурации после удаления фрагментов

5. Трассирования знаний в регулярных областях памяти

Пересечение областей памяти удобно для моделирования извлечения знаний из знаний. Результаты извлекаются удалением несущественных частей синтезированных знаний. Каждая такая часть связанна с областью памяти $I_\alpha \subseteq I$, которая удовлетворяет условию $I_\alpha \cap U(E) = \varnothing$. Выражение E определяет область памяти, в которой содержатся результаты процесса синтеза. Пусть $z \in M(E_1)$ — синтезированная конфигурация и E_2 — выражение, задающее фрагмент z. После замены фрагментов структуры z за пределами области $D(\lambda, E_1) \cap D(\lambda, E_2)$ на Λ конфигурация z превращается в объект z'.

Затем выполняется сжатие z' с помощью трассирования ξ:

$$\xi(\lambda) = \lambda;$$

$$\forall\, \alpha \in I, \sigma \in \{0,1\} \left(\xi(\alpha\sigma) = \begin{cases} \xi(\alpha)\beta, \beta = \delta_1\delta_2\ldots\delta_k \ \&\ \delta_1 = \\ \qquad = \sigma \ \&\ \xi(\alpha)\beta \in D(z')\backslash O(z') \ \&\ \forall\, j \in \{1,\ldots,k-1\}; \\ ([z']_{\xi(\alpha)\delta_1\delta_2\ldots\delta_{j-1}\bar{\delta}_j} = \Lambda)(\&\ [z']_{\xi(\alpha)\delta_1\delta_2\ldots\delta_j} \neq \Lambda)\ \& \\ \qquad \&\ ([z']_{\xi(\alpha)\delta_1\delta_2\ldots\delta_k 0} \neq \Lambda)\ \&\ ([z']_{\xi(\alpha)\delta_1\delta_2\ldots\delta_k 1} \neq \Lambda); \\ \xi(\alpha)\sigma, \text{ иначе.} \end{cases} \right).$$

Пусть $T(\xi, z)$ — размеченное бинарное дерево, построенное из вершин $D(z') \cap \xi(I)$. Следующее выражение определяет p-эндоморфизм $f_p : M \to M$.

$$f_p(z) = \begin{cases} T(\xi, z), T(\xi, z) \in \Sigma; \\ z, \text{ иначе.} \end{cases}$$

На рис. 9 показано, как объект z' трансформируется в значение $f_p(z)$.

Извлечение осуществляется путем удаления лишних частей z'. Удаляемые части z распознаются как внешние по отношению к памяти, задаваемой E. Вершины серого цвета представляют листья структуры знаний. Белые вершины $f_p(z)$ являются внутренними в представлении z'. Некоторые листья $O(z')$ отмечены пустыми знаниями. Они соответствуют удаляемым поддеревьям $D(z)$. Отображение ξ пропускает эти вершины.

Заключения и выводы

Применение специальных понятий расширяет возможности описания процессов синтеза знаний. Описание подобластей памяти определяет форматы размещаемых знаний при реализации целей системы. Эффективность регулярных форматов связана с доменами операций обработки знаний. Свойство регулярности также обеспечивает эффективность обработки структур синтезируемых знаний.

Операции усечения и сжатия структур конфигурации с помощью регулярных выражений основаны на формате отображений $f : M \times E \to M$ ($f : \Sigma \times E \to \Sigma$). Они определяют новый класс морфизмов знаний, удобный для исследования структур памяти и процессов мышления для формализации понятия когнитома [9].

Литература [References]

1. Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R., *Taxonomy of educational objectives: The classification taxonomy of educational goals. Handbook 1: Cognitive domain.* David McKay, New York, 1956.
2. Stanovich, K.E., *Rationality and the reactive mind.* Oxford Univ. Press, 2010.
3. Burgin, M., *Theory of knowledge: structures and processes.* World Scientific Series in Information Studies, Vol. 5, 2017. DOI 10.1142/8893
4. Костенко, К.И., Инварианты ядра фундаментальной модели интеллектуальной системы. *Программная инженерия*, 2021, т. 12, № 3, с. 157–168. [Kostenko, K.I., Invariants of the core of the fundamental model of an intelligent system. *Programmnaya inzheneriya = Software Engineering*, 2021, vol. 12, no. 3, pp. 157–168. (in Russian)] DOI 10.17587/prin.12.157-168
5. Kostenko, K., Knowledge flows processes at multidimensional intelligent systems. *Russian Advances in Artificial Intelligence: selected contributions to the Russian Conference on Artificial intelligence (RCAI 2020)*, October 10–16, 2020, Moscow, Russia, vol. 2648, pp. 74–84. URL: http://ceur-ws.org/Vol-2648/paper6.pdf
6. Костенко, К.И., Операции когнитивного синтеза формализованных знаний. *Программная инженерия*, 2018, т. 9, № 4, с. 174–184. [Kostenko, K.I., Operations of cognitive synthesis of formalized knowledge. *Programmnaya inzheneriya = Software Engineering*, 2018, vol. 9, no. 4, pp. 174–184. (in Russian)] DOI 10.17587/prin.9.174-184
7. Wooldridge, M., Intelligent Agents. In Weiss G. (ed.), *Multiagent Systems (A Modern Approach to Distributed Modern Approach to Artificial Intelligence).* The MIT Press, 1999, pp. 27–122.
8. Mesarovic, M., Takahara, Y., *General Systems Theory: Mathematical Foundations (Mathematics in Science and Engineering).* Elsevier, 1975.
9. Анохин, К.В., Когнитом: в поисках фундаментальной нейронаучной теории сознания. *Журнал высшей нервной деятельности им. И.П. Павлова*, 2021, т. 71, № 1, с. 39–71. [Anokhin, K.V., Cognit: in search of a fundamental neuroscientific theory of consciousness. *Zhurnal vysshey nervnoy deyatel'nosti im. I.P. Pavlova = I.P. Pavlov's Journal of Higher Nervous Activity*, 2021, vol. 71, no. 1, pp. 39–71. (in Russian)] DOI 10.31857/S0044467721010032
10. Костенко, К.И., Диаграммы процессов и правила синтеза знаний из элементов онтологий. *VIII Международная научная конференция «Знания-Онтологии-Теории»*, Новосибирск, 8-12 ноября 2021, с. 131–140. [Kostenko, K.I., Process diagrams and knowledge synthesis rules from ontology elements. *VIII Mezhdunarodnaya nauchnaya konferentsiya "Znaniya-Ontologii-Teorii" = VIII International Scientific Conference "Knowledge-Ontology-Theory"*, Novosibirsk, November 8-12, 2021, pp. 131–140. (in Russian)]