УДК 658.012.011.56

ОБОБЩЕННЫЙ ПОДХОД К МОДЕЛИРОВАНИЮ И АНАЛИЗУ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ АЛГЕБРЫ КОРТЕЖЕЙ

Б.А. Кулик
Институт проблем машиноведения РАН (ИПМаш РАН)
Россия, 199178, Санкт-Петербург, Большой пр. ВО, 61
E-mail: bkulik@msa2.ipme.ru, ba-kulik@yandex.ru

Ключевые слова: интеллектуальные системы, алгебра кортежей, структуры данных, алгоритмы, базы данных, базы знаний, логико-вероятностные методы, параллельная обработка данных
Key words: intelligent systems, cortege algebra, algorithms, data structures, databases, knowledge bases, logical and probabilistic methods, concurred processing

Алгебра кортежей – это математическая система для формализации многоместных отношений. Она основана на свойствах декартова произведения, которые соответствуют основным законам исчисления предикатов. Поэтому структуры алгебры кортежей хорошо совместимы со структурами баз данных и баз знаний.
A GENERALIZED APPROACH TO MODELLING AND ANALYSIS OF INTELLIGENT SYSTEMS ON THE CORTEGE ALGEBRA BASIS / B.A. Kulik (Institute of Problem of Mechanical Engineering of RAS, 61, Bolshoi pr. V.O., St. Petersburg 199178, Russia, E-mail: bkulik@msa2.ipme.ru). The cortege algebra is a mathematical system for formalization of multiplace relations. It is based on properties of the Cartesian product which correspond to fundamental laws of predicate calculus. Therefore structures of cortege algebra are well compatible with the structures of databases and knowledge bases.

1. Введение

В настоящее время методы искусственного интеллекта и логического анализа востребованы и широко применяются в информационно управляющих системах. Накопленный к настоящему времени опыт проектирования интеллектуальных систем (ИС) показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов. Проектирование ИС выполняется в основном на интуитивном уровне с применением неформализованных методов, основанных на искусстве, практическом опыте, экспертных оценках и дорогостоящих экспериментальных проверках качества функционирования ИС.

В интеллектуальных системах совместно с логическими структурами часто используются обычные структуры (сети, базы данных и т.д.), которые при традиционном подходе плохо совместимы с логическими структурами, и разработчикам интеллектуальных систем трудно использовать комплексный подход, в котором должны взаимодействовать структуры искусственного интеллекта с

традиционными структурами. При разработке программного обеспечения ИС возникает ряд проблем, для решения которых требуются большие затраты времени и средств. К этим проблемам относятся:

1) отличия в теоретических подходах к построению баз данных (БД) и баз знаний (БЗ) в интеллектуальных системах, следствием чего являются существенные различия в структурах данных для БД (таблицы, графы) и БЗ (продукции, фреймы, семантические сети) и большие затраты времени и средств на разработку методов сопряжения БД и БЗ в одной ИС;

2) трудности распараллеливания операций при использовании традиционных структур данных и знаний в интеллектуальных системах;

3) отсутствие простой и прозрачной интерпретации результатов программирования БЗ для пользователя (практически невозможно проследить логику работы системы, у которой многие программные модули представляют собой сложные подпрограммы на языке высокого уровня и, как правило, недоступны для «посторонних»);

Поиск решения этих проблем ведется по многим направлениям, но при этом используются традиционные структуры данных для БЗ, теоретические основы которых были разработаны в середине прошлого века, и которые по своим характеристикам плохо согласуются с архитектурой современных компьютеров и не пригодны для существенного улучшения ситуации. Для конструктивного решения этого комплекса проблем необходимо не только развитие мощного программного обеспечения и повышение быстродействия вычислительных комплексов, но и использование нового математического обеспечения, позволяющего унифицировать многообразные структуры данных, используемых в ИС, и обладающего свойствами естественного параллелизма. В качестве одного из возможных подходов к решению этой задачи в докладе рассматривается математическая система, названная *алгеброй кортежей* [1-7].

2. Основы алгебры кортежей

2.1. Основные термины и структуры

Алгебра кортежей (АК) – это математическая система для моделирования и анализа многоместных отношений, но в отличие от *реляционной алгебры*, применяющейся для формализации БД, в ней можно использовать все средства логического моделирования и анализа систем, входящие в математическую логику. В основе АК использованы известные свойства декартова (прямого) произведения множеств [8], которые, как показали исследования, соответствуют основополагающим законам математической логики. В АК для более четкого изложения и понимания этих свойств использованы следующие новации:

- введена новая более удобная система обозначений для декартовых произведений;
- введен ряд терминов и доказаны математические соотношения, связывающие декартовы произведения с произвольными многоместными отношениями и законами математической логики.

Начнем с обозначений. Декартово произведение (ДП) множеств $A_1, A_2, ..., A_k$ традиционно обозначается как $A_1 \times A_2 \times ... \times A_k$. В алгебре кортежей эта операция представлена как структура $[A_1 \;\; A_2 \;\; ... \;\; A_k]$ и называется ***C-кортеж*** (одна из пяти структур АК). Множества A_i называются ***компонентами*** *C*-кортежа. Если

эту структуру «развернуть», т.е. вычислить соответствующее ей декартово произведение, то будет получено множество кортежей (последовательностей) элементов. Эти кортежи в АК называются ***элементарными кортежами*** (еще одна структура АК).

Свойства ДП выражаются в структурах АК таким образом:

a) $[A_1\ \ A_2\ \ ...\ \ A_k] \cap [B_1\ \ B_2\ \ ...\ \ B_k] = [A_1 \cap B_1\ \ A_2 \cap B_2\ \ ...\ \ A_k \cap B_k]$;

b) если при этом существует хотя бы одно i такое, что $A_i \cap B_i = \varnothing$, то $[A1\ \ A2\ \ ...\ \ A_k] \cap [B_1\ \ B_2\ \ ...\ \ B_k] = \varnothing$;

c) $[A_1\ \ A_2\ \ ...\ \ A_k] \subseteq [B_1\ \ B_2\ \ ...\ \ B_k]$, если и только если $A_i \subseteq B_i$ для любого $i = 1, 2, ..., k$.

Если речь идет об объединении C-кортежей, то в результате этой операции может быть получен C-кортеж только в исключительных строго определенных случаях. Соответствующее свойство ДП в обозначениях АК выражается с помощью отношения включения множеств:

d) $[A_1\ \ A_2\ \ ...\ \ A_k] \cup [B_1\ \ B_2\ \ ...\ \ B_k] \subseteq [A_1 \cup B_1\ \ A_2 \cup B_2\ \ ...\ \ A_k \cup B_k]$.

В то же время в АК операцию объединения ДП можно выразить с помощью равенства:

e) $$[A_1\ \ A_2\ \ ...\ \ A_k] \cup [B_1\ \ B_2\ \ ...\ \ B_k] = \begin{bmatrix} A_1 & A_2 & ... & A_k \\ B_1 & B_2 & ... & B_k \end{bmatrix}.$$

В правой части этого равенства появляется третья, подобная матрице, структура АК, которая называется ***C-системой***. Интерпретируется она как объединение элементарных кортежей, содержащихся в соответствующих C-кортежах. C-система является универсальной структурой, так как с ее помощью можно представить любое многоместное отношение.

АК содержит ряд определений и более 30-ти теорем, с помощью которых решаются следующие задачи:

a) обосновывается ее изоморфизм с такими системами как теория многоместных отношений и исчисление высказываний и предикатов;
b) осуществляется погружение этой системы в вероятностное пространство;
c) разрабатывается алгоритмическая база для решения разнообразных задач логического анализа систем (логического вывода, поиска корректных гипотез и «скрытых аксиом», вероятностного анализа и т.д.).

В основе АК лежит понятие гибкого универсума. Пусть задана некоторая совокупность различных множеств, называемых ***сортами***. Каждому сорту приписываются некоторое множество имен (***атрибутов***) (в предыдущих публикациях по АК вместо этого термина использовался не совсем удачный термин «координата»). ***Доменом*** каждого атрибута является множество, равное соответствующему сорту. В математической логике доменам атрибутов соответствуют области определения переменных. ***Гибкий универсум*** состоит из некоторой совокупности ***частных универсумов*** – декартовых произведений доменов для заданной последовательности атрибутов. Последовательность атрибутов, определяющих данный частный универсум, называется ***схемой отношения***.

АК содержит всего 5 структур (***АК-объектов***): элементарный кортеж; C-кортеж; C-система; D-кортеж и D-система. АК-объекты, сформированные в одном и том же частном универсуме, называются ***однотипными***.

Предположим, что задан некий частный универсум в виде декартова произведения $\boldsymbol{S} = X_1 \times X_2 \times ... \times X_n$ произвольных множеств. Наглядно $\boldsymbol{S}$ можно представить как некоторое *пространство* признаков с *атрибутами* X_i. Домены этих

атрибутов соответствуют *шкалам* признаков. Тогда в пространстве $\boldsymbol{S}$ можно сформировать следующие подструктуры:

a) *проекции* — подпространства, в которых используются только некоторые атрибуты из множества атрибутов, образующих $\boldsymbol{S}$;
b) декартовы произведения в заданной схеме отношения; образующими компонентами этих декартовых произведений являются некоторые подмножества множеств X_i, представленных в заданной схеме отношения.

Рассмотрим примеры этих подструктур. Пусть $\boldsymbol{S} = X \times Y \times Z$, где $X = \{a, b, c, d\}$, $Y = \{f, g, h\}$, $Z = \{a, b, c\}$. Проекциями данного пространства могут быть декартовы произведения $X \times Y$, $X \times Z$ и т.д. или одиночные множества, например, X. Для простоты будем считать, что в этой системе каждому сорту соответствует единственный атрибут.

В пределах пространства $\boldsymbol{S}$ или какой-либо его проекции можно задать соответствующие подструктуры в виде декартовых произведений. Например, в $\boldsymbol{S}$ соответствующей подструктурой может быть декартово произведение $R[XYZ]=\{b, d\}\times\{f, h\}\times\{a, b\}$. Выражение $[XYZ]$ обозначает схему отношения.

Нетрудно убедиться, что $R \subset \boldsymbol{S}$ (свойство декартовых произведений). Аналогично некоторое подмножество элементарных кортежей проекции $Y \times Z$ можно задать как декартово произведение $Q[YZ] = \{f, g\}\times\{a, c\}$. Декартовы произведения представляют множества элементарных кортежей, эти множества при необходимости можно перечислить, хотя при выполнении вычислений со структурами АК это перечисление необязательно, так как практически все операции в АК выполняются с компонентами (множествами).

Элементарный кортеж – элемент декартова произведения или его проекции, т.е. последовательность из элементов, каждый из которых принадлежит домену соответствующего атрибута. Например, декартово произведение $Q[YZ] = \{f, g\}\times\{a, c\}$ содержит следующее множество элементарных кортежей: $\{(f, a), (f, c), (g, a), (g, c)\}$.

C-кортежем называется кортеж, заданный в полном пространстве или в какой-либо его проекции, компонентами которого являются подмножества соответствующих доменов атрибутов. *C*-кортеж интерпретируется как декартово произведение этих компонент, т.е. как некоторое множество элементарных кортежей. Для обозначения *C*-кортежей используются прямые скобки.

Например, приведенные выше отношения R и Q можно представить как *C*-кортежи: $R[XYZ] = [\{b, d\}\ \{f, h\}\ \{a, b\}]$; $Q[YZ] = [\{f, g\}\ \{a, c\}]$.

C-кортеж, у которого имеется хотя бы одна пустая компонента, является пустым. В АК, если речь идет о моделях исчисления высказываний или предикатов, это высказывание принято в качестве аксиомы, для которой имеется соответствующая интерпретация, основанная на свойствах декартовых произведений.

Чтобы обобщить операции, которые часто приходится выполнять со структурами с разными схемами отношений, вводятся ***фиктивные компоненты***. Они бывают двух видов, одна из них используется в *C*-кортежах и обозначается "∗". Другую фиктивную компоненту (∅), которая используется в *D*-кортежах, мы рассмотрим позднее. Фиктивные компоненты "∗" обозначают множества, равные доменам соответствующих атрибутов, их можно вставить в соответствующий *C*-кортеж вместо отсутствующих атрибутов и тем самым ввести в него новые атрибуты.

Например, *C*-кортеж $Q[YZ] = [\{f, g\}\ \{a, c\}]$ можно представить в схеме отношения $[XYZ]$ в виде *C*-кортежа $[*\ \{f, g\}\ \{a, c\}]$, используя фиктивную компоненту. Поскольку фиктивная компонента в Q соответствует атрибуту X, то справедливо равенство

$$[*\ \{f, g\}\ \{a, c\}] = [\{a, b, c, d\}\ \{f, g\}\ \{a, c\}].$$

Пересечение однотипных *C*-кортежей осуществляется покомпонентно: результатом пересечения является *C*-кортеж, содержащий пересечения компонент исходных *C*-кортежей, относящихся к одному и тому же атрибуту, например

$$[\{b, d\}\ \{f, h\}\ \{a, b\}] \cap [*\ \{f, g\}\ \{a, c\}] = [\{b, d\}\ \{f\}\ \{a\}].$$

Результатом пересечения *C*-кортежей может оказаться пустое множество (пустой *C*-кортеж), например

$$[\ \{b, d\}\ \{f, h\}\ \{a, b\}] \cap [\ *\ \{g\}\ \{a, c\}] = \varnothing,$$

так как результатом пересечения вторых компонент этих *C*-кортежей является пустое множество.

Многие отношения, заданные как подмножества некоторого декартова произведения, не всегда удается отобразить с помощью единственного *C*-кортежа. Поэтому целесообразно ввести универсальную структуру, которая является объединением *C*-кортежей.

***C*-системой** называется структура, являющаяся объединением произвольного числа однотипных *C*-кортежей.

C-системы так же, как и *C*-кортежи, ограничиваются прямыми скобками. Например, для заданного выше пространства $\boldsymbol{S}$ можно определить некоторое отношение *P* как *C*-систему

$$P[XYZ] = \begin{bmatrix} \{a,d\} & * & \{b,c\} \\ \{b,d\} & \{f,h\} & \{a,c\} \\ \{b,c\} & \{g\} & \{b\} \end{bmatrix}.$$

Проверка включения одного *C*-кортежа (C_m) в другой (C_n) осуществляется покомпонентно: $C_m \subseteq C_n$ тогда и только тогда, когда все компоненты C_m включены в соответствующие компоненты C_n.

На основе свойств декартовых произведений можно определить условия, при которых объединение двух *C*-кортежей C_m и C_n может быть преобразовано в один *C*-кортеж. Этих условий всего два.

Условие (i): Если $C_m \subseteq C_n$, то $C_m \cup C_n = C_n$.

Условие (ii): Если C_m и C_n отличаются только в одной (*i*-ой) компоненте, то $C_m \cup C_n$ можно представить как один *C*-кортеж, у которого все компоненты не изменяются, за исключением *i*-ой, которая становится равной объединению соответствующих компонент из C_m и C_n.

Зная, как получить пересечение *C*-кортежей, мы можем сформулировать алгоритм для получения пересечения *C*-кортежа с *C*-системой и *C*-системы с *C*-системой. Для этого нужно представить *C*-кортежи как обычные множества, элементами которых являются однотипные элементарные кортежи. Тогда *C*-система, содержащая *C*-кортежи *A*, *B*, ..., *L*, является объединением этих множеств. Исходя из этого, используя законы алгебры множеств, в частности, законы дистрибутивности, нетрудно получить следующие алгоритмы вычисления пересечений соответствующих структур.

Алгоритм 1. Вычисление пересечения *C*-кортежа *P* с *C*-системой *Q*:

a) вычислить пересечения *C*-кортежа *P* с каждым *C*-кортежем из *Q*;
b) из полученных результатов исключить пустые *C*-кортежи;

c) из оставшихся кортежей сформировать *C*-систему;
d) Конец алгоритма.

Алгоритм 2. Вычисление пересечения *C*-системы *P* с *C*-системой *Q*:

a) вычислить пересечения каждого *C*-кортежа из *P* с каждым *C*-кортежем из *Q*;
b) из полученных результатов исключить пустые *C*-кортежи;
c) из оставшихся кортежей сформировать *C*-систему;
d) Конец алгоритма.

В качестве примера вычислим пересечение двух *C*-систем, заданных на определенном ранее пространстве ***S*** (это означает, что символ ∗ на второй позиции *C*-кортежей соответствует множеству $X_2 = \{f, g, h\}$):

$$P[XYZ] = \begin{bmatrix} \{a,b,d\} & \{f,h\} & \{b\} \\ \{b,c\} & * & \{a,c\} \end{bmatrix}, \; Q[XYZ] = \begin{bmatrix} \{a,d\} & * & \{b,c\} \\ \{b,d\} & \{f,h\} & \{a,c\} \\ \{b,c\} & \{g\} & \{b\} \end{bmatrix}.$$

1) вычисляем пересечение всех пар *C*-кортежей, содержащихся в разных *C*-системах:

$$[\{a, b, d\}\ \{f, h\}\ \{b\}] \cap [\ \{a, d\} * \{b, c\}] = [\ \{a, d\}\ \{f, h\}\ \{b\}];$$
$$[\ \{a, b, d\}\ \{f, h\}\ \{b\}] \cap [\ \{b, d\}\ \{f, h\}\ \{a, c\}] = \varnothing;$$
$$[\ \{a, b, d\}\ \{f, h\}\ \{b\}] \cap [\ \{b, c\}\ \{g\}\ \{b\}] = \varnothing;$$
$$[\ \{b, c\} * \{a, c\}] \cap [\ \{a, d\} * \{b, c\}] = \varnothing;$$
$$[\ \{b, c\} * \{a, c\}] \cap [\ \{b, d\}\ \{f, h\}\ \{a, c\}] = [\{b\}\ \{f, h\}\ \{a, c\}];$$
$$[\ \{b, c\} * \{a, c\}] \cap [\ \{b, c\}\ \{g\}\ \{b\}] = \varnothing.$$

2) из оставшихся непустых *C*-кортежей формируем *C*-систему:

$$P \cap Q = \begin{bmatrix} \{a,d\} & \{f,h\} & \{b\} \\ \{b\} & \{f,h\} & \{a,c\} \end{bmatrix}.$$

Даже этот сравнительно несложный пример показывает нам некоторые возможности уменьшения трудоемкости алгоритмов за счет использования алгебры кортежей: тот же результат можно получить, если предварительно перевести исходные *C*-системы в множества элементарных кортежей. Но при этом трудоемкость вычислений увеличится, так как *C*-система *P* содержит 24 элементарных кортежа, *C*-система *Q* – 20, а *C*-система $P \cap Q$ – 8.

Объединения C-кортежей и *C*-систем вычисляются намного проще. Для этого надо из объединяемых структур сформировать новую *C*-систему, содержащую все *C*-кортежи этих структур. После этого в отдельных случаях можно объединить некоторые *C*-кортежи. Необходимо только не забывать, что реализуемые алгоритмы объединения и пересечения, а также проверки включения структур АК имеют смысл только в тех случаях, когда эти структуры однотипны или преобразованы в однотипные структуры с помощью добавления фиктивных атрибутов.

Если требуется вычислить ***дополнение*** *C*-кортежа, то, используя традиционные методы теории многоместных отношений, необходимо выполнить следующие операции:

a) «Развернуть» (т.е. разложить на элементарные кортежи) *C*-кортеж *R* и соответствующий ему частный универсум ***S***;
b) Из «развернутого» ***S*** исключить поодиночке все элементарные кортежи, содержащиеся в *R*.

Ясно, что такая операция в общем случае весьма трудоемка, однако она существенно упрощается, если воспользоваться нижеследующими соотношения-

ми. Определим сначала, что такое *дополнение компоненты C-кортежа*. Если многоместное отношение определено в пространстве, каждый атрибут которого представлен некоторым множеством, то очевидно, что универсумом для компоненты *C*-кортежа является домен соответствующего ей атрибута (частный универсум), а дополнением компоненты является множество, содержащее все элементы этого частного универсума, не принадлежащие данной компоненте. Например, в пространстве $\boldsymbol{S} = X \times Y \times Z$ задан *C*-кортеж $R = [R_1\ R_2\ R_3]$. Тогда соответственно

$$\overline{R_1} = X \setminus R_1;\ \overline{R_2} = Y \setminus R_2;\ \overline{R_3} = Z \setminus R_3.$$

Следующая теорема основана на свойствах декартова произведения.

Теорема 1. Дополнением *C*-кортежа $T = [R_1\ R_2\ \ldots\ R_n]$ является *C*-система

$$C = \begin{bmatrix} \overline{R_1} & * & \ldots & * \\ * & \overline{R_2} & \ldots & * \\ \ldots & \ldots & \ldots & \ldots \\ * & * & \ldots & \overline{R_n} \end{bmatrix}$$

размерности $n \times n$, в которой каждая диагональная компонента является дополнением соответствующей компоненты *C*-кортежа *T*, а все остальные компоненты – фиктивные.

Доказательство теоремы 1. Для доказательства теоремы достаточно доказать, что (i) $C \cap T = \varnothing$ и (ii) $C \cup T = \boldsymbol{S}$.

Докажем (i). Пусть C_i – *C*-кортеж с номером *i* в *C*-системе *C*. В каждом C_i компонента с номером *i* является дополнением компоненты с тем же номером в *T*. Отсюда ясно, что пересечение любого *C*-кортежа из *C* с *C*-кортежем *T* является пустым. Отсюда следует, что $C \cap T = \varnothing$.

Докажем (ii). Для доказательства равенства $C \cup T = \boldsymbol{S}$ достаточно доказать, что любой элементарный кортеж $(a_1, a_2, \ldots, a_n)$ из $\boldsymbol{S}$, если он не принадлежит *T*, то обязательно принадлежит *C*. Ясно, что к элементарным кортежам, не принадлежащим *T*, относятся все элементарные кортежи из $\boldsymbol{S}$, у которых, по крайней мере, один элемент не содержится в соответствующей компоненте *C*-кортежа *T*. Допустим, что это элемент a_i. Следовательно, $a_i \in \overline{R_i}$. Тогда этот элементарный кортеж принадлежит *C*-кортежу $C_i = [* \ldots * \ \overline{R_i}\ * \ldots *]$, который содержится в *C*-системе *C*. Отсюда следует, что любой элементарный кортеж, не принадлежащий *C*-кортежу *T*, принадлежит его дополнению. *Конец доказательства*.

Рассмотрим пример. Пусть в рассмотренном выше пространстве $\boldsymbol{S} = X \times Y \times Z$ задан *C*-кортеж $T = [\{b, d\}\ \{f, h\}\ \{a, b\}]$. Тогда его дополнением в соответствии с теоремой 1 будет *C*-система

$$\overline{T} = \begin{bmatrix} X \setminus \{b,d\} & * & * \\ * & Y \setminus \{f,h\} & * \\ * & * & Z \setminus \{a,b\} \end{bmatrix} = \begin{bmatrix} \{a,c\} & * & * \\ * & \{g\} & * \\ * & * & \{c\} \end{bmatrix}.$$

C-системы, изображающие дополнение *C*-кортежа по теореме 1, можно, используя только диагональные компоненты, представить как один кортеж множеств, но при этом вместо обычных прямых скобок записать в качестве ограничителей *перевернутые* прямые скобки. Тогда получается равенство

$$\overline{T} = \begin{bmatrix} \{a,c\} & * & * \\ * & \{g\} & * \\ * & * & \{c\} \end{bmatrix} =]\{a,c\} \quad \{g\} \quad \{c\}[.$$

Такое «сокращенное» представление диагональной *C*-системы образует новую структуру АК, которая называется ***D-кортежем***. Оказывается, эта структура не только позволяет компактно отобразить диагональные *C*-системы, но самостоятельно используется в некоторых операциях и поисковых запросах. Термины «*C*-кортеж» и «*D*-кортеж» выбраны не случайно: в простейшем случае *C*-кортеж соответствует конъюнкции, а *D*-кортеж – дизъюнкции одноместных предикатов с разными переменными.

Используя *D*-кортежи, можно сформировать еще одну (пятую) структуру АК – *D*-систему.

D-система – структура, подобная матрице, в строках которой содержатся однотипные *D*-кортежи, которая интерпретируется как пересечение множеств элементарных кортежей, содержащихся в этих *D*-кортежах. Например, дополнение *C*-системы

$$F[XYZ] = \begin{bmatrix} \{a,b,d\} & \{f,h\} & \{b\} \\ \{b,c\} & * & \{a,c\} \end{bmatrix},$$

заданной в пространстве ***S***, можно изобразить как *D*-систему

$$\overline{F} = \begin{bmatrix} X \setminus \{a,b,d\} & Y \setminus \{f,h\} & Z \setminus \{b\} \\ X \setminus \{b,c\} & Y \setminus * & Z \setminus \{a,c\} \end{bmatrix} = \begin{bmatrix} \{c\} & \{g\} & \{a,c\} \\ \{a,d\} & \varnothing & \{b\} \end{bmatrix}.$$

Таким образом, *D*-система является *пересечением D*-кортежей. Нетрудно видеть, что соотношения между *C*-объектами (*C*-кортежи и *C*-системы) и *D*-объектами (*D*-кортеж и *D*-системы) соответствуют законам двойственности де Моргана. В силу этого они названы ***альтернативными классами***.

Полное соответствие между структурами АК и формулами *исчисления предикатов* легко доказуемо. Здесь мы рассмотрим лишь некоторые основные соответствия. В исчислении предикатов *C*-кортежу в тривиальном случае (когда отдельные атрибуты не соотносятся с многоместными отношениями) соответствует ***конъюнкция*** одноместных предикатов с разными переменными. Например, логической формуле

$$H = P_1(x) \wedge P_2(y) \wedge P_3(z)$$

соответствует *C*-кортеж $P[XYZ] = [P_1\ P_2\ P_3]$, где $P_1 \subseteq X$; $P_2 \subseteq Y$; $P_3 \subseteq Z$.

Отрицанию формулы *H* (***дизъюнкции*** одноместных предикатов)

$$\neg H = \neg P_1(x) \vee \neg P_2(y) \vee \neg P_3(z)$$

соответствует *D*-кортеж $\overline{P} =]\overline{P_1}\ \ \overline{P_2}\ \ \overline{P_3}[$.

Пустой АК-объект соответствует ***тождественно ложной формуле***.

АК-объект, равный некоторому частному универсуму, соответствует ***общезначимой формуле***.

Непустой АК-объект соответствует ***выполнимой формуле***.

Элементарный кортеж, входящий в состав непустого АК-объекта, соответствует ***выполняющей подстановке*** логической формулы.

2.2. Операции с атрибутами

Универсальный характер АК обусловлен тем, что в ней помимо операций алгебры множеств с АК-объектами вводятся еще три ***операции с атрибутами***:

a) *перестановка атрибутов* и соответствующих им столбцов матрицы АК-объекта;
b) *добавление нового фиктивного атрибута*;
c) *элиминация атрибута*.

Определенные комбинации операций с атрибутами и операций алгебры множеств позволяют сравнительно легко осуществлять обращение, композицию, соединение отношений, логический вывод и операции с кванторами.

Перестановка атрибутов – операция, при выполнении которой в матрице АК-объекта меняются местами столбцы, вместе с этим в схеме отношения меняются местами соответствующие атрибуты.

В общем случае (исключением являются бинарные отношения) при перестановке атрибутов содержание отношения, заданного АК-объектом, не изменяется. Эта операция необходима для того, чтобы привести АК-объекты с одинаковыми, но расположенными в разном порядке атрибутами к такому виду, при котором удобно выполнять операции алгебры множеств.

Например, *C*-система $P[XYZ] = \begin{bmatrix} \{a,b,d\} & \{f,h\} & \{b\} \\ \{b,c\} & * & \{a,c\} \end{bmatrix}$ при перестановке атрибутов может быть преобразована в *C*-систему

$$P[YXZ] = \begin{bmatrix} \{f,h\} & \{a,b,d\} & \{b\} \\ * & \{b,c\} & \{a,c\} \end{bmatrix}.$$

В случае *бинарных отношений* перестановка атрибутов позволяет получить отношение, *обратное* исходному. Например, отношение, заданное *C*-системой $G[XY] = \begin{bmatrix} \{a\} & \{a,b\} \\ \{b,c\} & \{a,c\} \end{bmatrix}$, при перестановке атрибутов преобразуется в обратное отношение $G^{-1}[XY] = G[YX] = \begin{bmatrix} \{a,b\} & \{a\} \\ \{a,c\} & \{b,c\} \end{bmatrix}$. В этом случае при *обращении* АК-объекта все элементарные кортежи (x, y) исходного отношения преобразуются в обратные – (y, x). Если элементарный кортеж содержит одинаковые элементы (например, (b, b)), то при обращении он не изменяется.

Добавление фиктивного атрибута осуществляется только в том случае, если добавляемый атрибут отсутствует в схеме отношения АК-объекта. При выполнении этой операции одновременно в схему отношения добавляется имя нового атрибута, а в структуру добавляется на соответствующем месте новый столбец с фиктивными компонентами, при этом в *C*-кортежи и в *C*-системы добавляется фиктивные компоненты "∗", а в *D*-кортежи и *D*-системы – фиктивные компоненты "∅".

Непосредственно операции алгебры множеств с АК-объектами допускаются лишь при условии, что эти АК-объекты сформированы в одной схеме отношения. Если же заданы АК-объекты с разными схемами отношений, то для выполнения этих операций их предварительно необходимо привести к одной схеме отношения с помощью добавлений фиктивных атрибутов. Рассмотрим семантику этой операции. Например, если АК-объект

$$G[XZ] = \begin{bmatrix} \{a,c\} & * \\ \{a,c,d\} & \{b,c\} \end{bmatrix}$$

соответствует формуле $F(x, z)$ исчисления предикатов, то, добавив в этот АК-объект фиктивный атрибут Y, получим АК-объект

$$G_1[XYZ] = \begin{bmatrix} \{a,c\} & * & * \\ \{a,c,d\} & * & \{b,c\} \end{bmatrix},$$

который соответствует формуле $\forall yF(x, z)$, полученной из формулы $F(x, z)$ по одному из правил вывода исчисления предикатов, которое называется ***правилом обобщения***. Таким образом, с точки зрения исчисления предикатов добавление в АК-объект фиктивных атрибутов означает то, что вновь полученный АК-объект является следствием исходного.

Элиминация атрибута осуществляется так: из АК-объекта удаляется столбец, а из его схемы отношения – соответствующий атрибут. Но в отличие от предыдущей операции логический смысл этой операции зависит от того, к какому классу объектов она применяется. Если элиминируется атрибут (например, X) из C-системы, то это означает, что к этому объекту применен квантор $\exists x$, а если этот атрибут элиминируется из D-системы, то это означает, что к данному объекту применен квантор $\forall x$. Например, если заданы C-система и ее дополнение, выраженное как D-система:

$$Q[XYZ] = \begin{bmatrix} \{a,b,d\} & \{f,h\} & \{b\} \\ \{b,c\} & * & \{a,c\} \end{bmatrix} \text{ и } \overline{Q}[XYZ] = \begin{bmatrix} \{c\} & \{g\} & \{a,c\} \\ \{a,d\} & \varnothing & \{b\} \end{bmatrix},$$

то $\exists x(Q[XYZ]) = \begin{bmatrix} \{f,h\} & \{b\} \\ * & \{a,c\} \end{bmatrix}$ и $\forall x(\overline{Q}[XYZ]) = \begin{bmatrix} \{g\} & \{a,c\} \\ \varnothing & \{b\} \end{bmatrix}$.

В качестве иллюстрации использования этих операций рассмотрим пример вычисления *композиции* и *соединения* отношений. Пусть заданы два отношения

$R_1[X_2, X_3] = \{(a, a), (a, b), (b, a), (b, c), (c, a), (c, c)\}$ и

$R_2[X_1, X_2] = \{(a, b), (a, c), (b, a), (c, a)\}$.

У R_1 и R_2 разные схемы отношения, но в них содержится общий атрибут X_2, что дает возможность вычислить рассмотренные ниже операции.

Соединение $(R_1 \oplus R_2)$ пары отношений R_1 и R_2 традиционно вычисляется методом сопоставления пар элементарных кортежей из разных отношений и выбора таких пар, у которых второй элемент пары из отношения R_2 совпадает с первым элементом пары из отношения R_1 (эти совпадающие элементы соответствуют одному и тому же атрибуту X_2). Для каждой из таких пар результатом соединения является элементарный кортеж из трех элементов, при этом первые два его элемента повторяют элементарный кортеж из R_2, а третий элемент – это второй элемент из кортежа, принадлежащего R_1. Например, для кортежа (a, b) из R_2 подходящими парами являются такие кортежи из R_1, у которых первым элементом является b (в R_1 такими кортежами являются (b, a) и (b, c)), а элементами соединения для них являются кортежи (a, b, a) и (a, b, c). В итоге после просмотра всех возможных пар кортежей из разных отношений, выбора подходящих пар и вычисления их соединений получим

$R_1 \oplus R_2 = \{(a, b, a), (a, c, a), (a, b, c), (a, c, c), (b, a, a), (b, a, b),$
$(c, a, a), (c, a, b)\}$.

Композиция $(R_1 \circ R_2)$ отношений может быть вычислена после вычисления соединения отношений. Для этого нужно их всех полученных в результате соединения трехместных элементарных кортежей удалить средний элемент и, если при этом появятся одинаковые кортежи, удалить лишние. В итоге для нашего примера получим:

$R_1 \circ R_2 = \{(a, a), (a, c), (b, a), (b, b), (c, a), (c, b)\}$.

Теперь рассмотрим, как вычисляются эти операции в АК. Сначала отношения R_1 и R_2 представим как C-системы:

$$R_1[X_2, X_3] = \begin{bmatrix} \{a\} & \{a,b\} \\ \{b,c\} & \{a,c\} \end{bmatrix}; \quad R_2[X_1, X_2] = \begin{bmatrix} \{a\} & \{b,c\} \\ \{b,c\} & \{a\} \end{bmatrix}.$$

Далее производим вычисления по следующему алгоритму:

1) Добавим фиктивный атрибут X_1 в R_1 и фиктивный атрибут X_3 в R_2. Тем самым мы приведем эти структуры к одной схеме отношения $[X_1, X_2, X_3]$:

$$R_1[X_1, X_2, X_3] = \begin{bmatrix} * & \{a\} & \{a,b\} \\ * & \{b,c\} & \{a,c\} \end{bmatrix}; R_2[X_1, X_2, X_3] = \begin{bmatrix} \{a\} & \{b,c\} & * \\ \{b,c\} & \{a\} & * \end{bmatrix}.$$

2) Используя рассмотренные выше алгоритмы, вычислим пересечение этих *C*-систем:

$$R_1 \cap R_2 = \begin{bmatrix} * & \{a\} & \{a,b\} \\ * & \{b,c\} & \{a,c\} \end{bmatrix} \cap \begin{bmatrix} \{a\} & \{b,c\} & * \\ \{b,c\} & \{a\} & * \end{bmatrix} =$$

$$= \begin{bmatrix} \{b,c\} & \{a\} & \{a,b\} \\ \{a\} & \{b,c\} & \{a,c\} \end{bmatrix}.$$

3) Проверяем отсутствие пустых *C*-кортежей (если они имеются, то их необходимо удалить), после чего элиминируем атрибут X_2:

$$R_1 \circ R_2 = \begin{bmatrix} \{b,c\} & \{a,b\} \\ \{a\} & \{a,c\} \end{bmatrix}.$$

На шаге 2 мы получили соединение отношений R_1 и R_2, а на шаге 3 – их композицию.

Рассмотрим семантику этих операций в терминах математической логики. Пусть отношения R_1 и R_2 соответствуют предикатам $R_1(x_2, x_3)$ и $R_2(x_1, x_2)$. Если задана формула $R_1(x_2, x_3) \wedge R_2(x_1, x_2)$, то, применив к каждому из предикатов правило обобщения по отсутствующей переменной, получим формулу

$$\forall x_1(R_1(x_2, x_3)) \wedge \forall x_1(R_2(x_1, x_2)),$$

что соответствует соединению отношений. Если же к этой формуле применить квантор существования по переменной x_2, то получим формулу

$$\exists x_2(\forall x_1(R_1(x_2, x_3)) \wedge \forall x_1(R_2(x_1, x_2))),$$

которая соответствует композиции отношений.

Эта же семантика используется и в АК с использованием операций с атрибутами.

2.3. Техника вычислений в АК

2.3.1. Введение. АК-объекты представляют собой кортежи или матрицы компонент. При выполнении операций с ними и распознавании соотношений между ними (включения или равенства) основная вычислительная нагрузка ложится на операции с компонентами, которые являются подмножествами соответствующих доменов атрибутов. Таким образом, алгоритмы вычислений и сравнений АК-объектов сводятся к последовательностям операций или сравнений алгебры множеств. Эти последовательности операций определены с помощью схем отношений структур АК, потому операции с компонентами, принадлежащими разным атрибутам, не допустимы.

В предыдущих разделах были рассмотрены операции и сравнения лишь для некоторых сочетаний типов АК-объектов: пересечение, объединение и дополнение *C*-кортежей и *C*-систем, проверка включения *C*-кортежа в *C*-кортеж. Для того чтобы охватить этими операциями и сравнениями все возможные сочетания типов АК-объектов, требуются дополнительные соотношения. Здесь приводится без доказательств перечень всех необходимых соотношений. Их доказа-

тельства основаны на известных законах алгебры множеств. Предполагается, что используемые в операциях и сравнениях АК-объекты приведены к одной схеме отношений.

2.3.2. Вычисление дополнений. Дополнение произвольного АК-объекта P эквивалентно АК-объекту Q альтернативного класса и той же размерности, в котором каждая компонента q_{ij} равна дополнению соответствующей компоненты p_{ij} в P (аналог закона де Моргана).

2.3.3. Распознавание пустых множеств и частных универсумов.

- C-кортежи с хотя бы одной пустой компонентой и D-кортежи со всеми пустыми компонентами эквивалентны пустому множеству;
- D-кортежи с хотя бы одной полной компонентой (∗) и С-кортежи со всеми полными компонентами эквивалентны универсуму.

2.3.4. Пересечение.

- пересечение C-кортежей $[p_1\ p_2\ \ldots\ p_n]$ и $[q_1\ q_2\ \ldots\ q_n]$ эквивалентно C-кортежу $[r_1\ r_2\ \ldots\ r_n]$, где $r_i = p_i \cap q_i$ для всех $i = \{1, 2, ..., n\}$;
- пересечение C-систем P и Q эквивалентно C-системе, состоящей из всех непустых C-кортежей, образующихся при выполнении операций пересечения каждого C-кортежа из P с каждым C-кортежем из Q;
- пересечение множества D-кортежей или D-систем эквивалентно D-системе, содержащей все множества D-кортежей исходных АК-объектов.

2.3.5. Объединение.

- объединение D-кортежей $]p_1\ p_2\ \ldots\ p_n[$ и $]q_1\ q_2\ \ldots\ q_n[$ эквивалентно D-кортежу $]r_1\ r_2\ \ldots\ r_n[$, где $r_i = p_i \cup q_i$ для всех $i = \{1, 2, ... , n\}$;
- объединение D-систем P и Q эквивалентно D-системе, состоящей из всех не равных универсуму D-кортежей, образующихся при выполнении операций объединения каждого D-кортежа из P с каждым D-кортежем из Q;
- объединение множества C-кортежей или C-систем эквивалентно C-системе, содержащей все множества C-кортежей исходных АК-объектов.

2.3.6. Критерии включения одного АК-объекта в другой.

- для пары C-кортежей $P = [p_1\ p_2\ \ldots\ p_n]$ и $Q = [q_1\ q_2\ \ldots\ q_n]$ или D-кортежей $P =]p_1\ p_2\ \ldots\ p_n[$ и $Q =]q_1\ q_2\ \ldots\ q_n[$ справедливо $P \subseteq Q$, если и только если для каждого $i = \{1, 2, ..., n\}$ соблюдается $p_i \subseteq q_i$;
- для C-системы P и C-кортежа Q справедливо $P \subseteq Q$, если и только если для каждого C-кортежа P_i, содержащегося в P, выполняется $P_i \subseteq Q$;
- для C-кортежа $P = [p_1\ p_2\ \ldots\ p_n]$ и D-кортежа $Q =]q_1\ q_2\ \ldots\ q_n[$ справедливо $P \subseteq Q$, если и только если по крайней мере для одного i соблюдается $p_i \subseteq q_i$;
- для C-кортежа (или D-кортежа) P и D-системы Q справедливо $P \subseteq Q$, если и только если для каждого D-кортежа Q_j из Q выполняется $P \subseteq Q_j$;
- для C-системы P и D-системы Q справедливо $P \subseteq Q$, если и только если каждый C-кортеж из P включен в каждый D-кортеж из Q;
- два АК-объекта P и Q эквивалентны, если выполняется как $P \subseteq Q$, так и $Q \subseteq P$.

2.3.7. Преобразование АК-объектов в альтернативные классы. Приведенные выше соотношения не охватывают всех возможных сочетаний типов АК-объектов. Например, нет соотношений, позволяющих вычислить пересечения или объединения АК-объектов, относящихся к разным альтернативным классам, или соотношений, позволяющих проверить включение C-кортежа в C-систему. Поэтому необходимо иметь возможность преобразовывать АК-объекты, относящиеся к одному альтернативному классу, в эквивалентные им

АК-объекты другого альтернативного класса. Для этого предусматриваются следующие алгоритмы:

- каждый *C*-кортеж (*D*-кортеж) *P* преобразуется в эквивалентную ему *D*-систему (*C*-систему), в которой каждая нефиктивная компонента p_i, соответствующая атрибуту X_i исходного кортежа, представлена *D*-кортежем (*C*-кортежем), состоящим из компоненты p_i в атрибуте X_i и фиктивных компонент во всех остальных атрибутах;
- чтобы преобразовать *D*-систему *P* в эквивалентную ей *C*-систему, необходимо преобразовать в *C*-систему каждый *D*-кортеж из *P*, после чего вычислить пересечение всех полученных *C*-систем;
- чтобы преобразовать *C*-систему *P* в эквивалентную ей *D*-систему, необходимо преобразовать в *D*-систему каждый *C*-кортеж из *P*, после чего вычислить объединение всех полученных *D*-систем.

Рассмотрим примеры, иллюстрирующие приведенные выше алгоритмы.

Преобразование D-кортежа в C-систему:

$$\rceil\{a,c\}\ \ \{d\}\ \ \varnothing\ \{g,h\}\lceil = \begin{bmatrix} \{a,c\} & * & * & * \\ * & \{d\} & * & * \\ * & * & * & \{g,h\} \end{bmatrix}.$$

Преобразование D-системы в C-систему:

$$\begin{matrix} \rceil\{g\} & \{a,c\} \\ \{f,h\} & \{b\} \end{matrix}\lceil = \begin{bmatrix} \{g\} & * \\ * & \{a,c\} \end{bmatrix} \cap \begin{bmatrix} \{f,h\} & * \\ * & \{b\} \end{bmatrix} = \begin{bmatrix} \{g\} & \{b\} \\ \{f,h\} & \{a,c\} \end{bmatrix}.$$

Заметим, что операции преобразования *C*-систем в *D*-системы и *D*-систем в *C*-системы в отличие от всех рассмотренных ранее операций характеризуются экспоненциальной вычислительной сложностью. В частности, классическая задача «выполнимость КНФ», которая относится к *NP*-полным задачам [9], также характеризуется экспоненциальной вычислительной сложностью. В структурах АК алгоритм решения этой задачи реализуется как начальный этап алгоритма преобразования *D*-системы в *C*-систему. В АК разработаны методы, позволяющие уменьшить трудоемкость вычислений при реализации этих алгоритмов, а в некоторых случаях довести вычислительную сложность до полиномиальной.

2.3.8. Ортогонализация. Ортогональной называются *C*-система, у которой пересечение каждой пары содержащихся в ней *C*-кортежей равно пустому множеству. Ортогонализация необходима, в частности, при вероятностном анализе АК-объектов (см. ниже). Исследования показали, что при реализации алгоритма преобразования *D*-системы в *C*-систему использование ортогонализации позволяет существенно уменьшить требуемые вычислительные ресурсы. Методы ортогонализации широко используются в логико-вероятностных методах анализа надежности и безопасности сложных систем [10].

В основе методов ортогонализации лежит следующее соотношение, полученное для формул исчисления высказываний П.С. Порецким:

(1) $$H_1 \vee H_2 \vee \ldots \vee H_k = (H_1) \vee (\overline{H_1} \wedge H_2) \vee \ldots \vee (\overline{H_1} \wedge \overline{H_2} \wedge \ldots \wedge \overline{H_{k-1}} \wedge H_k).$$

Рассмотрим ортогонализацию при преобразовании *D*-системы в С-систему. Здесь первый этап алгоритма (преобразование *D*-кортежа в *C*-систему по Теореме 1) выполняется в соответствии с нижеследующей теоремой.

Теорема 2. *D*-кортеж вида $]Q_1\ Q_2\ \ldots\ Q_{m\text{-}1}\ Q_m[$ преобразуется в эквивалентную ему ортогональную *C*-систему: $\begin{bmatrix} Q_1 & * & \ldots & * & * \\ \overline{Q_1} & Q_2 & \ldots & * & * \\ & & \ldots & & \\ \overline{Q_1} & \overline{Q_2} & \ldots & Q_{m-1} & * \\ \overline{Q_1} & \overline{Q_2} & \ldots & \overline{Q_{m-1}} & Q_m \end{bmatrix}$.

Доказательство теоремы 2. Нетрудно убедиться, что *C*-система ортогональна, так как каждый ее *C*-кортеж T_i (i = 1, 2, …, m) имеет в i-м атрибуте компоненту Q_i, а все последующие *C*-кортежи содержат в этом же атрибуте дополнение этой компоненты. Равенство *D*-кортежа *C*-системе справедливо в силу (1). *Конец доказательства*.

Рассмотрим пример. Пусть в схеме отношения [*XYZ*], где $X = Y = Z = \{a, b, c, d\}$, задан *D*-кортеж $]\{a, c\}\ \{d\}\ \{b, d\}[$. Тогда в соответствии с теоремами 1 и 2 получим следующие равенства:

$$]\{a, c\}\ \{d\}\ \{b, d\}[= \begin{bmatrix} \{a,c\} & * & * \\ * & \{d\} & * \\ * & * & \{b,d\} \end{bmatrix} = \begin{bmatrix} \{a,c\} & * & * \\ \{b,d\} & \{d\} & * \\ \{b,d\} & \{a,b,c\} & \{b,d\} \end{bmatrix},$$

причем вторая *C*-система является ортогональной.

При реализации алгоритма преобразования *D*-системы в *C*-систему, если первый этап выполнен в соответствии с теоремой 2, то оказывается, что необходимо вычислить пересечение ортогональных *C*-систем. В результате этого пересечения получается ортогональная С-система. Этот результат основан на следующей теореме.

Теорема 3. Если *P* и *Q* – ортогональные *C*-системы, то их пересечение либо пусто, либо состоит из одного С-кортежа, либо является ортогональной *C*-системой .

Доказательство. Каждую из исходных *C*-систем можно представить как системы множеств $\{T(P_i)\}$ и $\{T(Q_j)\}$ элементарных кортежей, содержащихся в соответствующих *C*-кортежах. В силу ортогональности *C*-систем *P* и *Q* системы множеств $\{T(P_i)\}$ и $\{T(Q_j)\}$ являются разбиениями. При пересечении разбиений возможны три варианта: пустое множество, единственное множество или система множеств, являющаяся разбиением. Из этого следует утверждение теоремы. *Конец доказательства*.

Рассмотрим еще одно важное соотношение, позволяющее во многих случаях существенно сократить перебор при преобразовании АК-объектов в альтернативные классы. Дело в том, что *D*-кортеж, содержащий не менее двух непустых компонент, можно представить как ортогональную *C*-систему не единственным способом. Например, имеем равенства

$$]A\ B\ C[= \begin{bmatrix} A & * & * \\ * & B & * \\ * & * & C \end{bmatrix} = \begin{bmatrix} A & * & * \\ \overline{A} & B & * \\ \overline{A} & \overline{B} & C \end{bmatrix}.$$

Если в промежуточном результате переставить местами *C*-кортежи, то эквивалентность не нарушится, но при ортогонализации после перестановки мы получим другое изображение того же *D*-кортежа. В результате можно получить следующие равенства:

$$]A\ B\ C[= \begin{bmatrix} A & * & * \\ * & B & * \\ * & * & C \end{bmatrix} = \begin{bmatrix} * & B & * \\ * & * & C \\ A & * & * \end{bmatrix} = \begin{bmatrix} * & B & * \\ * & \overline{B} & C \\ A & \overline{B} & \overline{C} \end{bmatrix} = \begin{bmatrix} * & * & C \\ A & * & * \\ * & B & * \end{bmatrix} =$$

$$\begin{bmatrix} * & * & C \\ A & * & \overline{C} \\ \overline{A} & B & \overline{C} \end{bmatrix} = \ldots$$

Если целенаправленно использовать эти варианты преобразования, то можно существенно сократить как размерность вычисляемой *C*-системы за счет сокращения числа входящих в нее *C*-кортежей, так и трудоемкость вычислений за счет того, что в промежуточных вычислениях образуется значительно больше пустых *C*-кортежей, которые не используются в последующих вычислениях. Особенно этот прием эффективен при преобразовании *D*-систем в *C*-системы в рамках исчисления высказываний, но и для более широкого класса АК-объектов он также позволяет иногда получить существенное уменьшение трудоемкости. Рассмотрим пример ортогонализации *D*-системы

$$P[XYZ] = \begin{bmatrix} \{a,c\} & \{d\} & \{b,d\} \\ \varnothing & \{a,d\} & \{a,c\} \\ \{b,c\} & \varnothing & \{b\} \end{bmatrix},$$

заданной в универсуме $X \times Y \times Z = \{a,\ b,\ c,\ d\} \times \{f, g, h\} \times \{a,\ b,\ c\}$. Используя для каждого *D*-кортежа теорему 2 и перестановку *C*-кортежей, получим:

$$P = \begin{bmatrix} * & * & \{b,d\} \\ \{a,c & * & \{a,c\} \\ \{b,d\} & \{d\} & \{a,c\} \end{bmatrix} \cap \begin{bmatrix} * & * & \{a,c\} \\ * & \{a,d\} & \{b,d\} \end{bmatrix} \cap \begin{bmatrix} \{b,c\} & * & * \\ \{a,d\} & * & \{b\} \end{bmatrix}.$$

Для первого пересечения получим:

$$\begin{bmatrix} * & * & \{b,d\} \\ \{a,c & * & \{a,c\} \\ \{b,d\} & \{d\} & \{a,c\} \end{bmatrix} \cap \begin{bmatrix} * & * & \{a,c\} \\ * & \{a,d\} & \{b,d\} \end{bmatrix} = \begin{bmatrix} * & \{a,d\} & \{b,d\} \\ \{a,c\} & * & \{a,c\} \\ \{b,d\} & \{d\} & \{a,c\} \end{bmatrix}.$$

Для второго:

$$\begin{bmatrix} * & \{a,d\} & \{b,d\} \\ \{a,c\} & * & \{a,c\} \\ \{b,d\} & \{d\} & \{a,c\} \end{bmatrix} \cap \begin{bmatrix} \{b,c\} & * & * \\ \{a,d\} & * & \{b\} \end{bmatrix} = \begin{bmatrix} \{b,c\} & \{a,d\} & \{b,d\} \\ \{a,d\} & \{a,d\} & \{b\} \\ \{c\} & * & \{a,c\} \\ \{b\} & \{d\} & \{a,c\} \end{bmatrix}.$$

Нетрудно убедиться, что полученная *C*-система, содержащая 4 *C*-кортежа, ортогональна.

2.3.9. Экономия ресурсов при машинной реализации структур АК. Вычислительная сложность операций алгебры множеств и проверок включения зависит от того, к какому классу структур относятся используемые при этом АК-объекты. Например, проверка включения *C*-кортежа в *C*-систему выполняется в общем случае с помощью алгоритма экспоненциальной вычислительной сложности, в то время как алгоритм проверки включения *C*-кортежа и даже *C*-системы в *D*-систему имеет полиномиальную сложность. Для выполнения некоторых операций и проверок требуется преобразование АК-объекта в эквивалентный ему АК-объект альтернативного класса (например, преобразование *C*-системы в *D*-систему или наоборот), что для *C*-систем и *D*-систем выполняется

с помощью алгоритмов экспоненциальной сложности. Операция дополнения АК-объекта во всех случаях выполняется алгоритмом полиномиальной вычислительной сложности, но при этом система преобразуется в альтернативный класс. Операции пересечения и объединения АК-объектов, относящихся к одному классу, выполняются алгоритмами полиномиальной сложности, но, если они относятся к разным классам, то для выполнения этих операций требуется преобразование одного из них в другой класс.

В задачах, которые соответствуют в логике задачам дедуктивного вывода, часто требуется проверка включения одного АК-объекта в другой, а также выполнение кванторных операций. В таблице 1 приведены различные сочетания АК-объектов и знаком «+» помечены сочетания, для которых алгоритмы выполнения соответствующих операций являются полиномиальными при условии, что все домены атрибутов являются простыми множествами (т.е. не многоместными отношениями). При этом во всех случаях проверка того, содержится ли заданный элементарный кортеж в любой структуре, выполняется алгоритмом полиномиальной сложности.

Таблица 1. Операции АК, выполняемые с полиномиальной вычислительной сложностью.

Действие	*C*-кортеж	*C*-система	*D*-кортеж	*D*-система
Проверка включения С-кортежа в	+		+	+
Проверка включения С-системы в	+		+	+
Проверка включения D-кортежа в	+		+	+
Проверка включения D-системы в				
Кванторная операция ∀x	+		+	+
Кванторная операция ∃x	+	+	+	

Однако в рамках АК разработаны методы уменьшения трудоемкости экспоненциальных по вычислительной сложности алгоритмов, а также методы распознавания частных случаев структур, которые позволяют выполнить соответствующие операции, преобразования и проверки за полиномиальное время. Но даже в тех случаях, когда нет возможности использовать алгоритм полиномиальной вычислительной сложности, уменьшение требуемых вычислительных ресурсов достигается за счет использования присущего АК-объектам естественного параллелизма.

В отличие от традиционных структур данных, применяющихся при машинной реализации логического и логико-вероятностного анализа систем, структуры АК являются *матрицеподобными*, что позволяет при соответствующей программно-аппаратной реализации сравнительно легко уменьшить требуемые вычислительные ресурсы с помощью распараллеливания операций.

Машинная реализация АК-объектов позволяет использовать три уровня параллелизма: 1) на уровне компонент; 2) на уровне строк и 3) на уровне матриц. *На уровне компонент* можно представить домены и их подмножества в виде совокупности логических векторов и для реализации операций алгебры множеств и проверок включения использовать логические операции с целыми векторами. *На уровне строк* можно одновременно осуществлять операции или проверки включения со всеми парами компонент *C*-кортежей или *D*-кортежей. *На уровне матриц* можно одновременно выполнять операции алгебры множеств и проверки включения для множества пар, элементами которых являются строки (*C*-кортежи или *D*-кортежи) из разных АК-объектов. Например, при вычислении

пересечения *C*-системы с *C*-системой все используемые в этом алгоритме операции пересечения *C*-кортежей можно выполнять параллельно. Распараллеливание операций 2-го и 3-го уровней можно осуществить с помощью вычислительных комплексов параллельной обработки данных. Возможная архитектура подобных комплексов рассматривается в [11].

3. Представление используемых структур БД и БЗ структурами данных АК

3.1. Графы и сети

Обычно графы и сети представлены в компьютерах как списковые структуры [12]. В системах искусственного интеллекта логический вывод на графах и семантических сетях реализуется как алгоритм поиска достижимых вершин или построения транзитивного замыкания графа. Такие алгоритмы для списковых структур стали привычными. В то же время они недостаточно эффективны и плохо поддаются распараллеливанию, но их применение в системах искусственного интеллекта обусловлено тем, что многие другие структуры БЗ тоже представлены списковыми структурами и на основе этого реализуется определенная унификация данных. Рассмотрим, как представлены графы в структурах АК. В качестве примера используем граф, изображенный на рис. 1.

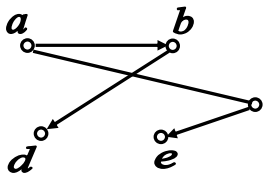

Рис. 1. Пример графа.

Этот граф можно представить как *C*-систему

$$G[XY] = \begin{bmatrix} \{a\} & \{b,c\} \\ \{b\} & \{d\} \\ \{c\} & \{a,e\} \end{bmatrix},$$

которая изоморфна матрице смежности этого графа. При этом схема отношения содержит атрибуты одного сорта.

Нетрудно убедиться, что если произвольный граф представить как *C*-систему, то справедливы следующие соотношения, которые легко реализуются и проверяются в структурах АК:

- у неориентированных графов $G = G^{-1}$;
- у ориентированных графов $G \cap G^{-1} = \varnothing$ и $G^{-1} \subseteq \overline{G}$;
- у смешанных графов $G \cap G^{-1} \neq \varnothing$ и $G \neq G^{-1}$.

На графах часто используется композиция $G \circ G$, т.е. композиция графа с самим собой. Сокращенно эта операция обозначается G^2. Используется и более высокая «степень» композиции, например $G^3 = G \circ G \circ G$ и т.д.

Вершина *y* графа *G* называется ***достижимой*** из вершины *x*, если в *G* существует путь из вершины *x* в вершину *y*. Часто бывает необходимо определить

для каждой вершины графа G множество всех достижимых из нее вершин. Информация об этом содержится в транзитивном замыкании графа, которое определяется следующим образом.

Транзитивным замыканием графа G, содержащего n вершин, называется граф G^+, у которого каждая вершина соединена дугой со всеми достижимыми вершинами.

Транзитивное замыкание можно построить с помощью следующей последовательности операций:

$$G^+ = G \cup G^2 \cup G^3 \cup \ldots \cup G^k, \text{ где}$$
$$k \le \lfloor log_2(n\text{-}2) \rfloor + 2.$$

Например, для графа, содержащего 60 вершин, для построения транзитивного замыкания достаточно, чтобы k не превышало 7. Практически во всех случаях преобразования конечного графа G в граф G^+ эта операция заканчивается прежде, чем будет получено последняя «степень» G^k. Наихудший случай, когда $k = \lfloor log_2(n\text{-}2) \rfloor + 2$, соответствует графу G, представляющего собой простую цепь вида $a_1 \to a_2 \to \ldots \to a_{n\text{-}1} \to a_n$. В других случаях критерием более раннего завершения этой операции является отсутствие в очередной «степени» графа не встречавшихся ранее дуг.

Рассмотрим вычисление транзитивного замыкания графа G с использованием структур АК.

Вычислим G^2. Для этого переименуем атрибуты, чтобы привести схему отношения к виду, соответствующему вычислению композиции. Тогда получим:

$$G[YZ] \circ G[XY] = \begin{bmatrix} * & \{a\} & \{b,c\} \\ * & \{b\} & \{d\} \\ * & \{c\} & \{a,e\} \end{bmatrix} \cap \begin{bmatrix} \{a\} & \{b,c\} & * \\ \{b\} & \{d\} & * \\ \{c\} & \{a,e\} & * \end{bmatrix}$$
$$= \begin{bmatrix} \{c\} & \{a\} & \{b,c\} \\ \{a\} & \{b\} & \{d\} \\ \{a\} & \{c\} & \{a,e\} \end{bmatrix}.$$

После элиминации атрибута Y и объединения C-кортежей с одинаковыми левыми компонентами получим $G^2 = \begin{bmatrix} \{a\} & \{a,d,e\} \\ \{c\} & \{b,c\} \end{bmatrix}$. Поскольку к вершинам a и c добавились новые связи, продолжим вычисление степеней. Вычислим $G^3 = G \circ G^2$.

$$\begin{bmatrix} * & \{a\} & \{b,c\} \\ * & \{b\} & \{d\} \\ * & \{c\} & \{a,e\} \end{bmatrix} \cap \begin{bmatrix} \{a\} & \{a,d,e\} & * \\ \{c\} & \{b,c\} & * \end{bmatrix} = \begin{bmatrix} \{a\} & \{a\} & \{b,c\} \\ \{c\} & \{b\} & \{d\} \\ \{c\} & \{c\} & \{a,e\} \end{bmatrix}.$$

После элиминации среднего атрибута получим $G^3 = \begin{bmatrix} \{a\} & \{b,c\} \\ \{c\} & \{a,d,e\} \end{bmatrix}$. При этом добавилась новая связь (c, d), поэтому для проверки надо вычислить $G^4 = G \circ G^3$. Убедившись, что в G^4 новых связей нет, можно вычислить транзитивное замыкание:

$$G^+ = G \cup G^2 \cup G^3 = \begin{bmatrix} \{a\} & \{b,c\} \\ \{b\} & \{d\} \\ \{c\} & \{a,e\} \end{bmatrix} \cup \begin{bmatrix} \{a\} & \{a,d,e\} \\ \{c\} & \{b,c\} \end{bmatrix} \cup$$

$$\cup \begin{bmatrix} \{a\} & \{b,c\} \\ \{c\} & \{a,d,e\} \end{bmatrix} = \begin{bmatrix} \{a\} & * \\ \{b\} & \{d\} \\ \{c\} & * \end{bmatrix}.$$

3.2. Дедуктивные базы данных

В реляционных БД основными объектами управления являются файлы, организованные в виде таблиц. С точки зрения АК эти таблицы состоят из множества элементарных кортежей, содержащихся в данной таблице (отношении). Можно также представить такую таблицу как *C*-систему, в которой все компоненты являются одноэлементными множествами. Но в ряде случаев такое представление не самое эффективное. Если таблицы имеют большой объем и не поддаются «сжатию» за счет объединения элементарных кортежей в *C*-кортежи, то для эффективного поиска информации в них, видимо, целесообразно воспользоваться средствами индексации и поиском по ключу, предусмотренными в современных СУБД.

Однако при использовании в СУБД средств логического анализа систем в них появляются и во многих случаях преобладают структуры (к ним относятся сети, правила логического вывода, предикаты и т.д.), которые целесообразно моделировать другими типами АК-объектов. В связи с этим появляется необходимость в том, чтобы поисковые запросы к таким структурам имели форму, сопоставимую со структурами АК. Теоретически эта задача в АК решается просто: надо сформулировать запрос в виде АК-объекта, при этом атрибуты в нем делятся на два типа. К первому типу относятся атрибуты с заданными значениями (или диапазонами значений), а ко второму типу – атрибуты, значения которых надо установить в процессе поиска. При формулировке запроса атрибуты, значения которых надо установить, вводятся как фиктивные. Тогда при пересечении АК-объекта, выражающего запрос, с соответствующими структурами БД будет получена структура, в которой на месте искомых атрибутов появляются ответы на заданные вопросы.

Рассмотрим пример. Пусть в БД используются АК-объекты

$$R_1[X_2, X_3] = \begin{bmatrix} \{d\} & \{g,h\} \\ \{e,f\} & \{g,i\} \end{bmatrix} \text{ и } R_2[X_1, X_2] = \begin{bmatrix} \{a\} & \{e,f\} \\ \{b,c\} & \{d\} \end{bmatrix}.$$

Будем считать, что атрибуту X_1 соответствует некоторое событие, атрибуту X_2 – место, где это событие произошло, а атрибуту X_3 – время появления этого события (алфавитный порядок в значениях этого атрибута соответствует временному порядку). Рассмотрим несколько возможных запросов и технику их реализации в структурах АК.

1) Q_1: Где произошло событие b?

Формулировка в АК: $Q_1[X_1, X_2] = [\{b\} \quad *]$.

Реализация: $Q_1[X_1, X_2] \cap R_2[X_1, X_2] = [\{b\} \quad \{d\}]$.

Ответ: местом, где произошло событие b, является d.

2) Q_2: В каких местах произошли события до момента времени i?

Формулировка в АК: $Q_2[X_2, X_3] = [* \quad \{g, h\}]$.

$$\textit{Реализация}: Q_2[X_2, X_3] \cap R_1[X_2, X_3] = \begin{bmatrix} \{d\} & \{g,h\} \\ \{e,f\} & \{g\} \end{bmatrix}.$$

Ответ: в моменты времени *g* и *h* события произошли в *d*, в момент времени *g* события произошли в местах *e* и *f*.

3) Q_3: Где и какие события произошли в период времени *h*?

$$\textit{Формулировка в АК}: Q_3[X_1, X_2, X_3] = [* \quad * \quad \{h\}].$$

$$\textit{Реализация}: Q_3[X_1, X_2, X_3] \cap (R_1 \oplus R_2) = [\{b, c\} \quad \{d\} \quad \{h\}].$$

Ответ: в момент времени *h* на месте *d* произошли события *b* и *c*.

При ответе на вопрос Q_3 потребовалось вначале вычислить соединение отношений R_1 и R_2. Это объясняется тем, что неизвестный атрибут «событие» (X_1) не содержится в той структуре, где атрибутом является «время» (X_3). Иногда выполнение операции соединения отношений требует больших объемов вычислений. В этом случае более эффективной будет следующая последовательность вычислений. Пусть сложный запрос имеет схему отношения, которая включает в себя схемы отношений двух или более структур. В нашем примере схема отношения запроса $Q_3[X_1, X_2, X_3]$ включает в себя схему отношения $[X_1, X_2]$ структуры R_2 и схему отношения $[X_2, X_3]$ структуры R_1. Тогда поиск выполняется в следующем порядке.

Шаг 1. В запросе выделяется проекция, схема отношения которой соответствует одному из объектов поиска. В нашем примере мы из запроса Q_3 выбираем проекцию $[X_2, X_3]$, которая соответствует схеме отношения структуры R_1. Получаем $Q_3^1[X_2, X_3] = [* \quad \{h\}]$.

Шаг 2. Находим пересечение этой проекции с соответствующей структурой:

$$Q_3^1[X_2, X_3] \cap R_1[X_2, X_3] = [* \quad \{h\}] \cap \begin{bmatrix} \{d\} & \{g,h\} \\ \{e,f\} & \{g,i\} \end{bmatrix} = [\{d\} \quad \{h\}].$$

Шаг 3. Из запроса выделяется другая проекция, которая соответствует другому объекту поиска. В нашем примере мы из запроса Q_3 выбираем проекцию $[X_1, X_2]$, которая соответствует схеме отношения структуры R_2. Получаем $Q_3^2[X_1, X_2] = [* \quad *]$.

Шаг 4. Находим пересечение этой проекции с соответствующей структурой. Здесь у нас отношение $R_2[X_1, X_2]$ не изменится, так как С-кортеж $Q_3^2[X_1, X_2]$ содержит только фиктивные компоненты.

Шаг 5. Находим соединение отношений, полученных на шагах 2 и 4.

$$[* \quad \{d\} \quad \{h\}] \cap \begin{bmatrix} \{a\} & \{e,f\} & * \\ \{b,c\} & \{d\} & * \end{bmatrix} = [\{b, c\} \quad \{d\} \quad \{h\}].$$

Получили тот же ответ. Если исходные структуры имеют большой объем, то экономия вычислительных ресурсов при использовании такой техники вычислений может быть значительной. Из примеров видно, что многие запросы могут быть сформулированы как *C*-кортежи. В более сложных случаях для формулировки запроса на языке АК потребуются более сложные структуры. Рассмотрим такой запрос:

Q_4: В каких местах произошло событие *a* независимо от времени или событие *b*, произошедшее в период времени *i*?

В этом случае для формулировки запроса необходимо использовать C-систему $Q_4[X_1, X_2, X_3] = \begin{bmatrix} \{a\} & * & * \\ \{b\} & * & \{i\} \end{bmatrix}$. Если выполним вычисления по одному из рассмотренных выше способов, то получим ответ $[\{a\}\ \{e, f\}\ \{h, g\}]$, из чего следует, что событие a происходило в местах e и f, а событие b при заданных условиях не происходило.

В общем случае можно выполнять операции соединения и композиции отношений для любых пар АК-объектов, у которых различаются схемы отношений. Пусть заданы две структуры $R_1[\boldsymbol{V}]$ и $R_2[\boldsymbol{W}]$, при этом $\boldsymbol{V}$ и $\boldsymbol{W}$ являются множествами атрибутов и $\boldsymbol{V} \neq \boldsymbol{W}$. Эти множества можно разложить на такие непересекающиеся подмножества $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, которые содержат следующие атрибуты: $\boldsymbol{X}$ – множество всех атрибутов, содержащихся в схеме отношения R_2, но не содержащихся в схеме отношения R_1, $\boldsymbol{Y}$ – множество всех атрибутов, содержащихся как в схеме отношения R_1, так и в схеме отношения R_2, $\boldsymbol{Z}$ – множество всех атрибутов, содержащихся в схеме отношения R_1, но не содержащихся в схеме отношения R_2. Тогда будут выполняться следующие соотношения

$$\boldsymbol{V} = \boldsymbol{Y} \cup \boldsymbol{Z} \text{ и } \boldsymbol{W} = \boldsymbol{X} \cup \boldsymbol{Y}.$$

С учетом этого данные отношения можно переписать так: $R_1[\boldsymbol{YZ}]$ и $R_2[\boldsymbol{XY}]$, где $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ могут быть в отличие от предыдущего примера не отдельными атрибутами, а некоторыми их совокупностями. В этом случае при выполнении операций соединения и композиции отношений, когда требуется привести оба отношения к одной и той же схеме отношения, в состав схемы отношения R_1 добавляется столько фиктивных атрибутов, сколько их содержится в множестве $\boldsymbol{Z}$, а в состав схемы отношения R_2 – столько фиктивных атрибутов, сколько их содержится в множестве $\boldsymbol{X}$. В остальном же выполнение всех необходимых операций выполняется аналогично.

Рассмотрим использование АК в дедуктивных базах данных. Считается, что *дедуктивная база данных* – это интеграция СУБД с системами логического вывода. Среди многих работ, посвященных этой теме, можно выделить два направления [13].

1) Работы, в которых предусматривается связывание систем управления реляционными БД с системами программирования на языке Пролог, который был разработан для систем искусственного интеллекта. Это так называемые *CPR*-системы (аббревиатура от Coupling Prolog to Relational databases).
2) Работы, в которых принят единый язык управления для баз данных и систем логического вывода. Этот язык программирования получил название Дейталог. Синтаксис этого языка во многом сходен с синтаксисом Пролога.

Но такие методы интеграции не позволяют использовать многие аналитические средства логического анализа систем. Недостатки дедуктивных баз данных во многом обусловлены отсутствием объединяющего математического подхода. Использование АК позволяет обеспечить единый подход к представлению реляционных баз данных и систем логического вывода. Кроме того, подход на основе АК упрощает реализацию поисковых алгоритмов в БД.

Рассмотрим основные элементы БД с точки зрения АК. Некоторые из отношений БД организованы в машинной памяти в виде файлов, структура которых соответствует таблицам. Эти файлы называются *экстенсиальными отношениями*. Однако возможны отношения, которые могут быть потенциально созданы в БД, но, как правило, создаются не в виде файлов, а как новые схемы отношений или же в виде файлов с новой схемой отношения, предназначенных

для кратковременного хранения. Такие отношения в БД называются *интенсиальными отношениями* или *взглядами* и служат обычно для реализации определенных запросов в БД. Из предыдущего ясно, что при формировании *взглядов* в АК можно воспользоваться методами, которые использовались при вычислении соединения отношений или использовать более быстрый способ с помощью декомпозиции поисковых запросов – в этом случае нет необходимости вычислять соединение отношений большого объема.

Математической основой СУБД является *реляционная алгебра* (RA), предназначенная для обработки многоместных отношений, заданных в виде таблиц, с помощью определенного набора операций. В ней предусмотрены пять основных операций. Это ***проекция***, ***объединение***, ***прямое произведение***, ***разность*** и ***селекция***. Остальные операции RA реализуются как комбинации основных операций.

Операция ***проекция*** в RA создает из отношения $Q[\boldsymbol{XY}]$ отношение $Q_p[\boldsymbol{X}]$. В результате этой операции из таблицы, в схеме отношения которой содержится множество атрибутов $\boldsymbol{X} \cup \boldsymbol{Y}$, элиминируется множество $\boldsymbol{Y}$ атрибутов. Такая операция в АК предусмотрена – многократная элиминация атрибутов. Кроме того, в АК становится понятен логический смысл этой операции – многократное использование квантора существования к выбранным для элиминации атрибутам.

Операция ***объединение*** в RA соответствует объединению АК-объектов с одинаковыми схемами отношений.

Операция ***прямое произведение*** (или *декартово произведение*) в RA соответствует в АК пересечению C-систем, у которых схемы отношений не содержат одинаковых атрибутов. Например, прямым произведением в RA двух отношений $P[XY] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ и $Q[ZV] = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$ является отношение

$$R[XYZV] = \begin{bmatrix} a & b & e & f \\ a & b & g & h \\ c & d & e & f \\ c & d & g & h \end{bmatrix}.$$

В АК реализация этой операции намного упрощается за счет использования фиктивных атрибутов. Чтобы выполнить эту операцию в АК, достаточно найти пересечение двух C-систем, приведенных с помощью добавления фиктивных атрибутов к единой схеме отношения:

$$P[XYZV] = \begin{bmatrix} a & b & * & * \\ c & d & * & * \end{bmatrix} \text{ и } Q[XYZV] = \begin{bmatrix} * & * & e & f \\ * & * & g & h \end{bmatrix}$$

и получить тот же результат.

Операция ***разность*** в RA соответствует в АК одноименной операции с однотипными отношениями, при этом можно воспользоваться соотношением:

$$P[\boldsymbol{X}] \setminus Q[\boldsymbol{X}] = P[\boldsymbol{X}] \cap \overline{Q[\boldsymbol{X}]},$$

которое дает возможность применять структуры и алгоритмы, не предусмотренные в RA (например, вычисление дополнений, использование D-кортежей и D-систем).

Операция ***селекция*** в RA предназначена для реализации запросов в БД и позволяет выбрать из заданного экстенсиального или интенсиального отношения набор кортежей, удовлетворяющих заданным ограничениям. Эти ограничения могут применяться к одному и более атрибутам данной схемы отношения и за-

даются для произвольных атрибутов X и Y в виде двух вариантов: 1) $X \Delta c$ и 2) $X \Delta Y$, где c – элемент или множество элементов домена, соответствующего данному атрибуту, Δ – одно из отношений сравнения, возможный список которых можно представить множеством $\{<, \leq, =, \neq, \geq, >\}$.

Рассмотрим некоторые примеры применения операции селекции в АК. Селекция – это реализация некоторого запроса. В каждом запросе содержатся ограничения на значения некоторых атрибутов (например, «возраст не более 25 лет») и список атрибутов, значения которых надо получить из базы данных. Запрос можно представить в виде *селектора*, т.е. структуры, которая при пересечении с соответствующим экстенсиальным или интенсиальным отношением даст ответ на этот запрос. Для случая, когда ограничение задано в виде равенств $X_i = c_i$, селектор можно представить как C-кортеж, у которого выбранные атрибуты X_i представлены элементами или подмножествами соответствующих доменов, а остальные атрибуты данной схемы отношения – фиктивными компонентами. Например, для отношения $P[XYZV]$ C-кортеж $[* \{a\} * \{d, f\}]$ является селектором с ограничениями $Y = a$; $V = d$ или f. Селекция по данным ограничениям формируется при пересечении этого C-кортежа с отношением P. Для ограничения типа $X_i \neq c_i$ в соответствующей позиции C-кортежа-селектора применяется компонента $\{\overline{c_i}\}$, т.е. множество, в котором содержатся все элементы соответствующего домена за исключением элемента c_i.

Для отношений типа $<$, $\leq$, $\geq$ или $>$, которые применимы только в частично упорядоченных множествах, также можно сформировать в качестве селекторов C-кортежи, у которых соответствующие компоненты заданы как интервалы или совокупности интервалов. Например, если домен атрибута X содержит элементы, у которых имеется естественный порядок (например, моменты времени), то ограничение $X < t_i$ означает, что компонента этого атрибута в селекторе будет содержать все элементы, предшествующие t_i.

Когда задаются ограничения типа 2 ($X \Delta Y$), в которых используются два разных атрибута, то в этом случае неявно предполагается, что атрибуты X и Y относятся к одному сорту. Например, атрибутами такого рода могут быть «время встречи» и «время расставания». Если встреча занимала незначительный промежуток времени в пределах точности отсчета, то можно найти в базе данных такие быстротечные встречи, если сформировать запрос, в котором приравниваются разные атрибуты X и Y. В качестве примера рассмотрим С-систему

$$P[XYZ] = \begin{bmatrix} \{a,b\} & \{b,c\} & \{a\} \\ \{b,c\} & \{a\} & \{a,c\} \\ (a,c\} & \{a,b,c\} & \{b,c\} \end{bmatrix},$$

для которой нужно реализовать следующий запрос: «Чему равны значения атрибута Z для случаев, когда значения атрибутов X и Y равны?»

Первый способ – это сформировать запрос в виде С-системы, у которой в атрибутах X и Y содержатся одинаковые одноэлементные множества, и затем найти пересечение этого запроса с отношением P. Запрос тогда можно выразить так:

$$Q = \begin{bmatrix} \{a\} & \{a\} & * \\ \{b\} & \{b\} & * \\ \{c\} & \{c\} & * \end{bmatrix}.$$

Второй способ позволяет значительно сократить объем вычислений, особенно в тех случаях, когда домены атрибутов, по которым выполняется селекция такого рода, содержат большое число элементов. Вычисления в этом случае выполняются так:

1) В *C*-системе *P* в каждой строке найти пересечение компонент сравниваемых атрибутов (в нашем случае *X* и *Y*) и результат записать вместо исходных значений.
2) Исключить из полученной *C*-системы пустые *C*-кортежи.
3) Если вычисленные компоненты содержат множества из двух и более элементов, то заменить такие *C*-кортежи совокупностью *C*-кортежей, содержащих одинаковые одноэлементные компоненты. Конец алгоритма.

Посмотрим выполнение этого алгоритма для нашего примера.

Шаг 1. Получаем $\begin{bmatrix} \{b\} & \{b\} & \{a\} \\ \varnothing & \varnothing & \{a,c\} \\ (a,c\} & \{a,c\} & \{b,c\} \end{bmatrix}$.

Шаг 2. $\begin{bmatrix} \{b\} & \{b\} & \{a\} \\ \{a,c\} & a,c\} & \{b,c\} \end{bmatrix}$.

Шаг 3. $\begin{bmatrix} \{b\} & \{b\} & \{a\} \\ \{a\} & \{a\} & \{b,c\} \\ (c\} & \{c\} & \{b,c\} \end{bmatrix}$.

Если бы мы остановились на шаге 2, то в этом случае декартово произведение $\{a, c\}\times\{a, c\}$ содержало бы не только нужные элементарные кортежи (a, a) и (c, c), но и кортежи (a, c) и (c, a), которые не соответствуют условию запроса.

Рассмотрим некоторые производные операции RA.

Операция ***пересечения*** двух отношений *R* и *S* в RA используется только в том случае, когда эти отношения однотипны, и реализуется как комбинация основных операций:

$$R \setminus (R \setminus S).$$

Используя законы алгебры множеств, получим:

$$R \setminus (R \setminus S) = R \cap \overline{R \cap \overline{S}} = R \cap (\overline{R} \cup S) = (R \cap \overline{R}) \cup (R \cap S) = R \cap S.$$

Это означает, что операцию пересечения в АК можно выполнять непосредственно, не применяя дважды операцию разности.

Операция ***соединения*** в RA имеет тот же смысл, что и одноименная операция в АК.

Для иллюстрации работы интеллектуальной БД рассмотрим пример из [13, с. 42-47]. Для этой базы данных с помощью языка Дейталог решается известная «Антитрестовская задача», суть которой заключается в следующем. В законодательстве некоторых стран существуют антитрестовские законы, не позволяющие фирмам непосредственно или косвенно (через другие фирмы) контролировать долю рынка, превышающую нормативную квоту для определенных товаров. Если какая-либо фирма имеет контрольный пакет акций другой фирмы (51% и выше), то она тем самым контролирует деятельность дочерней фирмы. Таким образом, возможна ситуация, когда доля рынка товаров самой фирмы не превышает квоты, но при учете других участников рынка, контролируемых этой фирмой, эта доля может превысить квоту для данного товара. Требуется найти таких «скрытых» нарушителей Антитрестовского закона.

В базе данных содержатся сведения об 11 фирмах. Каждая фирма (*A*, *B*, *C*, …, *M*, *N*) имеет определенную долю рынка в процентах для определенного вида товаров (T1 и T2). Эти сведения содержатся в таблице 2. В таблице 3 показаны пары фирм и проценты всех акций Фирмы2, являющиеся собственностью Фирмы1. В таблице 4 показаны наименования товаров и квота рынка в процентах для каждого товара. В данном случае необходимо выявить фирмы, торгующие товаром T1 и нарушающие антитрестовский закон.

Таблица 2. Распределение долей выпускаемых товаров по фирмам.

Фирма	Товар	Доля рынка (%)
A	T1	8
B	T1	7
C	T2	33
D	T1	13
E	T1	0
F	T1	12
G	T2	30
H	T1	8
I	T1	12
J	T1	2
K	T1	4
L	T1	19
M	T1	15
N	T2	35

Таблица 3. Отношение собственности между фирмами.

Фирма1	Фирма2	Процент акций
A	*B*	51
A	*D*	20
A	*E*	30
A	*J*	51
B	*H*	30
B	*I*	70
E	*D*	51
N	*C*	2
E	*F*	60
J	*K*	59
F	*L*	20
F	*M*	25
M	*L*	60

Таблица 4. Квота рынка товаров.

Товары	Предельная квота
T1	20
T2	38
T3	40

В [13] для постановки и решения этой задачи используется язык логического программирования. Здесь же мы поставим задачу в терминах АК и частично упорядоченных множеств. Задачу будем решать только для товара T1.

Начнем с выбора фирм, имеющих контрольный пакет акций в других фирмах. Для этого применим селектор [∗ ∗ {>50}] к таблице 3. Получим отношение $R1$, представленное в таблице 5.

Таблица 5. Отношение собственности между фирмами.

Фирма1	**Фирма2**	**Процент акций**
A	*B*	51
A	*J*	51
B	*I*	70
E	*D*	51
E	*F*	60
J	*K*	59
M	*L*	60

Из полученного отношения $R1$ выделим две проекции $R1$[**Фирма1 Фирма2**] и $R1$[**Фирма1**]. Первая проекция является графом (его можно выразить как *C*-систему, объединив некоторые элементарные кортежи), в котором содержатся направленные связи между фирмами и контролируемыми ими другими участниками рынка. Вторая проекция – это просто множество фирм, которые имеют контрольные пакеты акций каких-либо других фирм.

$$R1[\textbf{Фирма1 Фирма2}] = \begin{bmatrix} \{A\} & \{B,J\} \\ \{B\} & \{I\} \\ \{E\} & \{D,F\} \\ \{J\} & \{K\} \\ \{M\} & \{L\} \end{bmatrix};$$

$$R1[\textbf{Фирма1}] = \{A, B, E, J, M\}.$$

Далее выделим из множества $R1$[**Фирма1**] тех участников рынка, которые торгуют товаром T1. Для этого применим селектор [∗ {T1} ∗] к отношению «Фирмы» и в полученном отношении $R2$ выберем проекцию $R2$[**Фирма**], в результате чего получим множество, совпадающее с множеством $R1$[**Фирма1**]. Таким образом, для *C*-системы $R1$ корректировка не нужна.

Следующий этап: выявление множеств фирм, которые контролируются одной фирмой на рынке товара T1. Этот этап добавляет свойство «интеллектуальности» в обычную СУБД и для его реализации требуются рекурсивные алгоритмы. С математической точки зрения здесь необходимо получить транзитивное замыкание графа $R1$[**Фирма1 Фирма2**], вычисление которого, как было показано ранее, можно реализовать средствами АК. В результате получим *C*-систему

$$R1[\textbf{Фирма1 Фирма2}]^{+} = \begin{bmatrix} \{A\} & \{B,I,J,K\} \\ \{B\} & \{I\} \\ \{E\} & \{D,F\} \\ \{J\} & \{K\} \\ \{M\} & \{L\} \end{bmatrix}$$

Если фирма *X* контролирует товары фирм {*Y*, *Z*, ...}, то это означает, что фирма *X* контролирует товары не только дочерних фирм, но собственных товаров. Обозначим *C*(*X*) множество всех фирм, контролируемых фирмой *X*. Тогда для фирм, содержащих хотя бы одну дочернюю фирму, получим

$C(A) = \{A, B, I, J, K\}$; $C(B) = \{B, I\}$; $C(E) = \{E, D, F\}$;
$C(J) = \{J, K\}$; $C(M) = \{M, L\}$.

Чтобы получить долю рынка товара Т1, контролируемого фирмой X, необходимо сложить доли рынка этого товара для фирм, содержащихся в множестве $C(X)$. В результате такого подсчета получим:

для фирмы A – {8, 7, 12, 2, 4}; $\Sigma = 33$;
для фирмы B – {7, 12}; $\Sigma = 19$;
для фирмы E – {0, 13, 12}; $\Sigma = 25$;
для фирмы J – {2, 4}; $\Sigma = 6$;
для фирмы M – {19, 15}; $\Sigma = 34$.

Теперь, используя данные отношения «Трестовский лимит», можно определить, что нарушителями антитрестовского закона являются фирмы A, E и M.

Отсюда ясно, что задачу, для решения которой в силу интеллектуальной сложности рекомендуется использовать методы и средства логического программирования, можно решать, используя только средства АК.

3.3. Экспертные системы

В экспертных системах (ЭС) помимо данных, выраженных в основном в виде таблиц, предусматривается *база знаний*, которая состоит из набора *правил*, регламентирующих определенные преобразования данных. Этими правилами являются логические выражения определенного типа. Чаще всего в качестве таких выражений используются *продукции*, т.е. импликации типа

Если A_1 и A_2 и ... и A_k , *то* B_1 или B_2 или ... или B_n.

Если правило выражено в виде импликации $X \supset Y$, то логическое выражение X называется *телом правила*, а логическое выражение Y – *головой* (или *целью*) *правила*.

Одним из критических узлов в любой экспертной системе является *интерпретатор* – программно-математическая система преобразования данных в соответствии с правилами. Математические принципы работы интерпретаторов основаны на символьных преобразованиях математической логики, их программная реализация весьма сложна и, как правило, не показывается даже в самой подробной технической документации. В программной реализации интерпретаторов в основном используются списковые структуры данных, которые при машинной реализации больших систем характеризуются невысоким быстродействием и поэтому часто не могут использоваться в системах реального времени.

Рассмотрим пример, с помощью которого иллюстрируется работа интерпретатора в структурах АК. В статье [14] приведено описание реализованной на компьютере экспертной системы, предназначенной для принятия решений на уровне командира корабля в случае возникновения нештатных ситуаций на корабле или в окружающей обстановке, в частности, в боевых условиях. В качестве исходных данных для принятия решений использовались обобщенные данные по подсистемам корабля и окружающей среды, т.е. данные типа «утрата непотопляемости возрастает медленно (или быстро)», «возник пожар в энергоотсеках в одном эшелоне», «состояние моря 6 баллов» и т.д.

Пример перевода правил ЭС на язык АК рассмотрим для комплекта правил по анализу электрообеспечения корабля. В этом комплекте используются 4 фактора. Данные по факторм и соответствующие обозначения приведены в таблице 6.

Таблица 6. Факторы экспертной системы для анализа живучести корабля.

Наименования факторов и их значения	Обозначения факторов	Обозначения значений факторов
СНАБЖЕНИЕ ЭЛЕКТРОЭНЕРГИЕЙ	*W*	
обеспечено полностью		*a*
утрачено частично		*b*
утрачено полностью		*c*
ОСНОВНЫЕ ИСТОЧНИКИ	*X*	
в строю		*d*
вышли из строя в одном эшелоне		*e*
вышли из строя в обоих эшелонах		*f*
АВАРИЙНЫЕ ИСТОЧНИКИ	*Y*	
в строю		*g*
вышли из строя		*h*
ПРОВОДКА ЭНЕРГИИ	*Z*	
Не нарушена		*i*
нарушена частично		*j*
нарушена полностью		*k*

На предварительном этапе реализации системы был составлен следующий комплект правил для системы электрообеспечения.

***Инициализация*: W** = неизвестно.
***Цель*: W**.
***Правило 1*; приоритет 30:**
IF ($X = f$ AND $Y = h$) OR $Z = j$
THEN $W = c$
***Правило 2*; приоритет 20:**
IF $X = e$ OR $Z = j$
THEN $W = b$

Приоритет в данном случае устанавливает порядок применения правил: чем больше приоритет правила, тем раньше оно должно быть применено, при этом переход к следующему правилу происходит в случае безуспешной проверки. При успешной проверке происходит выход из комплекта правил с присвоением значения целевому фактору (в данном случае фактору *W*). Если в обоих правилах проверка безуспешна, то фактору *W* присваивается значение *a*.

С точки зрения АК данная структура состоит из 4-х атрибутов: $W = \{a, b, c\}$; $X = \{d, e, f\}$; $Y = \{g, h\}$; $Z = \{i, j, k\}$. Преобразуем условия правил 1 и 2 в АК-объекты, приведя их к единой схеме отношения [*XYZ*]. Тогда получим:

$$\text{Условие правила 1: } R_1[XYZ] = \begin{bmatrix} \{f\} & \{h\} & * \\ * & * & \{k\} \end{bmatrix},$$

$$\text{Условие правила 2: } R_2[XYZ] = \begin{bmatrix} \{e\} & * & * \\ * & * & \{j\} \end{bmatrix}.$$

Рассмотрим, как используются эти правила в АК. Пусть поступила информация: «Основные источники вышли из строя в одном эшелоне, аварийные источники в строю и проводка энергии нарушена частично». Эта информация преобразуется в *C*-кортеж $T = [\{e\}\ \{g\}\ \{j\}]$. Применим Правило 1, для чего найдем пересечение $T \cap R_1 = \varnothing$. Следовательно, ситуация не соответствует этому правилу. Применим Правило 2: $T \cap R_2 = [\{e\}\ \{g\}\ \{j\}]$.

Поскольку пересечение не пусто, то, следовательно, $W = b$.

Проанализируем корректность этих правил средствами АК. Во-первых, проверим неоднозначности, т.е. определим, являются ли некоторые ситуации общими для каждого из условий. Для этого вычислим пересечение

$$R_1 \cap R_2 = \begin{bmatrix} \{f\} & \{h\} & \{j\} \\ \{e\} & * & \{k\} \end{bmatrix}.$$

Непустота пересечения свидетельствует о том, что имеются ситуации, неразличимые для заданных правил. Но в самой экспертной системе неоднозначности не возникают, так как приоритет Правила 1, в котором предусмотрено это состояние, более высоки, и эта ситуация распознается правильно. Но возможна еще одна некорректность, когда некоторые критические состояния не включены ни в одно из условий для аварийных ситуаций. Для анализа вычислим множество таких состояний с помощью формулы $\overline{R_1 \cup R_2}$. Это позволяет получить множество возможных ситуаций, не учтенных в правилах.

$$R_1 \cup R_2 = \begin{bmatrix} \{f\} & \{h\} & * \\ * & * & \{k\} \\ \{e\} & * & * \\ * & * & \{j\} \end{bmatrix} = \begin{bmatrix} \{f\} & \{h\} & * \\ * & * & \{j,k\} \\ \{e\} & * & * \end{bmatrix}$$

В исходной С-системе можно объединить в один *C*-кортеж 2-й и 4-й *C*-кортежи

$$\overline{R_1 \cup R_2} = \begin{bmatrix} \{d,e\} & \{g\} & \varnothing \\ \varnothing & \varnothing & \{i\} \\ \{d,f\} & \varnothing & \varnothing \end{bmatrix}.$$

Получилась *D*-система. Преобразуем ее в *C*-систему (промежуточные выкладки здесь не показаны):

$$\begin{bmatrix} \{d,e\} & \{g\} & \varnothing \\ \varnothing & \varnothing & \{i\} \\ \{d,f\} & \varnothing & \varnothing \end{bmatrix} = \begin{bmatrix} \{d\} & * & \{i\} \\ \{f\} & \{g\} & \{i\} \end{bmatrix}.$$

С точки зрения специалиста одна из 3-х возможных ситуаций ([{*f*} {*g*} {*i*}]) является аварийной и соответствует состоянию $W = c$. Но она не включена ни в одно из условий, поэтому данные правила нуждаются в корректировке. В данном случае корректировка заключается в добавлении *C*-кортежа [{*f*} {*g*} {*i*}] в условие R_1. В результате получим

$$R_1 = \begin{bmatrix} \{f\} & \{h\} & * \\ * & * & \{k\} \\ \{f\} & \{g\} & \{i\} \end{bmatrix}.$$

Используя средства АК, можно также заменить комплект правил одним АК-объектом. Для этого объединим созданные в системе АК условия в один АК-объект и добавим атрибут, характеризующий состояние всей системы, в которой запишем соответствующие состояния. Тогда получим АК-объект, который распознает все возможные ситуации для данного комплекта правил. В частности, для рассмотренного примера таким АК-объектом является *C*-система.

$$R[XYZW] = \begin{bmatrix} \{f\} & \{h\} & * & \{c\} \\ * & * & \{k\} & \{c\} \\ \{f\} & \{g\} & \{i\} & \{c\} \\ \{e\} & * & * & \{b\} \\ * & * & \{j\} & \{b\} \\ \{d\} & * & \{i\} & \{a\} \end{bmatrix}.$$

Допустим, на вход поступила информация $X = e$, $Z = k$. Эта информация преобразуется в C-кортеж $T[XYZW] = [\{e\} * \{k\} *]$ и находится пересечение $T \cap R$. В данном случае после выполнения операции мы получим C-систему

$$\begin{bmatrix} \{e\} & * & \{k\} & \{c\} \\ \{e\} & * & \{k\} & \{b\} \end{bmatrix} = [\{e\} * \{k\} \ \{c, b\}]$$

с неоднозначным результатом в атрибуте W.

Однако эту неоднозначность можно устранить, если использовать правило приоритета. Приоритет здесь определяется порядком расположения C-кортежей в R: первый полученный непустой результат пересечения является окончательным. С учетом этого для заданного входа получается результат $W = c$.

Из этого примера видно, что на основе АК можно создавать «прозрачные» и сравнительно простые в реализации интерпретаторы экспертных систем и к тому же сравнительно легко проверять корректность правил на этапе проектирования системы.

3.4. Вероятностная логика на основе АК

Термин «вероятностная логика» получил большое распространение в работах по искусственному интеллекту после опубликования статьи известного специалиста по ИИ Н. Нильсона [15]. В этой и других публикациях по вероятностной логике ставилась следующая задача: заданы оценки вероятностей некоторого множества событий, представленных формулами исчисления высказываний, необходимо найти вероятностную оценку события, заданного логической формулой, отличающейся от исходных. Анализ статей по вероятностной логике показывает, что в них результатом соединения классических понятий «вероятность» и «логика» оказываются некоторые неклассические логики. Здесь предлагается рассмотреть концепцию вероятностной логики в классическом варианте через призму алгебры кортежей.

Совмещение понятий «логика» и «вероятность» вызывает немало трудностей. На первый взгляд, здесь все просто, если взять за основу аксиоматику вероятности, предложенную А.Н. Колмогоровым [16], в которой алгебра событий, погруженных в вероятностную меру, соответствует алгебре множеств. Например, для событий, представленных множествами A и B, вероятностная мера их объединения вычисляется по формуле

$$p(A \cup B) = p(A) + p(B) - p(A \cap B).$$

Таким образом, для точного вычисления вероятности события $(A \cup B)$ помимо вероятностей $p(A)$ и $p(B)$ необходимо знать вероятность $p(A \cap B)$, которая в рамках некоторых ограничений, в частности, таких: $p(A \cap B) \le min(p(A), p(B))$, не зависит от $p(A)$ и $p(B)$. Если при этом $A \cap B \neq \varnothing$, то события A и B являются зависимыми. Но если представить, что A и B являются не множествами, а разными логическими переменными, то вероятность дизъюнкции этих событий вычисляется по формуле:

$$p(A \vee B) = p(A) + p(B) - p(A)\,p(B),$$

для вычисления которой достаточно задать только вероятность событий A и B.

Спрашивается, почему для логических соотношений имеет место другая методика расчета вероятности, хотя многим представляется, что алгебра множеств и булева алгебра изоморфны? Ответ на этот вопрос оказывается ключевым при совмещении понятий «логика» и «вероятность». Дело в том, что в классической логике элементарные события, соответствующие разным логическим переменным, несовместимы, потому что любая логическая формула, содержащая n свободных переменных, изоморфна некоторому n-местному отношению, и события, соответствующие различающимся переменным, принадлежат разным атрибутам. Другими словами, логические переменные могут быть зависимыми, но не изначально, а только потому, что они содержатся в определенной логической формуле, которая и определяет зависимость между ними.

Абсурдность (с точки зрения математической логики) иного подхода видна из следующего примера, который иногда приводится в работах по вероятностной логике: для логических переменных (но не формул!) X и Y даются вероятности $p(X)$, $p(Y)$ и $p(X \wedge Y)$, причем последняя вероятность необязательно равна произведению предыдущих.

Отсюда ясно, что *при погружении логических систем в вероятностное пространство необходимо учитывать, что речь идет о системе, изоморфной в соответствии с аксиоматикой А.Н. Колмогорова алгебре множеств, но при этом сами множества по структуре являются множествами кортежей, содержащихся в многоместных отношениях*.

Именно это обстоятельство, которое явно или неявно учитывается в АК и в логико-вероятностных методах (ЛВМ) [10], не учитывается в различных версиях вероятностной логики. Предположение о том, что события, соответствующие разным логическим переменным, могут быть зависимыми сами по себе, т.е. без учета связывающей их логической формулы, означает нарушение законов математической логики. Другое дело, когда речь идет о самих формулах, в которых устанавливается зависимость между различными переменными, или о разных логических формулах, которые могут быть зависимыми только при условии, что они содержат хотя бы одну общую для них свободную переменную.

Рассмотрим методы вероятностного моделирования с помощью АК более подробно. Основным структурообразующим понятием АК является C-кортеж. Если известны вероятностные меры компонент C-кортежа, то мера самого C-кортежа вычисляется как произведение мер его компонент. Так, если C-кортеж $R = [A\ B\ C]$ задан в измеримых атрибутах, и меры его компонент равны соответственно $\mu(A)$, $\mu(B)$ и $\mu(C)$, то

$$\mu(R) = \mu(A) \cdot \mu(B) \cdot \mu(C)\,.$$

Если речь идет о погружении логических формул в вероятностное пространство, то все атрибуты пространства, в котором задана совокупность АК-объектов, имеют меру 1, а все АК-объекты этого пространства имеют меры, не превышающие 1. Это соответствует вероятностной мере не только по числовым соотношениям, но и потому, что система событий, моделируемых с помощью АК, изоморфна алгебре множеств.

Для вычисления мер АК-объектов, отличающихся от C-кортежей, необходима их ортогонализация, т.е. преобразование в эквивалентную C-систему, в которой пересечение любой пары C-кортежей является пустым множеством.

Если АК-объект является отображением формул исчисления высказываний, то он задан в универсуме $\{0,1\}^n$, где n – число логических переменных форму-

лы. Каждый столбец *C*-кортежа или *C*-системы «привязан» к определенной логической переменной. Столбцу с номером k соответствует переменная x_k. Литералу x_k в формуле соответствует состояние 1 в АК-объекте, а литералу $\neg x_k$ – состояние 0. Каждая строка (*C*-кортеж) в *C*-системе соответствует конъюнкции формулы, выраженной как ДНФ. Если в каком-либо конъюнкте отсутствуют переменные, входящие в состав формулы, то вместо них в соответствующем *C*-кортеже вставляется фиктивная переменная "∗". Рассмотрим пример.

Пусть задана формула исчисления высказываний

(2) $$F_Q = (x_1 \wedge \neg x_3) \vee (\neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2).$$

Поскольку здесь три логических переменных, то эту формулу можно представить как АК-объект Q в универсуме $\{0, 1\}^3$:

$$Q = \begin{bmatrix} \{1\} & * & \{0\} \\ * & \{0\} & \{1\} \\ \{0\} & \{1\} & * \end{bmatrix}.$$

Данная формула и соответствующий ей АК-объект являются ортогональными, поэтому Q можно непосредственно выразить в вероятностной мере. Пусть в АК-объекте Q вероятности событий заданы так: p_i – вероятность события 1 в i-ом атрибуте, $1–p_i$ – вероятность события 0 в i-ом атрибуте. Учитывая, что мера *C*-кортежа равна произведению мер его компонент, а мера ортогональной *C*-системы – сумме мер содержащихся в ней *C*-кортежей, получим формулу

(3) $$p(Q) = p_1(1–p_3)+(1–p_2)p_3+(1–p_1)p_2.$$

В ЛВМ формула (3) называется *вероятностной функцией* (ВФ) формулы (2). Эту функцию можно также получить из ортогональной *C*-системы с помощью замены элементов компонент на соответствующие им вероятности и преобразования системы в полином. На первый взгляд кажется, что структуры АК – лишь другой способ выражения для логических формул. Однако при усложнении моделей (в частности, при переходе к системам с многими состояниями) с помощью АК оказывается возможным существенно упростить алгоритмы решения ряда задач, решаемых в ЛВМ. Кроме того, в АК при погружении в вероятностное пространство вводится понятие «уравнение регрессии», которое позволяет осуществить постановку и решение задач вероятностной логики по Н. Нильсону. *Если в вероятностных функциях типа* (3) *считать p_i не фиксированными числами, а переменными, то такие формулы являются точным уравнением регрессии соответствующей логической формулы*. Доказательство этого утверждения приведено в [5]. Рассмотрим пример системы с многими состояниями.

Пусть $R = \begin{bmatrix} \{a_1, a_2\} & \{b_1, b_3\} \\ \{a_3\} & \{b_1, b_2\} \end{bmatrix}$ – ортогональная *C*-система с тремя состояниями задана в пространстве $\{a_1, a_2, a_3\} \times \{b_1, b_2, b_3\}$ с вероятностями $p(a_i)$ и $p(b_i)$, при этом $p(a_3) = 1– p(a_1) – p(a_2)$ и $p(b_3) = 1– p(b_1) – p(b_2)$.

Тогда вероятность события, выраженного АК-объектом R, при подстановке соответствующих вероятностей будет равна

$$p(R) = (p(a_1) + p(a_2))(1– p(b_2)) + (1– p(a_1) – p(a_2))(p(b_1)+ p(b_2)).$$

Изложенный подход соответствует *прямой задаче* логико-вероятностного анализа, когда при заданных вероятностях элементарных событий выполняется расчет вероятности сложного события. В *обратной задаче* постановка иная – на основе данных о вероятностях некоторых сложных событий нужно рассчитать вероятности элементарных событий, после чего можно рассчитать вероятности

других сложных событий. К ним относятся задачи, решаемые в вероятностной логике. Рассмотрим пример, приведенный в статье Н. Нильсона [15].

Дана совокупность событий, заданных формулами A и $A \supset B$ исчисления высказываний, при этом $p(A) = p_1$ и $p(A \supset B) = p_2$. Требуется оценить вероятность $p(B)$ события B.

Задачу будем решать методами АК. Имеются всего две логические переменные A и B, которые в данном случае являются также элементарными событиями. Предположим, что вероятность этих событий равна соответственно $p(A)$ и $p(B)$. Из условия задачи следует, что $p(A) = p_1$. Выразим заданные формулы в структурах АК, используя универсум $\{0, 1\}^2$:

$$A = [\{1\}\ *]; \quad B = [*\ \{1\}];$$

$$A \supset B = \overline{A} \vee B =]\{0\}\ \{1\}[= \begin{bmatrix} \{0\} & * \\ \{1\} & \{1\} \end{bmatrix}$$

(здесь D-кортеж, соответствующий формуле $A \supset B$, преобразован в ортогональную C-систему).

На основании этого получим вероятностные формулы для событий A и $A \supset B$:

$$P(A) = p_1;$$

$$P(A \supset B) = (1 - p(A)) + p(A)p(B) = p_2.$$

Получилась система из двух уравнений

$$p(A) = p_1;$$

$$(1 - p(A)) + p(A)\, p(B) = p_2.$$

Из этой системы несложно вывести

$$p(B) = \frac{p_1 + p_2 - 1}{p_1}.$$

В данной задаче получен точный ответ, в то время как в [15] ответ получен как неравенство $p_2 + p_1 - 1 \le p(B) \le p_2$.

В общем случае алгоритм решения задач вероятностной логики следующий. Пусть заданы исходные логические формулы F_i с заданными вероятностями $p(F_i)$ и формула G, вероятность которой $p(G)$ требуется вычислить. Тогда необходимо выполнить следующую последовательность операций:

a) формулы F_i и G преобразуются в ортогональные C-системы;
b) для каждой из этих систем составляется уравнение регрессии $E(F_i)$ и $E(G)$;
c) составляется и решается система уравнений $\{E(F_i)\}$;
d) если система уравнений $\{E(F_i)\}$ имеет единственное решение, то полученные значения переменных подставляются в формулу $E(G)$ и находится точный ответ.

Но точный ответ в подобных задачах возможен не во всех случаях. Рассмотрим пример.

Даны вероятности событий, заданных логическими формулами:

$$p(A \vee B) = a;$$

$$p(A \wedge B) = b.$$

Найти оценку $p(A)$ и $p(B)$.

Выразим заданные события в системе как ортогональные C-системы:

$$A \vee B \Leftrightarrow]\{1\}\ \ \{1\}[= \begin{bmatrix} \{1\} & * \\ \{0\} & \{1\} \end{bmatrix};$$

$$A \wedge B \Leftrightarrow [\{1\}\ \ \{1\}].$$

Составим систему уравнений:

$$p(A) + (1 - p(A))\, p(B) = a;$$
$$p(A)\, p(B) = b.$$

Решая данную систему уравнений, получим

$$p(A) = \frac{a + b \pm \sqrt{(a+b)^2 - 4b}}{2};$$
$$p(B) = \frac{a + b \mp \sqrt{(a+b)^2 - 4b}}{2}.$$

Видно, что полученные решения не дают однозначный ответ в тех случаях, когда подкоренное выражение не равно 0 (это возможно для случая $p(A) \neq p(B)$). Если учитывать, что обе исходные формулы симметричны, то такая неопределенность была предрешена условиями задачи.

Для приведенных примеров можно численно проверить выкладки, если построить соответствующие им вероятностные модели. Так, для последнего примера соответствующая вероятностная модель будет следующей: пусть одновременно бросаются две монеты, при этом известна вероятность выпадения хотя бы одного герба (этому событию соответствует формула $A \vee B$) и вероятность выпадения герба в двух бросаниях (формула $A \wedge B$). Зная вероятность выпадения герба (для правильных монет – 0.5), мы можем по законам теории вероятности подсчитать вероятности этих сложных событий: $p(A \vee B) = 0.75$; $p(A \wedge B) = 0.25$. Подстановка этих значений в приведенные выше формулы для $p(A)$ и $p(B)$ дает правильный ответ. Аналогичную проверку можно произвести и для «неправильных» монет, когда вероятности выпадения герба отличаются от 0.5.

Вероятностные соотношения, получаемые на основе АК, позволяют не только оценивать вероятность сложных событий при заданных функциях распределения событий в каждом атрибуте, но и решать обратную задачу вероятностного анализа – оценивать типы и параметры маргинальных распределений, т.е. распределений в атрибутах и некоторых проекциях.

Сначала рассмотрим следующую постановку задачи: система представлена в структурах АК или в виде системы логических функций, и для каждой переменной известно распределение вероятности. В многомерном пространстве распределение в атрибутах или в некоторых проекциях этого пространства называется ***маргинальным распределением***. Необходимо вычислить распределение вероятности самой системы и оценку устойчивости этого распределения. Логические системы, в которых решаются эти задачи, можно, используя ранее выведенные соотношения, преобразовать в измеримые системы АК.

Пусть каждый атрибут некоторой совокупности АК-объектов представлен конечной системой событий. Возможны два варианта задания системы событий в атрибуте X_i:

a) система событий является элементарной системой (т.е. системой попарно несовместимых событий);
b) система событий задана на непрерывном распределении вероятностей $p(x_i)$ в виде конечного множества пересекающихся в общем случае интервалов (a_i, b_i), где a_i, b_i – значения параметра x_i и $a_i < b_i$.

Для первого варианта достаточно сопоставить каждому событию его вероятность. Второй вариант можно свести к первому с помощью следующей процедуры:

a) произвести ***элементаризацию*** системы $\{(a_i, b_i)\}$ интервалов параметра x_i (т.е. в каждом атрибуте выполнить разбиение систем событий на множество попарно непересекающихся интервалов – ***квантов***);
b) преобразовать исходную систему событий в дискретную, сопоставив с каждым исходным событием некоторое множество квантов, объединение которых равно этому событию;
c) для каждого кванта e_r вычислить значение вероятности p_r, используя в качестве нижних и верхних пределов интегрирования для $p(x_i)$ концевые точки этого кванта, равные значениям параметра x_i.

Если эта процедура выполнена для каждого атрибута АК-объекта, то расчет его вероятности осуществляется в следующей последовательности:
a) осуществляется ортогонализация АК-объекта;
b) производится преобразование АК-объекта в полином, в котором каждому кванту приписывается соответствующее значение вероятности.

Рассмотрим пример. Система задана в пространстве $X \times Y$, где атрибуты представлены в виде интервалов, при этом $X = [0, 7]$ и $Y = [0, 5]$. Заданы также плотности распределения вероятностей $f_1(x, d_1, e_1)$ и $f_2(y, d_2, e_2)$ на атрибутах, где d_1, e_1, d_2, e_2 – параметры распределений. В этой системе задано некоторое событие в виде АК-объекта $R[XY] = \begin{bmatrix} \{a_1\} & \{b_1\} \\ \{a_2, a_4\} & \{b_2\} \\ \{a_3\} & \{b_3\} \end{bmatrix}$, где a_i, b_j – интервалы, заданные в таблице 7. Необходимо определить последовательность расчетов для вычисления вероятности события R.

Таблица 7. Интервалы исходных событий.

a_1	a_2	a_3	a_4	b_1	b_2	b_3
[0, 2.8]	[1.7, 3.4]	[3.4, 5.5]	[4.3, 6.4]	[0, 2.3]	[1.4, 3.2]	[2.3, 5.0]

Для решения задачи достаточно использовать только открытые интервалы. Для элементаризации системы построим сначала возрастающие ряды границ интервалов в атрибутах:

для X: 0; 1.7; 2.8; 3.4; 4.3; 5.5; 6.4; 7;

для Y: 0; 1.4; 2.3; 3.2; 5.

Тогда получим следующие множества элементарных интервалов для атрибутов X (таблица 8) и Y (таблица 9).

Таблица 8. Элементарные интервалы для атрибута X.

r_1	r_2	r_3	r_4	r_5	r_6	r_7
(0, 1.7)	(1.7, 2.8)	(2.8, 3.4)	(3.4, 4.3)	(4.3, 5.5)	(5.5, 6.4)	(6.4, 7)

Таблица 9. Элементарные интервалы для атрибута Y.

q_1	q_2	q_3	q_4
(0, 1.4)	(1.4, 2.3)	(2.3, 3.2)	(3.2, 5)

После замены исходных интервалов на соответствующие им множества квантов получим:

$$a_1=\{r_1, r_2\};\ a_2=\{r_2, r_3\};\ a_3=\{r_4, r_5\};\ a_4=\{r_5, r_6\};$$

$b_1 = \{q_1, q_2\}$; $b_2 = \{q_2, q_3\}$; $b_3 = \{q_3, q_4\}$.

После подстановки в исходную *C*-систему получим

$$R = \begin{bmatrix} \{r_1, r_2\} & \{q_1, q_2\} \\ \{r_2, r_3, r_5, r_6\} & \{q_2, q_3\} \\ \{r_4, r_5\} & \{q_3, q_4\} \end{bmatrix}$$

Для каждого кванта r_i или q_j вычислим соответствующую вероятность. Например,

$$p(r_3) = \int_{2,8}^{3,4} f_1(x, d_1, e_1)dx .$$

Теперь можно приступить к ортогонализации соответствующих сложных событий. Для события *R* вычислим $\overline{R}$ и после преобразования $\overline{R}$ в ортогональную *C*-систему найдем $p(R) = 1 - p(\overline{R})$. Тогда получим (промежуточные вычисления не показаны)

$$\overline{R} = \begin{bmatrix} \{r_3, r_4, r_5, r_6, r_7\} & \{q_1\} \\ \{r_4, r_7\} & \{q_2\} \\ \{r_1, r_7\} & \{q_3\} \\ \{r_1, r_2, r_3, r_6, r_7\} & \{q_4\} \end{bmatrix}.$$

Затем, подставляя вероятности квантов и используя соответствующие теоремы АК, получим

$$p(R) = 1 - p(\overline{R}) = 1 - ((p(r_3) + p(r_4) + p(r_5) + p(r_6) + p(r_7))p(q_1) + (p(r_4) + p(r_7))p(q_2) + (p(r_1) + p(r_7))p(q_3) + (p(r_1) + p(r_2) + p(r_3) + p(r_6) + p(r_7))p(q_4)).$$

При решении обратной задачи для систем с многими состояниями точное решение систем уравнений возможно не всегда, так как число переменных в уравнениях регрессии сопоставимо с числом всех квантов и может превышать число уравнений. Так, в данной задаче число квантов в атрибуте *X* равно 7, следовательно число неизвестных параметров только для этого атрибута будет на единицу меньше, т.е. 6. Однако задача может быть решена приближенно, если данную задачу представить как задачу аппроксимации. Допустим, атрибут X_i разделен на k_i квантов ($k_i > 2$). Тогда будем считать в качестве неизвестных не величины квантов, а типы и параметры непрерывных распределений вероятности для каждого атрибута. Число параметров распределений обычно не превышает двух – они и будут неизвестными величинами. Для их оценки можно использовать методы оптимизации, в которых управляющими воздействиями будут типы и параметры маргинальных распределений, а целевой функцией – обобщенный параметр, например, среднее значение абсолютных отклонений между расчетными и фактическими значениями вероятностей исследуемых сложных событий.

Использование АК позволяет решать прямую и обратную задачу вероятностного анализа логических систем в многомерном пространстве, не ограничиваясь каким-либо одним классом распределений. В качестве маргинальных распределений при решении прямой и обратной задачи можно использовать не только нормальное распределение, но и любое другое.

Заключение

Приведенные результаты исследований показывают, что алгебра кортежей является математической системой, позволяющая унифицировать разнообразные структуры данных в интеллектуальных системах. Структуры алгебры кортежей хорошо согласованы с архитектурой современных компьютеров и обладают естественным параллелизмом, что позволяет сравнительно легко реализовать интеллектуальные системы в вычислительных комплексах параллельной обработки данных.

Список литературы

1. Кулик Б.А. Система логического программирования на основе алгебры кортежей // Известия РАН. Техническая. кибернетика. 1993. № 3. С. 226-239.
2. Кулик Б.А. Математическая модель дедуктивной базы данных на основе алгебры кортежей // Известия РАН. Техническая. кибернетика. 1994. № 2. С. 161-169.
3. Кулик Б.А. Новые классы КНФ с полиномиально распознаваемым свойством выполнимости // АиТ. 1995. № 2. С. 111-124.
4. Кулик Б.А. Представление логических систем в вероятностном пространстве на основе алгебры кортежей. 1. Основы алгебры кортежей // АиТ. 1997. № 1. С. 126-136.
5. Кулик Б.А., Наумов М. В. Представление логических систем в вероятностном пространстве на основе алгебры кортежей. 2. Измеримые логические системы // АиТ. 1997. № 2. С. 169–179.
6. Кулик Б.А. Анализ надежности систем с многими состояниями на основе алгебры кортежей // АиТ. 2003. № 7. С. 13-18.
7. Кулик Б.А. Курс лекций по алгебре кортежей. http://logic-cor.narod.ru
8. Бурбаки Н. Теория множеств. М.: Мир, 1965. 455 с.
9. Гэри М., Джонсон Л. Вычислительные машины и труднорешаемые задачи. М.: Мир. 1982. 352 с.
10. Рябинин И.А. Надежность и безопасность структурно-сложных систем. СПб.: Политехника. 2000. 248 с.
11. Кулик Б.А. Быстродействующие ИПС на основе операций с логическими векторами // Управляющие Системы и Машины. 1989. № 4. С. 14-19.
12. Липский В. Комбинаторика для программистов. М.: Мир. 1988. 231 с.
13. Чери С., Готлоб Г., Танка Л. Логическое программирование и базы данных. М.: Мир. 1992. 352 с.
14. Индейцев А.И., Сергеев А.Г. Система интеллектуальной поддержки борьбы за живучесть надводного корабля // Методы и средства информационной поддержки борьбы за живучесть надводных кораблей. СПб.: ИПМаш РАН. 1995. С. 15-35.
15. Nilsson N.J. Probabilistic Logic // Artificial Intelligence. 1986. No 28. P. 71-87.
16. Колмогоров А.Н. Основные понятия теории вероятностей. М.: Наука. 1974. 120 с.