В.А. Фомичев

ФОРМАЛИЗАЦИЯ ПРОЕКТИРОВАНИЯ ЛИНГВИСТИЧЕСКИХ ПРОЦЕССОРОВ

МАКС ПРЕСС

МОСКВА 2005

В монографии описывается апробированная на практике новая система взаимосвязанных формальных моделей и алгоритмов, предназначенных для проектирования лингвистических процессоров (компьютерных систем, осуществляющих смысловую обработку письменных текстов или устной речи на естественном языке) в произвольных предметных областях. Значительное внимание уделяется изложению оригинального теоретического подхода к математическому описанию смысловой структуры не только предложений, но и сложных связных текстов (или дискурсов), относящихся к деловой прозе: текстов по медицине, экономике, юриспруденции и т.д. Анализируются возможности использования этого подхода в теории многоагентных систем, для разработки логико-информационных основ электронной коммерции и для устранения языкового барьера между пользователями сети Интернет из разных стран.

Значительная часть материалов монографии была опубликована в научных журналах “Информационные технологии”, “Качество и ИПИ (CALS)- технологии”, “Качество. Инновации. Образование”, “Informatica” (Словения), “Cybernetica” (Бельгия) и трудах международных научных конференций и симпозиумов, проходивших в России, Австрии, Великобритании, Германии, Дании, Нидерландах, Словении, Франции.

Книга не имеет аналогов в мировой научной литературе и будет полезна как опытным специалистам в области прикладных интеллектуальных систем или математической лингвистике, так и студентам и начинающим ученым.

The monograph describes a new system of interrelated formal models and algorithms tested in practice and destined for designing linguistic processors, or natural language processing systems (computer systems fulfilling the conceptual processing of written texts or oral speech in natural language) in arbitrary application domains. A considerable attention is drawn to setting forth an original theoretical approach to representing in a mathematical way the structured meanings (or conceptual structure, semantic structure) of not only separate sentences but also of complicated narrative texts (or discourses) pertaining to medicine, economy, law, and other fields of professional activity. The possibilities of using this approach in the multi-agent theory, for the elaboration of logical-informational foundations of electronic commerce (e-commerce), and for the elimination of the language barrier between the Internet users from various countries are analysed.

A considerable part of the stated materials was published in the scientific journals in Russian “Informational Technologies”, “Quality and IPI (CALS)-Technologies”, “Qulaity. Innovations. Education”, in the international scientific journals “Informatica” (Slovenia), “Cybernetica” (Belgium), and in the proceedings of the international scientific conferences and symposia which were held in Russia, Austria, Denmark, France, Germany, Slovenia, The Netherlands, and United Kingdom.

The monograph has no analogues in the world scientific literature and will be of use both to experienced specialists in the field of applied intelligent systems or mathematical linguistics and to the students and young scientists.

Моей семье:

Ольге Святославовне

Фомичевой,

Людмиле Дмитриевне

Удаловой,

Дмитрию Владимировичу

Фомичеву

посвящается

ОГЛАВЛЕНИЕ

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Всегда практика должна быть воздвигнута на
хорошей теории, ворота которой - перспектива
Леонардо да Винчи

В преподавании такой быстро развивающейся области,
какой является наука о вычислительных процессах,
правильный педагогический принцип состоит в том,
чтобы больше внимания уделять идеям, а не техни-
ческим подробностям реализации
А. Ахо, Дж. Ульман

За последние два десятилетия научно-техническое направление "искусственный интеллект" получило значительное развитие и нашло целый ряд успешных применений. Основная часть информации хранится и передается людьми с помощью естественного языка (ЕЯ), т.е. совокупности русского, английского, японского и других языков. Один из главных подклассов компьютерных систем с элементами искусственного интеллекта (СИИ) составляют программы, понимающие ЕЯ или синтезирующие выражения ЕЯ по некоторым внутренним представлениям. Такие программы называются системами обработки естественного языка (в англоязычной научной литературе: natural language processing systems), или лингвистическими процессорами (ЛП). Технологии, предусматривающие использование ЛП для обработки информации, составляют основной подкласс лингвистических информационных технологий (ЛИТ).

Другие виды современных ЛИТ связаны с разработкой и применением языков общения компьютерных интеллектуальных агентов (КИА) в многоагентных системах, языков построения протоколов переговоров, проводимых КИА в области электронной коммерции, и языков формирования контрактов, заключаемых КИА в ходе таких переговоров, а также семантически-структурированных языков нового поколения для представления информации во Всемирной Паутине (the World Wide Web, или WWW).

Несколько неформальных понятий, являющихся базовыми для теории смысловой обработки компьютером естественного языка, многократно используются в этой книге: семантика естественного языка, связный текст (или дискурс), структурированное значение выражения на ЕЯ, семантическое представление ЕЯ-выражения и алгоритм семантико-синтаксического анализа.

Под семантикой ЕЯ будем понимать совокупность закономерностей передачи информации средствами ЕЯ. Связным текстом (или дискурсом) называется последовательность взаимосвязанных по смыслу выражений на ЕЯ.

Если Т – некоторое выражение на ЕЯ (словосочетание, предложение, дискурс), то структурированным значением выражения Т является информационая структура, строящаяся мозгом человека, владеющего данным подязыком ЕЯ (русским, английским или другим), независимо от контекста, в котором услышано или прочитано выражение Т, т.е. строящаяся на основе только знаний о значениях элементарных лексических единиц и правил их комбинирования в данном языке.

Под семантическим представлением (СП) ЕЯ-выражения Т понимается формальная структура, являющаяся либо образом структурированного значения этого выражения, либо отражением смысла (или содержания) данного выражения в определенном контексте - в конкретной ситуации диалога, в контексте знаний о мире или в контексте предшествующей части дискурса.

Таким образом, СП ЕЯ-выражения Т является формальной структурой, первичными элементами которой являются, в частностия, обозначения понятий, конкретных объектов, множеств объектов, событий, имена функций и отношений, логические связки, обозначения чисел и цветов, а также обозначения смысловых отношений между значениями фрагментов текста или между объектами рассматриваемой предметной области.

СП текстов могут являться, например, строками и размеченными ориентированными графами (семантическими сетями).

Алгоритм семантико-синтаксического анализа строит по тексту на ЕЯ его СП, используя для этого знания о морфологии и синтаксисе подъязыка ЕЯ (русского, английского и др.), информацию о взаимосвязях лексических единиц с единицами семантического уровня и знания о мире. Семантическое

представление текста, построенное таким алгоритмом, интерпретируется прикладной интеллектуальной системой в зависимости от ее назначения, например, как задание на поиск ответа на вопрос, команда на выполнение физического действия автономным интеллектуальным роботом, фрагмент знаний о мире, предназначенный для пополнения базы знаний и т.д.

Научные результаты, изложенные в данной монографии, были получены автором в ходе цикла исследований, начатого более двадцати лет назад. Выбор направления исследований был реакцией на почти полное отсутствие в то время эффективных математических средств и методов проектирования ЛП.

Результаты данной монографии дают не только продвижение вперед, но и *качественный скачок* в области разработки формальных средств и методов проектирования алгоритмов семантико-синтаксического анализа ЕЯ-текстов. Этот качественный скачок обусловлен следующими основными факторами:

1. Разработчики ЛП получили систему правил (причем компактную, состоящую всего из 10 основных правил), позволяющих, по гипотезе автора, строить семантические представления произвольных текстов деловой прозы, т.е. текстов по экономике, технике, медицине, юриспруденции и т.д. Это означает, что эффективные процедуры построения СП ЕЯ-текстов и процедуры обработки СП ЕЯ-текстов (в контексте содержания предшествующей части текста или диалога, в рамках знаний о предметной области и т.д.) можно будет использовать в разных предметных областях и развивать возможности этих процедур при возникновении новых задач.
2. Построена формальная модель лингвистической базы данных, содержащей такие сведения о лексических единицах и их взаимосвязях с информационными единицами, которые достаточны для семантико-синтаксического анализа интересных для приложений подъязыков русского языка.
3. Разработан практически полезный сложный структурированный алгоритм семантико-синтаксического анализа, который описывается не средствами какой-либо системы программирования, а полностью с помощью

предложенной системы формальных понятий, что делает этот алгоритм независимым от программной реализации и предметной области.

СОДЕРЖАНИЕ КНИГИ

В главе 1 дается краткий обзор областей применения лингвистических процессоров, а также анализируются потребности расширения запаса эффективных формальных средств и методов для проектирования ЛП и разработки ЛИТ в области многоагентных систем и электронной коммерции.

В главе 2 описывается математическая модель, перечисляющая первичные единицы концептуального уровня, используемые ЛП, а также описывающая информацию, связанную с такими единицами и необходимую для соединения этих единиц в составные единицы, отображающие структурированные значения (СЗ) сколь угодно сложных ЕЯ-текстов.

В главе 3 (в развитие результатов главы 2) построена математическая модель для описания СЗ предложений и сложных связных текстов (дискурсов) на естественном языке (в частности, на русском, английском, немецком, французском языках). Модель представляет собою определение нового класса формальных языков, названных стандартными концептуальными языками (стандартными К-языками, СК-языками), и может рассматриваться как формальная грамматика нового вида. Сущность этой модели в том, что она задает 10 операций на концептуальных структурах, с помощью которых за конечное число шагов можно построить семантическое представление предложения или дискурса из чрезвычайно широкого подъязыка деловой прозы.

Проведено математическое исследование формальных объектов, задаваемых этой моделью – выражений СК-языков. В частности, доказана однозначность структурного анализа таких выражений.

Глава 4 посвящена исследованию выразительных возможностей класса СК-языков. Показано, что выражения СК-языков удобно использовать для: (а) построения СП предложений (выражающих высказывания, вопросы, команды) и сложных дискурсов на русском языке, (б) построения составных целей, (в)

представления знаний о мире, в том числе для построения формальных определений понятий и объектно-ориентированных модулей знаний..

Проведено сравнение выразительных возможностей СК-языков с выразительными возможностями других, наиболее часто используемых подходов к формальному представлению значений (смысловой структуры) ЕЯ-текстов: теории представления дискурсов, теории концептуальных графов, эпизодической логики, теории расширенных семантических сетей, теории неоднородных семантических сетей и компьютерной семантики русского языка. Показано, что выразительные возможности СК-языков значительно превосходят возможности перечисленных подходов и, в то же время, аппарат СК-языков позволяет моделировать механизмы представления информации, характерные для каждого из указанных подходов.

В главе 5 исследуются возможности использования аппарата СК-языков для решения ряда актуальных проблем информатики: разработки языков представления содержания посланий компьютерных интеллектуальных агентов, в частности, языков, предназначенных для формирования контрактов и протоколов переговоров в области электронной коммерции, создания семантического сетевого языка нового поколения, построения онтологий предметных областей, разработки новых языков представления знаний для решения информационно-сложных задач, проектирования интеллектуальных информационно-поисковых и вопросо-ответных Интернет-систем нового поколения.

В главе 6 вводится формальное понятие лингвистического базиса, которое интерпретируется как описание структуры лингвистической базы данных (ЛБД), используемой алгоритмом семантико-синтаксического анализа ЕЯ-текстов. ЛБД, структура которых отображается построенной моделью, позволяют устанавливать возможные смысловые отношения, в частности, в сочетаниях «Глагол + Предлог + Существительное», «Глагол + Существительное», «Существительное1 + Предлог + Существительное2», «Число + Существительное», «Прилагательное + Существительное», «Существительное1 + Существительное2», «Причастие + Существительное», «Причастие + Предлог

+ Существительное», «Вопросительно-относительное местоимение или Наречие + Глагол», «Предлог + Вопросительно- относительное местоимение + Глагол».

В главе 7 излагается новый метод преобразования ЕЯ-текстов в их семантические представления. Метод предусматривает использование предложенного автором матричного семантико-синтаксического представления (МССП) входного текста как промежуточного представления при переходе от ЕЯ-текста к СП текста, являющемуся выражением некоторого СК-языка (т.е. К-представлением текста). При этом не используется традиционное синтаксическое представление текста. Тексты могут быть, в частности, вопросами, сообщениями (описаниями фактов, ситуаций) или командами.

В главах 8 и 9 разработан сложный структурированный алгоритм семантико-синтак-сического анализа текстов из представляющих практический интерес подъязыков естественного (русского) языка (алгоритм SemSyn). Этот алгоритм, базирующийся на построенной в главе 6 формальной модели ЛБД и на введенном в главе 7 понятии МССП текста, устанавливает смысловые отношения между элементарными значащими единицами входного текста, отражая эти отношения посредством МССП, а затем строит СП текста, являющееся выражением некоторого СК-языка (К-представлением). Входные ЕЯ-тексты могут выражать высказывания (сообщения), команды, специальные вопросы (т.е. вопросы с вопросительными словами), общие вопросы (т.е. вопросы с ответом «Да»/ «Нет»)и могут, в частности, включать причастные обороты и придаточные определительные предложения. Алгоритм SemSyn позволяет устанавливать возможные смысловые отношения, в частности, в сочетаниях перечисленных выше видов.

В заключении к данной монографии делается вывод о том, что совокупность научных результатов, изложенных в главах 1 - 4, 6 - 9, и часть научных результатов главы 5 образуют новую теорию проектирования семантико-синтаксических анализаторов естественно-языковых текстов с использованием формальных средств представления входных, промежуточных и выходных данных; эта теория может быть названа теорией К-представлений.

Приложение содержит доказательства двух лемм и базирующегося на них доказательства одного из утверждений из главы 3. Нумерация утверждений сквозная внутри каждой главы (Утверждение 3.1, Утверждение 3.2 и т.д.).

В основе большей части содержания данной монографии лежат циклы лекций, читавшиеся автором с 1996 г. студентам Российского государственного технологического университета им. К.Э Циолковского – “МАТИ” по дисциплинам “Теоретические основы лингвистических информационных технологий”, “Математическая лингвистика”, “Проектирование лингвистических процессоров” и студентам Московского государственного института электроники и математики (технического университета) по дисциплинам “Лингвистические информационные технологии”, “Проектирование лингвистических процессоров” и “Глобальные информационные сети и дистанционное обучение”.

БЛАГОДАРНОСТИ

Я благодарен профессору, д.т.н., зав. кафедрой “Программное обеспечение вычислительных машин” Российского государственного социального университета Ю.П. Кораблину, профессору МАТИ Г.С. Плесневичу, профессорам МИЭМ Л.С. Воскову и А.К. Зыкову за обсуждение многих разделов данной монографии и полезные замечания, а также профессору, д.т.н., заслуженному деятелю науки и техники РСФСР , зав. кафедрой “Системы автоматического управления” МГТУ им. Н.Э. Баумана К.А. Пупкову за поддержку первых, самых трудных шагов исследования, результаты которого представлены в данной монографии.

С 1990-х годов положение науки в нашей стране, к сожалению, остается таким, что появление этой книги было бы невозможно без огромной поддержки, внимания, терпения моей жены - Ольги Святославовны Фомичевой, и мамы Ольги Святославовны - Людмилы Дмитриевны Удаловой.

Благодаря помощи моего сына Димы, выпускника факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, в освоении нескольких компьютерных технологий были подготовлены к печати многие работы, послужившие основой для этой монографии.

Я признателен директору издательства МАКС Пресс, Алле Николаевне Матвеевой, за предложение подготовить и издать эту книгу.

Большая помощь в подготовке в электронном виде материалов, послуживших основой для книги, была оказана многими студентами кафедры “Информационные технологии” МАТИ, особенно Я.В. Ахромовым, и студентами кафедры “Математическое и программное обеспечение систем обработки информации и управления” Московского государственного института электроники и математики (технического университета).

Глава 1

ФОРМАЛИЗАЦИЯ СЕМАНТИКИ ЕСТЕСТВЕННОГО ЯЗЫКА И ПОТРЕБНОСТИ ПРОЕКТИРОВАНИЯ ЛИНГВИСТИЧЕСКИХ ПРОЦЕССОРОВ

1.1. Области применения лингвистических процесоров

Прогресс, достигнутый за последние два десятилетия в области проектирования ЛП, выразился в появлении широкого спектра областей применения ЛП. Такими областями, в частности, являются: машинный перевод

письменных текстов (исторически первая область использования ЛП) и устной речи; естественно-языковые интерфейсы (ЕЯ-интерфейсы) прикладных интеллектуальных систем: экспертных систем, расчетно-логических систем, автономных интеллектуальных роботов; синтез текстов, представляющих рекомендации пользователю экспертной системы (медицинской диагностики, технической диагностики и др.) в естественно-языковой форме; проектирование концептуальных схем баз данных посредством преобразования ЕЯ-спецификаций предметной области в концептуальную схему базы данных; автоматизированное проектирование технических объектов (например, электронных блоков) с помощью преобразования ЕЯ-спецификации проектируемого объекта в формальную спецификацию и затем – в проектную документацию технического объекта.

Развитие исследований в области конструирования ЛП привело к появлению новых теоретических и практических задач.

Государственными и коммерческими организациями накоплены большие запасы информационных ресурсов, содержащих знания о предметных областях. Для повышения эффективности работы сотрудников с накопленными знаниями крупные компании в мире разрабатывают или уже разработали и используют системы управления знаниями. По имеющимся в литературе оценкам, более 70% ресурсов, накопленных в различных организациях, носит неструктурированный характер и образуется электронными текстовыми документами. Поэтому, по мнению ряда авторов, повышению эффективности работы сотрудников различных организаций с накопленными информационными ресурсами будет способствовать разработка интеллектуальных поисковых систем с ЕЯ-интерфейсами, способных осуществлять смысловой анализ естественно-языковых полей разнообразных используемых электронных документов и, как следствие, давать ссылки на документы, интересующие пользователя, или формулировать ответы на поставленные пользователем вопросы (Попов 2001, 2002; Королев 2003; Арлазаров, Емельянов 2003, 2004; Pohl 2003).

Создание таких интеллектуальных поисковых систем с ЕЯ-интерфейсами, и особенно Интернет-систем, представляется весьма актуальным направлением развития исследований по разработке CALS (ИПИ)-технологий

Непрерывная информационная поддержка жизненного цикла сложного изделия предполагает совместное использование субъектами виртуального предприятия (одной из современных форм реализации CALS (ИПИ)-технологий) единой базы знаний о рассматриваемых предметных областях (возможно, распределенной и с некоторыми ограничениями на конфигурацию базы знаний, доступную определенному субъекту виртуального предприятия) и эффективный обмен информацией между субъектами виртуального предприятия. В этой связи ЕЯ-интерфейсы обещают упростить и, как следствие, увеличить эффективность взаимодействия непрограммирующих специалистов с базами данных и базами знаний

Другой острой проблемой теории СИИ является автоматизация формирования баз знаний (БЗ) СИИ. Основная часть знаний, накопленных человечеством, хранится в виде естественно-языковых текстов (ЕЯ-текстов). Поэтому в последние годы реализован или реализуется ряд проектов, направленных на автоматическое извлечение знаний из ЕЯ-текстов. Значительное внимание в Германии, США, Японии и некоторых других странах уделяется проблеме автоматизации извлечения знаний из биологических и медицинских документов (отчетов об исследованиях, статей в научных журналах и т.д.). Проекты по этой проблеме составляют важную часть нового направления в информатике, получившего название *биоинформатика.*

Однако построенные системы извлечения знаний из ЕЯ-текстов обладают весьма узкими способностями понимания ЕЯ-текстов, особенно связных текстов (дискурсов), т.е. последовательностей взаимосвязанных по смыслу фраз на ЕЯ. Это выражается в использовании разнообразных узкоспециализированных шаблонов для извлечения знаний. Центральной причиной этого положения является недостаточная проработанность вопросов формального описания закономерностей передачи информации средствами ЕЯ, т.е. вопросов формализации семантики ЕЯ.

Благодаря бурному прогрессу компьютерной сети Всемирная Паутина (the World Wide Web, WWW, W3) пользователи сети во всем мире получили быстрый доступ к огромному количеству ЕЯ-текстов, относящихся к различным областям деятельности человека. С середины 1990-х годов специалисты в самых разных предметных областях работают не только с публикациями и базами данных (БД) своих организаций, но и стремятся использовать информационные ресурсы Паутины. Поэтому чрезвычайно актуальна задача организации взаимодействия на ограниченном естественном языке из различных предметных областей с огромным объемом накопленных информационных ресурсов Всемирной Паутины (Попов 2002; Хорошевский 2002).

ЕЯ-интерфейсы для взаимодействия с информационными ресурсами Паутины необходимы не только специалистам для решения профессиональных задач, но и конечным пользователям, перед которыми стоят задачи получения медицинской или юридической информации, расширения культурного кругозора, получения дополнительного профессионального образования и т.д.

В феврале 2001 г. консорциум сети Всемирная Паутина, обозначаемый в большинстве документов сокращением W3C (the World Wide Web Consortium), официально объявил о широком развертывании исследований по преобразованию существующей сети в Семантическую Всемирную Паутину (Semantic Web). Один из наиболее важных аспектов реализации этого крупномасштабного проекта заключается в том, что компьютерные интеллектуальные агенты (КИА) смогут анализировать информацию, представленную на Веб-сайтах, взаимодействуя между собой. Часть КИА сможет выполнять смысловой анализ естественно-языковых компонентов электронных документов, представленных в Веб-сайтах. Это даст возможность конечным пользователям осуществлять поиск информации в Паутине не по ключевым словам, а по смыслу, с помощью КИА (Semantic Web 2001).

Важные дополнительные возможности для пользователя предоставят речевые браузеры: они позволят использовать телефоны (в том числе мобильные) для взаимодействия с Семантической Паутиной на ЕЯ (Voice 2001).

Прогресс в разработке компьютеров, ЛП и средств телекоммуникации привел в 1990-е годы к реализации в ряде стран проектов создания электронных

библиотек (ЭлБ), называемых в англоязычной литературе цифровыми библиотеками. В нашей стране важными импульсами к развертыванию научно-технической программы в этом направлении стали Российско-американский семинар, проходивший в 1998 г, и первая национальная конференция по электронным библиотекам с участием ученых из Германии и США, состоявшаяся в 1999 г. в Санкт-Петербурге. В итоговых материалах этой конференции, в частности, отмечается, что одной из центральных научных задач, связанных с созданием ЭлБ, является автоматизация семантического анализа ЕЯ-текстов с целью смыслового поиска информационных источников.

Развитие гражданского общества в нашей стране существенно зависит от степени доступности государственных информационных ресурсов. Обеспечение такой доступности является одной из центральных задач федеральной целевой программы “Электронная Россия (2002 – 2010 годы)“. Огромную роль в обеспечении доступа общественности к государственным информационным ресурсам должны сыграть ЭлБ. Для обеспечения подлинной широты доступа пользователей ЭлБ к информационным ресурсам необходимы интеллектуальные поисковые системы с ЕЯ-интерфейсами, способные отыскивать информационные источники или находить ответы на вопросы конечных пользователей на основе осуществления смыслового анализа (а) запроса пользователя, (б) естественно-языковых полей разнообразных хранящихся электронных документов и сравнения содержания запроса пользователя с содержанием анализируемых текстовых полей электронных документов.

В свете перечисленных и ряда других направлений применения ЛП, разработка теории и методов компьютерного понимания ЕЯ-текстов и извлечения знаний из ЕЯ-текстов является важным направлением развития теории интеллектуальных компьютерных систем (Арлазаров, Журавлев, Ларичев и др. 1998). Этой проблеме было уделено значительное внимание на Научной сессии Отделения информационных технологий и вычислительных систем РАН, состоявшейся в мае 2003 года.

1.2. Значение формальных методов для разработки лингвистических информационных технологий

Накопленный опыт исследований по созданию ЛП показал, что огромное влияние на проектирование анализаторов ЕЯ-текстов оказывают используемые методы формального отображения содержания (или смысла) текстов, а также методы формального представления промежуточных результатов смыслового анализа текстов. Особую актуальность в 1990-е годы приобрела проблема формального представления содержания связных текстов (или дискурсов).

Во-первых, основной объем информации в текстовых БД и сети Интернет представлен дискурсами. Во-вторых, сформулированная Э.В. Поповым современная концепция разработки систем общения с БД на ограниченном естественном языке (ОЕЯ) предполагает, что на вход системы поступают не только предложения, но и дискурсы (Попов 2002). В-третьих, можно согласиться с высказанной Э.В. Поповым гипотезой о том, что повышению эффективности общения на ОЕЯ с большими БД будет способствовать реализация таких систем общения, когда активную роль в диалоге будет играть не только конечный пользователь, но и компьютер, располагающий моделью базы знаний, причем инициатива будет на протяжении диалога неоднократно переходить от одного участника общения к другому. Последовательность выражений на ОЕЯ (с указанием авторов выражений), сформированных участниками общения, образует дискурс.

Можно выделить несколько наиболее важных аспектов проблемы формального представления содержания (или смысла) ЕЯ-текстов в компьютерных системах.

Идея использования в системах машинного перевода искусственного языка-посредника для представления смысла ЕЯ-текстов была высказана еще в 1960-м году А.К. Жолковским, Н.Н. Леонтьевой и Ю.С. Мартемьяновым. В 1960-е – 1970-е годы эта идея получила значительное развитие в работах А.К. Жолковского и И.А. Мельчука по лингвистической модели “Смысл – Текст” (Жолковский, Мельчук 1969; . Мельчук 1974). В 1970-е годы усилению внимания к идее семантического языка-посредника способствовала теория

смысловой зависимости в ЕЯ Р. Шенка, нашедшая применение в нескольких экспериментальных системах компьютерной обработки ЕЯ (Schank 1972; . Schank и др. 1975).

Использование языка-посредника для представления содержания (смысла) ЕЯ-текстов позволяет перейти от неформализованного объекта, каким является ЕЯ-текст, к формальной структуре, что открывает возможности обработки этой структуры различными процедурами – “семантическими экспертами” в рамках базы знаний, представленных записями на формальном языке (языке представления знаний). На протяжении 1980-х – 2000-х годов в проектировании ЛП наиболее часто использовались языки-посредники, предоставляемые теорией семантических сетей, теорией фреймов, теорией концептуальных графов и эпизодической логикой. В нашей стране использовался также язык-посредник, разработанный в рамках компьютерной семантики русского языка, расширенные семантические сети, неоднородные семантические сети (см. параграф 4.15), стандартные К-языки, предложенные автором данной работы, и некоторые другие подходы.

В середине 1990-х годов возникла новая проблема, усилившая внимание исследователей к проблеме разработки языка-посредника для отображения содержания ЕЯ-текстов. С целью устранения языкового барьера между пользователями сети Интернет из разных стран мира, Х.Учида и М. Жу (Япония) предложили новый язык-посредник, использующий слова английского языка для обозначения информационных единиц и несколько специальных символов. Этот язык, названный универсальным сетевым языком (UNL, the Universal Networking Language), базируется на идее отображения содержания фраз с помошью бинарных отношений. С конца 1990-х годов ООН финансируется комплексный проект, направленный на разработку системы ЛП, преобразующих фразы на различных естественных языках в выражения языка UNL, а также преобразующих выражения языка UNL в предложения на различных естественных языках. Координатором проекта является Институт передовых исследований ООН Токийского университета. В настоящее время в проекте разрабатываются ЛП для шести официальных языков ООН (английского, арабского, испанского, китайского, русского и французского), а

также для хинди, индонезийского, итальянского, японского, латышского, немецкого, монгольского, португальского, суахили и тайского языков (Uchida, Zhu, Della Senta 1999; Uchida, Zhu 2001; Zhu, Uchida 2002).

Проблема создания широко применимых методов формального описания содержания (смысла) предложений и дискурсов (другими словами, описания структурированных значений ЕЯ-текстов) тесно соприкасается с потребностями развития таких бурно развивающихся направлений информатики, как многоагентные системы (МАС) и электронная коммерция. Взаимодействие компьютерных интеллектуальных агентов (КИА) осуществляется через обмен посланиями (messages), которые могут выражать сообщения, вопросы и команды. Для формирования таких посланий разрабатываются специальные языки общения интеллектуальных агентов (Agent Communication Languages, или ACL). Для координации деятельности исследовательских центров разных стран по разработке стандартных инструментальных средств в области МАС в 1996 г. образован международный Фонд интеллектуальных физических агентов (The Foundation for Intelligent Physical Agents, или FIPA), штаб-квартира которого находится в Женеве. В 1997 - 2000 годах в рамках этого фонда был разработан стандарт языка общения КИА, который в дальнейшем будет называться FIPA ACL. Часть этого языка, предназначенная для представления содержания посланий (в отличие от внешней информации - об отправителе, получателе и т.д.), названа семантическим языком (FIPA Semantic Language, или FIPA SL). Фондом поставлена задача разработки библиотеки языков представления содержания посланий КИА (Content Languages), совместимых с этим языком и охватывающих весь спектр применений МАС.

Многоагентные системы рассматриваются как ключевая технология для реализации электронной коммерции. Следовательно, выразительные возможности языка общения КИА должны быть достаточными для того, чтобы представлять содержание произвольных коммерческих переговоров и контрактов, заключенных в результате этих переговоров. Поэтому формальные языки для представления содержания коммерческих переговоров и контрактов являются предметами исследования в новых научных направлениях в области

МАС, называемых *электронными переговорами* (e-negotiations) и *электронным заключением контрактов* (electronic contracting).

Между тем, выразительные возможности семантического языка FIPA SL довольно далеки от того, чтобы быть удобными для решения этой задачи. В связи с этим актуальна задача создания методов разработки более совершенных формальных языков - таких, которые были бы удобны для представления содержания любых посланий КИА, в том числе и для представления содержания произвольных коммерческих переговоров и контрактов.

Проблема разработки формальных языков-посредников для отображения содержания (или смысла) ЕЯ-текстов (другими словами, языков семантических представлений, или семантических языков) исследуется специалистами разных стран в течение более трех десятилетий. В нашей стране ряд аспектов этой проблемы в различные периоды изучались Ю.Д. Апресяном, И.М. Богуславским, В.М. Брябриным, Б.Ю. Городецким, А.К. Жолковским, А.П. Ершовым, Ю.И. Клыковым, О.С. Кулагиной, Е.С. Кузиным, Л.Т. Кузиным, И.П. Кузнецовым, Д.Г. Лахути, Н.Н. Леонтьевой, Л.И. Литвинцевой, Ю.Я. Любарским, М.Г. Мальковским, А.Г. Мацкевичем, И.А. Мельчуком, Л.И. Микуличем, А.С. Нариньяни, Г.С. Осиповым, Г.С. Плесневичем, Э.В. Поповым, Д.А. Поспеловым, В.Ш. Рубашкиным, В.А. Тузовым, З.М. Шаляпиной, Г.С. Цейтиным, Л.Л. Цинманом и другими учеными.

За рубежом наибольший вклад в разработку методов математического описания содержания (смысла) ЕЯ-текстов внесли Р. Монтегю (грамматики Монтегю), Дж. Барвайз и Р. Купер (теория обобщенных кванторов, ситуационная теория), М. Кресвелл (теория структурированных значений предложений), Й. Гронендейк и М. Стокхоф (динамические грамматики Монтегю, динамическая предикатная логика), Дж. Сова (теория концептуальных графов), Л. К. Шуберт и Ч.Х. Хуан (эпизодическая логика), Г. Камп и У. Рейль (теория представления дискурсов)

Несмотря на усилия, предпринимавшиеся в течение многих лет учеными разных стран, до последнего времени многие существенные аспекты проблемы формального описания содержания ЕЯ-текстов оставались мало изученными. Одна из основных причин этой ситуации заключается в том, что внимание

уделялось, главным образом, формализации смысловой структуры отдельных фраз, а не дискурсов. Кроме того, недостаточно изученной является проблема формального описания смысловой структуры отдельных фраз, обозначающих высказывания и включающих описания множеств и/или придаточные цели и/или слова “понятие”, “термин”, а также структуры фраз, выражающих команды и вопросы.

Наконец, сегодня ясно, что понимание ЕЯ-текста осуществляется в контексте системы знаний о мире и о целях интеллектуальных систем. Однако выразительные возможности большинства известных подходов к математическому описанию смысловой структуры ЕЯ-текстов (а именно, грамматик Монтегю, теории обобщенных кванторов, ситуационной теории, теории структурированных значений предложений, динамических грамматик Монтегю, динамической предикатной логики) недостаточны для построения теорий компьютерного понимания ЕЯ в контексте системы знаний о мире и о целях интеллектуальных систем. Например, исследования по дескриптивным логикам, выросшие из работ по терминологическим языкам представления знаний (ЯПЗ), показали полезность включения в состав ЯПЗ составных обозначений понятий. Однако перечисленные непосредственно выше подходы не предоставляют такой возможности.

Проблема автоматизации формирования баз знаний СИИ посредством извлечения информации из ЕЯ-текстов с помощью ЛП, проблема разработки семантического языка-посредника для устранения языкового барьера между пользователями сети Интернет и ряд других актульных научно-технических проблем требуют создания эффективных средств формального представления содержания произвольных ЕЯ-текстов, относящихся к *деловой прозе* (термин А.П. Ершова, ставший широко популярным в компьютерной лингвистике), т.е. ЕЯ-текстов, относящихся к юриспруденции, бизнесу, медицине, технике и т.д.

Между тем, перечисленные наиболее популярные подходы к формальному представлению содержания ЕЯ-текстов имеют ограниченную сферу применения. В частности, эти подходы не предоставляют адекватных формальных средств для представления содержания произвольных предложений с описаниями множеств или составными обозначениями понятий,

дискурсов со ссылками на смысл фраз и более крупных частей текстов, с обозначениями сложных целей, с косвенной речью.

Так, язык-посредник UNL ориентирован на представление содержания отдельных предложений, а не дискурсов. Кроме того, в языке UNL нет формальных средств описания множеств, средств формального различения описаний объектов и описаний понятий, квалифицирующих эти объекты, средств представления ссылок на смысл фраз и более крупных фрагментов дискурсов.

В связи с этим актуальна проблема разработки более мощных математических методов описания смысловой структуры реальных предложений и связных текстов, относящихся к юриспруденции, бизнесу, медицине, технике, экономике и т.д.

Наибольшие трудности при разработке ЛП связаны с выполнением преобразования “ЕЯ-текст ➔ Семантическое представление (СП) текста”. Однако анализ как отечественных, так и зарубежных публикаций показывает, что при разработке преобразователей ЕЯ-текстов в СП текстов крайне недостаточно используются формальные средства. Это выражается в неформальном и фрагментарном описании структуры лингвистической базы данных (ЛБД), т.е. базы данных (БД) с морфологической и семантико-синтаксической информацией о лексических единицах, а также методов обработки информации основными подсистемами преобразователя “ЕЯ-текст ➔ СП текста”.

Основная часть исследований по разработке ЕЯ-интерфейсов и ЛП других видов была реализована для английского языка, синтаксис которого существенно отличается от синтаксиса русского языка (РЯ). Чрезвычайно существенно то, что полные описания информационного и программного обеспечения таких ЛП, как правило, недоступны специалистам в нашей стране. Кроме того, одним из следствий экономической ситуации, сложившейся в 1990-е годы в нашей стране, является отсутствие даже в центральных библиотеках огромного количества публикаций в области разработки ЛП, опубликованных за рубежом в 1990-е и 2000-е годы на английском и некоторых других языках. Все это серьезно затрудняет подготовку специалистов в нашей стране в области

проектирования ЛП и сужает возможности принятия оптимальных проектных решений, приводит к дополнительным трудозатратам на разработку ЛП.

Учитывая сказанное, актуальной является проблематика разработки методов формального описания структуры ЛБД, а также таких методов семантико-синтаксического анализа текстов из представляющих практический интерес подъязыков русского языка, которые более широко используют формальные средства описания входных, промежуточных и выходных данных по сравнению с известными методами.

Разработка ЛП многих видов, например, ЕЯ-интерфейсов больших БД, отличается большой трудоемкостью. В связи с этим в данной книге выдвигается гипотеза о том, что в долговременной перспективе сокращению затрат и времени на разработку семейства ЛП в рамках одной организации или нескольких взаимодействующих организаций будет способствовать реализация в проектировании информационного и алгоритмического обеспечения ЛП следующих двух принципов:

(1) **принципа стабильности** используемого языка семантических представлений (ЯСП) по отношению к многообразию решаемых задач, многообразию предметных областей и многообразию программных сред (стабильность понимается как использование единой системы правил для построения конструкций ЯСП и варьируемого набора первичных информационных единиц, определяемого предметной областью и решаемой задачей);

(2) **принципа преемственности** алгоритмического обеспечения ЛП на основе использования одной или нескольких совместимых формальных моделей лингвистической БД и единых формальных средств представления промежуточных и окончательных результатов семантико-синтаксического анализа ЕЯ-текстов по отношению к многообразию решаемых задач, предметных областей и программных сред (преемственность понимается как многократное, максимальное использование эффективных алгоритмов, реализуемых подсистемами ЛП, в разных проектах ЛП).

В данной работе предпринята попытка создания значительной части предпосылок для реализации этих двух принципов при проектировании ЛП.

1.3. Подходы к формализации семантики естественного языка, разработанные в конце 1960-х – первой половине 1980-х годов

Основные результаты в теории лингвистических процессоров (ЛП) были получены с конца 1960-х годов. В 1970-е годы было достигнуто значительное продвижение вперед в отношении принципов "понимания" компьютерной системой естественного языка (ЕЯ). В большой степени этому способствовали работы Т.Винограда (Winograd 1971), У.Вудса и Р.Каплана (Woods, Kaplan 1971), Й. Уилкса (Wilks 1973), Р. Шенка и его коллег Ч. Ригера, Н. Голдмана, Ч. Ризбека (Schank и др. 1975). Проекты, реализованные этими исследователями, показали, в частности, что (а) понимание ЕЯ-выражения осуществляется в рамках базы знаний (БЗ) о мире и (б) целесообразно использовать специальные формальные выражения для отображения смысла ЕЯ-выражений, причем выбор подхода к построению этих формальных выражений значительно влияет на процесс разработки ЛП.

Наиболее популярными подходами к построению таких формальных структур для отображения смысла ЕЯ-текстов в 1970-е годы являлись теория семантических сетей (ТСС), импульс к появлению которой был дан работами Куиллиана (Quillian 1968) и Р.Саймонса (Simmons 1973), а также теория смысловой зависимости в естественном языке (ТСЗЕЯ) Р. Шенка (Schank 1972; Schank и др. 1975). Как известно, семантические сети являются ориентированными графами со специальными метками вершин и ребер. Метки вершин обозначают понятия, реальные предметы, ситуации (в частности, события), числа, значения цветов и т.д., а метки ребер соответствуют смысловым отношениям между элементами текста и/или между понятиями. ТСЗЕЯ предложила способы построения диаграмм определенных видов для отображения смысла фраз и коротких связных текстов. Оба этих подхода не были математическими, но ТСС использовала математическое понятие

ориентированного графа для иллюстрации способов представления содержания простых фраз и текстов в виде размеченного графа.

В нашей стране в 1970-е и 1980-е годы в процессе развития теории семантических сетей И.П. Кузнецовым (Кузнецов 1976, 1978, 1986) возникла теория расширенных семантических сетей (см. параграф 4.15 данной книги).

Успехи 1970-х годов по реализации экспериментальных проектов понимания компьютером отдельных фраз на ЕЯ и текстов, состоящих из нескольких простых фраз, позволили выдвинуть в 1980-е годы перед исследователями следующие новые задачи: (а) переход от обработки простых фраз к обработке связных текстов (дискурсов), включающих пропуски слов в отдельных фразах (явление эллипсиса), ссылки на ранее упомянутые объекты ("для этого предприятия" и т.п.) и ссылки на смысл предыдущих фраз и более крупных частей текста ("об этом", "этот метод" и т.п.); (б) эффективный учет прагматики общения, т.е. интерпретация очередного ЕЯ-выражения в контексте всего диалога; (в) создание формальных, предметно-независимых методов проектирования ЛП с целью получения возможности широкого тиражирования эффективных проектных решений.

Вопрос о необходимости эффективных формальных методов для проектирования ЛП возник совершенно естественно. Дело в том, что опыт реализации в 1970-х годах экспериментальных проектов ЛП показал, что ЛП, обеспечивающие информационные потребности реального пользователя в той или иной предметной области, будут являться сложными программными комплексами. При разработке сложных технических систем в различных предметных областях широко используются математические методы. Например, для конструирования самолетов разработана и используется аэродинамика, а для проектирования кораблей и подводных лодок применяется гидромеханика.

Достаточно полное представление о том, какие формальные инструменты для изучения семантики ЕЯ были доступны к середине 1980-х годов разработчикам ЛП за рубежом, дают учебник (Thayse и др. 1988) по логическим методам в научном направлении Искусственный Интеллект (ИИ), учебник (Partee, ter Meulen. и Wall, 1990) по математической лингвистике и учебники (Grishman 1986; Gazdar, Mellish 1989) по компьютерной лингвистике. Анализ этих

источников и целого ряда других публикаций показывает, что в этот период запас формальных методов для изучения семантики ЕЯ был довольно бедным в отношении математического описания смысловой структуры реальных связных текстов (или дискурсов) на ЕЯ и соответствия между текстами и их семантическими представлениями (СП) с учетом базы знаний о мире.

Общепринято считать, что история современных подходов к формализации семантики ЕЯ начинается с фундаментальных работ американского логика Р. Монтегю (Montague 1970, 1974a, 1974b). Подход к формализации семантики ЕЯ, изложенный в этих работах, впоследствии был назван Грамматикой Монтегю. Рядом исследователей были предложены различные расширения Грамматики Монтегю; в частности, такие расширения рассматриваются в работах (Partee 1976; Thomason 1980). В 1980-х годах было предпринято несколько попыток использовать подход Монтегю в проектировании ЛП. Однако теория формализации семантики ЕЯ, получившая название Грамматики Монтегю, (а) не является универсальной, (б) недостаточно удобна для практики с точки зрения вычислительной эффективности, (в) изложена ее автором в трудно воспринимаемой форме, что тормозило прямое использование этой теории на практике. Поэтому подход Монтегю был переработан и дополнен (иногда существенно) с целью использования его в проектировании ЛП. Так, Клиффорд (Clifford 1983, 1988) разработал определение формального языка QE-III для представления содержания вопросов к историческим базам данных. Это было сделано на основе расширения Грамматики Монтегю. Хирст (Hirst 1988) разработал фреймоподобный семантический язык FRAIL, отойдя довольно далеко от Грамматики Монтегю. Язык FRAIL использовался в семантическом интерпретаторе ABSITY для представления результатов обработки ЕЯ-текстов с целью включения этих результатов в базу знаний. Джоуси (Jowsey 1987) предложил упрощенную версию Грамматики Монтегю для построения СП текстов в прикладной интеллектуальной системе, выполняющей рассуждения общего вида. Сембок и Райсберген (Sembok & van Rijsbergen, 1990) применили язык Джоуси, близкий к языку логики первого порядка, в экспериментальной информационно-поисковой системе.

Обобщенные грамматики фразовых структур (Gazdar, Klein, Pullum, Sag, 1985) также могут расматриваться как расширения Грамматики Монтегю, поскольку в этих грамматиках язык интенсиональной логики Монтегю используется для построения СП предложений.

Другими основными подходами к формальному изучению семантики ЕЯ в первой половине 1980-х годов были теория обобщенных кванторов (Barwise, Cooper 1981; Gaerdenfors 1987; Peres 1991), ситуационная семантика (Barwise, Perry 1983; Fenstad, Halvorsen и др. 1987; Cooper 1991), теория представления дискурсов (Kamp, 1981; Kamp, Reyle 1990). Все эти подходы имеют общую отправную точку – пионерские работы Р. Монтегю - и ряд взаимосвязей.

Анализ показывает, что глубокая связь с традициями математической логики является главной причиной очень большого разрыва между возможностями этих подходов и требованиями, предъявляемыми практикой проектирования ЛП. В частности, можно выделить следующие ограничения этих подходов к формальному изучению ЕЯ:

1. Неадекватность с точки зрения описания структурированных значений дискурсов на ЕЯ. В частности, отсутствие выразительных возможностей для описания семантической структуры дискурсов, содержащих ссылки на смысл фраз и более крупных частей текста (такие ссылки могут задаваться, например, словами и выражениями “поэтому”, “об этом”, “данный метод”, “’это распоряжение”, “поставленный вопрос”).

2. Идущая от логики внутренняя ориентация на рассмотрение выражений, представляющих высказывания. Между тем, еще существуют выражения, обозначающие вопросы, команды, цели, действия, пожелания, советы, обещания, назначения вещей, поэтому необходим формальный аппарат для описания смысловой структуры таких выражений. В этой связи следует отметить, что за рубежом первые шаги в этом направлении были сделаны теорией структурированных значений предложений (Cresswell 1985; Chierchia 1989), а в нашей стране первые шаги такого рода были сделаны в публикациях автора (Фомичев 1978а, 1981 а, б, 1983).

3. Игнорирование или недостаточно глубокое рассмотрение многих важных особенностей структуры выражений, обозначающих высказывания. В частности, можно отметить отсутствие адекватных формальных методов описания: (а) множеств,операций над множествами и отношений на множествах; (б) назначений вещей; (в) семантической структуры фраз, содержащих причастные обороты и придаточные определительные предложения; (г) стрктуры фраз, в которых логические связки “и“, “или“ соединяют не обозначения высказываний, а обозначения различных объектов, понятий, множеств или назначений вещей (“считывание или запись данных”, “прием и отправка груза” и т.п.); (д) структуры предложений со словами “понятие”, “термин”.

4. Структура данных, позволяющих поставить в соответствие выражению на ЕЯ одно или несколько возможных семантических представлений (СП) либо не моделировалась, либо моделировалась нереалистично с точки зрения разработки ЛП, способных анализировать тексты, относящися к науке, технике, экономике, медицине или юриспруденции.

5. Хорошо известно, что понимание ЕЯ-текста человеком может существенно зависеть от знаний этого человека (другими словами, реципиента текста) о реальности. Между тем, основные подходы к формализации семантики ЕЯ, популярные в 1980-е годы, не обладали выразительной силой, необходимой для эффективного описания знаний о реальности, для построения моделей концептуальной памяти и т.д.

6. Как следствие, за рубежом не разрабатывались модели соответствия “Текст – Система знаний – Семантическое представление (или представления) текста”.

7. В 1980-е годы ряд исследователей отмечали необходимость и важность моделирования процессов использования ЕЯ в общении, т.е. с учетом целей интеллектуальных систем, реализуемых в процессе общения, и их знаний о мире в целом и о другом участнике диалога (Попов 1982; Fomitchov, 1983, 1984; Narin’yani, 1984; Фомичев, 1988б; Fomichov,

1992). Однако наиболее популярные в 1980-е годы формализмы, использовавшиеся для изучения семантики ЕЯ, не предоставляли такой существенной возможности.

Представляется, что перечисленные ограничения являются наиболее важными с точки зрения проектирования семантико-синтаксических анализаторов дискурсов, относящихся к науке, технике, экономике, медицине, а также для разработки ЕЯ-интерфейсов больших баз данных и знаний.

1.4. Роль формальных систем семантических представлений с большими выразительными возможностями в проектировании лингвистических процессоров

Совокупность задач, поставленных перед теорией ЛП в начале 1980-х годов, оказалась чрезвычайно трудной. Как следствие, развитие теории ЛП в 1980-е годы сильно замедлилось. Несмотря на реализацию значительного количества проектов конструирования ЛП в разных странах мира, существенного продвижения вперед не удавалось достичь.

Главная причина этого замедления заключалась в следующем. В ЕЯ причудливым образом взаимодействуют многочисленные механизмы кодирования и декодирования информации. Поэтому часто для того, чтобы "понять" даже довольно простые для человека фразы или дискурсы, компьютер должен привлекать знания о закономерностях различных уровней языка (морфологическом, синтаксическом, семантическом), а также знания о мире и о конкретной ситуации диалога. Например, для того чтобы узнать, какие из нескольких ранее упомянутых объектов обозначаются местоимением "их", может потребоваться проведение умозаключений здравого смысла и логических рассуждений. Аналогичная ситуация имеет место и для задачи восстановления смысловой структуры фраз с пропусками слов (эллиптичных фраз) в контексте всего дискурса или всего диалога.

Поэтому, пытаясь формализовать понимание компьютером даже довольно простых текстов, исследователи быстро убеждались в том, что для решения их частных задач необходимо предварительно иметь теоретические решения, относящиеся к произвольным текстам группы естественных языков (например, русского, английского, немецкого, французского). В итоге в 1980-е годы в англоязычных публикациях даже возникла метафора "theory bottleneck" ("узкое горлышко теории"), отражающая значительные трудности создания адекватной теории понимания компьютером ЕЯ.

Наконец, несколькими группами исследователей из разных стран (в том числе и автором данной работы) была предложена идея, позволяющая найти выход из охарактеризованной тупиковой ситуации. Суть этой идеи заключается в следующем. Необходимо разработать такие формальные языки для представления знаний о мире и построения семантических представлений (СП) ЕЯ-текстов, чтобы можно было конструировать СП в виде выражений, отражающих многое структурные особенности самих текстов. Другими словами, нужны формальные языки (или формальные системы, поскольку множество их правильно построенных выражений образует язык) для описания структурированных значений (или смыслов) ЕЯ-текстов, обладающие выразительными возможностями, близкими к возможностям ЕЯ. Тогда можно будет выполнять смысловой анализ текста в два этапа:

ЕЯ-текст Т ➔ *Недоопределенное СП текста Т* ➔ *Целевое СП текста Т* .

Эту схему следует понимать следующим образом. Сначала должно быть построено промежуточное, предварительное СП текста, называемое недоопределенным семантическим представлением (НСП) рассматриваемого текста. Это выражение в большинстве случаев будет отображать смысл входного текста Т лишь частично, неполно. Например, в НСП текста Т может отсутствовать указание на конкретный объект, соответствующий конкретному вхождению в текст Т местоимения "ей" или не выбрано конкретное значение слова "станция", входящего в Т.

Однако НСП текста Т является формальным выражением, в отличие от исходного ЕЯ-текста Т. Поэтому на втором этапе обработки Т для снятия той или иной недоопределенности можно будет вызвать одну из многочисленных

специализированных процедур-"экспертов" по конкретным вопросам. Такие процедуры можно будет проектировать с примененинем формальных средств представления информации, покольку базы знаний ЛП состоят из выражений формальных языков представления знаний, а исходное НСП - вход процедуры и преобразованное НСП (в частности, совпадающее с целевым СП) являются формальными выражениями. Впервые эта идея была высказана в работах (Фомичев 1981а, 1981б; Fomitchov 1983, 1984).

С конца 1980-х годов по настоящее время идея опоры при проектировании ЛП на формальные системы семантических представлений с широкими выразительными возможностями является центральной для развития теории понимания компьютером ЕЯ. Росту популярности этой идеи способствовало появление серии публикаций по эпизодической логике (ЭЛ) (Schubert и Hwang, 1989, 2000; Hwang и Schubert, 1993а-1995), реализация на основе ЭЛ проекта TRAINS, направленного на формализацию проблемно-ориентированного диалога на естественном (английском) языке (Allen, Schubert и др. 1995), осуществление проекта машинного перевода Core Language Engine (CLE) в Кембриджском отделении (Великобритания) Стенфордского исследовательского института (Alshawi и van Eijck, 1989; Alshawi, 1990, 1992), а также осуществление в 1980-х - 1990-х годах проекта SnepS в США (Shapiro, 1996).

Отправной точкой исследований для авторов перечисленных выше работ был язык логики предикатов первого порядка. Характер предпринимавшихся усилий по расширению выразительных возможностей этого языка (точнее, класса логики предикатов первого порядка) можно проиллюстрировать следующими двумя примерами.

Пример 1. В ЕЯ-текстах встречается большое количество таких обозначений различных объектов, которые являются сочетаниями «Прилагательное + Существительное» (для русского языка) или «Артикль + Прилагательное + Существительное» (для английского, немецкого и французского языков). ЭЛ позволяет рассматривать формальные аналоги значений таких словосочетаний. Например, сочетанию «маленький дом» может соответствовать фрагмент

семантического представления (СП) текста, являющийся выражением $\exists y : [\, y\ ((атрибут, маленький) дом)]$ (Hwang и Shubert, 1993a).

В ЕЯ-текстах часто встречаются и однородные члены предложения, например, «комнату или маленький дом». ЭЛ дает возможность строить и формальные аналоги выражений такого вида. В частности, в СП текста сочетанию «комнату или маленький дом» может соответствовать формальное выражение

$\exists y : [[y \text{ комната}] \vee [y\ ((\text{ атрибут маленький}) \text{ дом})]$ (Hwang, Shubert 1993a).

В логике предикатов первого порядка нет средств построения формальных аналогов словосочетаний вида «Прилагательное + Существительное» или «Существительное1 + ‘или’ + Прилагательное + Существительное2». Поэтому различные части СП текста могут отражать различные компоненты смысла словосочетаний подобного рода.

Пример 2. В проекте «Базовый Языковой Механизм» (Core Language Engine, или CLE) формулы, используемые для построения недоопределенных семантических представлений (НСП) предложений, называются квазилогическими формами (КЛФ). В частности, выражению «the three firms» (три фирмы) будет соответствовать КЛФ (см. Alshawi 1990) q_term(<t=quant, n=plur, l=all>), S, [subset, S, q_term(<t=ref, p=def, l=the, n=number(3)>, X,[firm, X])]).

Таким образом, язык КЛФ позволяет строить формальные аналоги некоторых естественно-языковых обозначений множеств (Alshawi 1990).

Сегодня наиболее популярным за рубежом подходом к формализации семантически-ориентированных компьютерных методов анализа дискурсов является эпизодическая логика (ЭЛ). Ее создатели, Л.К. Шуберт и Ч.Х. Хуан, в указанных выше работах предложили логику семантических представлений с широкими выразительными возможностями. Создание ЭЛ представляло собою значительный вклад в формальную теорию построения и понимания ЕЯ-дискурсов и, как следствие, в формальную теорию использования ЕЯ.

В то же время анализ показывает, что выразительная сила класса формул, рассматриваемых в ЭЛ, недостаточна с точки зрения представления основных принципов использования ЕЯ интеллектуальными системами. В первую

очередь, выразительные возможности ЭЛ являются существенно ограниченными с точки зрения представления структурированных значений (СЗ) сложных целей, команд, дискурсов с описаниями множеств, дискурсов со ссылками на смысл фраз и более крупных частей текста. Кроме того, ЭЛ не предоставляет средств для рассмотрения составных обозначений понятий в качестве термов и для описания операций над понятиями. Те же ограничения (и ряд других) относятся к языкам проекта Core Language Engine и SnepS.

Логика предикатов первого порядка послужила в 1980-х годах отправной точкой для создания не только ЭЛ, но и нескольких других направлений, относящихся к области логического программирования. Одним из наиболее интересных направлений, развивающимся более 15 лет, является атрибутная логика (АЛ), или логика со значениями свойств (Johnson 1988; Carpenter 1992; Carpenter, Penn 2001). Для АЛ характерен переход от рассмотрения однородного множества логических формул, описывающих факты, ситуации, к группировке формул, характеризующих одну сущность (человека, фирму и т.д.). Перечисленные выше ограничения ЭЛ относятся и к атрибутной логике.

Проблема формализации семантики ЕЯ-текстов в течение многих лет привлекает внимание исследователей и в нашей стране. Наиболее часто это внимание было обусловлено задачей выявления и формализации таких явлений семантического уровня ЕЯ, которые можно было бы эффективно использовать для представления знаний в прикладных интеллектуальных системах. В этой связи можно отметить цикл работ Д.А. Поспелова, Ю.И. Клыкова и ряда других авторов, в которых выделяются бинарные смысловые отношения между элементами предложения для решения задач ситуационного управления (Поспелов 1975, 1981, 1986; Клыков и Горьков, 1980), Г.С. Плесневича по теории логического вывода на ассоциативных сетях и понятийно-ориентированным языкам (Плесневич 1997 - 2003), Г.С. Осипова по использованию неоднородных семантических сетей в интеллектуальных системах приобретения знаний (Осипов 1990, 1997), В.Н. Вагина по проблеме обобщения знаний, представленных семантическими сетями (Вагин 1988).

Другую часть исследователей интересовала разработка формальных средств отображения содержания (смысла) текстов, анализируемых ЛП. В работе Н.Н.

Леонтьевой (Леонтьева 1981), по-видимому, впервые в нашей стране был высказан тезис о том, что для формализации диалога, осуществляемого в ограниченных предметных областях с помощью довольно простых текстов, нужен семантический язык с выразительными возможностями, близкими к возможностям ЕЯ.

Отдельные аспекты формализации семантики ЕЯ нашли отражение в работах В.М. Брябрина, Б.Ю. Городецкого, А.К. Жолковского, А.П. Ершова, О.С. Кулагиной, Е.С. Кузина, Л.Т. Кузина, И.П. Кузнецова, Д.Г. Лахути, Н.Н. Леонтьевой, Л.И. Литвинцевой, Ю.Я. Любарского, М.Г. Мальковского, А.Г. Мацкевича, И.А. Мельчука, Л.И. Микулича, А.С. Нариньяни, Э.В. Попова, Д.А. Поспелова, В.Ш. Рубашкина, В.А. Тузова, З.М. Шаляпиной, Г.С. Цейтина, Л.Л. Цинмана и ряда других ученых.

Однако ни в какой из публикаций отечественных авторов, во-первых, не предлагается формального аппарата, удобного для отображения поверхностной смысловой структуры произвольных ЕЯ-текстов, относящихся к деловой прозе. Во-вторых, не предлагается лингвистической теории, которую можно было бы положить в основу построения математической модели, удобной для представления смысловой структуры ЕЯ-текстов деловой прозы. Таким образом, вопрос о формальных языках, удобных как для построения семантических представлений произвольных ЕЯ-текстов, относящихся к деловой прозе, так и для моделирования ЕЯ-диалога интеллектуальных систем, является чрезвычайно актуальным и остается в доступной литературе открытым (исключение составляют публикации автора данной работы).

Прогресс в решении этого вопроса означал бы существенный шаг вперед в решении фундаментальной проблемы разработки модели русского языка, сформулированной А.П. Ершовым еще в 1986 году следующим образом: “Мы хотим как можно глубже познать природу языка, и в частности русского. Одним из выражений этого познания должна стать модель русского языка. Это формальная система, которая должна быть адекватной и равнообъемной живому организму языка, но в то же время она должна быть анатомически отпрепарированной, разъятой, доступной для наблюдения, изучения и изменения” (Ершов 1986, с. 12).

Глава 2

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ОПИСАНИЯ СИСТЕМЫ ПЕРВИЧНЫХ ЕДИНИЦ КОНЦЕПТУАЛЬНОГО УРОВНЯ, ИСПОЛЬЗУЕМЫХ ЛИНГВИСТИЧЕСКИМ ПРОЦЕССОРОМ

2.1. Постановка задачи

Проанализировав по доступной литературе состояние исследований в области формализации семантики ЕЯ, Перегрин (Peregrin 1990) пришел к выводу о том, что существующие логические системы не позволяют формализовать все аспекты семантики ЕЯ, являющиеся важными для проектирования ЛП. Поэтому “мы не можем использовать имеющуюся форму логики как плавильную форму, в которую любой ценой необходимо втиснуть естественный язык”, и для создания адекватной формальной теории семантики ЕЯ необходимо выполнить системное лингвистическое исследование всех компонентов ЕЯ и установить взаимосвязи между логическими подходами к формализации семантики ЕЯ и лингвистическими моделями смысла.

В сущности, к тому же выводу (но значительно раньше, на рубеже 1970-х - 1980-х годов) пришел автор данной монографии. Этот вывод явился отправной точкой для разработки излагаемой ниже постановки задачи, а также постановки задачи в главе 3.

Представляется, что границы традиционной математической логики слишком узки для того, чтобы предоставить адекватную основу для компьютерно-ориентированной формализации семантики ЕЯ. Поэтому задача создания логических основ проектирования интеллектуально мощных систем смысловой обработки ЕЯ требует не только расширения логики первого порядка, но, скорее, разработки новых математических систем, совместимых с логикой предикатов первого порядка и позволяющих формализовать логику использования ЕЯ интеллектуальными системами.

Мы будем исходить из гипотезы о том, что существует единственный ментальный уровень для представления смысла ЕЯ-выражений, который можно

назвать концептуальным уровнем, но не семантический и концептуальный уровни отдельно. Эту гипотезу подддерживают многие ученые (см., например, Meyer 1994).

Анализ показывает, что первым шагом на пути создания широко применимого и предметно-независимого математического подхода к описанию структурированных значений ЕЯ-текстов должна являться разработка формальной модели, перечисляющей первичные (т.е. не составные) единицы концептуального уровня, используемые ЛП, а также описывающей информацию, связанную с такими единицами и необходимую для соединения таких единиц в составные единицы, отображающие структурированные значения сколь угодно сложных ЕЯ-текстов.

С целью построения формальной модели, обладающей указанным свойством, был, во-первых, проведен анализ лексического состава русского, английского, немецкого и французского языков.

Во-вторых, был изучен состав первичных информационных единиц, используемых в современных языках представления знаний в прикладных интеллектуальных системах, в частности, используемых в терминологических языках представления знаний.

На основании проведенного исследования в данной главе ставится задача разработки такой предметно-независимой математической модели для описания системы первичных единиц концептуального уровня, используемых ЛП, и информации, связанной с такими единицами, которая, во-первых, конструктивно учитывает существование следующих явлений естественного языка:

На множестве понятий задана иерархия по степени их общности. Например, понятие “физический объект” является частным случаем понятия “пространственный объект”.

Нередко один и тот же предмет может быть охарактеризован с помощью нескольких понятий, ни одно из которых не является частным случаем другого; такие понятия как бы дают значения “координат объекта” по разным “семантическим осям”. Например, каждый человек является физическим объектом, способным перемещаться в пространстве. С другой стороны, каждый

человек является интеллектуальной системой, поскольку люди могут решать задачи, читать, сочинять стихи и т.д.

В русском языке есть такие слова, как “некоторый”, “определенный”, “каждый” , “какой-нибудь” , “все” , “несколько”, “большинство” и ряд других, которые в предложениях всегда присоединяются к словам и словосочетаниям, обозначающим понятия. Например, мы можем построить выражения “каждый человек” , “какой-нибудь автомобиль”, “все люди” , “несколько книг” и т.д. Аналогичные слова есть в английском, немецком, французском и многих других языках.

Во-вторых, модель должна позволять различать формальным образом обозначения первичных единиц концептуального уровня, соответствующих:

(2.1) объектам, ситуациям, процессам в реальном мире и понятиям, квалифицирующим (характеризующим) эти объекты, ситуации, процессы;

(2.2) объектам и множествам объектов;

(2.3) понятиям, квалифицирующим объекты, и понятиям, квалифицирующим множества объектов тех же видов (“корабль” и “эскадра” и т.д.);

(2.4) упорядоченным n-местным наборам различных сущностей, где n > 1 (“упорядоченная пара” и т.д.) и множествам.

В-третьих, модель должна учитывать, что совокупность первичных единиц концептуального уровня включает:

(3.1) единицы, соответствующие логическим связкам “и” , “или” , “не” и логическим кванторам существования и всеобщности;

(3.2) именам нетрадиционных функций с аргументами и/или значениями, являющимися: (3.2.1) множествами предметов, ситуаций (событий); (3.2.2) понятиями, (3.2.3) множествами понятий; (3.2.4) семантическими представлениями (СП) ЕЯ-текстов, (3.2.5) множествами СП ЕЯ-текстов;

(3.3) единицу, соответствующую слову “понятие” и отличающуюся от концептуальной единицы “понятие “; первая из упомянутых единиц вносит, например, вклад в формирование значения выражения “важное понятие, используемое в физике, химии и биологии”.

Итогом решения поставленной задачи станет определение в параграфе 2.8 класса формальных объектов, называемых концептуальными базисами. Описание класса концептуальных базисов является математической моделью, перечисляющей первичные единицы концептуального уровня, используемые ЛП, а также описывающей информацию, связанную с такими единицами и необходимую для соединения этих единиц в составные единицы, отображающие структурированные значения сколь угодно сложных ЕЯ-текстов. Данная модель является первой частью теории К-представлений (концептуальных представлений).

2.2. Базовые обозначения и вспомогательные определения

2.2.1. Общематематические обозначения

$x \in Y$ элемент x принадлежит множеству Y

$x \notin Y$ элемент x не входит в множество Y

$X \subset Y$ множество X является подмножеством множества Y

$Y \cup Z$ объединение множеств Y и Z ; $Y \cap Z$ пересечение множеств Y и Z

$Y \setminus Z$ теоретико-множественная разность множеств Y и Z, т.е. совокупность всех таких элементов x из Y , что x не входит в Z

$Z_1 \times \ldots \times Z_n$ декартово произведение множеств Z_1 , …, Z_n , где $n > 1$

$\varnothing$ пустое множество

$\forall$ для любого, для любых $\exists$ существует

$\rightarrow$ следует, влечет за собой $\Leftrightarrow$ тогда и только тогда

2.2.2. Предварительные определения и обозначения из теории формальных грамматик и языков

Определение. Алфавитом называется конечное множество символов. Если A-произвольный алфавит, то $A^+ = \{d_1,\ldots, d_n \mid n \geq 1\}$, где для $i = 1,\ldots, n$ $d_i \in A$.

Обычно вместо $d_1,\ldots,d_n$ для упрощения пишут $d_1\ldots d_n$.

Пример. $A=\{0,1\}$, 011, 11011, $0,1 \in A^+$.

Определение. Элементы множества A^+ называются непустыми цепочками (или непустыми строками) в алфавите A (над алфавитом A).

Пусть A- произвольный алфавит, d – символ из A, тогда $d^1=d$, для $n>1$ $d^n = dd\ldots d$ (n раз).

Определение. Пусть $A^* = A^+ \cup \{e\}$, где e - пустая цепочка. Тогда цепочками (или строками) в алфавите A (или над алфавитом A) называются элементы множества A^*.

Определение. Для каждого $t \in A^*$ определено значение функции *Длина (t)* (обозначаемой также через t) следующим образом: (1) $e = 0$; (2) если $t=d_1\ldots d_n$, $n \geq 1$, для $i=1,\ldots, n$ $d_i \in A$, то $t = n$.

Определение. Пусть A - произвольный алфавит. Тогда формальным языком (или, для краткости, языком) в алфавите A (или над алфавитом A) называется произвольное подмножество L множества A^*, то есть $L \in A^*$.

Пример. Пусть $A=\{0,1\}$, $L_1 = \{0\}$, $L_2=\{e\}$, $L_3=\{0^{2k}1^{2k} \mid k \geq 1\}$, тогда L_1, L_2, L_3 - языки в алфавите A.

2.2.2. Используемые определения из теории алгебраических систем

Определение. Пусть $n \geq 1$, Z – произвольное непустое множество. Тогда декартовой n-степенью множества Z называется (и обозначается через Z^n) множество Z при $n = 1$ и множество всех упорядоченных наборов вида $(x_1, x_2, \ldots, x_n)$, где $x_1, x_2, \ldots, x_n$ - элементы множества Z, при $n>1$.

Определение. Пусть $n \geq 1$, Z – произвольное непустое множество. Тогда n-арным (или n-местным) отношением на множестве Z называется произвольное подмножество R множества Z^n - декартовой n-степени множества Z. . При $n = 1$ отношение R называется *унарным отношением (*в этом случае R является произвольным подмножеством множества Z*)*, а при $n = 2$ отношение R называется *бинарным отношением* (Ершов 1970).

Пример. Пусть *Z1* – множество всех целых чисел, и *Odd* – подмножество всех нечетных чисел. Тогда *Odd* является унарным отношением на *Z1*. Пусть *Less*

является множеством всех упорядоченных пар вида *(x,y)* , где *x, y* – произвольные элементы множества *Z1*, и число *x* меньше числа *y*. Тогда *Less* – бинарное отношение на множестве *Z1*.

Очень часто вместо записи $(b,c)\in R$, где *R* – бинарное отношение на произвольном множестве *Z* , *b, c* - произвольные элементы из *Z*, используется запись *b R c* .

Определение. Пусть *Z* – произвольное непустое множество, *R* – бинарное отношение на *Z*. Тогда

(а) если для любого $a\in Z$ $(a,a)\in R$, то *R* – рефлексивное отношение;

(б) если для любого $a\in Z$ $(a,a)\notin R$, то *R* – антирефлексивное отношение;

(в) если для любых *a, b, c*$\in Z$ из $(a,b)\in R$, $(b,c)\in R$ следует, что $(a,c)\in R$, то *R* называется транзитивным отношением;

(г) если для любых *a,b* $\in Z$ из $(a,b)\in R$ следует $(b,a)\in R$, то *R* – симметричное отношение;

(д) если для любых *a, b* $\in Z$ *из* $a\neq b$ и $(a,b)\in R$ следует, что $(b,a)\notin R$, то *R* – антисимметричное отношение;

(е) если *R* - рефлексивно, транзитивно и антисимметрично, то *R* – частичный порядок на *Z* (Johnsonbaugh 2001).

Пример. Бинарное отношение *Less* из предыдущего примера является транзитивным, антирефлексивным и антисимметричным.

Пример. Пусть *Z1* – множество всех целых чисел, и *Eqless* является множеством всех упорядоченных пар вида *(x,y)* , где *x, y* – произвольные элементы множества *Z1*, и число *x* равно числу y или меньше числа *y*. Тогда *Eqless* – бинарное отношение на множестве *Z1*. Это отношение является рефлексивным, транзитивным и антисимметричным. Таким образом, отношение *Eqless* является частичным порядком на *Z1*.

Пример. Пусть *Z2*– множество всех понятий, которые обозначают транспортные средства, и *Genrel* является множеством всех упорядоченных пар вида *(x,y)* , где *x, y* – произвольные элементы множества *Z2*, и понятие *x* совпадает с понятием *y* или является обобщением понятия *y*. Например, понятие *корабль* является обобщением понятия *ледокол* ; следовательно, пара (*корабль*, *ледокол)* входит в множество *Genrel*. Очевидно, что *Genrel* – бинарное

отношение на множестве *Z2*. Это отношение является рефлексивным, транзитивным и антисимметричным. Таким образом, отношение *Genrel* является частичным порядком на *Z2*.

Определение. Пусть Z – произвольное непустое множество, R – бинарное отношение на Z. Тогда элементы $a, b \in Z$ называются сравнимыми для R, если либо $(a,b) \in R$, либо $(b,a) \in R$.

2.3. Краткая характеристика предлагаемой математической модели для описания системы единиц концептуального уровня, используемых лингвистическим процессором

С математической точки зрения, решение задачи, поставленной в параграфе 2.1, является определением нового класса формальных объектов, называемых *концептуальными базисами (к.б.)*. Отдаленным прообразом этого понятия является понятие сигнатуры алгебраической системы (Ершов, Палютин 1979).

Каждый к.б. B является упорядоченным набором вида

$$((c_1, c_2, c_3, c_4), (c_5,..., c_8), (c_9,..., c_{15}))$$

с компонентами $c_1, c_2, ..., c_{15}$, являющимися (главным образом) конечными или счетными множествами символов и выделенными элементами таких множеств. В частности, $c_1 = St$ - конечное множество символов, называемых сортами и обозначающих наиболее общие рассматриваемые понятия, $c_2 = P$ - выделенный сорт "смысл собщения", $c_5 = X$ - счетное множество цепочек, используемых как "строительные блоки" для формирования модулей знаний и семантических представлений (СП) текстов, $c_6 = V$ - счетное множество переменных, $c_8 = F$ - подмножество множества X, элементы которого называются функциональными символами.

Компонент $c_3 = Gen$ является таким бинарным отношением (частичным порядком) на St, что если пара (s,u) входит в Gen, то либо $s = u$, либо понятие, соответствующее сорту u , является конкретизацией понятия, соответствующего сорту s. Компонент $c_7 = tp$ является отображением из объединения множеств X и V в некоторое счетное множество Tps цепочек, называемых типами и характеризующих элементы из X и V.

Предположим, например, что *X* включает элементы *интс* , *дин.физ.об*, *чел*, *редсовет*, *Д.И.Менделеев*, обозначающие сорт "интеллектуальная система", сорт “динамический физический объект”, понятия “человек”, "редакционный совет" и конкретного человека – выдающегося химика Дмитрия Ивановича Менделеева. Будем рассматривать символ $\uparrow$ как индикатор почти всех типов, связанных с понятиями. Тогда значениями отображения *tp* для элементов *чел*, *редсовет* , *Д.И.Менделеев* будут элементы $\uparrow$ *интс* дин.физ.об*, $\uparrow$ {*интс* дин.физ.об* } и *интс* дин.физ.об* соответственно. Если же в качестве элемента множества *X* мы рассматриваем обозначение редакционного совета конкретного издания, то для такой информационной единицы отображение *tp* примет значение *{интс*дин.физ.об}* . Таким образом, типы помогают различать (а) объекты и понятия, характеризующие эти объекты, (б) множества и понятия, характеризующие эти множества.

Определение класса концептуальных базисов будет использовано в главе 3 следующим образом. Каждому к.б. *B* будут поставлены в соответствие три множества формул *Ls* = *Ls*(*B*), *Ts* = *Ts*(*B*), Y*s* = *Ys*(*B*) (*l*-формулы, *t*-формулы, *y*-формулы). Множество *Ls*(*B*) будет названо *стандартным К-языком в базисе B*. Его цепочки подходят для построения семантических представлений (СП) текстов на естественном языке. Каждая формула из *Ts*(*B*) имеет вид *d* & *t*, где *d* $\in$ *Ls*(*B*), *t* – тип из *Tps*(*B*). Формулы из *Ys*(*B*) имеют вид a_1 & …& a_n & *d*, где a_1, …,a_n, *d* $\in$ *Ls*(*B*), *n* имеет разные значения для разных *d*, цепочка *d* строится из a_1, …,a_n как из элементарных информационных единиц (некоторые из них могут быть немного преобразованы) однократным применением некоторого правила построения.

Например, к.б. В можно определить так, чтобы выполнялись соотношения

Ls(B) ∋ страна, нек страна , нек страна : x1,

Столица (нек страна : x1), (Столица (нек страна : x1)≡Москва);

Ts(B) ∋ страна & ↑простр.об, нек страна & простр.об,

нек. страна : x1 & простр.об, Столица (нек страна : x1) & простр. об,

(Столица (нек страна : x1) ≡ Москва) & сообщ,

Ys(B) ∋ нек & страна & нек страна, нек страна & x1 & нек страна : x1,

Столица & нек страна : x1 & Столица (нек. страна : x1),

Столица (нек страна : x1) & ≡ & Москва & (Столица (нек страна : x1) ≡ Москва) , где *сообщ=P(B)* – выделенный сорт “смысл сообщения” для рассматриваемого к.б. *B, нек* – информационная единица, соответствующая словам “некоторый” “некоторая”, “некоторое”.

2.4. Основные идеи определения класса сортовых систем

Начнем решать поставленную задачу. Будем предполагать, что необходимо построить формальное описание некоторой предметной области (ПО), и рассмотрим первые шаги в этом направлении.

Шаг 1. Введем в рассмотрение конечное множество символов, обозначающих наиболее общие понятия ПО: пространственный объект, физический объект, интеллектуальная система, натуральное число и т.д. Будем считать, что каждое такое понятие характеризует сущность, не рассматриваемую как упорядоченный набор других сущностей или как множество, состоящее из каких-то других сущностей. Обозначим это множество символов через St и будем называть его элементы сортами.

Шаг 2. Выделим в St некоторый сорт, который будем связывать с семантическими представлениями (СП) ЕЯ-текстов, выражающих отдельные высказывания либо являющихся связными повествовательн ыми текстами. Обозначим такой сорт через P и назовем его сортом «смысл сообщения». Например, для каких-то применений роль выделенного сорта P может играть цепочка сообщ. Часть формул, которые мы будем рассматривать в этой работе, представима в виде F & t , где F - СП ЕЯ-выражения, a t – цепочка, классифицирующая данное выражение. Тогда, если t = P, то подформула F интерпретируется как СП простого или сложного высказывания (другими словами, сообщения). В частности, так может интерпретироваться формула *(Вес(нек блок1 : x3) ≡ 4/тонна) & сообщ* .

Шаг 3. Введем иерархию понятий на множестве сортов *St* с помощью некоторого бинарного отношения *Gen* на *St*, т.е. выделим некоторое подмножество $Gen \subset St \times St$. Например, могут выполняться соотношения

(цел, нат), (вещ, цел), (физ. об, дин. физ. об), (простр. об, физ. об) $\in$ *Gen.*

Шаг 4. Многие объекты могут быть охарактеризованы с разных точек зрения, у них есть «координаты» по разным «семантическим осям». Например, к конкретному университету можно подъехать или подойти, поэтому каждый университет имеет семантическую координату “пространственный объект”. У университета есть руководитель (ректор) , поэтому университеты имеют семантическую координату “организация”. Наконец, университет может разработать некоторую технологию или некоторый прибор; следовательно, представляется разумным считать, что университеты имеют семантическую координату “интеллектуальная система”.

Учитывая эти соображения, введем бинарное отношение совместимости (толерантности) *Tol* на множестве *St*. Это отношение интерпретируется следующим образом: если $(s,u) \in Tol \subset St \times St$, то существует такая сущность *x* в рассматриваемой ПО, что с *x* можно связать сорт *s* по одной семантической оси и сорт *u* по другой оси, причем сорт *s* и сорт *u* не являются сравнимыми для отношения *Gen*.

Например, множества *St* и *Tol* могут быть определены так, что *Tol* включает упорядоченные пары *(простр.объект, организация), (простр.объект, интел.система), (организация, интел.система), (организация, простр.объект), (интел.система, простр.объект), (интел.система, организация)* .

Из рассмотренной интерпретации отношения *Tol* вытекают следующие свойства: (*1)* $\forall u \in St\ (u,u) \notin Tol$, т.е. *Tol* – антирефлексивное отношение; *2)* $\forall u,t \in St$ из $(u,t) \in Tol$ следует, что $(t,u) \in Tol$, т.е. *Tol* – симметричное отношение.

Сортовой системой (с.с.) будем называть произвольную четверку *S* вида *(St, P, Gen, Tol)*, компоненты которой удовлетворяют определенным условиям.

2.5. Формальное определение сортовой системы

Определение. *Сортовой системой (с.с.)* будем называть произвольную упорядоченную четверку *S* вида

$$(St, P, Gen, Tol), \qquad (2.5.1)$$

где *St* – конечное множество символов, $P \in St$, *Gen* – непустое бинарное отношение на *St*, являющееся частичным порядком на *St* (т.е. рефлексивным,

транзитивным и антисимметричным), *Tol* – бинарное отношение на *St*, являющееся антирефлексивным и симметричным, и выполняются следующие условия:

(1) *St* не включает символы *‘↑’, ‘{‘ , ‘}’ , ‘(‘ , ‘)’ , ‘,’ , [сущн], [пон], [об], [↑сущн], [↑пон], [↑об];* (2) *St \ {P}≠∅ и ∀ u∈St \ {P} u, P* – несравнимы как для отношения *Gen*, так и для отношения *Tol*; (3) *∀ t,u∈ St* из *(t,u)∈ Gen* или *(u,t)∈ Gen* следует, что *t,u* несравнимы для отношения *Tol;* (4) *∀ t1,u1∈ St, t2,u2∈St* из *(t1,u1)∈ Tol, (t2,t1)∈ Gen, (u2,u1)∈ Gen* вытекает, что *(t2,u2)∈ Tol*.

Элементы множества *St* называются сортами; *P* – сортом «смысл сообщения»; *Gen ⊂ St×St* – отношением общности; *Tol ⊂ St×St* – отношением толерантности (совместимости). Если *(u,t)∈Gen*, то будем использовать эквивалентную запись *u→t* и говорить, что *t* – конкретизация сорта *u*, а *u* – обобщение сорта *t*. Если *(s,u)∈ Tol*, то будем использовать запись *s⊥u* и говорить, что сорт *s* совместим с сортом *u*.

Символы *‘↑’, ‘{‘ , ‘}’ , ‘(‘ , ‘)’ , ‘,’ , [сущн], [пон], [об], [↑сущн], [↑пон], [↑об]* будут играть особые роли при построении из сортов цепочек, называемых типами и классифицирующих сущности, рассматриваемые в выбранной предметной области (см. параграф 2.6).

Пример. Пусть *St_0={нат, цел, вещ, простр.об, физ.об., дин.физ.об, вообр.об, интс, орг, сит, соб, мом, сообщ}.* Элементы множества *St_0* обозначают понятия и интерпретируются следующим образом: *нат* – «натуральное число», *цел* – “целое число“, *вещ* – «вещественное число», *простр.об* – «пространственный объект», *физ.об* – «физический объект», *дин.физ.об* – «динамический физический объект», *вообр.об* – «воображаемый пространственный объект» (орбиты небесных тел, геометрические фигуры), *интс* – «интеллектуальная система», *орг* – “организация“, *сит* – «ситуация», *соб* – “событие“ (т.е. динамическая ситуация), *мом* – «момент времени», *сообщ* – «семантическое представление сообщения».

Пусть *P_0=сообщ, Ge1={(u,u) | u∈St_0}* ,

Ge2={(цел, нат), (вещ, цел), (вещ, нат), (простр.об, физ.об), (простр.об, вообр.об), (физ.об, дин.физ.об), (простр.об, дин.физ.об), (сит, соб)},

$Gen_0 = Ge1 \cup Ge2$,

$T1 = \{$(интс, дин.физ.об), (интс, физ.об), (интс, простр.об), (орг, интс), (орг, физ.об), (орг, простр.об)$\}$, $T2 = \{(u,s) \mid (s,u) \in T1\}$, $Tol_0 = T1 \cup T2$.

Пусть $S_0 = (St_0, P_0, Gen_0, Tol_0)$. Тогда легко проверить, что S_0 является сортовой системой, и сорт *сообщ* является выделенным сортом «смысл сообщения» этой системы. Из определения множества Gen_0 вытекают, в частности, следующие соотношения:

вещ→цел, цел→нат, вещ→нат, простр.об.→физ.об, простр.об.→вообр.об, физ.об.→дин.физ.об; интс⊥физ.об, интс⊥дин.физ.об, интс⊥орг, физ.об⊥интс, дин.физ.об⊥интс.

2.6. Типы, порождаемые сортовыми системами, и конкретизации типов

2.6.1. Определение множества типов

Предположим, что нам необходимо описать некую предметную область (ПО), и мы решили рассматривать некоторые сущности как элементарные сущности (люди, фирмы, числа, факты, понятия и т. д.). Тогда определим составные сущности для данной области как такие сущности, которые рассматриваются как упорядоченные наборы других сущностей или как множества, состоящие из каких-то других сущностей. Будем интерпретировать понятия(другими словами, концепты) как общие описания сущностей, относящихся к некоторым различаемым людьми классам сущностей. Объекты определим как такие сущности, которые не рассматриваются как понятия. Класс объектов включает, в частности, семантические представления (СП) текстов, множества СП текстов и множества понятий.

Определим для каждой с. с. *S* множество цепочек *Tp(S)*, элементы которого назовем типами системы *S* и будем понимать их как характеристики сущностей, рассматриваемых в рассуждениях о данной области. При построении типов используются сорта из *S* и специальные символы *[сущн]*, *[пон]*, *[об]*, *[↑сущн]*, *[↑пон]*, *[↑об]*, ‘↑’, ‘{‘ , ‘}’, ‘(‘ , ‘)’, ‘,‘ (запятая). Символы *[сущн]*, *[пон]*, *[об]*

будем называть, соответственно, типом «сущность», типом «концепт» (это наиболее общая характеристика понятий) и типом «объект» (это наиболее общая характеристика сущностей, не рассматриваемых как понятия). Символ «*» будет используется для соединения нескольких совместимых сортов (т.е. сравнимых для отношения толерантности *Tol*) при построении цепочек из множества *Tp(S).* Символ '↑' будем интерпретировать как индикатор типа понятия.

Предположим, что мы используем сортовую систему S_0 , построенную в примере из параграфа 2..5. Тогда мы сможем связать с понятием "человек" тип *↑интс*дин.физ.об* из *Tp(S_0)* , с каждым конкретным человеком – тип *интс*дин.физ.об* , с понятием "студенческая учебная группа" – тип *↑{интс*дин.физ.об}* , с конкретной студенческой группой М8-05 факультета прикладной математики МИЭМ – тип *{интс*дин.физ.об}.*

Рассмотрим интерпретацию специальных символов *[сущн]*, *[пон]*, *[об]*. Формализуя рассуждения, условимся исходить из следующих рекомендаций. Если природа сущности z, рассматриваемой в рассуждении, не играет роли, то поставим в соответсвие z тип *[сущн]* в ходе рассуждения. Если же важно то, что *z* представляет собой объект, то поставим в соответсвие *z* тип *[об]*. Если, напротив, в отношении *z* важно то, что *z* является понятием, то поставим в соответсвие *z* тип *[пон]*. Назначение типов *[сущн]*, *[пон]*, *[об]* станет понятным из следующих примеров.

Пусть E_1 и E_2 соответствуют выражениям "первая сущность, упомянутая на странице 12 выпуска газеты "The Moscow Times", опубликованного 1 октября 1994 г.", и "первый объект, упомянутый на странице 12 выпуска газеты "The Moscow Times", опубликованного 1 октября 1994 г.". Тогда можно связать типы [сущн] и [об] с сущностями, на которые ссылаются в E_1 и E_2 соответственно, в случае, если мы не читали страницу 12 указанного выпуска.

Однако, прочитав эту страницу, мы узнаем, что первая сущность и первый объект, упомянутые на этой странице, — город Мадрид. Следовательно, теперь мы можем связать с упомянутой сущностью (объектом) более информативный тип простр.об («пространственный объект»).

Пусть E_3 — выражение "понятие с меткой AC060, определенное в Longman Dictionary of Scientific Usage (Moscow, Russky Yazik Publishers, 1989)". Не читая

словаря, мы можем связать с понятием, упомянутым в E_3, только тип [пон]. Но после того, как мы найдем определение с пометкой AC060, мы узнаем, что это определение понятия “трубка” (полый цилиндр с длиной много больше диаметра). Следовательно, мы можем связать с понятием, упомянутым в Е3, более информативный тип ↑физ (обозначение понятия “физический объект”).

Будем предполагать, что цепочки [↑сущн], [↑пон], [↑об] – это типы семантических единиц, соответсвующих словам “сущность”, “понятие”, “объект”. Эти типы образуют множество специальных типов Spectp.

Условимся в последующих определениях считать символами как цепочки [сущн], [пон], [об], [↑сущн], [↑пон], [↑об] , так и элементы сортовых множеств.

Определение. Пусть S — с.с. вида (1), Spectp = {[↑сущн], [↑пон]}, [↑об]}, Toptp = {[сущн], [пон]}, [об]}. Тогда через Tp(S) обозначим наименьшее множество T, удовлетворяющее следующим условиям:

(1) Spectp ∪ Toptp∪ St∪ {↑s | s∈ St} ⊆ T ; элементы множеств Spectp и Toptp называются специальными типами и верхними типами, соответственно;

(2) Если k>1, для ∀ i=1,…,k s_i∈ St, ∀ i,j=1,…,k из i≠j следует, что $s_i \perp s_j$ (т.е. цепочки s_i, s_j сравнимы для отношения совместимости Tol), то цепочка $s_1*s_2*\ldots*s_k$ и цепочка ↑ $s_1*s_2*\ldots*s_k$ входят в T;

(3) Если n > 1, для i = 1,...,n t_i∈ T \ Spectp, то цепочка вида $(t_1,...,t_n)$ входит в T;

(4)Если t∈ T \ Spectp , то цепочка {t} входит в T;

(5)Если t∈T \ (Spectp ∪ Toptp), и t начинается с символа ‘(‘ или ‘{’, то цепочка ↑t входит в T.

Множество Tp(S) называется множеством типов, порождаемых с.с. S .

Определение. Если S — с.с., то Mtp(S) = Tp(S) \ Spectp; элементы множества Mtp(S) называются основными типами (обозначение этого множества происходит от английского словосочетания main types).

2.6.2. Интерпретация определения множества типов

Сформулируем принципы установления соответствия между сущностями, рассматриваемыми в предметной области с с.с. S и типами из множества Mtp(S).

Типы понятий, в отличие от типов объектов, начинаются с символа ‘↑’. С понятием, обозначаемым сортом с, свяжем тип ↑ с. Тип {t} соответствует любому множеству сущностей типа t. Если $x_1,...,x_n$ — сущности типов $t_1,...,t_n$, тогда тип $(t_1,...,t_n)$ соответствует n-местному упорядоченному набору $(x_1,...,x_n)$. Как следствие, множества, состоящие из упорядоченных наборов с типом $(t_1,...,t_n)$, будут иметь тип $\{(t_1,...,t_n)\}$.

Пример 1. Можно связать типы из Mtp(S) с некоторыми понятиями и объектами с помощью следующей таблицы:

ПОНЯТИЕ	ТИП
понятие “множество”	↑{[сущн]}
понятие “множество объектов”	↑{[об]}
понятие “множество понятий”	↑{[пон]}
понятие “человек”	↑интс*дин.физ.об
Д.И.Менделеев	интс*дин.физ.об
понятие “студенческая группа”	↑{интс*дин.физ.об }
Группа М8-05	{интс*дин.физ.об }
понятие “пара целых чисел”	↑(цел, цел)
пара (12,144)	(цел, цел)

Можно также связать с отношением “Меньше” на целых числах тип {(*цел*, *цел*)}, с отношением “Принадлежать множеству” — тип {([сущн], {[сущн]})}, с отношением “Объект *Y* характеризуется понятием *C*” — тип {([*об*], [*пон*])}, а с отношением “Понятие *D* является обобщением понятия *C*” — тип {([*пон*], [*пон*])}.

Основная цель введения типов заключается в том, чтобы задавать семантические ограничения на атрибуты отношения, в частности, на аргументы и значения функций. Идею такого использования типов можно пояснить следующим образом. Пусть *c* – обозначение понятия (в частности, $c \in St$). Тогда через *Dt(c)* будем обозначать множество всех сущностей, которые могут быть охарактеризованы понятием *c*, и называть *Dt(c) денотатом* понятия *c*.

Например, *Dt(книга)* = множество всех книг, *Dt(человек)* = множество всех людей.

Пусть *R* – обозначение *n*-арного отношения, *n>1*. Тогда ограничение $(x_1,\ldots,x_n) \in R \Leftrightarrow x_1 \in Dt(s_1), \ldots, x_n \in Dt(s_n)$, где $s_1,\ldots, s_n \in St$, будем указывать с помощью значения некоторого отображения *tp*, определенного для *R*, следующим образом: $tp(R)=\{(s_1,\ldots, s_n)\}$. Аналогично, ограничение $(x_1,\ldots, x_n) \in R \Leftrightarrow x_1 \in Dt(c_1), \ldots, x_n \in Dt(c_n)$ будем представлять в виде $tp(R)=\{(tc_1, \ldots, tc_n)\}$, где $tc_1, \ldots, tc_n$ – типы, характеризующие сущности из рассматриваемой предметной области.

Пример 2. Семантические ограничения на атрибуты отношений *Брат*, *Расстояние* (последнее отношение является функцией, ставящей в соответствие двум пространственным объектам некоторое значение длины) можно представить следующим образом:

$$tp(Брат)=\{(интс*дин.физ.об,\ интс.*дин.физ.об)\},$$

$$tp(Расстояние)=\{(простр.об,\ простр.об,\ длин)\}.$$

2.6.3. Отношение конкретизации на множестве типов

Пусть *S* –произвольная сортовая система (с.с.). Зададим на множестве типов *Tp(S)* некоторое бинарное отношение, обозначаемое символом $\vdash$ и называемое *отношением конкретизации*. На множестве сортов *St* отношение $\vdash$ совпадает с отношением общности $\rightarrow$. Следующая система примеров демонстрирует требования к отношению $\vdash$: *[сущн]* $\vdash$ *[об]*, *[сущн]* $\vdash$ *[пон]*, *физ.об* $\vdash$ *дин.физ.об*, *дин.физ.об* $\vdash$ *интс * дин.физ.об*, *[пон]* $\vdash \uparrow$*интс*, *[пон]* $\vdash \uparrow$*интс * дин.физ.об*, *[об]* $\vdash$ *физ.об*, *[об]* $\vdash$ *{физ.об}*, *[об]* $\vdash$ *{(вещ, вещ)}*.

Основная идея определения отношения конкретизации заключается в следующем. Мы хотим, чтобы расстояние могло быть определено и между неподвижными физическими объектами, и между динамическими физическими объектами, и между воображаемыми динамическими физическими объектами. Все объекты таких видов являются частными случаями пространственных объектов. Учитывая это, будем использовать отношение конкретизации $\vdash$ следующим образом.

Пусть R – обозначение n-арного отношения, где $n>1$, и некоторое отображение tp ставит в соответствие R описание семантических ограничений на атрибуты $\{(t_1,\ldots,t_n)\}$, т.е. $tp(R)=\{(t_1,\ldots,t_n)\}$, где $n>1$, $t_1,\ldots,t_n \in Tp(S)$.

Будем полагать, что выражение $R(x_1,\ldots,x_n)$ выражает тот же смысл, что и выражение $(x_1,\ldots,x_n) \in R$. Тогда будем считать выражение $R(x_1,\ldots,x_n)$ допустимым $\Leftrightarrow$ существуют такие $u_1, \ldots,u_n \in Tp(S)$, что $\forall k=1,\ldots n \; t_k \mid\!\!- u_k$ и $x_k \in Dt(u_k)$, т.е. x_k входит в денотат понятия u_k.

Пример 3. Так как выполняются соотношения

простр. об $\rightarrow$ *вообр.простр.об, простр.об* $\rightarrow$ *физ.об, физ.об* $\rightarrow$ *дин.физ.об,*

то *простр.об* $\mid\!\!-$ *вообр.простр.об, простр.об* $\mid\!\!-$ *дин.физ.об* .

Поэтому допустимыми будут являться выражения $Pасст(x_1,x_2,l_1)$, $Pасст(z_1,z_2,l_2)$, где x_1,x_2 – обозначения двух автомобилей, z_1,z_2 – обозначения орбит двух конкретных небесных тел, и l_1, l_2 – обозначения некоторых значений длины.

Пример 4. Студенческие учебные группы является частными случаями множеств. Семантические ограничения на аргумент и значение функции “Количество элементов множества”, обозначаемой символом *Колич-элем*, можно задать соотношением *tp(Колич-элем)={({[сущн]}, нат)}* . Предположим, что база знаний интеллектуальной системы включает идентификатор студенческой группы *М8-05*, и отображение *tp* связывает с этим идентификатором тип *{интс*дин.физ.об}*. Таким образом, данная гипотетическая интеллектуальная система рассматривает объект, обозначаемый идентификатором М8-05, как некоторое множество людей (каждый человек является каг интеллектеальной системой, так и динамическим физическим объектом). Пусть *tp(14)=нат*. Так как *нат* $\rightarrow$ *нат* , то в случае выполнения соотношения *{[сущн]}* $\mid\!\!-$ *{интс*дин.физ.об}* выражение *Колич-элем(М8-05,14)* допустимо.

Определение 3. Пусть S – произвольная с.с. вида (2.5.1). Тогда *элементарными составными типами* будем называть цепочки из $Tp(S)$ вида $s_1*s_2*\ldots*s_k$, где $k>1$, для $\forall\, i=1,\ldots,k \; s_i \in St$.

Пример 5. Цепочка *интс*дин.физ.об* является элементарным составным типом для с.с. S_0 .

Определение 4. Пусть S – с.с. вида (2.5.1). Тогда через $Elt(S)$ обозначим объединение множества сортов St с множеством всех элементарных составных типов. Элементы множества Elt будем называть *элементарными типами.*

Определение 5. Если S – с.с. вида (2.5.1), $t \in Elt(S)$, то *спектр* типа t, обозначаемый через $Spr(t)$, в случае $t \in St$ является множеством $\{t\}$, а в случае $t = s_1 * s_2 * \ldots * s_k$, где $k>1$, для $\forall\ i=1,\ldots,k\ s_i \in St$, является множеством $\{ s_1,\ldots, s_k\}$.

Пример 6. Для с.с. S_0 спектр $Spr(физ.об) = \{физ.об\}$, $Spr(интс * дин.физ.об) = \{интс, дин.физ.об\}$.

Определение 6. Пусть S – с.с. вида (2.5.1), $u \in St$, t – элементарный составной тип из $Tp(S)$. Тогда тип t называется *уточнением* сорта u $\Leftrightarrow$ когда спектр $Spr(t)$ включает такой сорт w , что $u \rightarrow w$ (т.е. $(u,w) \in Gen$).

Пример 7. Пусть u = *физ.об* , t = *интс*дин.физ.об* . Тогда спектр $Spr(t)$ = { *интс, дин.физ.об* }. Поэтому из *физ.об* $\rightarrow$ *дин.физ.об* вытекает, что t - уточнение сорта u . Напомним, что в данной работе сорта считаются символами, т.е. неделимыми единицами.

Определение 7. Пусть $u \in St$, $t \in Tp(S)$, и t включает символ u. Тогда вхождение символа u называется *свободным* $\Leftrightarrow$ когда либо $t = u$, либо это вхождение u в t не является вхождением в какую-либо подцепочку вида $s_1 * s_2 * \ldots * s_k$, где $k>1$, для $\forall\ i=1,\ldots,k\ s_i \in St$, и сущеспвует такое m, $1 \leq m \leq k$, что $u = s_m$.

Пример 8. С функцией «Друзья» можно связать тип t1={(интс*дин.физ.об, {интс * дин.физ.об})}. Как первое, так и второе вхождения в цепочку t1символа дин.физ.об. не являются свободными вхождениями. С функцией “Вес множества физических объектов” можно ассоциировать тип t2={({физ.об}, (цел, кг))}; вхождения символа физ.об в t2 и в t3 = ↑физ.об (возможный тип понятия “физический объект”) являются свободными.

Определение 8. Пусть S – с.с. вида (2.5.1), тогда $Tc(S)=\{t \in Tp(S) \setminus (Spectp \cup Toptp) \mid t$ начинается с символа $\uparrow\}$, где $Spectp = \{[\uparrow сущн], [\uparrow пон]\}, [\uparrow об]\}$, $Toptp = \{[сущн], [пон]\}, [об]\}$; $Tob=Tp(S)\setminus(Spectp \cup Toptp \cup Tc(S))$.

Элементы $Tc(S)$ интерпретируются как типы понятий (кроме наиболее общего типа $[сущн]$). Элементы $Tob(S)$ интерпретируются как типы объектов (сущности, не рассматриваемые как понятия).

Определение 9. Пусть S – с.с. вида (2.5.1). Тогда преобразования $tr_1,\ldots,tr_6$, частично применимые к элементам из *Tp(S)*, задаются следующим образом:

1. Если $t \in Tp(S)$, t включает символ *[сущн]*, то tr_1 и tr_2 применимы к t. Пусть *w1* – результат замены в t произвольного вхождения символа *[сущн]* на символ *[пон]* ; *w2* – результат замены в t произвольного вхождения символа *[сущн]* на символ *[об]* . Тогда *w1* и *w2* – возможные результаты применения к цепочке t преобразований tr_1 и tr_2, соответственно.
2. Если $t \in Tp(S)$, t включает символ *[пон]*, $u \in Tc(S)$, то tr_3 применимо к t , и результат замены произвольного вхождения символа *[пон]* на u является возможным результатом применения преобразования tr_3 к t.
3. Если $t \in Tp(S)$, t включает символ *[об]*, $z \in Tob(S)$, то tr_4 применимо к t, и результат замены в t произвольного вхождения символа *[об]* на тип z является возможным результатом применения преобразования tr_4 к типу t.
4. Если $t \in Tp(S)$, t включает символ $s \in St$, $u \in St$, $(s,u) \in Gen$, то тип, получающийся из t заменой какого-либо свободного вхождения s на сорт u, является возможным результатом применения преобразования tr_5 к типу t.
5. Если $t \in Tp(S)$, $u \in St$, z – элементарный составной тип из *Tp(S)*, являющийся уточнением сорта u, w получается из t заменой произвольного свободного вхождения сорта u в цепочку t на цепочку z, то w – возможный результат применения преобразования tr_6 к цепочке t.

Пример 9. Если S_0 – построенная ранее с.с., *t1=[об]*, *t2=простр.об*, *w1=интс*дин.физ.об*, *w2= дин.физ.об* то *w1* и *w2* – возможные результаты применения преобразования tr_4 и tr_5 к *t1* и *t2*, соответственно. Если *t3={физ.об}*, *w3={интс*дин.физ.об}*, то *w3* – возможный результат применения tr_6 к *t3*.

Определение 10. Пусть S – с.с. вида (2.5.1), $t,u \in Tp(S)$. Тогда тип u называется *конкретизацией* типа t, а тип t называется *обобщением* типа u (обозначается через $t|\!-u$) $\Leftrightarrow$ либо $t=u$, либо найдутся такие $x_1,\ldots,x_n \in Tp(S)$, где $n>1$, что $x_1=t$, $x_n=u$, и для $i=1,\ldots,n-1$ найдется такое $k[I] \in \{1,\ldots,6\}$, что преобразование $tr_{k[I]}$ применимо к x_i , и x_{i+1} является возможным результатом применения преобразования $tr_{k[I]}$ к x_i.

Пример 10. Для с.с. S_0 легко проверить, что *[сущн] |–[пон]*, *[сущн] |–[об]*, *[об] |–интс*, *интс|–интс*физ.об*, *физ.об |–дин.физ.об*, *[об] |–{интс}*,

*{интс}|–{интс*дин.физ.об.}, [об] |–(вещ, вещ), [об] |–{(вещ, вещ)}, [пон] |–↑интс, [пон] |–↑интс*дин.физ.об., [пон] |–↑{интс*дин.физ.об.}*

Утверждение 2.1. Пусть *S* - произвольная сортовая система. Тогда отношение конкретизации |– на множестве типов *Tp(S)* является частичным порядком.

Доказательство. Рефлексивность и транзитивность отношения |– следуют непосредственно из определения. Антисимметричность вытекает из свойств преобразований tr_1 ,..., tr_6 . В результате применения преобразования tr_1 или tr_2 количество вхождений символа *[сущн]* уменьшается на 1. После применения преобразования tr_3 или tr_4 на 1 уменьшается количество вхождений символа *[пон]* или символа *[об]*, соответственно. Если $t1,t2 \in Tp(S)$ и тип *t2* получен из *t1* в результате однократного применения преобразования tr_5 , то это означает, что найдутся такие $s, u \in St$, что $s \neq u$, $(s,u) \in Gen$, *t1* включает символ *s*, и *t2* получается заменой некоторого вхождения символа *s* в *t1* на символ *u* . Из антисимметричности отношения *Gen* на *St* следует, что обратное преобразование *t2* в *t1* невозможно. Если тип *t2* получен из типа *t1* однократным применением преобразования tr_6 , то количество символов в *t2* больше, чем количество символов в *t1*.

2.7. Концептуально-объектные системы

Предположим, что для описания какой-то предметной области (ПО) мы выбрали некоторую с.с. *S* вида (2.5.1). Тогда на следующем шаге выберем некоторое множество *X*, состоящее из таких элементарных информационных единиц, с помощью которых мы будем описывать сообщения, команды и вопросы, относящиеся к рассматриваемой ПО; это множество *X* будем называть *первичным информационным универсумом*. Затем выберем *V* – некоторое счетное множество символов, называемых *переменными* и используемых в качестве меток разнообразных сущностей, в том числе в качестве меток СП текстов и фрагментов СП текстов.

Далее зададим отображение $tp: X \cup V \to Tp(S)$ из объединения $X \cup V$ в множество типов, порождаемых с.с. S, тогда каждая переменная и каждая сущность получат тип. На последнем шаге выделим некоторое подмножество *F* множества

X так, что элементы F будут являться обозначениями функций, рассматриваемых в данной ПО. Тогда набор (X,V, tp,F) будет являться концептуально-объектной системой, согласованной с с.с. S.

Определение 1. Пусть S – с.с. вида (2.5.1). Тогда произвольную упорядоченную четверку Ct вида

$$(X, V, tp, F) \qquad (2.7.1)$$

назовем *концептуально-объектной системой (к.о.с.),* согласованной с с.с. S (или к.о.с. для S) ⇔ когда выполняются следующие условия:

(1) X, V – счетные непересекающиеся множества символов; tp - отображение вида $X \cup V \to Tp(S)$;

(2) (2) $F \subset X$, для каждого $r \in F$ цепочка $tp(r)$ начинается с подцепочки ‘{(‘ и заканчивается подцепочкой ‘)}’;

(3) (3) $St \subset X$, и для любого $s \in St$ $tp(s) = \uparrow s$;

(4) (4) $\{ v \in V \mid tp(v) = [сущн] \}$ – счетно.

Множество X называется *первичным информационным универсумом*, элементы множеств V и F называются, соответственно, *переменными* и *функциональными символами*. Если элемент $d \in X \cup V$, $tp(d)=t$, то будем говорить, что t – тип элемента d.

Пример. Построим некоторую к.о.с. Ct_0 для с.с. S_0. Пусть N – множество всех цепочек из цифр ‘0’, ‘1’, … , ‘9’, таких, что если первый символ цепочки 0, то и вся цепочка – 0. Будем полагать, что символы

чел, химик, биолог, студ.гр, тур.гр, П.Сомов, А.Зубов, И.Семенов, Друзья, Колич, Меньше, Знает, Явл1, Сейчас, Раньше, Включить1, Элем

являются соответственно обозначениями понятий “человек“, “химик”, “биолог”, “студенческая группа”, “туристическая группа”, трех конкретных людей, функции “Друзья”, функции “Количество элементов множества”, отношения “Меньше” на множестве вещественных чисел, отношения “В памяти некоторой интеллектуальной системы X1 в момент времени X2 имеется концептуальное представление некоторого сообщения X3”, отношения “Некоторый объект X1 характеризуется понятием X2” (пример реализации в тексте: “П.Сомов является химиком”), текущего момента времени, отношения “Раньше” на множестве моментов времени, отношения “Некоторая интеллектуальная система X1

включает сущность X2 в момент X3 в состав множества сущностей X4", отношения "Элемент множества", отношения "Подмножество". Символ *понятие* будем интерпретировать как информационную единицу, соответствующую словам "понятие" и "концепт".

Пусть *U1={чел, химик, биолог, студ.гр, тур.гр, П.Сомов, А.Зубов, И.Семенов, Друзья, Колич, Меньше, Знает, Явл1, Сейчас, Раньше, Включить1, Элем, понятие}*.

Зададим отображение *t1* из *U1* в $Tp(S_0)$ следующей таблицей:

x	*t1(x)*
чел, химик, биолог	*↑интс*дин.физ.об*
студ.гр, тур.гр	*↑{интс*дин.физ.об}*
П.Сомов, А.Зубов, И.Семенов	*интс*дин.физ.об*
Друзья	*{ (интс*дин.физ.об, {интс*дин.физ.об}) }*
Колич	*{ ({[сущн]}, нат) }*
Меньше	*{ (вещ, вещ) }*
Знает	*{ (интс, мом, сообщ) }*
Явл1	*{ ([об], [пон]) }*
Сейчас	*мом*
Раньше	*{ (мом, мом) }*
Включить1	*{ (интс, [сущн], мом, {[сущн]}) }*
Элем	*{ ([сущн], {[сущн]}) }*
понятие	*[↑пон]*

Табл. 2.1. Примеры соответствий между сущностями и типами

Будем полагать, что *АО_"Салют"*, *АО_"Старт"*, *НПО_"Радуга"* являются обозначениями организаций, *Поставщики*, *Персонал*, *Директор* – обозначения функций "Множество всех поставщиков данной организации", "Множество всех сотрудников данной организации" и "Директор данной организации",

соответственно. Пусть $U2$ = { *АО_"Салют", АО_"Старт", НПО_"Радуга", Поставщики, Персонал, Директор*}, и отображение $t2$ из $U2$ в $Tp(S_0)$ задается следующими условиями:

$t2$(*АО_"Салют"*) = $t2$(*АО_"Старт"*)= $t2$(*НПО_"Радуга"*)

= *орг *простр.об*интс*,

$t2$(*Поставщики*) = {(*орг*, {*орг*})}, $t2$(*Персонал*) = {(*орг*, {*интс*дин.физ.об*})},

$t2$(*Директор*) = {(*орг, интс*дин.физ.об*)} .

Пусть Vx={*x1, x2, …*}, Ve = {*e1, e2, …*} , Vp={*P1, P2, …*} ,

$Vset$ = {*S1, S2, …*} , $V_0 = Vx \cup Ve \cup Vp \cup Vset$, $X_0 = St_0 \cup N \cup U1 \cup U2$,

и отображение $tp_0 : X_0 \cup V_0 \to Tp(S_0)$ задается следующими соотношениями:

$d \in St_0$ => $tp_0(d) = \uparrow d$; $d \in Nat$ => $tp_0(d)$=*нат*; $d \in U1$ => $tp_0(d) = t1(d)$;

$d \in U2$ => $tp_0(d) = t2(d)$; $d \in Vx$ => $tp_0(d)$=[*сущн*]; $d \in Ve$ => $tp_0(d)$=*сит*;

$d \in Vp$ => $tp_0(d)$=*сообщ*, $d \in Vset$ => $tp_0(d)$= {[*сущн*]}.

Пусть F_0={*Друзья, Колич, Поставщики, Персонал, Директор*},

$Ct_0 = (X_0, V_0, tp_0, F_0)$, тогда нетрудно проверить, что Ct_0 – к.о.с. для с.с. S_0.

2.8. Системы кванторов и логических связок. Концептуальные базисы

Предположим, что мы определили с.с. S вида (2.5.1) и к. о. с. Ct вида (2.7.1) для описания рассматриваемой предметной области. Тогда предлагается выделить в первичном информационном универсуме X два непересекающихся и конечных (следовательно, непустых) подмножества Int_1 и Int_2 следующим образом: выделим в St два сорта int_1 и int_2 и предположим, что для $m=1,2$ $Int_m = \{x \in X | tp(x) = int_m\}$. Элементы Int_1 соответствуют значениям выражений "каждый", "какой-то", "некоторый", "произвольный" и т. д. в случаях, когда эти выражения являются частями групп слов, и эти группы связаны с единственным числом. Элементы Int_2 интерпретируются как семантические единицы, соответствующие выражениям "все", "несколько", "почти все", "многие" и т. д. ; минимальное требование к Int_2 заключается в том, чтобы Int_2 содержало семантическую единицу, соответствующую слову "все". Пусть Int_1 содержит выделенный элемент *ref*, рассматриваемый как аналог слова "некоторый" в

смысле “какой-то вполне определенный” (но, возможно, неизвестный). Если *Ct* — к. о. с. вида (2), $d \in X$, *d* обозначает понятие, и семантическое представление (СП) текста включает подцепочку вида *ref d* (например, цепочку *нек человек*, где *ref = нек, d = человек*), тогда будем полагать, что эта подцепочка обозначает некоторую конкретную сущность (но не произвольную), которая характеризуется понятием *d*.

Кроме того, будем предполагать, что X содержит элементы ‘≡’, ‘¬’, ‘∧’, ‘∨’, понимаемые как связки “тождественно”, “не”, “и”, “или”, и элементы ∀ и ∃, понимаемые как квантор всеобщности и квантор существования. Наконец, будем считать, что множество *St* включает выделенные сорта *eqv, neg, binlog, ext* , интерпретируемые, соответственно, как типы (а) связки ‘≡’ , (б) связки отрицания ‘¬ё’ (в) связок ‘∧’, ‘∨’ (конъюнкция и дизъюнкция) , (г) квантора всеобщности и квантора существования.

Эти предположения в наглядной форме отражает рисунок 2.1. На этом рисунке *[сущн]* – это тип “сущность”; элементы *интс, дин.физ.об, нат, сит, сообщ* – сорта “интеллектуальная система”, “динамический физический объект”, “натуральное число”, “ситуация”, “смысл сообщения”; *чел, нек, произвол, определ, все, нескол* – информационные единицы, соответствующие словам “человек”, “некоторый” (“некоторая”, “некоторое”), “произвольный” (“любой”), “определенный”, “все”, “несколько”; *Колич* – обозначение функции “Количество элементов множества”. Элемент *нек* интерпретируется как квантор референтности.

Определение 1. Пусть *S* — с.с. вида (2.5.1), *Ct* — к. о. с. вида (2.7.1) для *S*, $ref \in X$, различные элементы $int_1, int_2, eqv, neg, binlog, ext$ — некоторые выделенные сорта из $St \setminus \{P\}$, и каждая пара их несравнима для отношения общности *Gen* и несравнима для отношения совместимости *Tol.* Тогда упорядоченная семерка *Ql* вида

$$(int_1, int_2, ref, eqv, neg, binlog, ext) \qquad (2.8.1.)$$

называется *системой кванторов и логических связок (с. к. л. с.) для S и Ct* ⇔когда выполнены следующие условия:

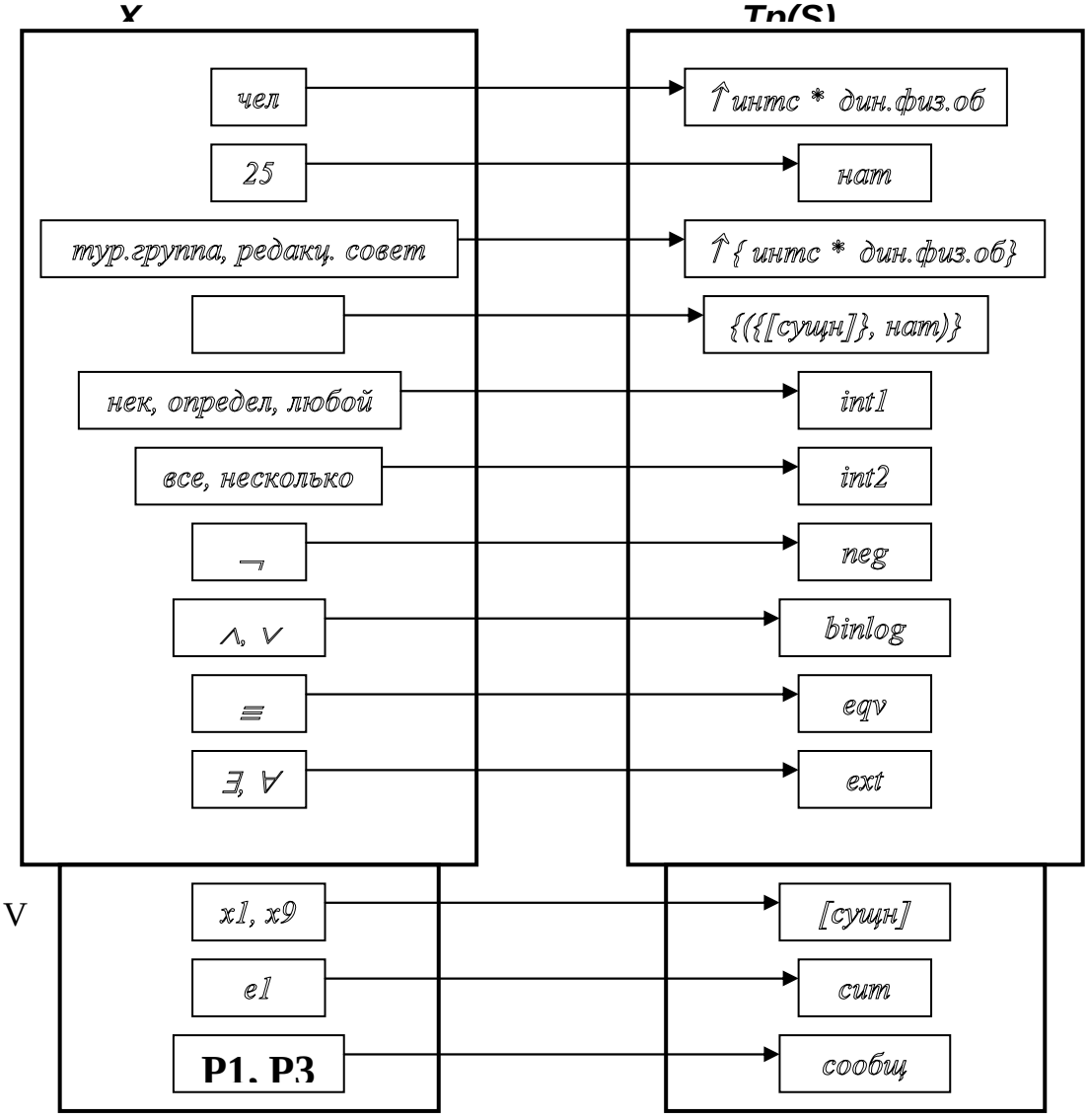


Рис. 2.1. Иллюстрация соответствия между элементами первичного информационного универсума *X*, переменными из множества *V* и их типами из множества *Tp(S)*, где *S* – сортовая система.

1. Для каждого $m = 1,2$ множество $Int_m = \{x \in X | tp(x) = int_m\}$ конечно; $ref \in Int_1$; Int_1 и Int_2 не пересекаются.
2. $\{ \equiv, \neg, \wedge, \vee, \forall, \exists \} \subset X$; кроме того, $tp(\equiv) = eqv$, $tp(\neg) = neg$, $tp(\wedge) = tp(\vee) = binlog$, $tp(\forall) = tp(\exists) = ext$.
3. Не найдется такого $d \in X \setminus (Int_1 \cup Int_2 \cup \{ \equiv, \neg, \wedge, \vee, \forall, \exists \})$ и такого $s \in \{ int_1, int_2, eqv, neg, binlog, ext\}$, что $tp(d)$ и s сравнимы для отношения общности *Gen* или сравнимы для для отношения совместимости *Tol*.
4. Для каждого $u \in \{ int_1, int_2, eqv, neg, binlog, ext \}$ сорт u и сорт P несравнимы для отношения общности Gen и несравнимы для отношения совместимости *Tol*.

Элементы Int_1 и Int_2 называются *интенсиональными кванторами*, *ref* называется *квантором референтности*, $\forall$ и $\exists$ называются *экстенсиональными кванторами*.

Пример 1. Пусть $S_0=(St_0, сообщ, Gen_0, Tol_0)$ – с.с., построенная в параграфе 2.5, $Ct_0=(X_0, V_0, tp_0, F_0)$ – к.о.с. для S_0, построенная в параграфе 2.7.. *Пусть* $Str=\{кв.инт1, кв.инт2, экв, не, бин, экст\}$, $Gen_1=Gen_0 \cup \{(s,s) / s \in Str\}$, $St_1=St_0 \cup Str$, $S_1=(St_1, сообщ, Gen_1, Tol_0)$. Тогда, очевидно, S_1 - сортовая система.

Определим теперь некоторую концептуально-объектную систему Ct_1 и некоторую систему кванторов и логических связок Ql_1. Пусть $Z=\{нек, все, \equiv, \neg, \wedge, \vee, \forall, \exists\}$, $X_1=X_0 \cup Str \cup Z$. Зададим отображение tp_1 из $X_1 \cup V_0$ в $Tp(S_1)$ следующим образом:

$u \in Str \Rightarrow tp_1(u) = \uparrow u$; $d \in X_0 \Rightarrow tp_1(u)=tp_0(u)$;

$tp1(нек)=кв.инт1$, $tp1(все)=кв.инт2$, $tp1(\equiv)=экв$, $tp1(\neg)=не$,

$tp1(\wedge) = tp1(\vee) = бин$, $tp1(\forall)=tp1(\exists)=экст$.

Пусть $Ct_1=(X_1, V_0, tp_1, F_0)$, $Ql_1= (кв.инт1, кв.инт2, нек, экв, не, бин, экст)$. Тогда легко проверить, что Ct_1 – к.о.с. для S_1, Ql_1 – с.к.л.с. для $S1$. В системе Ql_1 информационная единица *нек* интерпретируется как квантор референтности *ref* (т.е. как обозначение значения слов “некоторый”, “некоторая”, ‘некоторое”).

Определение 2. Упорядоченная тройка B вида

$(\ S, Ct, Ql\ )$ (2.8.2)

называется *концептуальным базисом (к.б.)* $\Leftrightarrow S$ – с.с., Ct – к.о.с. вида (2.7.1) для S, Ql – система кванторов и логических связок (с.к.л.с.) для S и Ct, и $(X \cup N) \cap \{$‘,’, ‘(‘, ‘)’, ‘:’, ‘*’, ‘<’, ‘>’, ‘&’$\} = \varnothing$.

Обозначим через *S(B), Ct(B), Ql(B)* компоненты произвольного к.б. вида (2.8.2). Каждый компонент *h* систем видов (2.5.1), (2.7.1), (2.8.1) будет обозначаться через *h(B)*. Например, сорт «смысл сообщения» будет обозначаться через *P(B)*, первичный информационный универсум *X* через *X(B)*, множество переменных *V* через *V(B)*, множество функциональных символов *F* через *F(B)*.

Каждый концептуальный базис будет интерпретироваться как формальное перечисление:

(а) первичных единиц, необходимых для построения семантических представлений (СП) ЕЯ-текстов, для описания знаний о реальности и для представления целей интеллектуальных систем; (б) информации, связанной с этими единицами и необходимой для построения СП текстов, формирования фрагментов знаний и представления целей интеллектуальных систем.

Пример 2. Пусть St_1, Ct_1, Ql_1 – соответственно с.с., к.о.с., с.к.л.с., определенные выше. Тогда очевидно, что упорядоченная тройка $B_1 = (\ St_1, Ct_1, Ql_1\ )$ является концептуальным базисом., и $St(B_1)=St_1$, $P(B_1)=$ *сообщ*, $X(B_1)=X_1$, $V(B_1)=V_0$.

Из этого примера следует, что множество всех концептуальных базисов не является пустым, поскольку мы построили формальный объект B_1, являющийся концептуальным базисом.

Введенные понятия дадут возможность в главе 3 для каждого к.б. *B* задать множество формул *Ls(B)*, удобных для описания содержания (т.е. структурированных значений) ЕЯ-текстов, представления знаний о мире и целей интеллектуальных систем.. Множество формул *Ls(B)* будет названо стандартным К-языком (концептуальным языком), порождаемым базисом *B*.

2.9. Обсуждение разработанной математической модели для описания системы первичных единиц концептуального уровня, используемых лингвистическим процессором

2.9.1. Особенности модели с математической точки зрения

По своей форме разработанная математическая модель для описания системы первичных единиц концептуального уровня, используемых лингвистическим процессором (ЛП), является оригинальной. Рассмотрим отличительные черты построенной модели, представляющиеся наиболее важными как с математической точки зрения, так и с точки зрения использования модели при проектировании ЛП.

1. Конструктивно учитывается существование иерархии понятий: для этого на множестве сортов St задается частичный порядок Gen, называемый отношением общности.
2. Многие сущности, рассматриваемые в той или иной предметной области, могут быть охарактеризованы с разных точек зрения. Например, люди являются, с одной стороны, интеллектуальными системами (поскольку могут читать, решать задачи и т.д.), но, с другой стороны, являются физическими объектами, способными перемещаться в пространстве. Поэтому многие понятия как бы имеют “координаты” по разным “семантическим осям”. Для учета этого важного явления в модель вводится бинарное отношение Tol (отношение совместимости, или толерантности) на множестве сортов. Накопленный опыт показал, что эта оригинальная черта модели является чрезвычайно важной для разработки алгоритмов семантико-синтаксического анализа текстов: дело в том, что появление одного и того же слова в несхожих контекстах может объясняться реализацией в этих контекстах различных “семантических координат” данного слова.
3. Фраза “это понятие используется в физике и химии” (относящаяся, например, к понятию “молекула”) для человека, владеющего русским языком, является очень простой. Между тем, смысловая структура данной

фразы не может быть адекватно отображена средствами основных известных подходов к формализации семантики ЕЯ. Причина заключается в том, что такие подходы не предлагают формального аналога информационной единицы, соответствующей слову “понятие”. Модель, построенная выше, во-первых, рассматривает специальный базовый тип [^пон], интерпретируемый как тип информационной единицы, соответствующей слову “понятие”. Во-вторых, компонент концептуально-объектной системы *Ct* вида (1.2), обозначаемый через X и называемый первичным информационным универсумом, может включать символ, интерпретируемый как информационная единица, соответствующая слову “понятие (см. пример в параграфе 1.7). Данная черта модели важна для разработки ЛП, обрабатывающих научные и научно-технические тексты, а также ЛП прикладных интеллектуальных систем, извлекающих знания из энциклопедических словарей или пополняющих электронные энциклопедических словари.

4. Одной из наиболее важных отличительных черт построенной модели является оригинальное определение множества типов, порождаемого произвольной сортовой системой, где типы рассматриваются как формальные характеристики сущностей, относящихся к выбранной предметной области. В соответствии с этим определением, (а) типы объектов из предметной области по своей форме отличаются от типов понятий, квалифицирующих данные объекты, (б) типы объектов по своей форме отличаются от типов множеств, состоящих из таких объектов, (в) типы понятий, обозначающих объекты, по своей форме отличаются от типов понятий, квалифицирующих множества данных объектов (например, тип понятия “человек” отличается от типа понятий “ученый совет”, “студенческая группа”) и т.д.
5. Модель связывает типы и с именами функций. При этом определение множества типов позволяет разумным образом связать типы с целым рядом довольно нестандартных, но практически важных функций. В частности, к ним относятся функции, значениями которых являются: (а) множество понятий, поясняемых в энциклопедическом словаре, (б)

множество понятий, входящих в определение данного понятия в данном словаре, (в) множество известных определений данного понятия, (г) количество элементов данного множества, (д) множество поставщиков данного предприятия, (е) множество сотрудников данной организации.

2.9.2. Сравнение модели с другими подходами к описанию первичных единиц концептуального уровня

Сравним построенную модель с подходами к описанию первичных единиц концептуального уровня, предлагаемыми логикой предикатов первого порядка, теорией представления дискурсов, теорией обобщенных кванторов, теорией концептуальных графов и эпизодической логикой.

В стандартной логике предикатов рассматриваются неструктурированные множества констант, функциональных символов и предикатных символов. В многосортных логиках предикатов множество констант разбито на непересекающиеся классы, каждый из которых характеризуется некоторым сортом. Разработанная выше модель предоставляет, в частности, следующие дополнительные возможности по сравнению с многосортными логиками предикатов: (1) благодаря введению отношения совместимости в качестве компонента сортовой системы, с первичной единицей концептуального уровня можно связать не только один, но и, во многих случаях, несколько сортов, как бы являющихся “координатами по ортогональным семантическим осям” сущностей, квалифицируемых или обозначаемых такими единицами; (2) “привязывание” типов к первичным информационным единицам означает, что множество таких единиц обладает развитой структурой; в частности, типы позволяют формально различать (а) типы объектов из предметной области и типы понятий, квалифицирующих данные объекты, (б) типы объектов и типы множеств, состоящих из таких объектов, (в) типы понятий, обозначающих объекты, и типы понятий, квалифицирующих множества данных объектов, (3) рассмотрение единиц концептуального уровня, соответствующих словам “некоторый”, “определенный”, “какой-нибудь”, “все”, “большинство”, “несколько”.

Кроме того, рамки построенной модели позволяют рассматривать функции, аргументами и.или значениями которых могут быть семантические представления (СП) высказываний и повествовательных текстов. Например, такая функция может ставить в соответствие каждому понятию, определяемому в энциклопедическом словаре, формулу – СП определения данного понятия. Между тем, в логике предикатов первого порядка аргументами и значениями функций могут быть только термы, но не формулы. Аналогичные ограничения должны выполняться и для атрибутов отношений, т.е. для аргументов предикатов.

Теорию представления дискурсов (ТПД) можно рассматривать как один из вариантов логики предикатов первого порядка, сочетающий использование формул и двумерных диаграмм для более наглядного представления информации. Поэтому перечисленные преимущества разработанной модели для описания системы первичных единиц концептуального уровня, используемых ЛП, относятся и к ТПД.

В теории обобщенных кванторов (ТОК), как и в построенной модели, рассматриваются единицы концептуального уровня, соответствующие словам “некоторый”, “определенный”, “все”, “большинство”, “несколько”. Однако все остальные перечисленные преимущества модели по сравнению с логикой предикатов первого порядка являются одновременно и преимуществами по сравнению с подходом ТОК.

В отличие от логики предикатов первого порядка, нотация теории концептуальных графов (ТКГ) позволяет различать обозначения конкретных объектов (конкретных компьютеров, предприятий, городов и т.д.) и обозначения понятий, квалифицирующих эти объекты (“компьютер”, “предприятие”, ‘город’). Остальные же перечисленные выше свойства модели являются преимуществами и по сравнению с ТКГ.

Наконец, указанные преимущества по сравнению с логикой предикатов первого порядка являются одновременно и преимуществами по сравнению с подходом к структурированию совокупности первичных единиц концептуального уровня, предлагаемым эпизодической логикой.

Очевидно, что если какая-либо модель предназначена как для описания системы первичных единиц концептуального уровня, используемых ЛП, так и для представления информации, связанной с этими единицами и определяющей возможности их соединения в правильные составные структуры, то подобная модель предлагает некоторый способ концептуальной структуризации рассматриваемых предметных областей. Поэтому на основании проведенного анализа можно придти к заключению, что построенная выше модель предлагает более “тонкоячеистую” структуризацию предметных областей по сравнению с основными известными подходами к формализации семантики ЕЯ, значительно увеличивает “разрешающую способность” формального инструментария, предназначенного для исследования различных предметных областей.

В конце 1990-х – начале 2000-х годов большую актуальность приобрели исследования по разработке онтологий различных предметных областей, т.е. по созданию формальных описаний систем понятий, относящихся к выбранной области, вместе с их определениями и фрагментами знаний, “привязанных” к понятиям. Первым шагом в каждом проекте такого рода является выбор некоторой начальной (другими словами, базовой) структуризации рассматриваемой предметной области или группы областей.

Представляется, что построенная в данной главе математическая модель для описания системы первичных единиц концептуального уровня, используемых ЛП, и для представления информации, связанной с этими единицами, может найти применение в проектах разработки более совершенных онтологий в произвольных предметных областях, поскольку разработанная модель предлагает формальный инструментарий с наибольшей “разрешающей способностью” по сравнению с другими известными подходами к формализации семантики ЕЯ и, как следствие, к концептуальной структуризации предметных областей.

Глава 3

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ОПИСАНИЯ СТРУКТУРИРОВАННЫХ ЗНАЧЕНИЙ ПРЕДЛОЖЕНИЙ И СВЯЗНЫХ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

3.1. Постановка задачи

В параграфе 2.1. была обоснована актуальность разработки широко применимых формальных языков для построения семантических представлений (СП) ЕЯ-текстов, выразительные возможности которых позволяют отображать многие особенности поверхностной структуры предложений и связных ЕЯ-текстов. В качестве первого шага на пути разработки определения такого класса формальных языков в главе 2 был определен класс формальных объектов, называемых концептуальными базисами.

В данной главе ставится задача разработки математической модели для описания структурированных значений предложений и связных естественно-языковых текстов.

По своей форме модель должна являться описанием такого соответствия между произвольным концептуальным базисом *B* и некоторым множеством формул *Forms(B)*, чтобы класс формальных языков {*Forms(B)*, где *B* – к.б.} был удобен для построения СП фраз и связных ЕЯ-текстов, отражающих многие особенности поверхностной структуры текстов.

С целью выработки критериев для построения такой модели был проведен системный анализ структурных особенностей (а) текстов на русском, английском, немецком и французском языках, (б) ряда искусственных языков, используемых для построения семантических представлений текстов лингвистическими процессорами, (в) выражений искусственных языков представления знаний в прикладных интеллектуальных системах (в частности, терминологических языков представления знаний).

Проведенный анализ показал, что есть несколько важных аспектов формализации семантики ЕЯ, которые до недавнего времени недооценивались или игнорировались большей частью исследователей. В частности, это относится к формальному исследованию смысловых структур (а) повествовательных текстов, включающих описания множеств; б) дискурсов со ссылками на смысл предложений и более крупных частей текста; (в) фраз, где логические связки “и”, “или” используются нетрадиционной способами и соединяют не фрагменты, выражающие высказывания, а описания объектов, множеств, понятий; (г) фраз с придаточными определительными и причастными оборотами; (д) фраз со словами "понятие", "термин".

Кроме того, несколько наиболее популярных подходов к математическому изучению семантики ЕЯ не принимают или недостаточно принимают во внимание роль знаний о мире в понимании ЕЯ и, следовательно, не изучают проблем формального описания фрагментов знаний (определения понятий и т. д.). Например, это относится к весьма популярной теории представления дискурсов (Kamp 1981; Kamp, Reyle 1993, 1996; van Eijck, Kamp 1996).

Надо добавить, что тексты имеют авторов, могут быть опубликованы тем или другим источниками, могут вводиться с того или другого терминалов и т. д. Информация об этих внешних связях текстов может быть важна для их смысловой интерпретации. Поэтому целесообразно рассматривать текст как некий структурированный объект, обладающий поверхностной структурой T, множеством значений S (в большинстве случаев S состоит из одного значения), соответствующим T, и некоторыми значениями V_1 ,..., V_N, обозначающими автора (авторов) T, дату написания (или коррекции) T, указывающими новую информацию в T и т. д. Но наиболее популярные подходы к математическому исследованию ЕЯ не предусматривают формальных средств для представления текстов как структурированных объектов подобного рода.

На основании проведенного системного исследования поставим задачу построения такой модели, чтобы ее формальные средства позволяли нам следующее:

(Свойство 1): Строить обозначения структурированных значений (СЗ) как фраз, выражающих высказывания, так и повествовательных текстов; такие

обозначения обычно называют семантическими представлениями (СП) ЕЯ-выражений.

(Свойство 2): Строить и различать формальными средствами обозначения СЗ повествовательных текстов, СЗ целей (выраженных неопределенными формами глаголов с зависимыми словами, таких как "окончить с отличием МГУ, подготовить и защитить кандидатскую диссертацию по биохимии") и СЗ вопросов.

(Свойство 3): Строить и различать обозначения единиц, соответствующих (а) объектам, ситуациям, процессам в реальном мире и (б) понятиям, квалифицирующим (характеризующим) эти объекты, ситуации, процессы.

(Свойство 4): Строить и различать обозначения: (3.1) объектов и множеств объектов; (3.2) понятий и множеств понятий; (3.3) СП текстов и множеств СП текстов.

(Свойство 5): Различать формальным образом понятия, квалифицирующие объекты, и понятия, квалифицирующие множества объектов тех же видов.

(Свойство 6): Строить составные обозначения понятий, т. е. строить формулы, отражающие поверхностно-семантическаю структуру ЕЯ-выражений, подобных выражению “человек, окончивший МГУ имени М.В. Ломоносова и являющийся биологом или химиком”.

(Свойство 7): Строить объяснения более общих понятий с помощью менее общих; в частности, строить цепочки вида (a=Des(b)), где a обозначает некоторое понятие, которое необходимо объяснить, a Des(b) обозначает описание некоторой конкретизации известного понятия b.

(Свойство 8): Строить обозначения упорядоченных n-местных наборов различных сущностей, где $n > 1$.

(Свойство 9): Строить (9.1) формальные аналоги составных обозначений множеств ("эта группа, состоящая из 12 туристов, являющихся химиками или биологами" и т.п.), (9.2) обозначения множеств упорядоченных наборов сущностей, (9.3) обозначения множеств, состоящих из множеств, и т.д.

(Свойство 10): Описывать теоретико-множественные отношения и операции над множествами.

(Свойство 11): Строить обозначения СЗ фраз, содержащих, в частности:

(11.1) слова “произвольный”, “некоторый”, “все”, “каждый”, и т. д.;

(11.2) выражения, полученные применением связок “и”, “или” к обозначениям (11.2а) предметов, событий; (11.2б) понятий; (11.2в) множеств;

(11.3) выражения , где связка “не” стоит непосредственно перед обозначением предмета, события и т. д.; (11.4) косвенную речь;(11.5) причастные обороты и придаточные определительные предложения;

(11.6) слова "понятие", "термин".

(Свойство 12): Строить обозначения СЗ дискурсов со ссылками на упомянутые объекты.

(Свойство 13): Указывать явно в СП дискурсов причинно-следственные и временные отношения между описываемыми ситуациями (событиями).

(Свойство 14): Описывать СЗ дискурсов со ссылками на смысл фраз и более крупных фрагментов рассматриваемых текстов.

(Свойство 15): Выражать суждения о тождественности двух сущностей.

(Свойство 16): Строить формальные аналоги формул логики предикатов первого порядка с кванторами существования и/или всеобщности.

(Свойство 17): Рассматривать нетрадиционные функции (и другие нетрадиционные отношения) с аргументами и/или значениями, являющимися: (17.1) множествами предметов, ситуаций (событий); (17.2) множествами понятий; (17.3) множествами СП текстов.

(Свойство 18): Строить концептуальные представления текстов как информационные объекты, отражающие не только смысл, но и значения внешних характеристик текста: авторов, дату, области применения результатов и т. д.

Эта постановка задачи отражена в публикациях (Фомичев 1981а, 1981б, 1983, 1988; 2002б, в; Fomitchov 1984; Fomichov 1992, 1996a, б, 2002b).

3.2. Краткая характеристика предлагаемого решения поставленной задачи

3.2.1. Краткая характеристика новых правил построения формул

В данной главе произвольному концептуальному базису (к.б.) *B* будут поставлены в соответствие три множества формул *Ls = Ls*(*B*), *Ts = Ts*(*B*), Y*s = Ys*(*B*) (*l*-формулы, *t*-формулы, *y*-формулы). Объединение этих множеств будет обозначено через *Forms(B).* Множество *Ls*(*B*) будет названо *стандартным К-языком в к.б. B.* Концептуальный базис *B* оказывается возможным определить таким образом, что цепочки языка *Ls = Ls*(*B*) будет удобно использовать для описания структурированных значений (другими словами, смысловых структур) ЕЯ-текстов, представления знаний о мире и представления целей интеллектуальных систем.. Другими словами, цепочки из языка *Ls = Ls*(*B*) окажется удобным использовать для построения семантических представлений (СП) текстов на естественном языке. Формулы из первого класса, т.е. *l*-формулы, будут называться также К-цепочками.

Каждая формула из множества *Ts*(*B*) представима в виде *d* & *t*, где $d \in Ls(B)$, *t* – тип из *Tps*(*B*). Формулы из множества *Ys*(*B*) являются выражениями вида a_1 & …& a_n & *d*, где $a_1, \ldots, a_n$, $d \in Ls(B)$, *n* имеет разные значения для разных *d*, и цепочка *d* строится из $a_1, \ldots, a_n$ как из элементарных информационных единиц (некоторые из них могут быть немного преобразованы) однократным применением некоторого правила построения.

В данной работе предлагается оригинальная схема подхода к определению трех класов выводимых формул; эта схема заключается в следующем. Будут сформулированы некоторые высказывания *P[0],...,P[10];* они будут интерпретироваться как правила построения семантических представлений (СП) ЕЯ-текстов из элементов первичного информационного универсума *X(B*), переменных из *V(B)* и нескольких специальных символов при условии, что *B* является концептуальным базисом для рассматриваемой области.

Каждое из этих правил фактически задает некоторую операцию на множестве всевозможных наборов, компоненты которых являются СП простых или

составных выражений естественного языка (ЕЯ). Всего 10 операций достаточно для построения формул, отображающих смысл (или структурированные значения) сколь угодно сложных ЕЯ-текстов. Поэтому можно сказать, что система этих правил задает некоторую полную систему квазилингвистических концептуальных операций.

Классы формул *Ls*, *Ts*, *Ys* для произвольного к.б. *B* определяются совместной индукцией правилами *P[0]* , *P[1]*, …, *P[10]* . Для любого к.б. *B* правило *P[0]* задает начальный запас формул.

Определение 1. Обозначим через *P[0]* высказывание "Если $d \in X(B) \cup V(B)$, $t \in Tp(S(B))$, $tp = tp(B)$, $tp(d) = t$, то $d \in L(B)$, и цепочка вида d & t входит в $T^0(B)$". □

Пусть *B* — произвольный к.б., *L(B)* и $T^0(B)$ — наименьшие множества, задаваемые утверждением *P[0]*, $Lnr_0(B) = L(B)$ (обозначение *"Lnr"* расшифровывается как *"L* нумерованное"). Тогда, очевидно, $Lnr_0(B) = X(B) \cup V(B)$, $T^0(B) = \{ b \mid b = d \,\&\, t, d \in X(B) \cup V(B), t \in Tp(S(B)), t = tp(d)\}$.

Таким образом, в соответствии с правилом P[0] информация о типах элементов первичного информационного универсума X(B) и переменных из V(B) отображается в структуре формул из множества T^0(B). □

Пример 1.. Пусть S_1 – с.с., построенная в примере из параграфа 2.8**,** $B_1 = (S_1, Ct_1, Ql_1)$ – к.б., определенный в параграфе 2.8**,** $B=B_1$. Тогда легко увидеть, что выполняются следующие соотношения:

$$чел,\ П.Сомов,\ НПО\_"Радуга",\ Друзья \in Lnr_0(B),$$

$$Персонал,\ Поставщики \in Lnr_0(B);\ чел\ \&\ \uparrow интс * дин.физ.об \in T^0(B),$$

$$П.Сомов\ \&\ интс * дин.физ.об \in T^0(B);$$

$$НПО\_"Радуга"\ \&\ орг * простр.об * интс \in T^0(B);$$

$$Друзья\ \&\ \{(интс * дин.физ.об, \{интс * дин.физ.об\})\} \in T^0(B),$$

$$Персонал\ \&\ \{(орг, \{интс * дин.физ.об\})\ \})\} \in T^0(B),$$

$$Поставщики\ \&\ \{(орг, \{орг\})\} \in T^0(B).$$

Правило *P[1]* предназначено для присоединения информационных единиц, соответствующих словам "некоторый", "каждый" , "какой-нибудь", "все", "несколько", "большинство" (такие информационные единицы в данной работе называются интенсиональными кванторами) к простым или составным

обозначениям понятий. Поэтому правило *P[1]* позволяет строить формальные аналоги выражений: "некоторый человек", "все люди", "большинство людей", "некоторый человек ростом 175 см", "все тридцатилетние люди", “все города Европы”. Примерами *l*-формул (К-цепочек) для *P[1]*, как последнего примененного правила, являются цепочки

*нек чел, все чел *(Возраст, 30/год), все город * (Регион, Европа)* .

Правило *P[2]* предназначено для построения цепочек вида $f(a_1, \ldots, a_n)$, где f – обозначение функции, $n \geq 1$, $a_1, \ldots, a_n$ – *l*-формулы, построенные с применением каких-то правил из списка *P[0]* , *P[1]*, …, *P[10]*. Например, после применения правила на последнем шаге вывода можно получить цепочки
Города(Европа), Колич-элем(Города(Европа)).

Правило *P[3]* позволяет строить цепочки вида $(a_1 \equiv a_2)$, где a_1, a_2 – *l*-формулы, полученные при помощи любых правил из *P[0]*, …, *P[10]*, и a_1, a_2 обозначают сущности, являющиеся однородными в некотором смысле. Примеры К-цепочек для *P[3]* как последнего примененного правила:

$(y_1 \equiv$ *нек город * (Название, ‘Саратов’))*,

(Директор(АО_”Салют”) $\equiv$ *П.Сомов)* .

Правило *P[4]* позволяет строить К-цепочки вида $r(a_1, \ldots, a_n)$, где r – *n*-арное отношение, $n \geq 1$, $a_1, \ldots, a_n$ – К-цепочки, полученные при помощи некоторых правил из *P[0]*, …, *P[10]* . Примеры К-цепочек для *P[4]* : *Принадлеж(Намюр, Города(Бельгия)), Подмнож(Города(Бельгия), Города(Европа)).*

Правило *P[5]* предназначено для построения К-цепочек вида $d : v$, где d – К-цепочка, не включающая v, v – переменная, и выполнены некоторые условия. При помощи правила *P[5]* можно помечать переменными в семантических представлениях текстов на естественном языке: а) описания различных сущностей, встречающихся в тексте (физических объектов, событий, понятий и др.), б) семантические представления предложений или более крупных фрагментов текста, на которые имеется ссылка в любой части текста. Примерами К-цепочек для правила *P[5*], примененного на последнем шаге вывода, являются выражения

все чел : Z1, Меньше(Возраст(П.Сомов), 30/ год) : P1.

Это правило дает возможность строить семантические представления текстов таким образом, чтобы они отражали референтную (ссылочную) структуру текстов. Демонстрирующие это утверждение примеры приведены ниже.

Правило *P[6*] позволяет строить К-цепочки вида $\neg d$, где d – К-цепочка, удовлетворяющая ряду условий. Примеры К-цепочек для *P[6]* :

¬биолог, ¬Принадлеж(Бонн, Города(Бельгия)). Здесь ¬ обозначает связку "не".

При помощи правила *P[7]* можно строить К-цепочки вида $(a_1 \wedge \ldots \wedge a_n)$ или $(a_1 \vee \ldots \vee a_n)$, где $n > 1$, $a_1, \ldots, a_n$ – К-цепочки, обозначающие однородные в некотором смысле сущности. В частности, $a_1, \ldots, a_n$ могут быть семантическими представлениями высказываний, описаниями физических объектов, описаниями множеств, состоящих из объектов одной природы, описаниями понятий. Следующие цепочки являются примерами К-цепочек (или *l*-формул) для *P[7]* :

(Финляндия ∨ Норвегия ∨ Швеция),

(Принадлеж((Намюр ∧ Гент), Города(Бельгия)) ∧ ¬Принадлеж(Бонн,

Города((Финляндия ∨ Норвегия ∨ Швеция))))).

Назначение правила *P[8]* состоит в том, что оно позволяет строить, в частности, К-цепочки вида $c * (r_1,b_1), \ldots, (r_n,b_n)$, где c – информационная единица из первичного универсума X, обозначающая понятие, для $i = 1, \ldots, n$, r_i – функция одного аргумента или бинарное отношение, b_i обозначает возможное значение r_i для объектов, характеризующихся понятием c. Например, если выбрать соответствующим образом первичные информационные единицы, то после применения на последнем шаге вывода правила P[8], можно получить К-цепочки *чел * (Имя, 'Петр')(Фамилия, 'Сомов'), поворот * (Направление, левое).*

Правило *P*[9] дает возможность строить, в частности, К-цепочки вида $\forall v(des)D$ и $\exists v(des)D$, где $\forall$ – квантор всеобщности, $\exists$ – квантор существования, *des* обозначает понятие ("человек", "город", "целое число" и др.) или составные понятия ("целое число, большее 200" и др.). *D* можно интерпретировать как семантическое представление высказывания с переменной *v* о любой сущности, характеризуемой понятием *des*. Примеры К-цепочек для *P[9]* как правила, примененного на заключительном шаге построения формулы:

∀x1(нат.ч.) ∃x2(нат.ч.) Меньше(x1, x2),

∃y(страна ∗(Регион, Европа))Больше(Колич(Города(y)), 15).

Правило *P[10]* позволяет строить, в частности, К-цепочки вида $<a_1,\ldots,a_n>$, где $n > 1$, $a_1,\ldots,a_n$ – К-цепочки. Цепочки, получаемые с использованием правила *P[10]* на последнем шаге вывода, интерпретируются как обозначения *n*-мерных векторов. Компонентами такого вектора могут быть не только обозначения чисел, объектов, но и семантические представления выражений, множеств, понятий и др. Используя правила *P[10]* и *P[4]* , можно построить цепочку

Учиться1(<Агент1, нек чел ∗(Имя, 'Петр')><Учеб.заведение, МГУ>,

<Начал. момент, 1996>), где *Агент1, Учеб.заведение, Начал. момент* – обозначения тематических ролей, т.е. обозначения отношений между значением глагола "учиться" и значениями зависящих от него в предложениях групп слов.

3.2.2. Схема определения трех классов формул, порождаемых концептуальными базисами

Рассмотрим более детально предлагаемую оригинальную схему подхода к определению трех класов выводимых формул.

Определение 2. Если *B* - произвольный концептуальный базис, то пусть

(а) $D(B) = X(B) \cup V(B) \cup \{',' , '(', ')', ':' , '*', '<', '>'\}$,

(б) $Ds(B) = D(B) \cup \{' \&'\}$, (в) $D^+(B)$ и $Ds^+(B)$ — множества всех непустых конечных последовательностей элементов из $D(B)$ и $Ds(B)$, соответственно. □

Если $1 \le i \le 10$, то для любого к.б. *B* и для $k = 1,\ldots,i$ утверждения $P[0],\ldots,P[i]$ определяют совместной индукцией некоторые множества формул $Lnr_i(B) \subset D^+(B)$, $T^0(B)$, $Tnr_i^1(B),\ldots$, $Tnr_i^i(B)$, $Ynr_i^1(B),\ldots$, $Ynr_i^i(B\ ) \ ) \subset Ds^+(B)$. Множество $Lnr_i(B)$ рассматривается как главный подкласс формул, порождаемых правилами $P[0],\ldots,P[i]$. Формулы из этого множества предназначены для описания содержания (смысловых структур) ЕЯ-текстов.

Если $1 \le k \le i$, то множество $Tnr_i^k(B)$ состоит из цепочек вида $b\&t$, где $b \in Lnr_i(B)$, $t \in Tp(S(B))$, и *b* понимается как результат применения правила *P[k]* к некоторым более простым формулам на последнем шаге вывода. Надо добавить, что при построении *b* из элементов *X(B)* и *V(B)* могут использоваться любые

правила *P[0],...,P[k],...,P[i]*; эти правила можно применять произвольно много раз. Если к.б. *B* выбран для описания некоторой области, то *b* можно понимать как СП текста или фрагмент СП текста, относящегося к данной области. В этом случае *t* можно рассматривать как описание вида сущностей, характеризуемых этим СП или фрагментом СП. Кроме того, *t* может квалифицировать *b* как СП повествовательного текста. Номер *i* интерпретируется в этих обозначениях как максимальный номер правила из списка P[0], P[1], … , P[10], которое мы используем для того, чтобы определить множества формул**.**

Таким образом, как будет показано ниже, $Lnr_4(B),..., Lnr_{10}(B)$ включа.ют формулы *Элем(П.Сомов, Друзья(И.Семенов)), Элем(АО_"Салют", Поставщики(НПО_"Радуга")),* и $Tnr_4^4(B),..., Tnr_4^{10}(B)$ включают формулы *Элем(П.Сомов, Друзья(И.Семенов)) & сообщ, Элем(АО_"Салют", Поставщики(НПО_"Радуга")) & сообщ,* где *сообщ* — выделенный сорт $P(B_1)$ ("смысл сообщения").

Каждая цепочка $c \in Ynr_i^k(B)$, где $1 \le k \le i$, может быть представлена в виде $c = a_1 \,\&\, a_2 \,\&\, ... \,\&\, a_m \,\&\, b$, где $a_1,...,a_m$, $b \in Lnr_i(B)$. Кроме того, найдется такое $t \in Tp(S(B))$, что цепочка $b \,\&\, t$ принадлежит $Tnr_i^k(B)$. Цепочки $a_1,...,a_m$ строятся с помощью любых правил из списка *P[0],...,P[10]*, а цепочка b построена из "блоков" $a_1,...,a_m$ (некоторые из них могут быть немного изменены) применением только один раз правила *P[k]*. Возможное количество "блоков" $a_1,...,a_m$ зависит от *k*. Таким образом, множество $Ynr_i^k(B)$ фиксирует результат применения правила *P[k]* один раз. Ниже мы увидим, что множества $Ynr_4^4(B_1),...,$ $Ynr_4^{10}(B_1)$ включают формулы

Элем & П.Сомов & Друзья((И.Семенов) & Элем(П.Сомов, Друзья((И.Семенов)),

Элем & АО_"Салют" & Поставщики(НПО_"Радуга") & Элем(АО_"Салют", Поставщики(НПО_"Радуга")).

Пусть для $i = 1,...,10$ $\quad T_i(B) = T^0(B) \cup Tnr_i^1(B) \cup ... \cup Tnr_i^i(B);$

$Y_i(B) = Ynr_i^1(B) \cup ... \cup Ynr_i^i(B); \quad Form_i(B) = Lnr_i(B) \cup T_i(B) \cup Y_i(B)$.

Будем интерпретировать $Form_i(B)$ как множество формул, порождаемых к.б. В. Это множество представляет собой объединение трех классов формул, главным из которых является $Lnr_i(B)$. Формулы из этих трех классов будем называть соответственно *l*-формулами, *t*-формулами, и *y*-формулами. Класс *t*-

формул необходим для того, чтобы связать тип из $Tp(S(B))$ с каждым $b \in Lnr_i(B)$, где i = 1,...,10. Для $i = 0,...,9$, $Lnr_i(B) \subseteq Lnr_{i+1}(B)$. Множество $Lnr_{10}(B)$ называется стандартным концептуальным языком (стандартным К-языком, СК-языком) в стационарном базисе B и обозначается через $Ls(B)$. Поэтому l-формулы будут часто называться К-цепочками.

Множество $T_{10}(B)$ обозначается через $Ts(B)$. Для любого к.б. B и для любой формулы A из множества $Ts(B)$ существуют такой тип $t \in Tp(S(B))$ и такая формула $C \in Ls(B)$, что $A = C \ \& \ t$. Для построения СП текстов будут использоваться только формулы из $Ls(B)$ и $Ts(B)$, т.е. l-формулы и t-формулы. Y-формулы рассматриваются как вспомогательные и нужны для того, чтобы сформулировать некоторые полезные свойства множеств $Ls(B)$ и $Ts(B)$.

3.3. Использование интенсиональных кванторов в формулах

В параграфе 2.8 было введено понятие *интенсионального квантора*. Этот термин используется для обозначения информационных единиц (другими словами, семантических единиц), соответствующих, в частности, словам и выражениям “каждый”, “некоторый”, “произвольный”, “какой-нибудь”, “определенный”, “все”, “несколько”, “большинство”, “почти все”. Совокупность интенсиональных кванторов делится на два подкласса, обозначаемые через Int_1 и Int_2 . Это осуществляется следующим образом. Компонентом каждого концептуального базиса B вида (2.8.2) является система кванторов и логических связок (с.к.л.с.) Ql вида (2.8.1). Компонентами с.к.л.с. Ql вида являются, в частности, два выделенных сорта int_1 и int_2 . Это дает возможность для $m=1,2$ определить Int_m как $\{x \in X | \ tp(x) = int_m\}$, где первичный информационный универсум X является одним из компонентов концептуально-объектной системы $Ct(B)$ вида (2.7.1)

Элементы множества Int_1 соответствуют значениям выражений “каждый”, “какой-то”, “некоторый”, “произвольный” и т. д. в случаях, когда эти выражения являются частями групп слов, и эти группы связаны с единственным числом. Элементы множества Int_2 интерпретируются как семантические единицы, соответствующие выражениям “все”, “несколько”, “почти все”,

“многие” и т. д. ; минимальное требование к Int_2 заключается в том,, чтобы Int_2 содержало семантическую единицу, соответствующую слову “все”.

Правило P[1] позволяет нам присоединять интенсиональные кванторы к простым или составным обозначениям понятий. В результате применения этого правила, во-первых, строятся: (а) *l*-формулы вида *Int.qr Conc.expr* , где *Int.qr* – интенсиональный квантор из *Int(B)*, а *Conc.expr* – простое или составное обозначение понятия. Во-вторых, строятся *t*-формулы вида *Int.qr Conc.expr & t* , где *t* – тип из множества *Tp(S(B))*.

Например, можно выбрать к.б. *B* так, что в этом базисе с помощью правил P[0] и P[1] можно будет построить *l*-формулы

*нек город, нек город * (Назв, “Чита”),*

каждый город, каждый человек(Квалиф., студент),*

все город, все город(Страна, Россия)*

и *t*-формулы *нек. город & простр. об, все город & {простр. об}* ,

каждый человек(Квалиф, студент) & интс * дин.физ.об* .

Определение 1. Если *B* — произвольный к.б., то для *m* = 1, 2

$Int_m(B) = \{q \in X(B) \mid tp(q) = int_m(B)\}$, $Int(B) = Int_1(B) \cup Int_2(B)$,

Tconc(B) = {t∈ Tp(S(B)) | t начинается с символа ‘↑’ }∪ Spectp ,

где Spectp = {[↑сущн], [↑пон], [↑об]}. □

Напомним, что в параграфе 2.6 выражения *[↑сущн], [↑пон], [↑об]* интерпретируются как такие символы (т.е. неделимые единицы), которые являются информационными единицами, соответствующими словам “сущность” , “понятие” (или “концепт”) и “объект”. В данной работе термин “сущность” является наиболее общим. Объектами называются все те сущности, которые не являются понятиями.

Используя правила P[0] и P[1], мы можем строить *l*-формулы вида *q d*, где *q∈Int(B), d∈ X(B), tp(d) ∈ Tconc(B).* Так, рассматривая к.б. B_1, определенный в параграфе 2.8, мы можем построить *l*-формулы

нек чел, нек тур.гр, нек понятие, все чел, все тур.гр, все понятие.

Можно также строить более сложные цепочки вида *q descr*, где *q* — интенсиональный квантор, *descr* – составное обозначение понятия. Для построения цепочки *descr* используются правила P[0], P[1], правило P[8] (см.

параграф 3.6), и, возможно, некоторые другие правила . Например, для к.б. B_1 будет возможно построить *l*-формулы *нек тур.гр * (Колич, 12), все тур.гр * (Колич, 12).* Эти формулы понимаются как семантические представления (СП) выражений “некоторая туристическая группа из 12 человек” и “все тургруппы из 12 человек”.

Переход от *l*-формулы *c*, обозначающей понятие, к *l*-формуле *q c*, где *q* — интенсиональный квантор, описывается с помощью специальной функции h.

Определение 2. Пусть B — произвольный к.б., S = S(B), Tp(S) – множество типов, порождаемых сортовой системой S . Тогда отображение

h: {1,2} × Tp(S) → Tp(S) задается следующим образом:

(а) если u∈ Tp(S) и цепочка ↑u входит в Tp(S), то h(1, ↑u) = u, h(2, u) = {u} ;

(б) h(1, [↑сущн]) = [сущн], h(1, [↑пон]) = [пон], h(1, [↑об]) =[об],

h(2, [↑сущн]) = {[сущн]}, h(2, [↑пон]) = {[пон]}, h(2, [↑об]) = {[об]} . □

С точки зрения построения СП текстов, отображение h описывает преобразование типов как следующие переходы:

(а) от понятия “человек”, “туристическая группа” к СП выражений “некоторый человек”, “каждый человек”, “любой человек”, “некоторая туристическая группа”, “любая туристическая группа” и. т. д. (в случае, когда первый аргумент h равен 1) и к СП выражений “все люди”, “все туристические группы”, и т. д. . (в случае, когда первый аргумент h равен 2); (б) от выражений “сущность”, “понятие”, “объект” к СП выражений “некоторая сущность”, “произвольная сущность”, “некоторое понятие”, “произвольное понятие”, “некоторый объект”, “произвольный объект”и т. д. (если первый аргумент h равен 1) и к СП выражений “все сущности”, “все понятия”, “все объекты” и т. д. (если первый аргумент h равен 2).□

Определение 3. Через P[1] обозначим высказывание

“Пусть c ∈ L(B) \ V(B), u∈Tconc(B), k∈{0, 8}, и цепочка c & u входит в T^k(B). Пусть m∈ {1, 2}, q∈Int_m, t = h(m,u), и b — цепочка вида q c. Тогда b∈L(B), цепочка вида b & t входит в T^1(B), и цепочка вида q&a&b входит в Y^1(B).” □

Пример 1. Пусть B — к.б. B_1, построенный в параграфе 2.8; *L(B), T^0(B), T^1(B), Y^1(B)* — наименьшие множества, совместно определяемые высказываниями P[0] и P[1]. Тогда легко убедиться в справедливости следующих соотношений:

*чел∈ L(B)\V(B), чел & ↑интс * дин.физ.об∈ T^0(B), нек∈ Int_1(B),*

h(1, ↑интс дин.физ.об) = интс * дин.физ.об ⇒*

*нек чел∈L(B), нек чел & интс * дин.физ.об ∈ T^1(B), нек & чел & нек чел∈ Y^1(B);*

все∈ Int_2(B), h(2, ↑интс дин.физ.об) = {интс* дин.физ.об } ⇒*

*все чел∈L(B), все чел & {интс * дин.физ.об }∈ T^1(B), все&чел&всечел∈ Y^1(B);*

тур.гр∈ L(B)\V(B),тур.гр& ↑{интс дин.физ.об }∈ T^0(B),*

h(1, ↑{интс дин.физ.об }) = {интс* дин.физ.об},*

*h(2, ↑{интс * дин.физ.об }) = {{интс* дин.физ.об }} ⇒*

нек тур.гр, все тур.гр∈L(B), нек тур.гр & {интс}, все тур.гр & {{интс}}}∈ T^1(B),

нек & тур.гр& нек тур.гр∈ Y^1(B); все & тур.гр & все тур.гр∈ Y^1(B);

понятие∈ L(B)\V(B), понятие&[↑пон]∈ T^0(B),

h(1,[↑пон]) = [пон], h(2,[↑пон]) = {[пон]} ⇒ нек понятие, все понятие∈L(B),

нек понятие&[пон], все понятие & {[пон]}∈ T^1(B),

нек & понятие & нек понятие ∈ Y^1(B), все & понятие & все понятие∈ Y^1(B).□

Комментарий к правилу P[1]. Фрагмент правила P[1] “Пусть c ∈ L(B) \ V(B), u∈Tconc(B), k∈{0, 8}, и цепочка c & u входит в T^k(B)” означает, что u – тип понятия (т.к. либо начинается с символа ‘↑’, либо является одним из символов [↑сущн], [↑пон], [↑об]), при k=0 c – простое обозначение понятия (c∈X(B)); при k=8 c – составное обозначение понятия. Такие составные обозначения понятий будут строиться с помощью правила P[8]; примерами таких выражений являются выражения человек* (Область.деят., биология), город*(Страна, Россия)).

Правило P[1] будет очень часто использоваться при построении СП текстов, т.к. оно нужно для построения семантических образов выражений с существительными. Например, пусть T1=”Откуда поступил двухтонный алюминиевый контейнер?”. Тогда в результате выполнения первого шага построения СП T1 можно получить выражение

E1 = *нек контейнер1 * (Вес, 2/Тонна>) (Материал, алюминий)* ,

а после выполнения второго шага – заключительное выражение

E2 = *Вопрос (x1, Ситуация (s1, поступление2 *(Место1, x1) (Объект1,*

*нек контейнер1 * (Вес, 2/Тонна>) (Материал, алюминий)))).*

Более подробно такого рода шаги при построении СП текстов рассматриваются ниже в данной главе и в главе 4.

Часто используемые обозначения. Рассмотрим обозначения, которые в дальнейшем будут часто использоваться в примерах. Для k = 1,...,10 правило P[k] утверждает, что некоторая формула b входит в L(B), некоторая формула b & t принадлежит $T^k(B)$, где $t \in Tp(S(B))$, и некоторая формула z принадлежит $Y^k(B)$. Если $1 \le i \le 10$, B - произвольный к.б., то правила P[0], P[1],... P[i] определяют совместной индукцией множества формул L(B), $T^0(B)$, $T^1(B)$,..., $T^i(B)$, $Y^1(B)$,..., $Y^i(B)$.

Обозначим эти множества через $Lnr_i(B)$, $T^0(B)$, $Tnr_i^1(B)$,... $Tnr_i^i(B)$, $Ynr_i^1(B)$,..., $Ynr_i^i(B$; семейство, состоящее из всех этих множеств, обозначим через $Globset_i(B)$.

Пусть $n \ge 1$, $Z_1,...,Z_n \in Globset_i(B)$, $w_1 \in Z_1,...,w_n \in Z_n$. Тогда, если эти соотношения для формул $w_1,...,w_n$ являются следствием применения некоторых правил $P[l_1],...,P[l_m]$, где $m \ge 1$, то обозначим этот факт выражением вида $B(l_1,..., l_m) \Rightarrow w_1 \in Z_1,...,w_n \in Z_n$. Последовательность $l_1,..., l_m$ может содержать повторяющиеся номера. В выражениях такого рода будет часто пропускаться символ B в обозначениях множеств $Z_1,...,Z_n$; кроме того, будут использоваться выражения $w_1,w_2 \in Z_1$, $w_3,w_4,w_5 \in Z_2$ и т. д. Используя эти обозначения, некоторые соотношения, полученные в Примере 1, можно представить следующим образом:

$B_1(0,1) \Rightarrow$ *все чел, все тур.гр* $\in Lnr_1$,

*все чел & {интс * дин.физ.об}, все тур.гр & {{интс * дин.физ.об}}* $\in Tnr_1^1$,

все & чел & все чел $\in Ynr_1^1$, *все& тур.гр & все тур.гр* $\in Ynr_1^1$.

Выражение $B_1(0,1) \Rightarrow$ *все чел* $\in Lnr_1$, *все чел & {интс * дин.физ.об}* $\in Tnr_1^1$ **равносильно выражению**

$B_1(0,1) \Rightarrow$ *все чел* $\in Lnr_1(B_1)$, *все чел & {интс * дин.физ.об}* $\in Tnr_1^1(B)$. □

Использование реляционных символов и разметка формул

3.4.1. Правила для применения реляционных символов

Правило P[2] позволяет нам, в частности, строить К-цепочки вида $f(a_1,...,a_n)$, где f обозначает функцию с n аргументами $a_1,...,a_n$. Правило P[3] предназначено для построения К-цепочек вида $(a_1 \equiv a_2)$, где a_1 и a_2 обозначают сущности, характеризующиеся типами, сравнимыми друг с другом для отношения конкретизации |—. Используя последовательно P[2] и P[3], мы сможем строить К-цепочки вида $(f(a_1,...,a_n) \equiv b)$, где b — значение f для аргументов $a_1,...,a_n$.

Отметим, что для с.с. S множество главных типов Mtp(S) = Tp(S)\ {[↑сущн], [↑об], [↑пон]} (см. параграф 2.6).

Определение 1. Пусть B — произвольный к.б., S = S(B). Тогда:

(а) $R_1(B)$ = {d∈ X(B) | найдется такой тип t∈ Mtp(S), что t начинается с символа '(' и tp(d) — цепочка вида {t} }; (б) для произвольного n>1, $R_n(B)$ = {d∈ X(B) | найдутся такие $t_1,...,t_n \in$ Mtp(S), что tp(d) - цепочка вида $\{(t_1,...,t_n)\}$ }; (в) для произвольного n>1, $F_n(B) = F(B) \cap R_{n+1}(B)$.

Если n≥1, то элементы множества $R_n(B)$ будем называть n-арными реляционными символами, а элементы $F_n(B)$ будем дополнительно называть n-арными функциональными символами.□

Легко показать, что для произвольного к.б. B и произвольных k,m>1 *из* k≠m *следует* $R_k(B) \cap R_m(B)=\varnothing$.

Определение 2. Обозначим через P[2] высказывание

"Пусть n ≥ 1, f∈ $F_n(B)$, tp = tp(B), $u_1,...,u_n,t \in$ Mtp(S(B)), tp(f) = {($u_1,...,u_n,t$)}; для j = 1,...,n, 0≤ k≤ i, $z_j \in$ Mtp(S(B)), $a_j \in$ L(B), цепочка a_j & z_j принадлежит $T^k_j(B)$; если a_j не входит в V(B), то u_j|— z_j (т. е. z_j является конкретизацией типа u_j); если $a_j \in$ V(B), то u_j и z_j сравнимы для отношения конкретизации |—. Пусть b — цепочка вида $f(a_1,...,a_n)$. Тогда

$b \in L(B), b \,\&\, t \in T^2(B), f \,\&\, a_1 \,\&\, ... \,\&\, a_n \,\&\, b \in Y^2(B)$".□

Перед тем, как сформулировать следующее утверждение, следует напомнить, что символ ‘≡’ является элементом первичного информационного универсума *X(B)* для произвольного к.б. *B*.

Определение 3. Обозначим через P[3] высказывание

“Пусть a_1, $a_2 \in L(B)$, u_1, $u_2 \in Mtp(S(B))$, типы u_1 и u_2 сравнимы для отношения конкретизации |—. Пусть для m = 1, 2, $0 \le k[m] \le i$, a_m & $u_m \in T^{k[m]}(B)$; P - сорт “смысл сообщения” для к.б. B, b — цепочка $(a_1 \equiv a_2)$.Тогда

$b \in L(B)$, $b\&P \in T^3(B)$, и цепочка a_1 & ≡ & a_2 & b входит в множество $Y^3(B)$.” □

В правилах P[2] и P[3] символ i обозначает неизвестное число, такое что $2 \le i \le 10$. Дело в том, что правила P[0] – P[3] и последующие правила будут использоваться вместе с объединяющим их определением, которое начинается с такой фразы: «Пусть B – произвольный к.б., $1 \le i \le 10$.» Число i будет интерпретироваться как максимальный номер правила, которое будет использоваться для построения формулы. Например, если I = 3, то мы можем использовать правила с номерами 0, 1, 2, 3, но не можем использовать правила с номерами 4 - 10. Параметр i в правилах вывода позволяет нам после введения очередного правила с номером i+1 определить формальный язык Lnr_{i+1} (B) и исследовать выразительные возможности этго языка.

Пример 1. Пусть B_1 — к.б., построенный в параграфе 2.8; $i = 3$;

b_1 = *Поставщики(НПО_”Радуга”)* , b_2 = *Колич (Поставщики(НПО_”Радуга”)* ,

b_3 = *(Колич (Поставщики(НПО_”Радуга”) ≡12)* , b_4 = *Колич(все понятие)*,

b_5 = *Колич(все химик)*, b_6 = *(все химик ≡x1)* , b_7 = *Колич(x1)* .

Тогда легко убедиться в справедливости следующих соотношений (принимая во внимание обозначения, введенные в конце предыдущего параграфа):

$B_1 \Rightarrow$ *Поставщики &{(орг, {орг})}* $\in T^0$,

*НПО_”Радуга” & орг * простр.об * интс, Колич&{({[сущн]}, нат)}* $\in T^0$;

$B_1(1, 2) \Rightarrow b_1$ *& {орг}* $\in Tnr_3^2$;

$B_1(0, 2, 2) \Rightarrow b_2 \in Lnr_3$, b_2 *& нат* $\in Tnr_3^2$, *Колич &* b_1 *&* $b_2 \in Ynr_3^2$;

$B_1(0, 2, 2, 0, 3) \Rightarrow b_3 \in Lnr_3$, b_3 *& сообщ* $\in Tnr_3^3$;

$B_1(0, 2, 2) \Rightarrow b_4, b_5 \in Lnr_3$, b_4 *& нат*, b_5 *& нат* $\in Tnr_3^2$;

$B_1(0, 1, 3) \Rightarrow b_6 \in Lnr_3$, b_6*&сообщ* $\in Tnr_3^3$;

x1∈V(B₁), tp(x1) = {[сущн]} , B₁(0, 2) ⇒ b₇∈Lnr₃, b₇ & нат∈ Tnr₃².□

Определение 4. Обозначим через P[4] высказывание

"Пусть n≥1, r∈ $R_n(B)\backslash F(B)$, $u_1,...,u_n \in Mtp(S(B))$, tp = tp(B), tp(r) — *цепочка* {($u_1,...,u_n$)} *при* n > 1 *или* { u_1} *при* n = 1*; для* j = 1,...,n, 0 ≤ k[j] ≤ i, $z_j \in Mtp(S(B))$, $a_j \in L(B)$, *цепочка* a_j & z_j *принадлежит* $T^{k[j]}_j(B)$; *если* $a_j \notin V(B)$*, то* u_j|— z_j*; если* $a_j \in V(B)$, *то* u_j *и* z_j *сравнимы для отношения конкретизации* |—. *Пусть* b — *цепочка вида* $r(a_1,...,a_n)$, P = P(B) — *сорт "смысл сообщения" для* B. *Тогда*

b∈ L(B), b & P∈ $T^4(B)$, и r& a_1&...& a_n & b∈ $Y^4(B)$".□

Пример 2. Пусть B_1 - к.б., рассмотренный в параграфе 2.8; *i =4*;

b_8 = *Меньше(10000, Колич(все химик)),*

b_9 = *Меньше(5000, Колич(все понятие)),*

b_{10} = *Элем(биолог, все понятие),*

b_{11} = *Элем(АО_"Старт", Поставщики(НПО_"Радуга"))* ,

b_{12} = *(П.Сомов ≡ Директор (АО_ "Старт"))* ,

b_{13} = *Знает(П.Сомов, (Колич(Поставщики(НПО_"Радуга"))≡12)),*

b_{14} = *Знает(И.Семенов, (П.Сомов ≡Директор (АО_ "Старт")))* ,

b_{15} = *Меньше(10000, Колич(x1)).*

Принимая во внимание определение к. о. с. $Ct(B_1)$ (см. пример в параграфе 2.7) и применяя правила P[0],...,P[4], мы получаем следующие соотношения:

$B_1(0, 1, 2, 3, 4) ⇒ b_8,..., b_{15} ∈ Lnr_4$, для m = 8,…, 15 b_m & сообщ∈ Tnr_4^4,

Меньше & 1000 & Колич(все химик) & Меньше (10000, Колич(все химик)) ∈ Ynr_4^4.

3.4.2. Правило, позволяющее помечать формулы

Основное назначение правила P[5] заключается в том, чтобы с помощью переменных из множества V(B) , где B – рассматриваемый к.б., помечать в семантических представлениях (СП) ЕЯ-текстов: (а) описания различных сущностей, упомянутых в тексте (физических объектов, событий, понятий и т. д.), (б) фрагменты, являющиеся семантическими представлениями предложений и более крупных частей текстов, на которые имеются ссылки в любой части

текста. С помощью этого правила (в сочетании с другими правилами) окажется возможным отражать референтную (ссылочную) структуру дискурсов, в которых имеются, в частности, выражения (а) этот прибор, на этом заводе, в этом городе, ему , о нем , (б) данный метод, это распоряжение, эту команду , этот вопрос, об этом, про это.

Определение. Обозначим через P[5] высказывание

"Пусть $a \in L(B) \setminus V(B)$, $0 \le k \le i$, $k \ne 5$, $t \in Mtp(S(B))$, $a\& t \in T^k(B)$; $v \in V(B)$, $z \in Mtp(S(B))$, $v \& z \in T^0(B)$, $z \text{ |— } t$, v не является подцепочкой цепочки a. *Пусть* b - цепочка вида a : v. Тогда $b \in L(B)$, $b\& t \in T^5(B)$, $a\& v\& b \in Y^5(B)$".□

Условие "k≠5 " вводится для того, чтобы правило P[5] нельзя было применять произвольное число раз подряд; в противном случае мы могли бы получать формулы с "избыточной разметкой", то есть выражения вида $a : v_1 : v_2 \ldots v_n$, где $n > 1$, и $v_1, v_2, \ldots, v_n \in V(B)$.

Пример 3. Рассмотрим, как м прежде, к.б. B_1, построенный в параграфе 2.8. Пусть i = 5, $a_1 = b_3 =$ *(Колич (Поставщики(НПО_"Радуга") ≡ 12),* $k_1 = 3$, $t_1 =$ *сообщ = P(*B_1*).* Тогда, очевидно, $a_1\& t_1 \in Tnr_5^3(B_1)$.

Предположим, что v_1 = P1, z_1 = сообщ = P(B_1). Тогда, в соответствии с определением базиса B_1, $v_1\& z_1 \in T^0(B_1)$; кроме того, $z_1 \text{|—} t_1$ (т. к. $z_1 = t_1$), и v_1 не является подстрокой a_1.

Пусть $b_{16} =$ *(Колич (Поставщики(НПО_"Радуга")) ≡ 12) : P1* . Тогда, в соответствии с правилом P[5], $b_{16} \in Lnr_5(B_1)$, b_{16} *&* *сообщ* $\in Tnr_5^5(B_1)$, $a_1\& v_1\& b_{16} \in Ynr_5^5(B_1)$.

Пусть $b_{17} =$ *Поставщики(НПО_"Радуга"): x3* . Тогда легко видеть, что

$$B_1(0, 2, 5) \Rightarrow b_{17} \in Lnr_5, b_{17} \&\{орг\} \in Tnr_5^5,$$

$$Поставщики(НПО\_"Радуга") \& x3 \& b_{17} \in Ynr_5^5.$$

В выражении b_{16} переменная P1 помечает СП фразы П1 = "У НПО "Радуга" имеется 12 поставщиков". Поэтому, если это выражение является частью длинной цепочки, то справа от вхождения b_{16} в такую цепочку можно использовать переменную P1 для повторного представления смысла указанной фразы П1 вместо значительно более длинного СП фразы П1. В цепочке b_{17}

переменная x3 является меткой множества, состоящего из всех поставщиков научно-производственного объединения “Радуга”.

3.5. Использование логических связок “не”, “и” , “или”

Правила P[6] и P[7] в сочетании с другими правилами позволяют по сравнению с языком логики предикатов более полно моделировать (на уровне семантических представлений текстов) способы использования связок “не”, “и”, “или” в предложениях на русском, английском и многих других языках. В частности, конструктивно учитывается существование фраз вида “Этот препарат выпускается не в Польше”, “Профессор Сухов работает не в МГУ”, “Этот патент внедрен в Австрии, Венгрии, Нидерландах и Великобритании”. С этой целью, во-первых, допускается присоединение связки $\neg$ (“не“) не только к семантическим представлениям (СП) фраз, обозначающих высказывания, но и к обозначениям предметов, событий, понятий. Во-вторых, разрешается соединять связками $\wedge$ (“конъюнкция”, т.е. логическое “и“) и $\vee$ (“дизъюнкция”, т.е. логическое “или”) не только СП высказываний, но и обозначения предметов, ситуций, понятий, множеств предметов и т.д.

Правило P[6] предназначено для построения l-формул вида $\neg$a, где a$\in$ Lnr_i(B).

Определение 1. Обозначим через P[6] высказывание «Пусть a$\in$ L(B), t$\in$ Mtp(S(B)), $0 \le k \le i$, k$\notin$ {2,5,10}, a& t$\in$ T^6(B), b — цепочка вида $\neg$ a. Тогда b$\in$ L(B), b & t$\in$ T^6(B), цепочка вида $\neg$ & a& b входит в множество Y^6(B).” □

Замечание 1. Условие k$\notin$ {2,5,10} означает следующее: если l-формула a выведена каким-либо образом с помощью правил P[0], P[1],...,P[i], то правило P[2] или P[5] или P[10] не может быть применено на последнем шаге вывода. Таким образом, условие k$\neq$2 означает, что не разрешается строить l-формулы вида $\neg f(d_1,\ldots,d_m)$, где f$\in$ F(B). Условие k$\neq$5 вводится для того, чтобы по любому выражению вида $\neg$ a:v, где v$\in$ V(B), можно было однозначно найти то правило, которое применялось последним; этим правилом будет P[5]. Условие k$\neq$10 запрещает строить выражения вида $\neg$ $<C_1, \ldots, C_m>$, т.е. запрещает присоединять связку “не“ к обозначениям упорядоченных наборов.

Пример 1. Пусть B_1 – к.б., построенный в параграфе 2.8, i=6. Тогда легко видеть, что выполняются следующие соотношения:

B_1 *(0, 6) => ¬ биолог* $\in Lnr_6$, *¬ биолог & ↑интс*дин.физ.об* $\in Tnr^6_6$.

B_1*(0,6,4,4) => Знает (П.Сомов, Сейчас, Явл1(И. Семенов, ¬ биолог))* $\in Tnr^4_6$;

B_1*(0,4,4,6) => ¬ Знает(П.Сомов, Сейчас, Явл1(И.Семенов, химик))* $\in Tnr^6_6$.

Правило P[7] позволяет строить *l*-формулы вида ($a_1 \wedge a_2 \wedge \ldots \wedge a_n$) и вида ($a_1 \vee a_2 \vee \ldots \vee a_n$). Например, формулы *(химик ∨ биолог), (математик ∧ художник), (Имя (x1, 'Сергей') ∧ Фам(x1, 'Жаворонков') ∧ Квалиф(x1, химик)).*

Определение 2. Обозначим через P[7] высказывание: "Пусть n>1, t∈ Mtp(S(B)), для m=1,…,n, $0 \le k[m] \le i$, a_m & t∈ $T^{k[m]}(B)$, s∈ {∧, ∨}, b – цепочка вида (a_1 s a_2 s … s a_n). Тогда *b∈ L(B), b & t∈ $T^7(B)$, s & a_1&…& a_n & b∈$Y^7(B)$"*.

Замечание 2. Данное правило требует, чтобы все выражения, соединенные за один шаг логической связкой, имели один и тот же тип. Так как *t* не обязательно является сортом «смысл сообщения», то по правилу P[7] можно соединять логическими связками не только семантические представления высказываний, но и обозначения различных объектов, простые и составные обозначения понятий, простые и составные обозначения целей интеллектуальных систем.

Пример 2. Пусть B_1 – к.б., построенный в в параграфе 2.8, i= 7 ,

b_1 *= (А.Зубов ∧ И.Семенов),* b_2 *= (химик ∨ биолог),*

b_3 *= ((Колич (Друзья (А.Зубов)) ≡3) :* P_1 *∧ Знает (П.Сомов, Сейчас,* P_1*)*

∧ ¬Знает (П.Сомов, Сейчас, Явл (А.Зубов, (химик ∨ биолог)))),

b_4 *= Элем ((АО_"Салют" ∧ АО_"Старт"), Поставщики (НПО_"Радуга"))* .

Тогда легко показать, что B_1*(0,7) =>* b_1, $b_2 \in Lnr_7$, b_1*& интс*дин.физ.об* $\in Tnr^7_7$,

b_2 *&* ↑*интс*дин.физ.об* $\in Tnr^7_7$, B_1*(0,2,2,3,5,4,7,4,4,6,7) =>* $b_3 \in Lnr_7$,

b_3 *& сообщ*$\in Tnr^7_7$; B_1*(0,7,0,2,4) =>* $b_4 \in Lnr_7$, b_4 *& сообщ*$\in Tnr^4_7$.

3.6. Построение составных обозначений понятий и объектов

Правило для построения составных обозначений понятий

Рассмотрим правило P[8], предназначенное для построения составных обозначений понятий. С помощью этого правила строятся *l*-формулы вида $a^*(r_1, d_1)\ldots(r_n, d_n)$ и *t*-формулы вида $a^*(r_1, d_1)\ldots(r_n, d_n) \,\&\, t$, где a – элемент первичного информационного универсума $X(B)$ и интерпретируется как простое обозначение понятия, $n \geq 1$, для $i=1,\ldots,n$, $r_i \in R_2(B)$, d_i – обозначение некоторой сущности. Например, выбирая подходящий конептуальный базис, можно построить *l*-формулы

*город * (Страна, Россия)* , *учебник * (Область, биология), понятие*(Имя.пон, "молекула")* , *туристич.группа*(Количество, 12)(Состав, (химик ∧ биолог))*

и t-формулу *город*(Страна, Россия)* & *↑простр.об* .

Используя правило P[8] вместе с правилом P[1] и другими правилами, можно будет строить составные обозначения объектов и множеств объектов в виде *q des*, где *q* — интенсиональный квантор (см. параграф 1.8), *des* — составное обозначение понятия, построенное с помощью правила *P[8]* на последнем шаге вывода. В частности, формулы

*некотор учебник * (Область, биология), все город * (Страна, Россия) , некотор чел*(Возраст, 18/год>), некотор понятие*(Имя.пон, "молекула"), некотор туристич.группа* (Количество, 12))(Качеств-состав, (химик ∧ биолог))* .

Определение 1. Для произвольного к.б. B $Tconc(B) = \{t \in Tp(S(B)) \mid t$ начинается с $\uparrow\} \cup Spectp$, где $Spectp = \{[\uparrow сущн], [\uparrow пон] , [\uparrow об]\}$.

Поясним обозначения, используемые ниже в определении правила P[8]. Каждый такой элемент c первичного информационного универсума $X(B)$, что $tp(c) \in Tconc(B)$, будем интерпретировать как обозначение понятия. Множество $R_2(B)$ состоит из бинарных реляционных символов (некоторые из них могут соответствовать функциям с одним аргументом); $F(B)$ — множество функциональных символов. Элемент $ref = ref(B)$ из $X(B)$ называется *квантором референтности* (см. параграф 2.8) и интерпретируется как информационная

единица (другими словами, семантическая единица), соответствующая значению слова “некоторый” в выражениях в единственном числе (“некоторая книга”, “некоторая страна” и т. д.); в рассматриваемых примерах *ref(B)* – это символ *нек* ; *P(B)* — обозначение сорта “смысл сообщения” к.б. *B*.

Определение 2. Обозначим через P[8] высказывание

“Пусть $a \in X(B)$, $tp = tp(B)$, $t \in Tconc(B)$, $t = tp(a)$, $P = P(B)$, $ref = ref(B)$. Пусть $n \geq 1$, $\forall m=1,\ldots,n$ $r_m \in R_2(B)$, c_m – цепочка вида *ref a*, $d_m \in L(B)$, h_m – цепочка вида $(r_m(c_m) \equiv d_m)$ в случае $r_m \in R_2(B) \cap F(B)$, и h_m – цепочка вида $r_m(c_m, d_m)$ в случае $r_m \in R_2(B) \setminus F(B)$; если $r_m \in F(B)$, то $h_m \,\&P \in T^3(B)$; если $r_m \notin F(B)$, то $h_m \,\&P \in T^4(B)$. Пусть b - цепочка вида $a^*(r_1,d_1)\ldots(r_n,d_n)$. Тогда $b \in L(B)$, $b\,\& t \in T^8(B)$, $a\,\& h_1 \&\ldots\& h_n \& b \in Y^8(B)$.” □

Пример 1. Предположим, что B_1 — к.б., определенный в параграфе 2.8, $i = 8$. Тогда рассмотрим возможный путь построения формулs b_1 (задаваемой ниже), соответствующей понятию “туристическая группа, состоящая из 12 человек”. При этом будем использовать обозначения, введенные в конце параграфа 3.3.

Пусть a = *тур.гр*, $t = tp(a) = \uparrow\{$*интс * дин.физ.об*$\}$, P = *сообщ* , ref = *нек*,

$$n = 1,\ r_1 = Колич,\ c_1 = ref\ a = нек\ тур.гр,\ d_1 = 12,$$

$$h_1 = (r(c) \equiv d) = (Колич(нек\ тур.гр) \equiv 12).$$

Тогда $B_1(0, 1, 2, 3) \Rightarrow h_1 \in Lnr_3(B_1)$, $h_1\,\&$ *сообщ* $\in Tnr_3^{\,8}(B_1)$.

Пусть $b_1 = a^*(r_1, d_1) =$ *тур.гр* (Колич, 12)*. Тогда из *P[8]* следует, что

$$b_1 \in Lnr_8(B_1),\ b_1\,\& \uparrow\{инс^*\ дин.физ.об\} \in Tnr_8^{\,8}(B_1).\ .\square$$

3.6..2. Построение составных обозначений объектов

Правило P[8] можно использовать для построения составных обозначений разных предметов, ситуаций и множеств предметов или ситуаций. Для построения составного обозначения предмета после правила P[8] применяется правило P[1] и строится выражение вида $ref\ a^*(r_1,d_1)\ldots(r_n,d_n)$. Например, таким образом можно построить выражения *нек город (Страна, Россия) (Колич. жит., 350000), нек чел* (Фам, “Сомов”)(Имя, “Петр”).*

Пример 2. Пусть b_2 = *нек биолог*(Элем, нек тур.гр*(Колич,12))*. Тогда

$B_1(0, 1, 2, 3, 8, 1, 1, 4, 8, 1) \Rightarrow b_2 \in Lnr_8(B_1)$, $b_2\,\&$ *интс* дин.физ.об* $\in Tnr_8^{\,1}(B_1)$.

Цепочку b_2 будем интерпретировать как составное обозначение какого-то (вполне определенного) человека, являющегося биологом и входящего в состав тургруппы из 12 человек. Подцепочку *нек тур.гр*(Колич,12))* будем интерпретировать как обозначение какой-то конкретной (в контексте ситуации общения) тургруппы, состоящей из 12 человек□

Пример 3. Исходя из предположений Примера 1, рассмотрим путь построения возможного СП фразы "А.Зубов включил И.Семенова в туристическую группу, состоящую из 12 человек". Пусть переменная *x1* обозначает момент времени, и b_3 = *(Включ1(А.Зубов , И.Семенов , x1, нек тур.гр*(Колич,12)) ∧Раньше (x1,Сейчас)).*

Тогда $B_1(0, 1, 2, 3, 8, 1, 4, 4, 7) \Rightarrow b \in Lnr_8(B_1)$, b_3 & *сообщ* $\in Tnr_8^7(B_1)$.

Пример 4. Пусть Т1 = "П.Сомов знает, что И.Семенов является директором фирмы, персонал которой включает 38 человек". Это предложение включает составное обозначение фирмы с персоналом из 38 человек. Пусть

b_4 = *Знает (П.Сомов, Сейчас, ((И.Семенов ≡Директор (нек фирма * (Описание, P1): x1)) ∧ (P1 ≡(Колич(Персонал (x1)) ≡38))))* .

Проследим путь вывода формулы b_4 с помощью правил Р[0], Р[1], …, Р[8] . Пусть $i=8$, B_1 – к.б., построенный в параграфе 2.8. Рассмотрим новый к.б. B_2, отличающийся от базиса B_1 тем, что $X(B_2) = X(B1) \cup \{$*фирма, Описание*$\}$, *tp(фирма) = ↑орг * простр.об * интс , tp(Описание)={([ob],P)}.*

Если *a=фирма, t=орг * простр.об * интс* , то, очевидно, *a* & $t \in T^0(B_2)$. По определению к.б. B_2, множество переменных $V(B_2)$ включает такую переменную *P1*, что $tp(P1)=P(B_2)$ = *сообщ* . Пусть

$n=1$, r_1= *Описание*, c_1= *ref a= нек фирма*,

$h_1 = r_1(c_1,d_1)$ = *Описание(нек фирма, P1)*, $b_5 = a^*(r_1,d_1)$= *фирма*(Описание,*P_1*).*

Тогда из правила Р[8] и правил Р[0], Р[1], Р[4] следует, что

$b_5 \in Lnr_i(B_2)$, b_1& *↑орг* простр.об * интс* $\in Tnr_i^8(B_2)$.

Пусть b_6 = *ref* b_1 = *нек фирма * (Описание,*P_1*).* Тогда

$B_2(0,1,4,8,1) \Rightarrow b_6 \in Lnr_i(B)$, b_6 & *орг* простр.об * интс* $\in Tnr_i^1(B_2)$.

Пусть $b_7 = b_6$ *: x1= нек фирма*(Описание,* P_1*): x1.* Тогда

$B_2(0,1, 4, 8,1,5) \Rightarrow b_7 \in Lnr_i$, b_7 & *орг* простр.об * интс* $\in Tnr_i^5$.

Пусть b_8= *(Р1≡(Колич(Персонал(x1))≡38))*. Тогда $B_2(0,2,2,3,3) \Rightarrow b_8 \in Lnr_i$, b_8 & *сообщ* $\in Tnr_i^3$. Пусть b_9 = *(И.Семенов≡Директор(b_7))*, тогда

$$B_2(0,1,4,8,1,5,2,3) \Rightarrow b5 \in Lnr_i,\ b5 \& сообщ \in Tnr_i^3.$$

Пусть b_{10} = *Знает (П.Сомов, Сейчас, ($b_9 \wedge b_8$))*. Тогда

$$B_2(0,1,4,8,1,5,2,2,3,7,4) \Rightarrow b_{10} \in Lnr_i,\ b_{10} \& сообщ \in Tnr_i^4.$$

Легко видеть, что b_{10} совпадает *с* b_4. Значит, мы построили вывод формулы b_4. Рассмотренный метод построения СП текста Т1 является весьма общим; этот метод может использоваться в самых разнообразных случаях для построения СП предложений со сложными причастными оборотами и придаточными определительными предложениями.

3.7. Использование в формулах кванторов существования и всеобщности. Построение обозначений упорядоченных наборов

3.7.1. Применение кванторов существования и всеобщности

Правило Р[9] позволяет строить формулы с кванторами ∃,∀, похожие на формулы логики предикатов первого порядка. Отличие, в частности, заключается в том, что явным образом ограничивается область действия кванторов, и переменные могут обозначать не только предметы, числа, но и множества различных сущностей. С помощью Р[9] можно построить l-формулы вида *Q v (concept) A (v)*, где $Q \in \{\exists, \forall\}$,$v \in V(B)$ – переменная; *concept* – это простое обозначение понятия (в этом случае *concept* – такой элемент первичного информационного универсума *X(B)*, что *tp(concept)* начинается с символа ↑) или составное обозначение понятия; например, *concept* = *страна*(Место, Европа))*, *A(v)* – формула, включающая v и интерпретируемая как СП высказывания.

Пример 1. Выбирая подходящий к.б. В, с помощью Р[9] и нескольких других правил можно построить СП предложения “В каждой стране Европы есть город с количеством жителей, превышающим 30 тысяч человек” следующим образом:
∀x1(страна*(Место,Европа)) ∃x2(город) (Место(x2, x1) ∧ Меньше (30000, Колич.элем. (Жители(x2)))) .

Определение 1. Через P[9] обозначим высказывание

“Пусть $Q\in\{\exists,\forall\}$, $A\in L(B)$, $P=P(B)$, $k\in\{3,4,6,7,9\}$, A & $P\in T^k(B)$, $v\in V(B)$, $tp(v)=[сущн]$ – базовый тип «сущность», A включает символ v, $m\in\{0,8\}$, $concept\in L(B)\backslash V(B)$, $u\in Tconc(B)$, где $Tconc(B)$ – множество всех типов из $Tp(S(B))$, начинающихся с символа $\uparrow$; цепочка вида $concept$ & u входит в множество $T^m(B)$, цепочка A не включает подцепочек видов : v, ∀v, ∃v, и цепочка A не имеет окончания вида :z, где z – произвольная переменная из $V(B)$.

Пусть b=Q v (concept) A. Тогда

$b\in L(B)$, b & $P\in T^9(B)$, Q & v & $concept$ & A& $b\in Y^9(B)$” .

Условие “$k\in\{3,4,6,7,9\}$” означает, что перед использованием правила P[9] (т.е. перед присоединением к формуле квантора существования или всеобщности) должно применяться одно из правил P[3], P[4], P[6], P[7] или P[9].

Пример 2. Пусть $i=9$ и существует такой к.б. B, что выполняются следующие соотношения:

$$город, страна, Европа, Место, Колич.элем., Жители, Меньше \in X(B),$$

$$простр.об, нат.чис, дин.физ.об, интс\in St(B),$$

$$tp(город)= tp(страна)= \uparrow простр.об,\ tp(Европа)=простр.об,$$

$$tp(Место)=\{(простр.об, простр.об)\},$$

$$tp(Колич.элем.)=\{(\{сущн]\}, нат.чис)\},$$

$$tp(Жители)=\{(простр.об, \{дин.физ.об * интс\})\},$$

$$tp(Меньше)=\{(нат.чис, нат.чис)\}, Колич.элем, Жители \in F(B) ,$$

$$x1, x2\in V(B) ,\ \ tp(x1)=tp(x2)=[сущн],\ 30000\in X(B) ,$$

$$tp(30000)=нат.чис, сообщ = P(B).$$

Перечисленные информационные единицы интерпретируются следующим образом: *простр.об, нат.чис, дин.физ.об, интс* – сорта “пространственный объект”, “натуральное число” , “динамический физический объект”, “интеллектуальная система”; *город, страна* – обозначения одноименных понятий; *Европа* – обозначение региона Европа; *Место* - обозначение бинарного отношения, связывающего пространственные объекты; *Колич.элем* – обозначение функции “Количество элементов множества”; *Жители* - обозначение функции, ставящей в соответствие населенному пункту множество

всех его жителей; *Меньше* - обозначение бинарного отношения “меньше” на множестве натуральных чисел.

Пусть $Q_1=\exists$, $v_1=x2$, $concept_1=$*город*, $A_1=$(*Место*(x2, x1)$\wedge$ *Меньше*(30000, *Колич.элем.* (*Жители*(x_2))) , $b_1=Q_1v_1(concept_1)A_1$. Тогда нетрудно проверить, что выполняется следующее соотношение: $B(0,2,2,3,4,4,7,9)\Rightarrow b_1\in Lnr_i$, b_1 & *сообщ*$\in Tnr_i^9$. Пусть $q_2=\forall$, $v_2=x1$, $concept_2=$ *страна* * (*Место*,*Европа*) , $A_2=b_1$, $b_2=Q_2v_2(concept_2)A_2$.Тогда $B(0,2,2,3,4,4,7,9,0,1,4,8,9)\Rightarrow b_2\in Lnr_{i=9}$, b_2 & *сообщ*$\in Tnr_i^9$, Q_2 &v_2 & $concept_2$ & b_1& $b_2\in Y_i^9(B)$.

3.7.2. Построение обозначений упорядоченных наборов

Правило P[10] предназначено для построения l-формул вида $<a_1,\ldots,a_n>$, где $n>1$ и $a_1,\ldots,a_n$ – обозначения некоторых сущностей. Такие формулы будут интерпретироваться как обозначения упорядоченных наборов.

Определение 2. Через P[10] обозначим высказывание:

“Пусть $n>1$, для $m=1,\ldots, n$ выполняются соотношения $a_m\in L(B)$, $u_m\in Tp(S(B))$, $0\le k[m]\le 10$, a_m & $u_m\in T^{k[m]}(B)$. Пусть t - цепочка вида $(u_1,u_2,\ldots,u_n)$, b - цепочка вида $<a_1,\ldots,a_n>$. Тогда

$$b\in L(B),\ b\ \&\ t\in T^{10}(B),\ a_1\&a_2\&\ldots\&\ a_n\ \&\ b\in Y^{10}(B).”$$

Пример 3. Пусть B_1 – к.б., построенный в параграфе 2.8, $i=10$, b_3 - цепочка (*Элем*(x3 ,S1) $\equiv$((x3 $\equiv$<*нек вещ* : x1, *нек вещ* : x2>) $\wedge$ (*Меньше* (x1 , x2)$\vee$(x1$\equiv$ x2)))) . Тогда *S1* можно интерпретировать как обозначение отношения «≤» на множестве вещественных чисел. То есть S_1 обозначает множество всех таких пар (x1, x2), что x1, x2 – вещественные числа, и x1 ≤ x2 . Легко проверить, что $B(0,4,1,5,1,5,4,10,3,7,7,3)\Rightarrow b_3\in Lnr_i(B_1)$, b_3 & *сообщ*$\in Tnr_i^{10}(B_1)$.

3.7.3. Сводная таблица правил P[0] – P[10]

Суммарный объем определений правил P[0] – P[10] и поясняющих их примеров довольно велик. При построении семантических представлений (СП) не только связных текстов, но и большинства отдельных предложений обычно используется значительная часть этих правил, причем в самых разнообразных комбинациях (анализу возможных применений этих правил посвящена Глава 4). В связи с этим представляется целесообразным дать сжатую, недетализированную характеристику каждого правила из списка P[0] – P[10] в приводимой ниже сводной таблице. Эта таблица облегчит анализ использования правил P[0] – P[10] при построении СП ЕЯ-текстов и при формировании фрагментов знаний о мире в примерах этой и последующих глав.

Правило	Результаты применения
P[0]	Начальный запас формул, определяемый первичным информационым универсумом *X(B)* , множеством переменных V(B) и отображением *tp* , задающим типы элементов из этих множеств
P[1]	*l* – формулы вида *q a* или вида *q a* $*(r_1,d_1)\ldots(r_n,d_n)$, где *q* – интенсиональный квантор, *a* – простое обозначение понятия, $1 \le n$, $r_1 \ldots r_n$ – характеристики сущностей
P[2]	*l*– формула вида $f(a_1,\ldots, a_n)$, $1 \le n$, где *f* – имя функции; *t* - формулы вида $f(a_1,\ldots, a_n)$ & *t*, где *t* – тип значения функции f для аргументов $a_1,\ldots, a_n$
P[3]	*l* – формулы вида $(a_1 \equiv a_2)$ и *t* – формулы вида $(a_1 \equiv a_2)$ & *P*, где *P* – сорт «смысл сообщения»
P[4]	*l* – формулы вида $r(a_1,\ldots, a_n)$ и *t* – формулы вида $r(a_1,\ldots, a_n)$ & *P*, где $n \ge 1$, *r* - n-арный реляционный символ
P[5]	*l* - формула вида *form : v*, где *v* – метка формулы *form*
P[6]	по *l* – формуле *form* строится *l* – формула $\neg$ *form* (отрицание)
P[7]	по логической связке $s \in \{\wedge, \vee\}$ и *l* -формулам $a_1,\ldots, a_n$ строится *l*-формула (a_1 s a_2 s … a_n), где $n>1$
P[8]	по простому обозначению понятия *conc*, характеристикам объектов $r_1,\ldots, r_n$ $(n \ge 1)$, *l* -формулам $d_1 \ldots d_n$ строится

	l-формула *conc* $*(r_1,d_1)\ldots(r_n,d_n)$ и *t*-формула *conc* $*(r_1,d_1)\ldots(r_n,d_n)$: *t* , где *t* начинается с символа $\uparrow$. Такие формулы интерпретируются как составные обозначения понятия. Пример: *страна* $*$ *(Место, Европа)* & $\uparrow$*простр. объект*
P[9]	строятся *l*-формулы вида *Qv(conc)A* и *t*-формулы *Qv(conc)A&P*, где Q$\in\{\exists,\forall\}$, v$\in$ V(B) , *conc* - простое или составное обозначение понятия, *A* - *l*-формула, обозначающая высказывание, *P* – сорт «смысл сообщения»
P[10]	Для построения *l*-формул вида $<a_1,\ldots a_n>$, где $n>1$, интерпретируемых как обозначения упорядоченных наборов

Табл. 3.1. Краткая характеристика правил P[1] – P[10].

3.8. Стандартные К-языки. Математическое исследование их свойств

Определение 1. Пусть *B* - произвольный концептуальный базис, тогда:

(а) $D(B) = X(B) \cup V(B) \cup \{',' , '(', ')', ':' , '*', '<', '>'\}$,

(б) $Ds(B) = D(B) \cup \{' \&'\}$, (в) $D^+(B)$ и $Ds^+(B)$ — множества всех непустых конечных последовательностей элементов из *D(B)* и *Ds(B)* соответственно. □

Таким образом, $Ds(B) = X(B) \cup V(B) \cup \{',', '(', ')', ':' , '*', '<', '>', '\&' \}$; каждое из множеств *D(B), Ds(B)* включает, в частности, символ “запятая”.

Определение 2. Пусть B – произвольный к.б., $1\le i\le 10$, и множества цепочек

$$\mathbf{L(B) \subset D^+(B)\ , T^0(B), T^1(B), \ldots , T^i(B), Y^1(B), \ldots , Y^i(B) \subset Ds^+(B)}$$

являются наименьшими множествами, совместно задаваемыми правилами P[0] – P[I]. Тогда обозначим эти множества соответственно через $Lnr_i(B)$, $T^0(B)$, $Tnr_i^1(B)$, ... , $Tnr_i^i(B)$, $Ynr_i^1(B)$, ... , $Ynr_i^i(B)$ и обозначим семейство (т.е. множество), состоящее из всех этих множеств, через $Globset_i(B)$. Кроме того, пусть

$$\mathbf{T_i(B) = T_0(B) \cup Tnr_i^1(B) \cup \ldots \cup Tnr_i^i(B)} \quad , \qquad \mathbf{(3.8.1)}$$

$$Y_i(B) = Ynr_i^1(B) \cup ... \cup Ynr_i^i(B) \quad , \qquad (3.8.2)$$

$$Form_i(B) = Lnr_i(B) \cup T_i(B) \cup Y_i(B) \quad . \qquad (3.8.3)$$

Определение 3 (итоговое). Если B-произвольный к.б., то

$$Ls(B)=Lnr_{10}(B) \quad , \qquad (3.8.4)$$

$$Ts(B)=T_{10}(B) \quad , \qquad (3.8.5)$$

$$Ys(B)=Y_{10}(B) \quad , \qquad (3.8.6)$$

$$Forms(B)=Form_{10}(B) \quad . \qquad (3.8.7)$$

$$Ks(B)= (B, Rls) , \qquad (3.8.8)$$

где Rls = {P[0], P[1], ... , P[10]} .

Упорядоченная пара *Ks(B)* называется К-исчислением (концептуальным исчислением) в базисе *B*; элементы множества *Forms(B*) называются формулами, выводимыми в базисе B. Формулы из *Ls(B), Ts(B) и Ys(B)* называются соответственно *l*-формулами, *t*-формулами, *y*-формулами. Множество *l*-формул *Ls(B)* называется стандартным концептуальным языком (стандартным К-языком, СК-языком) в базисе B.

Утверждение 3.1. **Если B-произвольный к.б., то (а) множество $Lnr_0(B)$ не является пустым; (б) если $1 \le i \le 10$, то $Lnr_{i-1}(B) \subseteq Lnr_i(B)$.**

Доказательство. (а) Для любого к.б. B первичный информационный универсум X(B) включает непустое множество сортов St(B), причем $\forall s \in St(B)$, $tp(s)=\uparrow s$. Тогда при i = 0 из правила P[0] следует, что $Lnr_0(B)$ включает X(B) и, как следствие, включает St(B) . Поэтому множество $Lnr_0(B)$ непусто.

(б) Структура правил P[1] – P[10] показывает, что добавление нового правила может либо не изменить множество выводимых формул, либо его расширить. В частности, если i = 2, то в случае, когда множество функциональных символов F(B) пусто, $Lnr_{i-1}(B) = Lnr_i(B)$.

Утверждение 3.2. **Если B - произвольный к.б., то множества Ls(B), Ts(B), Ys(B) не являются пустыми.**

Доказательство

Из Утверждения 3.1 следует, что $Lnr_0(B) \subset Ls(B)$. Поэтому Ls(B) непусто. Из определения концептуально-объектной системы (к.о.с.) вытекает, что существуют такие различные переменные $v_1, v_2 \in V(B)$, что $tp(v_1) = tp(v_2) =$ [сущн]. Тогда из правила P[3] вытекает, что цепочка $(v_1 \equiv v_2)$ & P входит в

множество Ts(B), и v_1 & v_2 & ($v_1 \equiv v_2$) ∈ Ys(B), где P=P(B)-сорт "смысл сообщения" . Поэтому множества Ts(B) и Ys(B) не являются пустыми..

Утверждение 3.3. **Если B - произвольный к.б., то: (а) Если $\tau \in$ Ts(B), то τ - цепочка вида α & t, где $\alpha \in$ Ls(B), t∈ Tp(S(B)), и такое представление, зависящее от τ, единственно для каждой цепочки τ; (б) Если $\gamma \in$ Ys(B), то найдутся такое n>1 и такие цепочки $\alpha_1, \alpha_2, \ldots, \alpha_n, \beta \in$ Ls(B), что γ - цепочка вида α_1 & α_2 & ... & α_n & β; кроме того, такое представление зависящее от γ, единственно для любого γ.**

Доказательство. Справедливость этого предложения непосредственно следует из определения к.б. и из структуры правил P[0] – P[10].

Утверждение 3.4. **Пусть B - произвольный к.б., d ∈ X(B) ∪ V(B). Тогда не найдутся такие k, n , где 1≤ k ≤ 10, n>1, и такие $\alpha_1, \alpha_2, \ldots, \alpha_n \in$ Ls(B), что**

α_1 & α_2 & ... & α_n & d ∈ Ynr_{10}^{k}(B). (*)

Интерпретация. **Смысл утверждения в том, что для каждого элемента d, входящего в первичный информационный универсум X(B) или являющегося переменной из V(B), нельзя получить этот элемент d с помощью каких-либо операций, задаваемых правилами P[1] – P[10].**

Доказательство **(от противного)**

Предположим, что существуют такие к.б. B, d ∈ X(B) ∪ V(B), натуральное число k , где 1≤ k ≤ 10, n>1, $\alpha_1, \alpha_2, \ldots, \alpha_n \in$ Ls(B), что справедливо соотношение (*). Для любого такого m, что 1≤ m ≤ 10, Ynr_{10}^{m}(B) включает цепочку α'_1 & α'_2 & ... & α'_n & d', где d' не включает символ &, только в том случае, когда цепочка d' построена из цепочек $\alpha'_1, \alpha'_2, \ldots, \alpha'_n$ применением правила P[m] на последнем шаге вывода. Но тогда из структуры правил P[1] – P[10] непосредственно следует, что цепочка d' должна содержать по крайней мере два символа. Но так как d ∈ X(B) ∪ V(B), то элемент d рассматривается как символ и поэтому имеет длину 0. Мы получили противоречие из нашего предположения, что доказывает Утверждение 3.4.

Утверждение 3.5. **Пусть B - произвольный к.б., z ∈ Ls(B) \ (X(B) ∪ V(B)). Тогда существует один и только один такой набор (k, n, $y_1, y_2, \ldots, y_n$), где 1 ≤ k ≤ 10, n > 1, $y_1, y_2, \ldots, y_n \in$ Ls(B), что**

y_1 & y_2 & ... & y_n & z ∈ Ynr_{10}^k(B).

Интерпретация. **Если l-формула z не входит в (X(B) ∪ V(B)), то тогда найдутся единственное правило P[k], где 1≤ k≤ 10, и единственный такой набор l-формул y_1, y_2, ... , y_n, что цепочка z построена из "блоков" y_1, y_2, ... , y_n применением ровно один раз правила P[k].**

Справедливость Утверждения 3.5 вытекает из двух лемм, рассматриваемых ниже. Для того, чтобы сформулировать эти леммы, потребуется

Определение 4. Пусть *B*-произвольный к.б., $n \geq 1$, для $i=1,\ldots,n$ $c_i \in D(B)$, $s=c_1\ldots c_n$, $1 \leq k \leq 10$. Тогда через $lt_1(s, k)$ и $lt_2(s, k)$ обозначим количество вхождений символа '(' и символа '<', соответственно, в подцепочку $c_{1\ldots}c_k$ цепочки $s=c_{1\ldots}c_n$. Через $rt_1(s, k)$ и $rt_2(s, k)$ обозначим количество вхождений символа ')' и символа '>' в подцепочку $c_{1\ldots}c_k$ цепочки *s*. Если в подцепочку $c_1\ldots c_k$ не входит символ '(' или символ '<', то, соответственно, $lt_1(s, k) = 0$, $lt_2(s, k) = 0$, $rt_1(s, k) = 0$, $rt_2(s, k) = 0$.

Лемма 1. Пусть *B*-произвольный к.б., $y \in Ls(B)$, $n \geq 1$, для $i = 1,\ldots, n$ $c_i \in D(B)$, $y = c_1\ldots c_n$. Тогда:

(a) при $n>1$ для каждого $k = 1,\ldots, n-1$ и каждого $m = 1, 2$ $lt_m(y, k) \geq rt_m(y, k)$;

(b) $lt_m(y, k) = rt_m(y, k)$.

Лемма 2. Пусть *B*-произвольный к.б., $y \in Ls(B)$, $n > 1$, $y= c_1\ldots c_n$, где для $i=1,\ldots,n$ $c_i \in D(B)$, цепочка y включает запятую или какой-либо из символов ≡ , ∧, ∨ , и k - такое произвольное натуральное число, что $1 < k < n$. Тогда:

(а) если c_k - один из символов ≡ , ∧, ∨ , то $lt_1(y, k) > rt_1(y, k) \geq 0$;

(б) если c_k - запятая , то выполняется по крайней мере одно из соотношений $lt_1(y, k) > rt_1(y, k) \geq 0$, $lt_2(y, k) > rt_2(y, k) \geq 0$.

Доказательства Леммы 1, Леммы 2 и Утверждения 3.5. изложены в Приложении к данной книге.

Определение 5. Пусть *B* – произвольный к. б., $z \in Ls(B) \setminus (X(B) \cup V(B))$, и существует такой набор (k ,n, $y_0,\ldots, y_n$), где $1 \leq k \leq 10$, $n > 1$, и $y_1,\ldots, y_n \in Ls(B)$, что

$$y_1 \&\ldots\& y_n \& z \in Ynr_{10}^k(B).$$

Тогда упорядоченный набор вида ($k, n, y_1,\ldots, y_n$) будем называть *формообразующим набором* цепочки z.

С учетом этого определения Утверждение 3.5 говорит о том, что для произвольного к.б. B каждая цепочка $z \in Ls(B)\backslash(X(B) \cup V(B))$ имеет единственный формообразующий набор.

Утверждение 3.6. Пусть B – произвольный концептуальный базис, $z \in Ls(B)$. Тогда существует один и только один такой тип $t \in Tp(S(B))$, что $z \,\&\, t \in Ts(B)$ **Доказательство.**

Рассмотрим два возможных случая.

Случай 1.

Пусть B – произвольный к. б., $z \in X(B) \cup V(B)$, $t \in Tp(S(B))$, $tp(z) = t$. Тогда из правила P[0] следует, что $z \,\&\, t \in Ts(B)$.

Предположим, что w - такой тип из $Tp(S(B))$, что $z \,\&\, w \in Ts(B)$. Анализ правил P[0] – P[10] показывает, что такое соотношение может вытекать только из правила P[0] . Но тогда w однозначно определяется данным правилом, поэтому w совпадает с t .

Случай 2.

Пусть B – произвольный к. б., $z \in Ls(B) \setminus (X(B) \cup V(B))$. В силу Предложения 6, найдутся такие натуральные k, где $1 \le k \le 10$, $n \ge 1$, и такие y_0 , $y_{1,\ldots,}$ $y_n \in Ls(B)$, что цепочка z построена из этих элементов в результате одного применения правила P[k]. Поэтому найдется такой тип $t \in Tp(S(B))$, что $z \,\&\, t \in Tnr_{10}^{k}(B)$, и, следовательно, $z \,\&\, t \in Ts(B)$.

Из Утверждения 3.5 вытекает, что z определяет однозначным образом такие k , n , y_0 , $y_{1,\ldots,}$ y_n . Но тогда набор (k , n , y_0 , $y_{1,\ldots,}$ y_n) однозначно определяет такой тип u , что $z \,\&\, u \in Tnr_{10}^{k}(B)$. Поэтому u совпадает с t .

Таким образом, Утверждение 3.6 говорит о том, что каждой цепочке стандартного К-языка $Ls(B)$, где B - произвольный к.б., можно поставить в соответствие единственный тип t из $Tp(S(B))$.

Определение 6. Пусть B – произвольный к. б., $z \in Ls(B)$. Тогда типом l-формулы z называется такой элемент $t \in Tp(S(B))$, обозначаемый через $tpl(z)$, что $z \,\&\, t \in Ts(B)$. □

Глава 4

ИССЛЕДОВАНИЕ ВЫРАЗИТЕЛЬНЫХ ВОЗМОЖНОСТЕЙ СТАНДАРТНЫХ К-ЯЗЫКОВ

Проведем дополнительный анализ выразительных возможностей стандартных К-языков (СК-языков) по сравнению с анализом возможностей математического описания структурированных значений ЕЯ-текстов, выполненным в предыдущих параграфах**.** Набор примеров, рассмотренных выше, недостаточно полно демонстрирует реальную мощность построенной модели. Поэтому рассмотрим ряд дополнительных примеров, иллюстрирующих некоторые важные возможности СК-языков.

Если цепочка *Expr* некоторого СК-языка является семантическим представлением выражения Т на естественном языке, то такую цепочку *Expr* будем называть возможным К-представлением (КП) выражения Т.

4.1. Удобный способ описания событий

Ключевую роль в формировании предложений играют глаголы и лексические единицы, являющиеся производными от глаголов - причастия, деепричастия и

отглагольные существительные, потому что они выражают разнообразные отношения между объектами рассматриваемой предметной области.

Тематической ролью (концептуальным падежом, семантическим падежом, глубинным падежом, семантической ролью) в компьютерной лингвистике называется смысловое отношение между значением глагольной формы и значением зависящей от нее в предложении группы слов (или отдельного слова).

В таких разных языках, как русский, английский, немецкий и французский языки, можно наблюдать следующую закономерность: в предложениях с одним и тем же глаголом, обозначающим событие, явно реализуется разное количество тематических ролей, связанных со значением данного глагола. Например, пусть Т1 = “Профессор Новиков прилетел вчера” и Т2 = “Профессор Новиков прилетел вчера из Праги”. Тогда в предложении Т1 явно реализуются тематическая роль, которой можно дать название Агент1 (Агент действия), а также тематическая роль Время. При этом в предложении Т2 явно реализуются тематические роли Агент1, Время и Место1 (отношение, связывающее событие перемещения в пространстве и исходный пространственный объект).

Рассмотрим столь же гибкий способ построения СП сообщений о событиях. Для этого потребуется сделать определенное предположение о свойствах рассматриваемого концептуального базиса (к.б.) В.

Предположение 1. Множество сортов *St(B)* включает выделенный сорт *cит* (“ситуация”); множество переменных *V(B)* включает счетное подмножество V_{sit} = *{e1, e2, e3, …},* такое, что для каждого $v \in V_{sit}$ *tp(v) = cит ;* первичный информационный универсум *X(B)* включает бинарный реляционный символ *Ситуация*, такой, что *tp(Ситуация)* - цепочка {(*cит*, $\uparrow$*cит*)}. □

Смысл выражения $\uparrow$*cит* в правой части соотношения *tp(Ситуация)={(cит*, $\uparrow$*cит*)}заключается в том, что мы сможем строить выражения вида *Ситуация* (e_k *, concept),* где e_k – переменная, обозначающая конкретное событие (продажа, покупка, отлет), *concept* – простое или составное понятие, являющееся семантической характеристикой события.

Уточним, что связь между меткой ситуации и видом ситуации будет осуществляться с помощью формул вида *Ситуация (v, conc * (r_1, d_1)…(r_n, d_n)),*

где *v* – переменная типа *сит*, *conc*$\in$*X(B)*, *conc* интерпретируется как понятие, характеризующее ситуацию, $n \geq 1$, для $i=1,\ldots, n$ r_i – характеристика ситуации, d_i – значение характеристики.

Пример. Пусть *Expr1 = ∃ e1(сит) (Ситуация(e1, прилет * (Время, x1)(Агент1, нек чел * (Квалиф, профессор)(Фамилия, 'Новиков') : x2)) ∧ Раньше (x1, Сейчас))* ,

*Expr2 = ∃ e1(сит) (Ситуация(e1, прилет * (Время, x1)(Агент1, нек чел * (Квалиф, профессор)(Фамилия, 'Новиков') : x2)(Место1, нек город * (Название, 'Прага') : x3)) ∧ Раньше (x1, Сейчас))* .

Тогда легко видеть, что можно построить такой к.б. B, что для него будет справедливо Предположение 1 и выполнено соотношение *B(0, 1, 2, 3, 4, 5,7, 8, 9) => Expr1, Expr2 ∈ Ls(B), Expr1 & сообщ ∈ Ts(B), Expr2 & сообщ ∈ Ts(B)* ,

где *сообщ = P(B)* – выделенный сорт "смысл сообщения" базиса *B*.

Данный способ описания сообщений будет многократно использован в этом и последующих параграфах. При этом чаще всего, по соображениям компактности, будет опускаться квантор существования при переменной, обозначающей событие. Например, вместо формулы *Expr1* будет рассматриваться формула *Expr3* вида

*(Ситуация(e1, прилет * (Время, x1)(Агент1, нек чел * (Квалиф, профессор)*

(Фамилия, 'Новиков') : x2)) ∧ Раньше (x1, сейчас)) .

4.2. Формализация предположений о структуре семантических представлений множеств

Сообщения, вопросы, команды могут включать обозначения множеств. Для обеспечения единства подхода в разных ситуациях (при рассмотрении сообщений, команд, вопросов) к построению СП описаний множеств целесообразно ввести ряд дополнительных предположений об используемых концептуальных базисах.

В связи с тем, что обозначения множеств в текстах часто включают количественные числительные или обозначения натуральных чисел ("два

алюминиевых контейнера" и т.п.), будем полагать, что для рассматриваемого к.б. В справедливо

Предположение 2. Множество сортов *St(B)* включает выделенный сорт *нат* ("натуральное число"), первичный информационный универсум *X(B)* включает подмножество цепочек *Nt*, такое, что $Nt = \{ d_1 \ldots. d_k \mid k \geq 1$, для $i = 1,.., k$ d_i – символ из множества {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'}, и из d_1 = '0' следует, что $k = 1\}$. При этом для каждого $z \in Nt$ $tp(z) = нат$.

Потребуем также, чтобы первичный информационный универсум *X(B)* включал выделенные элементы *множ*, *Колич*, *Кач-состав*, *Предм-состав*, интерпретируемые следующим образом: *множ* – это обозначение понятия "конечное множество", *Колич* – имя одноместной функции "Количество элементов множества", *Кач-состав* – имя бинарного отношения "Качественный состав множества", *Предм-состав* – имя бинарного отношения "Предметный состав множества".

Предположение 3. Первичный информационный универсум *X(B)* включает элементы *множ, Колич, Кач-состав, Предм-состав,* такие, что *tp(множ) = ↑{[сущн]}, tp(Колич) = {({[сущн]},нат)} , tp(Кач-состав) = {({[сущн]}, [пон])}, tp(Предм-состав) = {({[сущн]}, [сущн])}* , где *[сущн], [об], [пон]* – базовые типы “сущность” , “объект”, “понятие” (см. параграф 1.6).

Рассмотрим назначение выделенных элементов универсума *X(B)*, упоминаемых в Предположении 3.

Используя элементы *множ, Колич* и произвольную цепочку *numb* из *Nt*, мы сможем построить СП выражения "некоторое множество, содержащее *numb* элементов" в виде *нек множ * (Колич, numb)*, где *нек = ref(B)* – квантор референтности рассматриваемого концептуального базиса *B*.

Назначение бинарного реляционного символа *Кач-состав* заключается в следующем. Пусть *v* – переменная, обозначающая некоторое множество, и *conc* – простое или составное обозначение понятия. Тогда выражение *Кач-состав (v, conc)* обозначает высказывание "Каждый элемент множества *v* квалифицируется понятием *conc*", и это высказывание может быть истинным или ложным. Примерами выражений этого вида являются *Кач-состав(S1, контейнер1)*,

*Кач-состав(S2, статья1), Кач-состав(S3, контейнер1 * (Материал, алюминий)), Кач-состав(S4, статья1 * (Область1, биология)).*

С другой стороны, символ *Кач-состав* будет применяться и при построении составных обозначений множеств вида *ref множ * (Кач-состав, conc) : v,*
где *ref* – квантор референтности, *conc* – простое или составное обозначение понятия, *v* – переменная. В частности, концептуальный базис *B* можно выбрать так, чтобы язык *Ls(B)* включал выражения

*нек множ * (Кач-состав, контейнер1) : S1, нек множ * (Кач-состав, статья1) : S2, нек множ * (Кач-состав, контейнер1 * (Материал, алюминий)) : S3, нек множ * (Кач-состав, статья1 * (Область1, биология)) : S4.*

Фрагмент текста, обозначающий множество, может представлять собою явное перечисление элементов множества. Таким, в частности, является текст "Два заказчика, АО "Радуга" и ТОО "Зенит", не оплатили сентябрьские поставки".

Бинарный реляционный символ *Предм-состав* предназначен, в частности, для построения выражений вида *Предм-состав (v, ($x_1 \wedge x_2 \wedge \ldots \wedge x_n$))*,
где *v*, x_1 , x_2 ,…, x_n – переменные, причем *v* обозначает множество, а x_1 ,…, x_n – это обозначения всех элементов, входящих в состав множества *v*.

Например, будем считать, что выражение *(Предм-состав (y1, ($x_1 \wedge x_2$)) ∧ Явл (x1, АО) ∧ Явл (x2, ТОО) ∧ Имя (x1, "Радуга") ∧ Имя (x2, "Зенит"))* является СП высказывания "Множество y1 состоит из АО "Радуга" и ТОО "Зенит", причем первая организация обозначается через x1, а вторая – через x2.

В то же время мы должны иметь возможность (если это необходимо) строить формулы вида *ref множ * (Предм-состав, ($x_1 \wedge x_2 \wedge \ldots \wedge x_n$)) : y1,* где *ref* – квантор референтности, и *y1, x1, x2* – переменные типа *[сущн]* (базовый тип "сущность"). Например, мы должны располагать возможностью построения выражения *нек множ * (Предм-состав, (x1 ∧ x2)) : y1* .

Пример. Рассмотрим выражения Вр1 = "3 контейнера с керамикой из Индии" и Вр2 = "Партия керамики, состоящая из коробок с номерами 3217, 3218, 3219". Тогда можно построить такой к.б. *B,* для которого будут выполнены Предположения 2, 3, и *Ls(B)* включает формулы
*(1) нек множ. * (Колич, 3) (Кач-состав, Контейнер 1 * (Содерж1, нек множ **

*(Кач-состав, изделие * (Вид, керамика) (Страна, Индия)))),*

*(2) (нек партия2 * (Колич, 3)(Предм-состав, (нек коробка1 * (Номер, 3217) : x1 ∧ нек коробка1 * (Номер, 3218) : x2 ∧ нек коробка1 * (Номер, 3219) :x3)) :S1 .*

Построенные формулы будем интерпретировать как возможные КП выражений Вр1 и Вр2; здесь *x1, x2, x3* – метки коробок, *S1* – метка партии.

4.3. Построение семантических представлений вопросов с ролевыми вопросительными словами

Среди всех вопросительных местоимений и наречий можно выделить подмножество, включающее, в частности, слова “кто”, “что”, “кому”, “чем”, “когда”, “откуда”. Чтобы сформулировать свойство, выполняющееся для каждого элемента этого подмножества, введем обозначение *nil* для пустого предлога. Если в каком-либо вопросе некоторое вопросительное местоимение *qswd* употреблено без предлога, то условимся говорить, что этому местоимению *qswd* в данном предложении соответствует пустой предлог.

Для каждого местоимения qswd из рассматриваемого подмножества найдется предлог prep (возможно, он не является единственным), что паре (prep, qswd) соответствует некоторая тематическая роль *role*. Например, парам *(nil, кому), (для, кого), (от, кого),* могут соответствовать тематические роли *Адресат, Адресат, Источник1*. Таким образом, разным парам *(nil, кому), (для, кого)* соот-ветстветствует одна тематическая роль *Адресат*, поскольку в равной степени пра-вильными являются фразы “Кому прислана книга?” и “Для кого прислана книга ?”.

Местоимения и наречия, входящие в указанное подмножество, будем называть *ролевыми вопросительными словами.*

Предположение 4. Первичный информационный универсум *X(B)* концептуального базиса *B* включает символ *Вопрос*, и *tp(Вопрос) = {([сущн], P)}* , где *tp = tp(B)* - отображение, задающее тип информационной единицы, *[сущн]* – базовый тип “сущность”, *P = P(B)* – выделенный сорт “смысл сообщения”.

Пусть для к.б. *B* выполнено Предположение 4. Тогда СП вопроса с *n* ролевыми вопросительными словами можно представить в виде *Вопрос* (v_1 , *A*) при *n = 1* и в виде *Вопрос* (($v_1 \wedge .... \wedge v_n$), *A*) при *n > 1* , где *A* - формула, зависящая от переменных v_1 , ... , v_n и отображающая содержание высказывания (т.е. являющаяся семантическим представлением высказывания).

Пример 1. Пусть B1 = “Откуда поступил трехтонный алюминиевый контейнер?”, *Expr1* – цепочка вида *Вопрос (x1 , (Ситуация (e1, поступление2 * (Объект1, нек контейнер * (Вес, 3/тонна)(Материал, алюминий) : x2)(Место1, x1)(Время, t1)) ∧ Раньше(t1, сейчас)))* . Тогда нетрудно построить такой к.б. *B*, что для *B* выполняются Предположение 1 и Предположение 4, *P(B)* = сообщ, и *B (0, 1, 2, 3, 4, 5, 7, 8) => Expr1 ∈ Ls(B) , Expr1& сообщ ∈ Ts(B)* .

Цепочка *Expr1* является возможным КП вопроса B1. В этой цепочке символы *x1, x2, e1, t1* являются переменными, *поступление2* - информационная единица (другими словами, семантическая единица), соответствующая существительному “поступление” и передающая значение “перемещение некоторого физического объекта на пространственный объект” (в отличие от значения “поступление абитуриента в учебное заведение”).

Пример 2. Пусть B2 = “Откуда и когда поступил трехтонный алюминиевый контейнер?”. Тогда КП вопроса B2 может являться выражением *Вопрос ((x1 ∧ t1) , (Явл1 (e1, поступление2 * (Объект1, нек контейнер * (Вес, 3/тонна)(Материал, алюминий) : x2)(Место1, x1)(Время, t1)) ∧ Раньше(t1, Сейчас)))* .

4.4. Семантические представления вопросов о количестве предметов и о количестве событий

Предположение 5. Первичный информационный универсум *X(B)* вида *(S, Ct, Ql),* где *Ql* – система кванторов и логических связок вида *(ref,* int_1*,* int_2*, eq, neg, binlog, ext)* , включает элементы *произв, все, Элем*, такие, что *tp(произв) =* int_1*,* *tp(все) =* int_2*, tp(Элем) = {([сущн], {[сущн]})}* .

Ээлементы *произв, все, Элем* интерпретируются как информационные единицы “произвольный” (“каждый”), “все” и “Элемент множества” (имя отношения “Быть элементом множества”).

Следует заметить, что int_1 и int_2 — это выделенные элементы множества сортов *St(B).* По определению (см. параграф 2.8), элементы int_1 и int_2 являются типами интенсиональных кванторов соответственно первого и второго видов.

Пример 1. Пусть В1= «Сколько экземпляров книг А.П.Сомова имеется в библиотеке?». Тогда можно определить такой к.б. *B*, что для *B* выполняются Предположение 4 и Предположение 5, и цепочка

Вопрос(x1, (x1≡ Колич (все экземпляр1(Информ-объект, произв книга * (Автор, нек чел* (Инициалы, ‘А.П.’)(Фамилия, ‘Сомов’) : x2) : x3) (Место-хранения, нек библиотека: x4))))*

входит в *Ls(B).* Поэтому данное выражение является возможным К-представлением вопроса В1.

Пример 2. Если В2 = «Сколько человек участвовало в создании статистического сборника?», то возможным К- представлением В2 является выражение

Вопрос(x1, ((x1≡ Колич(все чел(Элем, S1))) ∧ Описание(произв чел* (Элем, s1) : y1, (Ситуация(e1, участие1* (Агент1, y1)(Время, x2) (Вид-деятельности, создание1* (Продукт1, нек сборник1* (Область1, статистика)))) ∧ Раньше(x2, #Сейчас#))))) .*

Пример 3. Пусть В3 = «Сколько книг поступило в январе этого года в библиотеку № 18?». Тогда возможным К-представлением В3 является формула

*Вопрос(x1, ((x1≡ Колич (все книга * (Элем, S1))) ∧ Описание (произв книга* (Элем, S1) : y1, (Ситуация(e1, поступление2* (Объект1, y1)(Время, <01, текущий- год>)(Место2, нек библиотека * (Номер, 18) : x2)))) .*

Пример 4. Вопрос В1= «Сколько раз Иван Михайлович Семёнов летал в Мексику?» может иметь следующее возможное КП:

*Вопрос(x1, (x1≡ Колич (все полёт *(Агент1, нек чел * (Имя, ‘Иван’) (Отчество, ‘Михайлович’)(Фамилия, ‘Семёнов’): x2)(Место2, нек страна* (Название, ‘Мексика’):x3)(Время, произв момент *(Раньше, #сейчас#))))) .*

4.5. Семантические представления вопросов с формами вопросительно-относительного местоимения “какой”

Метод, предложенный выше для построения К-представлений вопросов с ролевыми вопросительными словами, можно использовать и для построения КП вопросов с различными формами местоимения “какой”.

Пример 1. Пусть В1= «Какое издательство опубликовало роман «Ветры Африки»?». Тогда КП вопроса В1 может являться цепочкой

*Вопрос(x1, (Ситуация(e1, опубликование * (Время, x2) (Агент2, нек издательство: x1) (Объект3, нек роман1* (Название, ‘Ветры Африки’) :x3)) ∧ Раньше(x2, #сейчас#)))* .

Пример 2. Пусть В2= «С какими зарубежными издательствами сотрудничает писатель Игорь Сомов?». Тогда КП В2 может являться формулой

*Вопрос (S1, (Кач-состав (S1, издательство * (Вид-географич, зарубежное))*

∧ Описание(произв издательство (Элем, S1) : y1, Ситуация(e1, сотрудничество * (Агент1, нек чел* (Профессия, писатель)(Имя, ‘Игорь’)(Фамилия, ‘Сомов’): x1)(Организация1, y1)(Время, #сейчас#)))))* .

4.6. Построение семантических представлений вопросов общеудостоверительного актуально-синтаксического типа

Вопросами общеудостоверительного актуально-синтаксического типа в лингвистике называются вопросы с ответом “Да“ или “Нет“. Такие вопросы задаются для того, чтобы в целом удостовериться в правильности имеющейся у спрашивающего информации (Воробьева, Панюшева, Толстой 1975).

Оказывается, что предложенную выше форму отображения смысла вопросов с ролевыми вопросительными словами можно использовать и для построения СП общих вопросов. Для этого каждый такой вопрос будем интерпретировать как просьбу указать истинное значение некоторого высказывания. Например, вопрос В1 = "Является ли Гент городом Бельгии" можно интерпретировать как просьбу найти истинностное значение высказывания "Гент является одним из городов Бельгии". Для реализации этой идеи введем

Предположение 6. *St(B)* включает выделенный сорт *лог*, называемый "логическая величина"; *X(B)* включает различные элементы *ист*, *ложь*, причем *tp(ист) = tp(ложь) = лог* , *F(B)* включает одноместный функциональный символ *Ист-знач*, такой, что *tp(Ист-знач) = {(P, лог)}* , где *P = P(B)* – выделенный сорт "смысл сообщения".

Пример 1. Если для рассматриваемого концептуального базиса B выполняется указанное предположение, то КП вопроса B1 = "Является ли Гент городом Бельгии?" может являться формулой

*Вопрос(x1, (x1 ≡ Ист-знач (Элем(нек город * (Назв, "Гент") : x2,*

*Города(нек страна * (Назв, "Бельгия") : x3))))).*

В этой формуле символ *Города* интерпретируется как имя одноместной функции, ставящий в соответствие стране множество всех городов этой страны, *нек* – квантор референтности *ref(B); x1, x2, x3∈ V(B).*

Пример 2. Пусть B2= «Проходила ли в Азии международная научная конференция «COLING»?». Тогда К-представлением вопроса B2 может являться формула

Вопрос(x1, (x1≡ Ист-знач ((Ситуация (e1, прохождение2(Событие, нек конф* (Вид1, междун) (Вид2, научн) (Название, 'COLING') : x2) (Место, нек континент* (Название, 'Азия') :x3) (Время, x4)) ∧ Раньше(x4, #сейчас#)))) .*

4.7. Отображение смысловой структуры команд

Будем использовать две основные идеи. Во-первых, когда мы говорим о команде (или о приказе, распоряжении и т.д.), то всегда подразумеваем, что имеется одна интеллектуальная система, формирующая команду (обозначается выражением *#Оператор#),* и другая интеллектуальная система (или же конечное множество интеллектуальных систем), которая должна выполнить команду (обозначается выражением *#Исполнитель#).* Во-вторых, глагол в повелительном наклонении или неопределенную форму глагола будем заменять соответствующим отглагольным существительным.

Предположение 7. Множество сортов *St(B)* включает выделенные элементы *интс* (сорт "интеллектуальная система"), *мом* (сорт "момент времени");

первичный информационный универсум *X(B)* включает элементы *Команда, #Оператор#, #Исполнитель#, #сейчас#,* такие что *tp(Команда*) = {(*интс, интс, мом, ↑сит*)}, *tp(#Оператор#) = tp(#Исполнитель#) = интс, tp(#сейчас#) = мом*.

Пример. Пусть К1 = "Доставь ящик с деталями на склад № 3", где К1 - команда, отданная оператором гибкой производственной системы интеллектуальному транспортному роботу. Тогда базис В можно определить так, чтобы выполнялись Предположение 1, Предположение 7 и соотношение

B(0, 1, 2, 3, 4, 5, 8)=> Команда(#Оператор#, #Исполнитель#, #сейчас#, доставка1(Объект1, нек ящик * (Содерж1, нек множ * (Кач-состав, деталь)) : x1)(Место2, нек склад * (Номер, 3) : x2)) ∈ Ls(B).*

4.8. Представление теоретико-множественных отношений и операций на множествах

Пример 1. Пусть Т1а = "Намюр – один из городов Бельгии". Тогда рассмотрим текст Т1б = "Намюр входит в множество всех городов Бельгии".

Пусть *E1 = Элем (нек город * (Назв, 'Намюр') : x1, Города(нек страна * (Назв, 'Бельгия') : x2))* . Тогда ∃ такой к.б. *B1*, что *B1(0, 1, 2, 3, 8, 1, 5, 0, 1, 2, 3, 8, 1, 5, 2, 4) => E1 ∈ Ls(B1).* При построении нужно предполагать, что *Элем, страна, город, нек ∈ X(B), tp(Элем) = {([сущн], {[сущн]})}, tp(страна) = tp(город)= ↑ простр.об; Города ∈ F1(B1), tp(Города) = {(простр. об, {простр. об})}, нек = ref(B)* – квантор референтности базиса *B.*

Пример 2. Пусть Т2а = "Включи контейнер № 4318 в партию, отправляемую в Тамбов". Преобразуем Т2а в Т2б ="Некоторый оператор распорядился включить контейнер № 4318 в некоторую партию, отправляемую в город Тамбов". Тогда построим КП текстов Т2а и Т2б в виде

Команда((#Оператор#, #Исполнитель#, #сейчас#, включение1 (Объект1, нек контейнер * (Номер, 4318) : x1) (Целевое.множество, нек партия2 * (Место-назн, нек город*(Название, 'Тамбов') : x2) : S1)),*

где *S1* - метка партии продукции. Аналогично можно представить распоряжения о разделении множества объектов на несколько частей и об объединении

нескольких множеств в одно, например, при перегрузке деталей из нескольких ящиков в один.

4.9. Представление смысла фраз с придаточными предложениями цели и с косвенной речью

Пример 1. Пусть T1 = "Сергей поступил в МИЭМ, чтобы получить специальность "Прикладная математика" ", и *Sr1 = (Ситуация (e1, поступление1* (Агент, нек чел* (Имя, "Сергей") :x1)(Уч.заведение, нек вуз * (Название, 'МИЭМ') : x2)(Время, t1) (Цель, получение1* (Квалификация, нек специальность*(Название, 'прикладная математика') : x3)) ∧ Раньше(t1, #сейчас#)))).* Тогда ∃ такой.к.б. *B*, что *B(0, 1, 2, 3, 4, 5, 7, 8) => Sr1 ∈ Ls(B), Sr1 & Сообщ ∈ Ts(B).*

Пример 2. Пусть T2 = "Директор сказал, что на февраль запланирована реорганизация фирмы" и

*Sr2 = (Ситуация (e1, устное-сообщение * (Агент1, Директор(нек организация: x1)) (Время, t1)(Содержание1, Планируется(нек реорганизация* (Объект2, нек фирма: x1), Ближайший-месяц(февраль, t1)))) ∧ Раньше(t1, #сейчас#)) .*

Тогда легко построить такой к.б. *B*, что выполняются соотношения

B(0,2) => Ближайший-месяц(февраль, t1) ∈ Ls(B), Ближайший-месяц(февраль, t1) & врем.интервал ∈ Ts(B) ; B(0,1,2,4,5,7, 8) => Sr2∈ Ls(B), Sr2 & сообщ ∈ Ts(B).

4.10. Явное представление причинно-следственных отношений, передавамых дискурсами

Как уже отмечалось выше, в компьютерной и теоретической лингвистике дискурсом, или связным текстом, называется последовательность взаимосвязанных по смыслу предложений (полных или неполных). Соответствие между группами слов из текста и теми объектами, событиями, процессами, смыслами, которые эти группы слов обозначают, называется референтной структурой текста. СК-языки предоставляют широкие

возможности описания смысловой структуры дискурсов, в том числе их референтной структуры.

Пример. Пусть Т1 = "Первокурсник Петр Сомов не заметил, что расписание изменилось, поэтому он пропустил первую лекцию по линейной алгебре". В этом тексте проявляется, в частности, следующая особенность дискурсов: личное местоимение "он" используется вместо более длинного сочетания "первокурсник Петр Сомов". Говорят, что у этого последнего выражения и местоимения "он" есть один и тот же референт - некоторый человек, студент вуза.

Чтобы явно указать референтную структуру текста, нужно связать метки с сущностями, обозначаемыми некоторыми группами слов из этого текста или неявно упоминаемыми в тексте. Сделаем это таким образом: неявно упоминаемое учебное заведение – метка *x1*; “Первокурсник Петр Сомов”, “он” - метка *x2*; “Расписание” - метка *x3*; “первую лекцию по линейной алгебре” - метка *x4*; “не заметил” - метка *e1* (событие); “изменилось” - метка *e2*; *e3* – метка ситуации, описываемой первым предложением из Т1; “пропустил” - метка *e4* (событие). Будем полагать, что СП текста Т1 должно включать фрагмент *Причина(e3, e4).* Пусть

*Sr1 = ((Ситуация (e1, ¬ обращение-внимания * (Агент1, нек чел*(Имя, 'Петр')(Фам, ‘Сомов’)(Квалиф, студент*(Курс,1)(Уч-заведение, x1)) : x2)(Время, t1)(Объект-внимания, e2)) ∧ Раньше(t1, #сейчас#) ∧ Ситуация (e2, изменение * (Предмет, нек расписание : x3)(Время, t2)) ∧ Раньше(t2, t1)) : P1 ∧ Характеризует(P1, e3)).*

Тогда *Sr1* - возможное КП первого предложения П1 дискурса Т1.

Пусть *Sr2 = ((Ситуация (e4, пропуск1 * (Агент1, x2)(Объект3, нек лекция * (Дисциплина, лин-алгебра)(Уч-заведение, x1) : x4)(Время, t3)) ∧ Раньше(t2, #сейчас#).* Тогда *Sr2* - возможное КП второго предложения П2 из дискурса Т1.

. Пусть *Srd1 = (Sr1∧ Sr2 ∧ Причина(e3, e4)).* Тогда *Srd1* - возможное СП дискурса Т1, являющееся К-представлением текста Т1.

4.11. Построение семантических представлений дискурсов со ссылками на смысл фраз и более крупных частей текста

Пример. Пусть Т1 = "АО "Радуга" подпишет контракт до 15 декабря. Об этом сообщил заместитель директора Игорь Панов". Здесь сочетание "об этом" обозначает ссылку на смысл первого предложения дискурса Т1. Пусть

*Sr1 = (Ситуация (e1, подписание1 * (Агент1, нек организация * (Тип, АО) (Название, "Радуга"): x1)(Время, t1)(Объект3, нек контракт1: x2)) ∧ Раньше(t1, 15/декабрь/текущий-год))* ,

Srd2 = (Sr1 : P1 ∧ Ситуация (e2, сообщение1 (Агент1, нек чел * (Имя, 'Игорь')(Фамилия, 'Панов') : x3)(Время, t2)(Содержание2, P1)) ∧ Раньше (t2,, #сейчас#) ∧ Зам.директора(x3, нек орг : x4)))*.

Тогда найдется такой к.б. *B*, что *B(0,1, 2, 3, 4, 5, 7, 8) => Srd2 ∈ Ls(B), Srd2 & сообщ ∈ Ts(B)*.

Правило Р[5] позволяет приписать переменную *v* к СП *Sr* произвольного повест-вовательного текста и получить формулу *Sr : v* , где *v* - произвольная переменная сорта *P(B)* - сорта "смысл сообщения". Поэтому выражениям "об этом", "этот метод", "этот вопрос" и т.д. будет соответствовать переменная *v* сорта *P(B)* в СП всего дискурса (так же, как и в последнем примере).

4.12. Представление фрагментов знаний о мире

Пример 1. Пусть Т1 = "Понятие "молекула" используется в физике, химии, биологии." Можно определить такой к.б. *B*, что множество сортов *St(B)* включает элемент *область1* и первичный информационный универсум *X(B)* включает элементы *область1, цепочка, понятие, "молекула", Использ, Имя-понятия, физика, нек., химия, биология,* причем типы этих элементов задаются соотношениями

tp(понятие) = [↑пон], tp("молекула") = цепочка, tp(физика) = tp(химия) = tp(биология) = область1, tp(Использ) = {([пон], область1)}, tp(Имя-понятия) = {([пон], цепочка)}.

Пусть *нек* — квантор референтности базиса *B*, *Использ* и *Имя-понятия* — бинарные реляционные символы, не являющиеся именами функций, и

s_1 = *Имя-понятия (нек понятие, “молекула”),*

s_2 = *понятие * (Имя-понятия, “молекула”),*

s_3 = *Использ (нек понятие * (Имя-понятия, “молекула”),*

(физика ∧ химия∧ биология)).

Тогда *B(0, 1, 4) => s_1∈Ls(B); B(0, 1, 4, 8) => s_2∈Ls(B); B(0, 1, 4, 8, 1, 0, 7, 4) => s_3∈Ls(B).* Построенная формула s_3 является возможным КП для определения *T1*.

Пример 2. Пусть T2 = “Тинейджер — это человек в возрасте от 12 до 19 лет”; s –цепочка *((тинейджер ≡ человек * (Возраст, x1)) ∧ ¬ Меньше(x1, 12/год) ∧ ¬ Больше(x1, 19/год)).* Тогда s — возможное КП для T2.

Пример 3. В работе (Nebel, Peltason 1991) сформулировано определение: “Малое и среднее предприятие (*sme*) — это компания с числом служащих не более 50”. Это определение может иметь, в частности, следующие К-представления:

Определение(sme, ∀x1(компания1)(Явл1(x1, sme) ≡

¬Больше(Колич (Персонал (x1)), 50))) ,

((sme ≡ компания1(Описание, P1))∧(P1 ≡ ∀x1(компания1)*

(Явл1(x1, sme) ≡ ¬Больше(Колич(Персонал (x1)), 50)))) .

4.13. Объектно-ориентированные представления фрагментов знаний

Используя стандартные К-языки, мы можем строить сложные описания объектов и множеств объектов. Например, мы можем построить следующее К-представление описания международного журнала “Informatica”:

нек межд-науч-журнал (Название, 'Informatica’)(Страна, Словения)*

(Город, Любляна) (Области, (иск-интеллект ∧ когнитивная-наука

∧ базы-данных)) : k225 ,

где *k225* — метка модуля знаний с данными об этом журнале.

Постановка задачи, изложенная в параграфе 3.1, предусматривает возможность строить с помощью новых формальных средств концептуальные

представления текстов как информационные объекты, отражающие не только смысл, но и значения внешних характеристик текста (метаданные): авторов, дату, области применения изложенных результатов и т. д.

Пример. Используя идею построения К-представлений разнообразных объектов, проиллюстрированную на примере модуля знаний с данными о журнале "Informatica", мы можем построить модуль знаний, содержащий теорему Пифагора и указывающий ее автора и предметную область. Например, подобный модуль может быть следующим выражением некоторого СК-языка:

нек информ-объект (Вид, теорема)(Область, геометрия)(Автор, Пифагор)*
(Содержание, ∀x_1(геом) ∀x_2(геом) ∀x_3(геом) ∀x_4(геом)Если-то((Явл(x_1, прямоугольн) ∧ Гипотенуза(x_1, x_2)∧ Катет((x_3∧x_4), x_1)),
(Квадрат(Длина(x_2)) ≡Сумма(Квадрат(Длина(x_3)), Квадрат(Длина(x_4)))))) : k81.

4.14. Сравнение выразительных возможностей СК-языков с возможностями основных известных подходов к формальному представлению содержания ЕЯ-текстов

4.14.1. Сравнение с основными подходами, разработанными в нашей стране

Основными средствами формального представления содержания ЕЯ-текстов, разработанными в нашей стране и использовавшимимя в 1990-е – 2000-е годы для проектирования лингвистических процессоров, являются, помимо предложенных автором данной дисертации стандартных К-языков (СК-языков), расширенные семантические сети (Кузнецов 1976 - 1989; Кузнецов и др. 2000; Кузнецов, Мацкевич 2001, 2003; Кузнецов, Шарнин 2003; Kuznetsov, Matskevich 2002; Соловьева, Сомин 1993), формальные выражения, предоставляемые компьютерной семантикой русского языка, и неоднородные семантическте сети (Осипов 1990, 1997).

Расширенную семантическую сеть (РСС) можно представить как конечное множество выражений вида $R(c_1, c_2, \ldots, c_n, d)$ **,** где $n \geq 1$, R – имя n-арного отноше-ния, $c_1, c_2, \ldots, c_n$ - атрибуты отношения R , d – метка, являюшаяся уникальным именем (в рамках рассматриваемой базы данных) выражения $R(c_1,$

c_2, …, c_n) . Выражения вида $R(c_1, c_2, \ldots, c_n, d)$ называются элементарными фрагментами (ЭФ).

Каждый ЭФ вида $R(c_1, c_2, \ldots, c_n, d)$ можно аппроксимировать выражением некоторого СК-языка $R(c_1, c_2, \ldots, c_n) : d$, где при построении формулы $R(c_1, c_2, \ldots, c_n)$ на последнем шаге применялось правило P[4] , а формула $R(c_1, c_2, \ldots, c_n) : d$ построена в результате применения правила P[5] к операндам $R(c_1, c_2, \ldots, c_n)$ и d , причем d является переменной.

Использование элементарных фрагментов обеспечивает большую однородность представления информации в виде РСС, что создает предпосылки для унификации процедур обработки знаний, представленных с помощью РСС. Однако принципиальным недостатком использования РСС для отображения содержания текстов является огромный разрыв между структурой ЕЯ-текстов и структурой их семантических представлений. В связи с этим, располагая только аппаратом РСС, разработчик семантико-синтаксического анализатора ЕЯ-текстов остается один на один с многочисленными проблемами алгоритмизации перехода от ЕЯ-текста к его семантическому представлению.

Исходный запас идей для развития компьютерной семантики русского языка (КСРЯ) был изложен в 5-й главе монографии (Тузов 1984). В последующие годы эти идеи получили развитие, в частности, в публикациях (Тузов 2001; Каневский, Тузов 2002; Лезин , Тузов 2003).

Центральная идея КСРЯ заключается в следующем. Каждое слово русского языка (РЯ) интерпретируется как название (имя) функции, связанной с этим словом и называемой его семантикой. Семантическое представление предложения является суперпозицией функций, причем в качестве аргументов исходных функций берутся обозначения понятий (называемые лексемами).

Анализ публикаций по КСРЯ позволяет заметить, что более естественно было бы говорить в таких ситуациях о суперпозиции функций и отношений. Например, выражения *Caus(x, y)* (x является причиной события y), *Loc(x, y)* (x расположен в y) наиболее естественно рассматривать как атомарные формулы, в которых элементы *Caus*, *Loc* интерпретируются либо как имена бинарных отношений, либо как имена бинарных предикатов.

Выражения семантического языка КСРЯ можно аппроксимировать выражениями СК-языков, полученными применением правил Р[2] и Р[4], предназначенных для использования имен функций и имен n-арных отношений ($n \geq 1$), к исходным цепочкам вида *ref concept* , где *ref* – квантор референтности (информационная единица, соответствующая слову “некоторый”), *concept* – обозначение понятия из первичного информационного универсума.

Например, в работе (Тузов 2001) семантическое представление предложения “Собака охраняет кофейную плантацию” является выражением *Oper09_a1(СОБАКА$14224112~!%1,ОХРАНА$182036(Rel_o1(ПЛАНТАЦИЯ$12411~!%1,КОФЕ$14/112))).*

К-представлением этого предложения может быть, например, выражение *∃ e1(ситуация) Является (e1, охрана1 * (Агент1, нек собака)(Объект1, нек плантация * (Растения, нек множ * (Кач-состав, дерево * (Вид, кофейн))))))* .

СК-языки удобны и для построения шаблона семантической модели предложения (см. Лезин, Тузов 2003). Например, шаблон семантической модели предложения “Вручая книгу, старик окинул мальчика быстрым оценивающим взглядом” является последовательностью формальных выражений, включающей выражения *КНИГА $14110 : X1 , СТАРИК $12411 : X2, ВРУЧЕНИЕ $15210 : X3 (Oper, СУБЪЕКТ.X2, ОБЪЕКТ.X1, АДРЕСАТ.Z3).*

Эти выражения можно аппроксимировать К-цепочками *Явл(X1 , книга), Явл (X2, старик) , Явл (e1 , вручение1 * (Агент, X2)(Объект1, X1)(Адресат, Z3)).*

Аппарат СК-языков, предложенный в данной книге, обладает следующими основными преимуществами по сравнению с КСРЯ:

(а) ориентирован на построение СП не только отдельных предложений, но и связных текстов (в частности, позволяет отображать ссылки на смысл фраз и более крупных фрагментов дискурса); (б) позволяет строить формальные аналоги сложных составных обозначений понятий и множеств; (в) предоставляет возможность отображения смысловой структуры предложений со словом “понятие”, что необходимо для формального представления информации энциклопедического характера; (г) позволяет строить формальные представления составных целей; (д) конструктивно отражает существование

нескольких дополнительных способов использования логических связок в русском, английском и многих других языках по сравнению с языком логики предикатов.

Этими же основными преимуществами аппарат СК-языков обладает и по сравнению с аппаратом неоднородных семантических сетей (Осипов 1990, 197).

Таким образом, аппарат СК-языков значительно расширяет возможности формального отображения содержания ЕЯ-текстов по сравнению с КСРЯ и по сравнению с аппаратом неоднородных семантических сетей.

Обсуждавшиеся выше аппарат расширенных семантических сетей, компьютерная семантика русского языка и аппарат неоднородных семантических сетей. нашли применение в нескольких проектах разработки ЛП.

Совсем недавно, в последние три года, в публикациях С.В. Елкина и С.С. Елкина был предложен принципиально иной подход к построению семантических языков, истоки которого лежат не в области проектирования ЛП, а в философии. Этот подход рассматривает проблему формализации семантики понятий. Для решения этой проблемы предложен т.н. открытый семантический язык SL (Елкин 2003), построенный на основе информационного исчисления (Елкин С.В, Елкин С.С 2002а, 2002б).

Если не анализировать математическую сущность данного подхода к моделированию семантики понятий, то некоторые суждения из перечисленных непосредственно выше работ могут создать впечатление, что открытый семантический язык SL может широко использоваться для отображения смысла ЕЯ-текстов в компьютерных интеллектуальных системах. Например, в работе (Елкин С.В, Елкин С.С 2002б) отмечается, во-первых, что “наиболее полно проблема понимания текста на естественном языке может быть разрешена при помощи семантического языка” (с. 97). Во-вторых, что ряд особенностей универсального сетевого языка UNL (см. параграф 1.1) требуют его доработки или принципиальных изменений, эти особенности явились источником идей для создания языка SL.

Поскольку язык UNL разрабатывался в качестве языка-посредника для устранения языкового барьера между пользователями сети Интернет из разных стран мира, можно, на первый взгляд, сделать умозаключение о больших

выразительных возможностях открытого семантического языка SL с точки зрения построения семантических представлений предложений и дискурсов на ЕЯ. Однако внимательный анализ определения семантического языка SL показывает, что такой вывод был бы ошибочным: в действительности выразительные возможности семантического языка SL являются весьма ограниченными.

Представляется, что глобальной причиной этого является отсутствие даже в постановке задачи исследования закономерностей организации поверхностной и смысловой структуры предложений и дискурсов из широко распространенных естественных языков. Уровень рассмотрения проблематики является философским. Например, в работе (Елкин С.В 2003) отмечается, что “внешние отношения ПРИТЯЖЕНИЯ и ОТТАЛКИВАНИЯ связаны с внутренними процессами – ЛЮБОВЬЮ и НЕНАВИСТЬЮ. Будем описывать чувства и эмоции следующим образом: ($С_1$ * О * $С_2$) = Эм , (С * $О_2$* $О_1$) = Эм , ($С_1$ * $С_2$ * $С_3$) = Эм “.

Как следствие, данный подход не может рассматриваться в качестве эффективного теоретического инструмента для проектирования семантико-синтаксических анализаторов.

4.14.2. Сравнение с основными зарубежными подходами

В последнее десятилетие у зарубежных исследователей, работающих в области компьютерной лингвистики, по сравнению с 1980-ми годами значительно усилился интерес к методам формального исследования семантики ЕЯ-тестов. В этот период наибольшей популярностью пользовались три подхода: теория представления дискурсов (ТПД), возникшая в начале 1980-х годов (Kamp 1981), теория концептуальных графов (ТКГ), своим появлением обязанная работам (Sowa 1984, 1991), и эпизодическая логика (ЭЛ), предложенная Л. Шубертом и Ч.Х. Хуан (Schubert 1999, 2000; Shubert, Hwang 1989, 2000; Hwang 1992; Hwang, Schubert 1993a – 1993b).

Анализ показывает, что структура любых формальных выражений, использовавшихся в теории концептуальных графов или эпизодической логике

для отображения содержания ЕЯ-текстов и представления знаний о мире, может быть аппроксимирована выражениями стандартных К-языков (СК-языков). Например, выражение *[книга : {*} @ 50]* , являющееся в ТКГ СП словосочетания “50 книг“, может быть аппроксимировано К-формулой *нек множ * (Колич-элем, 50)(Качеств-состав, книга)* , где *нек* – информационная единица (квантор референтности), соответствующая словам с лексемой “некоторый”, *Колич-элем* и *Качеств-состав* – бинарные реляционные символы, обозначающие отношения “Количество элементов множества” и “Качественный состав множества”. В то же время теория К-исчислений обладает несколькими общими глобальными преимуществами по сравнению с перечисленными выше, а также другими известными подходами к формализации содержания ЕЯ-текстов.

Во-первых, модель, построенная в данной книге, представляет в математической форме *гипотезу* об общих внутренних (или ментальных) механизмах формирования сложных структур концептуального уровня (или семантических структур) из первичных единиц концептуального уровня. Ни ЭЛ, ни ТКГ не предпринимают попытки подобного рода. Во-вторых, построенная выше модель формулирует *гипотезу о полной системе квазилингвистических ментальных операций*, т.е. внутренних операций, позволяющих строить концептуальные структуры, выражающие смысл произвольных реальных предложений и дискурсов на ЕЯ, относящихся к любым областям деятельности человека. Другие же известные подходы к формальному описанию содержания ЕЯ-текстов лишь отмечают расширение выразительных возможностей (как правило, языка логики предикатов первого порядка), не выдвигая гипотезы о построении модели полной системы квазилингвистических ментальных операций и не обсуждая эту проблему.

В-третьих, форма описания как языка концептуальных графов в ТКГ, так и логических форм в ЭЛ не является строго математической. Стиль описания концептуальных графов в работе (Sowa 2001) напоминает стиль описания языка программирования, например, языка Паскаль. Это потенциально затрудняет разработку на основе этой теории алгоритмов обработки знаний, совместимых с процедурами вычислительной логики. Набор форм Бэкуса-Наура, используемый

в диссертации Ч.Х. Хуан (Hwang 1992) для описания базового логического синтаксиса, включает выражение *<1-местная-предикатная-константа>:= счастливый | человек | определенный | вероятный | ...* , а также несколько других выражений сходной структуры. Единственный способ избежать употребления многоточий в продукциях заключается в рассмотрении некоторого аналога понятия концептуального базиса, введенного выше.

Дополнительными общими преимуществами аппарата СК-языков по сравнению с ТКГ и ЭЛ являются: (1) более четкое структурирование предметных областей на основе определения множества типов, (2) рассмотрение отношения совместимости на множестве сортов предметной области, позволяющее связывать со многими сущностями не одну, а несколько "координат" по разным "семантическим осям", (3) наличие средств формального описания смысловой структуры таких дискурсов, в которых есть ссылки на смысл предыдущих фраз или более крупных частей текста, (4) возможность моделирования смысловой структуры фраз с прямой и косвенной речью, (5) возможность рассмотрения информационной единицы, соответствующей слову "понятие", что расширяет арсенал средств формального представления энциклопедической информации, (6) возможности использования логических связок "и", "или" для соединения обозначений объектов или понятий или целей, (7) возможность использования в формулах имен функций, аргументами и/или значениями которых могут быть множества объектов или понятий.

Наряду с общими преимуществами по сравнению с ТКГ и ЭЛ, теория СК-языков обладает рядом индивидуальных преимуществ по сравнению с каждым из этих подходов. Основными дополнительными преимуществами по сравнению с ТКГ являются (1) возможности построения составных обозначений целей, команд, (2) значительно большие возможности построения составных обозначений множеств объектов и множеств понятий. К дополнительным преимуществам по сравнению с ЭЛ, в частности, относятся возможности построения составных обозначений понятий и множеств объектов.

Примеры текстов, анализируемых в публикациях по теории представления дискурсов (ТПД), по своей сложности близки к сложности предложения "Если у

человека есть автомобиль, то он является владельцем кредитной карты". Поэтому, по всей видимости, причины широкой популярности ТПД являются не столько научными, сколько психологическими: простота восприятия изображений, состоящих из блоков с простыми формулами, импонирует широкому кругу лингвистов, интересующихся семантикой ЕЯ. Что же касается ценности ТПД для приложений, то можно согласиться с мнением профессора Л. Аренберга (Ahrenberg 1992, с. 7) о том, что "вопреки своему названию, ТПД, в сущности, может быть охарактеризована как формальная семантика для коротких последовательностей предложений, но не как теория дискурсов". В отличие от ТПД, предложенная в данной работе модель применима к построению СП широкого многообразия дискурсов на ЕЯ, целей и действий систем, построению определений понятий. Каждое из перечисленных выше преимуществ построенной модели по сравнению с ТКГ или ЭЛ является преимуществом модели по сравнению с ТПД.

Важным аспектом расширения в ЭЛ выразительных возможностей логики предикатов является введение специального оператора *Ка* для образования *видов действий*. Например, выражение "красить стену" в работах (Hwang и Schubert 1993a; Schubert и Hwang 2000a) рассматривается как обозначение вида действия. Поэтому с помощью оператора *Ка* строится выражение *Ка (красить стена)* , интерпретируемое как терм языка логики предикатов. Затем образуется выражение

(настоящ-время [Джон любит (*Ка (красить стена))]* .

Это выражение, с одной стороны, интерпретируется как СП предложения " Джон любит красить стену". С другой стороны, в указанных работах это выражение рассматривается как аналог некоторой формулы языка логики предикатов первого порядка.

Попытаемся ответить на вопрос, насколько математически корректным является выбранный способ расширения выразительных возможностей языка логики предикатов первого порядка (ЯЛП1) с помощью оператора Ка.

С одной стороны, неопределенные формы глаголов с зависимыми словами позволяют выражать цели интеллектуальных систем, назначения вещей (например, "подъемные краны нужны для того, чтобы поднимать, перемещать и

опускать грузы”), советы, желания, умения и т.д. Кроме того, к этому виду выражений легко привести и императивные предложения (в частности, команды).

С математической точки зрения, оператор Ка в указанных работах рассматривается как функция, одним аргументом которой является предикат “Красить”, а другим – терм (информационная единица, соответствующая слову “стена”). Но одно из принципиальных ограничений логики предикатов первого порядка заключается в том, что не расматриваются функции, аргументом которых могли бы быть предикаты. Кроме того, выражения, образованные неопределенными формами глаголов с зависимыми словами, могут быть сколь угодно сложными и длинными. Чтобы учесть все возможные случаи, нужно рассматривать счетное множество функций, аргументами которых могут быть как предикаты, так и термы.

Представляется, что такой подход полностью противоречил бы как нашей языковой интуиции, так и принципам математической логики. Именно поэтому в работах по ЭЛ рассматриваются только очень простые выражения подобного вида. Как правило, это неопределенная форма глагола с зависимым существительным.

Между тем, в теории СК-языков предлагается такое оригинальное решение этой проблемы, которок полностью соответствует нашей языковой интуиции и не является насилием над принципами какого-либо математического подхода. Это решение заключается в возможности построения сколь угодно сложных выражений, интерпретируемых как семантические представления выражений, образованных инфинитивами вместе с зависимыми словами. Например, цель “Поступить в Московский государственный университет им. М.В. Ломоносова, окончить его с отличием, подготовить и защитить кандидатскую диссертацию по физике” можно представить в виде К-цепочки

*поступление1 * (Учеб-заведение, нек университет * (Название, ‘МГУ им. М.В.Ломоносова’) : x1) ∧ окончание1 * (Учеб-заведение, x1) ∧ подготовка1 * (Объект1, нек диссертация * (Вид, кандидат)(Область1, физика) : x2 ∧ защита1 * (Объект1, x2)) .*

4.15. Обсуждение построенной математической модели

Многочисленные примеры, расмотренные в этой главе, показывают, что выразительные возможности СК-языков соответствуют постановке задачи в параграфе 3.1. Анализ литературы показывает, что разработанная модель для описания структурированных значений (СЗ) ЕЯ-текстов обладает целым рядом отличий от известных формальных и “полуформальных” подходов к описанию смысловой структуры текстов на естественном языке.

Представляется заслуживающим внимания обсуждение вопроса не только о содержании разработанной модели (т.е. о характере предлагаемых моделью операций на СЗ текстов), но и о форме новой модели. Поиск адекватной формы модели был сопряжен с рядом трудностей. Во-первых, на определенном этапе исследования стало ясно, что модель должна представлять собою индуктивное определение, задающее одно вспомогательное и десять основных правил построения формул. Однако получавшееся индуктивное определение было слишком сложным для восприятия, поскольку требовало одновременного понимания всех одиннадцати способов построения формул трех видов с учетом взаимодействия этих способов.

Поэтому в данной работе предлагается шаг за шагом задавать расширяющиеся множества формул $Forms_k$, где k = 0, 1, …, 10, определяемые совместной индукцией некоторыми правилами P[0], P[1], …, P[k] . Форма каждого из правил P[k] где k = 0, 1, …, 10, такова, что правило P[k] может использоваться без изменения в определениях множеств $Forms_k$, $Forms_{k+1}$, ..., $Forms_{10}$. Такой подход позволяет детально проиллюстрировать эффект введения нового правила P[k] , где k = 1, …, 10, с точки зрения расширения множества определенных на предыдущем шаге формул $Forms_{k-1}$.

Во-вторых, в логике предикатов отдельно определяются сначала множество термов, а затем множество формул, причем составные термы интерпретируются как обозначения различных объектов. Но мы знаем, что в русском, английском и многих других языках для построения составных обозначений объектов могут, в частности, использоваться причастные обороты и придаточные определительные предложения. Как следствие, структура таких составных

обозначений объектов различных видов может быть ничуть не проще, чем структура фраз, выражающих высказывания. Поэтому в данной работе предложено в одном определении задавать множество формул, часть из которых может интерпретироваться как составные обозначения объектов (т.е. как термы).

В-третьих, подход к концептуальномк структурированию предметных областей, предложенный в данной работе, существенно отличается от подхода логики предикатов. В логике предикатов исходят из того, что есть некоторые объекты, функции, заданые на объектах, и высказывания об объектах. Множество объектов не структурируется в традиционной логике предикатов и делится на классы (сорта) в многосортных логиках. Функции не могут быть определены на высказываниях (кроме функции, задающей истинностное значение высказывания), и их значениями не могут быть высказывания.

Между тем, на высказываниях могут быть определены различные функции, например, *Автор (Авторы)* и *Дата*. С другой стороны, естественным было бы рассматривать и функции, значениями которых являются высказывания. Например, если сущность является понятием, то значением функции *Определение* может быть формула, поясняющая смысл понятия.

В данной работе предложен новый подход к концептуальному структурированию предметных областей. Образно говоря, этот подход обладает значительно большей *"разрешающей способностью"* по сравнению с подходом логики предикатов. Это увеличение разрешающей способности структурирования действительности в данной работе обеспечивается:

(1) формальным различением (а) обозначений объектов и обозначений понятий, характеризующих эти объекты, (б) обозначений сущностей (предметов, событий, понятий) и обозначений множеств, состоящих из этих сущностей;

(2) наличием средств формального представления упоряоченных наборов и множеств, состоящих из упорядоченных наборов (т.е. *n*-арных отношений);

(3) возможностью рассматривать функции, аргументами и/или значениями которых могут быть семантические представления ЕЯ-текстов, множества объектов, множества понятий.

В свою очередь, для реализации перечисленных выше идей потребовалось задавать совместной индукцией три класса выводимых формул, а затем каждой формуле одного из этих классов (множества *Ls(B)*, где *B* - концептуальный базис) поставить в соответствие цепочку, называемую типом данной формулы и интепретируемую как концептуальную характеристику сущности, обозначаемой рассматриваемой формулой.

Таким образом, предложенный метод пошагового ввода правил построения формул с последующим соединением этих правил с помощью итогового индуктивного определения является оригинальным и может рассматриваться как один из научных результатов данной работы. Этот метод обогащает дискретную математику (в первую очередь – теорию формальных языков) и может использоваться для создания новых исчислений, формулы которых предназначены для описания структуры сложных линейных объектов.

В последние три десятилетия класс языков логики предикатов первого порядка (ЛППП) являлся стандартом, с которым сравнивались предлагавшиеся новые подходы к формальному представлению содержания ЕЯ-текстов. Чаще всего, такие новые подходы рассматривались их авторами как расширения языка ЛППП. Учитывая это, можно выделить следующие преимущества класса стандартных К-языков по сравнению с классом языков ЛППП (при этом будет использоваться нумерация свойств ЕЯ-текстов, введенная в параграфе 3.1):

(Св. 3) возможность строить и формально различать обозначения единиц, соответствующих (а) объектам, ситуациям, процессам в реальном мире и (б) понятиям, квалифицирующим (характеризующим) эти объекты, ситуации, процессы; (Св. 4.1) возможность строить и различать обозначения объектов и множеств объектов; (Св. 5) возможность различать формальным образом понятия, квалифицирующие объекты, и понятия, квалифицирующие множества объектов тех же видов;

(Св. 6) возможность строить составные обозначения понятий, т. е. строить формулы, отражающие поверхностно-семантическаю структуру ЕЯ-выражений, подобных выражению “человек, окончивший МГУ имени М.В. Ломоносова и являющийся биологом или химиком”; (Св. 8) возможность строить обозначения упорядоченных n-местных наборов различных сущностей, где $n > 1$;

(Св. 9). возможность строить (9.1) формальные аналоги составных обозначений множеств, (9.2) обозначения множеств упорядоченных наборов сущностей (9.3) обозначения множеств, состоящих из множеств;

(Св. 11) возможность моделировать смысловую структуру фраз, содержащих, в частности: (11.2) выражения, полученные применением связок “и”, “или” к обозначениям (11.2а) предметов, событий; (11.2б) понятий; (11.3) выражения , где связка “не” стоит непосредственно перед обозначением предмета, события, понятия и т. д.; (11.4) косвенную речь; (11.5) причастные обороты и придаточные определительные предложения ;

(Св. 14) возможность моделировать смысловую структуру дискурсов со ссылками на смысл фраз и более крупных фрагментов рассматриваемых текстов содержит семантическое представление дискурса “У А.Зубова есть три друга. П.Сомов это знает”; (Св. 17) возможность рассматривать нетрадиционные функции (и другие нетрадиционные отношения) с аргументами и/или значениями, являющимися множествами предметов, ситуаций.

Глава 5
АНАЛИЗ ВОЗМОЖНОСТЕЙ ПРИМЕНЕНИЯ АППАРАТА СК-ЯЗЫКОВ К РЕШЕНИЮ РЯДА АКТУАЛЬНЫХ ПРОБЛЕМ ИНФОРМАТИКИ

Данная глава посвящена исследованию возможностей применения аппарата стандартных К-языков (СК-языков) к разработке языков представления содержания посланий компьютерных интеллектуальных агентов, языков формирования контрактов и протоколов переговоров в области электронной коммерции, созданию семантического сетевого языка нового поколения, построению онтологий предметных областей, разработке новых языков представления знаний для решения информационно-сложных задач, проектированию интеллектуальных информационно-поисковых и вопросо-ответных Интернет-систем нового поколения. Содержание данной главы базируется на публикациях (Фомичев 1991, 1992б, 2001а, 2002б, 2005а – 2005в, 2005д; Fomichov 1998b - 2001, 2002b, 2004, 2005).

5.1. Определение класса стандартных К-языков как формальная метаграмматика для описания содержания посланий компьютерных интеллектуальных агентов

5.1.1. Проблема разработки языков общения компьютерных интеллектуальных агентов

Прогресс исследований в области искусственного интеллекта (ИИ) и компьютерных сетей привел к появлению теории многоагентных систем (МАС). Многоагентные системы являются одним из наиболее быстро развивающихся в 1990-е годы и начале 2000-х годов направлений информатики.

Главная причина постоянно растущего интереса к этому направлению заключается в следующем. В настоящее время можно прогнозировать бурное развитие в ближайшие годы *электронной коммерции* (electronic commerce, E-commerce), базирующейся на широчайших возможностях сети Интернет (Thome и Schihzer 1998); МАС рассматриваются как ключевая технология для конструирования систем электронной коммерции (Guilfoyle, Jeffcoate, Stark 1997; Wooldridge 1998).

В частности, одной из перспективных областей использования МАС является индустрия туристического сервиса. Задача проектирования компьютерных интеллектуальных агентов (КИА) для этой индустрии является весьма естественной, поскольку различные КИА многих поставщиков услуг (заказ авиабилетов и железнодорожных билетов, резервирование комнат в гостиницах, взятие напрокат автомобилей, организация культурной программы и т.д.) должны динамически обнаруживать друг друга и эффективно взаимодействовать для решения стоящих перед ними задач.

Многочисленные КИА (сконструированные разными научно-исследовательскими центрами, использующие различную аппаратуру и программное обеспечение) смогут эффективно взаимодействовать в процессе решения своих задач только в том случае, когда эти КИА будут располагать общим языком для обмена информацией и руководствоваться едиными правилами общения. Поэтому в 1990-е годы было разработано несколько *языков общения интеллектуальных агентов*; два из них являются наиболее широко применимыми.

Во-первых, это язык KQML, разработанный в США в рамках проекта разделения знаний, осуществлявшегося национальным агентством ДАРПА (DARPA). Исследования по разработке этого языка отражены, в частности, в публикациях (Finin и др. 1993; Labrou 1996; Finin, Labrou, Mayfield 1997; Labrou, Finin 1997, 1998). Значительную роль в этих исследованиях сыграли специалисты Стэндфордского университета.

Второй язык разработан в рамках международного Фонда интеллектуальных физических агентов, или ФИФА (the Foundation for Intelligent Physical Agents, или

FIPA), штаб-квартира которого находится в Женеве. Одним из важных результатов исследований, организованных этим фондом, стала разработка в 1997 - 1999 годах стандарта для представления посланий (messages) КИА, получившего название Языка общения агентов (FIPA Agent Communication Language = FIPA ACL) (FIPA 1998a). Теоретической основой для создания этого языка послужили принципы разработки языка KQML и языка KIF, или Knowledge Interchange Format (Genesereth, Fikes и др., 1992; Genesereth, 1999). Та часть языка FIPA ACL, которая предназначена для представления содержания посланий КИА, называется *семантическим языком* (FIPA Semantic Language = FIPA SL).

Проблема создания адекватных логико-информационных основ электронной коммерции предъявляет высокие требования к языку представления содержания посланий КИА. Этот язык должен позволять отображать содержание коммерческих переговоров. Однако язык FIPA SL обладает многими ограничениями в этом отношении. В связи с этим возникает проблема разработки математического описания такого класса языков, который был бы удобен для отображения содержания произвольных посланий КИА.

В нашей стране в последнее десятилетие проблематике многоагентных систем уделялось значительное внимание многими учеными. В частности, различные аспекты теории МАС и вопросы применения этой теории исследовались в работах В.И. Городецкого (Городецкий 1998), В.В. Емельянова (Emelyanov 2001), Э.С. Клышинского (Клышинский 1999), Г.С. Плесневича и В.Б. Тарасова (Тарасов 1998;.Plesniewicz, Tarassov 2001), Д.А. Поспелова (Поспелов 1998), Г.В. Рыбиной и В.Ю. Берзина (Рыбина, Берзин 2002).

Однако следует отметить, что проблематика разработки формальных языков с широкими выразительными возможностями для представления содержания посланий КИА не получила в трудах ученых нашей страны такого же внимания, как и в серии зарубежных проектов, выполненных в рамках международного Фонда интеллектуальных физических агентов.

5.1.2. Возможности стандартных К-языков для представления содержания посланий компьютерных интеллектуальных агентов

СК-языки обладают целым рядом свойств, делающих их удобными для представления содержания произвольных посланий КИА. Рассмотрим некоторые из этих свойств.

Свойство 1. К-языки позволяют строить формальные составные обозначения понятий.

Пример 1. Пусть П1 = “тургруппа, состоящая из 12 ученых”, тогда возможным К-представлением (КП) выражения П1 является цепочка

*тур-группа * (Колич-элементов, 12) (Качеств-состав, ученый)* .

Пример 2. Если П2 - выражение “керамика, выпущенная в Индии или Шри-Ланке”, то возможно первое КП П2: *керамика1 * (Производство, (Индия V Шри-Ланка))* и второе КП *керамика1 * (Производство, (нек страна * (Назв, ‘Индия’) : x1 ∨ нек страна * (Назв, ‘Шри-Ланка’) : x2))* .

Пример 3. Пусть П3 = “контейнер, содержащий 8 коробок с чайными сервизами из Китая и 4 коробки со столовыми сервизами из Индии или Шри-Ланки”. Тогда найдется такой концептуальный базис *B*, что *Ls (B)* включает следующее выражение, являющиеся возможным семантическим представлением П3:

*контейнер1 * (Содержание1, (нек множество * (Колич-элементов, 8)*
*(Кач-состав, коробка1 * (Содержание2, сервиз 1 * (Вид, чайный)*
*(Страна, нек страна1 * (Название, ‘Китай’) : x1))) : S1 ∧*
*нек множество * (Колич-элементов, 4)*
*(Кач-состав, коробка1 * (Содержание2, сервиз 1 * (Вид, столовый)*
*(Страна, нек страна 1 * (Название, “Индия” V “Шри-Ланка”) : x2))) : S2))* .

Свойство 2. СК-языки предоставляют широкие возможности построения определений понятий в виде (а) $(c \equiv b * (r_1,d_1)\ldots(r_n,d_n))$

или (б) $((c \equiv b * (r_1,d_1)\ldots(r_n,d_n)) \wedge A)$,

где c – вводимое понятие, b – базовое понятие (считается известным), $n \geq 1$, $r_1,\ldots, r_n$

бинарные реляционные символы; d_1 ,…, d_n – К-цепочки, A - К-цепочка, интерпретируемая как высказывание о свойствах объектов, характеризуемых понятием с..

Пример 1. Пусть П1 – определение “Freight forward – это груз, оплачиваемый в порту назначения (англ. язык)”. Тогда СП определения П1 может являться следующим выражением некоторого СК-языка:

*(freight-forward ≡ груз1 * (Описание1, < x1, Оплата (x1, нек порт 1 * (Пункт-назначения, x1)) >) (Язык, английский))* .

Пример 2. Пусть П2 = “Малое предприятие – это предприятие с количеством сотрудников, не превышающим 50 человек”. Тогда возможным КП определения П2 является следующее выражение:

Определение1 (Малое предприятие, x1, (Явл1 (x1, предприятие) ∧ ¬ Больше (Колич-элементов (Штат 1 (1x)), 50))) .

Здесь используется другая форма представления определения:

Определение1 (c, v, Des (v)) ,

где *c* – обозначение понятия (поясняемого), *v* – переменная (обозначает произвольную сущность, характеризуемую понятием *c*), *Des (v)* – К-цепочка (или *l*-формула), являющаяся СП высказывания о сущности с меткой *v*.

Свойство 3. С помощью СК-языков можно строить составные обозначения различных сущностей, в том числе обозначения множеств, для этого сначала строится составное обозначение понятия (см. свойство 1), причем применяется правило Р[8], а затем добавляется квантор референтности с помощью правила Р[1].

Пример 1. К-цепочка *коробка1 * (Содержание2, сервиз1 * (Вид, чайный))* , построенная в результате применения на последнем шаге правила Р[8] , интерпретируется как составное обозначение понятия “коробка с чайными сервизами”. В результате применения правила Р[1] можно получить выражения

нек коробка 1 (Содержание2, сервиз1 * (Вид, чайный))* , (*)

*все коробки1 * (Содержание2, сервиз1 * (Вид, чайный))* . (**)

Выражение (*) будем интерпретировать как обозначение коробки, где находится один или несколько чайных сервизов. Выражение (**) будем рассматривать как обозначение множества, состоящего из всех коробок, содержащих чайные сервизы.

Пример 2. Мы можем следующим образом обозначить конкретную запланированную серию из 5-ти поставок, каждая из которых включает 60 чайных сервизов № 53 и 36 столовых сервизов № 65:

*Нек множество * (Колич-элементов, 5) (Кач-состав, поставка1 * (Состав2, (нек множество * (Колич-элементов, 60) (Кач-состав, сервиз1 * (Вид, чайный) (Номер, 53)) ∧ нек множество * ((Колич-элементов, 36)*

*(Кач-состав, сервиз1 * (Вид, столовый) (Номер, 65))))) : S1* .

Свойство 4. Следствием свойства 3 является возможность моделировать способ использования йота-оператора в языке FIPA ACL, он включает, в частности, выражение *(iota ? x (UK Prime Minister ? x)* , интерпретируемое как семантическое представление выражения “премьер-министром Великобритании”. Тогда можно построить эквивалентное К-представление этого выражения

*нек чел * (Премьер-министр, UK) : x1* .

Для построения этого КП нужно выполнить следующие шаги:

(1) с помощью правил P[0], P[1] , P[4] построить цепочку

Премьер-министр (нек чел, UK);

(2) по правилу P[8] получить цепочку *чел * (Премьер-министр, UK);*

(3) по правилу P[1] слева приписывается квантор референтности *нек*;

(4) по правилу P[5] справа приписывается цепочка *: x1.*

Свойство 5. СК-языки удобны для построения семантических представлений простых и составных целей.

Пример 1. Пусть G1 – цель “Зарезервировать 12 одноместных номеров в трёх-звёздочных отелях Любляны.”. Тогда К-представлением этой цели может являться выражение

*Резервир1 * (Объект1, нек .множество * (Колич-элементов, 12) (Кач-состав, номер1 * (Вид1, одноместн) (Место1, произвольн отель * (Вид2, трёхзвёздочн) (Локализация, нек город * (Назв, ‘Любляна’) : x1))) : S1) : goal1.*

Пример 2. Пусть G2 = “Доставить фирме “Spencer & Co” в течение 12-19 февраля 2004 года 5 партий, каждая из которых состоит из 60 чайных сервизов № 53 и 56 столовых сервизов № 65”. Тогда цель (или распоряжение) G2 может иметь следующее К-представление:

*Доставка1 * (Адресат1, нек фирма1 * (Назв, ‘Spencer & Co’) : x1)*

(Время, <12.02.2004, 19.02.2004>) (Объект1, setdescr1),

где *setdescr1* – построенное выше КП выражения “серия из 5-ти поставок, каждая из включает 60 чайных сервизов № 53 и 36 столовых сервизов № 65.”

Свойство 6. СК-языки дают возможность представлять коммуникативные акты языка FIPA ACL. Предположим, что компьютерный интеллектуальный агент (КИА) “Клиент-агент” просит КИА “Антологический агент” (электронный поставщик знаний) сообщить, какие бывают виды цитрусовых. Тогда этот коммуникативный акт можно представить в виде следующего выражения некоторого СК-языка:

*нек коммуник-акт * (Вид, запрос)(Отправитель, Клиент-агент)*

*(Получатель, Онтол-агент) (Содержание1, (Вопрос (x1 ≡ нек множество **

*(Кач-состав, произв понятие * (Конкретизация, цитрус)))))*

(Язык-запроса, СК-язык)(Онтология, fipa-ontol-service- fruits-ontology)

(Метка-ответа, цитрус-запрос) (Язык-ответа, СК-язык).

Свойство 7. СК-языки удобны для построения СП вопросов.

Пример. Пусть В1 = “Сколько стоит двухтонный контейнер?”. Тогда возможное К-представление В1:

*Вопрос (x1, (x1 ≡ Цена (произв контейнер1 * (Вес, 2/ тонна)))).*

Свойство 8. СК-языки позволяют использовать ту же форму для построения общих вопросов, т.е. вопросов с ответом ДА / НЕТ.

Пример. Пусть В2 = “Работает ли Сергей Сомов в фирме IBM?”

Тогда К-представление В2 может являться следующим выражением:

*Вопрос (x1, (x1 ≡ Ист-знач. ((∃e1 (ситуация) Явл1 (e1, работа1 **

*(Агент1, нек человек * (Имя, ‘Сергей’) (Фамилия, ‘Сомов’) : x2) (Место3,*

*нек фирма * (Назв, ‘IBM’) : x3)) ∧ Время (e1, #сейчас#))))) .*

5.2. Анализ возможностей использования СК-языков для формирования контрактов и протоколов переговоров в области электронной коммерции

В течение нескольких последних лет в области электронной коммерции возникли два взаимосвязанных научных направления, получивших названия *электронные переговоры (e-negotiations)* и *электронное заключение контрактов (electronic contracting)*. Рождение этих направлений было формально обозначено проведением в начале 2000-х годов нескольких международных конференций и симпозиумов, в том числе конференции по электронным контрактам и вычислениям, базирующимся на контрактах (Цюрих, Швейцария, 2001); 6-й международной конференции по информационным системам для бизнеса (Колорадо Спрингс, США, 2003); симпозиума по теории и применениям электронных переговоров (Познань, Польша, апрель 2004); 1-го международного симпозиума по электронному заключению контрактов, организованного Международным институтом инженеров по электричеству и электронике (IEEE) в июле 2004 г. в Сан Диего, Калифорния, США.

К центральным задачам, стоящим перед исследователями в этих научных направлениях, относится создание формальных языков для представления содержания коммерческих переговоров, проводимых компьютерными интеллектуальными агентами (КИА), и для построения контрактов, заключаемых КИА в ходе таких переговоров. Эти задачи можно рассматривать как важные частные случаи проблемы разработки формальных языков общего назначения для бизнес-коммуникаций (Kimbrough и Moore 1997; Hasselberg и Weigand 2001).

В работе (Hasselberg и Weigand 2001) подчеркивается, что если послания в области электронной коммерции дожны обрабатываться автоматически, то значения (meanings) посланий должны быть формализованы. Эта идея совпадает с высказанным в статье (Kimbrough и Moore 1997) мнением о необходимости

создания логико-семантических основ конструирования формальных языков для бизнес-коммуникаций (ФЯБК).

Кимбру и Мур в указанной выше работе предлагают использовать как можно шире аппарат логики первого порядка для построения выражений любого ФЯБК. Однако выразительные возможности класса языков логики первого порядка очень ограничены с точки зрения описания семантической структуры произвольных бизнес-документов.

Анализ показывает, что протоколы коммерческих переговоров и контрактов могут формироваться с помощью выразительных механизмов естественного языка (ЕЯ), используемых для построения произвольных ЕЯ-текстов, относящихся к медицине, технике, юриспруденции и т.д. В частности, тексты из таких документов могут включать: (а) неопределенные формы глаголов (или инфинитивы) с зависимыми словами, выражающие цели, предложения (“продать 50 ящиков с яблоками”), обещания, обязательства и назначения предметов; (б) конструкции, образованные из инфинитивов с зависимыми словами с помощью логических связок “и”, “или”, “не” и являющиеся составными обозначениями целей, предложений, обещаний, обязательств и назначений предметов; (в) составные обозначения множеств (“партия, состоящая из 50 ящиков с яблоками”); (г) фрагментов, в которых логические связки “и”, “или” соединяют не обозначения высказываний, а обозначения предметов; (д) фрагментов, содержащих ссылки на смысл фраз или более крупных фрагментов дискурса (“это предложение”, “его распоряжение”, “это обещание” и т.д.); (е) обозначения функций, аргументами и/или значениями которых могут быть множества объектов (“персонал фирмы А”, “поставщики фирмы А”, “количество поставщиков фирмы А”); (ж) вопросы с ответом “Да” или “Нет”; (з) вопросы с вопросительными словами.

Между тем, логика первого порядка не позволяет строить формальные аналоги (на семантическом уровне) таких текстов из бизнес-документов, для которых выполняются перечисленные выше свойства (а) – (ж).

Поэтому проблема разработки формальных языков, позволяющих отображать содержание протоколов коммерческих переговоров, проводимых КИА, и

формировать контракты, заключаемые в ходе таких переговоров, очень сложна. В этой связи представляется разумным использовать для решения этой проблемы наиболее широко применимые теории представления структурированных значений ЕЯ-текстов, предоставляемые математической лингвистикой и математической информатикой.

СК-языки обладают рядом выразительных возможностей, необходимых для формального представления содержания контрактов. Для иллюстрации важной части таких возможностей рассмотрим сценарий взаимодействия деловых партнеров в ходе обработки страховой компанией поступившего заявления о повреждении автомобиля. Этот сценарий опубликован в работе (Xu, Jeusfeld 2003), посвященной электронным контрактам. Деловыми партнерами являются страховая компания “AGFIL”, фирмы с названиями “Europ Assist”, “Lee Consulting Services”(сокращенно “Lee C.S.”), а также сервисные центры и техничские эксперты. Фирма “Europ Assist” предоставляет владельцам страховых полисов возможность круглосуточного (24 часа) экстренного обращения по телефону. Фирма “Lee C.S.” ежедневно координирует и управляет операциями срочного обслуживания по соглашению с компанией “AGFIL”.

В целом процесс страхового обслуживания осуществляется следующим образом. Владелец страхового полиса звонит в “Europ Assist”, чтобы заявить о новом страховом случае. Фирма “Europ Assist” регистрирует поступившую заявку, предлагает подходящий сервисный центр и направляет извещение компании “AGFIL”, которая проверяет действительность полиса и то, покрывает ли полис заявку. После того, как компания “AGFIL” получает заявку, эта компания направляет детали заявки фирме “Lee C.S.”.

Компания “AGFIL” посылает владельцу полиса письмо, содержащее полную форму заявления. Фирма “Lee C.S.” соглашается с затратами на ремонт, если эксперт не требуется в связи с тем, что размер ущерба мал; в противном же случае назначается эксперт. Эксперт осматривает поврежденный автомобиль и договаривается с сервисным центром о затратах на ремонт. После получения от фирмы “Lee C.S.” договора о ремонте автомобиля сервисный центр начинает

ремонт. После завершения ремонта сервисный центр посылает счет фирме “Lee C.S.”, которая сравнивает счет с первоначальной оценкой. Фирма “Lee C.S.” возвращает все счета компании “AGFIL”. Эта компания осуществляет выплаты. Если заявка будет признана недействительной, то об этом будут проинформированы все стороны, участвующие в процессе, и процесс будут остановлен.

Данный сценарий позволяет проиллюстрировать ряд свойств СК-языков, делающих их удобным инструментом формального описания контрактов.

Свойство 1. Возможность построения составных обозначений целей.

Пример. Пусть Т1 = “Владелец полиса звонит в фирму “Europ Assist”, чтобы сообщить о повреждении автомобиля”. Тогда Т1 может иметь К-представление (КП)

*Ситуация (e1, телеф-разговор * (Агент1, нек чел * (Владеет1, нек полис1))(Объект2, нек фирма * (Назв, “Europ Assist”)(Цель, сообщение1 * (Тема1, нек повреждение1 * (Объект1, нек автомобиль))))*.

Свойство 2. Наличие средств компактного представления временных и причинно-следственных отношений между ситуациями.

Свойство 3. Возможность построения компактных семантических образов таких фрагментов предложений, которые получены в результате соединения логическими связками “И”, “ИЛИ” обозначений предметов, событий, понятий или целей.

Пример. Пусть Т2 = “После получения накладной по ремонту от фирмы “Lee C.S.” и заявления от владельца страхового полиса, компания “AGFIL” оплатит сервисному центру стоимость ремонта”. Тогда КП текста Т2 может являться выражением

*(Ситуация (e1, (получение1 * (Агент2, нек фирма * (Назв, “AFGIL”) : x1)(Объект1, нек накладная * (Тема, нек ремонт : e2) : x2)(Отправитель, нек фирма * (Назв, “Lee C.S.”) : x3) ∧ получение1 * (Агент2, x1)(Объект1, нек заявление1 : x4) (Отправитель, нек чел * (Владеет1, нек полис1 : x5) : x6))) ∧ Ситуация (e2, оплата1* (Агент2, x1)(Адресат1, нек сервис-центр : x7)(Сумма, Стоимость (e2))) ∧ Раньше (e1, e2))* .

Свойство 4. Существование средств формального представления содержания дискурсов со ссылками на смысл фраз и более крупных фрагментов текста.

Пример. Пусть Т3 = “Фирма “Europ Assist” предоставляет телефонное обслуживание владельцу страхового полиса; в частности, указывает сервисный центр для ремонта и извещает компанию “AGFIL” о заявлении владельца страхового полиса”. Тогда Т3 может иметь КП

*(Ситуация (e1, обслуживание1 * (Агент2, нек фирма * (Назв, “Europ Assist”) : x1)(Инструмент, нек телефон : x2)(Объект1, произвольн чел * (Владеет1, нек полис1 : x3) : x4)) : P1 ∧ Конкретизация (P1, (Ситуация (e2, указание1* (Агент2, x1)(Адресат, x4))(Объект3, нек сервис-центр * (Назначение1, ремонт) : x5)) ∧ Ситуация (e3, извещение1* (Агент2, x1)(Адресат1, нек фирма * (Назв, “AFGIL”) : x6)(Содержание1, нек заявление1* (Автор, x4) : x7))))) .*

Свойство 5. Возможность формального представления содержания контрактных обязательств, зависящих от условий.

Пример. Пусть Т4 = “Фирма “Lee C.S.” назначает эксперта для осмотра автомобиля в течение 41 часа с момента получения заявления о повреждении автомобиля, если стоимость ремонта не превышает 500 USD”. Тогда КП текста Т4 может являться выражением

*Если-то (¬ Больше1 (Стоимость (нек ремонт * (Объект1, нек автомобиль : x1) : e1), < 500, USD >) , (Ситуация (e2, назначение1* (Агент2, нек фирма * (Назв, “Lee C.S.”) : x2)(Персона1, нек эксперт : x3)(Цель1, нек осмотр * (Объект1, x1) : e3)(Момент, t1)) ∧ ¬ Больше1 (Разность (t1, t0), <41, час >) ∧ Ситуация (e4, получение1 * (Агент2, x2)(Объект1, нек заявление1* (Тема, нек повреждение1 * (Объект1, x1) : e5))(Время, t0)))) .*

Выходя за рамки обсуждавшегося в данном параграфе сценария взаимодействия деловых партнеров в процессе обслуживания заявления о страховом случае, сформулируем два дополнительных свойства СК-языков.

Свойство 6. Наличие средств построения составных обозначений множеств как компонентов семантических представлений ЕЯ-текстов, являющихся протоколами переговоров или контрактами (см. в предыдущем параграфе

обозначение запланированной серии из 5-ти поставок, каждая из которых включает 60 чайных сервизов № 53 и 36 столовых сервизов № 65).

Свойство 7. Возможность построения объектно-ориентированных СП протоколов переговоров и контрактов, т.е. формальных выражений вида

нек информ-объект * (Вид, concept)(Содержание1, content)(r_1 , u_1)…(r_n , u_n) ,

где concept – обозначение понятия “протокол переговоров“ или “контракт“, content – К-представление документа, r_1 , ..., r_n - обозначения внешних характеристик документа (задающих метаданные, например, характеристики Авторы, Дата, Язык), u_1, ..., u_n – цепочки, интерпретируемые как значения характеристик документа.

Дополнительные полезные свойства СК-языков с точки зрения построения СП контрактов и протоколов переговоров определяются возможностями явного указания тематических ролей (концептуальных падежей) в структуре семантического представления ЕЯ-текста, отображения содержания фраз с прямой и косвенной речью, со словами “понятие“, “термин”, рассмотрения функций, значениями и/или аргументами которых могут быть множества объектов (Поставщики, Ассортимент, Директор, Персонал и т.д.).

Автором данной работы был проведен сравнительный анализ выразительных возможностей СК-языков и явлений ЕЯ, отражающихся в структуре деловых контрактов и протоколов коммерческих переговоров (Fomichov 1999a, 2002b). Ряд результатов этого анализа был изложен выше. Проведенный анализ позволяет высказать предположение о том., что выразительных возможностей СК-языков достаточно для построения с их помощью формальных представлений контрактов и протоколов коммерческих переговоров.

С другой стороны, выразительные возможности других известных подходов к формальному представлению содержания ЕЯ-текстов недостаточны для построения СП произвольных контрактов и протоколов переговоров. В частности, это относится к теории представления дискурсов, теории концептуальных графов, эпизодической логике, компьютерной семантике русского языка.

Таким образом, аппарат СК-языков открывает новые возможности построения формальных представлений контрактов и протоколов коммерческих переговоров, осуществляемых компьютерными интеллектуальными агентами.

В то же время из проведенного анализа следует, что СК-языки предоставляют уникальный спектр возможностей для представления результатов семантико-синтаксической обработки лингвистическими процессорами дискурсов, являющихся контрактами или протоколами коммерческих переговоров.

5.3. Разработка семантического сетевого языка нового поколения

В параграфе 1.1 отмечалось возникновение во второй половине 1990-х годов нового направления в разработке семантических языков-посредников и ЛП, использующих такие языки. Это направление появилось как следствие разработки японскими учеными Х. Учидой и М. Жу формального языка для представления содержания предложений, названного ими универсальным сетевым языком (UNL, the Universal Networking Language). Первым центральным мотивом для создания языка UNL было стремление устранить языковой барьер между пользователями сети Интернет из разных стран мира. Второй центральный мотив заключался в попытке создать языковые средства, позволяющие представить в едином формате самые разные знания, накопленные человечеством, и, как следствие, создать объективные предпосылки для совместного использования этих знаний разнообразными компьютерными системами по всему миру.

С конца 1990-х годов Институтом передовых исследований ООН Токийского университета координируется ряд проектов в разных странах, цель которых заключается в создании семейства ЛП, преобразующих предложения на различных естественных языках в выражения языка UNL, а также строящих выражения языка UNL по предложениям на разных естественных языках. В целом эта система проектов охватывает 16 естественных языков, включая 6 официальных языков ООН (Uchida, Zhu, Della Senta 1999; Uchida, Zhu 2001).

Научные результаты, полученные в ходе реализации этих проектов, стали предметом обсуждения на международной конференции по универсальным знанию

и языку, состоявшейся в ноябре 2002 г. в Индии. Значительное внимание на этой конференции было уделено различным аспектам представления знаний о мире в едином формате с помощью языка UNL (Zhu, Uchida 2002).

Из результатов глав 3, 4 следует, что выразительные возможности класса СК-языков значительно превышают выразительные возможности языка UNL. В первую очередь, следует отметить, что язык UNL ориентирован на представление содержания только отдельных предложений, но не произвольных связных текстов. Кроме того, весьма ограничены возможности использования языка UNL для представления знаний о мире. Таким образом, по своим выразительным возможностям язык UNL не полностью, а лишь частично соответствует своему названию “универсальный сетевой язык”. Поэтому представляется обоснованной интерпретация языка UNL как одной из возможных версий семантического языка для сети Интернет, или семантического сетевого языка.

В этой связи можно провести аналогию между исследованиями по разработке семантического сетевого языка, одним из вариантов которого является язык UNL, и исследованиями по разработке языков формирования Web-документов. На протяжении 1990-х годов проходил бурный рост сети World Wide Web (Всемирной Паутины), причем для представления информации использовался преимущественно язык разметки гипертекстов HTML. Однако язык HTML не был предназначен для выделения смысловых частей электронных документов, что привело к большим трудностям принципиального характера при поиске документов, удовлетворяющих запросу пользователя.

Поэтому в конце 1990-х годов консорциумом Всемирной Паутины (обычно обозначается сокращением W3C) была начата подготовка к переходу к новым, семантически-структурированным средствам представления информации в Web-документах. Несколько лет предварительного этапа исследований позволили разработать язык для описания метаданных об информационных ресурсах RDF (Resource Description Framework) и язык RDF SSL - систему спецификации схем, являющихся выражениями языка RDF (RDF 1999; RDF SSL 2000), а затем объявить

о развертывании широкомасштабного проекта Семантической Всемирной Паутины (Semantic Web 2001).

Учитывая сказанное, можно предположить, что широко рекламируемый сегодня язык UNL является не окончательной, а лишь начальной версией семантического сетевого языка. Потребности формального представления содержания сложных связных текстов (например, относящихся к медицине, биологии, экономике, технике, экологии, юриспруденции), а также смысловой обработки семантических представлений (СП) таких текстов в рамках базы знаний о мире должны привести в ближайшие годы к разработке семантического сетевого языка нового поколения.

На основании анализа научной литературы по формализации семантики ЕЯ-текстов можно сделать вывод о том, что в качестве такого языка наиболее целесообразно выбрать СК-язык в некотором концептуальном базисе, построенном с учетом опыта разработки языка UNL. Например, в работе (Uchida, Zhu 2001) рассматривается бинарное отношение *ins* (Инструмент) , связывающее обозначение события и обозначение предмета, с помощью которого реализовано событие. Поэтому можно рассматривать концептуальные базисы с бинарным реляционным символом *ins* , имеюшим тип *tp(ins) = {(соб , [об])}* , где *соб* – сорт “событие” , об – базовый тип “объект” .

Проведенный анализ показал, что нетрудно аппроксимировать все выразительнве механизмы языка UNL средствами СК-языков. Главным образом, это обусловлено тем, что правило P[4] предназначено для конструирования формул с именами *n*-арных отношений, и правило P[8] позволяет строить составные обозначения понятий.

Пример 1. Рассмотрим UNL-выражение *to(train(icl > thing), London(icl > city))* ; это выражение, взятое из работы (Uchida, Zhu, Della Senta 1999), обозначает поезд на Лондон. Данное выражение можно аппроксимировать К-цепочкой вида *Назначение (нек поезд * (Конкретизация, вещь), нек город * (Назв, 'Лондон’))* или вида *нек поезд * (Конкретизация, вещь) (Назначение, нек город * (Назв, 'Лондон’))*.

В то же время, аппарат СК-языков предоставляет ряд важных преимуществ по сравнению с UNL с точки зрения разработки семантического сетевого языка нового поколения. Проиллюстрируем несколько таких преимуществ.

Пример 2. Рассмотрим определение Def1= “A flock (английский язык) – это большое количество птиц или млекопитающих (например, овец или коз), собирающихся вместе с определенной целью, такой, как питание, миграция или оборона”. Тогда определение Def1 может иметь следующее К-представление *Expr1*:

*Определение1 (flock, англ-яз, динамич-группа * (Кач-состав,*
*(птица ∨ млекопитающее * (Примеры, (овца ∨ коза)))), S1,*
(Оценка(Колич-элем(S1), большое) ∧ Цель-формирования (S1,
*Нек намерение * (Примеры, (питание ∨миграция ∨оборона))))) .*

Анализ этой формулы позволяет сделать вывод о том, что при построении семантических представлений (СП) ЕЯ-текстов удобно использовать: (1) обозначение 5-арного отношения *Определение1*, (2) составные обозначения понятий (в данном примере использованы выражения *млекопитающее * (Примеры, (овца ∨ коза))* и *динамич-группа * (Кач-состав, (птица ∨ млекопитающее * (Примеры, (овца ∨ коза))))*), (3) имена функций, аргументами и/или значениями которых могут быть множества (в примере использовано имя одноместной функции *Колич-элем* , значением которой является количество элементов множества), (4) составные обозначения намерений, целей (в примере – выражение *нек намерение * (Примеры, (питание ∨миграция ∨оборона))*).

Структура построенного К-представления *Expr1* в значительной мере отражает структуру исходного определения Т1. Между тем, попытка представить содержание этого определения на языке UNL, т.е. с помощью только обозначений бинарных отношений, привела бы к полному разрушению связи между структурой исходного определения Т1 и структурой UNL-представления данного определения.

Пример 3. Пусть D1 – относящийся к биологии и медицине дискурс “Все гранулоциты являются полиморфонуклеарными. Это означает, что их ядра

многодольны”. Тогда дискурсу D1 можно поставить в соответствие следующее К-представление *Expr2*:

(Свойство (произвольн гранулоцит : x1 , полиморфонуклеарный) : P1) ∧
Пояснение (P1, Следует-из (Ситуация (e1, обладание1 (Агент1, x1)*
(Объект1, нек ядро1 : x2)), Свойство (x2, многодольный)))) .

Ключевую роль в построении К-представления *Expr2* сыграло правило P[5], позволившее ввести метку *x1* для обозначения произвольного гранулоцита, метку *x2* для обозначения ядра клетки, и метку *P1* для обозначения семантического представления первого предложения из дискурса D1. Метка *P1* позволяет в структуре СП текста D1 эксплицировать ссылку на смысл первого предложения текста, даваемую сочетанием “Это означает”.

Язык UNL не включает средств представления ссылок на смысл фраз и более крупных фрагментов дискурса. Между тем, последний пример содержит один из наиболее коротких дискурсов такого рода. Учебники в различных областях знаний изобилуют значительно более сложными дискурсами со ссылками на смысл фраз и более крупных фрагментов.

Анализ показывает, что выразительные механизмы языка UNL нетрудно аппроксимировать средствами СК-языков, поскольку правило P[4] позволяет использовать бинарные реляционные символы. В то же время разработка семантического сетевого языка нового поколения.на основе определения класса СК-языков, в частности, позволит: (1) строить не только СП предложений, но и СП сложных связных текстов за счет средств представления ссылок на ранее упомянутые объекты и на смысл фраз и более крупных фрагментов текстов; (2) формировать составные обозначения множеств, понятий, целей интеллектуальных систем и назначений объектов; (3) соединять с помощью логических связок “и” , “или” не только обозначения высказываний, но и обозначения понятий, объектов, множеств объектов; (4) отображать смысловую сруктуру фраз со словами “понятие”, “термин”; (5) рассматривать нетрадиционные функции, аргументами и/или значениями которых могут быть множества объектов, множества понятий, СП текстов, множества СП текстов.

Таким образом, полученные в главах 2 - 4 результаты открывают реальные перспективы разработки семантического сетевого языка нового поколения, выразительные возможности которого будут значительно ближе к выразительным возможностям ЕЯ по сравнению с возможностями языка UNL, предложенного Х. Учидой и М. Жу (Uchida, Zhu, Della Senta 1999; Uchida, Zhu 2001; Zhu, Uchida 2002). Этот вывод опубликован в работе (Fomichov 2004).

5.4. Новые возможности для построения онтологий предметных областей и разработки языков представления знаний

5.4.1. Онтологии и их значение для глобальных информационных сетей

Работа прикладной интеллектуальной системы существенным образом зависит от ее базы знаний. Еще с 1970-х годов развиваются исследования по разработке все более совершенных языков представления знаний (ЯПЗ) в интеллектуальных системах. Важный класс ЯПЗ составляют терминологические языки представления знаний. В отличие от языка логики предикатов, в терминологических ЯПЗ есть специальные единицы, являющиеся обозначениями понятий, и есть средства построения из таких единиц составных обозначений понятий.

Например, во второй половине 1980-х годов и начале 1990-х годов в Германии по заказу фирмы IBM был реализован проект LILOG (LInguistics & LOGic), реализованный институтом представления знаний (Штутгарт) совместно с несколькими университетами. В рамках этого проекта были разработаны новые средства для представления знаний о мире и для проектирования ЕЯ-диалоговых систем (Pletat, von Luck 1990; Pletat 1991; Herzog, Rollinger 1991).

Терминологические языки представления знаний называют также в англоязычной литературе KL-ONE-like languages, потому что первым терминологическим языком представления знаний был язык KL-ONE,

разработанный в конце 1970-х – начале 1980-х (Brachman, Schmolze 1985). Одним из потомков языка KL-ONE стал язык L_{LILOG} , разработанный в проекте LILOG.

Развитие исследований по разработке терминологических ЯПЗ привело в 1990-е годы к появлению нового значения понятия “онтология“. Согласно Большому энциклопедическому словарю под редакцией А.М. Прохорова, изданному в 2000-м году, в философии онтологией называется учение о бытии (в отличие от гносеологии – учения о познании), в котором исследуются всеобщие основы, принципы бытия, его структура и закономерности.

В работах по информатике онтология понимается как спецификация (т.е. описание) концептуализации (Gruber 1993; Guarino 1998; FIPA 1998b). Термин “концептуализация” используется для указания способа, которым интеллектуальная система структурирует знания о мире, восприятие мира. Спецификация концептуализации дает значения терминам из словаря, используемого интеллектуальной системой для обработки знаний и взаимодействия с другими интеллектуальными системами.

На протяжении последнего десятилетия можно было наблюдать постоянный рост интереса исследователей к построению и изучению онтологий. Причина этого заключается в том, что ученые и разработчики компьютерных систем стали заинтересованы в повторном использовании или/и разделении (совместном использовании) знаний системами. Например, в нашей стране опубликованы, в частности, работы (Гаврилова, Хорошевский 2000; Гаврилова, 2001; Нариньяни 2001, 2002) и обстоятельный обзор (Смирнов, Пашкин, Шилов, Левашова 2002а, 2002б), посвященные созданию и применению онтологий.

Созданию и использованию онтологий для разработки системы автоматизированного контроля смысловой полноты технической документации, описывающей поведение оператора летного экипажа и бортовой аппаратуры в различных полетных режимах, посвящены работы (Добров, Лукашевич и др. 2004; Лукашевич 2004).

В основе исследования лежит частичный семантико-синтаксический анализ документации технических систем. Сущность анализа заключается в выделении

понятий, называемых в тексте или ассоциированных с упоминаемыми в тексте объектами и действиями. Выделенные понятия сопоставляются с фреймовой моделью предметной области, отражающей иерархическую сеть понятий и часть их взаимосвязей. Такая модель названа АвиаОнтологией.

Компьютерные системы используют различные понятия для описания предметных областей. Эти различия создают трудности для применения знаний одной системы в другой системе. Предположим, что мы построим онтологии, которые могут служить основой разработки баз знаний многих систем. В этом случае различные системы смогут применять общую терминологию, а это облегчит разделение и неоднократное использование знаний.

Примерно с начала 1990-х годов исследователями многих стран ведется поиск эффективных формальных подходов к построению онтологий и средств программной реализации онтологий. В 1990-е годы и начале 2000-х годов наибольшую известность получили компьютерные онтологии CYC (Lenat 1995; CYC 2001), LOOM (Loom 2001), OIL (Fensel и др., 2000; Horrocks 2000), DAML (DAML 2001).

В 1990-е годы исследования, направленные на создание терминологических ЯПЗ и применение их к построению онтологий, привели к возникновению нового научного направления в области математической теории прикладных интеллектуальных систем – дескриптивной логики. Общим для различных вариантов логик, разработанных в рамках данного направления, является то, что важный подкласс рассматриваемых правильно построенных формул образуют простые и составные обозначения понятий.

В связи с развертыванием широкомасштабного проекта Семантической Всемирной Паутины (Semantic Web) роль онтологий еще более возросла. Исследования по терминологическим ЯПЗ и дескриптивным логикам, опыт разработки в 1990-е годы компьютерных онтологий, а также разработка (в рамках подготовки проекта Семантической Паутины) языка описания метаданных об информационных ресурсах RDF (Resource Description Framework) позволили специалистам Западной Европы и США создать языковую систему для построения

онтологий DAML + OIL (Horrocks, van Harmelen, Patel-Schneider 2001). Эта система, получившая название языка для разметки онтологий (the ontological markup language), сегодня рассматривается специалистами как важная часть теоретического фундамента Семантической Паутины. Язык DAML + OIL основывается на идеях логики предикатов первого порядка и идеях, реализованных в фреймоподобных ЯПЗ.

Несмотря на интенсивность исследования проблемы, выразительные возможности разработанных формальных языков для построения онтологий являются довольно ограниченными. В частности, это относится к построению семантических представлений (СП) определений понятий, в которых либо упоминаются множества или назначения объектов или цели интеллектуальных систем, либо содержатся ссылки на смысл фраз и более крупных частей дискурса.

5.4.2. Анализ возможностей представления знаний о предметных областях средствами СК-языков

Авторы многих публикаций по онтологиям отмечают, что перспективный путь автоматизации конструирования онтологий заключается в разработке и использовании лингвистических процессоров для извлечения знаний из накопленных во всех областях текстов на естественном языке – монографий, статей, научных и технических отчетов, юридических документов и т.д.

Поэтому необходимы значительно более мощные (по сравнению с имеющимися) формальные средства для построения СП (а) ЕЯ-определений понятий и (б) предложений и дискурсов на ЕЯ, выражающих знания о предметной области.

В этой связи представляется целесообразным указать некоторые наиболее важные выразительные возможности СК-языков с точки зрения построения СП определений понятий и формального отображения знаний о предметных областях.

Сначала покажем (примеры 1 – 3), что СК-языки позволяют моделировать выразительные механизмы основных языков дескриптивной логики (другими словами, терминологических ЯПЗ).

Пример 1. Пусть Т1 = “Тинейджер – это человек в возрасте от 13 до 19 лет.” . Тогда на языке L_{LILOG} содержание текста Т1 может быть представлено выражением

teenager = person ∩ with-feature age in [13..19] .

Первым возможным К-представлением (КП) текста Т1 является выражение

*((тинейджер ≡ человек * (Возраст, x)) ∧ ¬ Меньше (x, <13, год>)*

∧ ¬ Больше (x, <19, год>)).

Вторым возможным КП текста Т1 является формула

*((тинейджер ≡ человек * (Возраст, x)) ∧ Диапазон (x, год, 13, 19))* .

Третье возможное КП текста Т1: *Определение (тинейджер, x, (Явл (x, тинейджер) ≡ (Явл (x, человек) ∧ Диапазон (Возраст (x), год, 13, 19))))* .

Пример 2. В языке L_{LILOG} семантическое представление текста Т2 = “Порше 911 – это автомобиль с двумя дверями типа кабрио” может выглядеть следующим образом (Pletat 1991) : *Constant Porsche-911: and (car, doors : {2}, body: {cabrio})*

Возможным К-представлением текста Т2 является выражение

*(Porsche-911 ≡ car * (Doors-number, 2) (Body-type, cabrio))* .

Пример 3. На языке L_{LILOG} информация о том, что различают типы корпуса автомобилей cabrio, coupe, hatch-back, sedan , формально представляется выражением

Sort body-type;

Atoms cabrio, coupe, hatchback, sedan. .

Эту же информацию можно представить с помощью следующего выражения некоторого СК-языка:

Kinds (body_type, (cabrio ∨ coupe ∨ hatch-back ∨ sedan)) .

Мы видим, что логические связки “и”, “или” позволяют соединять не только семантические представления высказываний, но и обозначения понятий.

Анализ показывает, что выразительные возможности языка L_{LILOG} и других разработанных терминологических ЯПЗ являются довольно ограниченными. Это касается построения составных обозначений понятий и целей интеллектуальных систем, описания множеств, отображения содержания ЕЯ-текстов со ссылками на смысл фраз и более крупных частей дискурса. В связи с этим рассмотрим некоторые возможности использования СК-языков в подобных случаях.

Пример 4. В предыдущем параграфе рассматривалось определение Def1 = “A flock (английский язык) – это большое количество птиц или млекопитающих (например, овец или коз), собирающихся вместе с определенной целью, такой, как питание, миграция или оборона” и была построена цепочка *Expr1* , являющаяся СП этого определения.

Определение Def1 взято из определенной книги, опубликованной в определенном году определенным издательством. СК-языки позволяют строить СП определений и других фрагментов знаний в объектно-ориентированной форме, отражая их внешние связи. Например, объектно-ориентированное СП определения Def1 может являться выражением

*нек информ-объект * (Вид, определ)(Содержание1, Expr1)*
*(Источник1, нек словарь * (Название, ‘Longman Dictionary*
of Scientifc Usage’)(Издательство, (Longman-Group-Limited/Harlow ∧
Russky-Yazyk-Publishers/Moscow))(Город, Москва)(Год, 1989)) .

Пример 5. Пусть ТЗ — определение «Евстахиева труба – это канал, ведущий от среднего уха к глотке ». ТЗ можно поставить в соответствие, в частности, следующую К-цепочку, интерпретируемую как СП текста ТЗ:

Определение1 (евстахиева-труба, русск-яз, канал2, x1,
∃z (чел) Вести1 (x1, нек среднее-ухо ∗ (Часть, z), нек глотка ∗ (Часть, z))) .

Пример 6. Пусть Т4 = «Сфигмоманометр — прибор, предназначенный для измерения кровяного давления», тогда Т4 может иметь следующее КП:

(сфигмоманометр ≡ прибор ∗ (Назначение, измерение1 ∗
(Парам, кровяное-давление)(Субъект, произв чел))) .

Семантическая единица *Назначение* в этом КП обозначает бинарное отношение. Если пара (A, B) принадлежит этому отношению, то A является физическим объектом, а B - формальным семантическим аналогом выражения, описывающего назначение этого физического объекта.

Пример 7. Пусть T5 — определение «Тромбин — это фермент, который помогает преобразовать фибриноген в фибрин во время коагуляции». Тогда следующая К-цепочка является возможным КП T5:

*(тромбин ≡ фермент ∗ (Назначение, оказание-помощи * (Действие, преобразование1* (Исх-объект, нек фибриноген)(Результат1, нек фибрин)(Процесс, нек коагуляция)))).*

Примеры, рассматриваемые ниже, покажут выразительные возможности стандартных К-языков в отношении описания семантической структуры дискурсов.

Пример 8. Рассмотрим текст T6 = «Адениновая основа на одной нити ДНК связана только с тиминовой основой противоположной нити ДНК. Подобным же образом, цитозиновая основа связана только с гуаниновой основой противоположной нити ДНК».

Для построения КП T6 полезно следующее пояснение. Молекула дезоксирибонуклеиновой кислоты (молекула ДНК) содержит тысячи нуклеотидов (комбинаций из трех основных элементов: дезоксирибозы, фосфатов и основы). Существует четыре вида основ: аденин, гуанин, цитозин и тимин. Нуклеотиды ДНК-молекулы образуют цепочку, которая формирует две длинные нити, сплетенные друг с другом. Приняв во внимание это замечание, с первым предложением из T6 можно связать КП A1 вида

$$\forall x1\ (\text{днк-молекула})\ (\text{Связывать1 (произв основа1} * (\text{Явл, аденин})\ (\text{Часть, произв нить1} * (\text{Часть, x1}) : y1) : z1, \text{нек основа1} * (\text{Явл, тимин})\ (\text{Часть, нек нить1} * (\text{Часть, x1})\ (\text{Противоположн, y1}) : y2) : z2) \wedge \neg \exists z3\ (\text{основа1})\ (\text{Явл}\ (z3, \text{тимин}) \wedge \text{Часть}\ (z3, y2) \wedge \text{Связывать}\ (z1, z3)) : P1 \quad (5.4.1)$$

В строке *A1* вида (5.4.1) переменные y1 и y2 используются как метки описаний двух нитей произвольной молекулы ДНК x1; переменные z1, z2, z3 помечают основы. Переменная *P1* (имеет сорт «смысл сообщения») используется для обозначения семантического представления первого предложения из Т6. Это позволяет построить компактное СП второго предложения Т6, так как вхождение выражения «подобным же образом» во второе предложение из Т6 означает ссылку на смысл первого предложения. В частности, второе предложение Т6 в контексте первого предложения может иметь К-представление *A2* вида

*(Подобно (P1, P2) ∧ (P2 ≡ ∀x1 (днк-молекула) (Связывать (произв основа1 * (Явл, цитозин) (Часть, произв нить1 * (Часть, x1) : y3) : z4, нек основа1 * (Явл, гуанин) (Часть, нек нить1 * (Часть, x1)(Противоположн, y3) : y4) : z5) ∧ ¬ ∃z6 (основа1) (Явл (z6, гуанин) ∧ Часть (z6, y4) ∧ Связывать (z4, z6)))))* . (5.4.2)

Таким образом, с текстом Т6 можно связать К-цепочку *A3* вида *(A1 ∧ A2)*, где *A1* и *A2* — цепочки видов (5.4.1) и (5.4.2) соответственно. Полученную строку можно рассматривать как возможное КП для текста Т6.

Цепочка *A3* иллюстрирует важную возможность, предоставляемую стандартными К-языками: можно помечать переменными фрагменты цепочек, являющиеся семантическими представлениями сообщений, неопределенных форм глаголов или вопросов. Эта возможность позволяет эффективно описывать структурированные значения дискурсов со ссылками на смысл фрагментов, являющихся сообщениями, целями (советами, пожеланиями) или вопросами. На наличие подобных ссылок в дискурсах часто указывают слова и словосочетания: «эта рекомендация», «например», «то есть», «рассмотренная идея», «другими словами» и ряд других.

Построенное КП *A3* для Т6 иллюстрирует еще одну особенность СК-языков: символ «≡» соединяет переменную *P2* и семантическое представление предложения.

Пример 9. Пусть Т7 = «Термин «цитозин» используется в генетике». Структурированное значение Т7 может быть представлено в виде К-цепочки

*Используется(нек понятие * (Название1, «цитозин»), генетика) .*

Следовательно, СК-языки позволяют описывать структурированные значения, предложений со словами «понятие», «термин» и т. п.

Таким образом, аппарат СК-языков открывает возможности построения новых терминологических языков представления знаний с очень большой выразительной силой и, как следствие, возможности разработки онтологий в произвольных предметных областях, поскольку дает мощные средства построения составных обозначений объектов, множеств, понятий, целей интеллектуальных систем, а также средства отображения ссылок на ранее упомянутые сущности и на смысл предшествующих фраз и более крупных частей связного текста. Перечисленные особенности СК-языков являются основными преимуществами предложенного подхода к формализации предметных областей по сравнению с языком разработки онтологий DAML + OIL, использующимся в проекте Семантической Паутины.

Поэтому одним из возможных применений аппарата СК-языков является совершенствование средств построения онтологий в проекте Семантической Паутины.

5.4.3. Разработка новых языков представления знаний для решения информационно-сложных задач

Анализ научной литературы говорит о том, что существует глубокая связь между проблемой формального описания содержания ЕЯ-текстов и проблематикой разработки информационных технологий (ИТ), основанных на представлении и обработке сложноструктурированных знаний. Как подчеркивается в работе (Кузин 2004), до последнего времени разработчики ИТ для автоматизации решения разнообразных практических задач основное внимание уделяли поиску методов решения алгоритмически-сложных задач. При этом не было широко осознано существование класса информационно-сложных задач, для которых необходимы языки представления знаний о проблемной среде с большими выразительными возможностями. Такие языки должны, в частности, позволять отображать большое

количество различных смысловых аспектов проблемной среды, обрабатывать информацию с разных точек зрения, строить многоуровневые обобщения и интегрировать информацию.

Анализ публикаций Е.С. Кузина по технологии функционально-ориентированного проектирования (ФОП-технологии) программных систем (Кузин 1996 - 2004) показывает, что основные идеи этих публикаций тесно взаимосвязаны **с** понятием онтология. Несмотря на то, что термин онтология не используется, по-существу, в том же смысле применяется термин модель проблемной среды, обозначающий целостную систему взаимосвязанных знаний о проблемной среде.

В качестве одного из ключевых направлений исследования проблематики автоматизации решения информационно-сложных задач указывается создание адекватной теории отображения объективного мира в программной системе (ПС) и разработка на этой основе языков представления знаний (ЯПЗ) о проблемной среде с очень высокими выразительными возможностями.

Список требований к таким ЯПЗ включает: (1) возможности отображения очень большого числа различных смысловых аспектов проблемной среды, которые являются существенными для решения задачи и, следовательно, должны конструктивно учитываться в ПС; (2) наличие средств представления не только детализированной информации о проблемной среде, но и более крупных информационных образований, которые получаются путем многоуровневых обобщений и интеграции информации и позволяют анализировать информацию под разными углами зрения.

В работах (Кузин 2003, 2004) описываются основные черты разработанного языка описания декларативных знаний (ЯОДЗ), удовлетворяющего перечисленным требованиям. Этот язык был создан в рамках новой семантической теории, названной конструктивной семантикой. Разработка ЯОДЗ была использована для создания системы управления базой знаний (СУБЗ), реализованной с помощью инструментальных языков C++ и Java в операционной системе Windows.

Построенная СУБЗ нашла успешное применение в опытно-конструкторских разработках, на ее основе созданы: (а) автоматизированная система “Персонал”,

предназначенная для ведения индивидуализированной информации о личностях, организациях и других объектах и настраиваемая на конкретные применения; (б) системы информационной поддержки управления поставками сложных компьютерных изделий и их сопровождения в течение жизненного цикла (Кузин 2003).

Сопоставление выразительных возможностей СК-языков и основных черт ЯОДЗ, описанных в (Кузин 2003), позволяет говорить о том, что СК-языки обладают всеми ценными свойствами ЯОДЗ. В частности, СК-языки позволяют: (а) формально различать конкретные объекты и понятия (типы почти всех понятий начинаются с вертикальной стрелки); (б) задавать семантические ограничения на аргументы отношений (для этого используется отображение *tp* , являющееся компонентом концептуально-объектной системы); (в) строить составные обозначения множеств объектов, включающие информационную единицу *все* и обозначения чисел, указывающих количество элементов множества.

В то же время есть ряд важных выразительных механизмов, реализованных в СК-языках, которые, насколько можно судить по работам (Кузин 2003, 2004), отсутствуют в ЯОДЗ. В частности, СК-языки позволяют: (а) строить составные обозначения целей интеллектуальных систем и назначений вещей, составные обозначения понятий и более сложные по сравнению с ЯОДЗ составные обозначения множеств, (б) представлять содержание фраз со словами “понятие”, “термин” и содержание дискурсов со ссылками на смысл фраз и более крупных фрагментов дискурса, (в) строить СП определений и других фрагментов знаний в объектно-ориентированной форме, отражая их внешние связи (см. выше пример 4).

Поэтому можно предположить, что научные результаты, полученные в главах 2 - 4 и в данной главе, окажут позитивное влияние на исследования в области автоматизации решения информационно-сложных задач в качестве теоретической базы для разработки ЯПЗ с большими выразительными возможностями, близкими к возможностям ЕЯ.

5.5. Возможности использования СК-языков в проектировании интеллектуальных информационно-поисковых и вопросо-ответных Интернет-систем нового поколения

5.5.1. Актуальность разработки вопросо-ответных Интернет-систем

Хотя сегодня системы информационного поиска в сети Интернет используются сотнями тысяч и миллионами людей во всем мире (в зависимости от языка запросов), их эффективность еще далеко не соответствует пожеланиям, по-видимому, большей части пользователей. Поэтому актуальной остается проблема совершенствования информационного поиска в сети Интернет.

Важный аспект проблемы заключается в том. что современные поисковые Интернет-системы обрабатывают только тематические запросы. Ответом на такой запрос обычно являются ссылки на большое количество документов – от десятков до десятков тысяч. Между тем, конечным пользователям очень часто нужно получить ответ на вопрос, и такой ответ должен быть числовым значением параметра (например, датой, номером телефона), коротким фрагментом текста или несколькими короткими фрагментами. Примерами таких вопросов являются “Сколько может стоить двухместный номер в трехзвездочном отеле в Будапеште?” и “Сколько провинций в Канаде?”.

В связи с этим как в отечественной научной литературе (см., например, Харин 2002), так и в зарубежных публикациях ставится проблема разработки информационно-поисковых Интернет-систем нового поколения, способных не только осуществлять тематический поиск документов (причем более точно и полно по сравнению с существующими системами), но и отвечать на вопросы конечных пользователей. Решение этой задачи является одной из основных целей реализации широкомасштабного проекта Семантической Всемирной Паутины (Semantic Web 2001).

Выразительные возможности класса СК-языков, исследовавшиеся в главе 4 и в предыдущих параграфах данной главы, показывают, что сегодня аппарат СК-

языков является наиболее удобным инструментом с точки зрения решения следующих задач проектирования информационно-поисковых Интернет-систем нового поколения, способных не только осуществлять тематический поиск документов, но и отвечать на вопросы конечного пользователя: (а) построения семантического представления (СП) запроса пользователя, (б) построения СП фрагмента анализируемого ЕЯ-текста (длина фрагмента может быть сколь угодно большой).

С одной стороны, выше было показано, что класс СК-языков обладает наибольшими выразительными возможностями по сравнению с другими известными подходами к формальному представлению содержания ЕЯ-текстов. С другой стороны, в главе 4 была высказана гипотеза о том, что СК-языки удобны для построения СП произвольных текстов деловой прозы.

Проиллюстрируем часть важных выразительных возможностей СК-языков на примерах, относящихся к двум актуальным научно-техническим проблемам. Первая проблема заключается в разработке методов и базирующихся на них компьютерных систем, обеспечивающих общественности в нащей стране доступ к государственным информационным ресурсам. Второй проблемой является разработка юридических полнотекстовых баз данных.

5.5.2. Электронные библиотеки и проблема обеспечения доступа общественности к государственным информационным ресурсам

Развитие гражданского общества в нашей стране существенно зависит от степени доступности государственных информационных ресурсов. Огромную роль в обеспечении доступа общественности к государственным информационным ресурсам должны сыграть электронные библиотеки (Елепов, Марчук, Бобров, Константинов 1997; Когаловский 2000; Калиниченко, Скворцов и др. 2000; Когаловский, Новиков 2000; Марчук, Осипов 2000; Антопольский 2002; Антопольский, Майорович, Чугунов 2005). Электронным библиотекам (ЭлБ) отводится важная роль в федеральной целевой программе “Электронная Россия

(2002 – 2010 годы)". Для обеспечения подлинной широты доступа пользователей ЭлБ к информационным ресурсам необходимы естественно-языковые интерфейсы (ЕЯ-интерфейсы), образующие важный подкласс лингвистических процессоров (ЛП). Кроме того, необходимы ЛП, способные преобразовать текстовый документ на естественном языке (ЕЯ) или фрагмент текстового документа в формальную структуру, отражающую его содержание, или смысл (семантическое представление документа или его фрагмента), а затем сравнить содержание запроса пользователя с этим семантическим представлением (СП).

Одной из первоочередных теоретических задач, связанных с разработкой ЛП указанных видов, является создание эффективных методов формального описания содержания (или смысла, или смысловой структуры) произвольных или почти произвольных текстов деловой прозы на русском и английском языках. Широкий спектр новых возможностей в этом направлении предоставляют СК-языки. Рассмотрим только один пример, далеко не исчерпывающий все такие возможности.

Пример. Пусть Т1 = "Какие решения правительства за 2000 – 2004 годы направлены на улучшение завоза продовольствия или расширение строительства жилья на северном побережье Восточной Сибири?". Тогда СП запроса Т1 может являться следующим выражением Е1 некоторого СК-языка:

*Вопрос (S1, (Качеств-состав(S1, решение1) ∧ Описание (произвольн решение1 * (Элемент, S1) : x1, (Принято (x1, Правительство (Россия), t1) ∧ Год (t1, (2000 ∨ 2001 ∨ 2002 ∨ 2003 ∨ 2004)) ∧ Цель (x1, (улучшение1 * (Процесс1, завоз1 * (Объект1, нек множ * (Качеств-состав, продукт-питания))(Место2, нек побережье * (Относит-расположение, север)(Регион, Восточн-Сибирь) : x2) ∨ расширение1 * (Процесс1, строительство * (Объект1, нек множ * (Качеств-состав, дом 1 * (Вид1, жилой)) : S3)(Место0, x2)))))).*

В построенном выражении Е1 отражены, в частности, следующие особенности предлагаемого подхода к формализации содержания ЕЯ-текстов.

1. Используются информационные единицы *нек* ("некоторый"), *произвольн* ("произвольный").

2. Можно строить составные обозначения: (а) понятий, характеризующих объекты: *дом 1 * (Вид1, жилой)* ; (б) понятий, характеризующих множества объектов: *множ * (Качеств-состав, продукт-питания)* ,
(в) множеств объектов: *нек множ * (Качеств-состав, продукт-питания)* ,
*нек множ * (Качеств-состав, дом 1 * (Вид1, жилой)) : S3*.
3. Можно присоединять (с помощью двоеточия) метки к составным обозначениям объектов. Например, в выражении

*нек побережье * (Относит-расположение, север)(Регион, Восточн-Сибирь) : x2*

переменная *x2* является меткой, поставленной в соответствие северному побережью Восточной Сибири.

4. Можно строить составные обозначения целей:

*улучшение1 * (Процесс1, завоз1 * (Объект1, нек множ * (Качеств-состав, продукт-питания))(Место2, нек побережье * (Относит-расположение, север)(Регион, Восточн-Сибирь) : x2)).*

5. Можно представлять сложные цели, соединяя логическими связками ∧ (и), ∨ (или) обозначения более простых целей:

*(улучшение1 * (Процесс1, завоз1 * (Объект1, нек множ * (Качеств-состав, продукт-питания) : S2)(Место2, нек побережье * (Относит-расположение, север)(Регион, Восточн-Сибирь) : x2)) ∨ расширение1 * (Процесс1, строительство * (Объект1, нек множ * (Качеств-состав, дом 1 * (Вид1, жилой)) : S3)(Место0, x2))).*

6. Логические связки ∧ (и), ∨ (или) могут соединять обозначения объектов (а не только высказываний, как в логике предикатов): *(2000 ∨ 2001 ∨ 2002 ∨ 2003 ∨ 2004).*
7. СК-языки позволяют связать с обозначением множества простое или составное обозначение понятия, являющегося концептуальной характеристикой каждого элемента этого множества:

Качеств-состав(S1, решение1), Качеств-состав(S2, продукт-питания),
*Качеств-состав(S3, дом 1 * (Вид1, жилой)).*

Глава 6
МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЛИНГВИСТИЧЕСКОЙ БАЗЫ ДАННЫХ

Построена широко применимая формальная модель лингвистической базы данных (ЛБД), т.е. базы данных, которая содержит информацию, используемую алгоритмом семантико-синтаксического анализа текстов для построения по ЕЯ-тексту его семантического представления (СП). Эта модель описывает логическую структуру ЛБД ЕЯ-интерфейсов баз данных и других прикладных компьютерных систем. В предложенной модели выражения стандартных К-языков (СК-языков) используются в качестве семантических единиц, соответствующих лексическим единицам, и в качестве СП естественно-языковых текстов. Предложена новая структура данных (названная матричным семантико-синтаксическим представлением текста), используемая в качестве промежуточной формы отображения семантической структуры входного ЕЯ-текста, для последующего построения СП текста. Разработан предметно-независимый метод преобразования входного ЕЯ-текста (вопроса, команды, сообщения) из подъязыка русского языка в его СП, являющееся выражением некоторого СК-языка.

6.1. Постановка задачи

Огромный рост парка персональных компьютеров, разработка многочисленных баз данных (БД) и баз знаний (БЗ) привели к тому, что к этим БД и БЗ получил доступ широкий круг пользователей, не являющихся программистами и не изучавших какие-либо формальные языки. Тем не менее, у этих пользователей возникает необходимость взаимодействия с такими БД или БЗ для решения профессиональных задач либо в повседневной жизни. Стремительное развитие сети Интернет существенно усиливает эту тенденцию: у очень большого

количества людей появляется желание получить какую-то информацию из источников, удалённых от терминала пользователя на тысячи километров.

Все эти факторы способствовали во второй половине 1990-х годов – первой половине 2000-х годов значительному усилению внимания к разработке и применению естественно-языковых интерфейсов (ЕЯ-интерфейсов) БД и БЗ. Одним из свидетельств возникновения новой ситуации в области проектирования и применения ЕЯ-интерфейсов является разработка научно-исследовательским центром фирмы Microsoft® англоязычного интерфейса баз данных English Query. С конца 1990-х годов этот интерфейс поставляется вместе с сервером SQL 6.5 или SQL 7.0 и может встраиваться в состав Web-узлов пользователей. Интерфейс English Query позволяет задавать вопросы к реляционной БД на структурно-ограниченном английском языке. Предусмотрены средства адаптации интерфейса к новым предметным областям (English Query 2000; Snyder 2001)..

Учитывая сказанное выше, можно сделать вывод о том, что насыщенность персональными компьютерами крупнейших промышленных и научных центров нашей страны, бурное развитие сети Интернет в последние годы говорят о большой актуальности проблемы создания интеллектуальных интерфейсов, предоставляющих массовому пользователю возможность взаимодействия с базами данных и/или базами знаний на структурно-ограниченном русском языке (РЯ) (Попов 1999).

Исследования по разработке ЛП для РЯ и некоторых других языков (в первую очередь, английского и французского) развиваются в нашей стране в течение более чем трёх десятилетий. В конце 1990-х – начале 2000-х годов потребности практики в развитии лингвистических информационных технологий (ЛИТ) привели к появлению ряда новых интересных проектов ЛП. Такие проекты были реализованы, в частности, в Институте проблем информатики РАН (Кузнецов и др. 2000; Кузнецов, Мацкевич 2001, 2003; Kuznetsov, Matskevich 2002), Институте проблем передачи информации РАН (Богуславский, Иомдин и др. 2000), распределенным коллективом разработчиков интеллектуальной метапоисковой системы “Сириус” из Института программных систем РАН, Институтf системного

анализа РАН и Российского государственного университета дружбы народов (Куршев, Осипов и др. 2002; Осипов, Куршев и др. 2003; Завьялова 2004; Тихомиров, Осипов и др. 2005), в Институте прикладной математики им. М.В. Келдыша (Кулагина 1998, 2001; Агранат, Кулагина 2000), РосНИИ информационных технологий и систем автоматизированного проектирования (Курбатов, Попов 2001; Курбатов 2002), РосНИИ искусственного интеллекта (Жигалов, Соколова 2001; Жигалов 2002; Жигалов В.А., Жигалов Д.В. 2002), ВИНИТИ (Кузнецов, Солнцева, Деревянкин, Закамская 2001), на факультете вычислительной математики и кибернетики МГУ им. М.В. Ломоносова (Мальковский, Шикин 1998; Болдасов, Соколова 2002), в МВТУ им. Н.Э. Баумана (Смирнов, Андреев, Березкин, Брик 1997), фирмой Abbey Software House (Перцова, Перцов 2002), ООО “Гарант-Парк-Интернет” (Ермаков 2002; Киселев, Ермаков, Плешко 2004), в Санкт-Петербургском государственном университете и Санкт-Петербургском экономико-математическом институте РАН (Тузов 2001; Каневский, Тузов 2002; Лезин, Тузов 2003), в Институте высокопроизводительных вычислений и баз данных Санкт-Петербургского государственного технического университета (Писарев, Самсонова 2002).

Важное значение для развития в нашей стране ЛИТ имеет также реализация нескольких проектов компьютерных семантических словарей (Лахути, Рубашкин 1998 - 2000; Леонтьева 2000 - 2002; Леонтьева, Семенова 2001 – 2003; Мальковский, Соловьев 2002 – 2004; Пацкин 2004).

Это продвижение в области проектирования и применения ЛП во многом было подготовлено теоретическими результатами и опытом проектирования ЛП, полученными в 1980-х годах и начале 1990-х годов, в частности, Ю.Д. Апресяном, И.М. Богуславским (Апресян, Богуславский и др. 1981, 1989), Г.Г. Белоноговым, И.А. Большаковым, В.М. Брябриным, А.В. Гладким (Гладкий 1985), Б.Ю. Городецким, В.И. Дракиным, А.П. Ершовым, Е.С. Кузиным, И.П. Кузнецовым, О.С. Кулагиной (Кулагина 1979, 1996), С.С. Курбатовым, Д.Г. Лахути, В.Ш. Рубашкиным (Рубашкин 1989; Лахути, Рубашкин 1993), Н.Н. Леонтьевой (Леонтьева 1981, 1986), Л.В. Литвинцевой, Ю.Я. Любарским (Любарский 1990),

М.Г. Мальковским (Мальковский 1985), Л.И. Микуличем, А.С. Нариньяни, А.П. Новоселовым, Г.С. Осиповым (Осипов 1990, 1997), Н.В. Перцовым, Р.Г. Пиотровским, Д.А. Поспеловым, Э.В. Поповым, А.Б. Преображенским (Попов 1982, 1987; Попов, Преображенский 1990; Дракин, Попов, Преображенский 1988), Г.В. Рыбиной, Г.В. Сениным, А.М. Степановым, В.А. Тузовым (Тузов 1984), В.С. Файном, Г.К. Хахалиным, В.Ф. Хорошевским, Г.С. Цейтиным, Л.Л. Цинманом (Апресян, Цинман 1982, Цинман 1986), а также рядом других исследователей.

Несмотря на появление в конце 1990-х – начале 2000-х годов новых примеров применения на практике в нашей стране лингвистических процессоров, можно констатировать, что в этот период в целом недостаточно внимания уделялось разработке эффективных формальных средств и методов проектирования ЛП.

Наибольшие трудности при разработке ЛП связаны с выполнением преобразования “ЕЯ-текст ➔ Семантическое представление (СП) текста”, где под СП ЕЯ-текста понимается формальная структура, отражающая содержание (или смысл) ЕЯ-текста. Однако анализ как отечественных, так и зарубежных публикаций показывает, что при разработке преобразователей ЕЯ-текстов в СП текстов крайне недостаточно используются формальные средства. Это выражается в неформальном и фрагментарном описании (а) структуры лингвистической базы данных (ЛБД), т.е. базы данных с морфологической и семантико-синтаксической информацией о лексических единицах, используемой алгоритмом семантико-синтаксического анализа текстов для построения по ЕЯ-тексту его семантического представления (СП) и (б) методов обработки информации основными подсистемами преобразователя “ЕЯ-текст ➔ СП текста”.

Основная часть исследований по разработке ЕЯ-интерфейсов и ЛП других видов была реализована для английского языка, синтаксис которого существенно отличается от синтаксиса русского языка (РЯ). В отличие от английского языка, РЯ относится к классу сильно флективных языков. Это выражается в том, что слова РЯ могут изменяться; например, окончания существительных меняются в зависимости от грамматического падежа и числа, окончания глаголов зависят от времени и лица и т.д. Другой важной особенностью РЯ является весьма свободный

порядок слов; например, в предложениях с глаголом в действительном залоге подлежащее может располагаться как перед сказуемым, так и после сказуемого.

Чрезвычайно существенно то, что полные описания информационного и программного обеспечения англоязычных ЛП, как правило, недоступны специалистам в нашей стране. Кроме того, одним из следствий экономической ситуации, сложившейся в 1990-е годы в нашей стране, является отсутствие даже в центральных библиотеках огромного количества публикаций в области разработки ЛП, опубликованных за рубежом в 1990-е и 2000-е годы на английском и некоторых других языках. Все это серьезно затрудняет подготовку в нашей стране специалистов в области проектирования ЛП и сужает возможности принятия оптимальных проектных решений, приводит к дополнительным трудозатратам на разработку ЛП.

Таким образом, актуальной является проблематика разработки методов формального описания структуры ЛБД, а также таких методов семантико-синтаксического анализа текстов из представляющих практический интерес подъязыков русского языка, которые более широко используют формальные средства описания входных, промежуточных и выходных данных по сравнению с известными методами.

Разработка ЛП многих видов, например, ЕЯ-интерфейсов больших БД, отличается значительной трудоемкостью. В связи с этим в параграфе 1.1. данной книги была выдвинута гипотеза о том, что в долговременной перспективе сокращению затрат и времени на разработку семейства ЛП в рамках одной организации или нескольких взаимодействующих организаций будет способствовать реализация в проектировании информационного и алгоритмического обеспечения ЛП следующих двух принципов:

(1) *принципа стабильности* используемого языка семантических представлений (ЯСП) по отношению к многообразию решаемых задач, многообразию предметных областей и многообразию программных сред (стабильность понимается как использование единой системы правил для построения

конструкций ЯСП и варьируемого набора первичных информационных единиц, определяемого предметной областью и решаемой задачей);

(2) *принципа преемственности* алгоритмического обеспечения ЛП на основе использования одной или нескольких совместимых формальных моделей лингвистической БД и единых формальных средств предствления промежуточных и окончательных результатов семантико-синтаксического анализа ЕЯ-текстов по отношению к многообразию решаемых задач, предметных областей и программных сред (преемственность понимается как многократное использование в различных лингвистических процессорах алгоритмов, реализуемых основными подсистемами ЛП).

Теоретическую основу для реализации принципа стабильности используемого ЯСП создают результаты, изложенные в главах 2 – 4 данной монографии. В главе 3 определен класс стандартных К-языков (СК-языков), позволяющих строить СП ЕЯ-текстов в произвольных предметных областях.

Данная глава базируется на результатах, отраженных в предыдущих главах и направлена на создание значительной части предпосылок для реализации принципа преемственности при проектировании алгоритмического обеспечения лингвистических процессоров..

В данной главе и главе 7 ставится и решается задача разработки нового метода преобразования ЕЯ-текста в семантическое представление для проектирования семантико-синтаксических анализаторов текстов из представляющих практический интерес подъязыков РЯ. С этой целью ставятся и достигаются следующие цели:

1. Формализовать структуру лингвистической базы данных, позволяющей устанавливать возможные смысловые отношения, в частности в сочетаниях «Глагол + Предлог + Существительное», «Глагол + Существительное», «Существительное1 + Предлог + Существительное2», «Число + Существительное», «Прилагательное + Существительное», «Существительное1 + Существительное2», «Причастие + Существительное», «Причастие + Предлог + Существительное», «Вопросительно-относительное местоимение + Глагол», «Предлог + Вопросительно-относительное местоимение + Глагол»,

“Местоименное наречие, играющее роль вопросительного слова + Глагол”, “Глагол + Обозначение числового значения параметра (обозначение числа + обозначение единицы измерения)”.

2. Формализовать структуру данных, используемых в качестве промежуточной формы отображения семантической структуры входного ЕЯ-текста, для последующего построения СП входного текста.
3. На основе решения задач 1 и 2 разработать предметно-независимый метод преобразования входного ЕЯ-текста (вопроса, команды, сообщения) из подъязыка русского языка в его СП.

В данной главе аппарат СК-языков применен к построению широко применимой формальной модели ЛБД. Эта модель описывает логическую структуру ЛБД ЕЯ-интерфейсов интеллектуальных баз данных и других прикладных компьютерных систем. В построенной модели выражения СК-языков используются, во-первых, в качестве семантических единиц, соответствующих лексическим единицам, и, во-вторых, для сборки СП текстов из элементов ЛБД.

Новый метод преобразования ЕЯ-текстов в их семантические представления (СП) предусматривает использование предложенного автором матричного семантико-синтаксического представления входного текста (это понятие было введено в работах (Fomichov 1998, 2002; Фомичев, Волчков 1999) как промежуточного представления при переходе от ЕЯ-текста к СП текста. При этом не используется традиционное синтаксическое представление текста.

Работоспособность предложенного метода доказана разработкой автором сложного структурированного алгоритма семантико-синтаксического анализа текстов из подъязыков естественного (русского) языка и успешным созданием на его основе семейства экспериментальных русскоязычных интерфейсов баз данных и баз знаний, реализованных в программных средах Турбо-Паскаль, версия 7.0, Си, Си++, Delphi 4.0, 5.0, PHP.

6.2. Формализация дополнительных требований к языку построения семантических представлений входных текстов лингвистического процессора

При построении семантических представлений (СП) ЕЯ-текстов в разных предметных областях возникает потребность в использовании небольшого инвариантного набора информационных единиц, в частности, предназначенных для формирования СП вопросов, команд и описаний множеств.

Предположения 1 – 7, сформулированные в процессе исследования выразительных возможностей стандартных К-языков (СК-языков) в главе 4, отражают существо важной части таких потребностей.

В этой и следующей главах мы будем рассматривать концептуальные базисы., для которых выполняются Предположения 1 – 7. Абстрагируясь от математических деталей, можно сказать, что такие концептуальные базисы будут названы размеченными концептуальными базисами. Цель вводимых ниже определений заключается в том, чтобы формально задать понятие размеченного концептуального базиса.

Определение. Пусть *B* – произвольный концептуальный базис, *St*(*B)* – множество сортов базиса *B, P(B)* – выделенный сорт “смысл сообщения”, *X*(*B)* – первичный информационный универсум базиса *B.* Тогда упорядоченный набор *Qmk* вида

(сит,Vsit, Ситуация, Вопрос, лог, ист, ложь, Ист-знач) (6.2.1)

называется разметкой вопросов для концептуального базиса B ⇔ когда

сит, лог ∈ St(B) \ { P(B) } , *X(B)* включает несовпадающие элементы *Ситуация, Вопрос, ист, ложь, Ист-знач,* и выполняются Предположения 1, 4, 6.

Определение. Пусть B – произвольный к.б. Тогда упорядоченный набор *Setmk* вида

(нат, Nt, множ, Колич, Кач-состав, Предм-состав, произв, все, Элем) (6.2.2)

называется теоретико-множественной разметкой базиса B ⇔ *нат ∈ St(B) \ { P(B) },*

Nt - подмножество первичного информационного универсума *X*(*B)*; *множ, Колич, Кач-состав, Предм-состав, произв, все, Элем* – различные элементы множества *X*(*B),* и для компонентов этого набора выполняются Предположения 2, 3, 5.

Определение. Пусть *B* – произвольный к. б., *Qmk* – разметка вопросов вида (6.2.1) для *B,* тогда упорядоченный набор *Cmk* вида

(интс, мом, #сейчас#, #Оператор#, #Исполнитель#, Команда) (6.2.3)

будет называться разметкой команд для базиса *B,* согласованной с разметкой вопросов *Qmk* ⇔ когда *интс, мом, #сейчас#, #Оператор#, #Исполнитель#, Команда* – различные элементы множества X(B); *интс, мом ∈ St(B) \ {P, сит, лог}* и выполняется Предположение 7.

Совокупность формальных понятий, рассмотренных выше в этом подразделе, позволяет сделать заключительный шаг и объединить эти понятия в определении класса размеченных концептуальных базисов.

Определение. *Размеченным концептуальным базисом (р.к.б.)* называется произвольный упорядоченный набор *Cb* вида

(B, Qmk, Setmk, Cmk) , (6.2.4)

где *B* – произвольный концептуальный базис, *Qmk* – разметка вопросов вида (4.2.1) для *B, Setmk* – теоретико-множественная разметка для *B, Cmk* – разметка команд вида (4.2.3) для *B,* согласованная с разметкой вопросов *Qmk,* и выполняются следующие условия: (а) все компоненты наборов *Qmk, Setmk, Cmk* , кроме компонента *Nt* набора *Setmk,* является несовпадающими (различными) элементами первичного информационного универсума X(B);

(б) если *Stadd = { сит, лог, интс, мом, нат },* то *Stadd* –подмножество множества *St(B) \ { P(B) },* причем любые два различные элементы подмножества *Stadd* являются несравнимыми как для отношения общности *Gen,* так и для отношения совместимости *Tol*; (в) если s – произвольный элемент подмножества *Stadd,* то *s* и *P* несравнимы как для отношения *Gen,* так и для и для отношения *Tol*.

Определение. Будем говорить, что размеченный концептуальный базис *Cb* является размеченным базисом стандартного вида ⇔ когда *Cb* – упорядоченный

набор вида (6.2.4), *Qmk* – набор вида (6.2.1), *Setmk* – набор вида (6.2.2) и *Cmk* – набор вида (6.2.3).

В дальнейшем будем рассматривать размеченные концептуальные базисы только стандартного вида.

Класс языков {*Ls(B)* | *B* – первый компонент произвольного р.к.б. *Cb*} будем использовать в качестве семантических языков при рассмотрении соответствий вида "ЕЯ-текст → Семантическое представление текста".

Данный класс языков удобен для построения семантических представлений высказываний, вопросов и команд, причем тексты каждого из указанных видов могут включать составные описания множеств. Многочисленные примеры использования выражений языков этого класса в качестве СП высказываний, вопросов и команд можно найти в главах 3 и 4.

6.3. Текстообразующие системы

Представим формально сведения об элементах, из которых состоят ЕЯ-тексты.

6.3.1. Морфологические базисы

Морфологией называется та часть языкознания, которая изучает закономерности изменения слов и словосочетаний (по числам, падежам, временам и т.д.). Лингвистическая база данных (ЛБД) должна включать *морфологическую базу данных* (МБД), содержание которой зависит от рассматриваемого языка. В отличие от английского языка русский язык является сильно флективным, т.е. слова в нем могут изменяться многими способами. Поэтому, если для английского языка МБД является лостаточно простой, то для русского языка (РЯ) это не так. Формализации морфологии РЯ посвящено много публикаций. Однако для разработки структурированного алгоритма семантико-синтаксического анализа текстов РЯ потребовалось предложить новый, более общий взгляд на морфологию русского языка по сравнению с имеющимися публикациями. Цель заключалась в том, чтобы

указать место морфологического анализа как части семантико-синтаксического анализа ЕЯ-текстов, избегая излишне детального рассмотрения проблем морфологии РЯ. Для достижения этой цели вводятся понятия морфологического детерминанта, морфологического пространства, морфологического базиса и морфологического базиса русскогоязычного типа (Р-типа).

Определение. *Морфологическим детерминантом* (*М-детерминантом*) будем называть произвольную упорядоченную тройку вида

$$(m, n, maxv)\ , \qquad (6.3.1)$$

где m,n - положительные целые числа; *maxv* - отображение из множества $\{1, 2, \ldots m\}$ в множество неотрицательных целых чисел N^+.

Пусть *Det* - М-детерминант вида (6.3.1), тогда m будем интерпретировать как количество всевозможных различных признаков (называемых морфологическими) слов из рассматриваемого языка; n - как максимальное количество различных наборов морфологических признаков, которые могут быть связаны с одним словом. Если $1 \le i \le m$, то *maxv(i)* интерпретируется как максимальное значение признака с номером i (см. рис. 6.1).

Например, со словом "книги" может быть связано три набора значений морфологических признаков (если "книги" - словоформа в единственном числе, то эта словоформа находится в родительном падеже; если "книги" - словоформа во множественном числе, то она может быть как в именительном, так и в винительном падежах). Поэтому $n \ge 3$.

Набор 1	Набор 2	…	Набор n

.

Рис.6.1. Структура массива морфологических признаков, связанных с одной словоформой.

Условимся считать, что морфологические признаки с порядковыми номерами 1 и 2 - это признаки "часть речи" и "подкласс части речи". Поэтому каждое целое k,

такое, что $1 \le k \le maxv(1)$, будем интерпретировать как обозначение какой-то части речи, и каждое целое r, такое, что $1 \le r \le maxv(2)$, будем интерпретировать как обозначение какого-то подкласса некоторой части речи.

Рис. 6.2. иллюстрирует структуру одного набора морфологических признаков с учетом этого соглашения.

Код части речи P_1	Код подкласса ч. речи P_2	Код признака P_3		Код признака P_k	Код признака P_m
1	2	3	…	k	

Рис. 6.2. Структура одного набора значений морфологических признаков.

Будем предполагать, что каждое слово из рассматриваемого языка относится только к одной части речи и к одному подклассу части речи. С одной стороны, это предположение выполняется для чрезвычайно широкого подмножества русского языка и, например, немецкого языка. С другой стороны, такое предположение позволит избежать усложнения (без ущерба для приложения) предлагаемой формальной модели ЛБД.

Определение. Пусть *Det* - М-детерминант вида (6.3.1.). Тогда *морфологическим пространством*, задаваемым детерминантом *Det*, называется множество *Spmorph*, состоящее из всех упорядоченных наборов вида

$$(x_1,\ldots,x_m, x_{m+1},\ldots, x_{2m}, x_{2m+1},\ldots, x_{nm}) \ , \qquad (6.3.2)$$

где: (а) для каждого k=1,…,n-1 $x_{km+1}=x_1$, $x_{km+2}=x_2$; (б) для каждого k=1,…,n и каждого q, такого, что $(k-1)m+1 \le q \le km$, выполняется неравенство $0 \le x_q \le maxv(q-(k-1)m)$.

Условия (а), (б) из данного определения интерпретируются следующим образом. В элементе морфологического пространства вида (6.3.2) x_1 - код части речи, и этот

код расположен во всех позициях, удаленных на расстояние *m, 2m,…, (n-1)m* от позиции 1; x_2 - код подкласса части речи, этот код расположен во всех позициях, удаленных на расстояние *m, 2m,…, (n-1)m* от позиции 2.

Отображение *maxv* для каждого числового кода названия признака q, где $1 \le q \le m$, задает диапазон его значений [1, *maxv(q)*]. Таким образом, для каждой позиции q, где $1 \le q \le m$, $0 \le x_q \le maxv(q)$ для набора вида (6.3.2). Если $1 \le q \le m$, x_q – компонент набора вида (6.3.2) , и $x_q = 0$, то это означает, что словоформа, которой соответствует данный элемент морфологического пространства, не обладает признаком с номером q. Например, у существительных нет признака “время”.

Каждый компонент x_s элемента морфологического пространства вида (6.3.2.), где $s=q+m,q+2m,\ldots,q+(n-1)m$, интерпретируется как какое-то возможное значение морфологического признака, что и в случае элемента x_q. Поэтому неравенство $0 \le x_s \le maxv(s-(k-1)m)$ задает диапазон допустимых значений элемента x_s, где целое число k в пределах от 1 до n однозначно определяется условием $(k-1)m+1 \le s \le km$.

Вводимое ниже определение морфологического базиса дает новую математическую интерпретацию понятия "морфологическая база данных". Если временно абстрагироваться от математических деталей, то под морфологическим базисом мы будем понимать произвольный упорядоченный набор *Morphbs* вида

(Det, A, W, Lecs, lcs, fmorph, propname, valname), (6.3.3)

где Det - морфологический детерминант, а остальные компоненты интерпретируются следующим образом. A – это произвольный алфавит (конечное множество символов); из элементов A образуются словоформы естественного языка. Пусть A^+ - множество всех непустых цепочек в алфавите A. Тогда W - это конечное подмножество A^+, элементы которого рассматриваются как слова и фиксированные словосочетания (например, "в течение"), используемые для построения ЕЯ-текстов. Элементы множества W будем называть словоформами. Lecs - это конечное подмножество множества W, элементы которого называются лексемами и интерпретируются как базовые формы слов и фиксированных словосочетаний (существительное в единственном числе и именительном падеже,

прилагательное в единственном числе, именительном падеже, мужском роде и т.д.).

Компонент *lcs* – это отображение вида $W \rightarrow Lecs$, которое каждой словоформе ставит в соответствие некоторую лексему; *fmorph* – это отображение, которое словоформе *wd* из *W* ставит в соответствие некоторый элемент морфологического пространства *Spmorph(Det).* Компонент *propname* (сокращение от *property-name*) является отображением, которое числовому коду морфологического признака словоформы ставит в соответствие цепочку - его имя. Например, может выполняться соотношение *propname(1)* = *часть-речи.* Точнее, это отображение $propname: \{1,2,\ldots,m\} \rightarrow A+ \setminus W$, где \ - знак теоретико-множественной разности.

Компонент *valname* – это отображение, которое числовому коду морфологического признака k и числовому коду значения данного признака p ставит в соответствие буквенное обозначение данного признака *valname(k, p).* В частности, может выполняться соотношение v*alname(1,1)=глаг* .

Определение. Пусть *A,B* – произвольные непустые множества и $f:A \rightarrow B$ - отображение из A в B. Тогда область значений *Range(f)* – это множество всех таких *y*, что существует такой *x* из *A* , для которого *f(x)= y.*

Определение. Морфологическим базисом называется произвольный упорядоченный набор *Morphbs* вида (6.3.3), где *Det* – М-детерминант вида (6.3.1), *A*-произвольный алфавит, *W* - конечное подмножество множества *A+* (множества всех непустых цепочек в *A*), *Lecs* – конечное подмножество множества *W*, $lcs: W \rightarrow Lecs$ - отображение из *W* на *Lecs* , $fmorph: W \rightarrow Spmorph(Det)$ – отображение из W в морфологическое пространство, порождаемое детерминантом *Det*, *propname* - отображение из $\{1,2,\ldots,m\}$ *в* $A+ \setminus W$, *valname* – частичное отображение из декартового произведения N+ x N+ в множество $A+ \setminus (W \cup Range(propname))$, определенное для пары (i,j) из $N+ \times N+ \Leftrightarrow 1 \leq i \leq m,\ 1 \leq j \leq maxv(i).$

6.3.2. Морфологические базисы Р-типа (русскоязычного типа)

Определение. Морфологический базис вида (6.3.3) называется морфологическим базисом Р-типа (русскоязычного типа) ⇔ выполняются следующие условия:

А – алфавит русского языка, дополненный знаком ‘ -‘ ;

Propname (1) = часть-речи

Propname (2) =подкласс-части-речи

Propname (3) = падеж

Propname (4) = число

Propname (5) = род

Propname (6) = залог

Propname (7) = время

Propname (8) = наклонение

Propname (9) = вид

Propname (10) = лицо

Propname (11) = возвратность

Valname (1,1) = глагол

Valname (1,2) = сущ

Valname (1,3) = прилаг

Valname (1,4) = предлог

Valname (1,5) = местоим

Valname (1,6) = прич

Valname (1,7) = наречие

Valname (1,8) = колич-числит

Valname (1,9) = порядк-числит

Valname (1,10) = союз

Valname (2,1) = сущ-нарицат

Valname (2,2) = сущ-собств

Valname (2,3) = личн-местоим

Valname (2,4) = вопр-относ-местоим

Valname (2,5) = местоим-наречие

Valname (2,6) = глаг-в-изъявит-накл

Valname (2,7) = глаг-в-повелит-накл

Valname (2,8) = глаг-в-неопред-форме

Valname (2,9) = действит-причастие

Valname (2,10) = страдат-причастие

Valname (3,1) = именительн

Valname (3,2) = родительн

Valname (3,3) = дательн

Valname (4,4) = винительн

Valname (5,5) = творительн

Valname (6,6) = предложный

Valname (4,1) = ед. числ.

Valname (4,2) = множ. числ.

Valname (5,1) = жен. род.

Valname (5,2) = муж. род.

Valname (5,3) = сред. род.

Valname (6,1) = действ

Valname (6,2) = страд

Valname (7,1) = прошед-время

Valname (7,2) = наст-время

Valname (7,3) = буд-время

Valname (8,1) = изъявит

Valname (8,2) = повелит

Valname (8,3) = сослаг

Valname (9,1) = несоверш-вид

Valname (9,2) = соверш-вид

Valname (10,1) = 1-ое-лицо

Valname (10,2) = 2-ое-лицо

Valname (10,3) = 3-е-лицо.

Valname (11,1) = действ

Valname (11,2) = страд

Valname (11,1) = вз

Valname (11,2) = нвз .

Здесь *прич* - обозначение части речи “причастие”, *действ* и *страд* – признаки действительного и страдательного залогов глагола, *вз* и *нвз* – признаки возвратных и невозвратных глаголов и причастий.

Определение. Пусть *Morphbs* –морфологический базис вида (6.3.3). Тогда *Parts(Morphbs) = { valname (1,1), … , valname(1,maxv(1)) },*

Subparts(Morphbs) = { valname (2,1), … , valname(2, maxv(2)) }.

Таким образом, *Parts(Morphbs)* – множество названий частей речи, *Subparts(Morphbs)* - множество названий подклассов частей речи для морфологического базиса *Morphbs.* Следовательно, если *Morphbs* – морфологический базис Р-типа, то *Parts(Morphbs)* $\supseteq$ *{ глаг, сущ, прилаг, предлог, местоим, прич, наречие, колич- числит, порядк- числит, союз}, Subparts(Morphbs)* $\supseteq$ *{сущ-нарицат, сущ-собств, вопр-относ-местоим}.*

Пусть *Morphbs* - морфологический базис вида (6.3.3), $z \in Spmorph(Det)$ - произвольный элемент морфологического пространства, и $1 \le i \le mn$, тогда *z [i]* – *i*-й компонент набора z (очевидно, *z* имеет $m \cdot n$ компонентов).

Определение. Пусть *Morphbs* - морфологический базис вида (6.3.3). Тогда отображение *prt* из *W* в *Parts(Morphbs*) и отображение *subprt* из *W* в *Subparts(Morphbs*) задаются следующим образом: для произвольной словоформы $d \in W$ *prt(d) = valname (1, morph(d) [1]), subprt(d) = valname (2, morph(d)[2])* .

Таким образом, *prt(d)* и *subprt(d)* – это соответственно названия части речи и подкласса части речи, к которым относится словоформа d.

Пример. Морфологический базис *Morphbs* Р-типа может быть определён так, что *W э контейнеров, откуда; prt(контейнеров) = сущ, subprt(контейнеров) = сущ-нарицат, prt(откуда) =наречие, subprt(откуда) =местоим-наречие*.

6.3.3. Понятие текстообразующей системы

В текстах могут встречаться не только слова, но и выражения, которые являются числовыми значениями различных признаков, например, 30°, 108%, 90 км/ч, 120 км. Назовём такие выражения *конструктами* и будем считать их единицами текстов. Это означает, что при построении формальной модели лингвистической базы данных, мы будем, например, рассматривать выражение 120_км как символ.

Разрабатывая компьютерные программы, конечно, нужно учитывать, что между “120” и ”км” есть пробел, и “120 км“ - это сочетание из двух элементарных выражений. Но построение всякой формальной модели включает идеализацию сущностей какой-то предметной области, поэтому мы рассматриваем конструкты как символы, т.е. как неделимые выражения.

Кроме слов и конструктов, в текстах встречаются разделители: точка, тире, знак вопроса и т.д., а также выражения в кавычках или апострофах, являющиеся названиями различных объектов.

Определение. Пусть *Cb* – размеченный концептуальный базис вида (6.2.4). Тогда текстообразующей системой (т.о.с.), согласованной с базисом *Cb*, называется произвольный упорядоченный набор *Tform* вида

$$(Morphbs, Constr, infconstr, Markers) \quad , \qquad (6.3.4)$$

где *Morphbs* - морфологический базис Р-типа вида (6.3.3), *Constr* – счетное множество символов, не пересекающееся с множеством словоформ *W* , *infconstr* – отображение из множества *Constr* в первичный информационный универсум *X(B),* где *B* – концептуальный базис, являющийся первым компонентом р.к.б. *Cb, Markers* – конечное множество символов, не пересекающееся с множествами *W* и *Constr,* и выполняются следующие условия: (а) для каждого *d* из множества *Constr* элемент *tp(infconstr(d))* является сортом из множества *St(B);* (б) множества *W, Constr, Markers* не включают апострофы и кавычки.

Элементы множеств *W, Constr* и *Markers* называются соответственно словоформами (или словами), конструктами и разделителями (или маркерами) системы *Tform.*

Очевидно, если задан *Morphbs* - морфологический базис Р-типа вида (6.3.3) , то заданы, в частности, алфавит А и множество словоформ W.

Определение. Пусть *Tform* – текстообразующая система вида (6.3.4). Тогда: *Names (Tform) = Names1 ∪ Names2*, где *Names1*={*‘x’*| *x* – цепочка в алфавите *A*}, *Names2*={*“y”*| *y* – цепочка в алфавите *A*}; *Textunits(Tform) = W ∪ Constr ∪ Names(Tform) ∪ Markers*; *Texts(Tform)* – множество всех конечных последовательностей вида $d_1,\ldots, d_n$, где $n \geq 1$, для $k=1,\ldots, n$ $d_k \in$ *Textunits(Tform).*

Определение. Пусть *Cb* – размеченный концептуальный базис, *Tform* – текстообразующая система вида (6.3.4), согласованная с базисом *Cb.* Тогда отображение *tclass* из *Textunits(Tform)* в *Parts(Morphbs) ∪{констр, имя}* и отображение *subclass* из *Textunits(Tform)* в *Subparts(Morphbs) ∪{nil}*, где *nil* –пустой элемент, задаются следующими соотношениями: (1) если *u∈ W(Tform),* то *tclass(u) = prt(u);* если *u ∈ Constr,* то *tclass(u) = констр;* если *u∈ Names(Tform),* то *tclass(u) = имя; ;* если *u∈ Markers,* то *tclass(u) =маркер*; (2) если *u∈ W(Tform),* то *subclass(u) = subprt(u);* если *u ∈ Constr,* то *subclass(u)= tp(infconstr(u)),* где *infconstr* и *tp* – отображения, являющиеся соответственно компонентами текстообразующей системы и первичного информационного универсума *X(B(Cb))*; если *u∈ Names(Tform) ∪ Markers,* то *subclass(u)= nil.*

6.4. Понятие лексико-семантического словаря

Рассмотрим модель словаря, ставящего в соответствие единицам текстов (“контейнеров”, “поступили” и далее др.) единицы семантического (или, другими словами, информационного) уровня; такие единицы в лингвистике называют *семами*. Лексико-семантический словарь (л.с.с.) является одним из основных компонентов ЛБД. Часть информационных единиц, соответствующих словоформам, мы будем считать символами; они являются элементами первичного информационного универсума *X(B(Cb)),* где *Cb* – размеченный концептуальный базис (р.к.б.), построенный для выбранной области, *B* – концептуальный базис (к.б.), являющийся первым компонентом *Cb*. Примеры таких единиц:

опубликование, *поступление1*, *станция1*, *станция2* и т.д. Другая часть информационных единиц имеет определенную структуру. Например, с прилагательным *"алюминиевый"* из *W* можно связать выражение *Материал (z, алюминий).*

Определение. Пусть *S* – сортовая система (с.с.) вида (2.5.1.). Тогда *семантической размерностью* системы S называется наибольшее такое число $k>1$, что найдутся сорта $u_1,\ldots,u_k \in St$, такие, что для любых $i,j=1,\ldots,k$ при $i \neq j$ u_i и u_j сравнимы для *отношения совместимости Tol* (т. е. $(u_i, u_j) \in Tol$). Это число *k* обозначается через *dim(S)* .

Таким образом, *dim(S)* - это наибольшее количество различных *"семантических осей"*, используемых для описания одной сущности в рассматриваемой области.

Пример. Рассмотрим понятия *"фирма"* и *"институт"*. Можно выделить три *семантических контекста* использования слов, соответствующих эти понятиям. Во-первых, фирма или институт могут разрабатывать прибор, технологию и т.д., поэтому в предложениях с этими словами может быть реализована *семантическая координата "интеллектуальная система"*. Во-вторых, мы можем сказать: "Эта фирма расположена возле м. Таганская," - тогда в этой фразе реализуется *семантическая координата "пространственный объект"*. Наконец, фирмы, институты имеют руководителя. Например, мы можем сказать: "Директор этой фирмы - А. Н. Семенов." В данной фразе реализована *семантическая координата "организация"*.

Мы будем предполагать в рассматриваемых примерах, что семантическая размерность используемых сортовых систем равна четырем или трем.

С содержательной точки зрения, под *лексико-семантическим словарем* мы будем понимать некоторое конечное множество *Lsdic*, состоящее из упорядоченных наборов вида

$$(i, lec, pt, sem, st_1,\ldots,st_k, comment) \quad , \qquad (6.4.1),$$

где $i \geq 1$ - порядковый номер набора (нужен для организации циклов), а остальные компоненты интерпретируются следующим образом. Компонент *lec* является элементом множества лексем *Lecs* рассматриваемого морфологического базиса; *pt* -

обозначение части речи лексемы *lec*; компонент *sem* является цепочкой, обозначающей одно из возможных значений лексемы *lec*.

Компонент *sem* для глаголов, причастий, деепричастий является информационной единицей, связанной с соответствующим отглагольным существительным. Например, глагол *"поступить"* имеет два значения: (1) поступление абитуриента в учебное заведение; (2) поступление физического объекта на какой-то пространственный объект (например, товара на склад). Поэтому, в частности, началом одного из наборов возможного лексико-семантическим словаря будет последовательность элементов i_1, *поступить*, *глаг*, *поступление1*, а началом другого набора - последовательность i_2, *поступить*, *глаг*, *поступление2*.

Число k является семантической размерностью рассматриваемой сортовой системы, т.е. $k = dim(S(B(Cb))$, где Cb – рассматриваемый размеченный концептуальный базис; $st_1,\ldots, st_k$ - различные *семантические координаты* сущности, характеризуемой понятием *sem*. Например, если *sem=фирма*, то st_1=*интс*, st_2=*простр.об*, st_3=*орг*, k= 3. Если же сущность, характеризуемая понятием *sem*, имеет различные семантические координаты $st_1,\ldots,st_p$, где $p<k$, то $st_{p+1},\ldots,st_k$ - это специальный *пустой элемент nil*. Компонент *comment* является пояснением на естественном языке смысла понятия *sem* либо пустым элементом *nil*.

Определение. Пусть *Cb* - размеченный концептуальный базис вида (6.2.4), *Morphbs* - морфологический базис вида (6.3.3), *Qmk* – разметка вопросов вида (6.2.1), первичный информационный универсум *X(B(Cb))* и множество переменных *V(B(Cb))* не включают символ *nil* (пустая сема). Тогда *лексико-семантическим словарем (л.с.с.)*, согласованным с р.к.б. *Cb* и морфологическим базисом *Morphbs*, называется произвольное конечное множество *Lsdic*, состоящее из упорядоченных наборов вида (6.4.1.), где $i \geq 1$, $lec \in Lecs$, $pt=prt(lec)$, $sem \in Lp(Cb) \cup \{nil\}$, $k=dim(S(B(Cb)))$; для каждого $p=1,\ldots, k$ $st_p \in St(B(Cb)) \cup \{nil\}$, $comment \in A^+ \cup \{nil\}$ и выполняются следующие условия:

(а) никакие два набора из *Lsdic* не могут иметь один и тот же первый компонент *i*; (бг) если два набора из *Lsdic* имеют разные значения компонента *sem*, то эти два набора имеют разные значения компонента *comment.*

Пример. *Lsdic* может быть определён так, что *Lsdic* включает следующие наборы:

(112, контейнер, сущ, контейнер1, дин. физ. объект, nil, nil, *“ёмкость”),*

(208, поступить, глаг, поступление1, ↑cит, nil, nil, *«поступить в вуз»),*

(209, поступить, глаг, поступление2, ↑cит, nil, nil, *«поступил груз»),*

*(311, алюминевый, прилаг, Материал(*z1, *алюминий),физ.об,* nil, nil, nil*),*

*(358, зеленый, прилаг, Цвет(*z1, *зелен), физ.об,* nil, nil, nil*),*

*(411, пассажирский, прилаг, Назначение(*z1, *перемещение1 * (Объект1, опред множ * (Кач-состав, человек))), дин.физ.об,* nil, nil, nil*),*

*(450, Италия, сущ, нек страна * (Назв, ‘Италия’), простр.об,* nil, nil, *«страна”),*

*(512, обувь, сущ, нек множ * (Кач-состав, * изделие1 * (Вид, обувн)), дин.физ.об,* nil, nil, *«термин, обозначающий различные множества обувных изделий»).*

6.5. Словари глагольно-предложных семантико-синтаксических фреймов

Ключевую роль в формировании предложений играют глаголы, причастия, деепричастия и отглагольные существительные, выражая разнообразные отношения между объектами рассматриваемой предметной области.

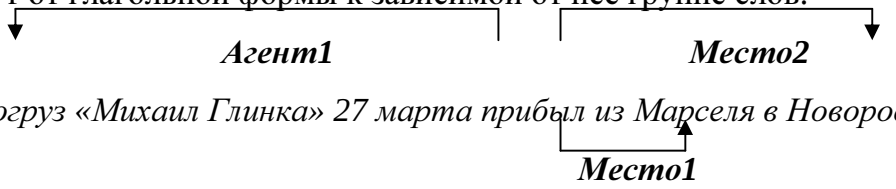
Тематической ролью называется смысловое отношение между значением глагольной формы (личной формы глагола, неопределенной формы глагола, причастия, деепричастия, отглагольного существительного) и значением зависящей от нее в предложении группы слов. Тематические роли называются также концептуальными падежами, семантическими падежами, глубинными падежами и семантическими ролями.

Впервые понятие глубинного падежа было предложено американским лингвистом Ч. Филлмором в 1968 году. Это понятие очень быстро стало широко популярным в компьютерной лингвистике, поскольку лежит в основе базовых

процедур установления смысловых отношений между значением глагольной формы и значением зависящей от нее группы слов.

Пример 1. Пусть Т1 = «Сухогруз «Михаил Глинка» 27 марта прибыл из Марселя в Новороссийск». Глагол прибыл обозначает некоторое событие вида прибытие, с которым можно связать метку e1 (event1). В Т1 упоминаются (называются) следующие объекты: некоторый корабль x1; некоторый город x2 с названием “Марсель”; некоторый город x3 с названием “Новороссийск”.

В событии *e1* объект *x1* играет роль «Агент-действия» (*Агент1*), *x2* играет роль «Исходное место для движения» (*Место1*), *x3* играет роль «Целевое место» (*Место2*). Тогда говорят, что в Т1 реализуются тематические роли *Агент1*, *Место1*, *Место2*, а также тематическая роль Время. Эта информация представляется размеченным текстом Т1, где стрелки с обозначением тематических ролей ведут от глагольной формы к зависимой от нее группе слов:



Пример 2. Пусть Т2 = «Сухогруз «Михаил Глинка» прибыл из Марселя.».

В Т2 явно реализуются только тематические роли *Агент1*, *Место1*, а тематические роли *Время* и *Место2* только подразумеваются в силу семантики глагола *прибывать*. Таким образом, фраза с одним и тем же глаголом в одном и том же значении могут явно выражать разные подмножества тематических ролей.

На формальном уровне мы будем интерпретировать тематические роли, как названия бинарных отношений, где первым атрибутом является ситуация, а вторым – реальный или абстрактный объект, играющий определенную роль в этой ситуации. При этом, если элемент $r \in R_2(B)$, где B – концептуальный базис, интерпретируется как тематическая роль, то его тип $tp(r)$= - это цепочка вида $\{(s,u)\}$, где s – конкретизация выделенного сорта *сит* (ситуация), а u – сорт из $St(B)$.

Словари глагольно-предложных фреймов содержат такие шаблоны (фреймы), которые позволяют представлять необходимые условия для реализации конкретной тематической роли в сочетании *Глагольная форма + Предлог + Зависимая группа*

слов, где *предлог* может быть пустым *(nil)*, а *зависимая группа слов* является либо существительным с зависимыми словами или без них, либо конструктом, то есть числовым значением параметра. Например, такими сочетаниями являются выражения “прибыть в порт”, “выехал из города”, “подготовить 4 статьи”, “купила итальянские туфли”, “ вернулся до 16:30”.

Определение. Пусть *Cb* – размеченный концептуальный базис вида (4.2.4), *Tform* – текстообразующая система вида (6.3.4), согласованная с *Cb*, *Morphbs* – морфологический базис вида (6.3.3) , *Lsdic* – лексико-семантический словарь, согласованный с *Cb* и *Tform*. Тогда словарем глагольно-предложных фреймов (с.г.п.ф.), согласованным с *Cb, Tform* и *Lsdic*, называется произвольное конечное множество *Vfr*, состоящее из упорядоченных наборов вида:

(k, semsit, form, refl, vc, sprep, grcase, str, trole, expl) (6.5.1)

где $k \geq 1$, $semsit \in X(B)$, $form \in \{неопр, личн, nil\}$, $refl \in \{возвр, невз, nil\}$, $vc \in \{действ, страд, nil\}$, $sprep \in W \cup \{nil\}$, где *nil* – пустой элемент, *W* - множество словоформ системы *Tform;* если $sprep \in W$, то $prt(sprep)=предлог$; $0 \leq grcase \leq 6$, $str \in St(B)$, *trole* – бинарный реляционный символ из первичного информационного универсума *X(B)*, причем $tp(trole)=\{(s,u)\}$, где $s, u \in St(B)$, причем *s* – конкретизация сорта “ситуация” *сит* (т.е. *сит* -> *s*); $expl \in A^{+} \cup \{nil\}$.

Компоненты произвольного набора вида (6.5.1) из *Vfr* интерпретируются следующим образом: *k* – порядковый номер набора; *semsit* – семантическая единица, обозначающая вид ситуации (прибытие, отлет, получение и др.); *form* - признак формы глагола, *неопр* – указатель неопределенной формы глагола; *личн* – указатель личной формы глагола, т.е. глагола в изъявительном или сослагательном наклонении; *refl* - признак возвратности глагола или причастия, *возвр* - указатель возвратной формы, *невз* - указатель невозвратной формы; *действ*, *страд* – признаки действительного и страдательного залога.

Компоненты *semsit, form, refl, vc* задают требования к глагольной форме, а компоненты *sprep, grcase, str* – требования к слову или группе слов, зависящей от глагольной формы и выражающей вместе с ней тематическую роль *trole*. Цепочка *sprep* – предлог, в том числе и составной (например, «в течение»), или nil; grcase –

код грамматического падежа от 1 до 6, либо 0 – как указатель отсутствия такой информации; *str* – семантическое ограничение на значение зависимой группы слов или слова; trole – та тематическая роль, необходимое условие реализации которой представляет данный набор (фрейм); *expl* – пример на ЕЯ, поясняющий тематическую роль, либо пустой пример nil.

Пример. Построим словарь. *Vfr1*, позволяющий устанавливать смысловые отношения в предложениях с глаголами *подготовить* и *поступить*. С каждым из этих глаголов будем связывать два значения. С глаголом *подготовить* – значения *подготовка1* (подготовка отчета, статьи и т.д.) и *подготовка2* (подготовка мастеров спорта и т.д.); с глаголом *поступить* – значения *поступление1* (поступление абитуриента в вуз) и *поступление2* (поступление контейнера на склад и т.д.).

Словарь *Vfr1* должен быть полезен, в частности, для семантического анализа текстов Т1 = “Профессор Семенов подготовил в июне отчет для НИИ «Заря»”; Т2 = “Профессором Семеновым в июне был подготовлен отчет для НИИ «Заря»”; Т3 = “Профессор Семенов подготовил в течение 1995-2003 годов трех кандидатов химических наук”; Т4 = “Контейнер поступил на склад в среду”;
Т5 = “В 1999 году Игорь поступил в МИЭМ”.

Можно построить такой р.к.б. *Cb*, что его первым компонентом является к.б. *B*, и выполняются следующие предположения: *St(B)* включает подмножество
{орг, интс, мом, инф.об, дин.физ.об, сит, квалиф, соб, простр.об, строка};
X(B) включает подмножество *{сейчас, иссл.инст, вуз, профессор, канд.хим.н, уч.завед, чел, ‘Семенов’, подготовка1, подготовка2, МИЭМ, контейнер1, склад1, отчет1, Назв, Фам, Месяц, июнь, среда, 3, 1995, 2002, 2003, Квал, Агент1, Объект1, Объект2, Продукт1, Время, Место1, Место2, Адресат1, Уч.зав}*;
сит → соб (т.е. событие является частным случаем ситуации);
tp (подготовка1) = tp (подготовка2) = tp (поступление1) =
tp (поступление2) = ↑соб ; *tp (иссл.инст) = tp (вуз) = ↑орг*простр.об*интс* ;
*tp (чел) = ↑интс*дин.физ.об* ; *tp (склад1) = ↑простр.об* ;
tp (контейнер1) = ↑дин.физ.об ; *tp (профессор) = tp (канд.хим.н) = квалиф* ;

*tp (МИЭМ) = орг*простр.об*интс tp (Агент1) = {(соб, интс)}* ;
tp (Адресат1) = {(соб, орг)}, tp (Время) = {(соб, мом)} ;
tp (Место1) = tp (Место2) = {(соб, простр.об)} ; *tp (Фам) = {(интс, строка)};*
tp ('Семенов') = строка ; *tp (Объект1) = {(соб, дин.физ.об)}* .

Тогда пусть *Vfr1* – множество, состоящее из следующих упорядоченных наборов:

k semsit form refl vc sprep grcase str trole expl

(1, подготовка1, личн, нвз, действ, nil, *1, интс, Агент1, 'И.П.Сомов подготовил (учебное пособие)')* ,
(2, подготовка1, личн, нвз, страд, nil,*5, интс, Агент1, 'Профессором Семеновым была подготовлена (книга)')* ,
(3, подготовка1, личн, нвз, действ, nil, *4, инф.об, Продукт1, '(И.П.Сомов) подготовил книгу'),*
(4, подготовка1, личн, нвз, страд, nil, *1, инф.об, 1, инф.об, 'Статья была подготовлена (за три недели)')* ,
(5, подготовка2, личн, нвз, действ, nil, *4, квалиф, Объект2, '(школа) подготовила 5 мастеров спорта')* ,
(6, подготовка1, nil, nil, *в, 0, мом, Время, 'подготовил в 2001-м году')* ,
(7, подготовка2, nil, nil, *в, 0, мом, Время, 'подготовит в 2003-м году')* ,
(8, поступление1, личн, действ, в, 4, орг, Уч.зав, '(Игорь) поступил в МИЭМ') ,
(9, поступление1, личн, действ, nil, *1, интс, Агент1, 'Игорь поступил (в МГУ)'),*
(10, поступление2, личн, нвз, действ, nil, *1, дин.физ.об, Объект1, 'контейнер поступил (на склад)')* ,
(11, поступление2, личн, нвз, действ, на, 4, простр.об, Место2, '(контейнер) поступил на склад') ,

(12, поступление2, личн, нвз, действ, в, 4, простр.об, Место2, ‘(контейнер) поступил вчера в магазин’) .

6.6. Формализация необходимых условий реализации данного смыслового отношения в сочетаниях вида “Глагол + Зависимая группа слов”

Пусть Tform – текстообразующая система, согласованная с размеченным концептуальным базисом (р.к.б.) Cb вида (6.2.4); $T \in$ Texts(Tform), длина (T) = nt, $1 \leq$ posn1 $\leq$ nt, posn1 – позиция существительного из T, $1 \leq$ posvb $\leq$ nt, posvb - позиция глагола из T; sem1, sem2 $\in$ X(B), где X(B) – первичный информационный универсум концептуального базиса B, являющегося первым компонентом р.к.б. Cb; prep - предлог из W или пустой предлог nil, $1 \leq$ grcase ≤ 6, rel – бинарный реляционный символ из X(B), интерпретируемый как название тематической роли. Тогда условимся, что запись

(T, posn1, sem1, prep, grcase, posvb, sem2, rel) $\in$ *Смысл-связь1*

интерпретируется следующим образом: если с элементом t_{posn1} , т.е. с существительным в позиции posn1, можно связать семантическую единицу sem1 и грамматический падеж с кодом grcase, к этому существительному в тексте T относится предлог prep (в частности, пустой предлог *nil*), с элементом t_{posvb} , т.е. с глаголом в позиции posvb, можно связать семантическую единицу sem2, то между элементами t_{posvb} и t_{posn1} может существовать смысловое отношение, являющееся тематической ролью с именем rel .

Пример 1. Предположим, что редакция некоторого научного журнала использует в своей работе интеллектуальную информационно-поисковую систему (ИПС), и этой системе задан вопрос B1 = “Когда поступила статья профессора Сомова?”. Если проставить после каждой текстообразующей единицы из B1 ее порядковый номер, то получится следующее (более удобное для анализа) представление вопроса B1: “Когда (1) поступила (2) статья (3) профессора (4) Сомова (5) ? 6)”.

Допустим, что лигвистическая база данных (ЛБД) редакционной ИПС включает компоненты, формальными моделями которых являются некоторый лексико-семантический словарь Lsdic и некоторый словарь глагольно-предложных фреймов Vfr, согласованные с размеченным концептуальным базисом (р.к.б.) Cb, причем первой составляющей Cb является концептуальный базис B. Пусть словарь Lsdic включает наборы

(k, поступить, глаг, поступление1, соб, nil, nil, 'поступление в вуз'),

(k+1, поступить, глаг, поступление2, соб, nil, nil, 'поступление контейнера на склад'),

(m, статья, сущ, статья1, инф.об, дин.физ.об, nil, 'статья, отправленная вчера в газету'),

(m+1, статья, сущ, статья2, инф.об, nil, nil, 'статья как часть юридического документа').

Тогда, очевидно, в вопросе B1 реализуется значение глагола "поступить", которому в словаре Lsdic соответствует семантическая единица *поступление2*, и реализуется значение существительного "статья", которому соответствует семантическая единица *статья1*.

Обозначим через *Объект1* тематическую роль, реализующуюся (в контексте вопроса B1) в сочетании "поступила статья", и предположим, что первичный информационный универсум X(B) включает бинарный реляционный символ *Объект1*. Заметим, что словоформа "статья" в вопросе B1 не связана с каким-либо предлогом, т.е. этой словоформе соответствует пустой предлог *nil*.

Пусть posn1 = 3 (позиция в вопросе B1 слова "статья") , posvb = 2 (позиция в вопросе B1 слова " поступила "). Слово "статья" в тексте B1 находится в именительном падеже, кодом которого является число 1. Тогда, с учетом сделанных предположений, выполняется соотношение

$$(B1, 3, статья1, nil, 1, 2, поступление2, Объект1) \in Смысл\text{-}связь1 .$$

Используя формальные средства, определим более точно смысл соотношения $(T, posn1, sem1, prep, grcase, posvb, sem2, rel) \in Смысл\text{-}связь1$.

Определение. Если Tform – текстообразующая система вида (6.3.4), Morphbs – морфологический базис Р-типа вида (6.3.3), то пусть

Nouns (Tform) = {d ∈ W │ prt(d) = сущ } , Prepositions (Tform) = {d ∈ W │ prt(d) = предлог } , Verbs (Tform) = {d ∈ W │ prt(d) = глаг}.

Таким образом, *Nouns(Tform), Prepositions(Tform)* и *Verbs(Tform)* – соответственно множества существительных, предлогов и глаголов, задаваемых текстообразующей системой Tform.

Определение. Пусть *Cb* - размеченный концептуальный базис вида (6.2.4), где первой составляющей *Cb* является концептуальный базис *B; Tform* – текстообразующая система, согласованная с р.к.б. *Cb*; *X(B)* – первичный информационный универсум базиса *B* ; m – семантическая размерность сортовой системы *S(B)* ; $R_2(B)$ – подмножество *X(B),* состоящее из всех бинарных редяционных символов; *Lsdic* – лексико-семантический словарь (л.с.с.), согласованный с р.к.б. *Cb*, *Vfr* - словарь глагольно-предложных фреймов, согласованный с р.к.б. *Cb* и л.с.с. *Lsdic*; *N* – множество положительных целых чисел.

Тогда подмножество *Смысл-связь1* декартова произведения

Texts(Tform) x N x X(B) x Prepositions (Tform) x {1,…, 6} x N x X(B) x $R_2(B)$

задается следующим условием:

упорядоченный набор *(T, posn1, sem1, prep, grcase, posvb, sem2, rel)* принадлежит множеству *Смысл-связь1* ⇔ когда *T ∈ Texts(Tform),* длина *(T) = nt, 1 ≤ posn1 ≤ nt,* t_{posn1} *∈ Nouns (Tform), 1 ≤ posvb ≤ nt,* t_{posvb} *∈ Verbs (Tform), prep ∈ Prepositions (Tform) U { nil }, 1 ≤ grcase ≤ 6, sem1, sem2 ∈ X(B), rel ∈* $R_2(B)$, т.е. *rel* – бинарный реляционный символ из первичного информационного универсума *X(B),*

существуют такие наборы (фреймы) *Fr1, Fr2* из *Lsdic* соответственно видов *(i1, lec1, сущ , sem1,* s_1*,…,* s_m*, comment1) ,*

(i2, lec2, глаг , sem2, st_1*,…,* st_m*, comment2)* и существует такой фрейм *Fr3* из *Vfr* вида *(k1, semsit, form, refl, vc, relat, sprep, grc, str, expl) ,*

что выполняется каждое из следующих условий:

Условие 1*: lcs (* t_{posn1}*) = lec1 , lcs (*t_{posvb} *) = lec2, semsit = sem2,*

sprep = prep , grc = grcase, relat = rel.

Условие 2: *grcase ∈ Падежи(morph(t_{posn1}) ,* где – множество числовых кодов всех грамматических падежей, которые могут соответствовать существительному в позиции *posn1*.

Условие 3: Пусть для произвольного сорта *s ∈ St Gener(s) = {u ∈ St(B) │ (u, s) ∈ Gen(B)}* , т.е. *Gener(s)* - множество всех сортов, являющихся обобщениями сорта s , включая сам сорт *s*. Тогда *str* входит в объединение множеств *Gener(s_i)* по всем сортам s_1,…, s_m , являющихся компонентами фрейма *Fr1* из *Lsdic* и отличных от пустого сорта *nil* .

Условие 4: Глагол имеет значение признака возвратности *refl,* значение формы *form* и значение залога *vc* .

Легко видеть, что смысл условия 3 заключается в том, что найдется такое *p, 1 ≤ p ≤ m,* что сорт s_p является конкретизацией сорта *str* – компонента глагольно-предложного фрейма *Fr3*.

Пример 2. Вернемся к размеченному представлению вопроса В1 “Когда (1) поступила (2) статья (3) профессора (4) Сомова (5) ? 6)”.

Допустим, что словарь глагольно-предложных фреймов Vfr включает набор вида *(n, поступление2, личн, нвз, действ, дин.физ.об, nil, 4, Объект1, ‘поступил контейнер’)* , где n – порядковый номер набора, личн – признак личной формы глагола (в отличие от неопределенной формы), действ - признак действительного залога глагола, дин.физ.об – сорт “динамический физический объект”, 1 – числовой код именительного падежа.

Пусть справедливы следующие соотношения (см. Пример 1): *posn1 = 3, posvb = 2, sem1 = статья1, sem2 = Lsdic[k+1].sem = поступление2, str = Lsdic[m].st_2 = дин.физ.об, prep = nil , = grcasy = 1, = rel = Объект1, lcs (статья) = статья , lcs (поступила) = поступить*. Тогда имеет место соотношение

(B1, posn1, sem1, prep, grcase, posvb, sem2, rel) ∈ Смысл-связь1 ,

т.е. *(B1, 3, статья1, nil, 1, 2, поступление2, Объект1) ∈ Смысл-связь1* ,

где *3 = posn1* – позиция слова “статья”, *2 = posvb* - позиция слова “поступила”.

Смысловые связи в предложении могут существовать не только между глаголами и существительными, но и между глаголами и конструктами, т.е. числовыми значениями различных параметров. Как и в случае существительных, на вид смысловой связи между глаголом и конструктом влият предлог, который может находиться перед конструктом. Например, в сочетаниях “нагрейти воду до 12 градусов” и “нагрейти воду на 12 градусов” реализуются разные смысловые отношения.

Пусть Tform – текстообразующая система, согласованная с размеченным концептуальным базисом.*Cb* вида (4.2.4); $T \in$ Texts(Tform), длина (T) = nt, $1 \leq$ posc1 $\leq$ nt, posc1 – позиция конструкта из T (т.е. $t_{posc1} \in$ Constr(Tform)), $1 \leq$ posvb $\leq$ nt, posvb - позиция глагола из T; semvb - семантическая единица из первичного информационного универсума X(B), соответствущая отглагольному существительному, образованному от глагола в позиции posvb; prep - предлог из W или пустой предлог nil, rel – бинарный реляционный символ из X(B), интерпретируемый как название тематической роли. Тогда будем считать, что запись

(T, posnc1, prep, posvb, semvb, rel)) $\in$ *Смысл-связь2*

интерпретируется следующим образом: если к конструкту в позиции posc1 относится предлог prep, то между элементами t_{posvb} и t_{posc1} может существовать смысловое отношение, являющееся тематической ролью с именем rel .

Пример 3. Рассмотрим предписания T1 = “ Нагрейти воду до 18 градусов” и T2 = “Нагрейти воду на 18 градусов”. Заменим эти предписания их размеченными представлениями “Нагрейти (1) воду (2) до (3) 18 градусов (4)” и “Нагрейти (1) воду (2) на (3) 18 градусов (4)”.

Пусть posc1 = 4, posvb = 1 , semvb = нагревание . Тогда можно определить размеченный концептуальный базис Cb, словари Lsdic и Vfr, согласованные с р.к.б. Cb, и отношение *Смысл-связь2* так, что будут выполняться соотношения

(T1, posnc1, до , posvb, semvb, Предельное-значение) $\in$ *Смысл-связь2* ,

(T2, posnc1, на , posvb, semvb, Приращение-значения) $\in$ *Смысл-связь2* .

6.7. Словари предложных семантическо-синтаксических фреймов

6.7.1. Формальное определение словаря предложных фреймов

Рассмотрим следующую проблему: каким образом в сочетании “Существительное1 + Предлог + Существительное2” или “Существительное1 + Существительное2” установить, какое именно смысловое отношение реализуется в этом сочетании. Рассмотрим идею решения на примерах.

Пример 1. Пусть С1 = ”Поезд из Праги”. Со словом “поезд” связано понятие *поезд1*, а этому понятию соответствует сорт *дин.физ.об* (“динамический физический объект”).. Словоформа “Праги” обозначает город. С понятием *город1* связан сорт *простр.об* (пространственный объект). Существительное “Прага” находится в родительном падеже, тогда можно представить, что лингвистическая база данных (ЛБД) включает семантическо-синтаксический шаблон вида

(k1, ‘из’, дин.физ.об, простр. об., 2, Место3, ‘посылка из Таганрога’) ,

смысл которого заключается в следующем: *k1* – номер шаблона; ‘из’ – предлог; *дин.физ.об*. – сорт “динамический физический объект”, связанный с первым существительным; *простр.об* – сорт, связанный со вторым существительным; 2 – код родительного падежа, причём второе существительное должно находиться в родительном падеже; *Место3* – обозначение смыслового отношения, которое реализуется в сочетании “Существительное1 + ‘из’ + Существительное2” при выполнении заданных условий; *‘посылка из Таганрога’* – пример выражения, в котором реализуется отношение *Место3*. Вместо *Место3* мы могли бы написать *Исходный-пространственный-объект*.

Легко видеть, что сочетание С1 совместимо с этим шаблоном, имеющим номер *k1*.

Пример 2. Пусть С2=”статья в журнале”, тогда ЛБД может включать шаблон вида

(k2, ‘в’, инф.об, инф.об, 6, Место4, ‘глава в книге’) ,

где *k2* – номер шаблона; ‘в’ – предлог; *инф. об*. – сорт “информационный объект”, 6 – код предложного падежа.

Со словом “статья” связаны понятия *статья1*, *статья2*. Понятие *статья1* интерпретируется как понятие, которому соответствует выражение “статья в журнале, газете и т.д.”; *статья2* – это отдельная смысловая часть документа (юридическое понятие). Сорт “информационный объект” связан с каждым из этих понятий. Поэтому выражение С2 совместимо с шаблоном, имеющим номер *k2*. Следовательно, в выражении С2 может реализовываться смысловое отношение *Место4*.

Пример 3. Пусть С3=”статья профессора”, и ЛБД включает шаблон вида

(k3, nil, инф.об, интс, 2, Авторы, ‘поэма Пушкина’) ,

где *nil* – пустой предлог; *интс* – сорт “интеллектуальная система”, 2 – код родительного падежа. С лексемой “профессор” можно связать сорт *интс* и сорт *дин.физ.об* (“динамический физический объект”). Поэтому сочетание С3 совместимо с шаблоном, имеющим номер *k3*.

Определение. Пусть *Cb* – размеченный концептуальный базис вида (6.2.4), *B=B(Cb), Morphbs* – морфологический базис вида (6.3.3); *Tform* - текстообразующая система вида (6.3.4), согласованная с р.к.б. *Cb*; *Lsdic* – лексико-семантический словарь, состоящий из записей вида (6.4.1), согласованный с *Cb* и *Tform*. Тогда словарём предложных семантико-синтаксических фреймов, согласованным с *Cb, Tform и Lsdic* , называется произвольное конечное множество *Frp*, состоящее из упорядоченных наборов вида

(i, prep, sr1, sr2, grc, rel, ex) (6.7.1)

где $i \geq 1$; $prep \in Lecs \cup \{nil\}$, *где nil* – цепочка, обозначающая пустой предлог; если $prep \in Lecs$, то *prt (prep) = предлог*; $sr1, sr2 \in St(B)$; $1 \leq grc \leq 6$; $rel \in R_2(B)$; $R_2(B)$ – множество бинарных реляционных символов, являющееся подмножеством первичного информационного универсума *X(B(Cb))*; $ex \in A+$.

Компоненты набора вида (6.7.1) из множества *Frp* интерпретируются следующим образом. Натуральное число $i \geq 1$ является порядковым номером набора (используется для организации циклов), *prep* – это предлог из множества лексем *Lecs* или пустой предлог *nil*. Элементы *sr1* и *sr2* интерпретируются как сорта, которые можно связать соответственно с

первым существительным и вторым существительным в лингвистически правильном сочетании “Сущ.1 + prep + Сущ.2”; *grc (grammatic case)* – код падежа, в котором должно находиться второе существительное в таком правильном сочетании; *rel* – обозначение смыслового отношения, которое может реализовываться в таком сочетании при выполнении указанных условий; *ex* – пример выражения, в котором реализуется то же самое отношение rel.

Пример. Можно построить такие размеченный концептуальный базис *Cb*, морфологический базис *Morphbs*, текстообразующую систему *Tform*, лексико-семантический словарь *Lsdic*, и словарь предложных семантико-синтаксических фреймов *Frp*, что *Frp* включает семантические шаблоны (фреймы): с номерами *k1, k2, k3,* рассмотренные в примерах 1 - 3, а также следующие шаблоны:

(k4, ‘от’, вещество, болезнь, 2, Против1, ‘таблетки от гриппа’);

(k5, ‘от’, вещество, дин.физ.об, 2, Против2, ‘мазь от комаров’);

(k6, ‘от’, физическое явление, физ.об, 2, Эффект1, ‘тень от дома’).

Потребуем, чтобы выполнялось следующее условие: в словаре *Frp* не найдётся таких наборов, в которых совпадают компоненты *prep ≠ nil, sr1, sr2, grc*, но не совпадают компоненты *rel* или *ex*. В таком случае четвёрка *(prep, sr1, sr2, grc)* однозначно определяет смысловое отношение *rel*.

6.7.2. Формализация необходимых условий существования определенного смыслового отношения в сочетании из двух существительных с учетом предлога

Определение. Пусть *B* – произвольный концептуальный базис (к.б.). Тогда для произвольного сорта $s \in St(B)$ $Gener(s)=\{u \in St(B) \mid (u,s) \in Gen(B)\}$, т.е. *Gener(s)* – множество всех сортов, являющихся обобщением сорта *s*, включая сам сорт s.

Определение. Пусть *Tform* – текстообразующая система (т.о.с.), согласованная с размеченным концептуальным базисом *Cb* вида (6.2.4.), *Morphbs* – морфологический базис Р-типа вида (6.3.3), $Nouns(Tform)=\{ d \in W \mid prt(d)=сущ\}$,

Prepositions(Tform)={ d∈W | prt(d)=предлог}, X(B) – первичный информационный универсум концептуального базиса *B=B(Cb), m* – семантическая размерность сортовой системы *S(B), $R_2(B)$* – подмножество *X(B),* состоящее из всех бинарных реляционных символов, N^+ - множество положительных целых чисел, *T* – текст из *Texts(Tform).* Тогда подмножество *Смысл-связь3* множества

Texts(Tform) ×N ×X(B) ×Prepositions(Tform) ×{1, … 6} ×N^+ ×X(B) ×R_2(B)

задается следующим условием:

(T, posn1, sem1, prep, grcase, posn2, sem2, rel)∈Смысл-связь3 ⇔

T∈Texts(Tform), 1 ≤posn1 < posn2 ≤длина (T), sem1, sem2 ∈X(B),

prep∈Prepostions(Tform) U {nil}, где *nil* – пустой предлог, *1 ≤ grcase ≤ 6, rel ∈ R_2(B),* и существуют такие фреймы *Fr1, Fr2 из Lsdic* соответственно видов

(i1, lec1, сущ, sem1, s_1, …, s_m, comment1) ,

(i2, lec2, сущ, sem2, st_1, …, st_m, comment2),

а также фрейм *Fr* из *Frp* вида *(k1, prep, sr1, sr2, grc, rel, ex),*

что *lcs(t_{posn1})=lec1, lcs(t_{posn2})=lec2, grc ∈ Падежи (fmorph(t_{posn2})),*

sr1 входит в объединение множеств *Gener(s_i)* по всем сортам s_1, …, s_m, отличным от *nil, sr2* входит в объединение множеств *Gener(st_i)* по всем сортам st_1,…, st_m, отличным от *nil*.

Пример. Проиллюстрируем применение определения отношения *Смысл-связь3* к проверке возможности реализации смыслового отношения *Против1* в сочетании “микстура от кашля”, являющегося фрагментом предложения Т1= “В аптеке # 18 продается новая микстура от кашля”.

Пусть множество сортов *St(B)* включает элементы *вещество, жидк.вещество, (вещество, жидк.вещество) ∈ Gen(B), Lsdic* включает элементы *Fr1, Fr2* соотвественно видов

(n1, микстура, микстура1, жидк.вещество, nil, nil, ‘лекарство’)

(n2, кашель, кашель1, болезнь, nil, nil, ‘вид заболевания’).

Пусть *posn1=6, posn2=8.* Тогда с учетом того, что словарь *Frp* включает набор (*k4, ’от’, вещество, болезнь, 2, Против1, ‘таблетки от гриппа’),* справедливо соотношение

(T, posn1, микстура1, от, 2, posn2, кашель1, Против1) ∈ Смысл-связь3.

6.8. Лингвистические базисы

6.8.1. Формализация семантической информации, связанной с вопросительными словами

Определим понятие системы вопросительных словосочетаний. Будем называть ролевыми вопросительными словосочетаниями пары вида *(qw, d),* где *qw* – это предлог, либо пустой предлог nil; $d \in W$ - некоторое слово, являющееся либо вопросительно-относительным местоимением, либо местоименным наречием. Например, такими сочетаниями являются пары *(nil, кто)* , *(nil, кому)* , *(для, кого)* , *(у, кого)* , *(nil, откуда)* .

Наша языковая интуиция позволяет связать с каждой из таких пар некоторое достаточно общее смысловое отношение:

(nil, кто) → Агент; (nil, кому) → Адресат; (для, кого) → Адресат;

(у, кого) → Источник1 (частные случаи: Продавец, Поставщик).

В связи с этим в состав лингвистической базы данных введём ещё один словарь.

Определение. Пусть выполняются предположения из определения словаря предложных семантико-синтаксических фреймов. Тогда системой ролевых вопросительных словосочетаний, согласованной с размеченным концептуальным базисом *Cb*, морфологическим базисом *Morphbs* и лексико-семантико-синтаксическим словарём *Lsdic* , называется произвольное конечное множество, состоящее из упорядоченных наборов вида

$$(i, prep, qw, relq) \quad , \qquad (6.8.1)$$

где $i \geq 1$, $prep \in Lecs \cup \{nil\}$, $qw \in W$, $prt(qw) \in \{местоим, наречие\}$, $relq \in R_2(B(Cb))$; в случае $prep \neq nil$ $prt(prep) = предлог$, $subprt(qw) = вопр\text{-}относ\text{-}местоим$; в случае $prep = nil$ $subprt(qw) \in \{вопр\text{-}относ\text{-}местоим$, $местоим\text{-}наречие\}$.

Пример. Определим *B, Cb, Morphbs, Lsdic, Rqs* так, что *Rqs* включает наборы

(1, nil, кто, Агент) , (2, nil, кому, Адресат) ,

(3, для, кого, Адресат), (4, у, кого, Источник1) , (5, на, чём, Инструмент)

(6, nil, когда, Время) , (7, nil, откуда, Место1) , (8, nil, куда, Место2).

(3, для, кого, Адресат), (4, у, кого, Источник1) , (5, на, чём, Инструмент).

6.8.2. Понятие лингвистического базиса

Лингвистические базисы являются формальными моделями лингвистических баз данных (ЛБД)..

Определение. Упорядоченный набор *Lingb* вида

$$(Cb, Tform, Lsdic, Vfr, Frp, Rqs) \qquad (6.8.2)$$

называется *лингвистическим базисом (л.б.)* ↔ когда *Cb* – размеченный концептуальный базис (р.к.б.) вида (6.2.4), *Tform* – текстообразующая система (т.о.с.) вида (6.3.4), согласованная с р.к.б. *Cb, Lsdic* – лексико-семантический словарь (л.с.с.), согласованный с р.к.б. *Cb* и т.о.с. *Tform*, *Vfr* – словарь глагольно-предложных семантико-синтаксических фреймов, согласованный с р.к.б *Cb*, т.о.с. *Tform*, л.с.с.*Lsdic*; *Rqs* – система вопросительных словосочетаний, согласованная с *Cb*, *Tform*, *Lsdic*.

Формальное понятие лингвистического базиса отражает наиболее существенные черты логической структуры широко применимых ЛБД. Это понятие конструктивно в том смысле, что на его основе можно проектировать ЛБД практически полезных лингвистических процессоров.

Понятие лингвистического базиса обобщает научные результаты автора, опубликованные в работах (Фомичев 1978б, 1979, 1980, 1986а, 1987б, 1988ж, 1990г, 1991б; Fomichov 1992, 2002a; Fomichov, Kochanov 2001).

Глава 7
НОВЫЙ МЕТОД ВЫПОЛНЕНИЯ ПРЕОБРАЗОВАНИЯ “ЕЯ-ТЕКСТ ➔СЕМАНТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ”

В этой главе предложена новая структура данных (названная матричным семантико-синтаксическим представлением текста), используемая в качестве промежуточной формы отображения семантической структуры входного ЕЯ-текста, для последующего построения СП текста. Описывается новый предметно-независимый метод преобразования входного ЕЯ-текста (вопроса, команды, сообщения) из подъязыка русского языка в его СП, являющееся выражением некоторого СК-языка.

Структуры данных, ассоциированные с текстом в рамках заданного лингвистического базиса

В следующем параграфе будет описан новый метод преобразования ЕЯ текста в его СП; метод использует новую форму представления промежуточных результатов анализа текста. - матричное семантико-синтаксическое представление (МССП) текста. Для определения МССП ЕЯ-текста опишем ряд вспомогательных структур данных, которые ассоциированы с входным текстом прикладной интеллектуальной системы в рамках рассматриваемого лингвистического базиса.

7.1.1. Компонентно-морфологическое представление текста

Морфологическое представление. Временно пропуская ряд математических деталей, условимся под морфологическим представлением текста *T* длины *nt* понимать двумерный массив *Rm* с индексами столбцов *base* и *morph*, элементы которого интерпретируются следующим образом. Пусть *nmr* – количество строк массива *Rm*, построенного для текста *T*, и *k* – номер строки массива *Rm*, т.е. $1 \le k \le nmr$. Тогда *Rm[k, base]* – это лексема, соответствующая некоторой словоформе в позиции *p* из текста *T*. При тех же предположениях *Rm[k, morph]* – это последовательность наборов значений морфологических признаков, соответствующих словоформе в позиции *p*.

Определение. Пусть *Tform* – текстообразующая система вида (6.3.4), *Morphbs* – морфологический базис вида (6.3.3), $T \in Texts(Tform)$, *nt* – длина (*T*). Тогда *морфологическим представлением текста T* называется двумерный массив *Rm* с индексами столбцов base и morph, для которого выполняются следующие условия:

1. Каждая строка массива *Rm* содержит информацию о какой-то словоформе из входного текста *T*, т.е., если *nmr* – количество строк массива *Rm*, то для каждого *i* от *1* до *nmr* в тексте *T* найдется такая позиция *p*, $1 \le p \le nt$, что $t_p \in W$, $Rm[i, base] = lcs(t_p)$, $Rm[i, morph] = fmorph(t_p)$.
2. Для каждой словоформы из текста *T* найдется строка в *Rm*, представляющая морфологическую информацию об этой словоформе, т.е. для каждой позиции *p* в тексте *T*, где $1 \le p \le nt$, $t_p \in W$, найдется такое *k*, где $1 \le k \le nmr$, что $lcs(t_p) = Rm[k, base]$, $fmorph(t_p) = Rm[k, morph]$.
3. Любые две строки массива *Rm* различаются либо значением лексемы в столбце *base*, либо набором значений морфологических признаков в столбце morph, т.е. если $1 \le k \le nmr$, $1 \le q \le nmr$, $k \ne q$, то либо $Rm[k, base] \ne Rm[q, base]$, либо $Rm[k, morph] \ne Rm[q, morph]$.

Таким образом, произвольная строка массива Rm указывает лексему (базовую форму) и совокупность наборов значений морфологических признаков, связанных

с какой-то лексической единицей из текста Т. В то же время для каждой лексической единицы из Т найдена соответствующая строка в Rm.

Пример. Пусть Т1= «В(1) каком(2) московском(3) издательстве(4) в(5) 2001-м(6) году(6) вышла(7) работа(8) по(9) искусственному(10) интеллекту(10) «Основы обработки знаний» (11) профессора(12) Сомова(13)?(14)» (после каждого элементарного выражения из текста указан порядковый номер элементарной значащей единицы текста, к которой относится данное выражение). Тогда морфологическое представление Rm текста Т1 будет иметь следующую форму:

base	morph
в	md_1
какой	md_2
московский	md_3
издательство	md_4
выходить	md_5
работа	md_6
по	md_7
искусственный интеллект	md_8
профессор	md_9
Сомов	md_{10}

Рис. 7.1. Структура морфологического представления Rm

Здесь md_1,…, md_{10} – числовые коды наборов морфологических признаков, связанных с соответствующими словами из входного текста Т1. В частности, md_4 кодирует следующие сведения: часть речи – существительное, подкласс части речи - существительное нарицательное, число – единственное, падеж – предложный. Набор неотрицательных целых чисел md_{10} кодирует следующую информацию: часть речи - существительное, подкласс часть речи – существительное собственное, падеж - родительный, число - единственное.

Классифицирующее представление. Пусть *Tform* – текстообразующая система вида (4.3.4), *T∈Texts(Tform), nt* – длина (*T*). Тогда, с содержательной точки зрения, классифицирующим представлением текста T, согласованным с морфологическим представлением Rm текста T, будет называться двумерный массив Rc с количеством строк nt и индексами столбцов unit, tclass, subclass, mcoord, элементы которого интерпретируются следующим образом.

Пусть k – номер произвольной строки массива Rc, т.е. 1≤k≤nt. Тогда Rc[k, unit] является одной из элементарных значащих единиц текста T , т.е если $T=t_1 \ldots t_{nt}$, то найдется такая позиция p, 1≤p≤nt, что Rc[k, unit]= t_p. Если Rc[k, unit] – словоформа, то Rc[k, tclass], Rc[k, subclass], Rc[k, mcoord] являются соответственно обозначениями части речи, подкласса части речи, последовательности наборов значений морфологических признаков.

Если Rc[k, unit] – конструкт (т.е. числовое значение параметра), то Rc[k, tclass] – цепочка *констр*, Rc[k, subclass] – сорт информационной единицы, соответствующей данному конструкту, Rc[k, mcoord]=0.

Пример. Пусть T1= «В каком московском издательстве в 2001-м году вышла работа по искусственному интеллекту «Основы обработки знаний» профессора Сомова?» Тогда классифицирующее представление *Rc* для текста T1,согласованное с *Rm*, может иметь следующий вид:

unit	tclass	subclass	mcoord
в	предлог	nil	1
каком	местоим	вопрос-относит-местоим	2
московском	прилаг	nil	3
издательстве	сущ	сущ-нарицат	4
в	предлог	nil	1
2001-м году	констр	момент	0
вышла	глаг	глаг-в-изъявит-накл	5
работа	сущ		6

по	предлог	сущ-нарицат	7
искусственному	сущ	nil	8
интеллекту		сущ-нарицат	
«Основы	имя		
обработки			
знаний»	сущ		9
профессора	сущ	сущ-нарицат	10
Сомова	маркер	сущ-собств	0
?			

Рис. 7.2. Структура классифицирующего представления Rc

Определение. Пусть Tform – т.о.с. вида (6.3.4), Morphbs – м.б. вида (6.3.3), nt – длина (T), T∈ Texts(Tform), Rm - морфологическое представление T. Тогда классифицирующим представлением текста T, согласованным с Rm, назовём двумерный массив Rc с индексами столбцов unit, tclass, subclass, mcoord и количеством строк nt, для которого выполняются следующие условия:

1. Для k = 1,.., nt Rc[k,unit] = tk.
2. Если 1≤k≤nt, tk ∈ W, то Rc[k, tclass] = prt(tk),

Rc[k, subclass] = subprt(tk), и найдется такое q, 1≤q≤nrm, где nrm количество строк в Rm, что Rc[k, mcoord]=q, fmorph(tk)=les(tk)=Rm[q,base], Rm[q,morph]

3. Если 1≤k≤nt, tk∈ Constr, Rc[k, tclass]=констр, Rc[k, subclass]=tp((tk)), Rc[k,mcoord]=0.
4. Если 1 ≤ k ≤ t, t_k∈ Names (Tform), то Rc[k, tclass]=имя, Rc[k, subclass]=ml, Rc[k, mcoord]=0.
5. Если 1≤k≤nt, tk∈ Markers, то Rc[k, tclass]=Маркер, Rc[k, subclass]=ml, Rc[k,mcoord]=0.

Таким образом, классифицирующее представление текста T задаёт следующие сведения:

1. Для каждой лексической единицы указывает часть речи, подкласс части речи (если он определён) и номер строки из морфологического представления Rm, перечисляющей числовые коды морфологических признаков, соответствующих данной лексической единице..

2. Для каждого конструкта задает класс *констр* и подкласс, являющийся сортом информационной единицы, соответствующей конструкту.

3. Для каждого элемента из множества Names(Tform) указывает класс *имя*, подкласс nil и число 0 в столбце mcoord.

4. Для каждого разделителя (знаки препинания) указывается класс *маркер*, подкласс nil и 0 в столбце mcoord.

Определение. Пусть *Tform* – т.о.с. вида (6.3.4), *T∈ Texts(Tform).* Тогда компонентно-морфологическим представлением текста *T* будем называть упорядоченную пару вида *(Rm, Rc),* где *Rm* – морфологическое представление текста *T*, *Rc* – классифицирующее представление текста *T,* согласованное с *Rm.*

7.1.2. Проекции компонентов лингвистического базиса на входной текст

Пусть *Lingb* – л.б. вида (6.8.2), и *Dic* – какой–либо из следующих компонентов *Lingb:* лексико-семантический словарь (л.с.с.) *Lsdic,* словарь глагольно-предложных фреймов *Vfr,* словарь предложных фреймов *Frp.* Тогда проекцией *Dic* на входной текст *T∈ Texts(Tform)* назовем двумерный массив, строки которого представляют всю информацию из *Dic,* которая относится к лексическим единицам из *T.* Вводимые ниже определения позволяют уточнить эту идею.

Определение. Пусть *Lingb* – л.б. вида (6.8.2), *T∈ Texts (Tform), nt* – длина *T, (Rm, Rc)* – компонентно-морфологическое представление *T.* Тогда проекцией л.с.с. *Lsdic* на входной текст *T* назовем двумерный массив *Arls* с индексами столбцов *ord, sem,* st_1,…, st_m, *comment,* где m - семантическая размерность сортовой системы S(B(Cb(Lingb))), удовлетворяющий следующим условиям:

1. Элементами столбца ord являются порядковые номера элементарных значащих единиц текста T, т.е. номера строк классифицирующего представления Rc.
2. Элементы столбцов *sem*, st_1,…, st_m, *comment* интерпретируются так же, как и одноименные компоненты л.с.с. *Lsdic* , т.е. *sem* – простая или составная семантическая единица, st_1,…, st_m – несравнимые для отношения совместимости *Tol* элементы сортового множества *St(B(Cb(Lingb)))*, *comment* – естественно-языковое описание смысла единицы *sem*.
3. Для каждой словоформы wd из W, входящей в строку $q \geq 1$ и столбец *unit* классифицирующего представления *Rc*, найдутся такая строка с номером $k \geq 1$ в массиве *Arls* и такой набор вида *(i, lec, pt, sem,* st_1,…, st_k, *comment)* из лексико-семантического словаря *Lsdic* , где $i \geq 1$, что компоненты этого набора *sem*, st_1,…, st_m, *comment* совпадают с элементами одноименных столбцов из строки *k*; *Rc[q, tclass] = pt*, и, если *Rc[q, mcoord] = m*, то *Rm[m, base] = lec*.
4. Строки массива *Arls*, в столбце *sem* которых расположены семантические единицы, представляющие разные значения одной и той же лексемы, следуют подряд и не могут перемежаться строками, в столбце *sem* которых отражены возможные значения каких-то других лексем.
5. Пусть для произвольной строки с номером *n* массива *Arls* элемент *q = Arls[n, ord]* является номером какой-то строки классифицирующего представления *Rc*, т.е. $1 \leq q \leq nt$. Тогда элементы столбцов *sem*, st_1,…, st_m, *comment* строки с номером *n* массива *Arls* совпадают с одноименными компонентами какого-то набора вида (*j, lec, pt, sem,* st_1,…, st_m, *comment*) из л.с.с. *Lsdic* , где $lcs(t_q) = lec$ - лексема, $prt(t_q) = pt$ – обозначение части речи.
6. В массиве *Arls* нет повторяющихся строк.

Легко видеть, что из сформулированных выше условий вытекает, что количество строк narls массива Arls равно сумме количеств значений лексем, соответствующих словоформам из входного текста T.

Пример. Пусть T1= «В(1) каком(2) московском(3) издательстве(4) в(5) 2001-м(6) году(6) вышла(7) работа(8) по(9) искусственному(10) интеллекту(10) «Основы(11) обработки(11) знаний(11)» профессора(12) Сомова(13)?(14)». Тогда массив Arls для T1 может иметь следующий вид:

N	**ord**	**sem**	**st1**	**st2**	**st3**	**st4**	**comment**
1	3	Город(z, Москва)	Простр .об	nil	nil	nil	nil
2	7	Издательство	орг	интс	простр.об	nil	nil
3	7	Выход1	сит	nil	nil	nil	‘Игорь вышел из комнаты’
4	7	Выход2	сит	nil	nil	nil	‘Книга вышла в 1988 году’
5	8	Работа1	соб	nil	nil	nil	‘Эта работа заняла 4 часа’
6	7	Работа2	инф.об	дин.физ. об	nil	nil	‘Работа профессора Новикова была отправлена экспресс-почтой’
7	10	Иск. интеллект	Науч. обл	nil	nil	nil	‘Научное направление «искусств. интеллект»’
8	12	Нек.чел *(Квалиф, ‘профессор’)	интс	дин.физ. об.	nil	nil	nil
9	13	Нек.чел * (Фам., ‘Сомов’)	интс	дин.физ. об	nil	nil	nil

Рис. 7.3. Структура массива Arls

Смысл рассмотрения двумерного массива Arvfr, называемого проекцией словаря глагольно-предложных фреймов Vfr на текст T, заключается в следующем: для каждой глагольной формы из текста T в этом массиве размещаются все шаблоны (фреймы) из словаря *Vfr,* позволяющие находить возможные смысловые отношения между значением данной глагольной формы и значением зависящей от нее в предложении из текста T группы слов.

Определение. Пусть Lingb – л.б. вида (6.8.2), T∈Texts(Tform), nt = длина(T), (Rm, Rc) – компонентно-морфологическое представление T, Arls – проекция лексико-семантического словаря Lsdic на текст T. Тогда назовем двухмерный массив Arvfr с индексами столбцов *nb, semsit, form, refl, vc, sprep, grc, str, trole, example* проекцией словаря глагольно-предложных фреймов Vfr на текст T, если выполняются следующие условия:

1. Элементами столбца *nb* являются порядковые номера элементарных значащих единиц текста T, являющихся глагольными формами (глаголами или причастиями), т.е. номера строк классифицирующего представления Rc.
2. Элементы столбцов *semsit, form, refl, vc, sprep, grc, str, trole, example* интерпретируются так же, как и одноименные компоненты словаря глагольно-предложных фреймов Vfr .
3. Пусть $q \geq 1$ – номер произвольной строки классифицирующего представления *Rc*, для которой Rc[q, tclass] ∈ {*глаг, прич*} (т.е. *q* – позиция произвольной глагольной формы из входного текста T); k - номер такой произвольной строки массиве *Arls*, что *Arls[k, ord] =q,* и *semunit* – элемент *Arls[k, sem]*. Тогда для каждого набора *d* из словаря *Vfr* вида

   (i, semsit, refl, form, vc, sprep, grc, str, trole, example) ,

   где $i \geq 1$ и *semsit = semunit* , в массиве *Arvfr* найдется такая строка с номером *m*, что *Arvfr [m, semsit] = semunit,* и элементы строки m, расположенные в столбцах *refl, form, vc, sprep, grc, str, trole, example,* совпадают с одноименными компонентами набора *d.*

Другими словами, для каждой глагольной формы t_q из текста T и любого значения *semunit* формы t_q массив *Arvfr* должен включать каждый глагольно-предложный фрейм из словаря *Vfr*, связанный со значением *semunit*.

4. Пусть *m* – номер произвольной строки массива *Arvfr* . Тогда найдутся такая строка массива Rc с номером q и такая строка массива Arls с номером k, что выполняются соотношения Rc[q, tclass] ∈ {*глаг*, *прич*}, *Arls[k, ord] =q*, *Arls[k, sem].= Arvfr [m, semsit]* (т.е. каждая строка массива *Arvfr* содержит глагольно-предложный фрейм из словаря Vfr, связанный с некоторым значением какой-либо глагольной формы из текста T).
5. Строки массива *Arvfr*, в столбце *semsit* которых расположены семантические единицы, представляющие разные значения одной и той же лексемы, следуют подряд и не могут перемежаться строками, в столбце *semsit* которых отражены возможные значения каких-то других лексем.
6. В массиве *Arvfr* нет повторяющихся строк

Пример. Вопрос T1 = “В каком московском издательстве в 2001-м году вышла работа по искусственному интеллекту “Основы обработки знаний” профессора Сомова?” включает глагольную форму “вышла”. Глагол “выходить” имеет, в частности, два значения, которым мы поставим в соответствие семантические единицы *выход1* и *выход 2*. Эти значения реализуются соответственно в предложениях “Теплоход вышел из порта в 8:30” и “Учебник вышел в 2003-м году”.

Будем считать, что со значением *выход1*связаны тематические роли Агент1 (Агент действия), Место1 (обозначение отношения между событием, заключающимся в перемещении физического объекта в пространстве, и исходным пространственным объектом), Место2 (обозначение отношения между событием, заключающимся в перемещении физического объекта в пространстве, и целевым пространственным объектом), Время, Длительность, Целевой-предмет (данное отношение реализуется, например, в предложении “Игорь вышел из дома за хлебом”).

Со значением *выход2* будем ассоциировать тематические роли Инф-объект (Информационный объект), Время, Организация (обозначение отношения между событием, заключающимся в опубликовании информационного объекта, и организацией, опубликовавшей этот объект).

Тогда массив *Arvfr*, построенный по тексту T1 с учетом массива Arls, рассмотренного в предыдущем примере, может иметь следующий вид:

nb	*semsit*	*fm*	*refl*	*vc*	*trole*	*sprep*	*grc*	*str*	*example*
6	Выход1	из	нвз	действ	Агент1	*nil*	1	дин. физ.об.	Он вышел (из дома)
6	выход1	из	нвз	действ	Место1	из	2	простр. об	(он) вышел из дома
6	выход1	из	нвз	действ	Длит	на	0	зн.длит.	Вышел на 2 часа
6	выход1	из	нвз	действ	Время	в	0	момент	(Теплоход) вышел (из порта) в 8:30
6	выход1	из	нвз	действ	Целе-вой пред-мет	за	5	Дин. Физ.об	Вышел (из дома) за хлебом
6	выход2	из	нвз	действ	Инф-объект	*nil*	1	инф.об.	(В издательстве “Белый город») вышел альбом
6	выход2	из	нвз	действ	Время	в	0	момент.	(Книга) вышла в 2002-м году
6	выход2	из	нвз	действ	Органи-зация	в	6	орг	(книга) вышла в издательстве

Рис. 7.4. Пример массива *Arvfr*

Связь с массивом *Arls* существляется через поле *semsit.* Шаблон из *Arvfr* связан со строкой с номером *k* массива *Arls*, если они относятся к одной и той же лексической единице из текста, и значение поля sem для массива *Arls* совпадает со значением поля *semsit* из массива *Arvfr*.

Аналогично строится массив *Arpfr* – проекция словаря предложных фреймов *Frp* на входной текст. Этот массив предназначен для отображения всех сведений из словаря Frp, относящихся к предлогам из текста T и к пустому предлогу *nil*.

Определение. Пусть Lingb – л.б. вида (6.8.2), T∈ Texts(Tform), nt = длина(T), (Rm, Rc) – компонентно-морфологическое представление T, Arls – проекция лексико-семантического словаря Lsdic на текст T. Тогда назовем двухмерный массив Arfrp с индексами столбцов *prep, sr1, sr2, grc, rel, ex* проекцией словаря предложных фреймов Frp на текст T, если выполняются следующие условия:

1. Пусть $q \geq 1$ – номер произвольной строки классифицирующего представления *Rc*, для которой Rc[q, tclass] = предлог (т.е. *q* – позиция произвольного предлога из входного текста T), и *pr = Rc[q, unit]*. Тогда в массиве Arfrp нйдется такая строка с номером $k \geq 1$, что Arfrp[k, prep] = pr, и в словаре Frp найдется набор вида (6.7.1), в котором prep] = pr, и компоненты *sr1, sr2, grc, rel, ex* совпадают с элементами одноименных столбцов массива Arfrp, расположенными в строке *k*.
2. Пусть *k*. – номер произвольной строки массива *Arfrp.* Тогда найдутся такие строка массива Rc с номером $q\ (1 \leq q \leq nt)$ и набор *d* вида (6.7.1) в словаре Frp, что *Arfrp[k, prep] = Rc[q, unit], Rc[q, tclass] = предлог*, и компоненты *sr1, sr2, grc, rel, ex* набора *d* совпадают с элементами одноименных столбцов массива Arfrp, расположенными в строке *k*.
3. Пусть *d* – произвольный набор из словаря предложных фреймов Frp, для которого компонентом *prep* является пустой предлог *nil*, и *h* - набор, получающийся из *d* удалением первого компонента (порядкового номера) набора. Тогда компоненты *sr1, sr2, grc, rel, ex* набора *h* совпадают с элементами одноименных столбцов некоторой строки массива Arfrp (т.е. Arfrp включает все шаблоны, или фреймы, позволяющие находить

возможные смысловые отношения в сочетаниях вида “Существительное1 + пустой предлог + Существительное2”).

4. Все строки массива Arfrp различны.

Пример. В тексте T1 = “В каком московском издательстве в 2001-м году вышла работа по искусственному интеллекту “Основы обработки знаний” профессора Сомова?” встречается предлог “по”. Этот предлог может использоваться, в частности, в сочетаниях “прогулка по городу” и “книга по физике”, причем во втором случае реализуется то же смысловое отношение, что и в тексте T2. Поэтому массив Arfrp для T1 может иметь вид, представленный на рисунке 7.5.

prep	sr1	sr2	grc	rel	ex
по	соб	простр.об	3	Место3	Прогулка по парку
по	инф.об	обл.деят	6	Область1	Книга по живописи

Рис. 7.5. Пример фрагмента массива Arfrp

7.2. Матричное семантико-синтаксическое представление ЕЯ – текста

Рассмотрим новую структуру данных, предлагаемую в данной работе в качестве промежуточной формы представления результатов семантико-синтаксического анализа ЕЯ-текстов и называемую матричным семантико-синтаксическим представлением (МССП) входного текста T.

МССП ЕЯ-текста T – это строково-числовая матрица Matr с индексами столбцов *locunit, nval, prep, posdir, reldir, mark, qt, nattr, contr*, позволяющая по информации о возможных видах коротких сочетаний слов найти смысловые отношения между элементами предложения T, а также указать одно из нескольких возможных значений каждой лексической единицы.

Таким образом, МССП текста – это матрица Matr следующего вида:

locunit	*nval*	*prep*	*posdir*	*reldir*	*mark*	*qt*	*nattr*	*contr*

Рис. 7.6. Структура матричного семантико-синтаксического представления текста

Количество строк матрицы Matr равно *nt* - количеству количество строк в классифицирующем представлении *Rc* , т.е. количеству выделенных элементарных значащих единиц текста.

В столбце ***locunit*** (*location of unit,* место единицы) указывается наименьший номер строки массива *Arls*, которая соответствует лексической единице с порядковым номером *k,* где *k* – это номер строки массива *Rc* и номер строки матрицы *Matr*. Массив *Arls* представляет все наборы из лексико-семантического словаря (л.с.с.) *Lsdic*, которые содержат информацию о лексических единицах из входного текста. Массив *Arls* выше был назван проекцией л.с.с. *Lsdic* на входной текст *T*.

Можно сказать, что значение поля *locunit* для *k*-той единицы текста является координатой входа по этой единице в массив *Arls*.

Столбец ***nval*** *(number of values,* количество значений*)* в начальный момент построения *Matr* указывает количество всех строк из *Arls,* соответствующих *k*-й лексической единице, где *k* – номер строки *Rc* и *Matr*. После завершения построения *Matr* в столбце *nval* на пересечении с каждой строкой, соответствующей лексической единице, должно находиться значение 1, поскольку для каждой лексической единицы было найдено одно из нескольких возможных значений.

Столбец ***prep*** *(preposition,* предлог*)* для каждой строки с номер *k* указывает предлог (возможно, пустой предлог *nil*), относящийся к *k*-й лексической единице.

Рассмотрим назначение группы столбцов ***posdir*** *(posdir*$_1$*, posdir*$_2$*, …, posdir*$_n$), где *n* – константа в пределах от 1 до 10, зависящая от программной реализации. Пусть $1 \le d \le n$. Тогда будем использовать обозначение *Matr[k, posdir, d]* для элемента, расположенного на пересечении строки k и столбца из группы ***posdir*** с порядковым номером *d* в данной группе.

Если $1 \le k \le nt$, $1 \le d \le n$, то *Matr[k, posdir, d] = m*, где *m* – это либо 0, либо порядковый номер *d*-й лексической единицы из входного текста *T*, управляющей единицей с порядковым номером *k*. Для глаголов в главном предложении в этих столбцах стоит 0, т.к. для них нет управляющей единицы. Условимся считать, что существительное управляет стоящими перед ним прилагательными, а также относящимся к нему числом или количественным числительным (например, в сочетании “5 научных статей”). В группе столбцов *reldir* содержатся обозначения смысловых отношений, отраженных в группе столбцов *posdir*.

Рассмотрим соотношения, являющиеся исходными для заполнения столбцов *posdir* и *reldir* **.** Эти соотношения базируются на определениях отношений Смысл.связь1, Смысл.связь2, Смысл.связь3 из подраздела 6.7.2. Начнем с соотношений для сочетаний вида “Глагольная форма + Предлог + Существительное”, где Предлог может быть пустым предлогом *nil*.

Пусть $1 \le k \le nt$, $1 \le d \le n$, *Rc[k, tclass] = сущ*, $1 \le posvb \le nt$, $k \ne posvb$, *Rc[posvb, tclass]*$\in$*{глаг, прич}, Matr[k, posdir, d]=posvb, Matr[k,locunit]=loc1, Matr[k, nval]=1, Matr[posvb, locunit]=loc2, Matr[posvb, nval]=1, Arls[loc1, sem]=sem1, Arls[loc2, sem]=sem2, Matr[k, reldir, d]=relcat, prep1=Matr[k,prep]* .

Тогда найдется такой код грамматического падежа *grcase*, где $1 \le grcase \le 6$, что *(T, k,sem1, prep1, grcase, posvb,sem2, relat)*$\in$ *Смысл.связь1*. При этом *grcase*$\in$ *Падежи(Rm[j, morph]), где j=Rc[k,mcoord], Падежи(Rm[j, morph])* – множество числовых кодов всех грамматических падежей, указанных в каком-либо наборе морфологических признаков из *Rm[j, morph]*.

Если $1 \le k \le nt$, *Rc[k,tclass] = констр* (т. е. *k*-я единица текста является конструктом), и *Matr[k,reldir, d]=rel*, где *rel* – некоторый бинарный реляционный символ из *X(B)*, то выполняются следующие соотношения:

(1) найдется такое целое posvb, где $1 \le posvb \le nt$, что $Matr[k,posdir, d]=posvb$

(2) $Rc[posvb, tclass] \in \{глаг, прич\}$; (3) Если $Matr[k,prep]=prep1$, $Matr[posvb, locunit]=locvb$, $Arls[locvb,sem]=sem1$,

то $(T, k, prep1, posvb, sem1, rel) \in Смысл.связь2$.

Таким образом, будем считать, что управляющей единицей текста для каждого конструкта (т.е. представления значения числового параметра) является либо глагол, либо причастие).

Если $1 \le posn1 < posn2 \le n2$, $Rc\ [posn1, fclass] = Rc\ [posn2, fclass] = сущ$, $Matr[posn2, posdir] = posn1$, $Matr[posn2, prep] = prep1$, $Matr[posn1, reldir] = rel$, то элемент *rel* является именем смыслового соотношения между существительными в позициях posn1 и posn2, где слово в позиции posn1 управляет словом в позиции posn2 с учетом предлога prep1, относящегося ко второму существительному.

В этом случае должно выполняться следующее условие: если

$Matr[posn1, locunit] = loc1$, $Matr[posn1, nval] = 1$,

$Arls[loc1, sem] = sem1$, $Matr[posn2, locunit] = loc2$,

$Matr[posn2, nval] = 1$, $Arls[loc2, sem] = sem2$,

то найдется такое целое число grcase, где $1 \le grcase \le 6$, что $(T, posn1, sem1, prep,. Grcase, Posn2, sem2, rel) \in Смысл.связь3$, причем $grcase \in Cases(fmorph(Rc[posn2, unit))$.

Столбец ***mark*** (метка) предназначен для хранения переменных, обозначающих различные сущности из входного текста (в том числе события, на которые указывают глаголы, причастия, деепричастия, отглагольные существительные).

В столбце ***qt*** *(quantity)* – количество, помещается либо 0, либо число, которое указывается в тексте перед существительным и относится к существительным.

В столбце ***nattr*** *(number of attributes)* – количество атрибутов, указывается либо 0, либо количество прилагательных относящихся к существительному представленному в данной строке *k*, т.е. мы предполагаем, что $R_1[k].unit$ –это существительное.

В столбце ***contr*** (*control*, управление) помещается либо 0, либо число, позволяющее установить связь между главным предложением и причастным оборотом или придаточным предложением.

Пример. Пусть В1 = «Сколько контейнеров, поступивших в пятницу из Новороссийска, были отправлены АО “Радуга”?».

Тогда $k = 2$ – порядковый номер слова «контейнеров»; $p = 4$ – порядковый номер слова «поступивших», и *Matr[k, contr] = p, Matr[p, contr] =k.* Таким образом, если *k,* - позиция существительного, к которому ”прикреплено” причастие, то *Matr[k, contr]* – позиция этого причастия. Наоборот, если *p* - позиция причастия, то *Matr[p, contr]* – позиция существительного, к которому “прикреплено” это причастие.

Пусть П1 = ”Профессор Сомов работает в институте, который он закончил в 1978 году”, k = 5 (позиция словоформы “институте”), m = 9 (позиция словоформы “закончил”). Тогда *Matr[k, contr] = m,* и *Matr[m, contr] =k.*

Если придаточное определительное предложение соединено с главным предложением с помощью вопросительно-относительного местоимения в позиции j , то Matr[j, contr] = m, где m – позиция существительного из главного предложения, к которому прикреплено придаточное предложение.

Возможность использовать столбец contr в двух противоположных смыслах обусловлена тем, что каждая строка, соответствующая лексической единице, одназначно определяет ее часть речи.

Проиллюстрируем форму матричное семантико-синтаксическое представления (МССП) Matr.

Пример. Построим МССП текста Т1= «В(1) каком (2) московском (3) издательстве (4) в (5) 2001-м году (6) вышла (7) работа (8) по(9) искусственному интеллекту (10) ”Основы обработки знаний” (11) профессора (12) Сомова (13) ? (14)». В параграфе 4.9 для текста Т1 были построены массивы Arls, Arvfr, Arfrp. С учетом этого МССП Matr для текста Т1 может иметь следующий вид:

	Loc-unit	nval	prep	Pos-dir	reldir	Ma-rk	qt	nattr	Co-ntr
1	0	0	в	0, 0	nil, nil	nil	0	0	0
2	0	0	в	0, 0	Nil, nil	nil	0	0	0
3	1	1	в	4, 0	Место(z, Москва), nil	nil	0	0	0
4	2	1	в	7, 0	Простр.объект, nil	X1	0	1	0
5	0	0	в	0, 0	nil, nil	nil	0	0	0
6	0	0	в	7, 0	Время, nil	nil	0	0	0
7	3	1	nil	0, 0	nil, nil	L1	0	0	0
8	6	1	nil	7, 0	Объект 3, nil	X2	0	0	0
9	0	0	no	0, 0	nil, nil	nil	0	0	0
10	7	1	no	8, 0	Область 1, nil	X3	0	0	0
11	0	0	0	8, 0	Название, nil	nil	0	0	0
12	8	1	nil	8, 0	Авторы, nil	X4	0	0	0
13	9	1	nil	12, 0	Фамилия(z, ’Сомов’), nil	X4	0	0	0
14	0	0	0	0, 0	nil, nil	nil	0	0	0

Рис. 7.7. Пример матричного семантико-синтаксического представления текста

Построенная матрица отражает финальную конфигурацию МССП Matr. Это значит, что найдены все смысловые соотношения между единицами текста.

7.3. Новый метод преобразования ЕЯ-текстов в их семантические представления

7.3.1. Принципы установления соответствия между матричным семантико-синтаксическим представлением текста и его К-представлением

Как уже отмечалось выше, матричное семантико-синтаксическое представление (МССП) ЕЯ-текста Т строится как промежуточная структура для представления результатов семантико-семантического анализа Т. Следующий шаг должен заключаться в построении по МССП *Matr* некоторого К-представления текста Т, т.е. выражения некоторого стандартного К-языка, интерпретируемого как семантическое представление (СП) текста Т. В связи с этим ниже излагаются наиболее общие принципы преобразования МССП ЕЯ-текста Т в некоторое К-представление текста Т. На основе этих принципов в главе 9 разработан алгоритм преобразования МССП текста в его К-представление.

Рассмотрим структуры данных, позволяющие осуществить преобразование МССП текста в его К-представление.

Массив Sitdescr предназначен для построения семантических описаний ситуаций (в частности, событий), упоминаемых во входном тексте. Количество заполненных строк этого массива равно количеству глаголов и причастий в тексте. Столбец с индексом mrk хранит метки ситуаций (связь с Matr осуществляется через метки из этого столбца). Столбец с индексом expr предназначен для хранения семантических описаний ситуаций.

Рассмотрим пример заполнения массива Sitdescr. Пусть В1 = “На каких предприятиях, для которых поставляет картон АО “Старт”, выпускают мебель для кухни?”. Тогда массив Sitdescr может иметь следующий вид:

mrk	expr
e1	*Ситуация(e1, выпуск1 *(Агент1, нек множ *(Кач-состав, предприятие) : x1) (Объект1, нек множ * (Кач-состав, дин. физ. об. * (Класс1, мебель)(Цел место, нек кухня))))*
e2	*Ситуация(e2, поставка1 *(Агент1, нек орг *(Тип, АО)(Назв, "Старт") : x3) (Объект1, нек множ *(Кач-состав, дин.физ.об. *(Вещество, картон)) : x2)(Адресат, x1))*

Рис. 7.8. Пример конфигурации массива описания ситуаций Sitdescr

По такому массиву Sitdescr можно построить следующее КП *Semrepr* вида *Вопрос (x1, (Sitdescr[1] ∧ Sitdescr [2]))* , т.е. выражение

*Вопрос (x1, (Ситуация(e1, выпуск1 *(Агент1, нек множ *(Кач-состав, предприятие) : x1) (Объект1, нек множ * (Кач-состав, дин. физ. об. * (Класс1, мебель)(Цел место, нек кухня))))*

*∧ Ситуация(e2, поставка1 *(Агент1, нек орг *(Тип, АО)(Назв, "Старт") : x3) (Объект1, нек множ *(Кач-состав, дин.физ.об. *(Вещество, картон)) : x2)(Адресат, x1)))*

Начальным шагом формирования строки массива Sitdescr с меткой ситуации *ek* является построение выражения вида *Ситуация (ek, concept * ,* где *concept* – семантическая единица, квалифицирующая ситуацию и являющаяся значением поля *semnoun* массива Arls (проекции лексико-семантического словаря Lsdic на входной текст T) для строки, номер которой указан в столбце locunit матрицы Matr.

Например, в случае рассмотрения вопроса B1 первая и вторая строки массива Sitdescr получат соответственно значения *Ситуация(e1, выпуск1 * и Ситуация(e2, поставка1* **.**

После того, как сформировано начальное значение рассматриваемой строки массива Sitdescr, необходимо добавить в эту строку описания участников ситуации и соответствующие тематические роли. Для этого используется массив Performers

(“Исполнители-ролей”). Количество строк в этом массиве совпадает с количеством строк в классифицирующем представлении Rc и в МССП Matr

Наиболее простую структуру имеют семантические представления таких описаний участников ситуаций, которые являются сочетаниями вида “Существительное + Имя”, где Имя – это выражение в кавычках или апострофах. Пусть k – порядковый номер в тексте существительного из такого выражения, т.е. порядковый номер строки классифицирующего представления, соответствующей этому существительному. Тогда Performers [k] = *нек conc * (Назв, Имя)* ,
где *conc* – простое обозначение понятия, соответствующего данному существительному.

Например, для вопроса В1 Performers [9] = *нек акц-общ * (Назв, “Старт”)* ,
поскольку выражение АО занимает 9-е по порядку место в вопросе В1.

В лаконичной форме принципы заполнения массива Performers иллюстрирует слкдуюшая таблица. Первый столбец таблицы соответствует существительному с порядковым номером k, где $k \geq 1$; второй столбец – контексту для данного существительного, т.е. виду сочетания, в которое входит данное существительное; в третьем столбце указывается значение строки k массива Performers.

t_i	контекст	Performers [I]
контейнер	Поступил контейнер	*нек контейнер1*
контейнеры	Поступили контейнеры	*нек множ *(Кач-состав, контейнер1)*
контейнера	3 контейнера	*нек множ * (Колич, 3)(Кач-состав, контейнер1)*
контейнера	3 алюминиевых контейнера	*нек множ * (Колич, 3)(Кач-состав, контейнер1 * (Материал, алюм))*
контейнера	3 зел. алюмини-евых контейнера	*нек множ * (Колич, 3)(Кач-состав, контейнер1 * (Цвет, зел)(Материал, алюм))*

Ростов	Ростов	*нек город * (Назв, "Ростов")*
АО "Заря"	АО "Заря"	*нек орг *(Тип, АО)(Назв, "Заря")*
году	в 1998 году	*1998/nil/nil*
февраль	с февраля 1998	*1998/февраль/nil*
керамикой	с керамикой	нек множ *(Кач-состав, дин.физ.об. * (Вид, керамич-изделие))
обувью	с обувью	нек множ * (Кач-состав, дин.физ.об. * (Вид, обувн-изделие))
обувь	с обувью из Италии	нек множ *(Кач-состав, дин.физ.об. * (Вид, обувн изделие)(Изготовитель, нек страна * (Назв, "Италия")))
пластмассу	огнеустойчивую пластмассу	нек множ *(Кач-состав, дин. физ.об. * (Вещество, пластм *(Характ, огнеустойчив)).

Таблица 7.1. Виды элементов массива Performers

Пример. На основе указанных принципов по построенному в параграфе 7.2 МССП Matr текста T1 = "В каком московском издательстве в 2001-м году вышла работа по искусственному интеллекту «Основы обработки знаний» профессора Сомова ?" можно построить следующее К-представление (КП):

Вопрос (x1, Ситуация(e1, выход 1 (Простр.объект, нек изд-во1* (Место, Москва) : x21)(Время, 2001/год)(Объект3, нек работа2 * (Назв, 'Основы обработки знаний')(Область1, иск-интеллект)(Авторы, нек чел *(Квалификация, профессор)(Фамилия, 'Сомов') : x4) : x3))))) .*

Для фомирования массивов Performers и Sitdescr используются вспомогательные массивы Sembase и Semdes с количеством строк nt (количество единиц текста в класифицирующем представлении Rc входного текста T. Первоначально элементы этих масствов заполняются цепочкой nil.

Массив Sembase предназначен для явного отображения информации о семантике прилагательных и существительных из входного текста T.

Пример. Пусть В1 = “Какие европейские фармацевтические компании участвовали в выставке “ЭКСПОХИМ-2003”?”. Тогда в в класифицирующем представлении Rc вопроса В1 словоформам “европейские”, “фармацевтические”, “компании”, “выставке” будут соответствовать строки с номерами 2, 3, 4, 7. Лингвистический базис может быть определен таким образом, что в результате применения описываемого ниже алгоритма “Начало-постр-СемП” будут выполненв следующие операторы присваивания:

Sembase[2] := (Регион, Европа) ,

Sembase[3] := (Регион, Европа) (Область1, фармацевтика) ,

Sembase[4] := компания1 * (Регион, Европа) (Область1, фармацевтика) ,

Sembase[7] := выставка1 .

Цепочку Sembase[4] будем интерпретировать как составное обозначенияе понятия “европейская фармацевтическая компания”, но не как обозначение какой-то конкретной компании, цепочку Sembase[7] - как обозначение понятия “выставка”.

Массив Semdes предназначен для построения главных частей семантических представлений выражений из входного текста, обозначающих объекты или множества объектов (т.е. выражений, включающих существительные).

Пример. В контексте вопроса В1 выражение “европейские фармацевтические компании” обозначает некоторое конкретное множество компаний, а слово “выставка” – некоторую конкретную выставку. Поэтому

*Semdes[4] := нек множ * (Кач-состав, компания1 * (Регион, Европа) (Область1, фармацевтика)) ,*

*Semdes[7] := нек выставка1 * (Назв, 'ЭКСПОХИМ-2003’)* .

Здесь *нек* - информационная единица, интерпретируемая как квантор референтности (см. параграф 2.8). Поскольку выражения Semdes[4] и Semdes[7] начинаются с этой информационной единицы, постольку эти

выражения интерпретируются как обозначения конкретных сущностей, упоминаемых в тексте, а не как обозначения понятий.

Так как в позициях 2, 3 вопроса В1 расположены прилагательные, то Semdes[2] = Semdes[3] = nil .

В массиве Performers отличными от цепочки nil являются только элеиенты, соответствуюшие конструктам (числовым значениям параметров) или существительным. Если элемент в позиции k является конструктом, то Sembase[k] = Semdes[k] = nil , Performers [k] := Rc [k, unit] , где Rc - класифицирующее представление входного текста Т. Например, если позиции k соответствует значение цены учебника 112 рублей , то Performers [k] := 112/рубль .

Если же k – номер строки из Rc , соответствующей существительному, то Performers [k] := Semdes[k] + ' : v ‘ , где v – переменная, являющаяся меткой сущности в СП входного текста; здесь символ ‘+ ‘ будем интерпртировать как знак операции конкатенашии, т.е. операции приписывания справа к одной цепочке другой цепочки.

Пример. Для вопроса В1 *Performers [4] := Semdes[4] + ' : S1 ‘ ,*
Performers [7] := Semdes[7] + ' : x1 ‘ , т.е.
*Performers [4] := нек множ * (Кач-состав, компания1 * (Регион, Европа) (Область1, фармацевтика)) : S1 ,*
*Performers [7] := нек выставка1 * (Назв, 'ЭКСПОХИМ-2003’) : x1 .*

7.3.2. Формулировка метода

Введенные выше понятия и изложенные принципы позволяют сформулировать новый метод преобразования ЕЯ-текста (в частности, запроса, сообщения или команды) в СП текста. Эта метод предназначен для проектирования диалоговых систем и включает следующие три этапа преобразования:

Преобразование1: Компонентно-морфологический анализ входного текста..

Сущность преобразования заключается в следующем. По тексту Т на естественном языке строится одно или несколько компонентно-морфологических представлений (КМП) текста Т, т.е. один или несколько наборов вида (Rc, Rm), где Rc - классифицирующее представление текста и Rm – морфологическое представление текста, т.е. представление возможных значений морфологических признаков для тех компонентов текста Т, которые являются лексическими единицами (в отличие от числовых значений признаков, разделителей, выражений в кавычках или апострофах).

В большинстве случаев отдельным фразам из входного текста будет соответствовать единственное КМП. Если же либо входной текст Т неоднозначно разбивается на элементарные значащие единицы текста, либо неоднозначно определяется часть речи какой-либо единицы текста, то задаются уточнящие вопросы пользователю диалоговой системы, и неоднозначности снимаются после обработки ответов пользователя на эти вопросы.

Преобразование 2: Построение матричного семантико-синтаксического представления (МССП) текста.

Цель второго преобразования заключается в том, чтобы связать с каждым словом какое-то одно из возможных нескольких значений и в том, чтобы установить смысловые отношения между различными единицами текста.

Так как это делается постепенно, шаг за шагом, то МССП сначала является недоопределенным. Чтобы снять неоднозначности, могут задаваться уточняющие вопросы пользователю. Но, главным образом, используются сведения из лингвистической базы данных (ЛБД) о допустимых способах комбинирования разных единиц текста в лингвистически правильные сочетания.

Преобразование 3: Сборка семантического представления текста, являющегося К-представлением, по его МССП *Matr*.

Алгоритм, преобразующий МССП *Matr* в некоторое формальное выражение $Semrepr \in Ls(B)$, где *B* – концептуальный базис, являющийся первым компонентом используемого размеченного концептуального базиса (р.к.б.) *Cb*, *Ls(B)* – СК-язык в базисе *B*, будем называть *алгоритмом семантической сборки*.

7.3.3. Принципы выбора формы семантического представления для текстов различных видов

Форма семантического представления (СП) ЕЯ-текста Т, строящегося по МССП текста Т, должна зависеть от вида входного текста. Рассмотрим на примерах рекомендации по выбору формы СП, являющегося выражением стандартного К-языка (СК-языка) в некотором концептуальном базисе, т.е. К-представлением (КП) входного текста. В этих примерах СП входного текста Т будет являться значением строковой переменной Semrepr (Semantic representation).

Пример. Пусть Т1 = “Профессор Игорь Новиков преподает в Томске”. Тогда Semrepr = *Ситуация(e1, преподавание * (Время, #сейчас#)(Агент1, нек чел * (Квалиф, профессор)(Имя, ‘Игорь‘(Фамилия, 'Новиков’) : x2))(Место1, нек город * (Название, 'Томск’) : x3)).*

Пример. Пусть Т2 = “"Доставь ящик с деталями на склад № 3.". Тогда

Semrepr = *(Команда(#Оператор#, #Исполнитель#, #Сейчас#, e1) ∧ Цель (e1, доставка1*(Объект1, нек ящик * (Содерж1, нек множ * (Кач-состав, деталь)) : x1)(Место2, нек склад * (Номер, 3) : x2)) .*

Пример. Пусть Т3 = “Проходила ли в Азии международная научная конференция “COLING”?”. Тогда

Semrepr = *Вопрос(x1, (x1≡ Ист-знач (Ситуация (e1, прохождение2* (Время, нек мом * (Раньше ,#сейчас#) : t1)(Событие, нет конф* (Вид1, междун) (Вид2, научная) (Название, ‘COLING’) : x2) (Место, нек континент* (Название, ‘Азия’) :x3))))).*

Пример. Пусть Т4 = “Какое издательство опубликовало роман «Ветры Африки»?”. Тогда Semrepr = *Вопрос(x1, Ситуация(e1, опубликование * (Время, нек мом * (Раньше, #сейчас#) : t1) (Агент2, нек издательство: x1) (Объект3, нек роман1* (Название, ‘Ветры Африки’) :x3))) .*

Пример. Пусть Т5 = “ С какими зарубежными издательствами сотрудничает писатель Игорь Сомов?”. Тогда Semrepr = *Вопрос (S1, (Кач-состав (S1, издательство * (Вид-географич, зарубежное)) ∧ Описание (произв издательство**

*(Элем, S1) : y1, Ситуация(e1, сотрудничество * (Время, #сейчас#)(Агент1, нек чел* (Профессия, писатель)(Имя, 'Игорь')(Фамилия, 'Сомов'): x1)(Организация1, y1)))))* .

Пример. Пусть T6 = "Кем выпускается препарат "Зиннат"?".
Тогда Semrepr = *Вопрос (x1, Ситуация (e1, выпуск1 * (Время, #сейчас#) (Агент1, x1)(Продукция1, нек препарат1 * (Название, 'Зиннат') : x2)))* .

Пример. Пусть T7 = "Откуда и для кого поступил трехтонный алюминиевый контейнер?". Тогда Semrepr = *Вопрос ((x1 ∧ x2) , Ситуация (e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1) (Место1, x1) (Адресат, x2) (Объект1, нек контейнер * (Вес, 3/ тонна))(Материал, алюминий) : x3))))* .

Пример. Пусть T8 = "Сколько человек участвовало в создании статистического сборника?". Тогда Semrepr = *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, чел) ∧ Описание(произв чел* (Элемент, S1) : y1, Ситуация(e1, участие1* (Время, нек мом * (Раньше, #сейчас#) : t1) (Агент1, y1)(Вид-деятельности, создание1* (Продукт1, нек сборник1 * (Область1, статистика) : x2)))))* .

Пример. Пусть T9 = "Сколько раз Иван Михайлович Семёнов летал в Мексику?". Тогда Semrepr = *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, сит) ∧ Описание(произв сит * (Элемент, S1) : e1, Ситуация (e1, полёт * (Время, нек мом * (Раньше, #сейчас#) : t1)(Агент1, нек чел.*(Имя, 'Иван')(Отчество, 'Михайлович')(Фамилия, 'Семёнов'): x2)(Место2, нек страна* (Название, 'Мексика') : x3)))))* .

7.4. Обсуждение разработанного метода преобразования ЕЯ-текстов в семантические представления

Изложенный метод, базирующийся на построенной формальной модели лингвистической базы данных (ЛБД) и на введенном понятии матричного семантико-синтаксического представления (МССП), направлен на непосредственное установление смысловых отношений между элементарными

значащими единицами входного текста, отражая эти отношения посредством МССП, и на последующее построение семантического представления (СП) текста, являющегося выражением некоторого СК-языка (К-представлением). Рассматриваемые тексты могут выражать высказывания (сообщения), команды, специальные вопросы (т.е. вопросы с вопросительными словами), общие вопросы (т.е. вопросы с ответом «Да»/ «Нет»). Тексты могут, в частности, включать причастные обороты и придаточные определительные предложения.

Метод позволяет устанавливать возможные смысловые отношения, в частности, в сочетаниях «Глагол + Предлог + Существительное», «Глагол + Существительное», «Существительное1 + Предлог + Существительное2», «Число + Существительное», «Прилагательное + Существительное», «Существительное1 + Существительное2», «Причастие + Существительное», «Причастие + Предлог + Существительное», «Вопросительно-относительное местоимение или местоименное наречие + Глагол», «Предлог + Вопросительно--относительное местоимение + Глагол».

Работоспособность изложенного метода реализации преобразования «Текст→СП» доказана созданием на его основе сложного структурированного алгоритма семантико-семантического анализа ЕЯ-текстов (см. главы 8, 9) и серии семантико-семантических анализаторов вопросов, команд и сообщений в системах программирования TurboPascal 7.0., C, C++, Delphi 4.0, 5.0, PHP.

Предложенный метод намечает принципиально новый подход к семантико-синтаксическому анализу ЕЯ-текстов.

Метод явно учитывает многозначность слов, что чрезвычайно важно для приложений и является его существенным преимуществом.

Важная особенность метода заключается в том, что он не предусматривает использования синтаксического уровня представления текста (как результата выполнения синтаксического анализа), в то время

как синтаксический уровень представления используется в течение нескольких десятилетий как в нашей стране, так и за рубежом.

Характер данных, описываемых формальной моделью ЛБД, и направленность предложенного метода на непосредственное выявление смысловых отношений между элементами текста с целью построения его СП позволяют провести некоторые параллели между разработанным методом и идеями компьютерной семантики русского языка.

Например, в статье (Тузов 2001) одно из возможных значений предлога “к” (значение “Куда”, т.е. значение “Направление движения”) представлено выражением *@ Куда К ($ 12 ~ @ Дат острову)* .

То же значение предлога “к” может быть отображено предложным фреймом вида *(j, ‘к ‘ , соб , простр.об , 4 , Куда, ‘к острову’)* из словаря *Frp* (см. параграф 6.7), являющегося компонентом лингвистического базиса Lingb. В последнем выражении *j* является порядковым номером фрейма, *соб* – сорт “событие”, *простр.об* – сорт “пространственный объект”, 4 - код винительного падежа, *Куда* – обозначение смыслового отношения “Направление движения”, *'к острову’* – пример реализации этого отношения.

Таким образом, в этом случае мы видим использование, по существу, одной и той же структуры данных. Однако необходимо отметить, что, в отличие от содержания данной главы и главы 7, в публикациях по компьютерной семантике русского языка не построена формальная модель ЛБД, не намечены контуры такой модели и не предлагается структура данных для представления промежуточных результатов семантико-синтаксического анализа ЕЯ-текстов.

Использование аппарата СК-языков для построения СП входных текстов ЛП позволило преодолеть трудности принципиального характера, касающиеся отображения содержания команд, а также вопросов нескольких видов: с вопросительными словами “какие”, “каких” и т.д., со словом “сколько”, относящимся к количеству предметов, и с ответом “Да /Нет”.

Важное преимущество изложенного нового подхода к разработке алгоритмов ССА заключается в создании предпосылок для облегчения подготовки специалистов в области лингвистических информационных технологий. Предложенный подход направлен на непосредственный поиск смысловых отношений между участниками ситуаций, и эти смысловые отношения понятны специалистам из рассматриваемой конкретной области (при этом предметная область может меняться). Как следствие, разработанный подход не требует овладения обширной лингвистической терминологией. Для понимания метода достаточно знакомства с базовыми математическими понятиями (множество, последовательность, цепочка, *n*-арное отношение, функция) и рядом понятий из курса русского языка по программе средней школы .

Глава 8
АЛГОРИТМ ПОСТРОЕНИЯ МАТРИЧНОГО СЕМАНТИКО-СИНТАКСИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ЕСТЕСТВЕННО-ЯЗЫКОВОГО ТЕКСТА

8.1. Постановка задачи разработки алгоритма семантико-синтаксического анализа текстов

8.1.1. Назначение и крупноблочная структура алгоритма

Цель данной главы и главы 9 заключается в разработке алгоритма семантико-синтаксического анализа текстов из подъязыков русского языка (РЯ), реализующего предложенный в главе 7 новый метод выполнения преобразования “ЕЯ-текст – Семантическое представление (СП) текста”. При этом предложенное в главе 6 формальное понятие лингвистического базиса интерпретируется как описание структуры лингвистической базы данных (ЛБД), используемой алгоритмом. Рассматриваемые тексты могут выражать высказывания (сообщения), команды, специальные вопросы (т.е. вопросы с вопросительными словами), общие вопросы (т.е. вопросы с ответом «Да»/ «Нет»). Тексты могут, в частности, включать причастные обороты и придаточные определительные предложения.

Для реализации предложенного в главе 7 нового метода выполнения преобразования “ЕЯ-текст –> СП текста” ставится задача разработки алгоритма SemSyn, являющегося композицией некоторых алгоритмов BuildMatr и BuildSem, удовлетворяющих следующим условиям:

(1) BuildMatr – алгоритм преобразования текстов из некоторых практически интересных подъязыков русского языка в их матричные семантико-синтаксические представления (МССП);

(2) BuildSem – алгоритм сборки семантического представления ЕЯ-текста по его МССП, причем построенное СП текста является выражением

некоторого стандартного К-языка (т.е. является К-представлением входного текста).

Отправной точкой для разработки алгоритма являлся анализ поверхностной и смысловой структуры текстов из следующих подъязыков русского языка и английского языка, представляющих практический интерес:

- вопросы и сообщения на естественном языке к поисковой Интернет-системе нового поколения, касающиеся научных публикаций и участия специалистов в различных научных конференциях;
- команды и вопросы транспортно-погрузочному интеллектуальному роботу, в частности, роботу, действующему на автоматизированном складе, и роботу, действующему в аэропорту;
- вопросы и сообщения для базы данных, касающиеся выпуска, экспорта и импорта продукции различными предприятиями, фирмами;
- вопросы, с которыми оператор автоматизированного склада обращается к интеллектуальной базе данных;
- вопросы потенциальных покупателей к интеллектуальной базе данных Интернет-магазина.

8.1.2. Набор текстов, рассматривавшийся в качестве ориентира при разработке алгоритма

Познакомимся с примерами входных текстов, которые могут анализироваться алгоритмом, разработанным в данной главе. Тексты в примерах будут рассматриваться в качестве типичных представителей определенных подклассов входных текстов.

1. Сообщения (описания фактов)

1. Профессор Игорь Петрович Сомов опубликовал в 1996 - 2001 годах 8 статей в зарубежных научных журналах.
2. Академик Иван Петрович Павлов разработал теорию условных рефлексов.
3. Фирма “GlaxoWelcome” выпускает лекарства для больных астмой.
4. АО “Парус” с 1999 года экспортирует продукцию в Болгарию.

2. Команды

1. Отправить контейнеры фабрике “Заря” до 16:30.
2. Подтащи металлическую балку к самосвалу.

3. Погрузи 4 контейнера на рейс компании «Люфтганза».

3. Частноутвердительные (или общие) вопросы

1. Экспортирует ли продукцию в Болгарию АО “Парус”?

4. Вопросы с вопросительно-относительным местоимением “какой”

1. Какой контейнер предназначен для АО “Радуга”?
2. Каким рейсом улетел профессор Семенов?
3. Какие препараты, выпускаемые фирмой “GlaxoWelcome”, предназначены для больных астмой?
4. Какую монографию опубликовал в прошлом году профессор Семенов?
5. В каком университете работет профессор Игорь Сергеевич Сомов, о котором писала газета “Поиск” в ноябре 2002-го года?
6. В какие страны экспортируют продукцию предприятия, расположенные в Саратовской области?
7. Какие контейнеры с индийской керамикой, поступившие в пятницу, предназначены для АО “Парус“?
8. В каких странах выпускается препарат бекатит?
9. Какие европейские страны выпускают препарат серетид?
10. В какие страны экспортирует станки объединение “Радуга”?
11. Из каких стран получает оборудование завод “Старт”?
12. Какие статьи опубликованы профессором Семеновым в 2001 г.?

5. Вопросы частноинформативного актуально-синтаксического типа

В работе (Воробьева, Панюшева, Толстой 1975) к этому классу вопросов относят такие вопросы, когда спрашивающий не знает часть информации о ситуации и просит собеседника восполнить неизвестный ему аспект интересующего его факта.

Вопросы этого класса можно было бы назвать ролевыми, поскольку они начинаются с одного или нескольких вопросительных слов или словосочетаний, каждое из которых связано с определенной тематической ролью. Примеры таких вопросов приведены ниже:

1. Откуда и для кого поступили 3 двухтонных контейнера с индийской керамикой ?

2. Где выступал в 2003-м году профессор Новосельцев из МВТУ?
3. Где работает профессор Сомов?
4. Кто разработал теорию условных рефлексов ?
5. Кто выпускает препарат серетид ?
6. Где находится центральный офис фирмы “GlaxoWelcome”?
7. Для кого предназначены два контейнера с итальянской обувью ?

6. Вопросы относительно количества предметов

1. Сколько статей, опубликованных Игорем Сомовым с 1996 года, относятся к искусственному интеллекту?

2. Сколько трехтонных контейнеров, поступивших в пятницу из Ростова, предназначены для АО «Парус»?
3. Откуда и для кого поступили 3 двухтонных контейнера с индийской керамикой ?

7. Вопросы относительно количества событий

1. Сколько раз в этом году запрашивался учебник Коробова?
2. Сколько раз в прошлом году профессор Коробов участвовал в международных научных конференциях?

8.2. Формализация исходных предположений о рассматриваемых подъязыках естественного (русского) языка

Представим в формальном виде основную часть исходных предположений о структуре текстов из рассматриваемых подъзыков ЕЯ. Хотя анализ будет

касаться подъязыков русского языка, тот же метод применим и для описания поверхностной структуры текстов на английском, немецком, французском и многих других языках. Теоретической основой анализа являются понятие лингвистического базиса, введенное в предыдущей главе, и понятие бесконтекстной грамматики – одно из центральных понятий теории формальных грамматик, языков и трансляторов.

Будем использовать аппарат бесконтекстных грамматик для описания расширенного входного языка лингвистического анализатора. Построим бесконтекстную грамматику G, которая будет описывать слова тех частей речи, которые могут встречаться во входном языке, и способы комбинирования слов, относящихся к различным частям речи, чисел и числовых значений параметров.

Как известно, бесконтекстной грамматикой (или контекстно-свободной грамматикой, КС-грамматикой) называется упорядоченная четверка G вида (N, T, P, s_0), где N, T - конечные непересекающиеся множества символов, P - конечное множество выражений вида $s \rightarrow u$, где $s \in N$, u – цепочка (возможно, пустая) в алфавите $N \cup T$, $s_0 \in N$.

Элементы множеств N и T называются нетерминальными и терминальными символами, элементы множества P называются продукциями, s_0 называется начальным символом.

Будем использовать компактную форму записи $s \rightarrow u_1 \mid u_2 \mid u_3 \mid \ldots \mid u_n$ для нескольких продукций вида $s \rightarrow u_1$, $s \rightarrow u_2$, …. , $s \rightarrow u_n$. Кроме того, будем использовать в продукциях символ ::= вместо символа → .

Таким образом, из соображений компактности мы будем рассматривать ниже бесконтекстные грамматики в форме Бэкуса-Наура (Johnsonbaugh 2001).

Построим некоторую бесконтекстную грамматику в форме Бэкуса-Наура. Нетерминальными символами (или нетерминалами) этой грамматики будут служить выражения вида $< s_1, \ldots, s_n >$, где $n \geq 1$, для $k = 1, \ldots, n$ s_k - буква русского алфавита, цифра или знак – (тире). Такие выражения будут интерпретироваться как метки определенных фрагментов текста либо метки частей речи. Например, нетерминалами строящейся грамматики будут являться выражения <вопрос> , <команда> , <сообщение> , <вопрос-на-перечисление>, <да-нет-вопрос>, <зависимое-выражение>, <описание-участников-события> ,

<числовая-часть> . Начальным символом грамматики будет являться нетерминал <текст>.

Терминальными символами (или терминалами) этой грамматики будут служить элементы “?”, “,” , “.” (т.е. вопросительный знак, запятая и точка), “Сколько” , “ Сколько раз” , “ли” и элементы (какой), [сущ] , [прилаг] , [глаг] , [прич], [предлог], [колич-числит] , [число], [наречие] , [конструкт] , [имя], {сущ-нарицат} , {сущ-собств} , {глаг-в-неопред-форме}, {глаг-в-изъявит-накл} , {глаг-в-повелит-накл} , {вопр-относит-местоим}, {местоим-наречие}.

Рассмотрим следующую систему продукций:

<текст > ::= <вопрос> | <команда> | <сообщение> ;

<вопрос> :: = <да-нет-вопрос> | <вопрос-о-тематич-ролях> |

<колич-вопрос1> | <колич-вопрос2> | <вопрос-на-перечисление> ;

<да-нет-вопрос>: : = {глаг-в-изъявит-накл} “ли” <зависимое-выражение> “?” ;

<зависимое-выражение>:: = <описание-сущности> | [конструкт] | <зависимое-выражение><зависимое-выражение>;

<описание-сущности> : := <возм-предлог ><числовая-часть> <атрибутная-часть> <ядро-описания-сущности> ;

<возм-предлог > : := | <предлог > ; (5.2.1)

<числовая-часть> :: = | [число] | [колич-числит];

<атрибутная-часть > : := | [прилаг] [прилаг] <атрибутная-часть> ;

<ядро-описания-сущности> : : = [сущ] | {сущ-нарицат} [имя] | {сущ-нарицат} <послед-сущ-собств > | <послед-сущ-собств > | [сущ] <причастн-оборот> |

[сущ] <придаточное-опред-предложение> ;

<послед-сущ-собств > : := {сущ-собств} | {сущ-собств} <послед-сущ-собств > | {сущ-нарицат} <послед-сущ-собств > ;

<вопрос-о-тематич-ролях> ::= < ролевое-вопр-сочетание> <сообщение> “?” ;

< ролевое-вопр-сочетание> ::= < местоим-сочетание> | {местоим-наречие} | < ролевое-вопр-сочетание> <связка> < ролевое-вопр-сочетание> ;

< местоим -сочетание> ::= <возм-предлог > {вопр-относит-местоим} ;

<связка> ::= “ , “| “и “ ;

<колич-вопрос1> ::= “Сколько” <описание-сущности> <неполное-сообщение> “?” ;

<колич-вопрос2> :: =

“Сколько раз” <описание-сущности> <неполное-сообщение> “?” ;

<вопрос-на-перечисление> :: = <возм-предлог > (какой) <сообщение> “?” ;

<команда> : : = <действие> <описание-участников-события>;

<описание-участников-события> :: = <зависимое-выражение> |

<зависимое-выражение> <описание-участников-события> ;

<действие> : : = {глаг-в-повелит-накл} | {глаг-в-неопред-форме};

<причастн-оборот> : : = <возм-запятая> [прич] <простое-описание-участников-события> ;

<простое-описание-участников-события> ::= <простое-зависимое-выражение> |

<простое-зависимое-выражение>< простое-зависимое-выражение> ;

<придаточн-опред-предложение>::=<присоединяющая-часть><простое-сообщение>;

<присоединяющая-часть> ::= <возм-запятая> <возм-предлог> <вопр-относит-местоим> ;

<возм-запятая> ::= | “,” ;

<вопр-относит-местоим> ::= (который)(какой);

<простое-сообщение>::= <простое-описание-участников-события>{глаг-в-изъявит-накл}<возм-точка> | <простое-описание-участников-события>{глаг-в-изъявит-накл}<возм-точка><простое-описание-участников-события> ;

<возм-точка>. ::= | “.” ;

<простое-зависимое-выражение> ::= <простое-описание-сущности> | [конструкт] ;

<простое-описание-сущности> ::= <возм-предлог > <числ-часть><атрибутная-часть>< простое-ядро-описания-сущности> ;

<простое-ядро-описания-сущности> ::= [сущ] | {сущ-нарицат} [имя] | {сущ-нарицат} <послед-сущ-собств > | <послед-сущ-собств > ;

<расширенное-ядро-описания-сущности> ::= <простое-ядро-описания-сущности> > | [сущ] <причастн-оборот> |

[сущ] <придаточн-опред-предложение> ;

<сообщение> ::= <описание-участников-события> {глаг-в-изъявит-накл} <правая-часть-сообщения> <возм-точка> ;

<правая-часть-сообщения> :: = | <описание-участников-события> ;

<неполн-сообщение> :: = {глаг-в-изъявит-накл} <описание-участников-события>.

Пример. Пусть В1 = “В каком петербургском издательстве в 2000-м году вышла книга “Базы знаний интеллектуальных систем”? ”. Тогда в грамматике с системой продукций (5.2.1) можно выполнить следующую систему замен нетерминалов на правые части продукций, приводящую к выводу цепочки, дающей обобщенное описание структуры вопроса В1:

<текст > = > <вопрос> ,

<вопрос> = > <вопрос-на-перечисление> ;

<вопрос-на-перечисление> = > <возм-предлог > (какой) <сообщение> “?” ;

<возм-предлог > (какой) <сообщение> “?” = >

[предлог] (какой) <сообщение> “?” ;

<сообщение> => <описание-участников-события> {глаг-в-изъявит-накл} <правая-часть-сообщения> <возм-точка>

=> <описание-участников-события><описание-участников-события> {глаг-в-изъявит-накл} <описание-участников-события> =>

[прилаг] [сущ] [предлог] [конструкт] {глаг-в-изъявит-накл} [сущ-нарицат] [имя] .

Поэтому, очевидно, <текст > => Expr1 , где Expr1 – цепочка вида

[предлог] (какой) [прилаг] [сущ] [предлог] [конструкт]

{глаг-в-изъявит-накл} [сущ-нарицат] [имя] “?” . (8.2.2)

Определение. Пусть *G = (N, T, P, s)* – бесконтекстная грамматика с множеством нетерминальных символов (нетерминалов) *N* , множеством терминальных символов (терминалов) *T,* множеством продукций *P* и начальным символом *s*. Тогда через *L (G)* обозначим множество всех цепочек в алфавите *T* , выводимых из *s* с помощью продукций из *P* .

Определение. Пусть *Lingb* – лингвистический базис вида (8.8.2), *Qmk* – разметка вопросов вида (6.2.1), *G = (N, T, <текст>, P)* – произвольная бесконтекстная грамматика с нетерминалами вида <a > и терминалами видов *(b) ,“c”, [d], {h}* , где *a* – некоторое выражение в русском алфавите,

обогащенным символом ‘-‘ ; $b \in Lecs$, *где* $Lecs$ – множество лексем текстообразующей системы, являющейся компонентом $Lingb$; с $\in W$, где W – множество словоформ морфологического базиса, являющегося компонентом $Lingb$; $d \in Classes(Lingb)$, $h \in Subclasses(Lingb)$, и <*текст*> – начальный символ G .

Тогда через $Linp(G, Lingb)$ обозначим множество всех цепочек, каждая из которых может быть получена из некоторой цепочки x из языка $L(G)$ выполнением одного или нескольких из следующих преобразований: (1) терминал вида (b) заменяется на произвольную словоформу $u \in W(Morphbs(Tform))$, такую, что $lcs(u) = b$ (т.е. b является лексемой словоформы u) ; (2) терминал вида *“c”* заменяется на c ; (3) терминал вида $[d]$ заменяется на произвольный такой элемент $u \in Textunit$s $(Tform)$, что $tclass(u) = d$; (4) терминал вида $\{h\}$ заменяется на произвольный такой элемент z $\in Textunits(Tform)$, что $subclass(z) = h$.

Пример. Нетрудно определить такой лингвистический базис $Lingb$, что язык $Linp(G, Lingb)$ включает вопрос B1 из предыдущего примера, и B1 может быть получен из цепочки Expr1 вида (8.2.2) применением перечисленных выше преобразований.

Таким образом, выше предложен новый метод формального описания предположений о структуре входных текстов лингвистического процессора на основе комбинированного использования аппарата бесконтекстных грамматик и понятия лингвистического базиса, введенного в 6-й главе.

8.3. Начальные этапы разработки алгоритма построения матричного семантико-синтаксического представления входного текста лингвистического процессора

8.3.1. Назначение алгоритма

Разрабатываемый в данной главе алгоритм BuildMatr предназначен для преобразования входных текстов из подъязыков русского языка (РЯ) в их матричные семантико-семантические представления (МССП). Операции, осуществляемые по входному тексту, зависят от выбранного лингвистического

базиса (л.б.) *Lingb*, интерпретируемого как формальная модель лингвистической базы данных (ЛБД).

При этом входной текст T должен являться выражением языка *Linp (G, Lingb),* где *G* – бесконтекстная грамматика с системой продукций вида (8.2.1), построенная в предыдущем параграфе. Главными выходными данными алгоритма должны являться строка *kindtext*, задающая вид входного текста (т.е. классифицирующая этот текст), и строково-числовая матрица *Matr* – МССП входного текста.

Известно, что проблема семантико-синтаксического анализа (ССА) компьютером ЕЯ-текстов включает много аспектов, и разные аспекты этой проблемы проработаны в разной степени. Начальный этап ССА текстов на РЯ включает нахождение базовых форм словоформ из входного текста, нахождение возможных значений морфологических признаков (число, падеж, лицо, время и т.д.) для этих словоформ и разбиение текста на такие сегменты, которые соответствуют определенным элементарным единицам смыслового уровня. К таким сегментам относятся, например, выражения “были отправлены”, “будет подготовлен”, “Олимпийские игры”, “840 км” , “1999-й год” , “2 часа”, “пять градусов“.

Вопросы автоматизации морфологического анализа текстов на русском языке исследованы во многих публикациях.

Вопросы, касающиеся автоматического выделения таких коротких сегментов текста, которые обозначают элементарные единицы смыслового уровня, оказываются не слишком сложными с логической точки зрения и могут решаться непосредственно на уровне программной реализации алгоритма.

Анализ литературы и собственный опыт автора говорят о том. что основные трудности логического характера при автоматизации ССА ЕЯ-текстов касаются поиска смысловых отношений между компонентами входного текста.

В связи с этим при разработке алгоритма BuildMatr основной акцент делается на формализации и алгоритмизации процесса поиска смысловых отношений между компонентами входного текста. Реализация этого акцента достигается следующим образом:

1. Во многих случаях входной текст T алгоритма является некоторой абстракцией по сравнению с реальным входным текстом лингвистического процессора. Дело в том, что единицами входного текста алгоритма могут быть не только словоформы, но и короткие словосочетания (“были отправлены”, “будет подготовлен”, “Олимпийские игры”, “840 км” , “1999-й год” , “2 часа”), являющиеся обозначениями элементарных единиц смыслового (семантического, концептуального) уровня.
2. Постулируется существование отображений, ставящих в соответствие словоформам их базовые формы (лексемы) и наборы значений морфологических признаков, а также связываюшие с каждым конструктом (числовым значением какого-то параметра) некоторую семантическую (или концептуальную) единицу; с этой целью в главе 4 было введено понятие текстообразующей системы.

Известны два основных подхода к разработке алгоритмов: проектирование сверху вниз (нисходящее проектирование) и проектирование снизу вверх (восходящее проектирование). Представляется целесообразным при разработке сильно структурированного алгоритма *Semsyn* сочетать оба этих метода. Это сочетание методов нисходящего и восходящего проектирования алгоритмов нашло отражение в описании алгоритма BuildMatr в последующих параграфах данной главы и в описании алгоритма BuildSem в главе 9.

В тех случаях, когда вспомогательный алгоритм является весьма простым (непосредственно программируемым) либо его разработка не представляет теоретических трудностей с учетом имеющихся научных публикаций, в структурированных описаниях алгоритмов BuildMatr и BuildSem указывается только внешняя спецификация такого алгоритма, т.е. описание назначения, входных и выходных данных алгоритма.

Для описания алгоритмов в данной главе используется один из возможных вариантов языка разработки алгоритмов, или псевдокода. Служебные слова *нач*, *кон*, *кесли, кцикл, квыбор* интерпретируются следующим образом: *нач = начало*, *кон = конец*, *кесли = конец-если, кцикл = конец-цикла*, *квыбор= конец оператора выбора.*

8.3.2. Внешняя спецификация алгоритма BuildMatr

Входные данные:

Lingb – лингвистический базис (см. параграф 6.8);

T – текст из языка *Linp(G, Lingb)* , где *G* – бесконтекстная грамматика вида (8.2.1).

Основные выходные данные:

nt – целое, количество единиц текста;

Rc – классифицирующее представление входного текста T (см. подраздел 7.1.1);

Rm – морфологическое представление входного текста (см. подраздел 7.1.1);

kindtext – строковая переменная, значение которой позволяет отнести входной текст к одному из подклассов текстов;

Arls – двумерный массив – проекция лексико-семантического словаря **Lsdic** на входной текст **T** (см. подраздел 7.1.2);

Arvfr – двумерный массив – проекция словаря глагольно-предложных фреймов **Vfr** на входной текст **T** (см. подраздел 7.1.2);

Arfrp – двумерный массив - проекция словаря предложных семантико-синтаксических фреймов **Frp** на входной текст **T** (см. подраздел 7.1.2);

Matr - матричное семантико-синтаксическое представление (МССП) входного текста (см. параграф 7.2).

8.3.3. Разработка плана алгоритма BuildMatr

План алгоритма BuildMatr

Нач Построение-компон-морфол-представления (T, Rc, nt, Rm)
Построение-проекции-лексико-семантич-словаря (Lsdic, nt, Rc, Rm, Arls)
Построение-проекции-словаря-глаг-фреймов (Arls, Vfr, nt, Rc, Rm, Arvfr)
Постр-проекции- словаря-предложных-фреймов (Arls, Frp, nt, Rc, Rm, Arfrp)
Формирование-начальных-значений-данных
Выявление-вида-текста (nt, Rc, Rm, leftprep, mainpos, kindtext, pos)
Цикл-до pos := pos + 1
Class := Rc[pos, tclass]
выбор class **из**

предлог: Обработка-предлога

прилаг: Обработка-прилаг

колич-числит: Обработка-колич-числит

сущ: Обработка-сущ

местом: Обработка-местоим

наречие: Обработка- наречия

глаг, прич: Обработка-глаг-формы

союз: Пустой оператор

констр: Обработка-конструкта

имя: Обработка-названий

маркер: если Rc[pos, unit] = ',' {запятая }

то Обработка-запятой

кесли

квыбор

выход-при *(pos = nt)*

конец

Алгоритм “Формирование-начальных-значений-данных”

Нач Обнулить все целочисленные переменные.

Присвоить значение nil (пустой элемент) всем строковым переменным.

Обнулить все числовые столбцы и заполнить цепочкой nil все строковые столбцы используемых одномерных и двумерных массивов.

Для каждой такой строки с номером k классифицирующего представления Rc, что эта строка соответствует словоформе из входного текста,

Matr[k, locunit] := наименьший номер строки из массива Arls (проекции лексико-семантического словаря Lsdic на входной текст) с информацией, соответствующей данной словоформе;

Matr[k, nval] := количество строк из Arls, соответствующих этой словоформе

кон

Последующие параграфы данной главы посвящены детализации этого плана, т.е. разработке первой части алгоритма семантико-синтаксического анализа текстов из представляющих практический интерес подъязыков естественного (русского) языка.

8.4. Описание алгоритма выявления вида входного текста

8.4.1. Назначение алгоритма

Алгоритм “Выявление-вида-текста” предназначен для отнесения текста к определенному классу; вид класса является значением выходной переменной kindtext. Спектр значений, которые может принимать переменная kindtext, представлен в следующей таблице:

Высказывания (сообщения)	АО “Парус” с 1999 года экспортирует продукцию в Болгарию.	stat
Команды	Отправить контейнеры фабрике “Заря”до 16:30.	imp
Частноутвердительные (общие) вопросы	Экспортирует ли продукцию в Болгарию АО “Парус”?	genqs
Вопросы о количестве предметов	Сколько статей по органической химии опубликовано профессором Игорем Сомовым в прошлом году?	specqs-quant1
Вопросы о количестве событий	Сколько раз в этом году запрашивался учебник Коробова?	specqs-quant2
Ролевые вопросы	Откуда поступили трехтонные контейнеры?	specqs-rol
Вопросы с вопросительно-относительным местоимением “какой” в единственном числе	В каком институте работает профессор Игорь Павлович Сомов?	specqs-relat1
Вопросы с вопросительно-относительным местоимением “какой” во множественном числе	Из каких стран импортирует комплектующие АО “Радуга”?	specqs-relat2

Табл. 8.1. Примеры входных текстов разных видов

Значение переменной kindtext используется в алгоритме построения семантического представления (СП) текста по его матричному семантико-синтаксическому представлению (параграф 5.10). Например, пусть В1 = “Откуда поступили трехтонные контейнеры?”, В2 = “Сколько раз в этом году запрашивался учебник Коробова?”. Для вопроса В1 kindtext = specqs-rol, и СП вопроса В1 может являться К-формулой *Вопрос (x1, (Ситуация(e1, поступление2 * (Место1, x1)(Время, x2)(Объект1, нек множество *(Качеств-состав, контейнер * (Вес, 3/ тонна)) : S1)) ∧ Раньше (x1, #сейчас#)))*.

В случае вопроса В2 kindtext = specqs-quant2, и СП вопроса В2 может являться К-формулой

*Вопрос (x1, (x1 ≡ Колич-элем(все запрос1 *(Время, текущий-год) (Предмет-запроса,*

*нек учебник * (Автор, нек человек *(Фамилия, “Коробов”) : x2)))))* .

Кроме переменной kindtext, к выходным данным алгоритма “Выявление-вида-текста” относятся переменные mainpos и pos. Значением переменной mainpos является номер позиции, занимаемой в тексте вопросительным словом. Например, для вопросов В3 = “Из каких стран импортирует комплектующие АО “Радуга”? и В1 = “ Откуда поступили трехтонные контейнеры?” переменная mainpos принимает соответственно значения 2 и 1. Для команд, сообщений и вопросов с ответом “Да/Нет” переменная mainpos принимает значение 1.

8.4.2. Внешняя спецификация алгоритма “Выявление-вида-текста”

Вход: Lingb – лингвистический базис, Rc и Rm – классифицирующее и морфологическое представления входного текста T (см. подраздел 7.1.1.).

Выход: kindtext, leftprep – строковые переменные для обозначения соответственно вида входного текста и предлога в начале текста; mainpos, pos - целочисленные переменные для обозначения соответственно позиции вопросительного слова и позиции, после которой следует продолжить обработку текста.

Внешние спецификации вспомогательных алгоритмов

Спецификация алгоритма-функции “Число”

Вход: d – элемент морфологического пространства Spmorph(Tform(Lingb)), соответствующий словоформе, которая может иметь число.

Выход: number1 – значение 1 для единственного числа, 2 для множественного числа, 3 в тех случаях, когда словоформа может быть отнесена как к единственному, так и множественному числу (пример: “реки”).

Спецификация алгоритма “Форма-глагола”

Вход: d – элемент морфологического пространства Spmorph(Tform(Lingb)), соответствующий глаголу.

Выход: form1– строка со значением ‘неопр’, если d соответствует глаголу в неопределенной форме; значением ‘изъявит’ для представления глагола в изъявительном наклонении (форме, с которой связано грамматическое время); значением 'повелит’, если d соответствует глаголу в повелительном наклонении.

5.4.3. Алгоритм “Выявление-вида-текста”

```
Нач      leftprep := nil, kindtext := nil
     Если     kindtext = nil
     То       log1 := (Rc[1,subclass] = вопр-относ-местоим);
          log2:=((Rc[1,tclass] = предлог) И (Rc[2,subclass] = вопр-относ-местоим));
               если (log1 = Ист) то mainpos := 1 кесли;
               если (log2 = Ист) то mainpos := 2 кесли;
{(log1 = Ист) – признак вопроса Класса 1А. Пример:
Какие препараты фирмы  GlaxoWelcome выпускаются в Польше?}
{(log2 = Ист) – признак вопроса Класса 1Б. Пример:
“Из каких стран импортирует комплектующие АО “Радуга”?}
               если (log1 = Ист) или (log2 = Ист)
               то нач m1 := Rc[mainpos,mcoord]; baseform:= Rm[base, m1];
                    number1:= Число (Rm[m1, morph]);
                 если baseform = ‘какой‘
                 то если number1= 1 то kindtext := specqs-relat1
```

иначе kindtext := specqs-relat2 кесли

если (log2 = Ист) то leftprep := Rc[1,unit] кесли;

если number1= 1 то var1 := ‘x1’ {признак вопроса о единственном объекте} иначе var1:= ‘S1’{признак вопроса о множестве объектов} кесли; pos := mainpos кесли кон кесли;

Если kindtext = nil

То log3 := ((Rc[1,subclass] = вопр-относ-местоим) ИЛИ ((Rc[1,subclass] = местоим-наречие) И (Rc[1, unit] – одно из слов (местоименных наречий)”когда”, “куда”, “откуда”, “где”)));

log 4 := ((Rc[1,tclass] = предлог) И (Rc[2,subclass] = вопр-относ-местоим));

если (log3 = Ист) ИЛИ (log4 = Ист)

то если (log3 = Ист) то mainpos := 1 кесли;

если (log4 = Ист) то mainpos := 2 кесли;

если Rc[mainpos, subclass] = вопр-относ-местоим

то m1 := Rc[mainpos,mcoord]; baseform:= Rm[base, m1];

если baseform не является лексемой ‘какой‘

то kindtext := specqs-rol;

если (log4 = Ист) то leftprep := Rc[1,unit] кесли

var1 := ‘x1’; pos := mainpos – 1;

{(log3 = Ист) – признак вопроса Класса 2А. Пример:

“Откуда поступили трехтонные контейнеры?”}

{(log4 = Ист) – признак вопроса Класса 2Б. Пример:

“Для кого предназначены контейнеры с керамикой”?} кесли кесли кесли

Если kindtext = nil

То если ((Rc[1,tclass] =глаг) И (Rc[2,unit] = “ли”))

то kindtext := genqs кесли

{ Признак вопроса Класса 3 (вопроса с ответом “Да”/”Нет”).

Пример: “Экспортирует ли продукцию в Болгарию АО “Парус”?}

Если kindtext = nil

То log5 := (Rc[1,unit] = ‘сколько’); log6 := ((Rc[2,tclass] = сущ) ;

log7 := (Rc[2,unit] = ‘раз’);

если ((log5 = Ист) И (log6 = Ист))

то var1 := ‘S1’ {признак вопроса о множестве объектов}

если (log7 = Ложь) то kindtext := specqs-quant1, mainpos := 1

{признак вопроса класса 4. Пример: “Сколько статей по органической химии опубликовано профессором И.П. Сомовым в прошлом году?”}

иначе {т.е. в случае log7 = Ист}

kindtext := specqs-quant2, mainpos := 2 кесли

{признак вопроса класса 5. Пример: “Сколько раз в этом году запрашивался роман Сергея Коробова?”} pos := mainpos кесли

Если kindtext = nil

То log8 := (Rc[1, tclass] = ‘глаг’);

если (log8 = Ист)

то нач pos := mainpos; m1 := Rc[mainpos,mcoord];

form1:= Форма-глаг (Rm[m1, morph]);

если ((form1 = неопр) ИЛИ (form1 = повелит))

то kindtext := imp {признак команды. Пример : “Отправить контейнеры фабрике “Заря”до 16:30”} кесли кон кесли кесли

Если (kintext = nil) то kindtext := stat {признак сообщения. Пример :

“ АО “Парус” с 1999 года экспортирует продукцию в Болгарию. ”} кесли конец

8.5. Принципы обработки ролевых вопросительных словосочетаний

Условимся говорить, что предложение начинается с вопросительного слова wd, если wd является первым словом данного предложения или первым словом предложения является некоторый предлог, за которым следует wd. Например, будем считать, что каждый из вопросов В1=”Кого командировали в Пензу?” и В2=”Для кого поступили три двухтонных контейнера?” начинается с вопросительного слова “кого”.

Разобьем на две группы все вопросительные слова, с которых могут начинаться рассматриваемые вопросы. К первой группе относятся вопросительно-относительные местоимения “какой”, “какие” и их формы, соответствующие различным грамматическим падежам, а также слово “сколько”. Во вторую группу включим вопросительно-относительные

местоимения “кто”, “кому”, “кого”, “чем” и т.п., а также местоименные наречия “где”, “когда”, “куда”, “откуда”. Каждое слово из этой группы вместе с определенным предлогом предназначены для выражения определенной тематической роли, т.е. смыслового отношения между значением глагола и значением выражения, зависящим в предложении от данного глагола. Если вопросительное слово wd не требует предлога для выражения определенной тематической роли, то будем говорить, что это слово используется вместе с пустым предлогом nil. Например, пара (nil, кого) в вопросе В1 используется для выражения тематической роли “Объект действия”, а пара (для, кого) в предложении В2 позволяет выразить тематическую роль “Адресат”. Учитывая сказанное, слова из второй группы будем называть *ролевыми вопросительными словами.* Начальная обработка вопросительных местоимений из первой группы осуществляется алгоритмом “Выявление-вида-текста”. Позиция вопросительного слова является значением выходной переменной mainpos. Значение переменной kindtext указывает на подкласс, к которому относится данное вопросительное слово.

Ролевые вопросительные слова (т.е. слова из второй группы) могут вместе с предлогами образовывать последовательности, являющиеся левыми сегментами вопросов. Примером может послужить вопрос В3 = “Когда, для кого откуда поступили три алюминиевых контейнера?”. В связи с этим необходим специальный алгоритм “Обработка-ролевых-вопрос-слов”. Этот алгоритм использует множество наборов Rqs, являющееся частью лингвистического базиса (ЛБД). Наборы из Rqs имеют следующую структуру:

nb (номер)	prep (предлог)	qswd (вопросительное слово)	relq (тематическая роль)
1	nil	откуда	Место1
2	для	кого	Адресат

Табл. 8.2. Структура набора из множества Rqs

Алгоритм “Обработка-ролевых-вопрос-слов” вызывается для обработки ролевых вопросительных слов, располагающихся в начале многих вопросов. Такие слова являются вопросительно-относительными местоимениями (“кто”, “кому”, “чем” и т.д.) или местоименными наречиями (“когда”, “куда”, “откуда”).

Внешняя спецификация алгоритма “Обработка-ролевых-вопрос-слов”

Вход: nt – цел - количество единиц текста; **Rc** – классифицирующее представление входного текста T (см. параграф 7.1); **pos** – цел – позиция вопросительного слова в Rc**; Rqs** – словарь вопросительных словосочетаний (одна из составляющих Lingb);
Matr - матричное семантико-синтаксическое представление (МССП) входного текста (см. параграф 7.2); **leftprep** - строка – значение предлога слева; **numbent** – цел – количество объектов, упомянутых в тексте; **numbqswd** – цел – количество уже найденных вопросительных слов в тексте; **posqswd** – одномерный массив длины nt, где для k≥ 1 posqswd [k] – либо позиция в Rc k-го вопросительного слова, либо 0.
Выход: Matr, numbent, numbqswd, posqswd, leftprep.

Алгоритм “Обработка-ролевых-вопрос-слов”

нач {Условие вызова:(subclass = вопр-относ-местоим) ИЛИ (subclass = местоим-наречие)}

numbqswd:=numbqswd+1; { количество вопросительных слов в тексте } posqswd[numbqswd]:=pos; word1:=Rc[pos,unit]; {запоминается позиция вопросительного слова для последующего связывания с глаголом}

если (subclass = вопр-относ-местоим)

то Найти в множестве Rqs - одной из составляющих Lingb – набор с таким порядковым номером k1, что

Rqs[k1,prep] = leftprep, Rqs[k1, qswd]:= word1

иначе Найти в множестве Rqs набор с таким порядковым номером k1, что Rqs [k1, qswd] = word1 кесли

Затем role:=Rqs[k1, relq], Matr[pos, reldir, 1] := role

{ для кого=> leftprep=’для’, word1=’кого’, role:=’Адресат’}

{ кто=> leftprep=nil, word1=’кто’, role:=’Агент’}

numbent := numbent+1; var1:= var(‘x’, numbent);

Matr[pos, mark] := var1; leftprep:=nil кесли кон

8.6. Принципы и методы обработки причастных оборотов и придаточных определительных предложений

8.6.1. Принципы обработки

Каждому причастному обороту и придаточному определительному предложению ставится в соответствие определенный номер уровня вложения в главное предложение; этот номер является значением переменной depth (“глубина” в переводе с английского). Пусть pos – целочисленная переменная, значение которой является порядковым номером единицы текста, анализируемой в данный момент вычисления. Тогда, если pos указывает какую-либо текстовую единицу в главном предложении, то depth = 1. При переходе из главного предложения в причастный оборот или придаточное определительное предложение уровень глубины увеличивается на единицу. Увеличение уровня глубины на единицу происходит и в том случае, когда осуществляется переход из фрагмента текста с уровнем глубины depth > 1 в причастный оборот или придаточное определительное предложение.

Позиция глагольной формы (глагола или причастия) во фрагменте с глубиной *depth* анализируемого предложения задается элементом verbposmag[depth] анализируемого массива verbposmag длины 4. Размер массива определяется предположением о том, что в реальных фразах максимальный уровень глубины

depth равен 4, причем даже уровень глубины 3 достигается редко. Перед началом анализа фразы элементы массива verbposmag обнуляются.

Пример. Пусть T1 = “Профессор Игорь Сергеевич Сомов, о котором пишет газета “Поиск” в номере, поступившем в субботу, работает в МИФИ.”, E1 – фрагмент “Профессор Игорь Сергеевич Сомов,”, E2 – фрагмент “о котором пишет газета “Поиск” в номере,”, E3 – фрагмент “поступившем в субботу,“, E4 – фрагмент “работает в МИФИ.”. Тогда, если переменная *pos* указывает какую-либо текстовую единицу во фрагментах E1, E2, E3, E4, то переменная depth принимает соответственно значения 1, 2, 3, 1.

Если pos = 4 (позиция слова “Сомов”), то depth = 1 и verbposmag[depth] = 0, поскольку глагол из главного предложения еще не найден. Пусть pos = 9 (позиция слова “газета”), тогда depth = 2 и verbposmag[depth] = 8 (позиция глагола “пишет”), при этом в позициях 1, 3, 4 массива verbposmag расположен 0.

Матричное семантико-синтаксическое представление (МССП) Matr и вспомогательный целочисленный массив posconnectword используются для отображения информации о взаимосвязях главного предложения и либо причастного оборота, либо придаточного определительного предложения. Количество строк в Matr и длина массива posconnectword равны nt - количеству строк в классифицируюшем представлении Rc входного текста, т.е. количеству элементарных значащих единиц текста.

Напомним, что если pos – позиция причастия, с которого начинается причастный оборот, и это причастие “прикреплено” к существительному в позиции m, то Matr[pos, contr] = m.

Пример. Пусть T2 = “Сколько контейнеров с индийской керамикой, поступивших из Новороссийска, были отправлены АО “Радуга”?”. Рассмотрим размеченное представление текста T2 (проставим после каждой элементарной значащей единицы текста ее номер): “Сколько (1) контейнеров (2) с (3) индийской (4) керамикой (5) , (6) поступивших (7) из (8) Новороссийска (9) , (10) были отправлены (11) АО (12) “Радуга” (13) ? (14)”. Тогда Matr[7, contr] = 2, где 7 и 2 – позиции слов “поступивших” и “контейнеров”.

Ненулевые элементы массива posconnectword предназначены для установления взаимосвязи между позицией pos глагола в придаточном определительном предложении и союзным словом, “прикрепляющим” это придаточное предложение к главному.

Пример. Пусть Т3 – размеченный текст “Сколько (1) контейнеров (2) с (3) индийской (4) керамикой (5) , (6) которые (7) в (8) пятницу (9) поступили (10) из (11) Новороссийска (12) , (13) были отправлены (14) АО (15) “Радуга” (16) ? (17)”. Тогда posconnectrword[10] = 7, где 10 и 7- позиции слов “поступили” и “которые”. В остальных позициях массива posconnectrword расположен 0.

Если pos – позиция глагола в придаточном определительном предложении с уровнем глубины depth > 1, k - позиция союзного слова в том же придаточном предложении, m – позиция существительного, на которое дается ссылка союзным словом, то Matr[pos, contr] = Matr[k, contr] = m.

Пример. Рассмотрим размеченное представление текста Т1:

“Профессор (1) Игорь (2) Сергеевич (3) Сомов (4) , (5) о (6) котором (7) пишет (8) газета (9) “Поиск” (10) в (11) номере (12) , (13) поступившем (14) в (15) субботу (16) , (17) работает (18) в (19) МИФИ(20) . (21)”

Тогда *Matr[8, contr] = Matr[7, contr] =1* (позиция слова “профессор”) ,

posconnectword [8] = 7 (позиция слова “ котором ”) ,

Matr[14, contr] =12 (позиция слова “ номере ”) .

Назначение двумерного целочисленного массива pos-free-dep[1: 4, 1: nt] с количеством строк 4 и количеством строк nt (равным количеству значащих строк Rc - классифицирующего представления входного текста) заключается в следующем. Пусть $1 \leq \text{depth} \leq 4$. Тогда строка с номером depth массива pos-free-dep содержит позиции таких единиц фрагмента текста с уровнем глубины depth, для которых в данный момент анализа текста еще не найдена смысловая связь с глаголом на том же уровне глубины.

Пример. Рассмотрим размеченное представление текста Т4 = “С (1) 2001-го года (2) АО (3) ”Радуга” (4) экспортирует (5) станки (6) в (7) Болгарию (8) . (9)”. Тогда в момент рассмотрения единицы ”Радуга” в позиции 4 массив pos-free-dep должен иметь следующую конфигурацию:

2	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Табл. 8.3. Структура массива pos-free-dep

Значением элемента numb-free-dep[depth] одномерного массива numb-free-dep длины 4 является количество существительных и конструктов в анализируемом фрагменте с уровнем глубины depth рассматриваемого предложения, для которых в данный момент обработки предложения еще не найдена смысловая связь с глагольной формой из того же фрагмента с уровнем глубины depth.

Пример. Для сформулированного выше вопроса Т3 после обработки словоформы “пятницу” в позиции 9 numb-free-dep[1] = 2, поскольку в главном предложении (уровень глубины 1) встретились существительные “контейнеров” и “керамикой”, и numb-free-dep[2] = 2, так как в придаточном предложении встретилось союзное слово “которые” и существительное “пятницу”.

Одномерный массив nmasters[1 : nt] предназначен для отображения количества семантических отношений между элементарной значащей единицей текста с произвольным порядковым номером k, $1 \leq k \leq nt$, и другой элементарной значащей единицей текста.

Пример. В классифицирующем представлении Rc входного текста (вопроса) Т3 слову “контейнеров” соответствует строка с номером k = 2. Между значением этого слова и значениями слов “поступили”, “ были отправлены” (порядковые номера 10 и 14 в Rc) существуют семантические отношения R1, R2. Поэтому nmasters[2] = 2.

8.6.2. Описание алгоритма “Обработка запятой”

Внешняя спецификация алгоритма “Обработка запятой”

Вход: Rc – классифицирующее представление входного текста T (см. параграф 7.1);

p – цел – позиция запятой; kindtext – строка – обозначение вида текста; depth – цел - значение уровня глубины к моменту рассмотрения запятой в позиции p; verbposmag[1 : 4], pos-free-dep[1: nt, 1:4], numb-free-dep[1 : 4] - целочисленные массивы (см. выше описание принципов использования этих массивов).

Выход: p, pos - целочисленные переменные; verbposmag[1 : 4], pos-free-dep[1: nt, 1:4], numb- free-dep[1 : 4] - целочисленные массивы.

Алгоритм «Обработка запятой»

Нач если ((kindtext = specqs-rol) И (depth = 1) И (verbposmag[1] = 0)
И ((Rc[p + 1, subclass] = вопр-относ-местоим)
ИЛИ (Rc[p + 1, subclass] = местоим-наречие)
ИЛИ ((Rc[p + 1, tclass] = class) = предлог)
И (Rc[p + 2, subclass] = вопр-относ-местоим)))
То Пустой оператор {в этом случае запятая разделяет вопросительные слова в начале вопроса}
иначе ind:=0;

Если Rc[p+1, tclass] = ’причаст’ то ind:=1 кесли
Если Rc[p+1, subclass] = ’вопр-относ-местоим’
То ind:=2 кесли
Если ((Rc[p+1, tclass] =’предлог’) И (Rc[p+2, subclass] = ’вопр-относ-местоим’)) то ind := 3 кесли

Выбор ind из
0: {Возвращение на предыдущий уровень глубины}
verbposmag[depth] : = 0; numb-free-dep[depth] := 0;
Обнулить элементы строки с номером depth массива pos-free-dep ;
Depth := depth-1 ;
1, 2, 3 : depth := depth + 1{переход на следующий уровень глубины}
Квыбор кесли конец

8.6.3. Описание алгоритма «Обработка-местоимения»

Условие вызова алгоритма: Rc[p, tclass] = ’местоим’, причем либо местоимение в позиции p входит в вопросительное ролевое словосочетание в начале вопросительного предложения, либо местоимение играет роль союзного слова, соединяющего придаточное определительное предложение с главным предложением.

Описание вспомогательного алгоритма «Поиск-существительного»

Условие вызова алгоритма: в позиции pos массива Rc располагается причастие или словоформа с лексемой «который» или «какой». При этом в главном предложении слева от союзного слова есть по крайней мере одно существительное. Слева от позиции pos ищется такое существительное, к которому в первом случае «прикреплено” причастие, а во втором рассматриваемом случае это существительное является референтом союзного слова.

Внешняя спецификация алгоритма

Вход: Rc – классифицирующее представление текста, p – цел – позиция причастия или словоформы с лексемой «который» или «какой»; массив verbposmag[1:4]; Matr – МССП текста.

Выход: poscontr – цел – позиция существительного, к которому “прикреплены” причастный оборот или придаточное определительное предложение.

Алгоритм “Поиск-существительного”

```
Нач   log:=ложь,    k1:= pos , posvb1:= verbposmag[depth-1]
      цикл-до       k1:= k1-1;
                    part1:= Rc[k1, tclass];
                    если (part1=’сущ’) и
                     ((Matr[k1, posdir, nmasters[k1]]= posvb1) ИЛИ
                     (Matr[k1, posdir, nmasters[k1]]= 0))
                    то      log:=  Ист;    poscontr := k1
      выход при (log = Ист)       кцикл
кон
```

Пример: Пусть T = Сколько контейнеров с индийской керамикой, которые в пятницу поступили из Новороссийска, предназначены для АО «Парус»?". В про-цессе разбора T слева направо найдем зависимость [контейнеров] --->[керамикой]. Так как у элемента 'контейнеров' пока не найдено управляющего слова, то этот элемент является референтом слова «который».

Внешняя спецификация алгоритма «Обработка-местоимения»

Вход: Rc и Rm– классифицирующее и морфологическое представления входного текста T (см. параграф 4.9); Rqs – словарь вопросительных словосочетаний; subclass – строка – значение подкласса местоимения; p – цел – позиция местоимения; leftprep – строка – значение предлога слева (включая пустой предлог nil); depth – цел - значение уровня глубины к моменту рассмотрения местоимения; verbposmag[1 : 4], pos-free-dep[1: nt, 1:4], numb-free-dep[1 : 4] - целочисленные массивы (см. выше описание принципов их использования).
Выход: p - целочисленная переменная; verbposmag[1 : 4], pos-free-dep[1: nt, 1:4], numb- free-dep[1 : 4] - целочисленные массивы.

Алгоритм «Обработка-местоимения»

Нач если ((subclass] = вопр-относ-местоим) И (depth = 1)
И (verbposmag[depth] = 0))
то Обработка-ролевых-вопрос-слов (Rc, Rm, Rqs, leftprep)
иначе нач если (((Rc[p – 1, unit] ≠ ',') И (Rc[p-1, tclass] ≠ предлог))
ИЛИ ((Rc[p-1, tclass] = предлог) И (Rc[p – 2, unit] ≠ ',')))
то {во входном тексте пропущена запятая перед придаточным определительным предложением} depth := depth + 1 кесли
Поиск-существительного (p, poscontr)
Matr [p, contr] := poscontr; numb-free-dep[depth] := 1;
{в придаточном определительном предложении с уровнем глубины depth найдено первое слово, для которого пока отсутствует управляющая стрелка (с меткой семантического отношения) из глагольной формы в позиции verbposmag[depth] (пока в этой позиции расположен 0)}
pos-free-dep[depth, 1] := p кон
кон

8.6.4. Описание алгоритма “Обработка наречия”

Внешняя спецификация алгоритма

Вход: Rc – классифицирующее представление входного текста T (см. параграф 7.1);

p – цел – позиция наречия; subclass – строка – значение подкласса наречия; depth – цел - значение уровня глубины к моменту рассмотрения наречия в позиции p; numbqswd – цел – количество вопросительных слов; posqswd [1 : nt] – массив для представления позиций вопросительных слов; verbposmag[1 : 4], pos-free-dep[1: nt, 1:4], numb-free-dep[1 : 4] - целочисленные массивы (см. выше описание принципов использования этих массивов).

Выход: p, numbqswd - целочисленные переменные; posqswd [1 : nt], verbposmag[1 : 4], pos-free-dep[1: nt, 1:4], numb- free-dep[1 : 4] - целочисленные массивы.

Алгоритм “Обработка наречия”

Нач если (subclass = местоим-наречие) И (depth = 1)

To leftprep := nil;

Обработка-вопрос-слов (p, Rc, Rm, Rqs, leftprep, numbqswd, posqswd, Matr)

кесли кон

8.7. Разработка алгоритма поиска возможных смысловых связей между глагольной формой и значением зависящей от нее группы слов

8.7.1. Основные идеи формализации необходимых условий существования смысловой связи между значением глагольной формы и значением зависящей от нее группы слов.

Субстантивными выражениями будем называть существительные, а также существительные с зависимыми словами, обозначающие понятия, предметы и множества предметов (сочетание noun substantive означает в английском языке имя существительное). Например, пусть T1=“Откуда и для кого поступили два алюминиевых контейнера с керамической плиткой?”, T2=“Когда поступила статья профессора А.П.Сомова?” и T3 =“Поставь синюю коробку на зеленый

ящик”. Тогда сочетания “два алюминиевых контейнера”, “статья профессора А.П.Сомова”, “синюю коробку” являются субстантивными выражениями.

Под глагольной формой будем понимать глагол в личной или неопределенной форме либо причастие. Установление возможных смысловых отношений между глагольной формой и словосочетанием, включающем существительное или вопросительно-относительное местоимение, играет важную роль в процессе осуществления семантико-синтаксического анализа ЕЯ-текста.

Будем полагать, что posvb - это позиция в представлении Rc глагольной формы, posdepword - позиция в представлении Rc существительного или вопросительно-относительного местоимения..

Входными данными алгоритма “Найти-множ-тематич-ролей” являются натуральные числа posvb, posdepword, а также двумерные массивы Arls, Arvfr, где Arls – проекция лексико-семантического словаря Lsdic на входной текст, Arvfr – проекция словаря глагольно-предложных фреймов Vfr на входной текст.

Назначение алгоритма “Найти-множ-тематич-ролей” заключается, во-первых в нахождении целого числа nrelvbdep – количества возможных смысловых отношений между значениями единиц текста с номарами p1 и p2 в представлении Rc.

Во-вторых, этот алгоритм должен строить вспомогательный двумерный массив arrelvbdep, хранящий информацию о возможных смысловых связях между единицами Rc с номерами p1 и p2. Строки этого массива представляют информацию о комбинациях значений глагольной формы и зависимой группы слов (или одного слова). Структура каждой строки представлена на Рис. 8.1.

Arrelvbdep

Linenoun	Linevb	trole	example

Рис. 8.1. Структура строки вспомогательного массива Arrelvbdep

Для k-й заполненной строки массива Arrelvbdep (k>=1) linenoun - номер строки из массива Arls, соответствующего слову в позиции p1; linevb – номер набора из массива Arls, соответствующего глагольной форме в позиции p2; trole – обозначение смыслового отношения (тематической роли), связывающего

глагольную форму в позиции p2 и зависимое слово в позиции p1; example – пример выражения на ЕЯ, в котором реализуется та же самая тематическая роль.

Поиск возможных смысловых отношений между значением глагольной формы (ГФ) и значением зависимой группы слов (ЗГС) осуществляется с помощью проекции на входной текст словаря глагольно-предложных фреймов (с.г.п.ф.) Arvfr. В этом словаре ищется такой шаблон (или шаблоны), который был бы совместим с некоторыми семантико-синтаксическими характеристиками ГФ в позиции posvb и ЗГС, имеющей номер posdepword в Rc.

К таким характеристикам, во-первых, относится множество кодов грамматических падежей Grcases, ассоциированных с текстообразующей единицей, порядковым номером, которым в Rc является величина posdepwd.

Предположим, что Rc[posvb, tclass] = глаг, Rc[posdepword, tclass] = сущ

или Rc[posdepword, subclass]= вопрос-относ-местоим.

Тогда Grcases – это множество грамматических падежей, соответствующих существительному или вопросительно-относительному местоимению в позиции posdepword.

Пример. Пусть B1=”Какие(1) лекарственные(2) препараты(3) выпускаются(4) на(5) фабрике(6) “Рассвет”(7) ?(8)” и B2=”Где(1) работает(2) профессор(3) И.П.(4) Семенов(5) ,(6) о (7) котором(8) пишет(9) газета(10) «Поиск»(11) в(12) последнем(13) номере(14) ?(15)”

Пусть для B1 posvb=4, posdepword=6 (позиция слов «выпускаются» и «фабрике»), для B2 posvb=9, posdepword=8 (позиции слов «пишет» и «котором»). Тогда в первом случае Grcases:={3,6}, т.к. словоформа «фабрике» может находиться как в дательном падеже (код 3), так и в предложном падже (код 6). Для второго случая Grcases:={6}, поскольку словоформа «котором» относится к предложному падежу.

При поиске возможных смысловых отношений в сочетаниях «Существительное+Причастие» («препарат, выпускаемый», «сотрудники, работающие») используются шаблоны из множества Arvfr, связанные со значениями глаголов, от которых образованы причастия.

Например, при поиске возможных смысловых отношений между причастием и существительным в сочетании «препарат, выпускаемый» используется

семантико-синтаксический шаблон, позволяющий найти тематическую роль (роли) в сочетании «препарат был выпущен».

Чтобы найти смысловое отношение (отношения) в сочетании «сотрудников, работающих», будем фактически искать тематическую роль (роли) в сочетании «сотрудники работают».

Учитывая сказанное выше, процесс поиска в предложении тематической роли, связывающей глагольную форму в позиции posvb и слово (существительное или относительное местоимение) в позиции posdepword, к которому относится предлог prep (возможно, пустой предлог nil) можно пояснить следующим образом:

В тройном цикле по параметрам i1, i2, k1, где i1, i2 - номера наборов строк из множества Arls, соответствующие существительному или вопросительно-относительному местоимению в позиции posdepword и глагольной форме в позиции posvb, k1 – номер набора из проекции словаря глагольно-предложных фреймов Arvfr, ищется такое сочетание значений параметров i1, i2, k1, что выполняются следующие условия:

Если sem1 - значение поля sem для набора с номером i1 из Arls, sem2 – значение поля sem для набора с номером i2 из Arls, и *semsit1, trole1, sprep1, grc1* – значения полей semsit, trole1, sprep, grcase набора с номером k1 из Arvfr, такие, что

Semsit1=sem2, sprep1=prep, grc1 ∈ Grcases, то выполняется соотношение (см. параграф 4.6)

(T, posdepword, sem1, prep, grc1, posvb, sem2, relat1) ∈ Смысл-связь1.

8.7.2. Описание алгоритма поиска возможных смысловых связей между глагольной формой и субстантивным выражением

Назначение алгоритма “Найти-множ-отнош-глаг-сущ”

Установить тематическую роль, связывающую глагольную форму в позиции **posvb** и слова (существительного или союзного слова) в позиции **posdepword** с учетом возможного предлога перед этим словом. Как следствие, выбрать одно из нескольких возможных значений глагольной формы и одно из нескольких возможных значений слова в позиции **posdepword**. Для этого потребуются три

вложенных цикла: (1) по возможным значениям слова в позиции **posdepword**, (2) по возможным значениям глагольной формы; (3) по глагольно-предложным фреймам, связанным с данной глагольной формой.

Внешняя спецификация алгоритма алгоритма “Найти-множ-отнош-глаг-сущ”

Вход **Rc** - классифицирующее представление, **nt** – цел - количество единиц текста в классифицирующем представлении Rc, т.е. количество строк в Rc,
Rm – морфологическое представление лексических единиц, входящих в Rc,
posvb – цел – позиция глагольной формы (глагола в личной или неопределенной форме, причастия), **posdepword** – цел – позиция существительного или вопросительно-относительного местоимения (“котором” в сочетании “о котором”, являющемся началом придаточного определительного предложения и т.д.), **depth** – цел - значение уровня глубины вложенности для слова в позиции **posdepword** ,
Matr – начальное значение МССП текста; **Arls** – массив – проекция лексико-семантического словаря (ЛСС) **Lsdic** на входной текст **T**; **Arvfr** – массив – проекция словаря глагольно-предложных фреймов **Vfr** на входной текст **T**.
Выход

arrelvbdep – одномерный массив, предназначенный для представления информации о значении зависимого слова, значении глагольной формы и о смысловом отношении между глагольной формой в позиции posvb и зависимым словом в позиции posdepword;
nrelvbdep – цел - количество значащих строк в массиве arrelvbdep.

Внешние спецификации вспомогательных алгоритмов

Спецификация алгоритма “Признаки-глаг-формы”

Вход: p1 – номер строки из Rc, соответствующей глаголу или причастию.
Выход:.form1, refl1, voice1 – строки, значения которых определяются следующим образом. Если p1 – позиция глагола, то form1 может иметь одно из следующих значений: *изъявит* (признак изъявительного наклонения), *неопр* (признак неопределенной формы глагола), *повелит* (признак повелительного

наклонения). Если p1 – позиция причастия, то form1 := *изъявит.* Строка refl1 получает значение *действит* (признак действительного залога) или *страд* (признак страдательного залога). Значения параметров form1, refl1, voice1 вычисляются по набору числовых кодов значений морфологических признаков, связанных с текстовой единицей с порядковым номером p1.

Спецификация алгоритма “Спектр-сорта”

Вход: z - сорт, т.е. элемент множества St(B(Cb(Lingb))), где Lingb – лингвистический базис.

Выход: spectrum - множество всех сортов, являюшихся обобщениями сорта z, включая сорт z..

Алгоритм “Найти-множ-отнош-глаг-сущ”

Нач Признаки-глаг-формы (posvb, form1, refl1, voice1)

nrelvbdep := 0

{Далее вычисляется предлог}

если ((Rc[posvb, tclass] = прич) И (posdepword = Matr[posvb, contr])

то prep := nil иначе prep := leftprep кесли

{ Вычисление posn1 – позиции существительного, которое определяет множество сортов для текстовой единицы в позиции posdepword }

если (Rc[posdepword, subclass] = вопрос-относ-местоим)

и (Rc[posdepword, unit] – слово с лексемой “который” или “какой»)

то posn1 := Matr[posdepword, contr]

иначе posn1 := posdepword кесли

{ Далее вычисляется множество грамматических падежей Grcases, которое будет связано со словом в позиции posdepword для нахожднгтя множества смысловых отношений между словами в позициях posvb и posdepword }

t1 := Rc[posvb, tclass]; t2 := Rc[posvb, subclass];

если t1 = прич то если posdepword = Matr[posvb, contr]

то Grcases := {1} кесли

иначе {в случае posdepword ≠ poscontrword [posvb]} переход к L1 кесли

иначе {т.е. в случае t1 = глаг} переход к L1 кесли

L1: p1 := Rc[posdepword, mcoord];

Grcases := Падежи (Rm[p1, morph])

кесли

line1 := Matr[posn1, locunit]; numb1 := Matr[posn1, nval]

{количество строк в Arls со значениями существительного}

цикл для i1 от line1 до line1 + numb1 – 1 {цикл по строкам массива Arls, соответствующих существительному в позиции posn1}

Set1 := пустое множество

цикл для j от 1 до m {m – семантическая размерность сортовой системы S(B(Cb(Lingb))), т.е. наибольшее количество несравнимых сортов, которые могут характеризовать одну сущность}

current-sort := Arls[i1, st_j];

если current-sort ≠ nil

то Спектр-сорта (current-sort, spectrum);

Set1 := Объединение множеств Set1 и spectrum кесли

{для произвольного сорта z значением spectrum является множество всех сортов, являюшихся обобщениями сорта z, включая сорт z} кцикл {по j}

{Далее следует цикл по значениям глагольной формы}

line2 := Matr[posvb,locunit]

numb2 := Matr[posvb,nval]

{количество строк в Arls со значениями глагольной формы}

цикл для i2 от line2 до line2 + numb2 - 1

{цикл по строкам массива Arls соотв. глаг. в позиции posvb}

current-pred := Arls[i2, sem]

<u>цикл</u> для k1=1 до narvfr

<u>если</u> Arvfr[k1, semsit] = current-pred

<u>то</u> <u>нач</u>

s1 := Arvfr[k1, str]

<u>если</u> ((prep=Arvfr[k1, sprep] и (s1∈ Set1) и (form1 =Arvfr[k1, form]) и и (refl1 =Arvfr[k1, refl]) и (voice1 =Arvfr[k1, voice]))

<u>то</u> grc := arvfr[k1, grcase]

<u>если</u> (grc ∈ Grcases)

<u>то</u> {отношение существует}

nrelvbdep:=nrelvbdep+1; arrelvbdep[nrelvbdep, linevb] := i2 ;

arrelvbdep[nrelvbdep, linenoun] := i1 ; arrelvbdep[nrelvbdep, gr] := grc ;

arrelvbdep[nrelvbdep+1, role] := arvfr[k1, trole]

кесли

кесли

конец

кесли

кесли

кесли

конец

Комментарий к алгоритму "Найти-множ-отнош-глаг-сущ"

Найдено количество nrelvbdep смысловых отношений между глагольной формой и зависимым от нее существительным. Рассматривается такой подъязык русского языка, что в вопросах всегда после глагола находится хотя бы одно существительное. Информация о таких комбинациях значений глагола V и существительного N1, которое даёт хотя бы одно смысловое отношение между V и N1, отображена во вспомогательном массиве arrelvbdep:

linevb	linenoun	role	example
c1	c2		Поступила цистерна
…			

Рис. 8.2. Структура строки вспомогательного массива arrelvbdep.

В столбце linevb помещается c1 – номер строки массива Arls, для которой Arls[c1, numb] = posvb, т.е. строка c1 указывает какое-то одно значение глагола V в позиции posvb. Например, для B1 = "Откуда и для кого поступили три алюминиевых контейнера с керамикой?" в столбце linevb ставится c1 – номер строки массива Arls, такой, что Arls[c1,sem] = поступление2.

В столбце linenoun ставится c2 – номер строки массива Arls, такой что Arls[c2, numb] = posn1 (позиция существительного n1). Например, для B1 Arls[c2,

sem]=контейнер. Солбец role предназначен для отображения возможных отношений между глаголом V и существительным N1.

Если nrelvbdep = 0, то не найдено смысловых отношений. Будем предполагать, что это невозможно для рассматриваемого входного языка. Если nrelvbdep = 1, то однозначно определены значение глагола V (по строке c1), значение существительного N1 (по строке c2) и значение смыслового отношения arrelvbdep [nrelvbdep, role]. Например, для вопроса B1 выполняются соотношения V=”посту-пили”, N1=”контейнера”, nrelvbdep = 1, arrelvbdep [nrelvbdep, role] = Объект1.

Если nrelvbdep > 1 то необходимо вызвать процедуру, которая задает уточняющие вопросы пользователю, и сформировать эти вопросы на основе примеров в столбце example.

8.7.3. Описание алгоритма обработки конструктов

Назначение алгоритма “Найти-множ-отнош-глаг-конструкт”: установить тематическую роль, связывающую глагольную форму в позиции **posvb** и конструкт в позиции **posdep** с учетом возможного предлога перед этим конструктом. Как следствие, выбрать одно из нескольких возможных значений глагольной формы. Для этого потребуются два вложенных цикла: (1) по возможным значениям глагольной формы; (2) по глагольно-предложным фреймам, связанным с данной глагольной формой.

Внешняя спецификация алгоритма

Вход: posvb – цел – позиция глагольной формы (глагола в личной или неопределенной форме, причастия), posdep (сокращение от “position of dependent word”) – цел – позиция конструкта (выражения, обозначающего числовое значение параметра), subclass1 – строка – обозначение сорта конструкта, Matr – МССП текста; Arls – проекция лексико-семантического словаря на входной текст; Arvfr - проекция словаря глагольно-предложных фреймов на входной текст;

prep1 – строка – предлог, относящийся к конструкту, или пустой предлог nil.

Выход:

arrelvbdep – двумерный массив, предназначенный для представления информации о значении глагольной формы и смысловом отношении между глагольной формой в позиции posvb и конструктом в позиции posdep;

nrelvbdep – цел – количество значащих строк в массиве arrelvbdep.

Алгоритм “Найти-множ-отнош-глаг-конструкт”

```
Нач     startline  := Matr[posvb, locunit] – 1 ;
        Line1 := startline;    numbvalvb := Matr[posvb, nval]
{количество возможных значений глагольной формы}
цикл-до
        Line1:= Line1 + 1;   Current-pred := Arls[line1, sem]
        K1:=0; log1:=false
        Цикл-до
                k1:=k1+1
                если (Arvfr[k1, semsit] =current-pred)
                то     если (Arvfr[k1, str] = subclass1) и (Arvfr[k1, sprep] = prep1)
                         то {отношение существует}
                     nrelvbdep := 1; arrelvbdep[1, linevb] := line1
                arrelvbdep[1, role] := Arvfr[k1, trole];   Log1:=true
        Выход-при (log1=true)      Кцикл
Выход-при ((log1=true) или (line1 = startline + numbvalvb))          кцикл
          leftprep := nil       кон
```

8.7.4. Описание алгоритма “Найти-множ-тематических-ролей”

Назначение алгоритма: установить множество тематических ролей, связывающих глагольную форму в позиции **posvb** и слово (существительного, союзного слова, конструкт) в позиции **posdep** с учетом возможного предлога перед этим словом. Для этого потребуются три вложенных цикла: (1) по возможным значениям слова в позиции **posdep**, (2) по возможным значениям глагольной формы; (3) по глагольно-предложным фреймам, связанным с данной глагольной формой.

Внешняя спецификация

Вход

Rc - классифицирующее представление текста, nt – цел - количество единиц текста в Rc, т.е. количество строк в Rc, Rm – морфологическое представление лексических единиц, входящих в Rc, posvb – цел – позиция глагольной формы (глагола в личной или неопределенной форме, причастия),

posdep (сокращение от “position of dependent word”) – цел – позиция зависимого слова (существительного, вопросительно-относительного местоимения, конструкта - выражения, обозначающего числовое значение параметра),

Matr – строково-числовая матрица – исходное МССП текста,

depth – цел - значение уровня глубины вложенности для слова в позиции **posdep**,

Arls – массив – проекция лексико-семантического словаря (ЛСС) Lsdic на входной текст T; Arvfr – массив – проекция словаря глагольно-предложных фреймов Vfr на входной текст T,

nmasters [1:nt] – массив для отображения количества управляющих слов для каждой единицы текста.

Выход:

class1 – строка – обозначение класса текстовой единицы в позиции posdep,

subclass1 - обозначение подкласса текстовой единицы в позиции posdep,

arrelvbdep – двумерный массив, предназначенный для представления информации о значении зависимого слова, значении глагольной формы и о смысловом отношении между глагольной формой в позиции posvb и зависимым словом в позиции posdep,

nrelvbdep – цел - количество значащих элементов в массиве arrelvbdep.

Алгоритм “Найти-множ-тематических-ролей”

Нач Заполнить числом 0 все числовые позиции массива arrelvbdep и заполнить пустым элементом *nil* все строковые позиции массива arrelvbdep

class1 := Rc[posdep,tclass]

subclass1 := Rc[posdep, subclass]; prep1 := Matr [posdep, prep]

если (class1 = сущ) ИЛИ ((class1 = местоим) И (subclass1 = вопросит-относит-местоим))

то Найти-множ-отнош-глаг-сущ (Rc, Rm, posvb, posdep, prep1, depth, Arls, Arvfr, Matr, nmasters, nrelvbdep, arrelvbdep) **кесли**

если (class1 = констр) **то** Найти-множ-отнош-глаг-конструкт (posvb, posdep, prep1, subclass1, depth, Arls, Arvfr, Matr, nmasters, nrelvbdep, arrelvbdep) **кесли конец**

8.7.5. Описание алгоритма поиска смысловой связи между глагольной формой и зависимым выражением

Назначение алгоритма "Смысл-связь-глаг-формы": установить тематическую роль, связывающую глагольную форму в позиции **posvb** и выражение (существительное, союзное слово, конструкт) в позиции **posdep** с учетом возможного предлога перед этим выражением. Как следствие, выбрать одно из нескольких возможных значений глагольной формы и одно из нескольких возможных значений слова в позиции **posdep**.

Занести полученную информацию о значении глагольной формы, значении зависимой единицы текста и о смысловом отношении (т.е. о тематической роли) в МССП Matr.

Внешняя спецификация алгоритма

Вход: **Rc** - классифицирующее представление, **nt** – цел - количество единиц текста в Rc, т.е. количество строк в Rc,

Rm – морфологическое представление лексических единиц, входящих в Rc,

posvb – цел – позиция глагольной формы (глагола в личной или неопределенной форме, причастия), **posdep** – цел – позиция существительного или относительного местоимения ("котором" в сочетании "о котором", являющемся началом придаточного определительного предложения и т.д.)

depth – цел - значение уровня глубины вложенности для слова в позиции **posdep**,

Arls – массив – проекция лексико-семантического словаря (ЛСС) **Lsdic** на входной текст **T**; **Arvfr** – массив – проекция словаря глагольно-предложных фреймов **Vfr** на входной текст **T**;

Matr – начальное значение МССП текста; **nmasters [1:nt]** – массив для отображения количества управляющих слов для каждой единицы текста.

Выход: **Matr –** строково-числовая матрица – преобразованное значение исходной матрицы **Matr**.

Внешняя спецификация алгоритма “Выбор-тематич-роли”

Вход: posvb –цел – позиция глагольной формы; posdep - цел – позиция зависимой единицы текста (существительного или конструкта); arrelvbdep – двумерный массив, представляющий информацию о возможных комбинациях значения глагольной формы в позиции posvb, значения зависимой единицы в позиции posdep и тематической роли rel, реализующейся в таком сочетании (см. описание массива arrelvbdep в подпараграфе 5.7.2); nrelvbdep – цел – количество значащих строк в массиве arrelvbdep, т.е. количество возможных смысловых отношений между рассматриваемыми глагольной формой и зависимой единицей.

Выход: m1 **–** цел – номер некоторой значащей строки массива arrelvbdep. Параметр m1 приобретает ненулевое значение в результате обработки ответа пользователя на уточняющий вопрос ЛП. Пользователю предлагается указать, какое из нескольких смысловых отношений реализуется в сочетании “Глагольная форма в позиции posvb + Зависимая единица в позиции posdep”. Для этого пользователю с помощью столбца example даются примеры сочетаний, в которых реализуется такое же смысловое отношение, как и потенциально возможное отношение между единицами текста в позициях posvb и posdep.

Алгоритм “Смысл-связь-глаг-формы”

нач Найти-множ-тематических-ролей (Rc, Rm. posvb, posdep, depth, Arls, Arvfr, Matr, nmasters, nrelvbdep, arrelvbdep)

{Найти количество элементов массива nrelvbdep и массив arrelvbdep, описывающий возможные смысловые отношения между глагольной формой и зависимой единицей текста}

если nrelvbdep = 1 то m1 := 1

иначе Выбор-тематич-роли (posvb, posdep, nrelvbdep, arrelvbdep, m1)

{m1 — номер строки в массиве arrelvbdep, дающей реализуемое в Т сочетание значения глагольной формы, значения зависимой единицы текста, и смыслового отношения между глагольной формой в позиции posvb и единицей текста в позиции posdep с учетом предлога, который может относиться к позиции posdep}

кесли

rel1 := arrelvbdep[m1,role]

locvb := arrelvbdep [m1,linevb] {строка из Arls}

если (class1 = сущ) то locnoun := arrelvbdep [m1,linenoun] {строка из Arls} кесли

⇒ {Внесение информации в Matr (см. описание Matr в главе 4)}

Matr[posvb].posdir := 0, Matr[posvb,locunit] := locvb, Matr[posvb,nval] := 1

если (class1 = сущ) то Matr[posdep,locunit] := locnoun кесли

Matr[posdep,nval] := 1, Matr[posvb].ndep := Matr[posvb].ndep + 1

Matr[posdep1].posdir := posvb, Matr[posdep].reldir := rel1

Конец

Комментарий к алгоритму "Смысл-связь-глаг-формы". Если nrelvbdep >1 то необходимо задать уточняющие вопросы пользователю, используя поле example массива arrelvbdep, и найти: locnoun – строку из Arls, указывающую значение единицы текста в позиции posdep, если эта единица является существительным; locvb – строку из Arls, указывающую значение глагольной формой в позиции posvb; rel1 – смысловое отношение (тематическую роль), между единицами текста в позициях posvb и posdep.

Если nrelvbdep =1, то m1 := 1, поэтому locvb= arrelvbn [1,linevb];

Locnoun = arrelvbn [1,linenoun]; rel1= arrelvbn [1,role].

Далее полученную информация запоминается в матрице Matr*:*

Matr[posdep,locunit]:=locnoun; Matr[posdep].posdir:=posvb;

Matr[posndep].reldir:=rel1.

В результате проведенного анализа однозначно определяются значения как глагольной формы в позиции posvb, так и существительного в позиции posdep. Поэтому Matr[posvb,nval] := 1 , Matr[posdep,nval] := 1 .

8.7.6 Заключительная часть описания алгоритма обработки глагольных форм

Внешняя спецификация алгоритма «Обработка-глаг-формы”

Вход: Rc, Rm**,** МССП Matr, Arls, Arvfr, одномерные массивы verbposmag[1:4], posqswd[1:nt], двумерный массив pos-free-dep[1: nt, 1:4], одномерный массив numb-free-dep[1: 4], целочисленные переменные pos, nsit, numbqswd, переменная depth (номер уровня глубины вложения рассматриваемого фрагмента текста), class – строка, обозначающая часть речи глагольной формы.

Выход: преобразованное значение МССП Matr..

Алгоритм “Обработка-глаг-формы”

Нач nsit := nsit + 1

(nsit – количество уже упомянутых в тексте ситуаций}

Matr[pos].mark := Var (‘e’, nsit)

verbposmag [depth] := pos,

Если ((class = прич)

То если Rc[pos, unit] ≠ ‘,’ то depth := depth+1 кесли

{Учтена возможность отсутствия запятой перед причастием, с которого начинается причастный оборот}

Поиск-существительного(pos, poscontr, Rc, Matr)

{см. описание алгоритма Поиск-существительного в подразделе 8.6.3}

Matr [pos, contr] := poscontr

{Пример. Пусть T1 = “Сколько предприятий, расположенных в Саратовской области, экспортируют продукцию в Болгарию?, и pos = 4. Тогда Matr [4, contr] := 2 (позиция слова “предприятий”)}

posvb := pos; posdepword := poscontr;

{Далее находится смысловое отношение между причастием в позиции posvb и управляюшим существительным в позиции poscontr}

class1 := сущ ; subclass := Rc[posdep, subclass];

nmasters[posdep] := nmasters[posdep] + 1;

Смысл-связь-глаг-формы (Rc, nt. Rm. posvb, posdep, class, subclass, class1, subclass1, depth, Arls, Arvfr, nmasters, Matr)

Кесли {завершение начальной части обработки причастия}

Если ((class = глаг) И (depth = 1) И (numbqswd > 0))

То {от позиции глагола в главном предложении проводятся управляющие стрелки с метками смысловых отношений (тематических ролей) к позициям вопросительных слов}

Цикл для k1 от 1 до numbqswd

P1 := posqswd [k1]; Matr[p1, posdir, 1] := pos

Кцикл

numbqswd := 0;

Кесли

Если numb-free-dep [depth] > 0 {на том же уровне глубины вложения существуют свободные единицы текста, т.е. такие единицы, для которых пока не найдено семантико-синтаксическое управление от другой единицы}

То Цикл для m1 от 1 до numb-free-dep [depth]

Смысл-связь-глаг-формы (Rc, nt, Rm, pos, pos-free-dep [depth, m1], depth, Arls, Arvfr, Matr, nmasters)

Кцикл конец

Пример. Пусть B1 – следующее размеченное представление вопроса: “Когда (1) и (2) где (3) будет проходить (4) очередная (5) международная (6) научная (7) конференция (8) “COLING” (9) ? (10)”. Тогда, если pos = 4, то будут проведены помеченные стрелки от позиции 4 к позициям 1 и 3 (в цикле по параметру k1 со значениями от 1 до numbqswd).

8.8. Обработка прилагательных , предлогов, количественных числительных, названий и существительных

8.8.1. Обработка прилагательных

Описание алгоритма “Обработка-прилаг”

Внешняя спецификация

Вход : pos – целое – порядковый номер единицы текста, являющейся прилагательным; Matr –МССП текста; Arls - массив – проекция лексико-семантического словаря (ЛСС) Lsdic на входной текст **T**.

Выход : nattr – целое – количество подряд идущих прилагательных; Attributes массив, имеющий следующую структуру:

Attributes

place	prop
позиция в Rc	семантическая единица для очередного прилагательного

Рис. 8.3. Структура строки вспомогательного массива Attributes

Пример. Пусть B1 = “Откуда (1) поступили (2) 2 (3) зеленых (4) алюминиевых (5) контейнера (6) ? (7)”. Тогда nattr:=2, а массив Attributes имеет следующий вид:

place	prop
4	Цвет (z1, зел)
5	Материал (z1 , алюм)
0	пустая строка

Рис. 8.4. Пример вспомогательного массива Attributes

Замечание. В конце алгоритма Обработка-сущ выполняются, в частности, следующие действия: nattr:=0; столбец place обнуляется, столбец prop заполняется цепочкой nil – обозначением пустой строки.

Алгоритм “Обработка-прилаг”

Нач nattr:=nattr+1, k1:= Matr[pos, locunit],
semprop:=Arls[k1, sem] ;
Attributes [nattr, place]:= pos,
Attributes [nattr, prop]:= semprop кон

8.8.2. Обработка предлогов, количественных числительных и названий

Алгоритмы “Обработка-предлога” и “Обработка-колич-числит” очень просты. Первый из них предназначен для запоминания предлога в рассматриваемой позиции pos с помощью переменной leftprep (“предлог слева”). Второй алгоритм преобразует лексическую единицу, относящуюся к классу количественных числительных, в число, обозначаемое данной лексической единицей. Например, слову “трех” и сочетанию “двадцать три“ соответствуют числа 3 и 23. Для запоминания числа предназначена переменная leftnumber (“число слева”). Входными параметрами этих алгоритмов являются классифицирующее представление текста Rc и переменная pos (номер строки в Rc).

Алгоритм “Обработка-предлога”

Нач Leftprep := Rc[pos, unit] кон

Алгоритм “Обработка-колич-числит”

Нач Leftnumber := Число (Rc[pos, unit]) кон

Описание алгоритма “Обработка-названий”

Внешняя спецификация алгоритма

Вход: Rc – классифицирующее представление текста, pos – позиция выражения в кавычках или апострофах, Matr – МССП текста.

Выход: преобразованное значение Matr.

Алгоритм

Нач Matr[pos, posdir, 1] := pos – 1; Matr[pos, reldir, 1] := ‘Название’

{Смысл операций: проведена управляющая стрелка с меткой ‘Название’ от выражения в кавычках или апострофах к существительному, стоящему слева от него} кон

8.8.3. Описание алгоритма поиска возможных смысловых связей между двумя существительными с учетом предлога

Назначение алгоритма “Найти-множ-отношений-сущ1-сущ2”

Алгоритм “Найти-множ-отношений-сущ1-сущ2” (“Найти-множество-смысловых-отношений-между-Существительным-1-и-Существительным-2”) позволяет установить смысловые отношения, которые могут существовать между существительным в позиции posn1 (в дальнейшем обозначается выражением Сущ1) и существительным в позиции posn2 (в дальнейшем обозначается выражением Сущ2) при условии, что ко второму существительному относится некоторый предлог, расположенный в позиции между posn1 и posn2.

Для этого потребуются три цикла: (1) по возможным значениям слова в позиции posn1, (2) по возможным значениям слова в позиции posn2, (3) по предложным фреймам, связанным с рассматриваемым предлогом..

Внешняя спецификация алгоритма алгоритма “Найти-множ-отношений-сущ1-сущ2”

Вход **Rc** - классифицирующее представление, **nt** – цел - – количество единиц текста в классифицирующем представлении R1, т.е. количество наборов в R1,

Rm – морфологическое представление лексических единиц, входящих в R1,

Posn1 – цел – позиция первого существительного, **Posn2** – цел – позиция второго существительного, **Matr** –МССП текста;

Arls – массив – проекция лексико-семантического словаря (ЛСС) **Lsdic** на входной текст **T**; **Arfrp** – массив – проекция словаря предложных фреймов **Frp** на входной текст **T**.

Выход **arrelvbdep** – двумерный массив, предназначенный для представления информации о значении первого существительного, значении второго существительного и о смысловом отношении между словом в позиции posn1 и зависимым словом в позиции posn2,

nreln1n2 – цел - количество значащих строк в массиве arrelvbdep.

Алгоритм "Найти-множ-отношений-сущ1-сущ2"

Нач nreln1n2 := 0

{Вычисление предлога} prep1 := Matr[posn2,prep]

{Вычисление множества грамматических падежей}

p1 := Rc[posn2,mcoord]; Grcases := Падежи (Rm[p1].morph)

line1 := Matr[posn1, locunit], numb1 := Matr[posn1, nval]

{количество строк в Arls со значениями существительного}

цикл для n1 от line1 до line1 + numb1 – 1 {цикл по строкам массива Arls, соответствующих существительному в позиции posn1}

Set1 := пустое множество

цикл для j от 1 до m {m – семантическая размерность сортовой системы S(B(Cb(Lingb))), т.е. наибольшее количество несравнимых сортов, которые могут характеризовать одну сущность}

current-sort := Arls[n1, st_j];

если current-sort ≠ nil то Спектр-сорта(current-sort, spectrum);

Set1 := Объединение множеств Set1 и spectrum кесли

{для произвольного сорта z spectrum (z) – это множество всех сортов, являюшихся обобщениями сорта z, включая сорт z} кцикл {по j}

{Пример Если u = дин.физ.об, то

spectrum (u) = { дин.физ.об, физ.об, простр.об }}

{цикл по значениям Сущ2}

line2 := Matr[posn2, locunit], numb2 := Matr[posn2, nval]

{количество строк в Arls со значениями Сущ2}

цикл для n2 от line2 до line2 + numb2 – 1 {цикл по строкам массива Arls, соответствующих существительному в позиции posn2}

Set2 := пустое множество

цикл для q от 1 до m {m – семантическая размерность сортовой системы S(B(Cb(Lingb))), т.е. наибольшее количество несравнимых сортов, которые могут характеризовать одну сущность}

current-sort := Arls[n2, st_q];

если current-sort ≠ nil то Спектр-сорта(current-sort, spectrum);

Set2 := Объединение множеств Set2 и spectrum кесли кцикл {по q}

цикл для k1=1 до narfrp {количество строк в массиве Arfrp – проекции словаря предложных фреймов Frp на входной текст}

если Arfrp[k1, prep] = prep1 {найден нужный предлог}

то нач s1 := Arfrp [k1, sr1]; s2 := Arfrp [k1, sr2];

если (s1∈ Set1) И (s2∈ Set2)

то если grc ∈ Grcases

то {отношение существует}

nreln1n2 := nreln1n2 + 1

arreln1n2 [nreln1n2, locn1] := n1; arreln1n2 [nreln1n2, locn2] := n2

arreln1n2 [nreln1n2, relname] := arfrp [k1, rel]

кесли

кесли

конец

кесли

кесли

кесли

конец

Комментарий к алгоритму “Найти-множ-отношений-сущ1-сущ2”

Найдено количество nreln1n2 смысловых отношений между существительными в позициях posn1 и posn2. Информация о таких комбинациях значений первого и второго существительных, которые дают хотя бы одно смысловое отношение между элементами в позициях posn1 и posn2, отображена во вспомогательном массиве arreln1n2:

Locn1	Locn2	relname	example
n1	n2	Против2	лекарство от астмы
…			

Рис. 8.5. Структура строки вспомогательного массива Attributes

- В столбце locn1 помещается n1 – номер строки массива Arls, задающей возможное значение существительного в позиции posn1.

- В столбце locn2 находится n2 – номер строки массива Arls, задающей возможное значение существительного в позиции posn2.

- Столбец relname предназначен для отображения возможных отношений между существительными в позициях posn1 и posn2.

Если nreln1n2 = 0, то не найдено смысловых отношений. Будем предполагать, что это невозможно для рассматриваемого входного языка.

Если nreln1n2 = 1, то однозначно определены значение существительного в позиции posn1 (по строке n1), значение существительного в позиции posn2 (по строке n2) и значение смыслового отношения arreln1n2 [nreln1n2, relname].

Если nreln1n2 > 1 то необходимо вызвать процедуру, которая задает уточняющие вопросы пользователю, и сформировать эти вопросы на основе примеров в столбце example.

План алгоритма “Обработка-сущ”

Нач

Занесение в Matr информации о стоящих (возможно) слева числе (или количественном числительном) и прилагательных посредством вызова алгоритма «Запись-атрибутов»

Генерация метки элемента и типа метки (вызов алгоритма «Вычисление-метки»)

Если Rc[pos +1, tclass] = сущ-собств то Обработка- сущ-собств кесли

Поиск смысловой зависимости от ближайшего слева существительного, управляемого глаголом в позиции verbposmag[depth]

Если такой зависимости нет

То в случае verbposmag[depth] ≠ 0 поиск смысловой зависимости от глагольной формы в позиции verbposmag[depth]

иначе (т.е. в случае verbposmag[depth] = 0) номер позиции pos заносится в массив свободных единиц текста pos-free-dep в строку depth, где depth – уровень глубины вложенности рассматриваемого фрагмента текста, включающего единицу в позиции pos кон

Внешняя спецификация алгоритма “Обработка-сущ”

Вход **Rc** - классифицирующее представление, **nt** – цел - – количество единиц текста в классифицирующем представлении Rc, т.е. количество наборов в Rc,

Rm – морфологическое представление лексических единиц, входящих в Rc,

pos – цел – позиция существительного, **depth** – цел - значение уровня глубины вложенности для слова в позиции **pos** ,

Matr – начальное значение МССП текста;

Arls – массив – проекция лексико-семантического словаря (ЛСС) **Lsdic** на входной текст **T**; **Arvfr** – массив – проекция словаря глагольно-предложных фреймов **Vfr** на входной текст **T**;

Arfrp – массив – проекция словаря предложных фреймов **Frp** на входной текст **T**.

Выход **pos** – цел - позиция единицы текста; **Matr** - преобразованное значение исходной матрицы **Matr**.

Внешние спецификации вспомогательных алгоритмов

Спецификация алгоритма “Найти-сущ-слева”

Вход: pos –цел – позиция существительного.

Выход: posleftnoun – цел – позиция ближайшего слева к позиции pos существительного, которое может оказаться управляющим словом для существительного в позиции pos (см. ниже подраздел “Описания вспомогательных алгоритмов”).

Спецификация алгоритма “Обработка-сущ-собств”

Вход: pos –цел – позиция существительного нарицательного или собственного, после которое следует хотя бы одно существительное собственное; Arls – проекция лексико-семантического словаря Lsdic на входной текст; Matr – исходное значение МССП текста.

Выход: Matr – преобразованное значение МССП текста (см. ниже подраздел “Описания вспомогательных алгоритмов”).

Спецификация алгоритма “Обработка-названий”

Вход: pos - позиция существительного нарицательного, после которого следует выражение в кавычках или апострофах; Matr – исходное значение МССП текста.

Выход: Matr – преобразованное значение МССП текста (см. подпараграф 8.8.2).

Спецификация алгоритма “Найти-множ-тематич-ролей”

Спецификация этого алгоритма и алгоритм приведены в подпараграфе 8.7.4.

Спецификация алгоритма “Смысл-связь-глаг-формы”

Спецификация этого алгоритма и алгоритм приведены в подпараграфе 8.7.5.

Спецификация алгоритма “Обработка-названий”

Спецификация этого алгоритма и алгоритм приведены в подпараграфе 8.8.2.

Спецификация алгоритма “Найти-множ-отношений-сущ1-сущ2“

Спецификация этого алгоритма и алгоритм приведены в подпараграфе 5.8.3.

Спецификация алгоритма “Выбор-управления-глаг-сущ”

Вход: **pos** – цел - позиция единицы текста; posvb – цел – позиция глагольной формы; posleftnoun – цел – позиция существительного слева; prep – строка – значение предлога, относящегося к позиции pos.

Выход: **res** – строка – получает значение 1 или 2 в результате уточняющего диалога с пользователем; если существительное в позиции pos непосредственно зависит от глагольной формы в позиции posvb, то res := 1; если существительное в позиции pos (с учетом предлога) непосредственно зависит от стоящего слева существительного в позиции posleftnoun, то res := 2.

Спецификация алгоритма “Выбор-отнош-между-сущ”

Вход: posleftnoun –цел – позиция существительного 1; pos - цел – позиция существительного 2, стоящего правее существительного 1; prep – строка – значение предлога (возможно, пустого предлога nil), относящегося к

существительному 2; arreln1n2 – двумерный массив, представляющий информацию о возможных комбинациях значения существительного 1, существительного 2 и смыслового отношения между ними с учетом предлога prep (см. описание массива arreln1n2 в подпараграфе 8.8.3); nreln1n2 – цел – количество значащих строк в массиве arreln1n2, т.е. количество возможных смысловых отношений между рассматриваемыми существительными.

Выход: m2 **–** цел – номер некоторой значащей строки массива arreln1n2. Параметр m2 приобретает ненулевое значение в результате обработки ответа пользователя на уточняющий вопрос ЛП. Пользователю предлагается указать, какое из нескольких

смысловых отношений реализуется в сочетании “Существительное 1 в позиции posleftnoun + зависимое Существительное 2 в позиции pos” с учетом предлога prep . Для этого пользователю с помощью столбца example даются примеры сочетаний, в которых реализуется такое же смысловое отношение, как и потенциально возможное отношение между единицами текста в позициях posleftnoun и pos.

Спецификация алгоритма “Выбор-тематич-роли”

Внешняя спецификация этого алгоритма приведена в подпараграфе 8.7.5.

Алгоритм “Обработка-сущ”

Нач если leftnumber > 0 то Matr[pos, qt] := leftnumber кесли

Если nattr > 0 то цикл для m от 1 до nattr

p1 := Attributes[m, place]; Matr[p1, posdir, 1] := pos;

Semprop:=Attributes[m, prop]; Matr[p1,.reldir, 1] := semprop кесли

leftnumber := 0 ; nattr := 0; Matr[pos, prep] := leftprep; leftprep := nil

Linenoun := Matr[pos, locunit] {номер набора из Arls, содержащего начальное значение существительного}

Sort1 := Arls[linenoun, st1]

Если Sort1 ≠ сит {ситуация}

То numbent := numbent + 1 {количество сущностей, упомянутых в просмотренной части текста} кесли

gramnumber := Число (Rc[pos, mcoord])

если gramnumber = 1 то Var1 := Varstring (‘x’, numbent) кесли

если gramnumber - число 2 или 3 то Var1 := Varstring (‘S’, numbent) кесли

Matr[pos, mark] := var1

Найти-сущ-слева (pos, posleftnoun)

Если posleftnoun = 0 {слева от позиции pos нет существительных, которые, возможно, управляют существительным в позиции pos}

То если verbposmag [depth] = 0

То numb-free-dep[depth] := numb-free-dep[depth] + 1

K1 := numb-free-dep[depth]; pos-free-dep[depth, k1] := pos

иначе posvb := verbposmag [depth]

Смысл-связь-глаг-формы (posvb, pos, Matr)

Иначе {в случае posleftnoun >0}

Найти-множ-отношений-сущ1-сущ2 (posleftnoun, leftprep, pos, Matr, nreln1n2, arreln1n2) {находятся возможные смысловые связи (и их количество) между рассматриваемым существительным в позиции pos и ближайшим слева существительным в позиции postleftnoun}

если (nreln1n2 = 0) {нет семантико-синтаксического управления от предыдущего существительного}

то posvb := verbposmag [depth

если posvb > 0

то Смысл-связь-глаг-формы (posvb, pos, Matr)

иначе { в случае posvb = 0}

numb-free-dep[depth] := numb-free-dep[depth] + 1

K1 := numb-free-dep[depth] ; pos-free-dep[depth, k1] := pos

кесли

кесли {случай nreln1n2 = 0 рассмотрен }

если (nreln1n2 > 0) {существует возможность семантико-синтаксического управления от предыдущего существительного}

то posvb := verbposmag [depth]

если posvb > 0

то Найти-множ-тематических-ролей (posvb, pos, class1, subclass1, Matr, nrelvbdep, arrelvbdep)

если (nrelvbdep = 0) {нет смысловой связи с глагольной формой}

то если (nreln1n2 = 1)

то m2 := 1 {m2 — номер эл-та массива arreln1n2, откуда берется инф-ция для Matr о связи между posn1 и posn2}

иначе Выбор-отнош-между-сущ (posn1, prep, posn2, nreln1n2, arreln1n2, m2) кесли

Добавление в Matr информации о связи между единицами текста в позициях posn1 и posn2, эта информация берется из позиции m2 массива arreln1n2;

кесли {случай nrelvbn2 = 0}

если (nrelvbdep > 0) {возможна связь с глаголом}

то если (nreln1n2>0) {возможна связь и с предыдущим существительным}

то Выбор-управления-глаг-сущ (posvb, prep, posn1, n2, res)

{res=1 ⇒ связь с глаголом; res=2 ⇒ связь с сущ. в позиции posn1}

если (res=1)

то если (nrelvbdep = 1) то m1:=1

иначе Выбор-тематич-роли (posvb, prep, posn2, nrelvbdep, arrelvbdep, m1)

{запись в Matr информации о связи между глагольной формой в позиции posvb и существительным в позиции pos, которая берется из строки m1 массива arrelvbdep}

nmasters[pos] := nmasters[pos] + 1;

{найдена новая управляющая стрелка, ведущая в позицию pos}

d := nmasters[pos]; Matr[pos, posdir, d] := posvb;

Matr[pos, reldir, d] := arrelvbdep [m1, role] ;

Matr[posvb, locunit] := arrelvbdep [m1, linevb]; Matr[posvb, nval] := 1;

Matr[pos, locunit] := arrelvbdep [m1, linenoun]; Matr[pos, nval] := 1

кесли кесли

если (res=2) {нет связи с глагольной формой, но есть связь с существительным. в позиции posleftnoun} то если (nreln1n2=1) то m2:=1

иначе

Выбор-отнош-между-сущ (posleftnoun, prep, pos, nreln1n2, arreln1n2, m2)

Кесли

{запись в Matr информации о смысловой связи между существительными в позициях posleftnoun и pos с учетом предлога prep (возможно, prep – это пустой предлог nil), которая берется из строки m2 массива arrelvbdep}

Matr[posleftnoun, locunit] := arreln1n2 [m2, locn1]; Matr[posleftnoun, nval] := 1;

Matr[pos, locunit] := arreln1n2 [m2, locn2]; Matr[pos, nval] := 1;

Matr[pos, posdir, 1] := posleftnoun;

Matr[pos, reldir, 1] := arreln1n2 [m2, role] Кесли

Если Rc[pos + 1, subclass] = сущ-собств

То logname := (слова в позициях pos и pos + 1 могут быть связаны с одним и тем же грамматическим падежом) И (семантические единицы, соответствующие этим словам в массиве Arls, имеют один и тот же набор сортов в Arls)

Если logname = Истина То Обработка-сущ-собств (pos) кесли кесли

Если Rc[pos + 1, subclass] = имя то Обработка-названий (pos) кесли

Leftprep := nil; leftnumber := 0; nattr := 0; обнулить столбец place массива Attributes; обнуляется, заполнить цепочкой nil – обозначением пустой строки - столбец prop массива Attributes.

Конец {алгоритма “Обработка-сущ”}

Описания вспомогательных алгоритмов

Описание алгоритма “Найти-сущ-слева”

Внешняя спецификация (см. выше)

Алгоритм

Нач posleftnoun := 0; p1 := pos

Цикл-до p1 := p1 – 1; classleft := Rc[p1, tclass]

Если classleft = сущ то posleftnoun := p1 кесли

Выход-при (p1 = 1) ИЛИ (posleftnoun > 0)

ИЛИ nlassleft ∈ {глаг, прич, наречие, местоим, констр, маркер}

кцикл кон

Пример. Пусть В1 = “Сколько контейнеров с индийской керамикой поступило из Новороссийска?”. Преобразуем вопрос В1 в следующее размеченное представление: “Сколько (1) контейнеров (2) с (3) индийской (4) керамикой (5) поступило (6) из (7) Новороссийска (8) ? (9)”. Пусть pos = 5 (позиция словоформы “керамикой”). Тогда после завершения работы алгоритма posleftnoun = 2 (позиция словоформы “контейнеров”).

Описание алгоритма «Обработка-сущ-собств»

Внешняя спецификация (см. выше)

Алгоритм

Нач k1 := pos + 1

Пока Rc[k1, tclass]= сущ-собств цикл

m1:= Matr [k1, locunit] {Найдена первая и единственная строка массива Arls с информацией о единице Rc[k1, unit]}

Matr[k1, posdir, 1]:= pos {Проведена управляющая стрелка от элемента в позиции pos к элементу в позиции k1}

sem1:=Arls[m1, sem]; Matr[k1, reldir, 1] :=sem1; k1:=k1+1 кцикл

pos : = k1 – 1 кон

Пример. Пусть В2 = “Сколько статей профессор Игорь Петрович Сомов опубли-ковал в 2003-м году?”. Тогда в результате вызова алгоритма Обработка-сущ-собств с параметром pos = 3 (позиция слова ”профессор”) будут как бы проведены управ-ляющие стрелки (посредством преобразования МССП Matr) от позиции pos к позициям pos + 1, pos + 2, pos + 3, соответствующим фрагменту “Игорь Петрович Сомов”.

8.9. Завершение разработки алгоритма построения матричного семантико-синтаксического представления входного текста

8.9.1. Описание головного модуля алгоритма

Для облегчения понимания головного модуля алгоритма построения МССП входного текста ниже приводится его внешняя спецификация (разработанная в параграфе 8.3).

Внешняя спецификация алгоритма SemSyn

Входные данные:

Lingb – лингвистический базис (л.б.);

T – текст из языка *Linp(G, Lingb)* , где - *G* – бесконтекстная грамматика вида (8.2.1).

Выходные данные:

nt – целое, количество единиц текста; **Rc** – классифицирующее представление входного текста T (см. параграф 7.1);

Rm – морфологическое представление входного текста (см. параграф 7.1);

Arls – множество упорядоченных наборов – проекция лексико-семантического словаря (ЛСС) Lsdic на входной текст T;

Arvfr – множество упорядоченных наборов – проекция словаря глагольно-предложных фреймов Vfr на входной текст T;

Arfrp – множество упорядоченных наборов - проекция словаря предложных семантико-синтаксических фреймов **Frp** на входной текст **T**;

Matr - матричное семантико-синтаксическое представление (МССП) входного текста (см. параграф 7.2).

numbqswd – переменная, отображающая количество вопросительных слов в предложении; одномерные массивы **posvbmag, numb-free-dep, posconnectword, nmasters**, двумерный массив **pos-free-dep** (структура и принципы использования этих массивов описаны в параграфе 5.6).

8.9.2. Внешние спецификации вспомогательных алгоритмов

Спецификация алгоритма “Построение-компон-морфол-представления”

Вход: **Lingb** – лингвистический базис; **T** – текст из Linp(G, Lingb), где G – бесконтекстная грамматика вида (8.2.1).

Выход: **Rc** - классифицирующее представление текста T; **nt** – цел – количество единиц текста в классифицирующем представлении Rc, т.е. количество значащих строк в Rc; **Rm** – морфологическое представление текста T**.**

Спецификация алгоритма “Построение-проекции-лексико-семантич-словаря”

Вход: **Rc**, **nt, Rm; Lsdic** - лексико-семантический словарь (см. параграф 4.4).

Выход: **Arls –** двумерный массив – проекция словаря Lsdic на входной текст T.

Спецификация алгоритма "Построение-проекции-словаря-глаг-фреймов"

Вход: **Rc**, **nt, Rm, Arls; Vfr** – словарь глагольно-предложных семантико-синтаксических фреймов (см. параграф 6.5).

Выход: **Arvfr –** двумерный массив – проекция словаря глагольно-предложных фреймов Vfr на входной текст T.

Спецификация алгоритма "Построение- проекции-словаря-предложных-фреймов"

Вход: **Rc**, **nt, Rm, Arls; Frp** – словарь предложных семантико-синтаксических фреймов (см. параграф 4.7).

Выход: **Arfrp –** двумерный массив – проекция словаря предложных фреймов Frp на входной текст T.

8.9.3. Алгоритм построения МССП входного текста

Алгоритм BuildMatr

Нач Построение-компон-морфол-представления (T, Rc, nt, Rm)
Построение-проекции-лексико-семантич-словаря (Rc, nt, Rm, Lsdic, Arls)
Построение-проекции-словаря-глаг-фреймов (Rc, nt, Rm, Arls, Vfr, Arvfr)
Построение-проекции-словаря-предложных-фреймов
(Rc, nt, Rm, Arls, Frp, Arfp)
Формирование-начальных-значений-данных
Выявление-вида-текста (nt, Rc, Rm, leftprep, mainpos, kindtext, pos)
Цикл-до pos := pos + 1
Class := Rc[pos, tclass]
выбор class **из**
предлог: Обработка-предлога (Rc, pos, leftprep);
прилаг: Обработка-прилаг (Rc, pos, nattr, Attributes)
колич-числит: Обработка-колич-числит (Rc, pos, numb);
сущ: Обработка-сущ (Rc, Rm, pos, Arls, Arfrp, Matr, leftprep, numb, nattr, Attributes)

местом: Обработка-местоим (Rc, Rm, pos, Arls, Rqs, Arfrp, Matr, leftprep)

наречие: Обработка- наречия (Rc, Rm, pos, Arls, Rqs, Matr)

глаг, прич: Обработка-глаг-формы (Rc, Rm, pos, Arls, Rqs, Arvfr, Matr, leftprep)

союз: Пустой оператор

констр: Обработка-конструкта

имя: Обработка-названий

маркер: если Rc[pos, unit] = ',' {запятая }

то Обработка-запятой (Rc, Rm, pos, Arls, Matr) кесли

квыбор

выход-при (pos = nt)

кон

Таким образом, в этом и предыдущих параграфах данной главы разработан алгоритм BuildMatr, находящий: (а) смысловые отношения между единицами ЕЯ-текста, (б) конкретные значения глагольных форм и существительных из текста. Эта информация отражена в строково-числовой матрице Matr.

Обрабатываемые алгоритмом тексты могут выражать сообщения (факты), вопросы и команды и могут включать глаголы (в неопределенной форме, изъявительном и повелительном наклонениях), причастия, существительные, прилагательные, числовые значения параметров (конструкты), количественные числительные и цифровые представления чисел, вопросительные слова (являющиеся вопросительно-относительными местоимениями и местоименными наречиями), союзные слова, являющиеся вопросительно-относительными местоимениями с лексемой “какой”. Входные тексты могут включать придаточные определительные предложения, составные описания множеств.

Построенный алгоритм BuldMatr является оригинальным и обладает рядом преимуществ по сравнению с известными подходами к алгоритмизации поиска смысловых отношений в ЕЯ-текстах. Эти преимущества и особенности алгоритма обсуждаются в заключительной части главы 9. Следует отметить, что алгоритм BuldMatr позволяет реализовать семантико-синтаксический анализ текстов из представляющих практический интерес подъязыков естественного (русского) языка.

Глава 9

АЛГОРИТМ СБОРКИ СЕМАНТИЧЕСКОГО ПРЕДСТАВЛЕНИЯ ТЕКСТА ПО ЕГО МАТРИЧНОМУ СЕМАНТИКО-СИНТАКСИЧЕСКОМУ ПРЕДСТАВЛЕНИЮ

9.1. Начальный шаг построения семантических представлений входных текстов

Алгоритм, преобразующий матричное семантико-синтаксическое представление (МССП) *Matr* в некоторое формальное выражение *Semrepr* $\in$ *Ls(B*) , где *B* – концептуальный базис, являющийся первым компонентом используемого размеченного концептуального базиса (р.к.б.) *Cb*, *Ls(B)* – СК-язык в базисе *B*, в параграфе 7.3 был назван *алгоритмом семантической сборки.*

Рассмотрим алгоритм “Подготовка-к-постр-СемП”, являющийся начальной частью разрабатываемого в данной главе алгоритма семантической сборки. Алгоритм “Начало-постр-СемП” строит начальное значение семантического представления (СП) входного текста, являющееся начальным значением строки *Semrepr* (“Semantic representation”) и зависящее от вида входного текста, т.е. от значения переменной kindtext, формируемого алгоритмом BuildMatr.

Выбор формы семантического представления входного текста в зависимости от значения переменной kindtext осуществляется на основе анализа, проведенного в параграфе 7.3. Некоторые примеры из этого параграфа используются ниже в алгоритме в качестве комментариев, показывающих контекст построения начального значения переменной. Для упрощения формы СП входного текста кванторы существования (когда они должны быть в соответствии с подходом, изложенным в главе 4) явно не указываются, а только подразумеваются.

Описание алгоритма “ Подготовка-к-постр-СемП”

Внешняя спецификация

Вход : Rc – массив – классифицируюшее представление входного текста; Rm – массив – морфологическое представление входного текста; kindtext – строка, характеризующая вид входного текста (возможными значениями этой строки

являются Stat, Imp, Genqs, Specqs-relat, Specqs-rol, Specqs-quant1, Specqs-quant2 (см. параграф 8.4); mainpos – целое число – позиция вопросительного слова в начале текста; Matr – МССП текста.

Выход : Semrepr – строка - начальное значение семантического представления входного текста.

Алгоритм “Подготовка-к-постр-СемП”

Нач Выбор kindtext из

Stat: Semrepr := пустая строка ;

{Пример. Пусть Т1 = “Профессор Игорь Новиков преподает в Томске”.

Тогда сначала Semrepr := пустая строка.

После завершения работы алгоритма BuildSem

Semrepr = Ситуация(e1, преподавание * (Время, #сейчас#)(Агент1, нек чел * (Квалиф, профессор)(Имя, ‘Игорь‘)(Фамилия, 'Новиков’) : x2)(Место1, нек город * (Название, 'Томск’) : x3)). }

Imp: Semrepr = *(Команда(#Оператор#, #Исполнитель#, #сейчас#, e1)*

{Пример. Пусть Т2 = “"Доставь ящик с деталями на склад № 3.".

Тогда сначала Semrepr := *(Команда(#Оператор#, #Исполнитель#, #сейчас#, e1)*

.

После завершения работы алгоритма BuildSem

Semrepr = *(Команда(#Оператор#, #Исполнитель#, #Сейчас#, e1) ∧ Цель (e1, доставка1*(Объект1, нек ящик * (Содерж1, нек множ * (Кач-состав, деталь)) : x1)(Место2, нек склад * (Номер, 3) : x2)))* }

Genqs: Semrepr := *Вопрос(x1≡Ист-знач (*

{Пример. Пусть Т3 = “Проходила ли в Азии международная научная конференция “COLING”?”. Тогда сначала

Semrepr := *Вопрос(x1, (x1≡Ист-знач (* .

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос(x1, (x1≡ Ист-знач (Ситуация (e1, прохождение2* (Время, нек мом * (Раньше ,#сейчас#) : t1)(Событие, нет конф* (Вид1, междун) (Вид2,*

научная) (Название, ‘COLING’) : x2) (Место, нек континент (Название, ‘Азия’) :x3))))).* }

Specqs-relat1, Specqs-relat2:

начало k1 : = R1 [mainpos, mcoord];

numb := Число (R2 [k1, morph]) {Значением переменной numb является код грамматического числа, соответствующего вопросительному слову с лексемой “какой” из начального сегмента входного вопроса; 1 - код единственного числа, 2 - код множественного числа }

если kindtext = Specqs-relat1 то Semrepr := ‘*Вопрос(x1,’

иначе Semrepr := ‘*Вопрос (S1, (Кач-состав (S1,'* конец

{Пример 1. Пусть T4 = “Какое издательство опубликовало роман «Ветры Африки»?”. Тогда сначала Semrepr := ‘*Вопрос(x1,’* . После завершения работы алгоритма BuildSem *Semrepr = Вопрос(x1, Ситуация(e1, опубликование * (Время, нек мом * (Раньше, #сейчас#) : t1) (Агент2, нек издательство: x1) (Объект3, нек роман1* (Название, ‘Ветры Африки’) :x3))) . }*

{Пример 2. Пусть T5 = “ С какими зарубежными издательствами сотрудничает писатель Игорь Сомов?”. Тогда сначала Semrepr := *Вопрос (S1, (Кач-состав (S1, .*

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос (S1, (Кач-состав (S1, издательство * (Вид-географич, зарубежное)) ∧ Описание (произв издательство* (Элем, S1) : y1, Ситуация(e1, сотрудничество * (Время, #сейчас#)(Агент1, нек чел* (Профессия, писатель)(Имя, ‘Игорь’)(Фамилия, ‘Сомов’): x1)(Организация1, y1))))) .* }

Specqs-rol: Semrepr := '*Вопрос (‘*

{Пример 1. Пусть T6 = “Кем выпускается препарат “Зиннат”?”.

Тогда сначала Semrepr := *Вопрос (.*

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос (x1, Ситуация (e1, выпуск1 * (Время, #сейчас#) (Агент1, x1)(Продукция1, нек препарат1 * (Название, ‘Зиннат’) : x2)))*

Пример 2. Пусть T7 = “Откуда и для кого поступил трехтонный алюминиевый контейнер?”. Тогда сначала Semrepr := *Вопрос (.*

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос ((x1 ∧ x2) , Ситуация (e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1) (Место1, x1) (Адресат, x2) (Объект1, нек контейнер * (Вес, 3/тонна)(Материал, алюминий) : x3))) .* }

Specqs-quant1: Semrepr := '*Вопрос(x1, ((x1≡ Колич(’ ;*

{Пример. Пусть T8 = “Сколько человек участвовало в создании статистического сборника?”. Тогда сначала Semrepr := 'Вопрос(x1, ((x1≡ Колич(’ .

После завершения работы алгоритма BuildSem

Semrepr = Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, чел) ∧ Описание(произв чел (Элемент, S1) : y1, Ситуация(e1, участие1* (Время, нек мом * (Раньше, #сейчас#) : t1) (Агент1, y1)(Вид-деятельности, создание1* (Продукт1, нек сборник1 * (Область1, статистика) : x2))))) .*

Specqs-quant2:

sortsit := выделенный сорт *сит (* “ситуация“) используемого концептуального базиса;

Semrepr := *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1,+ sortsit + ') ∧ Описание(произв’ + sortsit + ‘* (Элемент, S1) : e1,‘*

{Пример. Пусть T9 = “Сколько раз Иван Михайлович Семёнов летал в Мексику?”.

Тогда сначала Semrepr := *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, сит) ∧ Описание(произв сит * (Элемент, S1) : e1, .*

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, сит) ∧ Описание(произв сит * (Элемент, S1) : e1, Ситуация (e1, полёт * (Время, нек мом * (Раньше, #сейчас#) : t1)(Агент1, нек чел* (Имя, ‘Иван’)(Отчество, ‘Михайлович’)(Фамилия, ‘Семёнов’): x2)(Место2, нек страна* (Название, ‘Мексика’):x3)))))’ .* } квыбор кон

9.2. Построение семантических представлений коротких фрагментов входного текста с помощью алгоритма “Начало-постр-СемП”

9.2.1. Основные используемые структуры данных

В параграфе 7.3 были рассмотрены главные структуры данных, позволяющие по матричному семантико-синтаксическому представлению (МССП) входного текста построить его семантическое представление (СП), являющееся К-представлением, т.е. выражением стандартного К-языка в используемом концептуальном базисе. Такими структурами являются одномерные массивы Sembase (“Семантическая основа”), Semdes (“Семантическое описание”), Performers (“Исполнители ролей в ситуациях, упоминаемых во входном тексте ”) и двумерный массив Sitdescr (“Описание ситуаций”).

В данном параграфе разрабатывается алгоритм “Начало-постр-СемП”, предназначенный для формирования массивов Sembase, Semdes, Performers и начальной конфигурации массива Sitdescr.

9.2.2. Вспомогательные алгоритмы

Расмотрим алгоритмы, взаимодействие которых позволяет сформировать массивы Sembase, Semdes, Performers и начальную конфигурацию массива Sitdescr.

Описание алгоритма “Вычисление-вида-случая“

Внешняя спецификация

Вход : Rc – массив – классифицируюшее представление входного текста; k1 – номер строки классифицируюшего представления входного текста, т.е. порядковый номер единицы текста; Arls – двумерный массив – проекция лексико-семантического словаря (ЛСС) Lsdic на входной текст T; Matr – МССП текста; class1 – строка, задающая класс единицы текста; sem1 – семантическая единица, соответствующая k1–й единице текста.

Выход : casemark – строка, принимающая значения case1 – case7 в зависимости от вида обрабатываемого фрагмента классифицируюшего представления текста.

Алгоритм

Нач если class1 = прилаг то casemark := ‘Case1’ кесли

если class1 = констр то casemark := ‘Case2’ кесли

если class1 = сущ

то если Rc[k1 + 1, tclass] = имя то casemark := ‘Case3’

иначе numb1 := Matr[k1, qt]

{число, относящееся к существительному в позиции k1}

ref := нек **{**квантор референтности}

beg1 := sem1[1] {первый символ цепочки sem1, если считать каждый элемент первичного информационного универсума X(B(Cb(Lingb))) и каждую переменную из V(B) одним символом}

setind1 := 0 {признак обозначения индивида, а не множества индивидов}

len1 := Длина (sem1)

если (len1 ≥ 2) И (sem1[2] = ' множ ')

то setind1 := 1 кесли

{ sem1[2] – 2-й символ структурированной семантической единицы sem1 , если интерпретировать как символы элементы первичного информационного универсума *X(B(Cb))*, где *Cb* – используемый размеченный концептуальный базис (р.к.б.)}

если ((numb1 = 0) ИЛИ (numb1 = 1)) И

(beg1 = ref) И (setind1 = 0)

{т.е. RC[k1, unit] – обозначение индивида, а не множества }

то casemark := ‘Case4’ {Пример: ‘Бельгия’} кесли

если (numb1 = 0) И (beg1 ≠ ref) И (sem1 не является обозначением функции из F(B(Cb)), где Cb – используемый размеченный концептуальный базис) то нач loc1 := Rc[k1, mcoord], md1 := Rm[loc1, morph];

если (Число(md1) = 1) то casemark := ‘Case5’ {Пример: ‘конференция’}

иначе { т.е. в случае Число(md1) = 2) casemark := ‘Case6’

{Примеры: ‘5 статей’ , ‘3 международные конференции’} кесли кон кесли

если (numb1 = 0) И (setind1 = 1) {Пример: ‘с индийской керамикой’}

то casemark := ‘Case7’ кесли

конец

Описание алгоритмов Buildsemdes1 – Buildsemdes7

Внешняя спецификация каждого из алгоритмов Buildsemdes1 – Buildsemdes7

Вход : Rc – массив – классифицирующее представление входного текста; k1 – цел – номер строки из Rc; Arls –двумерный массив – проекция лексико-семантического словаря (л.с.с.) Lsdic на входной текст T; Matr – МССП текста; sem1 – строка – семантическая единица, соответствующая единице текста с номером k1; casemark – строка, принимающая значения case1 – case7 в зависимости от вида обработанного фрагмента классифицирующего представления входного текста; массивы Sembase, Semdes, Performers.

Выход: массивы Sembase, Semdes, Performers (эти массивы были описаны в параграфе 7.3), хранящие блоки для образования финального значения переменной Semrepr – семантического представления входного текста.

Описание алгоритма Buildsemdes1

Описание вспомогательных алгоритмов

Функция Transform1

Аргументы: s – строка вида r(z, b), где r – обозначение бинарного отношения, b – второй атрибут отношения, или вида (f(z) ≡ b) , где f - имя одноместной функции, b – строка, обозначающая значение функции, z – буква ‘z', интерпретируемая как переменная.

Значение: строка t вида (r, b) в первом случае и вида (f, b) во втором случае.

Пример. Пусть T1 = “Сколько двухтонных алюминиевых контейнеров поступило из Пензы?”. Тогда лингвистический базис может быть определен так, что для k1 = 2 sem1 := (Вес (z) ≡ 2/тонна) , Transform1(sem1)= (Вес ,

2/тонна) , для k1 = 3 sem1 := Материал (z , алюминий) , Transform1(sem1)= (Материал, алюминий).

Алгоритм Buildsemdes1

Нач {Отображение семантики прилагательных в массиве sembase}

Если Matr[k1 – 1, nattr] = 0

{непосредственно слева от позиции k1 нет прилагательных, т.е. в позиции k1 расположено первое прилагательное из группы идущих подряд прилагательных}

то Sembase[k1] := Transform1(sem1)

иначе {непосредственно слева от позиции k1 есть прилагательное}

Sembase[k1] := Sembase[k1 - 1] + Transform1(sem1)

{здесь знак + обозначает операцию конкатенации, т.е. операцию приписывания строки справа} кесли кон

Пример. В процессе использования алгоритма Buildsemdes1 для обработки вопроса Т1 = “Сколько двухтонных алюминиевых контейнеров поступило из Пензы?” будут выполнены операторы Sembase[2] := (Вес , 2/тонна) , Sembase[3] := (Вес , 2/тонна) (Материал, алюминий) .

Алгоритм Buildsemdes2

Нач {Обработка конструкта }

Sembase[k1] := sem1; Performers[k1] := Rc[k1, unit]

{Пример. Performers[k1] := ‘720/км’ } кон

Описание алгоритма Buildsemdes3 (“Обработка названий”)

Назначение: построение семантического представления (СП) фрагмента текста Т, являющегося сочетанием вида “Существительное + Выражение в кавычках или апострофах ”.

Условие вызова: в позиции k1 расположено существительное, в позиции k1 + 1 расположено выражение в кавычках или апострофах.

Пример. Пусть Т2 = «Кем выпускается препарат “Зиннат”?. Тогда в результате применения этого алгоритма будет выполнено присваивание

Performers[k1] := *нек препарат1 * (Назаниев, ‘Зиннат’)* .

Алгоритм

Нач name := RcT[k1 + 1, unit] ;

Если (Performers [k1] не включает символ *)

То Performers[k1] := Performers [k1] + ‘* (Название,’ + name + ‘)’

Иначе Performers[k1] := Performers [k1] + ‘ (Название,’ + name + ‘)’ кон

Алгоритм Buildsemdes4

Нач {Обработка существительных собственных}

{Пример контекста – опубликовал в Бельгии }

Sembase[k1] := sem1; Semdes[k1] := Sembase[k1]

Var1 := Matr[k1, mark] ; Performers[k1] := Semdes[k1] + ‘ : ‘ + var1

{Пример. Performers[k1] := ‘нек страна * (Название, ‘Бельгия) : x2’ } Кон

Алгоритм Buildsemdes5

Нач {Обработка нарицательных существительных }

{Пример контекста – опубликовал монографию}

если Matr[k1, nattr] ≥ 1 {слева есть прилагательные}

то Sembase[k1] := sem1 + ‘*’ + sembase[k1 – 1]

иначе Sembase[k1] := sem1 кесли

Ref := ‘нек‘ ; Semdes[k1] := ref sem1

Var1 := Matr[k1, mark] ; Performers[k1] := Semdes[k1] + ‘ : ‘ + var1

{Пример 1. Performers[k1] := ‘нек монография : x3’ }

{Пример 2. Performers[k1] := ‘нек принтер * (Вид, струйный) : x4’ } кон

Алгоритм Buildsemdes6

Нач {Обработка сочетаний с существительными, обозначающих множества объектов. Пример контекста – “Поступили 5 трехтонных контейнеров”}

numb1 := Matr[k1, qt] ; Sembase[k1] := sem1

Если numb1 > 0 то Semdes[k1] := *‘нек множ * (Колич,’ + numb1 + ‘)(Кач-состав,’ + sembase[k1] + ‘)’*

иначе Semdes[k1] := *‘нек множ * (Кач-состав,’ + sembase[k1] + ‘)’*
кесли

beg1 := sem1[1] {первый символ цепочки sem1, если считать символами элементы первичного информационного универсума X(B(Cb(Lingb))) и переменные}

Var1 := Matr[k1, mark] ; Var2 := Varsetmember(var1);

{Переменная var2 обозначает произвольный элемент множества с меткой var1. Пример. Если var1 = S2, то var2 = y2 }

Performers[k1] := ‘произвольн’ + beg1 + ‘* Элем(' + Semdes[k1] +‘ : ‘ + var1 + ‘): ‘ + var2 { Пример. Performers[k1] := ‘произвольн контейнер1 * (Элем, нек множ * (Колич, 5)(Кач-состав, контейнер1 * (Вес, 3/тонна)): S1) : y1’} кон

Описание алгоритма Buildsemdes7 (“Обработка собирательных существительных”)

Назначение: Построение фрагмента семантического представления (СП) текста T, являющегося сочетанием, включающим собирательное существительное (“индийская керамика”, “итальянская обувь” и т.п.).

Условие вызова: в позиции k1 расположено собирательное существительное.

Пример. Пусть ТЗ = “Откуда поступили три контейнера с индийской керамикой? “. Словоформа “керамикой“ в вопросе ТЗ имеет порядковый номер 7. Лингвистический базис может быть определен так, что в результате применения алгоритма Buildsemdes7 будут выполнены операторы

Semdes[7] := *нек множ * (Кач-состав, керамич-изделие * (Географич-локализация, нек страна * (Назв, ‘Индия‘))) ,*

Performers [7] := *нек множ * (Кач-состав, керамич-изделие * (Географич-локализация, нек страна * (Назв, ‘Индия‘))) : S1 .*

Описание вспомогательных алгоритмов

Функция Transform2

Аргументы: s – строка, отображающая семантику прилагательного или последовательности прилагательных; например, s может отображать семантику прилагательного “индийская “ и являться строкой *(Географич-локализация, нек*

*страна * (Назв, ‘Индия‘))* ; t – строка, являющаяся структурированной семантической единицей, соответствующей собирательному существительному и включающая подцепочку *(Кач-состав*, (например, t может соответствовать существительному “керамика“ и являться строкой *нек множ * (Кач-состав, керамич-изделие)*).

Значение: строка u , формируемая следующим образом. Пусть pos1 – позиция первой левой скобки (в подстроке *(Кач-состав*, строки s, и пусть pos2 – позиция правой скобки) , закрывающей скобку в позиции pos1. Пусть h – подстрока строки t, лежащая между подстрокой *(Кач-состав*, и правой скобкой в позиции pos2. Тогда u получается из строки t заменой подстроки h на строку h * s .

Пример. В контексте вопроса ТЗ = “Откуда поступили три контейнера с индийской керамикой? “ пусть *s = (Географич-локализация, нек страна * (Назв, ‘Индия‘)) , t = нек множ * (Кач-состав, керамич-изделие)* .
Тогда *h = керамич-изделие, u = Transform2 (s, t) = нек множ * (Кач-состав, керамич-изделие * (*Географич-локализация, *нек страна * (Назание, ‘Индия‘)))* .

Алгоритм Buildsemdes7

Нач если Rc[k1 – 1, tclass] ≠ прилаг то semdes[k1] := sem1

иначе prop1 := sembase [k1 – 1]; Semdes[k1] := Transfrom2 (prop1, sem1)

{Пример. Semdes[k1] := *нек множ * (Кач-состав, керамич-изделие * (Место-производства, нек страна * (Назв, ‘Индия‘)))* }

Описание алгоритма ProcessSit

Алгоритм ProcessSit предназначен для представления в массиве Sitdescr структурированных единиц концептуального уровня (другими словами, семантических единиц), соответствующих тем ситуациям, которые упоминаются во входном тексте с помощью глаголов или причастий.

Внешняя спецификация

Вход : Rc – массив – классифицируюшее представление входного текста T; k1 – номер строки классифицируюшего представления текста T, т.е. порядковый

номер единицы текста, являющейся глагольной формой; Rm – массив – морфологическое представление текста T; kindtext – строка – обозначение вида текста T; Arls – проекция лексико-семантического словаря Lsdic на входной текст T; Matr – МССП текста; Sitdescr – исходная конфигурация массива описания ситуаций, упоминаемых в тексте; timevarnumb – максимальный номер переменной, обознача-ющей момент времени.

Выход: Sitdescr - преобразованная конфигурация массива для описания упоминаемых в тексте ситуаций .

Описание вспомогательных алгоритмов

Функция Numb

Аргумент: v – строка вида *R S* , где *R* - буква латинского алфавита, *S* – строка, представляюшая натуральное число. **Значение:** *N* - натуральное число, ассоции-рованное со строкой *S* .

Пример. Для строки *e3 Numb(e3)* – это число 3.

Функция Stringvar

Аргументы: *R* - буква латинского алфавита, *N* - натуральное число.

Значение: строка вида *R S* , где *S* – строка, представляюшая натуральное число *N*.

Пример. Если *R* - буква ‘t’, *N* – число 2, то *Stringvar(R, N)* – это строка *t2*.

Функция Time

Аргументы: M- набор морфологических признаков, связанный с произвольной глагольной формой vbform (глаголом или причастием)

Значение: цифра ‘1’, если форме vbform соответствует прошедшее время; цифра ‘2’, если форме vbform соответствует настоящее время; цифра ‘3’, если форме vbform соответствует будущее время.

Алгоритм ProcessSit

Нач pos1 := Rc[k1, mcoord]

armorph := Rm[pos1, morph] {набор морфологических признаков, связанный с глагольной формой в позиции k1}

timevarnumb := timevarnumb + 1

Vartime := Stringvar (‘t’ , timevarnumb)

time1 := Time(armorph)

Выбор Time1 из

‘1’: timesit := ‘(Время, нек мом * (Раньше, #сейчас#) : ‘ + vartime + ‘)’

‘2’: timesit := ‘(Время, #сейчас#) ‘

‘3’: timesit := ‘(Время, нек мом * (Позже, #сейчас#) : ‘ + vartime + ‘)’

Квыбор

linesit := Matr[k1, locunit] ; concsit := Arls[linesit, sem]

var1 := Matr[k1, mark] ; numbsit := Numb (var1)

если (kindtext = Imp) И (numbsit = 1)

то Sitdescr [numbsit, expr] := ‘Цель (‘ + var1 + ‘,’ + concsit + ‘*’

иначе Sitdescr [numbsit, expr] := ‘Ситуация (‘ + var1 + ‘,’ + concsit + ‘*’ + timesit

кесли

{Пример 1. Sitdescr [1, expr] := *‘Ситуация (e1, выпуск1 *‘(Время, нек мом * (Раньше, #сейчас#) : t1)‘* }

{Пример 2. Sitdescr [1, expr] *:= 'Цель (e1, доставка1*(Объект1, нек контейнер : x1)(Место2, нек склад * (Номер, 4) : x2))'* } кон

Описание алгоритма “Начало-постр-СемП”

Внешняя спецификация

Вход : Rc – массив – классифицирующее представление входного текста T; Rm – массив – морфологическое представление текста T; Arls –проекция лексико-семантического словаря (ЛСС) Lsdic на текст T; kindtext – строка, характеризующая вид входного текста T; mainpos – целое число – позиция вопросительного слова в начале текста; Matr – МССП текста.

Выход : Semrepr – строка - начальное значение семантического представления входного текста; Performers – одномерный массив, содержащий семантические представления коротких фрагментов входного текста.

Алгоритм

Нач Подготовка-к-постр-СемП (Rc, Rm, Matr, kindtext, mainpos, Semrepr)

{Пример: Если kindtext = genqs (общий вопрос, т.е. вопрос с ответом “Да/Нет”), то Semrepr := ‘Вопрос (x1, (x1 ≡ Ист-знач (‘ }

цикл для k1 от 1 до nt

{формирование массивов Sembase, Semdes, Performers и начальной конфигурации массива Sitdescr}

class1 := Rc[k1, tclass]

если (class1 ≠ конструкт) И (class1 ≠ имя) И (class1 ≠ маркер)

то loc1 := Matr[k1, locunit] ; sem1 := Arls[loc1, sem] кесли

если (class1 = глаг) ИЛИ (class1 = прич)

то ProcessSit (k1, Rc, Rm, k1, Arls, Matr , Sitdescr, timevarnumb) кесли

если (class1 – элемент множества {прилаг, констр, сущ})

то Вычисление-вида-случая (Rc, k1, Arls, Matr, class1, sem1, casemark1);

Выбор casemark из

‘Case1’: Buildsemdes1 (List1),

где List1 – список параметров Rc, k1, Arls, Matr, Sembase, Semdes, Performers, casemark1;

‘Case2’: Buildsemdes2 (List1);

‘Case3’: Buildsemdes3 (List1);

‘Case4’: Buildsemdes4 (List1);

‘Case5’: Buildsemdes5 (List1);

‘Case6’: Buildsemdes6 (List1);

‘Case7’: Buildsemdes7 (List1)

квыбор

кон

9.3. Заключительные этапы разработки алгоритма сборки семантического представления входного текста по его матричному семантико-синтаксическому представлению

9.3.1. Основные идеи алгоритма “Отображение-ситуаций”

К моменту вызова алгоритма сформированы массив Performers и начальная конфигурация массива описания ситуаций Sitdescr. В массиве Performers представлены семантические единицы (первичные и составные), соответствующие конструктам (числовым значениям параметров), существительным и сочетаниям видов “Группа прилагательных + Существительное”, “Число + Существительное”, “Число + Группа прилагательных + Существительное”, “Количественное числительное + Существительное”, “Количественное числительное + Группа прилагательных + Существительное”.

Напомним (см. параграф 7.3), что количество заполненных строк массива Sitdescr равно количеству глаголов и причастий в тексте. В столбце mrk размещается метка ситуации (связь с МССП Matr осуществляется через элементы в этом столбце); столбец expr (сокращение от “expression” – “выражение”) предназначен для хранения семантических описаний ситуаций (событий), упоминаемых в тексте (см. таблицу в подразделе 7.3.1).

В рассматриваемом алгоритме “Отображение-ситуаций” преобразование информации осуществляется в два последовательных этапа. Первый этап представляет собою цикл по m от 1 до nt, где m – номер строки классифицирующего представления Rc, nt – количество элементов текста. В этом цикле информация о семантико-синтаксических отношениях в сочетаниях “Глагольная форма (глагол или причастие) + Зависимый фрагмент предложения” отображается в элементах столбца expr массива Sitdescr (каждый

из таких элементов является описанием определенной ситуации, упоминаемой во входном тексте).

При этом под зависимым фрагментом предложения понимается конструкт, либо существительное, либо сочетание одного из видов “Группа прилагательных + Существительное”, “Число + Существительное”, “Число + Группа прилагательных + Существительное”, “Количественное числительное + Существительное”. “Количественное числительное + Группа прилагательных + Существительное”.

Примерами зависимых фрагментов предложения являются выражения “в 2002-м году”, “европейские научные издательства”, “двухтонных контейнеров”, “5 контейнеров”, “двенадцать персональных компьютеров”.

Пример. Пусть B1 = “Сколько двухтонных контейнеров с индийской керамикой, поступивших из Новороссийска, было отправлено фирме “Парус”?

Тогда на первом этапе выполнения алгоритма выражения, являющиеся элементами столбца expr массива Sitdescr, пополняются информацией о семантико-синтаксических связях в сочетаниях “поступивших + двухтонных контейнеров”, “поступивших + Новороссийска”, “отправлено + двухтонных контейнеров”, “отправлено + фирме “Парус”.

В результате выполнения данного этапа алгоритма для вопроса B1 массив Sitdescr приобретет следующую конфигурацию:

mrk	expr
e1	*Ситуация(e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1) (Объект1, произв контейнер1 * (Элем, нек множ * (Кач-состав, контейнер1 * (Вес, 2/тонна) : S1) : y1)(Место1, нек город * (Название, 'Новороссийск‘) : x2))*
e 2	*Ситуация (e2, отправка1 * (Время, нек мом * (Раньше, #сейчас#) : t2) (Объект1, y1)(Адресат, нек фирма * (Название, 'Парус‘) : x3))*

Рис. 9.1. Структура массива Sitdescr на промежуточном этапе построения семантического представления входного текста

Вспомогательный массив Used длины nt позволяет избежать многократного повторения в столбце expr массива Sitdescr семантического представления одного и того же выражения, обозначающего объект или множество объектов. Первоначально для каждого m от 1 до nt Used [m] = 0. Если для некоторого k строка Performers [k] включается в состав некоторой строки массива Sitdescr, то Used [k] := 1. Поэтому в состав других строк массива Sitdescr (если такая необходимость возникает) включается не строка Performers [k] , а переменная, являющаяся окончанием строки Performers [k].

Например, в первую строку массива Sitdescr (рис. 9.2) входит выражение *произв контейнер1 * (Элем, нек множ * (Кач-состав, контейнер1 * (Вес, 2/тонна) : S1) : y1* , являющееся элементом Performers [3]. Во второй же строке массива Sitdescr вместо этого выражения использована переменная *y1*.

Фрагменты предложения, непосредственно управляемые глагольными формами (глаголами или причастиями), назовем зависимыми элементами 1-го уровня.

Второй этап алгоритма “Отображение-ситуаций” заключается в поиске по МССП Matr таких конструктов или сочетаний с существительным, которые непосредственно управляются зависимыми элементами 1-го уровня, т.е. в поиске зависимых элементов 2-го уровня. Например, в вопросе В1 фрагмент “двухтонных контейнеров” управляется глагольной формой “было отправлено**”** и поэтому является зависимым элементом 1-го уровня. В то же время сочетание “двухтонных контейнеров” управляет сочетанием “с индийской керамикой”.

Формальное представление *descr1* информации, передаваемой сочетанием вида “Зависимый элемент 1-го уровня X + Зависимый элемент 2-го уровня Y”, приписывается справа с помощью конъюнкции к элементу Sitdescr [k, expr], где k – порядковый номер ситуации, участника которой обозначает зависимыый элемент 1-го уровня X.

Вспомогательный массив Conj первоначально заполняется нулями. Если в результате выполнения второго этапа алгоритма к элементу Sitdescr [k, expr] приписывается справа с помощью конъюнкции (conjunction) некоторое

выражение, то Conj[k] := 1. Это значение 1 является сигналом об обрамлении элемента Sitdescr [k, expr] скобками (,) перед включением этого элемента в семантическое представление входного текста.

Пример. В результате выполнения второго этапа алгоритма для вопроса В1 к элементу Sitdescr [1, expr] с помощью конъюнкции будет приписано справа выражение *Содержание1 (y1, нек множ * (Кач-состав, керамич-изделие * (Географич-локализация, нек страна * (Назв,'Индия'))))* .

Так как использовалась конъюнкция, то элементу Conj [1] будет присвоено значение 1. В итоге массив Sitdescr приобретет следующую конфигурацию:

mrk	expr
e1	*Ситуация(e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1) (Объект1, произв контейнер1 * (Элем, нек множ * (Кач-состав, контейнер * (Вес, 2/тонна) : S1) : y1)(Место1, нек город * (Назв, 'Новороссийск') : x2)) ∧ Содержание1 (y1, нек множ * (Кач-состав, керамич-изделие * (Географич-локализация, нек страна * (Название,'Индия')))*
e2	*Ситуация (e2, отправка1 * (Время, нек мом * (Раньше, #сейчас#) : t2) (Объект1, y1)(Адресат, нек фирма * (Название, 'Парус') : x3))*

Рис. 9.2. Структура массива Sitdescr на заключительном этапе построения семантического представления входного текста

9.3.2. Описание алгоритма “Отображение-ситуаций”

Рассматриваемый ниже алгоритм предназначен для отображения информации об упоминаемых в тексте ситуациях (событиях) в массиве Sitdescr .

Внешняя спецификация алгоритма

Вход : Rc – массив – классифицируюшее представление входного текста; nt - целое число – длина входного текста (количество строк в Rc и Matr); kindtext –

строка – обозначение вида входного текста; Matr – МССП текста; Performers – двумерный массив, содержащий семантические образы участников ситуаций; maxnumbsit – цел – количество ситуаций, упоминаемых в тексте.

Выход : Sitdescr - массив, отображающий информацию о ситуациях, упоминаемых во входном тексте; Used[1 : nt] – одномерный массив для хранения признаков неоднократности использования структурированной семантической единицы в строках массива Sitdescr; Conj[1 : maxnumbsit] - одномерный массив для хранения признаков использования конюнкции в строках массива Sitdescr.

Алгоритм “Отображение-ситуаций”

Нач {Обработка прямых смысловых связей между глагольной формой и существительным или конструктом}

Цикл для j1 от 1 до nt Used [j1] := 0 кцикл

Цикл для j2 от 1 до maxnumbsit Conj [j2] := 0 кцикл

Цикл для m от 1 до nt {первый проход строк из Rc}

Class1 := Rc [m, tclass]

Если (class1 = сущ) ИЛИ (class1 = констр)

То нач d := nmasters[m]; {найдено количество единиц текста, управляющих m–й единицей текста}

Если d >0

То цикл для q от 1 до d

Нач p1 := Matr [m, posdir, q]; class2 := Rc[p1, tclass]

Если (class2 = глаг) ИЛИ (class2 = прич)

То var2 := Matr [p1, mark]; {метка ситуации, которую обозначает управляющая глагольна форма} numbsit := Numb (var2);

role := Matr [m, reldir, q] кесли

если (class1 = сущ)

то если kindtext не входит в множество {specqs-relat2, specqs-quant1}

то если Used [m] = 0 то actant := Performers[m]

иначе если (class1 = сущ)

то actant := Varbuilt(Matr[m, mark]) кесли;

если (class1 = констр) то actant := Rc[m, unit]) кесли

иначе {в случае kindtext входит в множество {specqs-relat2, specqs-quant1}}

если ((Used[m] = 1) ИЛИ ((Used[m] = 0) И (Matr[m, mark] = S1))

то actant := Varbuilt(Matr[m, mark])

иначе actant := Performers[m] кесли

кесли {kindtext}

{Varbuilt (x_j) = x_j ; Varbuilt (S_j) = y_j }

Sitdescr [numbsit, expr] :=

Sitdescr [numbsit, expr] + '(' + role + ',' + actant + ')'

Кцикл {конец цикла по q}

Кцикл {конец цикла по m }

{Пример 1. Пусть рассматривается вопрос B1 = "Откуда поступили 5 алюминиевых двухтонных контейнеров", и перед применением алгоритма "Отображение-ситуаций" выполнялось соотношение

Sitdescr[1, expr] = *Ситуация (e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1)* .

Тогда после применения алгоритма "Отображение-ситуаций" при определенном выборе размеченного концептуального базиса будет иметь место соотношение

Sitdescr [1, expr] = *Ситуация (e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1) (Место1, x1)(Объект1, произв контейнер1 * (Элем, нек множ * (Колич, 5) (Кач-состав, контейнер1 * (Вес, 2/тонна)(Материал, алюминий)) : S1) : y1*}

цикл для k1 от 1 до nt

{заполнение Sitdescr – второй проход Rc– обработка единиц текста, управляемых существительными, зависящими от глагольной формы}

class1:=Rc[k1, tclass]

если (class1= сущ) ИЛИ (class1= констр)

{примеры сочетаний: "контейнеров с индийской керамикой" (class1= сущ), "с лампами по 60 ватт" (class1= констр) }

то posmaster: := Matr[k1, posdir, 1]

{позиция словоформы “контейнеров” или словоформы “ лампами“ для указанных выше сочетаний}

если posmaster = 0 то вывод (‘Неправ. текст’)

иначе class2 := Rc[posmaster, tclass]

если class2 = сущ

то rel1 := Matr [k1, reldir, 1]

varmaster := Matr[posmaster, mark]

если (class1= констр) то arg2 := sembase [k1] кесли

если (class1= сущ)

то vardep:=Matr[k1, mark]

если Used[k1] = 0 то arg2 := Performers[k1]

иначе arg2 := vardep кесли

letter := первый символ цепочки varmaster {переменная varmaster соответствует существительному, управляющему единицей текста в позиции k1, причем для самого этого существительного управляющей единицей является глагольная форма – глагол или причастие}

если letter=’x’ то descr1:=rel1(varmaster, arg2)

иначе {т.е. в случае letter=’S’}

semhead:= первый элемент sembase[posmaster]

descr1 := rel1 + ‘(произв‘ + semhead + ‘*(Элем,‘

+ varmaster + ‘),’ + arg2 + ‘)’

{Затем с помощью конъюнкции $\wedge$ к выражению Sitdescr [numbsit, expr] , характеризующему рассматриваемую ситуацию с номером. numbsit, справа приписывается выражение descr1 }

Пример: Пусть B1 = “Какие писатели из Томска участвовали в конференции?”, и после выполнения первого цикла (для m от1 до nt) массив Sitdescr имет следующую конфигурацию:

mrk	expr
e1	*Ситуация(e1, участие * (Время, нек мом * (Раньше, #сейчас#) : t1)(Агент1, произв писатель * (Элем, нек множ * (Кач-состав,писатель) :S1) : y1)* *(Событие1, нек конференция : x1))*

Рис. 9.3. Структура массива Sitdescr после выполнения цикла с параметром m, изменяющимся от1 до nt.

Во втором цикле (для k1 от1 до nt) выполняется оператор Descr1:= *Географич-локализация (y1, нек город * (Назв, ‘Томск ‘) : x2)* , а затем массив Sitdescr приобретает новую конфигурацию:

Sitdescr

mrk	expr
e1	*Ситуация(e1, участие * (Время, нек мом * (Раньше, #сейчас#) : t1)(Агент1,, произв писатель * (Элем, нек множ * (Кач-состав, писатель) : S1) : y1)(Событие1, нек конференция : x1)) ∧ Географич-локализация (y1, нек город * (Назв, ‘Томск ‘) : x2)* .

Рис. 9.4. Структура массива Sitdescr после выполнения цикла с параметром k1, изменяющимся от1 до nt.

Конец примера}

конец

possit := Matr[posmaster, posdir, 1]

varsit := Matr[possit, mark] ; numbsit := Numb(varsit)

Sitdescr [numbsit, expr] := Sitdescr [numbsit, expr] + ‘∧’ + descr1

Если Conj [numbsit] = 0 то Conj [numbsit] := 1 кесли

{признак использования конъюнкции в строке Sitdescr [numbsit, expr] }

кесли кесли кесли кцикл

конец {алгоритма }

9.3.3. Описание алгоритма “Заключит-операции“

Рассматриваемый алгоритм предназначен для передачи информации, отраженной в масиве описания ситуаций Sitdescr, в финальном значении строки Semrepr - семантического представления входного текста.

Внешняя спецификация

Вход : Rc – массив – классифицируюшее представление входного текста; Rm – массив – морфологическое представление входного текста; kindtext – строка, характеризующая вид входного текста; mainpos – целое число – позиция вопросительного слова в начале текста; Matr – МССП текста; Performers – двумерный массив, содержащий семантические образы участников ситуаций; numbsit – цел – количество ситуаций, упоминаемых во входном тексте, т.е. количество заполненных строк массива Sitdescr; numbqswd – цел – количество вопросительных слов во входном тексте; Sitdescr - массив, отображающий информацию о ситуациях, упоминаемых во входном тексте; Semrepr – строка - исходное значение семантического представления текста.

Выход : Semrepr – строка - финальное значение СП входного текста.

Описание вспомогательных алгоритмов

Функция Right

Аргументы: pos1 – номер строки из классифицирующего представления Rc входного текста, т.е. номер позиции единицы входного текста; class1 – строка – обозначение класса единицы текста. **Значение:** pos2 – позиция ближайшей справа (к позиции pos1) единицы текста, относящейся к классу class1.

Функция Stringvar

Аргументы: *R* - буква латинского алфавита, *N* - натуральное число.

Значение: строка вида *R S* , где *S* – строка, представляюшая натуральное число *N*.

Пример: Если *R* - буква ‘t’, *N* – число 2, то *Stringvar(R, N)* – это строка *t2*.

Алгоритм

Нач pos2 := pos1

цикл-до pos2 = pos2 + 1; class2 := Rc [pos2, tclass]

выход-при (class2 = class1) конец

Алгоритм “Заключит-операции“

Начало цикл для k от 1 до maxnumbsit

event := Sitdescr [k, expr]

если Conj[k] = 1 то event := '(' + event + ‘)’ кесли

если k = 1 то situations := event

иначе situations := situations + ‘∧’ + event кесли кцикл

если maxnumbsit > 1 то situations := '(' + situations + ‘)’ кесли

{строка situations описывает ситуации, упоминаемые во входном тексте}

Выбор kindtext из

Stat: Semrepr := Situations

{Пример. Пусть T1 = “Профессор Игорь Новиков преподает в Томске”.

Тогда сначала Semrepr := пустая строка.

После завершения работы алгоритма BuildSem

Semrepr = *Ситуация(e1, преподавание * (Время, #сейчас#)(Агент1, нек чел * (Квалиф, профессор)(Имя, ‘Игорь‘)(Фамилия, 'Новиков’) : x2)(Место1, нек город * (Название, 'Томск’) : x3)).* }

Imp: Semrepr := Semrepr + ‘∧’ + *Situations* + *‘)’*

{Пример. Пусть T2 = "Доставь ящик с деталями на склад № 3".

Тогда сначала Semrepr := ‘*(Команда(#Оператор#, #Исполнитель#, #сейчас#, e1)*‘ .

После завершения работы алгоритма Buildsem

Semrepr = *(Команда(#Оператор#, #Исполнитель#, #Сейчас#, e1) ∧ Цель (e1, доставка1*(Объект1, нек ящик * (Содерж1, нек множ * (Кач-состав, деталь)) : x1)(Место2, нек склад * (Номер, 3) : x2))* }

Genqs: Semrepr := Semrepr + + situations + ‘)))’;

{Пример. Пусть T3 = “Проходила ли в Азии международная научная конференция “COLING”?”. Тогда сначала Semrepr := ‘*Вопрос(x1, (x1≡Ист-знач (‘* .

После завершения работы алгоритма Buildsem

Semrepr := ‘*Вопрос(x1, (x1≡ Ист-знач (Ситуация (e1, прохождение2* (Время, нек мом * (Раньше ,#сейчас#) : t1)(Событие, нет конф* (Вид1, междун) (Вид2, научная) (Название, ‘COLING’) : x2) (Место, нек континент* (Название, ‘Азия’) :x3)))))*. }

Specqs-relat1, Specqs-relat2:

нач Если Semrepr = ‘*Вопрос(x1,'*

{Пример. T4 = “Какое издательство опубликовало роман «Ветры Африки»?”}

то Semrepr = Semrepr + situations + ‘)’

{Пример. Для вопроса T4 Semrepr := ‘*Вопрос(x1, Ситуация(e1, опубликование * (Время, нек мом * (Раньше, #сейчас#) : t1) (Агент2, нек издательство: x1) (Объект3, нек роман1* (Название, ‘Ветры Африки’) :x3)))’* }

иначе {т.е. если Semrepr = ‘*Вопрос (S1, (Кач-состав (S1,'* }

posmainnoun:= Right (mainpos, сущ)

{Пример . Пусть T5 = “ С какими зарубежными издательствами сотрудничает писатель Игорь Сомов?”. Тогда mainpos = 2 (позиция вопросительного слова “какими“, posmainnoun := 4 (позиция слова “издательствами”) }

sem1 := sembase [posmainnoun]

если sem1 не включает символ ‘*’

то semhead := sem1

иначе loc1 := Matr [posmainnoun, locunit]

semhead := Arls [loc1, sem]

{Пример. Для вопроса Т5 sem1 := *издательство * (Вид-географич, зарубежное)* ,

semhead := *издательство* }

кесли

Semrepr = Semrepr + sem1 + *') ∧ Описание (произв'* + semhead + *'* (Элем . S1) : y1,'* + situations + '))'

{Пример. Для вопроса Т5 = " С какими зарубежными издательствами сотрудничает писатель Игорь Сомов?"

Semrepr := *Вопрос (S1, (Кач-состав (S1, издательство * (Вид-географич, зарубежное)) ∧ Описание(произв издательство* (Элем, S1) : y1, Ситуация(e1, сотрудничество * (Время, #сейчас#) (Агент1, нек чел* (Профессия, писатель)(Имя, 'Игорь')(Фамилия, 'Сомов'): x1)(Организация1, y1)))))* . }

Specqs-rol: {Пример 1. Пусть Т6 = "Кем выпускается препарат "Зиннат"?".

Тогда сначала Semrepr := *Вопрос (* .}

Unknowns := *'x1'*

Если numbqswd > 1

То цикл для k от 1 до numbqswd – 1

Vrb := Stringvar ('x', k) ; Unknowns := unknowns + '∧' + vrb кцикл

Unknowns := '(' + unknowns + ')' кесли

Semrepr := Semrepr + unknowns + ',' + situations + ')'

{конец Specqs-rol}

{ Пример 1. После завершения работы алгоритма BuildSem в случае вопроса Т6 = "Кем выпускается препарат "Зиннат"?"

Semrepr = *Вопрос (x1, Ситуация (e1, выпуск1 * (Время, #сейчас#) (Агент1, x1)(Продукция1, нек препарат1 * (Название, 'Зиннат') : x2)))*

Пример 2. Пусть Т7 = "Откуда и для кого поступил трехтонный алюминиевый контейнер?". Тогда сначала Semrepr := 'Вопрос ('.

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос ((x1 ∧ x2) , Ситуация (e1, поступление2 * (Время, нек мом * (Раньше, #сейчас#) : t1) (Место1, x1) (Адресат, x2) (Объект1, нек контейнер * (Вес, 3 тонна))(Материал, алюминий) : x3))) .* }

Specqs-quant1

posmainnoun:= Right (mainpos, сущ)

{Пример . Пусть T8 = “Сколько человек участвовало в создании статистического сборника?”. Тогда mainpos = 1 (позиция вопросительного слова “ сколько”, posmainnoun := 2 (позиция слова “ человек”) }

sem1 := sembase [posmainnoun]

если sem1 не включает символ ‘*’

то semhead := sem1

иначе loc1 := Matr [posmainnoun, locunit]

semhead := Arls [loc1, sem]

{Пример. Для вопроса T8 sem1 := *чел* , semhead := *чел* }

кесли

Semrepr = Semrepr + sem1 + *‘) ∧ Описание (произв’* + semhead + *‘* (Элем . S1)* :

y1,’ + situations + ‘))’

{Пример. Для вопроса T8 = “Сколько человек участвовало в создании статистического сборника?”

Semrepr := ‘Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, чел) ∧ Описание(произв чел (Элемент, S1) : y1, Ситуация(e1, участие1* (Время, нек мом * (Раньше, #сейчас#) : t1) (Агент1, y1)(Вид-деятельности, создание1* (Продукт1, нек сборник1 * (Область1, статистика) : x2))))) ‘ .*

Specqs-quant2: *Semrepr := Semrepr + situations + ‘)))’*

{Пример. Пусть T9 = “Сколько раз Иван Михайлович Семенов летал в Мексику?”

Тогда сначала Semrepr := *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, сит) ∧ Описание(произв сит * (Элемент, S1) : e1, .*

После завершения работы алгоритма BuildSem

Semrepr = *Вопрос(x1, ((x1≡ Колич(S1)) ∧ Кач-состав (S1, сит) ∧*

*Описание(произв сит * (Элемент, S1) : e1, Ситуация (e1, полёт * (Время, нек мом * (Раньше, #сейчас#) : t1)(Агент1, нек чел.*(Имя, ‘Иван’)(Отчество, ‘Михайлович’)(Фамилия, ‘Семёнов’): x2)(Место2, нек страна* (Название, ‘Мексика’):x3))))) .* }

9.3.4. Полный алгоритм сборки семантического представления текста

Описание алгоритма BuildSem (“Сборка-СемП”)

Внешняя спецификация

Вход : Rc – массив – классифицируюшее представление входного текста T; Rm – массив – морфологическое представление текста T; nt - целое число – длина текста T (количество строк в Rc и Matr); kindtext – строка, характеризующая вид входного текста; mainpos – целое число – позиция вопросительного слова в начале текста; Matr – МССП текста; Arls – проекция лексико-семантического словаря Lsdic на входной текст.

Выход : Performers – массив; Sitdescr - массив; Semrepr – строка - К-представление входного текста (семантическое представление входного текста, являющееся выражением некоторого стандартного К-языка).

Алгоритм BuildSem

Нач Подготовка-к-постр-СемП (Rc, Rm, Matr, kindtext, mainpos, Semrepr)
Начало-постр-СемП (Rc, Rm, Matr, kindtext, numbqswd, Arls, Performers, Sitdescr, Semrepr)
Отображение-ситуаций (Rc, Matr, Performers, Sitdescr)
Заключит-операции (Rc, kindtext, numbqswd, Matr, Performers, Sitdescr, Semrepr)
Конец

Таким образом, разработанный в данной главе алгоритм BuildSem (“Сборка-СемП”) преобразует МССП вопроса, команды или сообщения из широкого многообразия текстов на естественном (русском) языке в семантическое представление, являющееся выражением стандартного К-языка, задаваемого рассматриваемым размеченным концептуальным базисом – компонентом лингвистического базиса.

9.4. Алгоритм семантико-синтаксического анализа текстов на естественном (русском) языке

9.4.1. Описание алгоритма SemSyn (“Семантико-синтаксич-анализ-текста”)

Внешняя спецификация

Вход : *Lingb* – лингвистический базис; T - текст из множества *Texts(Tform(Lingb))* , где *Tform* - текстообразующая система, являющаяся одним из компонентов *Lingb*.

Выход : Semrepr – строка - К-представление входного текста (семантическое представление входного текста, являющееся выражением некоторого стандартного К-языка).

Алгоритм

Нач BuildMatr (T, Rc, Rm, Arls, kindtext, Matr, mainpos, numbqswd)

BuildSem (Rc, Rm, Arls, kindtext, Matr, mainpos, numbqswd, maxnumbsit, Semrepr) кон

Пример. Пусть T1 = «В каком московском издательстве в 2001-м году вышла работа по искусственному интеллекту «Основы обработки знаний» профессора Сомова ?». Лингвистический базис *Lingb* может быть определен таким образом, что на разных стадиях обработки T1 алгоритм SemSyn построит двумерные массивы Rc и Rm – классифицирующее и морфологические представления текста T1 (см.примеры в подразделе 7.1.1), рассмотренные в подразделе 7.1.2 двумерные массивы Arls (проекцию лексико-семантического словаря Lsdic на входной текст), Arvfr (проекцию словаря глагольно-предложных семантико-синтаксических фреймов Vfr на входной текст), Arfrp (проекцию словаря предложных семантико-синтаксических фреймов Frp на входной текст), а затем матричное семантико-синтаксическое представление (МССП) Matr (см. пример в параграфе 7.2).

На заключительном этапе работы процедура BuildSem построит К-представление текста Т1, являющееся строкой Semrepr вида

Вопрос (х1, Ситуация(е1, выход 1 (Издатель, нек издательство1* (Место, Москва) : х2)(Время, 2001/год)(Информ-объект, нек работа2 * (Название, ‘Основы обработки знаний’)(Область1, иск-интеллект)(Авторы, нек чел *(Квалификация, профессор)(Фамилия, ‘Сомов‘) : х4) : х3)))) .*

9.4.2. Обсуждение разработанного алгоритма семантико-синтаксического анализа текстов

Разработанный выше алгоритм SemSyn, базирующийся на построенной в главе 6 формальной модели лингвистической базы данных (ЛБД) и на введенном понятии матричного семантико-синтаксического представления (МССП), устанавливает смысловые отношения между элементарными значащими единицами входного текста, отражая эти отношения посредством МССП, а затем строит семантическое представление (СП) текста, являющееся выражением некоторого СК-языка (К-представлением). Входные ЕЯ-тексты могут выражать высказывания (сообщения), команды, специальные вопросы (т.е. вопросы с вопросительными словами), общие вопросы (т.е. вопросы с ответом «Да»/ «Нет»)и могут, в частности, включать причастные обороты и придаточные определительные предложения.

Алгоритм SemSyn позволяет устанавливать возможные смысловые отношения, в частности, в сочетаниях «Глагол + Предлог + Существительное», «Глагол + Существительное», «Существительное1 + Предлог + Существительное2», «Число + Существительное», «Прилагательное + Существительное», «Существительное1 + Существительное2», «Причастие + Существительное», «Причастие + Предлог + Существительное», «Вопросительно-относительное местоимение или местоименное наречие, играющее роль вопросительного слова + Глагол», «Предлог + Вопросительно-относительное местоимение + Глагол».

Вместе с результатами глав 6 и 7 алгоритм SemSyn выражает принципиально новый подход к семантико-синтаксическому анализу ЕЯ-текстов.

Чтобы продемонстрировать преимущества этого нового подхода по сравнению с подходами, отраженными в современной научной литературе, сравним алгоритм BuildMatr с алгоритмом семантико-синтаксического анализа ЕЯ-текстов, изложенным в монографии Дж.Ф. Люгера “Искусственный интеллект. Стратегии и методы решения сложных проблем”. 4-е издание этой монографии было опубликовано на английском языке в 2002-м году, перевод на русский язык опубликован в 2004 г. В книге, в частности, отмечается, что 4-е издание содержит обновленный материал по вопросам обработки естественных языков.

Процедура Sentence; {Анализ предложения}

Начало

вызвать процедуру noun_phrase для получения представления подлежащего;

вызвать процедуру verb_phrase для получения представления сказуемого с зависимыми словами;

с помощью объединения и ограничения связать понятие существительного, возвращаемое для подлежащего, с агентом графа для глагольной конструкции

конец

Процедура noun_phrase;

Начало вызвать процедуру noun для получения представления существительного;

выбор случая:

Неопределенный артикль и единственное число: понятие, определяемое существительным, является общим;

Определенный артикль и единственное число: связать маркер с понятием, определяемым существительным;

Множественное число: указать, что существительное во множественном числе

конец выбора случая

конец;

Процедура verb_phrase;

Начало вызвать процедуру verb для получения представления глагола;

если с глаголом связано дополнение

то начало вызвать процедуру noun_phrase *для получения представления дополнения;*

с помощью объединения и ограничения связать понятие, определяемое дополнением, с дополнением, соответствующим сказуемому

конец

конец;

Процедура verb*;*

Начало получить падежный фрейм для глагола конец;

Процедура noun*;*

Начало получить понятие для существительного конец.

Приведенный выше алгоритм отражает основные характерные черты доминирующего как в отечественной, так и зарубежной научной литературе подхода к описанию алгоритмов семантико-синтаксического анализа ЕЯ-текстов. Этими характерными чертами являются отсутствие модели лингвистической базы данных (заменяемое отдельными неформальными примерами используемых данных), отсутствие формального или достаточно четкого неформального описания структуры входных текстов и, как следствие, отсутствие в публикациях реальных алгоритмов семантико-синтаксического анализа (ССА) текстов или даже подробных методов выполнения ССА.

По существу, приведенный выше текст с названием *Процедура Sentence* является не алгоритмом, а лишь *пожеланием* разработать такой алгоритм. Разные специалисты в области компьютерной обработки ЕЯ разработают по этому пожеланию *разные* алгоритмы. Это относится не только к приведенному выше фрагменту из монографии Дж.Ф. Люгера, но и к подавляющему большинству публикаций, посвященных семантико-синтаксическому анализу ЕЯ-текстов.

Результаты данной монографии, изложенные в главах 6 - 9, дают не только продвижение вперед, но и *качественный скачок* в области разработки формальных средств и методов проектирования алгоритмов ССА ЕЯ-текстов. Разработчики ЛП впервые получили широко применимый *формальный аппарат для описания структуры данных,* с которыми работает алгоритм ССА, а также

детальный метод описания алгоритмов ССА и *оригинальный алгоритм ССА*, базирующийся на формальной модели ЛБД.

Существенным преимуществом разработанного алгоритма SemSyn является явный учет многозначности слов, что чрезвычайно важно для приложений.

Анализ построенного алгоритма SemSyn показывает работоспособность предложенного в главе 7 нового метода выполнения преобразования “ЕЯ-текст ➔ СП текста”. Важная особенность этого метода и алгоритма SemSyn заключается в том, что они не предусматривают использования синтаксического уровня представления (как результата выполнения синтаксического анализа) текста. Разработка алгоритма SemSyn показала, что такие традиционные понятия синтаксиса, как, например, подлежащее и дополнение, являются избыточными для компьютерной обработки ЕЯ-текста: семантическое представление текста может быть построено без опоры на эти понятия.

В этой связи можно отметить, что с учетом характера используемых данных из ЛБД и принципов применения этих данных для построения СП ЕЯ-текста без выполнения синтаксического анализа текста центральные идеи алгоритма SemSyn имеют некоторые общие черты с идеями компьютерной семантики русского языка.

Например, согласно работе (Тузов 2001), процесс компьютерного анализа текста делится на три части: морфологический анализ, предварительная пословная обработка текста и собственно семантический анализ текста. Последний этап характеризуется как выбор конкретного морфо-семантического значения словоформы и связывания всех слов предложения в единую семантическую структуру, причем на данном этапе используется семантический словарь.

С точки зрения материалов глав 7 - 9 данной книги, морфологический анализ и предварительная пословная обработка текста примерно соответствуют построению компонентно-морфологического представления текста. Для реализации собственно семантического анализа текста в данной монографии разработан сложный структурированный алгоритм семантико-синтаксического анализа ЕЯ-текстов SenSyn, описанию которого посвящены главы 6 – 9.

Однако уровень проработанности вопросов формального описания структуры ЛБД, структуры промежуточных данных и алгоритма преобразования ЕЯ-текстов в семантические представления в главах 6 – 9 данной монографии значительно выше, чем в публикациях по компьютерной семантике русского языка (Тузов 2001; Лезин, Каневский, Тузов 2002; Тузов 2003). В частности, в указанных публикациях отсутствуют формальная модель ЛБД и четкое описание алгоритма построения СП текста.

Разработка аппарата СК-языков в главах 2, 3 и применение этого аппарата в модели ЛБД (глава 6) и в алгоритме SemSyn позволили преодолеть трудности принципиального характера, касающиеся отображения содержания команд, а также вопросов нескольких видов: с вопросительными словами “какие”, “каких” и т.д., со словом “сколько”, относящимся к количеству предметов, и с ответом “Да /Нет”.

Алгоритм семантической сборки BuildSem, являющийся частью алгоритма SemSyn, существенно использует ряд новых выразительных возможностей, предоставляемых определением класса СК-языков.

Пример 1. Пусть В1 = “ С какими зарубежными издательствами сотрудничает писатель Игорь Сомов?”. Тогда для некоторого лингвистического базиса Lingb алгоритм SemSyn построит по вопросу В1 его К-представление (КП) в виде цепочки

Semrepr1= *Вопрос (S1, (Кач-состав (S1, издательство * (Вид-географич, зарубежное)) ∧ Описание(произв издательство* (Элем, S1) : y1, Ситуация(e1, сотрудничество * (Время, #сейчас#) (Агент1, нек чел* (Профессия, писатель)(Имя, ‘Игорь’)(Фамилия, ‘Сомов’): x1)(Организация1, y1)))))* .

Фрагментами цепочки Semrepr1 являются: (а) составное обозначение понятия *издательство * (Вид-географич, зарубежное)* , (б) семантическая характеристика произвольного элемента множества *произв издательство* (Элем, S1) : y1*, (в) составное обозначение объекта *нек чел* (Профессия, писатель)(Имя, ‘Игорь’)(Фамилия, ‘Сомов’): x1* . Правило Р[5] позволило связать метку (переменную) *y1* с произвольным элементом искомого множества *S1* , а затем использовать только эту метку для последующих ссылок на эту характеристику.

Пример 2. Пусть B2 = “Проходила ли в Азии международная научная конференция “COLING”?”. Тогда в рамках некоторого лингвистического базиса Lingb алгоритм SemSyn построит КП вопроса B2 в виде цепочки

Semrepr2 = *Вопрос(x1, (x1≡ Ист-знач (Ситуация (e1, прохождение2* (Время, нек мом * (Раньше ,#сейчас#) : t1)(Событие, нет конф* (Вид1, междун) (Вид2, научная) (Название, ‘COLING’) : x2) (Место, нек континент* (Название, ‘Азия’) :x3)))))*.

В выражении Semrepr2 цепочка *Ист-знач* интерпретируется как обозначение функции, аргументом которой является СП высказывания, а значением – логическая величина Истина или Ложь.

Таким образом, использование СК-языков для построения СП входных текстов лингвистического процессора позволило расширить возможности отображения особенностей смысловой структуры входных текстов по сравнению с другими известными подходами к построению СП ЕЯ-текстов. В частности, это относится к командам и к текстам с составными описаниями множеств.

По глубине проработки вопросов преобразования компонентно-морфологического представления текста в его СП и ясности описания предложенных решений разработанный алгоритм не имеет аналогов как в отечественной, так и зарубежной научной литературе на английском языке.

Содержание данной главы и глав 6 - 8 отражено в публикациях (Фомичев 1978б – 1980, 1986а – 1989, 1990в – 1992б, 2002а, б, в; Фомичев, Волчков 1999; Фомичев, Люстиг 2004; Fomichov 1992 – 1994, 1996a – 1998a, 2002a, 2002b, 2005в – 2005e; Fomichov, Akhromov 2001; Fomichov, Chuykov 2000; Fomichov, Kochanov 2001; Fomichov, Lustig 2001; Fomichova, Fomichov 2004).

9.5. Применение разработанного алгоритма к проектированию русскоязычных интерфейсов прикладных компьютерных систем

9.5.1. Русскоязычные интерфейсы баз данных и баз знаний

Работоспособность предложенного структурированного алгоритма семантико-синтаксического анализа текстов на естественном (русском) языке, разработанного в главах 8 и 9, доказана тем, что на его основе в рамках учебного процесса сконструирован целый спектр лингвистических процессоров (ЕЯ-интерфейсов) баз данных и баз знаний, реализованных в программных средах Turbo-Pascal 7.0, Borland-Pascal 7.0, Delphi 4.0, Delphi 5.0, C, C++, Visiul C++, Action Script, PHP.

Реализация ЕЯ-интерфейсов (русскоязычных интерфейсов) осуществлялась в МИЭМ в рамках курсового и дипломного проектирования, а также в “МАТИ” – Российском государственном технологическом университете им. К.Э. Циолковского в рамках выполнения курсовых работ по дисциплине “Проектирование лингвистических процессоров” и дипломного проектирования.

Рассмотрим характеристики наиболее интересных из этих программных реализаций.

ПРИМЕР 1. ЕЯ-интерфейс “Фармацевт” предназначен для обработки запросов к базам данных, хранящим сведения о различных лекарственных препаратах, а также для ввода в базу данных содержания сообщений о препаратах. Для реализации использовалась программная среда C++ 3.1 фирмы Borland.

Примерами входных текстов ЛП являются вопросы “Откуда поступил анальгин?”, “Сколько стоит анальгин?”, “Кто призводит анальгин?”, “Для кого поступил анальгин?”, “Когда и откуда поступил анальгин?”, “Откуда и для кого поступил анальгин?”, “Какие препараты, выпускаемые фирмой “GlaxoWelcome”, предназначены для больных астмой?”, “В каких странах выпускается препарат бекатит?”, “Какие европейские страны выпускают препарат серетид?” и сообщения “Анальгин поступил из Польши”, “Бромгексин поступил на склад”, “Со склада поступил инсулин”.

ПРИМЕР 2. “РУСЛАН-1” (РУСскоязычный Лингвистический АНализатор – Первая версия) – семантико-синтаксический анализатор вопросов к БД и команд интеллектуальному транспортно-погрузочному роботу. Включает подсистемы морфологического анализа и семантико-синтаксического анализа. Реализация в среде Turbo-Pascal 7.0.

ПРИМЕР 3. ЛП “СКЛАД ГПС”. Модельная предметная область: обработка ЕЯ-запросов к БД оператора автоматизированного склада ГПС. Реализация в среде Borland Delphi 4.0, язык Object Pascal. Отлажена в среде Windows 2000. Пример запроса: “Откуда и для кого поступили 3 двухтонных алюминиевых контейнера?”

ПРИМЕР 4. ЕЯ-интерфейс интеллектуальной вопросо-ответной системы АНТЕК-1 (АНАлиз ТЕКстов, версия 1) является семантико-синтаксическим русскоязычным анализатором вопросов и сообщений о событиях (о публикации научных работ, участии в научных конференциях, разработке новых приборов, получении и отправке товаров и т.д.).

Интерфейс реализован на языке C++ с использованием библиотек классов MFC и OLE DB Template Library фирмы Microsoft. Это позволило использовать в качестве хранилища базы знаний реляционные СУБД Microsoft Access или Microsoft SQL-Server, которые позволяют оперировать большими объемами данных.

При реализации алгоритмов поиска в базе данных существует возможность использовать для этих целей средства самой СУБД, что позволяет в некоторых случаях значительно упростить задачу поиска.

Пример 5. В среде Visual C++ разработан прототип русскоязычного интерфейса интеллектуальной базы данных, предназначенной для подбора вин и составления ресторанной винной карты в ходе взаимодействия конечного пользователя с Web–сайтом Российской ассоциации сомелье (РАС) и Web–сайтом Интернет-магазина, разработанного при поддержке РАС .

Ниже приведено несколько примеров входных текстов этого интерфейса; каждый из текстов может являться входом алгоритма семантико-синтаксического анализа, разработаннjuj в данной монографии:

• Посоветуйте белое французское вино к рыбному столу. • Какие красные вина, производимые в провинции Бургундия, импортирует фирма “Радуга”? • По какой технологии производят вино “Божоле”?

• Сколько фирм поставляют сухое белое вино “Бельканто”? • Выращивают ли виноград “Изабелла” в провинции Прованс?

•Откуда и кто поставляет сухое белое вино “Бельканто”?

•Из какого винограда производят красное вино “Божоле”?

•Сколько раз в апреле заказывали французское вино “Мерло“?

ПРИМЕР 6. Одна из исходных версий разработанного в данной главе алгоритма семантико-синтаксического анализа текстов была применена Я.В. Ахромовым (в рамках дипломного проектирования на кафедре информационных технологий МАТИ) к конструированию лингвистического процессора анимационной системы, предназначенной для имитации взаимодействия с интеллектуальным транспортно-погрузочным роботом, действующим в аэропорту (см. рисунок 9.5).

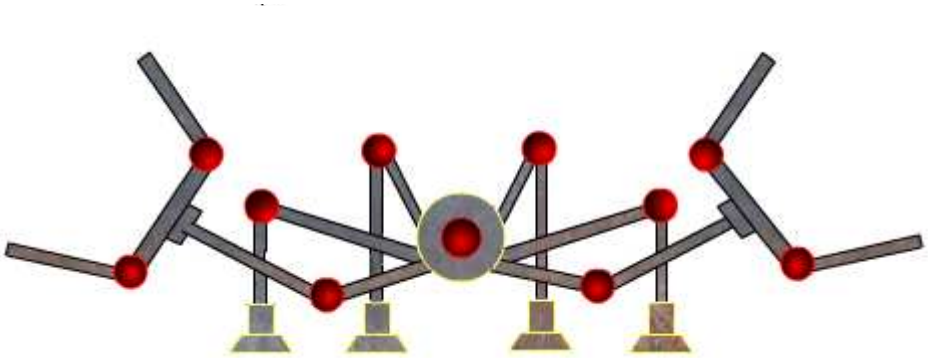


Рис. 9.5. Внешний вид возможного авиаробота

Анимационная система с ЛП, обрабатывающим команды и вопросы авиароботу , проектировалась с использованием среды разработки Flash 5. Этот выбор был сделан в связи с тем, что среда Flash 5 может быть использована как графическая программа, программа обработки звука и среда программирования.

Для реализации ЛП использовался язык программирования Action Script, встроенный в среду Flash 5.

Работа ЛП и связанных с ним подсистем направлена на осуществление общения оператора погрузочно-разгрузочных работ с авиароботом. Разработанный робот является только виртуальной моделью и не имеет реальных прототипов на практике. Задача оператора заключается в том, чтобы вводить команды авироботу средствами естественного языка, например: “Погрузи 3 контейнера в самолёт авиакомпании SAS, рейс (SK-6787).” Авиаробот является в этой системе некоторым интеллектуальным агентом, который способен оценивать текущую ситуацию с погрузкой в соответствии с динамическим расписанием погрузочно-разгрузочных работ и отслеживать верность запрашиваемой оператором команды.

Входной текст ЛП (команда или вопрос) подвергается анализу на корректность. Например, если оператор введет количество контейнеров для погрузки больше, чем заявлено в расписании погрузочно-разгрузочных работ, система должна будет отреагировать на это сообщением об ошибке. Другим примером может быть ситуация, при которой оператор указывает рейс, который уже произвел взлет. Тогда система должна будет уведомить оператора, что данный рейс является недоступным для погрузочных работ.

Помимо этого, оператор может задавать авироботу вопросы нескольких типов, например, “Сколько контейнеров было погружено на рейс (SK-6787)?”, “Грузил ли контейнеры к терминалу <номер терминала>?”, “Сколько контейнеров погрузил на все рейсы авиакомпании <имя авиакомпании>?”.

Это даёт возможность оператору следить за ходом работы авиаробота, за динамикой погрузки, прогнозировать следующие погрузочные работы.

Разработку семантико-синтаксического анализатора письменных вопросов и сообщений авироботу можно рассматривать как шаг на пути организации устного взаимодействия оператора и транспортно-погрузочного авиаробота.

Одной из базовых процедур алгоритма BuildMatr (алгоритма построения матричного семантико-синтаксического представления входного текста) является алгоритм “Найти-множ-тематич-ролей” (параграф 8.7), существенно использующий в работе словарь глагольно-предложных семантико-

синтаксических фреймов (см. параграф 6.5). Применяя терминологию статей (Баллард 1989; Хейз, Гауптман, Карбонелл, Томита 1989), можно сказать, что работа алгоритма “Найти-множ-тематич-ролей” основывается на применении семантических падежных фреймов.

Во второй из указанных статей на основе экспериментальных исследований, проведенных в Университете Карнеги-Меллон, сделан вывод о перспективности использования семантических падежных фреймов для семантико-синтаксического анализа устной речи. В этой связи можно сделать заключение о перспективности использования разработанного в данной главе алгоритма семантико-синтаксического анализа ЕЯ-текстов (алгоритма SemSyn) при проектировании анализаторов устной речи, т.е. при решении одной из актуальных проблем разработки лингвистических информационных технологий.