Информационные процессы, Том 16, № 4, 2016, стр. 312–332

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Категорная технология создания и развития интеллектуальных систем, основанных на знании[1]

А.В.Жожикашвили

Институт проблем передачи информации, Российская академия наук, Москва, Россия
e-mail: zhozhik@iitp.ru

Поступила в редколлегию 27.10.2016

Аннотация—В статье рассмотрены определенного вида продукции, которые являются полезными для создания интеллектуальных компьютерных систем, основанных на знании. Построена математическая теория таких продукций, основанная на аппарате теории категорий. Изложена основаная на этой теории технология создания и развития продукционных систем.

КЛЮЧЕВЫЕ СЛОВА: представление знаний, продукции, теория категорий.

1. ВВЕДЕНИЕ

В первые десятилетия своего существования искусственный интеллект развивался как наука сугубо теоретическая. То, что основным инструментом для научных исследований являлось компьютерное моделирование, не меняло сути дела. Задачи, решаемые с помощью компьютера, не были в полной мере прикладными задачами. Если программа умела доказывать элементарную теорему, разумно переставлять кубики или понимать текст, состоящий из нескольких фраз, понятных ученику младших классов, это считалось серьезным достижением. Рассчитывать на что-то большее не позволяли ограниченные мощности тогдашних компьютеров.

Сейчас мощность компьютеров возросла на много порядков, и это серьезно изменило ситуацию. Программы, созданные на основе исследований по искусственному интеллекту, активно входят в жизнь, можно даже сказать – в быт. Слова artificial intelligence можно сейчас увидеть на пылесосе или стиральной машине. Может быть в следствии этого, может быть по каким-то общим законам развития науки фундаментальные исследования несколько отошли в тень. Исследователи больше сосредоточены на создании полезных продуктов, нежели на разработке общей теории. А такая теория нужна. Отсутствие теории рано или поздно остановит даже разработку приложений, которая уже становятся все более трудоемкой. Между тем, общие моменты в различных прикладных работах, безусловно, имеются, и они требуют теоретического осмысления.

Предлагаемая читателю статья является некоторым шагом в этом направлении. Излагаемый в статье материал – лишь часть теории, охватывающая небольшую область искусственного интеллекта. Если говорить точнее, статья охватывает широкий круг проблем искусственного интеллекта, но касается лишь одной стороны этих проблем. Однако эта сторона представляется автору достаточно важной.

Речь в работе пойдет об интеллектуальных системах, основанных на знаниях. Тема эта является одной из центральных в искусственном интеллекте. Она в той или иной мере касается большинства компьютерных интеллектуальных систем, ибо почти все они используют

[1] Работа частично финансировалась по проекту РФФИ 15-07-07486 и по программе N211 Президиума РАН.

различные методы работы со знаниями. По мнению автора, одним из основных действий, совершаемых интеллектуальными системами, основанными на знании, является узнавание ситуации, с которой встретилась система в процессе работы, и преобразование этой ситуации в соответствии с хранящейся в памяти системы информацией о ней. Для формального описания такого действия автором было использование понятие продукции, понимаемой в весьма общем смысле. Именно, была введена в рассмотрение продукция, называемая далее продукцией вида *сопоставление-конкретизация*, и построен формализующий такие продукции математический аппарат, основанный на языке теории категорий [1, 2, 3]. Описанные в статье продукции могут быть полезны при исследовании самых разных действий, связанных со знаниями.

Возможность описать различные сущности с единых позиций, на одном языке, могут быть полезны во многих отношениях. В теоретическом плане она позволяет ставить и решать задачи в общем виде, так что решения могут быть затем применены в различных исследованиях. В практическом плане такое единообразное описание позволяет строить компьютерные системы, которые, с незначительными изменениями, могут использоваться для решения существенно различных задач. В настоящей работе основное внимание уделено именно этому второму аспекту. В работе предлагается технология разработки продукционных систем, снижающая трудоемкость создания таких систем и позволяющая перенастраивать их для работы с разными структурами знаний [4]. Несмотря на прикладной характер исследований, связанных с этой технологией, при их проведении часто возникают теоретические, иногда чисто математические задачи. Для полноты изложения некоторые из этих задач также приведены в работе.

Структура работы следующая. В главе 2 вводится понятие продукции вида сопоставление-конкретизация и излагается математический аппарат, используемый для формализации этого понятия. Обсуждается место построенной теории в ряду теоретических исследований, посвященных продукционным системам и представлению знаний в искусственном интеллекте. В главе 3 описана собственно технология создания продукционных систем и приведены примеры ее использования.

2. ПРОДУКЦИЯ КАК ОДИН ИЗ ОСНОВНЫХ ЭЛЕМЕНТОВ СИСТЕМ, ОСНОВАННЫХ НА ЗНАНИИ

2.1. Классические схемы представления знаний

С семидесятых годов 20 века в искусственный интеллект пришло понимание того, что человек в своей интеллектуальной деятельности использует очень большие объемы знаний, значительно большие, чем, казалось бы, достаточно для решения поставленной им задачи. Это позволяет ему быстро ориентироваться в обстановке и находить путь к решению порой быстрее, чем компьютеру, формально превосходящему мозг человека по быстродействию. Тематика, связанная с представлением знаний, стала одной из основных на всех крупных конференциях. Появилось специальное понятие: системы, основанные на знаниях, и первые системы искусственного интеллекта, принесшие практическую пользу – экспертные системы – безусловно относятся к этому классу систем. С участием многих известных специалистов обсуждался вопрос о том, что такое знания, чем знания отличаются, скажем, от данных. Не вдаваясь в детали этого обсуждения, отмечу еще раз, какое свойство знаний было для автора статьи важным при построении описываемой теории. Знания – это не информация, являющаяся с формальной точки зрения необходимой для решения некоторой задачи, а информация, позволяющая быстро найти путь к этому решению. Если, например, говорить о математике, формально необходимой для решения задачи информацией являются аксиомы и правила вывода. Однако математик в процессе учебы изучает множество теорем и решает многочисленные задачи. В результате решения задач у него остаются воспоминания о разнообразных ситуациях и о том, какие действия в этих ситуациях приводили к успеху. Это и есть математический опыт, позволяющий математику быстро находить решения и по возможностей избегать рассмотрения тупиковых путей.

В работах по искусственному интеллекту был выработан ряд способов представления знаний и создана классификация этих способов, ядро которой сохранилось и поныне. Согласно этой классификации способы представления знаний делятся на продукционные – основанные на правилах, сетевые и фреймовые. Не подвергая сомнению обоснованность такой классификации, лучшим доказательством которой служит то, что она дожила до наших дней, рискну высказать к ней некоторые претензии. На взгляд автора статьи, указанная классификация характеризует не разные способы представления знаний, она характеризует разные свойства, которыми должна обладать структура, используемая для представления знаний, причем желательно – всеми сразу.

В чем основная идея М. Минского, из работы которого [5] выросла фреймовая модель представления знаний? Она состоит вовсе не в использовании определенных конструкций – фреймов и слотов. Идея состоит в том, что знания не должны представлять множество никак не сгруппированных элементов. Знания должны объединяться в блоки, соответствующие типичным ситуациям, с которыми сталкивается в своей деятельности человек или интеллектуальная компьютерная система. При этом объем и детализация блока напрямую зависит от того, насколько часто возникает в работе соответствующая ситуация и насколько она важна для достижения нужного результата. Фрейм – описание стереотипной ситуации, это – почти дословная цитата из упомянутой выше работы Минского.

Но эта идея не противоречит ни подходу, основанному на правилах, ни подходу, основанному на семантических сетях. Построение больших баз знаний, основанных на правилах, трудно представить себе без структурирования таких баз. Правила обязательно должны быть разделены на блоки, соответствующие подзадачам. Автор принимал участие в создании в коллективе В.Л. Стефанюка ряда экспертных систем, в которых правила разбивались на модули и каждый модуль подгружался по мере необходимости – см., например, [6, 7]. Ниже будет идти речь о продукционной сети, которая представляет собой совокупность отдельных продукционных баз.

Сетевая организация базы знаний также реализует некоторую общую идею: база знаний должна представлять не набор разрозненных фрагментов, а набор фрагментов, связанных определенным образом друг с другом, т.е. объединенных в сеть, отражающую связи, существующие между описываемыми этими фрагментами сущностями в реальном мире. Следование этой идее является уместной и при построении продукционных, и при построении фреймовых баз знаний. Минский в своей работе прямо говорит том, что интерес представляет не столько фрейм, сколько сеть фреймов. Упомянутая уже продукционная сеть привносит сетевые идеи в продукционную базу.

О том, какую общую идею реализует продукционный подход, будет сказано чуть ниже. Пока же можно сделать такой вывод: система представления знаний должна быть до некоторой степени и продукционной, и сетевой, и фреймовой. Другое дело, что для каждого из трех перечисленных способов представления знаний разработаны свои технологии и свой инструментарий. Когда разработчик говорит, что создал продукционную систему, это может означать, что он использовал технологию, предназначенную для разработки таких систем. Возможно, он пользовался соответствующими инструментами, скажем, языками для описания правил, или пустой оболочкой, основанной на правилах. При этом он вполне мог следовать и сетевым идеям организации знаний, но уже не опираясь на поддержку этих инструментов. Так, программист, пишущий на ассемблере, вполне может при этом мыслить в терминах структурного или объектного программирования, хотя ассемблер не имеет средств поддержки соответствующих архитектур. То, что имеющийся инструментарий требует от программиста выбора одной из схем представления знаний и в дальнейшем навязывает следование канонам этой схемы, можно считать некоторым недостатком существующий технологий создания систем, основанных на знаниях. Этот недостаток был одной из причин, побудившей автора взяться за то, что в этой работе называется категорной технологией создания интеллектуальных систем. Эта технология не связана с определенными схемами вывода. Она, правда, основана на использовании продукций, но, как будет видно из написанного далее, продукция понимается несколько более абстрактно, чем в классической теории продукционных баз знаний.

С таким абстрактным пониманием продукции связано название статьи. Хотя весь дальнейший материал связан с продукциями, автор посчитал не совсем правильным писать в названии о технологии создания продукционных систем. Дело в том, что в понимании многих специалистов "продукционная система" означает "система, осуществляющая вывод на продукциях", причем алгоритмы этого вывода являются достаточно устоявшимися. Разработка классической машины вывода относится, конечно, к задачам, решаемым с помощью категорной технологии, однако, как будет сказано далее в статье, возможность строить подобные машины не является самой выигрышной ее стороной. В разделе 2.5 перечислены четыре направления прикладных работ, связанные с категорной технологией, и построение классической машины вывода является только одним из них. Считая понимаемую достаточно абстрактно продукцию одним из основных элементов любой интеллектуальной системы, основанной на знаниях, автор посчитал более правильным говорить о технологии разработки именно таких систем.

2.2. Продукции вида сопоставление-конкретизация

Итак, выше было сказано, что и сетевые, и фреймовые подходы к представлению знаний основаны на некоторой глобальной идее о том, как устроены знания. Какую же глобальную идею развивает продукционный подход? На взгляд автора статьи, понятие продукции реализует основную особенность интеллектуальных систем, основанных на знании, которые отличают эти системы от интеллектуальных систем других видов. Встретившись с некоторой ситуацией, система, основанная на знании, не начинает разбирать ее с нуля, как это сделала бы, скажем, система логического вывода. Такая система, основанная, например, на методе резолюции Робинсона, выводит все из небольшого набора аксиом, строя довольно длинную цепочку умозаключений. В противоположность этому система, основанная на знании, должна узнать ситуацию, т.е. найти в базе знаний описание этой ситуации, а вместе с этим найти в базе знаний рекомендации о том, что в этой ситуации следует предпринять. Именно использование такой операции является отличительной особенностью систем, основанных на знаниях. Такая система, следовательно, должна хранить в памяти описание большого числа типовых ситуаций, с которыми она может встретиться в своей работе, и описание действий, которые нужно или можно совершить в этой ситуации.

Вот этот основной элемент базы знаний – описание ситуации и способ преобразования ее в новую ситуацию – и есть продукция в том смысле, который придается этому понятию в статье. Продукция, изучаемая в статье, состоит из левой и правой части, причем левая часть – это описание ситуации, к которой продукция применима, правая часть – описание ситуации, возникающей после ее применения. Продукция, следовательно, преобразует ситуацию, заменяя старую, которую описывает левая часть, на новую, которую описывает правая.

Что такое "описание ситуации"? Если левая часть продукции описывает в точности одну ситуацию, такая продукция может быть применена только к этой ситуации. Полезность такой продукции невелика. Правильнее требовать от продукции, чтобы она была применима к целому классу сходных в чем-то ситуаций. Достигается это тем, что описание ситуации, стоящее в левой части продукции, не является полным. Оно описывает какую-то часть ситуации, важную для того, чтобы продукция была бы применима. Другие же ее части могут варьироваться.

Такое неполное описание ситуации будем называть *образцом.* Образец, таким образом, описывает некоторое множество ситуаций, у которых охватываемая им часть совпадает, в других же частях они могут отличаться друг от друга. Если некоторая ситуация входит в это множество, будем говорить, что она подходит под образец. Операция, определяющая, подходит ли ситуация под образец, называется операцией *сопоставления* ситуации с образцом. Если ситуация подходит под описание, задаваемое образцом, будем говорить, что сопоставление прошло успешно, ситуация сопоставима с образцом, в противном случае сопоставление окончилась неудачей, ситуация не сопоставима с образцом.

Поскольку образец – неполное описание ситуации, всякая сопоставимая с ним ситуация отличается от образца тем, что содержит больше информации. Таким образом, ситуация, сопоставимая с образцом, может быть получена добавлением к образцу недостающей информации. Процесс добавления к образцу недостающей информации называется *конкретизацией* образца. В результате конкретизации образец превращается в ситуацию[2]. Ситуация, таким образом, сопоставима с образцом, если образец может быть конкретизирован таким образом, чтобы в результате получилась эта ситуация.

Теперь мы можем уточнить определение продукции. Продукция состоит из пары образцов, называемых левой и правой частью продукции. Продукция применима в некоторой ситуации (или, как удобнее говорить, применима к некоторой ситуации, ибо продукция имеет целью преобразовать ситуацию), если эта ситуация сопоставима с левой частью продукции. В этом случае результатом применения продукции является ситуация, получающаяся из правой части, если конкретизировать ее так же, как была конкретизирована левая часть при сопоставлении с ситуацией.

Итак, мы можем, хоть и в весьма общих терминах, сказать о том, как устроена продукция и как она действует. Именно, продукция представляет собой пару образцов, а действие продукции представляет собой последовательное выполнение двух операций – сопоставления и конкретизации. Продукции, устроенные таким образом, и называются в статье продукциями вида *сопоставление-конкретизация.*

2.3. Связь с классическим пониманием продукции

Продукционные системы являются, пожалуй, наиболее распространенными и наиболее исследованными системами представления знаний. В искусственном интеллекте встречались различные определения термина "продукция", но суть их была одна. Именно, под продукцией понималась конструкция, состоящая из двух компонент: условия, которое должно было быть проверено с целью установить, применима ли продукция, и действия, которое следовало совершить, если условие оказывалось выполненным. Как соотносится это определение – условия применимости-действие – с тем, которое было приведено выше – сопоставление-конкретизация? В более ранних работах автор часто использовал термин "обобщенная продукция", имея в виду то, что определенная им продукция не связана с конкретными структурами данных, а основана на общих понятиях, таких, как ситуация, образец, сопоставление. Со временем, однако, он пришел к выводу, что продукция в терминологии данной статьи является не более общим, а, наоборот, более специальным понятием, чем классическая продукция, изучае-

[2] Стефанюк отмечал, что такое понимание означает, что образец описывает всю ситуацию, хотя и не во всех деталях. В реальности часто бывает нужно, чтобы образец описывал часть ситуации (опять же не во всех деталях), важную в тех задачах, для решения которых он предназначен. Конкретизация образца, следовательно, превращает его не в ситуацию, а в часть этой ситуации. Хотя с абстрактно-теоретической точки зрения формулировки "описывает часть ситуации" и "описывает ситуацию на во всех деталях" мало чем отличаются, в практическом отношении смысл у этих формулировок достаточно разный. Это соображение показывает, что теория еще далека от завершения, ее, возможно, следует развить с целью получить возможность описать и такую конкретизацию.

мая в искусственном интеллекте. Ведь продукция вида сопоставление-конкретизация подходит под общее определение: сопоставление с образцом можно рассматривать, как проверку условия применимости, а конкретизацию – как действие, дающее новую ситуацию. Но обратное не верно: не всякое условие можно описать как сопоставление и не всякое действие – как конкретизацию (по крайней мере так, чтобы это не приводило к неоправданно сложным конструкциям). Продукция вида сопоставление-конкретизация предназначена для выполнения весьма специфической операции: узнавания ситуации и замену ее другой ситуацией, в общих чертах описанной в продукции, но с отдельными деталями, взятыми из исходной ситуации. Повторю, однако, что подобная операция является одной из основных операций, используемых в своей работе интеллектуальной системой, основанной на знании. Это подтверждается и тем, что многие продукции, встречающиеся в таких системах, подходят под сформулированное определение.

Идея использовать в левой части сопоставление с образцом – не нова, целый ряд авторов в качестве первой компоненты классической продукции – условия применимости – рассматривают именно процедуру сопоставления. Однако мысль о том, что и в качестве правой части может выступать образец, является абсолютно новой.

2.4. Продукции на языке множеств и отображений

Выше было сказано следующее: результатом применения продукции является ситуация, получающаяся из правой части, если конкретизировать ее так же, как была конкретизирована левая часть при сопоставлении с ситуацией. Одной из целей, которая преследовалась при построении описываемой в статье теории, была формализация термина "конкретизировать так же". Это было сделано следующим образом.

Информацию, которая добавляется к образцу при его конкретизации, будем называть *конкретизатором.* С каждым образцом связано множество допустимых конкретизаторов. Пусть задано множество ситуаций S и образец, который можно сопоставлять с ситуациями из множества S. Обозначим множество допустимых конкретизаторов этого образца символом X. Каждый конкретизатор из множества X превращает образец в ситуацию из множества S. Таким образом, образец определяет отображение $\phi\colon X \to S$. Это отображение мы будем называть S-образцом с областью конкретизаторов X. Если $s \in S$ – ситуация, она является сопоставимой с образцом ϕ если $s \in \mathrm{Im}(\phi)$, т.е. если для некоторого $x \in X$ имеем $\phi(x) = s$.

Теперь мы можем дать более точное определение продукции. Пусть S и T – два множества ситуаций. Продукцией из S в T называется пара образцов $\phi\colon X \to S$ и $\psi\colon X \to T$ с совпадающей областью конкретизаторов. Продукция применима к ситуации $s \in S$, если эта ситуация сопоставима с ее левой частью – образцом $\phi\colon X \to S$, т.е. если для некоторого $x \in X$ имеем $\phi(x) = s$. В этом случае результатом применения продукции является ситуация $\psi(x) \in T$. Таким образом, продукция переводит ситуации из множества S в ситуации из множества T. Наиболее интересным является случай $S = T$. В этом случае продукция преобразует ситуацию из множества S в новую ситуацию из того же множества, и к ней можно применять следующую продукцию.

Отметим, что мы не требуем инъективности отображения ϕ, поэтому элемент $x \in X$ определяется равенством $\phi(x) = s$ неоднозначно, а следовательно, неоднозначно определяется и $\psi(x)$, т.е. результат действия продукции. Продукция показывает лишь возможные переходы от ситуации к ситуации, и с продукцией может быть связано несколько возможных переходов.

2.5. Что дает теория продукций вида сопоставление-конкретизация?

Переход от понятия “продукция” в общем смысле к продукциям вида сопоставление-конкретизация, т.е. сосредоточение внимания на некотором специальном виде продукций, дает возможность построить более развитую теорию таких продукций. Во-первых, они поддаются хорошей математической формализации, о чем будет рассказано ниже. Во-вторых, понимаемые так продукции и соответствующую математическую теорию можно использовать в более широком круге задач, чем продукции в классическом понимании. Последние очень удобны для теоретического исследования классических алгоритмов вывода на продукциях [8]. Однако поскольку автор считает продукцию в изложенном выше понимании основным элементом любой системы представления знаний, он должен иметь такую теорию продукций, которая позволит исследовать более широкий круг операций со знаниями. Вот примеры нескольких направлений исследования, в которых может быть полезна излагаемая в работе теория.

Наряду с обычной логикой работы продукционной системы может быть рассмотрена какая-либо более сложная схема вывода, например, алгоритм вывода на продукционной сети, изложенный ниже в работе. В терминах классического определения продукции – условие применимости-действие – такой алгоритм описан быть не может, ибо он требует прерывания проверки условия применимости и продолжения этой проверки после использования дополнительных продукций, а при осуществлении указанного в продукции действия должны быть использованы результаты применения этих дополнительных продукций. Изложенный в работе подход к продукциям позволяет все это сделать, ибо дает более детальное описание как строения продукции, так и алгоритма ее применения.

В своих исследованиях автор занимался задачей анализа базы знаний. Такой анализ может преследовать различные цели. Например, такой целью может быть оптимизация базы знаний путем удаления из нее лишних продукций, т.е. таких продукций, что любой сделанный с их помощью вывод может быть получен и без их использования. Некоторые полезные для таких исследований операции над продукциями описаны ниже, их строгое определение стало возможным в результате перехода к пониманию продукции, развитому в работе, и построению соответствующего математического формализма.

Наконец, в рамках излагаемой теории могут формулироваться не только алгоритмы использования знаний, но и алгоритмы автоматического формирования знаний, в частности, построения продукций. Эта тема тоже будет освещена ниже в работе.

Но прежде чем переходить к описанию этих приложений описываемой в работе теории изложим математический аппарат, на котором эта теория построена.

2.6. Продукции на языке теории категорий

В сложной базе знаний разные множества могут играть роль множества ситуаций и множества конкретизаторов для разных образцов. Множество конкретизаторов одного образца может выступать в качестве множества ситуаций для другого. Будем говорить, что задана система образцов, если определено, какие множества могут выступать в качестве множества ситуаций и множества конкретизаторов, и какие отображения допускаются в качестве образцов. Потребуем, чтобы класс всех таких отображений был бы замкнут относительно композиции и включал в себя тождественные отображения. Это означает, что задать систему образцов – значит задать конкретную категорию $\mathbf{C}$, т.е. подкатегорию категории множеств и отображений. При этом образцом можно считать любой морфизм $\phi \in \mathbf{C}(X, Y)$ этой категории.

С теоретической точки зрения разумно отказаться от требования, чтобы категория $\mathbf{C}$ была бы подкатегорией категории множеств. Это тем более является правильным, ибо некоторые важные с прикладной точки зрения системы образцов описываются категориями, которые формально не являются подкатегориями категории множеств. К таковым, например, относится P-кратная категория, о которой пойдет речь ниже. Однако если объекты категории не являются множествами, а морфизмы – отображениями множеств, нельзя считать ситуациями элементы множества S. Поэтому в дальнейшем ситуация будет считаться частным случаем образца, т.е. тоже морфизмом. Это вполне согласуется с пониманием продукции и образца на идейном уровне. Действительно, если образец отличается от ситуации тем, что в нем недостает некоторой информации, то ситуация – это образец, в котором эта недостающая информация – нулевая.

Все это приводит к следующим определениям. Системой образцов называется категория $\mathbf{C}$, в которой для каждой пары объектов X и Y определено некоторое множество $\mathbf{C}_{\mathrm{s}}(X, Y) \subset \mathbf{C}(X, Y)$ морфизмов, причем выполнено следующее *условие ситуационной замкнутости*: если заданы морфизмы $\phi\colon X \to S$, $\psi\colon Y \to S$ и $\sigma\colon X \to Y$ такие, что $\phi = \psi\sigma$, то $\phi \in \mathbf{C}_{\mathrm{s}}(X, S)$ тогда и только тогда, когда $\psi \in \mathbf{C}_{\mathrm{s}}(Y, S)$. S-образцом будем называть морфизм $\phi\colon X \to S$ где X – произвольный объект категории, S-ситуацией – морфизм $\alpha\colon X \to S$, $\alpha \in \mathbf{C}_{\mathrm{s}}(X, S)$. Объект X будем называть областью конкретизаторов S-образца ϕ. Ситуация $\alpha\colon A \to S$ считается сопоставимой с образцом $\phi\colon X \to S$, если для некоторого морфизма $\beta\colon A \to X$ имеем $\alpha = \phi\beta$.

Продукцией из S в T называется пара образцов – S-образец и T-образец – с общей областью конкретизаторов, т.е. пара морфизмов $\phi\colon X \to S$ и $\psi\colon X \to T$. Такая продукция действуют на S-ситуации, превращая их в T-ситуации. Продукция (ϕ, ψ) считается применимой к ситуации $\alpha\colon A \to S$, если эта ситуация сопоставима с образцом ϕ, т.е. $\alpha = \phi\beta$ для некоторого $\beta\colon A \to X$. Результатом применения продукции к ситуации α в этом случае считается ситуация $\psi\beta\colon A \to T$ (этот морфизм является ситуацией в силу условия ситуационной замкнутости).

К такой теоретико-категорной формализации автор шел довольно долго, перебрав целый ряд иных математических конструкций. Одной из задач, к которой он постоянно возвращался в своих исследованиях, была задача автоматического построения базы знаний на основе изучения конкретных примеров. В этих исследованиях основным механизмом для построения знаний выступал процесс обобщения. Суть этого процесса в следующем. Пусть заданы две ситуации, описываемые на каком-то языке выражениями a и b, и пусть c является обобщением как a, так и b. Если некоторой действие приводит к цели в каждой из ситуаций a и b, разумно выдвинуть гипотезу, что это действие приводит к цели и в ситуации, описываемой выражением c. При этом возможны, конечно, неоправданно широкие обобщения, и чтобы уменьшить риск получить такое обобщение, следует выбирать в качестве c наименее общее обобщение a и b. Все эти рассуждения хорошо формализуются на языке теории решеток. Определим порядок на множестве выражений так: $a \leq c$ означает, что c является обобщением a. Тогда в качестве обобщения a и b следует взять их точную верхнюю грань $a \vee b$.

Теория решеток и была использована на первом этапе для формализации подобных рассуждений. Однако при переходе от отдельных образцов к продукциям, т.е. к парам образцов, автор стал ощущать некоторую бедность языка теории решеток. Для того, чтобы применить продукцию $c \to d$ к ситуации a мало знать, что a есть частный случай c, т.е. что $a \leq c$. Нужно знать, каким образом a получается из c, т.е. в чем состоит конкретизация. Другими словами, вопрос “$a \leq c$?” должен иметь не два возможных ответа – да, или нет, а много разных ответов, показывающих, как именно следует уменьшить c, чтобы получить a. Всякий, изучавший теорию категорий, знаком с классическим примером малой категории: объекты – элементы частично упорядоченного множества, если a и b – объекты, то $\hom(a, b)$ состоит из единственного элемента, если $a \leq b$ и пусто в противном случае. Обратно, если в малой категории между любой парой объектов есть не более одного морфизма, эта категория задается частичным порядком описанным выше способом. Таким образом, решетка – это частный случай малой категории, в которой поиск морфизма между парой объектов имеет только два результата: есть морфизм, или нет, в произвольной же категории он может иметь много результатов, находя один из существующих морфизмов. Как раз этого и не хватало в языке теории решеток. Так произошел переход на язык теории категорий. Дальнейшая работа доказала разумность такого выбора. В частности, многие конструкции, имеющие смысл в области представления знаний, оказываются близки к теоретико-категорным конструкциям или строятся на основе последних, что в некоторой степени будет продемонстрировано и в оставшейся части этой работы.

Для того, чтобы перевести на теоретико-категорный язык продукции, ориентированные на решение какой-то задачи, нужно построить категорию, описывающую структуру знаний, используемых для решения этой задачи. В публикациях, посвященных этой теме, были построены категории, пригодные для формализации весьма разнообразных видов образцов и продукций – см., например уже цитированную работу [3]. Среди них были продукции, используемые в интеллектуальных компьютерных системах, основанных на знании, в частности, в экспертных системах, позволяющие вывести новые факты из установленных ранее. Были продукции, строящие некоторые выражения на основе уже построенных. Такие продукции используются в системах формально-логического вывода, скажем, систем автоматического доказательства теорем. Были продукции, преобразующие строки символов. Такие продукции полезные в задачах понимания текстовой информации. Разнообразие примеров использования теоретико-категорного языка довольно велико, что показывает широкую применимость построенного формализма.

3. КАТЕГОРНАЯ ТЕХНОЛОГИЯ

3.1. Два уровня разработки продукционной системы

Остальная часть работы посвящена описанию развитой автором технологии разработки и развития продукционных систем представления знаний, называемой в работе категорной. Суть этой технологии состоит в том, что работы, связанные с созданием такой системы, делятся на два уровня – верхний и нижний. Речь идет не только о разработке алгоритмов, но и о реальном программировании системы.

На верхнем уровне алгоритмы и программы реализуются в терминах теории категорий. Базовые элементы, на которых основаны эти алгоритмы, элементарные действия, из которых они строятся, должны соответствовать теоретико-категорным операциям, таким например, как композиция морфизмов. Ниже этот тезис будет продемонстрирован на простых примерах.

Нижний уровень представляет собой программную реализацию категории, наиболее адекватно описывающей структуру знаний, соответствующую данной задаче. Программно реализовать категорию – это значит определить структуры данных, описывающие морфизмы категорий, и определить процедуры, выполняющие операции над этими структурами. Среди этих операций обязательно должна быть композиция, без которой категория не определена, но могут быть и другие операции, которые требуются для работы того или иного алгоритма, работающего со знаниями.

Надо отметить, что все работы, и верхнего, и нижнего уровня, касаются алгоритмов работы со знаниями, а не содержания этих знаний. Так, перенос продукционной экспертной системы на новую предметную область не меняет ни строения правил, ни алгоритмов работы с этими правилами. Меняется только содержание правил. Формирование списка правил – это работа инженера по знаниям, не относящаяся ни к верхнему, ни к нижнему уровню. Специалисты по разработке продукционной системы должны лишь сообщить инженеру по знаниям синтаксис правил, который последний должен использовать для кодирования знаний.

Специалист нижнего уровня должен включиться в работу в том случае, если меняется само строение ситуаций, образцов и продукций. Автор статьи перестроил систему, выводящую новые факты из ранее установленных, превратив ее в систему обработки текстов, заменяющую некоторые фрагменты этих текстов другими. Это было сделано заменой категории, т.е. эта работа относилась к нижнему уровню. Другой случай, когда потребовалась модификация нижнего уровня – переход к нечетким знаниям, где каждый факт может быть установлен лишь с некоторой степенью уверенности. Об этом говорится в разделе 3.6. Категорная технология позволила сделать это не меняя алгоритмы вывода, т.е. без проведения работ на верхнем уровне.

Необходимость же в проведении работ верхнего уровня возникает тогда, когда меняются сами алгоритмы работы с продукциями. Пример – перехода от прямого вывода – от данных, к обратному – от цели. Другой пример – переход к неклассическим алгоритмам вывода, чему посвящен раздел 3.3. Наконец, на верхнем уровне реализуются алгоритмы, осуществляющие не вывод на продукциях, а что-то иное – см. разделы 3.4 и 3.5.

Далее мы рассмотрим конкретные задачи, решаемые на каждом уровне.

3.2. Верхний уровень – вывод на продукциях

Начнем с обсуждения верхнего уровня и рассмотрим, какого рода алгоритмы на нем могут быть реализованы. Остановимся первым делом на алгоритмах классического вывода на продукциях. Один шаг такого вывода состоит в определении того, применима ли продукция к данной ситуации, и, если да, в применении этой продукции. Как уже говорилось, продукция (ϕ, ψ), $\phi\colon X \to S$, $\psi\colon X \to T$, считается применимой к ситуации $\alpha\colon A \to S$, если $\alpha = \phi\beta$ для некоторого $\beta\colon A \to X$, и результатом применения продукции в этом случае считается ситуация $\psi\beta\colon A \to T$. Мы видим, что программисту, реализующему операцию применения продукции к ситуации, должны быть доступны две операции, реализованные на морфизмах: бинарная операция – композиция, и бинарная операция, строящая по морфизмам α и ϕ морфизм β такой, что $\alpha = \phi\beta$. Последняя операция не является ни всегда выполнимой, ни однозначной. Ее программная реализация может, например, выдавать список значений, возможно – пустой. На этих двух операциях и должен быть построен алгоритм вывода. Чтобы он заработал, надо проделать работы нижнего уровня, т.е. закодировать морфизмы и реализовать обе операции так, чтобы это соответствовало структуре знаний, используемых в решаемой задаче. Перенос того же алгоритма на другую задачу потребует перепрограммирования категории.

Способность записывать классические алгоритмы вывода на продукциях не является главной выигрышной стороной категорной технологии. Классическое понимание продукции как условия применимости-действие позволяет сделать практически то же самое. Алгоритм вывода может базироваться на двух операциях: проверке условия применимости и выполнения действия, связанного с продукцией. Как только эти две операции будут запрограммированы соответственно потребностям задачи, алгоритм можно будет использовать для ее решения. Тем не менее, даже для таких алгоритмов категорная технология имеет ряд достоинств. Во-первых, сопоставление с образцом – операция более низкого уровня, чем произвольное действие, производимое продукцией. На нашем языке действие – это последовательное выполнение операции сопоставления и конкретизации, т.е. более сложное понятие – действие, разбито на два более простых, причем однотипных, решающих одну и ту же задачу в двух противоположных направлениях. Более простые операции легче программировать и особенно переносить на другие задачи. Описываемая в статье категорная технология не зря названа технологией создания и развития. Возможность развития системы, т.е. переноса ее на решение более сложных задач, является сильной стороной этой технологии.

Отметим также деталь, хорошо понятную программистам. В описании действия продукции вида сопоставление-конкретизация явно указано, что при конкретизации используется морфизм, найденный в результате сопоставления. В классическом понимании – условие применимости-действие – ничего не говорится о том, что действие должно использовать результаты проверки условия (не считая, естественно, того, что результаты проверки условия применимости определяют, надо ли вообще совершать действие). Однако в реальных задачах часто бывает так, что проверка условия применимости – это уже половина действия. Скажем, если продукция представляет собой продукцию Поста и ее задача – заменить в строке одну подстроку другой, то проверка условия применимости состоит в поиске подстроки, действие же не будет включать в себя повторный поиск подстроки, оно произведет замену того, что уже найдено при проверке условия применимости.

3.3. Верхний уровень – продукционные сети

Более интересной особенностью категорной технологии является предоставляемая ей возможность создавать неклассические алгоритмы вывода на продукциях. Ниже описан вывод на системе продукций, организованных определенным образом. Такая организация, более сложная, чем просто список продукций, была названа автором продукционной сетью [9].

Большинство продукционных систем имеют одну общую черту: продукции используются независимо друг от друга. После того, как завершила работу очередная продукция, осуществляется выбор продукции, которая будет применена на следующем шаге, а затем и применение этой продукции. Действие продукции зависит только от текущей ситуации, и не зависит от других продукций базы.

Такой подход имеет как сильные, так и слабые стороны. Достоинства его состоят в том, что продукционная база знаний создается проще, чем базы, основанные на других способах представления знаний. Продукционная база знаний легко пополняется. Можно добавить новую продукцию, не меняя ничего в уже существующих. Можно легко собирать базу знаний из фрагментов, созданных разными специалистами, полученных разными путями, из разных источников. Именно простота продукционных баз знаний и привела к их столь широкому распространению в прикладных системах. Однако эта же простота стала некоторым затруднением на пути их дальнейшего развития. Для того чтобы обеспечивать сложное поведение системы столь простыми средствами, приходилось максимально увеличивать количество продукций в базе. Оказалось, что, когда объем базы становился достаточно большим, система становилась неконтролируемой, на практике она часто вела себя не так, как рассчитывали создатели, и отладка такой системы оказывалась крайне сложным делом.

Все эти сложности имеют достаточно глубокие корни. Продукционные системы реализуют идею разбиения сложной задачи на последовательность простых шагов, состоящих в применении отдельных продукций. Такая система является двухуровневой: верхний уровень сложности – сама решаемая задача, нижний – отдельная продукция. Но скачок с наиболее высокого уровня сложности – уровня задачи, решаемой системой, к наиболее низкому – уровню подзадачи, описываемой отдельной продукцией, является слишком стремительным. Человек в своей мыслительной деятельности поступает более сложно. Он тоже может свести решение исходной задачи к цепочке подзадач, но если исходная задача была сложной, то и эти подзадачи могут оказаться сложными. Тогда они, в свою очередь, разбиваются на последовательность еще более простых подзадач и т.д. Таким образом, подзадачи образуют не линейную последовательность, а дерево, распространяющееся и вширь, и вглубь.

Продукционная сеть реализует такое использование продукций. В продукционной сети продукция в процессе выполнения может инициировать применение других продукций. В случае если их применение приводит к успеху, она выполняется, причем в своей работе использует результаты, полученные этими другими продукциями.

Приведем соответствующие теоретико-категорные определения. Пусть задана категория $\mathbf{C}$. Под продукционной базой понимается тройка (S, T, P), где S и T – объекты категории, а P –

множество продукций, действующих из S в T. Продукции, входящие в продукционную базу, могут быть двух видов. Простая продукция, действующая из S в T – это продукция в обычном смысле, т.е. пара морфизмов $\phi\colon X \to S$ и $\psi\colon X \to T$. Сложная продукция из S в T представляет собой тройку (ϕ, ψ, P), где $\phi\colon X \to S$ и $\psi\colon Y \to T$ – образцы, а P – продукционная база, действующая из X в Y. Мы, таким образом, сохраняем понимание продукции как составленной из двух образцов, но отказываемся от требования к этим образцам иметь общую область конкретизаторов, заменяя его требованием существования соответствующей продукционной базы.

Алгоритм применения продукции к ситуации следующий. Простая продукция применяется обычным способом. Сложная продукция (ϕ, ψ, P) применяется к ситуации $\alpha\colon U \to S$ следующим образом. Ситуация α сопоставляется с левой частью продукции – с образцом $\phi\colon X \to S$. Если сопоставление невозможно – продукция неприменима. Пусть в результате сопоставления найден морфизм $\alpha'\colon U \to X$ такой, что $\alpha = \phi\alpha'$. В силу условия ситуационной замкнутости, этот морфизм сам будет ситуацией. К этой ситуации может быть применена продукционная база P. Если эта операция не приведет к успеху, исходная продукция считается не применимой к ситуации, если же результатом применения продукционной базы P к ситуации α' окажется ситуация $\beta'\colon V \to Y$, то результатом применения исходной продукции к ситуации α считается ситуация $\beta = \psi\beta'\colon V \to T$.

Алгоритм применения сложной продукции является рекурсивным. Для того, чтобы рекурсия не была бы бесконечной, при построении базы должно быть обеспечено, чтобы каждая цепочка вызывающих друг друга продукций заканчивалась бы простой продукцией.

Продукционной сетью называется множество продукционных баз, одна из которых выбрана в качестве начальной базы. Эта база в первую очередь применяется к ситуации, но в процессе рекурсивного применения могут быть вовлечены и другие.

Важную роль в продукционных сетях играет операция разложения в прямое произведение. Она возникает в тех случаях, когда конкретизатор, полученный при сопоставлении ситуации с образцом, распадается на несколько частей, каждая из которых должна быть проанализирована независимо. Приведем соответствующие определения.

Будем понимать под n-арным образцом морфизм $\phi\colon X_1 \times X_2 \times ... \times X_n \to S$, где знаком $\times$ обозначено прямое произведение. Отметим, что это не какое-то новое определение образца, а обычный образец, область конкретизаторов которого имеет специальный вид. Пусть $\alpha\colon U \to S$ – ситуация. Если она сопоставима с образцом ϕ, т.е. $\alpha = \phi\beta$ для некоторой ситуации $\beta\colon U \to X_1 \times X_2 \times ... \times X_n$, то ситуация β может быть представлена набором из n ситуаций $\beta_i\colon U \to X_i$, $\beta_i = \pi_i\beta$, где $\pi_i\colon X_1 \times X_2 \times ... \times X_n \to X_i$ – проекция прямого произведения на компоненту. Теперь n-арную продукцию можно определить как объект, состоящий из двух n-арных образцов $\phi\colon X_1 \times X_2 \times ... \times X_n \to S$ и $\psi\colon Y_1 \times Y_2 \times ... \times Y_n \to T$ и набора продукционных баз $(P_1, P_2, ..., P_n)$, где P_i действует из X_i в Y_i.

Как видно из этого описания, для реализации алгоритма вывода на продукционной сети нужны те же две операции, что и для классического вывода, плюс операция разложения в прямое произведение. Последняя операция, как известно, реализуема не в любой категории. Это означает, что, построив категорию, соответствующую решаемой задаче, разработчик должен выяснить вопрос, является ли эта категория категорией с прямыми произведениями. Если нет, приходится признать, что использование продукционной сети в данной задаче невозможно. Эта ситуация является типичной для категорной технологии. Создатель алгоритма должен выписать список требований, которым должна удовлетворять категория, чтобы этот алгоритм был бы в ней реализуем, и разработчик, планирующий применить этот алгоритм, должен проверить, удовлетворяет ли его категория этим требованиям.

3.4. Верхний уровень – операции над образцами и продукциями

Выше уже говорилось, что на категорном языке могут быть описаны некоторые операции над образцами и продукциями, полезные в некоторых задачах манипулирования знаниями.

Начнем с того, что свойства как образцов, так и продукций, могут быть разделены на две группы. К первой группе отнесем свойства, основанные на том, как устроены эти образцы и продукции. Ко второй – свойства, описывающие, с какими ситуациями эти образцы сопоставимы и как эти продукции действуют на ситуации. Свойства первого вида будем называть *интенсиональными*, свойства второго вида – *экстенсиональными.*

Образец был задуман как способ описания некоторого множества ситуаций. В то же время реальные образцы могут быть заданы выражениями, записанными на некотором языке. Как правило, они представляют собой некоторые синтаксические конструкции. Образец, как описание множества ситуаций, и образец, как синтаксическая конструкция – объекты, тесно связанные друг с другом, но не тождественные. К примеру, разные по строению образцы могут описывать одно и то же множество ситуаций.

То же самое можно сказать о продукциях. Продукция, так же как образец, представляет собой некоторую синтаксическую конструкцию. Эта же продукция задает некоторый алгоритм преобразования ситуации.

Обычно, интенсиональные свойства образцов и продукций проверяются намного проще, чем экстенсиональные свойства, ибо проверка первых требует рассмотрения самого образца, а проверка вторых – рассмотрения множества (возможно, бесконечного) сопоставимых с ним ситуаций. Однако поскольку образец служит для описания множества ситуаций, а продукция – для преобразования ситуаций, экстенсиональные свойства в некотором смысле являются более важными. Поэтому естественно желание установить связь между интенсиональными и экстенсиональными свойствами. Многие операции очевидным образом определяются экстенсионально, в то же время аналогичное интенсиональное определение бывает менее очевидным. Это хорошо видно на рассмотренном ниже примере композиции продукций. Теоретико-категорный язык часто позволяет доказать, что те или иные экстенсиональные операции сводятся к интенстональным, последние же могут быть реализованы с помощью стандартных операций теории категорий. Это позволяет описывать соответствующие операции в терминах верхнего уровня и включать их в программы, работающие со знаниями, для которых реализована соответствующая категория.

Приведем примеры сказанному выше. Начнем с образцов. Образец был введен как способ описания некоторого множества близких ситуаций. Можно, следовательно, сравнивать образцы, сравнивая те множества, которые они описывают.

Пусть $\phi\colon X \to S$ – образец. Обозначим символом S_ϕ множество S-ситуаций, сопоставимых с образцом ϕ.

Определение. Образец ϕ является *частным случаем* образца ψ, если $S_\phi \subset S_\psi$.

Это определение является экстенсиональным. Дадим его интенсиональный аналог.

Определение. Пусть $\phi\colon X \to S$ и $\psi\colon Y \to S$ – два образца. Будем говорить, что первый из них сопоставим со вторым, если существует морфизм $\chi\colon X \to Y$ такой, что $\phi = \psi\chi$. В этом случае будем также называть образец ϕ *подобразцом* образца ψ.

Определение полностью повторяет данное ранее определение сопоставимости ситуации с образцом, которое теперь может рассматриваться как частный случай сопоставимости двух образцов.

Теорема. *Если образец $\phi\colon X \to S$ сопоставим с образцом $\psi\colon Y \to S$ то всякая ситуация, сопоставимая с образцом $\phi\colon X \to S$, сопоставима и с образцом $\psi\colon Y \to S$, т.е. если один образец является подобразцом другого, то он является также его частным случаем.*

Доказательство Пусть образцы $\phi\colon X \to S$, $\psi\colon Y \to S$ и морфизм $\chi\colon X \to Y$ удовлетворяют условиям определения. Если ситуация $\alpha\colon A \to S$ такова, что $\alpha \in S_\phi$, т.е. для некоторого морфизма $\beta\colon A \to X$ имеем $\alpha = \phi\beta$, то морфизм $\chi\beta\colon A \to Y$ удовлетворяет равенству $\alpha = \psi\chi\beta$, откуда $\alpha \in S_\psi$. □

Обратное утверждение, вообще говоря, не верно, однако оно верно во многих важных частных случаях.

Перейдем теперь к продукциям. В качестве примера того, как экстенсиональные операции, определенные на продукциях, сводятся к интенсиональным, рассмотрим операцию композиции продукций.

Естественно определить композицию так.

Определение. Композиция двух продукций – это такая продукция, которая преобразует ситуацию α в ситуацию β в том и только в том случае, когда существует такая ситуация γ, что первая продукция преобразует α в γ, а вторая – γ в β.

Это определение, очевидно, является экстенсиональным. Дадим интенсиональный его аналог. Для того, чтобы это было возможно, будем считать, что в категории существуют декартовы квадраты. В работе [3] доказана следующая теорема.

Теорема. *Пусть* $S \xleftarrow{\phi} X \xrightarrow{\sigma} S$ *и* $S \xleftarrow{\psi} Y \xrightarrow{\tau} S$ *– продукции, а*

$$\begin{array}{ccc} Z & \xrightarrow{\mu} & Y \\ \downarrow{\scriptstyle\lambda} & & \downarrow{\scriptstyle\psi} \\ X & \xrightarrow{\sigma} & S \end{array}$$

декартов квадрат. Тогда продукция $S \xleftarrow{\phi\lambda} Z \xrightarrow{\tau\mu} S$ *является композицией продукций* $S \xleftarrow{\phi} X \xrightarrow{\sigma} S$ *и* $S \xleftarrow{\psi} Y \xrightarrow{\tau} S$.

Построение декартова квадрата – классическая операция теории категорий. Если доказано, что в категории, описывающей некоторую структуру знаний, существуют декартовые квадраты и написан алгоритм для их построения, открывается вариант для построения композиции. Построение же декартового квадрата само может быть сведено к более элементарным операциям, например, построению произведений и уравнителей. Это – известный факт теории категорий.

Отметим, что обратить высказанное выше утверждение нельзя, композиция не всегда имеет указанный вид.

3.5. Верхний уровень – обобщение образцов

На излагаемом в работе теоретико-категорном языке могут быть описаны не только алгоритмы вывода, т.е. получения результатов на основе знаний, но и алгоритмы автоматического формирования новых знаний. Как уже отмечалось, основным инструментом для такого формирования является операция обобщения. Обобщение продукций – задача довольно сложная, и далеко еще не решенная. В более простой задаче – задаче обобщения образцов – автору удалось продвинуться дальше. Эта задача состоит в том, чтобы по паре образцов построить такой третий образец, что каждый из двух исходных был бы его частным случаем, причем по возможности выбрать наименее общий из таких образцов. Выше уже говорилось, что эта задача соответствует задаче нахождения точной верхней грани в соответствующим образом построенной решетке образцов. Сформулируем все это более строго.

Определение. Пусть задана малая категория $\mathbf{C}$, и пусть S – объект этой категории. Обозначим символом $\mathbf{C}/S$ множество морфизмов вида $X \to S$, где X – произвольный объект категории $\mathbf{C}$. Если $\phi\colon X \to S$ и $\psi\colon Y \to S$ – два образца, т.е. $\phi, \psi \in \mathbf{C}/S$, то положим $\phi \leq \psi$, если существует морфизм $\chi\colon X \to Y$ такой, что $\phi = \psi\chi$.

Это определение фактически повторяет определение сопоставимости образцов, данное в разделе 3.4.

Теорема. *Отношение $\leq$ является предпорядком на множестве $\mathbf{C}/S$.*

Доказательство. Рефлексивность следует из того, что для единичного морфизма $1_X\colon X \to X$ имеем $\phi = 1_X\phi$. Пусть теперь для трех образцов $\phi\colon X \to S$, $\psi\colon Y \to S$ и $\chi\colon Z \to S$ имеем $\phi \leq \psi \leq \chi$. Это означает, что для некоторых $\sigma\colon X \to Y$ и $\tau\colon Y \to Z$ выполнены равенства $\phi = \psi\sigma$ и $\psi = \chi\tau$. Но тогда $\phi = \psi\tau\sigma$, т.е. $\phi \leq \chi$, чем доказана транзитивность. □

Известно, что если на некотором множестве P задано отношение предпорядка $\leq$, то отношение $\sim$, определяемое условием $x \sim y \iff z \leq y \& y \leq x$, является отношением эквивалентности, и $\leq$ индуцирует частичный порядок на фактормножестве множества P по этому отношению, которое мы далее будем обозначать символом $\overline{P}$. Переход от множества $\mathbf{C}/S$ к множеству $\overline{\mathbf{C}/S}$ часто бывает оправдан, ибо если ϕ, ψ – S-образцы, то эквивалентность $\phi \sim \psi$, т.е. $\phi \leq \psi$, $\psi \leq \phi$ означает, в свете теоремы, доказанной в начале раздела 3.4, что $S_\phi = S_\psi$, т.е. эквивалентные объекты, различаясь в интенсиональном смысле, совпадают в экстенсиональном, ибо описывают одно и то же множество образцов.

Теперь дадим основное определение, связанное с теорией обобщений образцов.

Определение. Образец $\sigma\colon Z \to S$ называется наименьшим обобщением образцов $\phi\colon X \to S$ и $\psi\colon Y \to S$, если

1. существуют морфизмы λ и μ такие, что диаграмма

$$\begin{tikzcd} S \arrow[r, equal] & S \arrow[r, equal] & S \\ X \arrow[u, "\phi"] \arrow[r, "\lambda"'] & Z \arrow[u, "\sigma"] & Y \arrow[u, "\psi"] \arrow[l, "\mu"] \end{tikzcd}$$

коммутативна;

2. если Z' – объект, а σ', λ' и μ' – морфизмы, для которых диаграмма

$$\begin{tikzcd} S \arrow[r, equal] & S \arrow[r, equal] & S \\ X \arrow[u, "\phi"] \arrow[r, "\lambda'"'] & Z' \arrow[u, "\sigma'"] & Y \arrow[u, "\psi"] \arrow[l, "\mu'"] \end{tikzcd}$$

коммутативна, то существует морфизм , делающий коммутативной диаграмму

$$\begin{tikzcd} S \arrow[r, equal] & S \\ Z \arrow[u, "\sigma"] \arrow[r, "\tau"'] & Z' \arrow[u, "\sigma'"'] \end{tikzcd}$$

Как несложно видеть, наименьшее обобщение является переводом на язык морфизмов определения точной верхней грани в множестве $\mathbf{C}/S$.

Теоретико-категорный язык не только позволяет дать определение наименьшего обобщения в довольно общем виде. Он позволяет также свести процесс построения наименьшего обобщения к более простым операция теории категорий. Одним из примеров того, как это можно сделать, служит следующая теорема.

Теорема. *Пусть заданы образцы $\phi\colon X \to S$, $\psi\colon Y \to S$, и пусть объект Z вместе с морфизмами $\sigma\colon X \to Z$ и $\tau\colon Y \to Z$ – прямая сумма объектов X и Y. Рассмотрим морфизм $\chi\colon Z \to S$, делающий коммутативной диаграмму*

$$\begin{tikzcd} S \arrow[r, equal] & S \arrow[r, equal] & S \\ X \arrow[u, "\phi"'] \arrow[r, "\lambda"'] & Z \arrow[u, "\chi"'] & Y \arrow[u, "\psi"'] \arrow[l, "\mu"] \end{tikzcd}$$

(существование и единственность такого морфизма следует из определения прямой суммы). Тогда образец $\chi\colon Z \to S$ есть наименьшее обобщение $\phi\colon X \to S$ и $\psi\colon Y \to S$.

Доказательство. Первая диаграмма определения наименьшего обобщения есть повторение диаграммы из условия теоремы. Пусть теперь имеется коммутативная диаграмма

$$\begin{tikzcd} S \arrow[r, equal] & S \arrow[r, equal] & S \\ X \arrow[u, "\phi"'] \arrow[r, "\lambda'"'] & Z' \arrow[u, "\chi'"'] & Y \arrow[u, "\psi"'] \arrow[l, "\mu'"] \end{tikzcd}$$

Морфизмы $\lambda'\colon X \to Z'$ и $\mu'\colon Y \to Z'$ определяют морфизм $\xi\colon Z \to Z'$, для которого $\lambda' = \xi\lambda$ и $\mu' = \xi\mu$. Тогда

$$\chi'\xi\lambda = \chi'\lambda' = \phi = \chi\lambda$$

и

$$\chi'\xi\mu = \chi'\mu' = \psi = \chi\mu,$$

откуда $\chi'\xi = \chi$, что заканчивает доказательство. □

Условие теоремы не является необходимым. Наименьшее обобщение образцов $\phi\colon X \to S$ и $\psi\colon \to S$ может существовать и в том случае, когда не существует прямой суммы объектов X и Y. Поскольку в определении наименьшего обобщения нет никаких требований единственности, нет возможности утверждать, что наименьшее обобщение определяется однозначно с точностью до изоморфизма. Кроме того, наименьшее обобщение определено не для пары объектов, а для пары образцов, т.е. в общем случае может зависеть не только от объектов X и Y, но и от морфизмов ϕ и ψ, в то время, как прямая сумма строится по паре объектов.

Как уже говорилось, описанное понятие является основным инструментом в алгоритмах формирования знаний посредством обобщения. В работе [10] показано, как на теоретико-категорном языке можно описать некоторые самые простые алгоритмы, используемые в работе ДСМ-метода [11]. Эти описания могут быть построены на верхнем уровне. Применение их к конкретной задаче сводится к построению категории, соответствующей структуре знаний, используемых для решения этой задачи. Построив такую категорию разработчик должен убедиться, что в ней для любого объекта S и для любой пары S-образцов существует наименьшее обобщение – в противном случае алгоритмы не применимы. После этого он должен определить процедуру, строящую такое обобщение. В частности, если построенная категория является категорией с конечными суммами, он может определить процедуру, строящую прямую сумму, и свести к ней процедуру построения наименьшего обобщения на основе доказанной выше теоремы. После этого можно будет использовать алгоритмы верхнего уровня.

3.6. Нижний уровень – изменение характера знаний

До сих пор говорилось о том, какие задачи решает разработчик алгоритмов верхнего уровня. Теперь обсудим вопрос о том, что можно делать на нижнем уровне. Возможность изменять нижний уровень независимо (или – почти независимо) от верхнего, позволяет легко перенастраивать одни и те же алгоритмы и даже программные модули для решения задач работающие с данными и знаниями иного формата. Автор этих строк многие годы занимался разработкой продукционных экспертных систем, выводящих новые факты из фактов, установленных ранее. Когда для решения задачи выявления конструкций со значением обусловленности [12] потребовалась система преобразования текстов, в качестве такой системы была взята та же продукционная система, но нижний уровень ее был переписан так, что теперь она работала не с множествами фактов, а с текстами.

Но в этой статье хотелось бы коснуться несколько иного способа модификации знаний, который касается не изменения способа кодирования этих знаний, а изменению того, что автор в своих работах называл характером знаний. Знания разного характера могут записываться одинаково и одинаково выводиться с помощью продукций, но на использование этих знаний при этом накладываются некоторые ограничения. Наиболее известными системами с некоторыми особенностями в характере знаний являются нечеткие и динамические системы. В нечетких системах факты можно использовать лишь с большей или меньшей степенью уверенности. Существуют различные представления нечетких знаний, от нечетких множеств Заде, до коэффициентов уверенности, впервые предложенных в системе MICIN. В динамических системах истинность раз установленного факта не является вечной, он может утратить истинность по истечении определенного времени или при изменении каких-либо условий. В этом случае истинность как самого факта, так и выведенных из него следствий нуждается в переоценке. При использовании категорной технологии характер знаний иногда может быть изменен перепрограммированием нижнего уровня, т.е. заменой категории.

Начнем с динамических систем. Динамические системы в изложенном выше понимании были исследованы В.Л.Стефанюком. Им же были предложены некоторые алгоритмы вывода в таких системах, например, замена динамической системы квазистатической [13, 14]. Автором эти алгоритмы были переформулированы на теоретико-категорном языке.

Как уже говорилось, в динамической системе истинность раз установленного факта не является абсолютной, она зависит от контекста и при определенных условиях может быть пересмотрена. Для формализации понятия “контекст” введем в рассмотрение решетку P, которую назовем контекстной решеткой, а ее элементы – контекстами. С каждым фактом f мы связываем некоторый контекст $p \in P$, в котором этот факт верен. Если текущее состояние системы описывается контекстом $q \in P$, и если $p \leq q$, будем считать, что факт f является истинным, если же неравенство $p \leq q$ не выполнено, истинность факта f следует перепроверить заново.

Для описания динамических продукционных систем была предложена P-кратная категория, построенная по некоторой категории и решетке. Пусть задана решетка P, обладающая наименьшим элементом 0. Построим категорию $\tilde{P}$, имеющую единственный объект, множеством эндоморфизмов которого является сама решетка P, роль композиции играет операция взятия нижней грани $\vee$ в этой решетке, роль тождественного морфизма – наименьший элемент 0 решетки.

Определение. Пусть $\mathbf{C}$ – некоторая категория. P-кратной категорией над $\mathbf{C}$ называется категория $\mathbf{C}_P = \mathbf{C} \times \tilde{P}$.

Таким образом, морфизм P-кратной категории имеет вид (ϕ, p), где ϕ – морфизм категории $\mathbf{C}$, а $p \in P$, т.е. с каждым фактом, который на категорном языке представлен морфизмом, связана информация о контексте, в котором он определен. Композиция морфизмов в P-кратной категории определяется формулой $(\phi, p) \circ (\psi, q) = (\phi \circ \psi, p \vee q)$, т.е. композиция двух морфизмов определена в контексте, в котором определен каждый из них.

Основной вопрос теоретического исследования P-кратной категории связан с тем, в каких случаях те или иные свойства категории $\mathbf{C}$ переносятся на категорию $\mathbf{C}_P$. К важному классу таких свойств относятся свойства порядка на множестве образцов. В работе [15] доказана следующая теорема.

Теорема. *Пусть $\mathbf{C}$ – категория, S – объект этой категории, P – решетка. Если частично упорядоченное множество $\overline{\mathbf{C}/S}$ образует решетку, частично упорядоченное множество $\overline{\mathbf{C}_P/(S,*)}$ также образует решетку. Символом * здесь обозначен единственный объект категории $\tilde{P}$.*

Другим важным свойством категории является существование в ней пределов и слабых пределов. На основе слабых пределов строятся многие конструкции, полезные при изучении систем образцов, поэтому исследование вопроса, когда из существования слабых пределов в категории $\mathbf{C}$ следует их существование в категории $\mathbf{C}_P$, представляет интерес. В работе [15] показано, что такое наследование свойств имеет место при ряде ограничений на контекстную решетку. В частности, были получены следующие результаты.

Пусть P – решетка, $Q \subset P$. Малой суммой[3] множества Q будем называть наименьший элемент множества $\{m \in P \colon \forall x, y \in Q \quad m \vee x = m \vee y\}$.

Теорема. *Если в решетке существуют конечные малые суммы, такая решетка сохраняет слабые пределы функтора из конечных категорий со слабым начальным объектом.*

Теорема. *Если решетка является булевой алгеброй, то в ней существуют конечные малые суммы.*

Теорема. *Во вполне дистрибутивной решетке существуют конечные малые суммы.*

Доказательства этих утверждений можно найти в [15].

Общая теория часто позволяет увидеть сходство в том, что, на первый взгляд кажется различным. Автор нигде не встречал замечаний о сходстве между нечеткими и динамическими системами, однако такое сходство становится заметным при использовании теоретико-категорного подхода. Это сходство состоит в том, что в обеих системах с каждым фактом связан некоторый дополнительный параметр, описывающий характер использования этого факта. В динамических системах таким параметром является элемент контекстной решетки (т.е. параметр не обязательно является числовым). В нечетких системах в качестве такого параметра можно взять коэффициент уверенности. В работе [16] рассмотрены другие задачи, в которых с каждым фактом связан некоторый параметр, вводящие ограничения на использование этого факта. В.Л.Стефанюк предложил называть этот параметр ретенцией. Множество всех возможных значений ретенции обозначим символом M. Таким образом, если F – множество фактов, допустимых в работе системы, мы можем считать, что задано отображение

[3] Термин В.Б.Шехтмана

$R\colon F \to M$, ставящее в соответствие каждому факту $f \in F$ его ретенцию $R(f) \in M$. Достаточно естественно предположить, что ретенция факта $f\&g$ зависит только от ретенций фактов f и g, т.е. на множестве определена бинарная операция, которую будем обозначать мультипликативно. В контекстной решетке роль такой операции играет операция $\vee$ взятия верхней грани. В нечетких системах есть обширная литература о комбинировании свидетельств, см. напр. [17]. Очевидные соотношения $f\&g = g\&f$ и $f\&(g\&h) = (f\&g)\&h$ означают, что эта операция является коммутативной и ассоциативной, т.е. M является коммутативной полугруппой. В каждой системе бывают безусловно истинные факты, истинность которых не зависит ни от чего. Если обозначить такой факт символом e, то ретенция факта $f\&e$ очевидно равна ретенции f, т.е. ретенция e играет роль единицы. Мы видим, что M является коммутативным моноидом. Аналогично тому, как по категории $\mathbf{C}$ и решетке P была построена категория $\mathbf{C}_P = \mathbf{C} \times \tilde{P}$, по категории $\mathbf{C}$ и моноиду M может быть построена M-кратная категория $\mathbf{C}_M = \mathbf{C} \times \tilde{M}$, где символом $\tilde{M}$ обозначена категория, состоящая из одного объекта, моноидом эндоморфизмов которого является M.

Техника перехода от некоторой заданной категории к кратной категории может быть полезной на практике, позволяя с помощью незначительной доработки превращать продукционные системы в нечеткие, динамические или иные системы, отличающиеся характером знаний. Переход этот осуществляется следующим образом. Допустим, мы имеем некоторую продукционную систему, построенную по категорной технологии. То, что она построена по категорной технологии, означает, что для этой системы определена категория, описывающая соответствующую структуру знаний. Для перехода к системе с более сложным характером знаний следует прежде всего описать соответствующий этому характеру моноид ретенции. После этого следует описанным выше способом определить кратную категорию. Затем нужно провести работы нижнего уровня, состоящие в перепрограммировании для построенной категории теоретико-категорных операций, используемых в системе. После этого систему можно использовать как нечеткую, динамическую и т.д.

4. ЗАКЛЮЧЕНИЕ

Повторим вкратце способ применения категорной технологии для разработки продукционных систем. Выше уже говорилось, что эта технология может быть использована как для описания алгоритмов функционирования такой системы, так и для программной реализации этих алгоритмов.

Разработчик, работающий на верхнем уровне, должен построить алгоритмы работы со знаниями таким образом, чтобы в них использовались только теоретико-категорные операции. При этом он должен составить список этих операций, который должен стать обязательной частью описания алгоритма. После этого он должен написать программы, использующие неопределенные у этому моменту процедуры нижнего уровня так, чтобы при добавлении модуля, содержащего определение этих процедур, программа стала бы работоспособной. Это может быть набор программных модулей, которые требуют для компиляции и линковки добавления модуля с соответствующим определениями процедур. В технологии объектного программирования удобно построить программу на использовании абстрактных классов, от которых на нижнем уровне должны быть произведены конкретные классы. Возможны и другие способы.

Разработчик, работающий на нижнем уровне, должен описать категорию, соответствующую структуре знаний, с которой будет работать создаваемая им программа. Затем он должен выбрать пригодный алгоритм верхнего уровня для решения своей задачи. Для этого он, кроме всего прочего, должен удостовериться в том, что в построенной им категории выполнимы те операции, который указал разработчик верхнего уровня в описании алгоритма. Если эти операции достаточно сложны, он может свести их в рамках построенной категории к более простым теоретико-категорным операциям, подобно тому, как в разделе 3.5 наименьшее обобщение было сведено к прямой сумме. Затем он должен в своей программе определить структуры данных, соответствующие категорным понятиям, прежде всего – морфизмам, и запрограммировать нужные операции над этими структурами.

В заключение дадим ответ на естественный вопрос: в каких случаях категорная технология работает, а в каких ее применение невозможно или затруднительно. Ответ состоит в том, что категорная технология применима в том случае, когда продукции, с которыми должна работать создаваемая система, действительно относятся к виду сопоставление-конкретизация. Если же условие применимости продукции проверяется с помощью вычисления некоторого предиката или действие, производимое продукцией, представляет собой сложную процедуру, построить категорию, описывающую такие продукции, скорее всего не удастся, или она будет настолько сложна, что исследование ее свойств окажется невозможным, что сделает категорную технологию неприменимой.

СПИСОК ЛИТЕРАТУРЫ

1. V.L. Stefanuk, A.V. Zhozhikashvili. Productions and rules in artificial intelligence, *KYBERNETES, The International Journal of Systems & Cybernetics. MCB University Press*, 2002, Vol. 31, no. 5,6, pp. 817-826.

2. А.В.Жожикашвили, В.Л.Стефанюк. О понятии продукции в искусственном интеллекте, *Известия академии наук, серия "Теория и системы управления"*, 2002, № 4, стр.76–81.

3. Жожикашвили А.В., Монады для формализации процедуры сопоставления с образцом *Программирование*, 1014, № 3, стр. 15–29.

4. Zhozhikashvili A.V., Stefanuk V.L. Theory of Category Approach to Knowledge Based Programming *11th Joint Conference, JCKBSE 2014*, Volgograd, Russia, September 17–20, 2014, pp. 735–746.

5. Минский М., Фреймы для представления знаний. В кн. *Психология машинного зрения.* М.: Мир, 1978.

6. Стефанюк В.Л., Жожикашвили А.В., Трегуб А.Н. Прикладная экспертная система для выбора свай и расчета их несущей способности "СВАЯ 1.0". *II Всесоюзная конференция "Искусственный интеллект – 90" г. Минск. Тезисы докладов. Серия Выставка.* Тверь, Центрпрограммсистем, 1990, т. 3, стр. 136–145.

7. Стефанюк В.Л., Жожикашвили А.В. *Создание и сопровождение базы знаний в динамической экспертной системе. Проблема "Сейсмичность".* Препринт. Ин-т физики Земли АН СССР, Москва, 1992.

8. Нильсон Н. *Принципы искусственного интеллекта.* – М.: Радио и связь, 1985.

9. Жожикашвили А.В., Стефанюк В.Л. Продукционные сети: развитие теории ТК-продукций. *Вестник Российского университета дружбы народов. Серия "Прикладная и компьютерная математика"*, 2003, Т. 2, с. 118–126.

10. Жожикашвили А.В. Метод сходства и метод различия Д.С. Милля на языке теории категорий. *Искусственный интеллект и принятие решений*, 2014, № 4, С. 25–30.

11. Финн В.К. Правдоподобные выводы и правдоподобные рассуждения. В кн.: *ИТОГИ НАУКИ и ТЕХНИКИ сер. теория вероятностей, математическая статистика.* Москва: ВИНИТИ, 1988, стр. 1–83.

12. Жожикашвили А.В., Савинич Л.В.; Стефанюк В.Л. Процедуры выявления знаний об отношениях обусловленности. *Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010*, Тверь, 2010, стр. 92–99.

13. V.L. Stefanuk. Dynamic expert systems. *KYBERNETES, The International Journal of Systems & Cybernetics.* 2000, V. 29, no. 5/6, pp. 702–709.

14. Стефанюк В.Л. Поведение квазистатической оболочки в изменяющей нечеткой среде. *Труды IY национальной конференции с международным участием "Искусственный интеллект – 94"*, Рыбинск, 1994, Т. 1, стр.199–203.

15. Жожикашвили А.В., Стефанюк В.Л. Свойства кратной категории, построенной по категории и решетке. *Проблемы передачи информации*, 2011, № 4, т. 47, стр.1–13.

16. Жожикашвили А.В., Стефанюк В.Л. Расслоенные категории для описания динамических продукционных систем. *Известия Российской академии наук. Теория и системы управления*, 2008, № 5, стр. 71–76.

17. Стефанюк В.Л. Некоторые аспекты теории экспертных систем. *Известия АН СССР. Техническая кибернетика*, 1987, № 2.

Category-Theoretic Technology of Creation and Development of Knowledge-Based Intelligent Systems

A. Zhozhikashvili

Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia

The article details productions of a special form which could be used in knowledge-based intelligent systems. The mathematical concept of these productions is based on the category theory. A method of creation and development of production systems, that is based on this theory, is also described.

KEYWORDS: knowledge representation, production, category theory.