МЕТОДЫ И СРЕДСТВА СИСТЕМ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

С. Л. Крывый, Н. П. Дарчук, И.С. Ясенова,
А.Л. Головина, А.С. Соляр

Аннотация: Рассматривается краткий обзор средств систем анализа, обработки и представления знаний, построения онтологий по естественноязыковым текстам. В частности, представлен обзор средств Дескриптивных Логик, которые активно развиваются и применяются. Приводится пример представления конкретной предметной области в виде онтографа с описанием операций на онтографах и на онтологиях.

ACM Classification Keywords: Systems of knowledge representation, Description Logics, ontograph, operations over ontologies

Введение

Одним из современных способов управления является информация. Информационное влияние – это форма управления, отличающаяся от других форм управления средством влияния, которым является содержание информации. Степень влияния информации на личность зависит от многих факторов: семантического содержания информации, способом представления и источника информации, уровня информационной культуры личности и т. п.. Степень влияния информации базируется на ее свойствах, интересующих личность, а также на ее надежности и достоверности.

Развитие информационных технологий увеличило количество источников информационных потоков, созданных для открытой публичной передачи информации при помощи разных технических инструментов. Так, наряду с традиционными СМИ (печатные СМИ, радио, телевидение) в формировании информационных потоков большую роль играют онлайновые социальные сети (ОСС). ОСС являются каналами распространения информации не только официального содержания или непредвзятого освещения фактов, но и источниками, которыми есть политики, известные личности. Таким образом, ОСС – это одно из средств общения, которое позволяет каждому участнику выразить свой субъективный комментарий или оценку, или же самому быть источником информации – лично формировать контент.

Контент (заимствовано в 1990-е годы из английского content “содержание”) – основа коммуникации в ОСМ и способ построения взаимосвязей между участниками.

Каждый день ОСМ генерируют миллионы фактов в минуту как глобально-информативного, так и субъективно-эмоционального характера. Контент является средством влияния участников ОСМ один на другого и, соответственно способом анализа окружающей действительности с отслеживанием общественно-политических настроений. Распространенную разными участниками ОСМ факты часто могут быть связаны, а значит, подтверждаться и взаимодополняться. Но возможны случаи дублирования фактов, что может свидетельствовать либо о действительном множестве реальных информационных сведений, либо о информационной атаке, основной целью которой является распространение неправдивой или обманчивой информации. Очевидно, что для определения достоверности сведений связанное множество фактов весомее чем одиночные факты.

Проблемы анализа содержания информации

В связи со сказанным приобретают актуальность системы автоматического выделения различных фактов с целью их анализа и интеграции. Наиболее распространенной формой подачи фактов в ОСМ являются естественноязыковые тексты. Они легко воспринимаются, порождаются, тиражируются и модифицируются. Соответственно возникает проблема извлечения знаний из текстов. Неформально будем считать, что знания – это системно зафиксированная в сознании человека или информационной системы совокупность фактов, которые более менее адекватно отображают реальные взаимосвязи материальных и абстрактных объектов окружающего мира [1]. Исследованиям в сфере анализа текстов естественного языка посвящено большое количество отечественных и зарубежных публикаций, среди которых нужно выделить генеративную грамматику Хомского [2; 3], предикатно-аргументные структуры Ч. Гиллмора [4], модель концептов Р. Шенка [5; 6] и т.д. В трудах Хомского реализован подход к исследованию глубинной синтаксической структуры текста и построения дерева синтактико-семантических зависимостей и выявления семантических аномалий. В трудах Ч. Гиллмора и Р. Шенка были введены понятия "концепт" и "фрейм": структуры типа "предикат-аргумент". В трудах И. Мельчука [7] разработана теория, ориентированная на многоуровневое превращение содержания в текст и текста в содержание. В трудах Н. Дарчук разработаны основы теоретического и экспериментального обоснования лингвистических и процедурных принципов компьютерного аннотирования текста с созданием на этой основе компьютерной грамматики (система АГАТ) для автоматического анализа текстов естественного языка [8].

Модели, в которых используются синтаксические зависимости и толково-комбинаторные словари (см. [9]) дали начало современным тезаурусу и онтологии. Так, под семантикой языковых знаков в прикладной лингвистике понимают информацию, связанную со словом, его значением, которое представлено в толковом словаре. Под содержанием понимают функцию интерпретации конечной последовательности символов, в рамках априори согласованной семантики [1].

Анализ естественноязыковых текстов требует глубинного семантического анализа, который связан с построением модели фрагмента знаний, полученных из этих текстов. На сегодня ведущей парадигмой структурирования информации по тематике, разделам и содержанию является онтологическая парадигма, которая предусматривает создание логикоориентованной формальной концептуальной модели предметной области Δ. Одним из средств автоматизации процесса построения таких моделей в данный момент выступают Дескриптивные Логики (ДЛ) [10], которые активно развиваются и служат базой для построения систем представления знаний, построения онтологий и исследования свойств полученных фактов из разных источников информации. Создание онтологий в виде концептуальной базы знаний является фундаментальной проблемой в сфере компьютерной лингвистики и инженерии знаний. Для решения этой проблемы необходимо разработать:

– методы анализа естественноязыковых текстов с целью получения знаний;

– методы проверки знаний на совместимость и непротиворечивость;

– методы построения онтологий.

Сложность решения данной проблемы связана с целым рядом лингвистических, логических, теоретических и технических проблем. Что касается лингвистических проблем, то естественный язык содержит много метафор – переносных значений слов, которые вытесняют начальные значения, присутствие которых значительно усложняет анализ текстов. Переносные значения характерны для каждой языковой культуры и понимаются субъектами коммуникации в зависимости от контекста. Соответственно, это же должны делать информационные системы формирования онтологий. Кроме этого, существует проблема концептов – присутствие у субъекта коммуникации некоторого представления о фрагменте мира или части такого фрагмента. Такой фрагмент имеет сложную структуру, выраженную разными группами признаков, которые реализуются разными языковыми способами и средствами [11]. Концепт отображает категориальные и ценностные характеристики знаний о некотором фрагменте мира, имеющего значение для субъекта коммуникации. Решения этих вопросов натыкается на ряд трудностей, учитывающих факторы сложности естественного языка и психологии восприятия информации, заложенной в этом языке. Значительной преградой на пути автоматизации процесса анализа текстов естественного языка является то, что часть значимой информации, как правило, находится за пределами данного текста. Восстановление этой недостающей информации людьми в процессе коммуникации является процессом естественным, а в формальных системах это совсем не так.

Важной задачей в этом процессе является построение соответственной терминологии. Значимость терминологии проявляется в том, что рассуждения выражаются определенной словесной формой в виде предложений некоторого полуформального языка L, а концепты – в виде терминов [12; 13]. В идеале каждый термин должен иметь единственный точно определенный смысл (это абсолютизируется в математике), чего нельзя достичь в естественном языке в связи с неоднозначностью его семантики. Поскольку для естественного языка в целом этого достичь нельзя, то рассматривается несколько ослабленное требование: интерпретация (терминов) концептов должна быть как можно более точной. Чем точнее интерпретация концептов, тем более убедительными становятся рассуждения и результаты анализа. Это ослабленное требование дальше сужается к отдельно взятой предметной области (ПО). В некоторых ПО оно может выполняться в полном объеме, а в некоторых, более сложных, частично, но более менее удовлетворительно. С этой целью свойства объектов предметной области выражаются в некотором формальном языке L', который служит основой построения концептуальной формальной модели ПО и средством решения задач в этой ПО. Схематически такой процесс можно представить так:

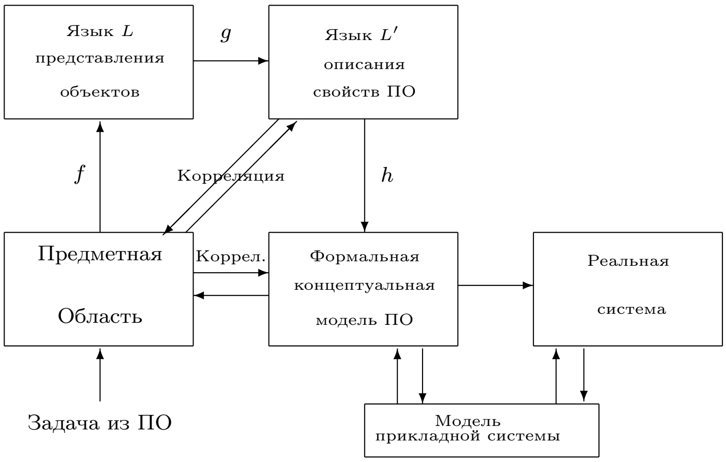

Рис 1. Общая схема

В процессе решения задачи в ПО возникают вопросы, связанные с языком представления объектов ПО, которые фигурируют в ПО.

Выбор языка представления объектов зависит от ПО и поставленной задачи. В идеале таким языком мог бы быть язык предикатов первого порядка или другой язык формальной логики с достаточными выразительными свойствами. Наличие такого языка облегчает задачу определения свойств как самого языка, так и объектов ПО. А это, в свою очередь, облегчает построение формальной модели.

После того как проведен анализ возникает необходимость проверки правильности найденных свойств. Эта проверка, названная Корреляцией, сводится к проверке истинности свойств на объектах ПО. Она может привести к пересмотру постановки задачи или к уточнению постановки задачи. Аналогичная потребность возникает при проверке построенной модели на соответствие поставленной задаче. Если модель решает задачу в рамках ограничений абстракции, то можно приступать к созданию реального изделия (если оно предусмотрено заданием) или использовать эту модель для других целей.

Названные проблемы объединяют сферы когнитивной и компьютерной лингвистики, инженерии знаний, теории поиска доказательств в разных языках математической логики, теории графов, психологии и т.д. В связи с названными проблемами в последнее время появилось много попыток решить (хотя бы частично) эти проблемы. Если средства построения онтологий более менее разработаны, то проблемы получения знаний и проверки их совместимости (непротиворечивости) являются одними из основных в этой области.

Одной из попыток построения систем представления и манипулирования знаниями, как отмечалось, являются дескриптивные логики [10], которые в данный момент активно развиваются и используются и в которых большое внимание уделяется формализации построения терминологии ПО.

Неформальное описание ДЛ

ДЛ – это совокупность формализмов для представления знаний некоторой ПО (такую ПО иногда называют “миром”). С помощью формализмов ДЛ сначала определяют концепты данной ПО, а потом, пользуясь этими концептами, специфицируют свойства объектов и индивидуумов (констант) этой ПО. Далее, опираясь на найденные свойства, проверяют некоторые гипотезы, тезисы или факты, являются ли они следствиями этих свойств.

Само название ДЛ выражает одну из характеристик этих логик, а именно, формальную логико-ориентированную семантику. Другой характерной особенностью ДЛ является возможность проводить логический вывод новых фактов из существующих, что составляет ее основной сервис. ДЛ поддерживает средства вывода не только свои собственные, но и средства других систем, с которыми она сотрудничает или в которые она встроена. Классификация концептов определяет подконцепты, надконцепты и отношения (которые называют отношениями включения) между концептами в заданной терминологии, что дает возможность структурировать терминологию в виде некоторой иерархии. Такого типа иерархия открывает полезную информацию на связанных между собой концептах, что ускоряет процесс вывода в центральном сервисе. Поскольку ДЛ является формализмом для представления знаний, то, как правило, предполагается, что система представления знаний должна всегда отвечать на вопрос пользователя в разумном интервале времени. В связи з этим возникает интерес к процедурам вывода в ДЛ и исследования их алгоритмической разрешимости. В отличии от пруверов в логике первого порядка, такие процедуры должны быть терминальными (то есть, всегда заканчивать свою работу) независимо от того, позитивным или негативным будет ответ. Следовательно, проблема поиска разрешимых процедур вывода, которые работают в разумном интервале времени и их временные характеристики, является актуальной в ДЛ.

Сложность решения проблемы вывода зависит от выразительных возможностей ДЛ. Эта сложность заключается в том, что существует некоторое противоречие между выразительными возможностями логики и сложностью вывода в ней: большие выразительные возможности логики имеют малоэффективные процедуры вывода и даже могут приводить к алгоритмической неразрешимости проблемы вывода, а слабые выразительные возможности с эффективными процедурами вывода становятся неспособными определить важные свойства концептов в данной ПО. Решение этой противоречивости является одной из самых важных проблем, которые особенно интересуют исследователей. Первые шаги и идеи на пути преодоления указанной противоречивости при разработке ДЛ были взяты из так называемых “структурно-наследственных рабочих станций” [14; 15]. Следующие идеи были заимствованы из этих работ:

а) базовыми синтаксическими блоками являются атомарные концепты (унарные предикаты), атомарные роли (бинарные базовые предикаты) и индивидуумы (константы);

б) выразительные возможности ДЛ ограничивались только использованием относительно небольшого множества конструкторов для построения более сложных концептов и ролей;

в) простейшие знания о концептах и индивидуумах можно вывести автоматически с помощью процедур выведения. В частности, отношение включения между концептами и отношения между индивидуумами играют при этом ключевую роль.

Построенная на этой основе семантика языка, которая была названа KL-One, основывалась на логике первого порядка. Это дало возможность использовать процедуры вывода в этой логике и придать точное понимание фактам, что в результате привело к формальному исследованию свойств таких языков. Первые результаты показали, что логика первого порядка достаточно выразительна, но в ней алгоритмически неразрешима проблема выполнимости отношений включения, а временная сложность не является полиномиальной [16]. Позже были найдены более менее приемлемые способы решения этих проблем. Прежде чем перейти к их обзору и задачам, которые при этом возникают, приведем основные понятия ДЛ и структуру систем представления знаний на основе ДЛ.

Языки ДЛ

01 Базовые формализмы ДЛ

Система представления знаний (СПЗ) в ДЛ включает две компоненты:

– TBox, которая содержит терминологию, то есть словарь ПО;

– ABox, которая включает утверждения о найденных индивидуумах в терминах введенной терминологии, то есть в терминах словаря ПО.

Словарь состоит из концептов, которые обозначают множества индивидуумов, и ролей, которые являются бинарными отношениями между индивидуумами. Дополнительно к атомарным концептам и ролям (именам концептов и ролей) пользователем добавляются в ДЛ более сложные описания концептов и ролей. Следовательно, TBox может использоваться для присваивания имен сложным описаниям этих объектов в ДЛ-системе. Язык, который используется для построения описаний, является характеристикой каждой ДЛ-системы и разные ДЛ-системы имеют разные языки описания. Язык описаний имеет теоретико-модельную семантику. Следовательно, утверждения в TBox и в ABox могут быть представлены некоторой формулой логики предикатов первого порядка или, в некоторых случаях, в ее незначительном расширении. ДЛ-система не только сохраняет терминологию и утверждения, но также и дополнительные сервисы, с помощью которых можно делать выводы. Типичным заданием такого типа является проверка противоречивости описаний объектов, или проверка того, является ли некоторое описание более общим, чем другое. Важной проблемой для ABox является определение, будет ли некоторое множество утверждений совместимым, то есть будет ли это множество иметь модель.

В многих использованиях систем представления знаний ДЛ встроены в более общие системы, а их компоненты взаимодействуют с другими компонентами другой системы путем использования запросов к базе знаний. Таким образом, система представления знаний с использованием ДЛ принимает вид:

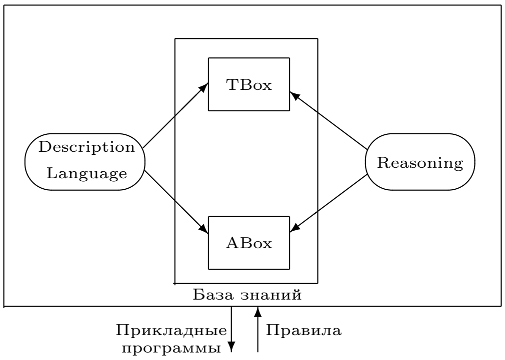

Рис 2. Архитектура системы представления знаний на основе ДЛ

02 Базовый атрибутный язык AL

Одним из основных средств ДЛ являются языки, с помощью которых представляются знания и осуществляется их обработка. Из вышесказанного следует, что изначально существует два вида объектов, которые назывались атомарными концептами и атомарными ролями. Атомарные концепты обозначаются буквами A, B, C, D, а атомарные роли символами R, S. Из атомарных концептов и ролей как элементарных описаний индуктивно строятся более сложные описания с помощью соответственных конструкторов. Поскольку конструкторы могут быть разные, то и языки описаний отличаются по типам этих конструкторов. Начальным языком описания является Атрибутный Язык (AL), основные объекты которого определяются следующим образом.

Пусть C и D – произвольные концепты. Описание базисного AL выполняется в соответствии с таким синтаксисом:

$C, D ::= A$ (атомарный концепт) | $\top$ (универсальный концепт)

| $\bot$ (пустой концепт)| $\neg A$ (отрицание атомарного концепта)

| $C \sqcap D$ (пересечение)| $\forall R.C$ (ограниченное значение)

| $\exists R.\top$ (ограниченный квантор существования).

Заметим, что в AL отрицание применяется только к атомарным концептам, а для квантора существования областью его действия является универсальный концепт.

Для демонстрации того, что можно выразить в AL, рассмотрим пример. Пусть $Person$ и $Female$ являются атомарными концептами. Тогда $Person \sqcap Female$ и $Person \sqcap \neg Female$ являются концептами в AL, которые означают, что “персона является женщиной” и что “персона не является женщиной”. Если дополнительно предположить, что $hasChild$ означает атомарную роль, то можно образовать концепты $Person \sqcap \exists hasChild.\top$ и $Person \sqcap \forall hasChild.Female$, которые описывают те персоны, которые имеют ребенка и те персоны, которые имеют детей и являются женщинами. Используя $\bot$-концепт, можно описать тех персон, которые не имеют детей, с помощью такого концепта $Person \sqcap \forall hasChild.\bot$.

Для определения формальной семантики AL, рассмотрим интерпретацию I над непустой областью Δ^I (область интерпретации), которая является функцией, которая присваивает каждому атомарному концепту A некоторое множество $A^I \subseteq \Delta^I$ и каждой атомарной роле R бинарное отношение $R^I \subseteq \Delta^I \times \Delta^I$. Функция интерпретации расширяется на описание концептов с помощью такого индуктивного определения:

– $\top^I = \Delta^I$,

– $\bot^I = \emptyset$,

– $(\neg A)^I = \Delta^I \setminus A^I$,

– $(C \sqcap D)^I = C^I \cap D^I$,

– $(\forall R.C)^I = \{a \in \Delta^I | \forall b\ (a,b) \in R^I \to b \in C^I\}$,

– $(\exists R.\top)^I = \{a \in \Delta^I | \exists b\ (a,b) \in R^I\}$.

Два концепта C и D считаются эквивалентными и это записывается как $C \equiv D$, если $C^I = D^I$ для произвольной интерпретации I. Например, из определения семантики концептов следует, что $\forall hasChild.Female \sqcap \forall hasChild.Student$ и $\forall hasChild(Female \sqcap Student)$ эквивалентны.

03 Семейство дескриптивных языков

Более выразительные дескриптивные языки можно получить, если ввести дополнительные конструкторы в язык AL. Такими конструкторами выступают:

– Объединение концептов (обозначение $\mathcal{U}$) $C \sqcup D$ с интерпретацией

$$(C \sqcup D)^I = C^I \cup D^I,$$

– Полный квантор существования (обозначение $\mathcal{E}$) $\exists R.C$ с интерпретацией

$$(\exists R.C)^I = \{a \in \Delta^I | \exists b.(a,b) \in R^I \land b \in C^I\},$$

– Числовое ограничение (обозначение $\mathcal{N}$) $\geq nR$ (не меньше) и $\leq nR$ (не больше) с интерпретациями

$$(\geq nR)^I = \{a \in \Delta^I || \{b | (a,b) \in R^I\}| \geq n\},\ \ (\leq nR)^I = \{a \in \Delta^I || \{b | (a,b) \in R^I\}| \leq n\}$$

соответственно,

– Отрицание произвольного концепта (обозначение $\mathcal{C}$) $\neg C$ с интерпретацией

$$(\neg C)^I = \Delta^I \setminus C^I.$$

С помощью введенных конструкторов можно выразить, например, описание тех персон, которые имеют не больше одного ребенка, и тех женщин, которые имеют не меньше трех детей:

$$Person \sqcap (\leq 1 hasChild \sqcup (\geq 3 hasChild \sqcap \exists hasChild.Female)).$$

Расширение AL с помощью произвольного подмножества введенных конструкторов порождает соответствующие AL-языки. Каждый из этих языков обозначается аббревиатурой вида

$$AL[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}].$$

Например, $AL\mathcal{E}\mathcal{N}$ является AL-языком, в котором присутствуют конструкторы $\mathcal{E}$ и $\mathcal{N}$.

С семантической точки зрения не все эти языки являются разными, поскольку имеет место некоторая семантическая эквивалентность. Например, $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ и $\exists R.C \equiv \neg\forall R.\neg C$. Следовательно, объединение и полный квантор существования можно выразить, используя отрицание. Наоборот, комбинация объединения и полного квантора существования дает возможность выразить отрицание концептов через их эквивалентную нормальную негативную форму (форма, в которой отрицания стоят только перед атомарными концептами). Поэтому без ограничения общности считается, что объединение и полный квантор существования присутствуют в каждом AL-языке, который включает отрицание и наоборот. Отсюда следует, что все AL-языки могут быть записаны только с помощью $\mathcal{U}, \mathcal{E}, \mathcal{N}$. Несложно понять, что все восемь языков, которые можно получить таким образом, попарно не эквивалентны. Далее аббревиатуры $AL\mathcal{C}$, $AL\mathcal{U}\mathcal{E}$, $AL\mathcal{C}\mathcal{N}$ и $AL\mathcal{U}\mathcal{E}\mathcal{N}$ означают один и тот же язык.

04 AL-языки как фрагменты логики предикатов

Семантические понятия AL-языков показывают, что эти языки являются фрагментами логики предикатов первого порядка. На основе того, что интерпретация I присваивает каждому атомарному концепту и роли соответствующие унарные и бинарные отношения на множестве Δ^I, то на атомарные концепты и роли можно смотреть как на соответствующие унарные и бинарные предикаты. Тогда каждый концепт C можно транслировать в формулу логики предикатов $\psi_C(x)$ с одной свободной переменной x такой, что для произвольной интерпретации I множество элементов из Δ^I, которое выполняет $\psi_C(x)$, в точности совпадает с множеством C^I. При этом атомарный концепт A транслируется в формулу $A(x)$, а конструкторы пересечения, объединения и отрицания транслируются в логическую конъюнкцию, дизъюнкцию и отрицание соответственно. Если C уже странслированное в $\psi_C(x)$ и R – атомарная роль, то кванторы существования и общности выражаются формулами

$$\psi_{\exists R.C}(y) = \exists x.R(y,x) \wedge \psi_C(x),$$
$$\psi_{\forall R.C}(y) = \forall x.R(y,x) \to \psi_C(x),$$

где y – новая переменная, а числовое ограничение значений выражаются формулами

$$\psi_{\geq nR}(x) = \exists y_1, \ldots, \exists y_n . R(x, y_1) \wedge \ldots \wedge R(x, y_n) \wedge \bigwedge_{i<j} y_i \neq y_j,$$

$$\psi_{\leq nR}(x) = \forall y_1, \ldots, \forall y_{n+1} . R(x, y_1) \wedge \ldots \wedge R(x, y_{n+1}) \wedge \bigvee_{i<j} y_i = y_j.$$

Следует заметить, что предикат равенства "=" необходим, поскольку с его помощью выражаются числовые ограничения. Но так как концепты транслируются в логику предикатов, то вводить специальный синтаксис нет необходимости.

Терминологии

Из сказанного следуют способы построения сложных описаний концептов для представления классов объектов. Рассмотрим терминологические аксиомы, которые выражают связи между концептами и ролями. Выбирая определения как специфические аксиомы и идентифицируя терминологию как множество определений, мы можем вводить атомарные концепты как аббревиатуры или имена для сложных концептов. Если определения в терминологии создают цикл, то к этой ситуации адаптируются семантики неподвижной точки с целью сделать терминологии однозначными. Будут рассмотрены также типы терминологий, для которых существуют модели семантик неподвижной точки.

05 Терминологические аксиомы

В наиболее общем виде терминологические аксиомы имеют вид

$$C \sqsubseteq D \ (R \sqsubseteq S) \text{ или } C \equiv D \ (R \equiv S),$$

где C, D – концепты (R, S – роли). Аксиомы первого типа называются включениями, а аксиомы второго типа – равенствами. Далее будут рассматриваться только аксиомы для концептов, потому что для ролей они аналогичны.

Интерпретация I выполняет включение $C \sqsubseteq D$ тогда и только тогда, когда $C^I \subseteq D^I$ и выполняет равенство $C \equiv D$ тогда и только тогда, когда $C^I = D^I$. Если $\mathcal{T}$ терминология с множеством аксиом, то интерпретация I выполняет $\mathcal{T}$ тогда и только тогда, когда I выполняет каждую аксиому терминологии $\mathcal{T}$. Если интерпретация I выполняет аксиомы $\mathcal{T}$, то говорят, что I является моделью для $\mathcal{T}$. Две аксиомы или два множества аксиом эквивалентны, если они имеют одну и ту же модель.

06 Определения

Равенство, в которой левая часть является атомарным концептом, называется определением. Определение используется для введения символических имен для сложных описаний. Например, с помощью аксиомы

$$Mother \equiv Women \sqcap \exists hasChild.Person$$

мы достаем описание правой части для имени $Mother$. Символические имена могут быть использованы как аббревиатуры в других описаниях. Если, например, имеем определение $Father$, аналогичное $Mother$, то можно определить имя $Parent$ у виде

$$Parent \equiv Mother \sqcup Father.$$

Множество определений может быть неоднозначным. Неоднозначность влияет на однозначность терминологии.

Конечное множество определений $\mathcal{T}$ называется терминологией или TBox, если нет ни одного имени, которое определяется больше одного раза, то есть когда для каждого атомарного концепта A существует не больше одной аксиомы в $\mathcal{T}$, левая часть которой – концепт A. На рис. 3 показана терминология, которая описывается семейные отношения.

Пример 1. Терминология (TBox) с концептами про семейные отношения:

$$
\begin{array}{rcl}
\text{Woman} & \equiv & \text{Person} \sqcap \text{Female} \\
\text{Man} & \equiv & \text{Person} \sqcap \neg\ \text{Female} \\
\text{Mother} & \equiv & \text{Women} \sqcap \exists\ \text{hasChild.Person} \\
\text{Father} & \equiv & \text{Man} \sqcap \exists\ \text{hasChild.Person} \\
\text{Parent} & \equiv & \text{Father} \sqcup \text{Mother} \\
\text{Grandmother} & \equiv & \text{Mother} \sqcap \exists\ \text{hasChild.Parent} \\
\text{Grandfather} & \equiv & \text{Father} \sqcap \exists\ \text{hasChild.Parent} \\
\text{MotherWithManyChildren} & \equiv & \text{Mother} \sqcap \geq 3\ \text{hasChild} \\
\text{MotherWithout Daughter} & \equiv & \text{Mother} \sqcap \forall\ \text{hasChild.}\neg\ \text{Women} \\
\text{Wife} & \equiv & \text{Women} \sqcap \exists\ \text{hasHusband.Man}
\end{array}
$$

Рис. 3. Терминология (TBox) для семейных отношений

Пусть $\mathcal{T}$ – некоторая терминология. Атомарные символы, которые входят в $\mathcal{T}$, делятся на два подмножества:

– символы-имена $\mathcal{N}_{\mathcal{T}}$, которые входят только в левые части аксиом;

– базовые символы $\mathcal{B}_{\mathcal{T}}$, которые входят только в правые части аксиом.

Символы-имена часто называют определенными концептами, а базовые символы – примитивными концептами. Таким образом, терминология уточняет определение имен концептов в терминах базовых концептов.

Базовой интерпретацией терминологии $\mathcal{T}$ называется интерпретация, которая интерпретирует только базовые символы. Пусть $\mathcal{J}$ – базовая интерпретация. Интерпретация $\mathcal{I}$, которая интерпретирует символы-имена, называется расширением $\mathcal{J}$, если она имеет ту же самую область интерпретации, что и $\mathcal{J}$ (то есть $\Delta^I = \Delta^J$) и согласуется с $\mathcal{J}$ для базовых символов.

Терминология $\mathcal{T}$ называется дефиниторной, если каждая базовая интерпретация имеет единственное расширение, что образует модель $\mathcal{T}$. Другими словами, если нам известна интерпретация базовых символов и $\mathcal{T}$ – дефиниторная, то понимание символов-имен полностью детерминированное (однозначное). Очевидно, что когда терминология дефиниторная, то произвольная терминология, которая ей эквивалентна, тоже дефиниторная.

Дефиниторность терминологии зависит от того, будет или нет множество определений цикличным. Например, терминология, которая состоит только из одной аксиомы

$$Human' \equiv Animal \sqcap \forall hasParent.Human', \tag{1}$$

имеет цикл, поскольку каждый $Human'$ определяется в терминах $Human'$.

Пусть A и B – атомарные концепты терминологии $\mathcal{T}$. Говорят, что концепт A непосредственно использует концепт B, если B встречается в правой части определения A. Транзитивное замыкание отношения непосредственного использования называется отношением использования.

Терминология $\mathcal{T}$ включает цикл тогда и только тогда, когда существует атомарный концепт в $\mathcal{T}$, который использует сам себя. В противном случае терминология $\mathcal{T}$ называется ациклической.

Если терминология имеет цикл, то она может иметь много расширений. Например, если рассмотреть терминологию с единственной аксиомой (1), то $Human'$ – это символ-имя, а $Animal$ и $hasParent$ являются базовыми символами. Интерпретация, которая $hasParent$ относит каждого $Animal$ к его предку, может иметь несколько возможных расширений интерпретации для $Human'$, удовлетворяющих аксиому (1). $Human'$ может интерпретироваться как множество всех зверей, как некоторый вид или какое-нибудь другое множество зверей со свойством, что каждый зверь имеет предка.

Если же $\mathcal{T}$ ациклическая, то она дефиниторная. Действительно, мы можем с помощью подстановок итеративным способом заменить в $\mathcal{T}$ все вхождения имен в правые части определений на базовые имена. Такой процесс всегда заканчивается в связи с ацикличностью $\mathcal{T}$ и в результате получаем терминологию $\mathcal{T}'$, которая называется расширением терминологии $\mathcal{T}$.

Следует отметить, что размеры расширения могут быть экспоненциальными в сравнении с размерами оригинальной терминологии [17].

Пример 2. Терминология (TBox) из примера 1 ациклическая и ее расширение получает вид:

$$
\begin{array}{rcl}
\text{Woman} & \equiv & \text{Person} \sqcap \text{Female} \\
\text{Man} & \equiv & \text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}) \\
\text{Mother} & \equiv & (\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasChild.Person} \\
\text{Father} & \equiv & (\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists\ \text{hasChild.Person} \\
\text{Parent} & \equiv & ((\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists\ \text{hasChild.Person}) \\
 & & \sqcup\ ((\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasChild.Person}) \\
\text{Grandmother} & \equiv & ((\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasChild.Person}) \\
 & & \sqcap\ \exists\ \text{hasChild.}(((\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female})) \sqcap \exists\ \text{hasChild.Person}) \\
 & & \sqcup\ (\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasChild.Person})) \\
\text{MotherWithManyChildren} & \equiv & ((\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasChild.Person}) \sqcap \geq 3\ \text{hasChild} \\
\text{MotherWithout Daughter} & \equiv & ((\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasChild.Person}) \sqcap \forall\ \text{hasChild.}(\neg(\text{Person} \sqcap \text{Female})) \\
\text{Wife} & \equiv & (\text{Person} \sqcap \text{Female}) \sqcap \exists\ \text{hasHusband.}(\text{Person} \sqcap \neg(\text{Person} \sqcap \text{Female}))
\end{array}
$$

Рис. 4. Расширение TBox для семейных отношений

Имеет место

Утверждение 1.. Пусть терминология $\mathcal{T}$ – ациклическая и $\mathcal{T}'$ ее расширение. Тогда

а) $\mathcal{T}$ и $\mathcal{T}'$ имеют одинаковое множество базовых символов;

б) $\mathcal{T}$ и $\mathcal{T}'$ эквивалентны;

в) $\mathcal{T}$ и $\mathcal{T}'$ дефиниторные.

Теорема 1.. Каждая дефиниторная $\mathcal{ALC}$ терминология эквивалентна ациклической терминологии.

Известно, что $\mathcal{ALC}$-язык является вариантом пропозициональной модальной логики K_n [18].

07 Семантика неподвижных точек для терминологических циклов

Рассмотрение семантик неподвижной точки мотивируется тем фактом, что существуют ситуации, где циклические определения могут быть содержательными и это содержание описывается семантикой или наименьшей неподвижной точки, или наибольшей неподвижной точки.

Пример 3. Пусть необходимо описать концепт “мужчина, который имеет наследников только мужского пола” (сокращение $Momd$). В частности, такой мужчина имеет только сыновей (обозначим его Mos). Mos можно определить без циклов

$$Mos \equiv Man \sqcap \forall hasChild.Man.$$

Для $Momd$ мы хотим создать новые утверждения с помощью транзитивного замыкания роли $hasChild$. Это принимает такой вид: мужчина, который имеет наследников только мужчин, сам является таким мужчиной, а все его сыновья тоже являются мужчинами, которые имеют наследниками только мужчин. Следовательно,

$$Momd \equiv Man \sqcap \forall hasChild.Momd. \tag{2}$$

Для достижения однозначного понимания, необходимо обосновать это определение с помощью подходящей семантики неподвижной точки.

Далее будет показано, что семантика наибольшей неподвижной точки обосновывает ситуацию такого типа.

Пример 4. Пусть имеем множество объектов, которые называются $Trees$ и бинарное отношение $has\text{-}branch$ между объектами, которые порождают их поддеревья. Тогда бинарные деревья являются такими, что имеют не больше двух поддеревьев и которые, в свою очередь, тоже являются бинарными деревьями. Тогда определение $BinaryTree$ принимает вид:

$$BinaryTree \equiv Tree \sqcap \leq 2has\text{-}branch \sqcap \forall has\text{-}branch.BinaryTree.$$

Как и в случае с $Momd$, семантика неподвижной точки дает желаемое понимание, но в данном случае – это семантика наименьшей неподвижной точки.

Определение семантик неподвижной точки. В терминологии $\mathcal{T}$ каждый символ A входит только один раз в левую часть аксиомы $A \equiv C$. Поэтому на $\mathcal{T}$ можно смотреть как на отображение, которое ассоциирует с символом-именем A его концепт $\mathcal{T}(A) = C$. В этих обозначениях интерпретация $\mathcal{I}$ являет собой модель терминологии $\mathcal{T}$ тогда и только тогда, когда $A^I = (\mathcal{T}(A))^I$.

Рассмотрим семейство этих отображений, для которых интерпретация является моделью терминологии $\mathcal{T}$ тогда и только тогда, когда она является неподвижной точкой некоторого отображения из этого семейства.

Пусть $\mathcal{T}$ – некоторая терминология и $\mathcal{J}$ – фиксированная базовая интерпретация. Пусть $Ext_{\mathcal{J}}$ означает расширение интерпретации $\mathcal{J}$ до интерпретации $\mathcal{T}_{\mathcal{J}}(\mathcal{I})$, которое определяется как

$$A^{\mathcal{T}_{\mathcal{J}}(\mathcal{I})} = (\mathcal{T}(A))^{\mathcal{J}}$$

для каждого символа-имени A.

Интерпретация $\mathcal{I}$ является неподвижной точкой $\mathcal{T}_{\mathcal{J}}$ тогда и только тогда, когда $\mathcal{I} = \mathcal{T}_{\mathcal{J}}(\mathcal{I})$, то есть тогда и только тогда, когда $A^{I} = A^{\mathcal{T}_{\mathcal{J}}(I)}$ для всех символов-имен. Это означает, что для каждого определения $A \equiv C$ в $\mathcal{T}$ имеет место $A^{I} = A^{\mathcal{T}_{\mathcal{J}}(I)} = (\mathcal{T}(A))^{I} = C^{I}$, а это означает, что $\mathcal{I}$ является моделью для $\mathcal{T}$.

Утверждение 2.. Пусть $\mathcal{T}$ – терминология, $\mathcal{I}$ – интерпретация и $\mathcal{J}$ – сужение $\mathcal{I}$ к базовым символам терминологии $\mathcal{T}$. Тогда $\mathcal{I}$ является моделью $\mathcal{T}$ тогда и только тогда, когда $\mathcal{I}$ неподвижная точка отображения $\mathcal{T}_{\mathcal{J}}$.

С этого утверждения следует, что терминология $\mathcal{T}$ дефиниторная тогда и только тогда, когда каждая базовая интерпретация $\mathcal{J}$ имеет единственное расширение, которое является неподвижной точкой $\mathcal{T}_{\mathcal{J}}$.

Пример 5. Рассмотрим пример терминологии, который показывает почему циклическая терминология не является дефиниторной. Это терминология $\mathcal{T}^{Momd}$, которая включает единственную аксиому (2). Пусть базовой интерпретацией $\mathcal{J}$ является такая интерпретация:

$\Delta^{\mathcal{J}} = \{Charles_1, Charles_2, \ldots\} \cup \{James_1, \ldots, James_{Last}\}$,

$Man^{\mathcal{J}} = \Delta^{\mathcal{J}}$,

$hasChild = \{Charles_i, Charles_{i+1} | i \geq 1\} \cup \{James_i, James_{i+1} | 1 \leq i < Last\}$.

Это означает, что династия $Charles$ живая, в то время как династия $James$ обрывается на $James_{Last}$.

Найдем неподвижные точки $\mathcal{T}_{\mathcal{J}}^{Momd}$. Отметим, что бездетный индивидуум, то есть без элементов у $hasChild$, всегда будет присутствовать в интерпретации $\forall hasChild$ независимо от того, как интерпретируется $Momd$. Следовательно, если $\mathcal{I}_1$ – неподвижная точка расширения $\mathcal{J}$, то $James_{Last}$ буде элементом в $(\forall hasChild)^{\mathcal{I}_1}$, а значит и элементом $Momd^{\mathcal{I}_1}$. Отсюда получаем, что каждый $James$ является $Momd$. Пусть $\mathcal{I}$ будет расширением $\mathcal{J}$ таким, что $Momd^{\mathcal{I}_1}$ охватывает только династию $James$. Тогда легко увидеть, что $\mathcal{I}$ является неподвижной точкой. Если к династии $James$ добавляется кто-то из династии $Charles$ как $Momd$, то все члены династии $Charles$ до и после этого члена должны принадлежать концепту $Momd$. Снова легко понять, что расширение $\mathcal{I}_2$, которое интерпретирует $Momd$, является неподвижной точкой и других неподвижных точек нет.

Для того чтобы дать дефиниторный вид циклической терминологии $\mathcal{T}$, необходимо определить отдельную неподвижную точку, если этих точек несколько. Для этого введем частичный порядок "$\preceq$" на расширениях базовой интерпретации $\mathcal{J}$:

$$\mathcal{I} \preceq \mathcal{I}' \Leftrightarrow A^{\mathcal{I}} \subseteq A^{\mathcal{I}'}$$

для каждого символа-имени A в $\mathcal{T}$. В вышеописанном примере $Momd$ является только символом-именем. Поскольку $Momd^{\mathcal{I}} \subseteq Momd^{\mathcal{I}'}$, то $\mathcal{I} \preceq \mathcal{I}'$.

Неподвижная точка $\mathcal{I}$ отображения $\mathcal{T}_{\mathcal{J}}$ называется наименьшей неподвижной точкой, если $\mathcal{I} \preceq \mathcal{I}'$ для каждой другой неподвижной точки $\mathcal{I}'$. Интерпретация $\mathcal{I}$ называется наименьшей неподвижной точкой терминологии $\mathcal{T}$, если $\mathcal{I}$ наименьшая неподвижная точка $\mathcal{T}_{\mathcal{J}}$ для некоторой базовой модели $\mathcal{J}$.

В рамках семантики наименьшей неподвижной точки допускаются только модели наименьшей неподвижной точки терминологии $\mathcal{T}$ как единственные интерпретации. Наибольшая неподвижная точка модели и семантики определяются аналогично. В примере $Momd$ интерпретация $\mathcal{I}_1$ является наименьшей неподвижной точкой, а интерпретация $\mathcal{I}_2$ – наибольшей неподвижной точкой отображения $\mathcal{T}_{\mathcal{J}}$.

Следует заметить, что модели неподвижной точки не всегда существуют. Например, пусть имеем терминологию с единственной аксиомой

$$A \equiv \neg A. \tag{3}$$

Если $\mathcal{I}$ – некоторая модель этой аксиомы, то $A^I = \Delta^I \backslash \Delta^I = \emptyset$, что абсурдно. Аксиома (3) не имеет никакой модели ни наименьшей ни наибольшей неподвижной точки.

Существуют случаи моделей неподвижной точки, для которых невозможно существование ни наименьшей ни наибольшей неподвижной точек. Например, рассмотрим терминологию $\mathcal{T}$ с единственной аксиомой

$$A \equiv \forall R.\neg A. \tag{4}$$

Пусть $\mathcal{J}$ – базовая интерпретация $\mathcal{T}$, где $\Delta^{\mathcal{J}} = \{a, b\}$ і $R^{\mathcal{J}} = \{(a, b), (b, a)\}$. Тогда существует два расширения $\mathcal{I}_1$ и $\mathcal{I}_2$ интерпретации $\mathcal{J}$, которые определяются с помощью $A^{\mathcal{I}_1} = \{a\}$, $A^{\mathcal{I}_2} = \{b\}$. Но эти расширения не сравниваются по отношению “$\preceq$“.

Для обоснования семантик неподвижной точки терминологии напомним некоторые понятия из теории полных решеток [20].

Полной решеткой называется алгебра, носитель которой является частично упорядоченное множество с двумя операциями пересечения и объединения, которые также называются соответственно взятием наибольшей нижней и наименьшей верхней грани, в которой существует единичный элемент и каждое непустое подмножество имеет наименьшую верхнюю грань.

Множество интерпретаций $Ext_{\mathcal{J}}$ относительно частичного порядка $\preceq$ является полной решеткой. Действительно, для семейства интерпретаций $\mathcal{I}_i$, $i \in N$, определим $\mathcal{I}_0 = \bigsqcup_{i \in N} \mathcal{I}_i$ как поэлементное объединение $\mathcal{I}_i$-х, то есть для каждого символа-имени A будем иметь $A^{\mathcal{I}_0} = \bigcup_{i \in N} A^{\mathcal{I}_i}$. Тогда $\mathcal{I}_0$ становится наименьшей верхней гранью для $\mathcal{I}_i$-х, которая показывает, что $(Ext_{\mathcal{J}}, \preceq)$ является полной решеткой.

Функция $f : L \to L$ полной решетки L в себя называется монотонной, если $f(x) \preceq f(y) \Leftrightarrow x \preceq y$. Имеет место

Теорема 2 (Тарский). . Для монотонной функции на полной решетке существует непустое множество неподвижных точек и это множество тоже является полной решеткой [19].

Следовательно, для монотонной функции в полной решетке существует ее наименьшая и наибольшая неподвижные точки.

Определение 1.. Терминология $\mathcal{T}$ называется монотонной, если отображение $\mathcal{T}_{\mathcal{J}}$ монотонное для произвольной базовой интерпретации $\mathcal{J}$.

Из теоремы Тарского следует существования для монотонной терминологии наибольшей и наименьшей неподвижных точек.

Утверждение 3.. Если $\mathcal{T}$ – монотонная терминология и $\mathcal{J}$ – базовая интерпретация, то существуют расширения $\mathcal{J}$, которые являются моделями наименьшей верхней неподвижной и наибольшей нижней неподвижной точки терминологии $\mathcal{T}$ соответственно [10].

Для того чтобы найти синтаксический критерий монотонности, рассмотрим операторы AL-языка, которые использовались для концептов, а именно: $\sqcap, \sqcup, \neg, \forall, \exists$. Пусть эти операторы интерпретируются на области Δ. Им соответствуют теоретико-множественные операции пересечения, объединения, дополнения с теми самыми названиями на области Δ. Поскольку $\forall$ и $\exists$ применяются к ролям и концептам, то введем для бинарного отношения r на Δ унарную операцию A_r, которая отображает некоторое подмножество $S \subseteq \Delta$ в

$$A_r(S) = \{a \in \Delta | \forall b.\ (a, b) \in r \to b \in S\}$$

и унарную операцию E_r, которая отображает множество $S \subseteq \Delta$ в

$$E_r(S) = \{a \in \Delta | \exists b.\ (a, b) \in r \wedge b \in S\}.$$

Отсюда следует, что эти операции, за исключением дополнения, являются монотонными в том смысле, что когда они применяются к большим аргументам, то значение результата тоже будет большим. Это очевидно для пересечения и объединения, а для A_r и E_r нетрудно убедиться в том, что когда $S \subseteq S'$, то $A_r(S) \subseteq A_r(S')$ и $E_r(S) \subseteq E_r(S')$. К сожалению, подобные аналогии для отрицания не выполняются.

Определение 2.. Терминология называется $\neg$-свободной, если в ней нет ни одного знака отрицания.

Утверждение 4.. Произвольная $\neg$-свободная $\mathcal{ALCN}$-терминология монотонна.

Расширим это достаточное условие, анализируя, какие из знаков отрицания не влияют на монотонность. Отрицание может влиять на монотонность, если в его области действия находится некоторой символ-имя. Однако отрицание может нейтрализоваться другим отрицанием, если оно входит в область действия другого отрицания. Отсюда следует такое

Определение 3.. Концепт C называется синтаксически монотонным, если каждый символ-имя в C входит в область действия парного числа знаков отрицания.

Терминология называется синтаксически монотонной, если каждый концепт в правой части равенства является синтаксически монотонным.

Утверждение 5.. Произвольная синтаксически монотонная $\mathcal{ALCN}$-терминология монотонна.

08 Терминологии с аксиомами включения

Некоторые концепты невозможно определить в полном объеме. В таких случаях можно только формулировать необходимые условия принадлежности концепта к некоторому множеству, используя включения. Включения, левая часть которого атомарная, называется специализацией.

Например, когда инженер по знаниям считает, что определение “women” в нашем примере TBox (рис. 3) неисполнимое и он понимает, что не в состоянии полностью определить концепт “women”, то он потребует, чтобы все *woman* были *person* с помощью специализации

$$Woman \sqsubseteq Person. \tag{5}$$

Если такая специализация в терминологии допустима, то терминология теряет дефиниторность даже, если она ациклическая. Множество аксиом терминологии $\mathcal{T}$ называется обобщенной терминологией, если левые части ее аксиом являются атомарными концептами и для каждого атомарного концепта существует не более одной аксиомы, где он входит в левую часть.

Выполним преобразование обобщенной терминологии $\mathcal{T}$ в некоторую регулярную терминологию $\bar{\mathcal{T}}$, которая включает только такие определения, что $\bar{\mathcal{T}}$ эквивалентна $\mathcal{T}$ в некотором определенным смысле. В результате получим $\bar{\mathcal{T}}$ из $\mathcal{T}$ путем выбора для каждой специализации $A \sqsubseteq C$ в $\mathcal{T}$ новый базовый символ $\bar{A}$ и заменой специализации $A \sqsubseteq C$ на определение $A \equiv \bar{A} \sqcap C$. Терминология $\bar{\mathcal{T}}$ называется нормализацией $\mathcal{T}$.

Если TBox включает специализацию (5), то нормализация включает определение

$$Woman \equiv \overline{Woman} \sqcap Person.$$

Интуитивно дополнительный базовый символ $\overline{Woman}$ для равенства призван выделить женщин среди персон. Результат нормализации в TBox с определением для $Woman$ является подобным тому, как это делалось для $Female$ в TBox.

Утверждение 6.. Пусть $\mathcal{T}$ обобщенная терминология и $\bar{\mathcal{T}}$ ее нормализация. Тогда

а) каждая модель $\bar{\mathcal{T}}$ является моделью и для $\mathcal{T}$;

б) для каждой модели $\mathcal{I}$ терминологии $\mathcal{T}$ существует модель $\bar{\mathcal{I}}$ терминологии $\bar{\mathcal{T}}$, которая имеет ту же самую область интерпретации, что и $\mathcal{I}$ и согласуется с $\mathcal{I}$ на атомарных концептах и ролях $\mathcal{T}$.

Доказательство. Первый пункт выполняется в связи с тем, что модель $\bar{\mathcal{I}}$ терминологии $\bar{\mathcal{T}}$ удовлетворяет равенству $A^{\bar{\mathcal{I}}} = (\bar{A} \sqcap C)^{\bar{\mathcal{I}}} = \overline{A}^{\mathcal{I}} \cap C^{\mathcal{I}}$, а это означает что $A^{\bar{\mathcal{I}}} \subseteq C^{\bar{\mathcal{I}}}$.

Наоборот, если $\mathcal{I}$ модель $\mathcal{T}$, то расширение $\bar{\mathcal{I}}$ терминологии $\mathcal{T}$, определенное как $\overline{A}^{\bar{\mathcal{I}}} = A^{\mathcal{I}}$, является моделью $\bar{\mathcal{T}}$, поскольку $A^{I} \subseteq C^{I}$, а отсюда следует, что $A^{I} = \overline{A}^{I} \cap C^{I} = A^{\bar{\mathcal{I}}} \cap C^{\bar{\mathcal{I}}}$. Следовательно, $\bar{\mathcal{I}}$ выполняет $A = \bar{A} \sqcap C$. ■

Отсюда следует, что аксиомы включения не влияют на выразительность терминологии TBox. Но, с практической точки зрения, они являются удобным средством введения термов в терминологию, которую невозможно полностью определить.

Описание миров

Другая компонента базы знаний является ABox – описание области интерпретации (мир, в котором интерпретируются концепты).

09 Утверждения об индивидуумах

ABox описывает свойства объектов области интерпретации в терминах концептов и ролей. Некоторые атомарные концепты и роли в ABox могут быть определены как имена TBox. В ABox вводятся индивидуумы путем предоставления им имен и описания свойств этих индивидуумов. Индивидуумы обозначаются $a, b, c, \ldots$. Используя концепты C и роли R, можно создать в ABox свойства таких типов:

$$C(a) \text{ i } R(b, c).$$

Первый тип называется свойством концепта, которое означает, что a принадлежит C. Второй тип называется свойством роли, которое означает, что c является наполнителем роли R для b. Например, если PETER, PAUL и MARY имена индивидуумов, то FATHER(PETER) означает, что PETER является отцом и HasChild(MARY,PETER) означает, что Peter является ребенком Mary. Следовательно, ABox $\mathcal{A}$ является конечным множеством таких утверждений. На рис. 5 показан пример ABox.

MatherWithoutDaughter(MARY)	Father(PETER)
hasChild(MARY,PETER)	hasChild(PETER,HARRY)
hasChild(MARY,PAUL)	

Рис. 5. Описание ABox

С точки зрения пользователя, ABox выглядит как объект реляционной базы данных, который состоит только из унарных и бинарных отношений. Однако в классических базах данных отношения замкнуты в рамках данной предметной области, в то время как в ABox база знаний является открытой для данной предметной области, поскольку нормализованная база знаний является системой, в которой нет предположений о полноте ее знаний. Кроме этого, TBox связывает семантическими отношениями концепты и роли в ABox, что не имеет аналога в семантиках баз данных.

Семантика ABox определяется путем расширения на индивидуальные имена. Следовательно, интерпретация $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ отображает не только атомарные концепты и роли во множество отношений, а дополнительно отображает каждое индивидуальное имя a в некоторый концепт $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. При этом считается, что разные индивидуальные имена обозначают разные объекты. Поэтому это отображение должно соответствовать уникальным именам, то есть если a и b разные имена, то $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. Интерпретация $\mathcal{I}$ выполняет свойство концепта $C(a)$, если $a^{\mathcal{I}} \in C^{\mathcal{I}}$ и выполняет свойство роли $R(a,b)$, если $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Интерпретация $\mathcal{I}$ выполняет ABox $\mathcal{A}$, если она выполняет все свойства в $\mathcal{A}$. В этом случае говорят, что $\mathcal{I}$ является моделью для ABox. Наконец, $\mathcal{I}$ выполняет свойство α или ABox $\mathcal{A}$ относительно TBox $\mathcal{T}$, если она является моделью не только для α или ABox, а и для TBox $\mathcal{T}$. Следовательно, модель $\mathcal{A}$ і $\mathcal{T}$ является интерпретацией конкретной ПО, концепты которой интерпретируются как подмножества этой области в терминах концептов и ролей из ABox.

Для индивидуальных имен в языках ДЛ допускается конструктор $\{a_1, \ldots, a_n\}$, где a_i – индивидуальное имя с интерпретацией $\{a_1, \ldots, a_n\}^{\mathcal{I}} = \{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$.

Вторым конструктором, который включает индивидуальные имена, является “fills”-конструктор для роли R. Семантика этого конструктора имеет вид:

$$(R : a)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | (d, a^{\mathcal{I}}) \in R^{\mathcal{I}}\},$$

т. е. $(R : a)$ означает множество объектов, для которого a является наполнителем роли R. Оба конструкторы не добавляют ничего нового, поскольку $(R : a)$ и $\exists R.\{a\}$ эквивалентны.

Проблемы вывода в ДЛ

Системы представления знаний, которые основываются на ДЛ, могут осуществляют разного рода логические выведения. Как отмечалось выше, ДЛ с заданными ABox и TBox имеют семантики, которые делают их эквивалентными множеству аксиом логики предикатов первого порядка. Следовательно, в ДЛ можно выполнять выведения новых фактов. Например, с TBox на рис. 3 и ABox з рис. 5 можно вывести, что $Mary$ является бабушкой, поскольку этот факт явно не фигурирует в TBox. Кроме логики первого порядка в ДЛ допустимы и другие виды выведения. Их делят на выводы только для концептов, выводы в ABox, выводы в TBox и выводы в TBox и ABox вместе.

010 Вывод для концептов

В процессе моделирования конкретной ПО строится терминология, например $\mathcal{T}$, путем определения новых концептов, возможно, в терминах других концептов, которые были определены ранее. В этом процессе важным является идентификация того, будут ли нововведенные концепты совместимыми или они будут противоречить уже введенным концептам. С логической точки зрения, концепт непротиворечив, если существует некоторая интерпретация, которая удовлетворяет аксиомы $\mathcal{T}$ (т. е., является моделью $\mathcal{T}$). Концепт с таким свойством называется выполнимым относительно терминологии $\mathcal{T}$ и невыполнимым в противоположном случае.

Проверка выполнимости концептов является ключевой задачей для процесса вывода. Например, для того чтобы проверить корректность модели ПО или оптимизировать запросы, которые формулируются в виде концептов, необходимо знать, что некоторый концепт является более общим, чем другой. Эта задача называется проблемой включения (subsumption problem). Другими интересными отношениями между концептами являются эквивалентность и попарное непересечение их подмножеств (disjointness). Эти свойства имеют такие формальные определения.

Пусть $\mathcal{T}$ – терминология (TBox). Тогда

Выполнимость: Концепт C называется выполнимым в терминологии $\mathcal{T}$, если существует модель $\mathcal{I}$ терминологии $\mathcal{T}$ такая, что $C^{\mathcal{I}} \neq \emptyset$. В этом случае говорят, что $\mathcal{I}$ является моделью C.

Включение: Концепт C включается в концепт D в терминологии $\mathcal{T}$, если $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ для каждой интерпретации $\mathcal{I}$ терминологии $\mathcal{T}$. Это обозначается как $C \subseteq_{\mathcal{T}} D$ или $\mathcal{T} \models C \sqsubseteq D$.

Эквивалентность: Концепты C и D эквивалентны в терминологии $\mathcal{T}$, если $C^{\mathcal{I}} = D^{\mathcal{I}}$ для каждой интерпретации $\mathcal{I}$ терминологии $\mathcal{T}$. Это обозначается как $C \models_{\mathcal{T}} D$ или $\mathcal{T} \models C \equiv D$.

Непересечение: Концепты C и D не пересекаются в терминологии $\mathcal{T}$, если $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ для каждой интерпретации $\mathcal{I}$ терминологии $\mathcal{T}$.

Пример 6. В терминологии TBox з рис. 3 $Person$ включает $Woman$, $Woman$ и $Parent$ оба включают $Mother$, а $Mother$ включает $Grandmother$, $Woman$ и Man, $Father$ и $Mother$ не пересекаются. Отношение включения следует из семантики операций $\sqcap$ и $\sqcup$.

Традиционными правилами выведения в ДЛ системах являются следующие редукции.

Утверждение 7 (Редукция к включению). Для концептов C и D справедливы такие эквивалентности в заданной терминологии $\mathcal{T}$:

1. C не выполняется тогда и только тогда, когда C включается в $\perp$;

2. C и D эквивалентны тогда и только тогда, когда C включается в D и D включается в C;

3. C и D не пересекаются тогда и только тогда, когда $C \sqcap D$ включается в $\perp$.

Дополнительными правилами выведения в ДЛ выступают правила, построенные на свойствах конструктора $\sqcap$.

Утверждение 8 (Редукция к невыполнимости). Для концептов C и D справедливы такие эквивалентности в заданной терминологии $\mathcal{T}$:

1. C включается в D тогда и только тогда, когда $C \sqcap \neg D$ не выполняется;

2. C и D эквивалентны тогда и только тогда, когда ни $C \sqcap \neg D$ ни $\neg C \sqcap D$ не выполняются;

3. C и D не пересекаются тогда и только тогда, когда $C \sqcap D$ не выполняется.

Приведенные правила имеют очевидную теоретико-множественную интерпретацию.

011 Удаление TBox

В практическом использовании желательно, чтобы TBox была пустой. Это можно сделать путем подстановок так, как это было показано в примере 2. Нетрудно понять, что когда терминология ациклическая, то это всегда можно сделать и таким образом свести проблему выведения в $\mathcal{T}$ к проблеме выведения в пустой терминологии TBox. Поскольку $\mathcal{T}$ и ее расширение $\mathcal{T}'$ эквивалентны, то произвольный концепт C и его расширение C' будут иметь одинаковую интерпретацию. То есть,

– $C \equiv_{\mathcal{T}} C'$ и

– C выполняется в $\mathcal{T}$ тогда и только тогда, когда выполняется C'.

Аналогично получаем

– $\mathcal{T} \models C \sqsubseteq D$ тогда и только тогда, когда $\models C' \sqsubseteq D'$ и

– $\mathcal{T} \models C \equiv D$ тогда и только тогда, когда $\models C' \equiv D'$ и

– C і D не пересекаются в $\mathcal{T}$ тогда и только тогда, когда C' і D' не пересекаются.

В этом случае возникает вопрос о сложности процедур вывода в $\mathcal{T}$, поскольку ее размеры могут возрастать экспоненциально. Анализ сложности процедур показывает, что они эффективнее работают в терминологии с пустой компонентой TBox [17; 21].

Отношения произвольной арности

Как следует из описания ДЛ, основными отношениями в ней являются бинарные отношения. В реальных использованиях возникает потребность в отношениях произвольной арности. Рассмотрим одно из расширений ДЛ, которое естественным образом вводит отношение произвольной арности и которое описано в работах [22; 23]. Полученный в результате такого расширения ДЛ-язык называется ДЛR.

Базовыми элементами ДЛR являются атомарные отношения и атомарные концепты, которые обозначаются P и A соответственно. Произвольное отношение арности $2 \leq n \leq n_{max}$ и произвольные концепты строятся по такому синтаксису:

$R ::= \top_n | P | (\$i/n : C) | \neg R | R_1 \sqcap R_2,$

$C ::= \top_1 | A | \neg C | C_1 \sqcap C_2 | \exists [\$i] R | \leq k[\$i] R,$
где i – i-я компонента отношения, то есть целое число $1 \leq i \leq n_{max}$, $n_{\max}$ – максимальная арность отношений P, R, R_1, R_2 и k – некоторое натуральное число.

Концепты и отношения должны быть согласованы по арности, т. е. комбинировать можно только отношения одной и той же арности.

Семантика ДЛR определяется обычным образом с помощью интерпретации $\mathcal{I} = (\Delta^{\mathcal{I}.I})$, в которой функция интерпретации $.I$ присваивает каждому концепту C подмножество $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, и каждому n-арному отношению R – подмножество $R^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$ так, что

$\top_n^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$

$P^{\mathcal{I}} \subseteq \top_n^{\mathcal{I}}$

$(\neg R)^{\mathcal{I}} = \top_n^{\mathcal{I}} \setminus R^{\mathcal{I}}$

$(R_1 \sqcap R_2)^{\mathcal{I}} = R_1^{\mathcal{I}} \sqcap R_2^{\mathcal{I}}$

$(\$i/n : C)^{\mathcal{I}} = \{(d_1, \ldots, d_n) \in \top_n^{\mathcal{I}} | d_i \in C^{\mathcal{I}}\}$

$\top_1^{\mathcal{I}} = \Delta^{\mathcal{I}}$

$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$

$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \sqcap C_2^{\mathcal{I}}$

$(\exists[\$i]R)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} | \exists(d_1, \ldots, d_n) \in R^{\mathcal{I}}.d_i = d\}$

$(\leq k[\$i]R)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} || \{(d_1, \ldots, d_n) \in R^{\mathcal{I}}.d_i = d\}| \leq k\},$

где P, R, R_1, R_2 отношения арности n. Отметим, что $\top_1$ обозначает область интерпретации, в то время как $\top_n$ обозначает не декартовое произведение этой области, а только некоторое его подмножество, которое включает все n-арные отношения, что определялись на этой области. Отсюда следует, что отрицание на отношениях определяется как разница отношений.

Конструкция $(\$i/n : C)$ означает все n-ки из $\top_n$, которые являются i-ми компонентами концепта C. Эта конструкция являет собой разновидность операции селекции. Квантор существования и числовые ограничения на отношениях являются обобщениями соответствующих конструкций для ролей.

На этом обзор средств ДЛ закончим и рассмотрим свойства отношений общего типа, с помощью которых описываются ПО.

Отношения описания ПО общего типа

Отношения общего типа разделим на семантические и математические отношения и рассмотрим связь между этими типами отношений.

Семантические отношения, о которых пойдет речь далее, выработаны в лингвистике, и поэтому их часто называют лингвистическими. Основными типами таких отношений являются [8]:

– Род-вид (А есть родовым к В),

– Синонимии (А синонимичен к В),

– Часть-целое (В часть А),

– Корреляции (А противоположный к В),

– Ассоциации (А ассоциируется с В),

– Оперирования (В есть операцией для А),

– Функциональное (А выражает В),

– Абстрагирования (А ограничивает В),

– Импликации (если A, то B),

– Классификации (A относится к классу B),

– Принадлежности (A является объектом B) и другие.

Наиболее важными семантическими отношениями являются отношения Род-вид, Синонимии, Часть-целое и Корреляции в связи с тем, что они охватывают более 70 процентов имеющихся терминов в украинском, русском и других языках.

Математические отношения. Наряду с вышеприведенными семантическими отношениями, имеются и отношения общего типа, играющие фундаментальную роль в математике. Эти отношения будем называть математическими отношениями. В целях автоматизации процесса анализа естественноязыковых текстов необходима формализация этого процесса и, в частности, отношения следующих типов:

– n-арные отношения ($n \geq 3$),

– бинарные,

– подобия,

– аналогии,

– обобщения,

– специализации,

– функциональные и другие отношения более частного характера.

Наиболее важными среди этих отношений являются бинарные отношения в связи с тем, что большинство из перечисленных отношений можно выразить в терминах бинарных.

Бинарные отношения делятся на отношения

– тождества,

– рефлексивные,

– иррефлексивные,

– симметрические,

– транзитивные,

– антисимметричные,

– асимметричные,

– репродуктивные,

– эвклидовы,

– плотности и другие.

Отношением подобия в терминах бинарных отношений называется рефлексивное и симметричное бинарное отношение. С помощью этих отношений определяются бинарные отношения, которые в математике имеют фундаментальное значение. К ним относятся отношения

– эквивалентности,

– частичного порядка и его вариации,

– функциональное и его вариации.

Отношение эквивалентности – это рефлексивное, симметричное и транзитивное отношение.

Отношение частичного порядка – это рефлексивное, антисимметричное и транзитивное отношение. Его вариациями являются отношения квазипорядка – рефлексивное и транзитивное отношение; отношение строгого порядка – иррефлексивное и транзитивное отношение; линейного порядка – это отношение частичного порядка с условием тотальности (любые два элемента сравнимы); отношение полного порядка – это отношение линейного порядка с условием существования наименьшего элемента в непустых подмножествах множества, на котором это отношение определено.

Функциональное отношение – это отношение, которое ставит в соответствие некоторым элементам одного и того же множества, или разных множеств не более одного элемента.

Взаимоотношения между отношениями. Рассмотрим отношения Род-вид, Часть целое, Принадлежности. Эти семантические отношения удовлетворяют свойствам рефлексивности, транзитивности и антисимметричности, а поэтому являются математическим отношением частичного порядка. Отсюда следует, что их представление носит иерархический характер и их элементы можно представлять в виде такой структуры данных как дерево или ациклический граф.

Отношение Синонимии и Классификации (имеется в виду разбиение на непересекающиеся подмножества) в общем случае являются отношениями эквивалентности, поскольку они рефлексивны, симметричны и транзитивны (разбиение индуцирует отношение эквивалентности).

Слова "в общем случае", относящиеся к этим отношениям, означают некоторую осторожность. Дело в том, что близкими отношениями к отношениям Синонимии и Классификации являются отношения Подобия и Аналогии. Существенная разница между этими отношениями состоит в том, что Подобие и Аналогии не являются транзитивными отношениями. Это связано с тем, что отношение подобия определяется на некотором уровне абстракции. Это проявляется в том, что в процессе поиска подобия выделяются особые свойства объектов (или элементов) и абстрагируются от других свойств, несущественных с нашей точки зрения. И два объекта считаются подобными, если они обладают одинаковыми выделенными свойствами. Отношение Подобия, в свою очередь, тесно переплетается с Аналогией, поскольку в процессе поиска подобия на первый план выходит отношение аналогии, по которому и ищутся общие свойства или устанавливаются различие между этими свойствами. Отсюда вытекает, что между одними и теми же объектами может существовать несколько аналогий и это нарушает свойство транзитивности этого отношения. Однако, часто отношение подобия является рефлексивным и симметричным в смысле одинаковости выделенных свойств.

Отношение Корреляции, Импликации и Оперирования в некотором смысле моделируются (или моделируют) каузуальные отношения, отношения следования (в смысле логического следования).

Отношения Ассоциации, Оперирования и Связывания в некотором смысле являются функциональными отношениями. Поскольку степень тождественности зависит от контекста использования отношений, семантической (смысловой) интерпретации и других обстоятельств, то полной уверенности в правильности принятых предположений не может быть. Это требует дополнительного более глубокого анализа.

Пример обработки ПО "Дискретная математика"

Принимая во внимание сказанное выше, рассмотрим пример построения терминологического тезауруса ПО "Дискретная математика" и на его основе онтографа, представляющего эту ПО.

Структура тезауруса. Тезаурус строится таким образом, что вместе с концептами в нем присутствуют семантические отношения. Семантические отношения – бинарные отношения, виды которых рассматривались выше. Тезаурус в данном случае играет роль Терминологии (аналог TBox) , а отношения, определенные на этой терминологии, можно считать ролями (в некотором смысле аналог ABox), которые присутствуют в ДЛ.

Структура тезауруса имеет вид:

Концепт	Опред-ние	X	Отнош1	Y	X	Отнош2	Y	$\cdots$
Множество	Интуит. опред.	-	-	-	-	-	-	$\cdots$
Элемент x	Объект мн-ва Y	x	R_1	Y	-	-	-	$\cdots$
A1:Акс. экст. X, Y	$X = Y$	X	R_2	Y	A1	R_3	A2	$\cdots$
A2:Акс. объемн. X, Y	$X \subseteq Y \wedge Y \subseteq X$	X	R_2	Y	A2	R_3	A1	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

,

где отношение R_1 – это отношение принадлежности, R_2 – отношение включения, R_3 – отношение синонимии и т.д.

В общем случае построенный тезаурус включает около 700 концептов, основными отношениями на которых являются отношения Синонимии, Часть-целое, Род-вид, Корреляции, Классификации, Принадлежности, Функциональное.

Имея в распоряжении терминологию и семантические отношения, начинается построение онтографа для выбранной ПО. Это построение выполняется с учетом взаимосвязей семантических отношений с математическими отношениями. На первый план здесь выступают отношения частичного порядка и эквивалентности. Отношение частичного порядка позволяет определить структуру онтографа, а отношение эквивалентности наполнение вершин этого графа.

Отношение синонимии играет важную роль в процессе интеграции онтологий, заданных своими онтографами, о чем будет сказано далее.

В результате программная система построения онтографа с учетом сказанного строит размеченный онтограф $G = (V, E)$ (см. рисунок 1 в Приложении 1 в конце статьи).

Дальнейшая работа с построенным онтографом выполняется путем реализации операций алгебры орграфов [20]. Сигнатура операций этой алгебры включает

– нульарную операцию $\Lambda = (\emptyset, \emptyset)$ – “абсолютно пустой граф”;

– бинарную операцию введения вершины u в орграф $G = (V, E)$, результатом которой есть орграф $G = (V \cup \{u\}, E)$;

– бинарную операцию введения ребра e в орграф $G = (V, E)$, результатом которой есть орграф $G = (V, E \cup \{e\})$;

– бинарную операцию удаления вершины u из орграфа $G = (V, E)$, результатом которой есть орграф $G = (V \setminus \{u\}, E \setminus \bigcup\limits_{e=(u,v)} e)$;

– бинарную операцию удаления ребра e из орграфа $G = (V, E)$, результатом которой есть орграф $G = (V, E \setminus \{e\})$.

Реализация операции Λ означает заготовку места для онтографа.

Известно, что перечисленные пять операций составляют полное множество операций, с помощью которых может быть реализована любая операция на конечных орграфах [20]. Отсюда следует, что относительно этих операций и носителя, каким является множество конечных орграфов, имеем универсальную алгебру, носителем которой являются конечные орграфы над некоторым универсальным множеством U (из которого берутся отметки вершин). Обычно в качестве множества U выступает множество натуральных чисел N или его декартова степень.

Программная система, реализующая эти операции, приведена в Приложении 1 в конце статьи.

Метаоперации на онтологиях: автоматная интерпретация

Пусть имеется несколько онтологий, которые представляются в виде своих онтографов $G_i = (V_i, E_i)$, $(i = 1, 2 \ldots, k)$. Рассмотрим онтограф $G = (V.E)$, множество вершин V которого представляет множество предметных областей, которые будем называть метаконцептами) а множество ребер E – бинарное отношение между этими предметными областями. Для построения орграфа $G = (V.E)$ применяются операции на онтографах $G_i = (V_i, E_i)$, которые будем называть метаоперациями.

С каждым таким орграфом $G = (V, E)$ будем ассоциировать конечный (вообще говоря) частичный детеминированный автомат без выходов $A = (V, X = V, f, S, F)$, где V – множество состояний, которое также служит входным алфавитом данного автомата, S – подмножество начальных состояний, $F \subseteq V$ – подмножество заключительных состояний (которое, в частности, может быть пустым), а функция переходов данного автомата определяется следующим образом: $f(u, v) = v$ тогда и только тогда, когда $(u, v) \in E$ и не определено в остальных случаях.

Рассмотрим пример представления фрагмента онтологии для ПО “Комбинаторика”.

Пусть задана онтология, отражающая некоторую часть этой предметной области (разделы взяты из математической энциклопедии), в виде следующего онтографа:

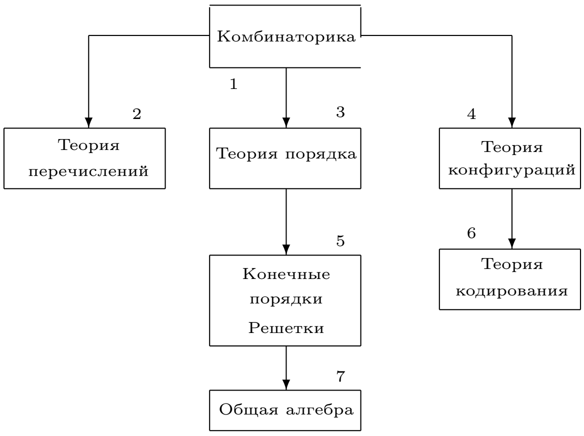

Рис. 6. Онтология C

Конечный автомат, соответствующий данной онтологии, имеет вид $A = (V = \{1, 2, 3, 4, \ 5, 6, 7\}, X = \{1, 2, 3, 4, 5, 6, 7\}, f, \{1\}, \{7\})$, где f задана таким графом переходов:

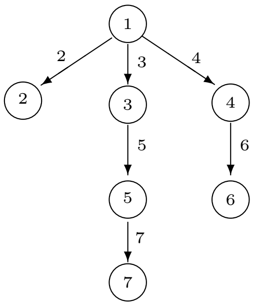

Рис. 7. Автомат A онтологии C

Это значит, что $f(1, 2) = 2, f(1, 3) = 3, f(1, 4) = 4, f(3, 5) = 5, f(5, 7) = 7, f(4, 6) = 6$. Остальные переходы в данном автомате неопределены.

Метаоперации на онтологиях. Представление онтологий в виде конечного автомата без выходов позволяет ввести метаоперации на онтологиях. Основными такими операциями являются:

– объединение – теоретико-множественное объединение множества состояний и множества переходов данных автоматов-аргументов;

– пересечение – теоретико-множественное пересечение множества состояний и множества переходов, пополненное транзитивным замыканием отношения достижимости на автоматах-аргументах;

– конкатенация или умножение двух автоматов – частный случай операции объединения, когда объединение выполняется только по множеству начальных состояний второго автомата;

– итерация – повторяемая конечное число раз операция умножения, применяемая в рамках одной онтологии с целью уточнения и пополнения этой онтологии (с помощью этой операции осуществляется пошаговое уточнение и пополнение онтологий);

– обращение – ориентация в противоположном направлении переходов в автомате, представляющем данную онтологию, т. е. построение функции переходов $g(v, u) = u$ тогда и только тогда, когда $f(u, v) = v$ и неопределено в остальных случаях.

Приведем примеры, иллюстрирующие перечисленные выше операции. Пусть имеем такую онтологию:

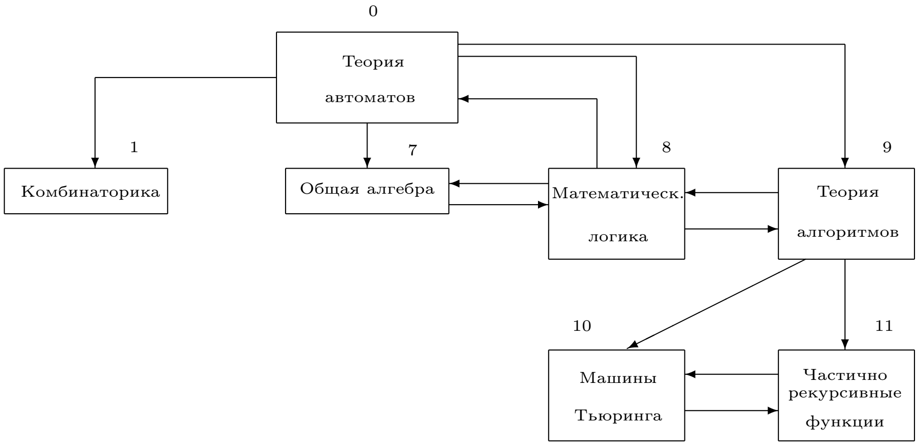

Рис. 8. Онтология O_1

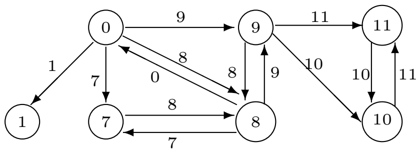

Рис. 9. Автомат A_1 онтологии O_1

Объединение онтологий C и O_1. При реализации операции объединения находятся общие вершины онтографа, по которым и выполняется объединение онтологий. Поскольку такие вершины могут иметь разные метки, то поиск ведется с учетом отношения синонимии. Если в описании вершин находятся синонимичные концепты, то эти вершины считаются одинаковыми и их можно отождествить. После отождествления вершин нужно удалить общие подграфы.

Тогда введенные выше операции дают такие результаты, если их применить к автоматам A и A_1 (автомат A показан на рис. 7):

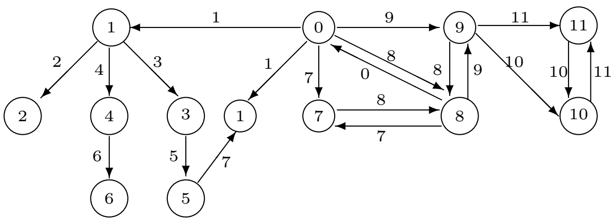

Рис.10. Объединение автоматов A и A_1

Пересечение онтологий C O_1. При выполнении операции пересечения подграфов (как и при выполнении операции объединении) учитывается отношение синонимии. Одинаковыми считаются вершины онтографа, у которых имеются синонимичные концепты. В данном примере пересечение дает такой результат:

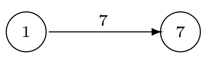

Рис. 11. Автомат $A \cap A_1$

Итерация: пополнение (уточнение) онтологии O_1. Если имеется онтология некоторой подобласти более общей онтологии, то ее пополнение выполняется путем склеивания общих вершин в онтографе. Например, пусть имеем такую онтологию:

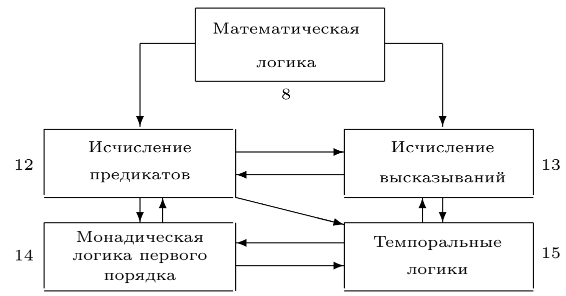

Рис. 12. Онтология O_2

Автомат A_2, соответствующий онтологии O_2, принимает вид:

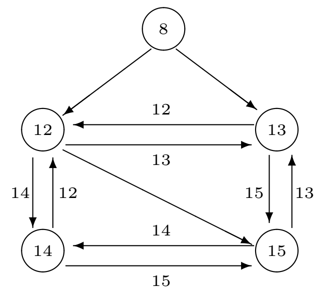

Рис. 13. Автомат A_2 для онтологии O_2

Конкатенируя автоматы A_1 и A_2 по начальному состоянию 8 автомата A_1, получаем автомат, представляющий пополненную (уточненную) онтологию $O_1 * O_2$:

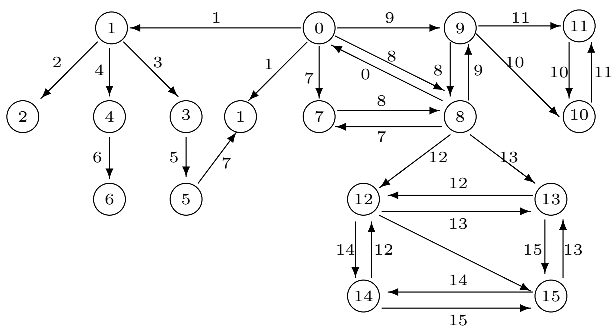

Рис. 14. Пополнение онтологии O_1

Обращение. Применяя эту операцию к A_1, получаем автомат:

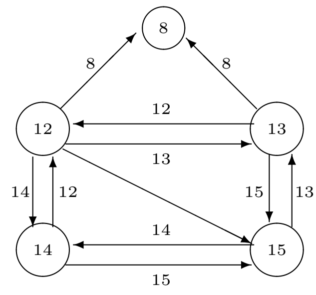

Рис. 15. Обращение автомата онтологии O_2

Приведенное множество операций (в случае надобности) можно расширять по крайней мере в двух направлениях. Одним из них является расширение операций на графах (соединения графов, изоморфного соединения, декартового произведения и т. п.).

Другим направлением является алгебра отношений. Поскольку каждая онтология является представлением некоторой совокупности отношений, то можно вводить операции реляционной алгебры.

Какое из возможных направлений будет выбрано, зависит от практических потребностей использования онтологий. Представленные операции над онтологиями оказываются полезными при анализе, синтезе и манипулировании онтологиями и онтологическими объектами.

Выводы

Подводя итог сказанному, выделим некоторые проблемы, возникающие при построении и работе с онтологиями как концептуальными моделями ПО.

Первая проблема (и возможно основная при работе с онтологиями) связана с автоматизацией процесса извлечения знаний из естественноязыковых текстов. Одним из возможных направлений в решении этой проблемы является некоторая предварительная обработка текста. Такая обработка должна связывать семантические и математические отношения, например с помощью аннотирования текста или другого способа, облегчающего дальнейшую обработку.

Проблемы реализации вышеприведенных операций на онтологиях, связаны с тем, что корректное выполнение этих операций требует создания некоторого общего глоссария предметных областей и понятий, с помощью которого можно было бы однозначно идентифицировать соответствующие объекты. По видимому, эта проблема является не только проблемой на пути реализации введенных операций, но и в некотором смысле общей проблемой на пути построения онтологий и работы с онтологиями. Здесь приходит на помощь отношение синонимии (которое предполагается отношением эквивалентности). Вершины онтографов могут иметь разные названия, но если эти вершины содержат синонимичные объекты, то они склеиваются в одну вершину при реализации таких операций как объединение и пересечение.

Следующая проблема, возникающая при реализации операций, связана с имеющейся иерархией областей и понятий. Дело в том, что в различных онтологиях одни и те же понятия и объекты могут находиться на разных уровнях иерархии и это необходимо учитывать при применении операций. В предлагаемом подходе эта проблема решается с помощью построения транзитивного замыкания отношения достижимости на состояниях автоматов, представляющих данные онтологии. Однако, нет уверенности в том, что этого замыкания достаточно для полного решения проблемы. Здесь необходимы эксперименты с реальными онтологиями и их представлениями.

И еще одна проблема связана с полнотой знаний, имеющихся в представленных онтологиях. Эта проблема является основной в процессе создания программного и технического обеспечения систем. Здесь же эта проблема состоит в построении в некотором смысле полной онтологической картины той или иной предметной области.

Приложение 1

Инструкция

Программная система строит онтограф некоторой предметной области и выполняет операции на этом онтографе, которые позволяют искать, добавлять и удалять концепты, а также устанавливать связи между концептами. Система имеет интерфейс, который позволяет выводить на экран онтограф.

Если нужно добавить новый концепт, то необходимо нажать на вкладку Concepts

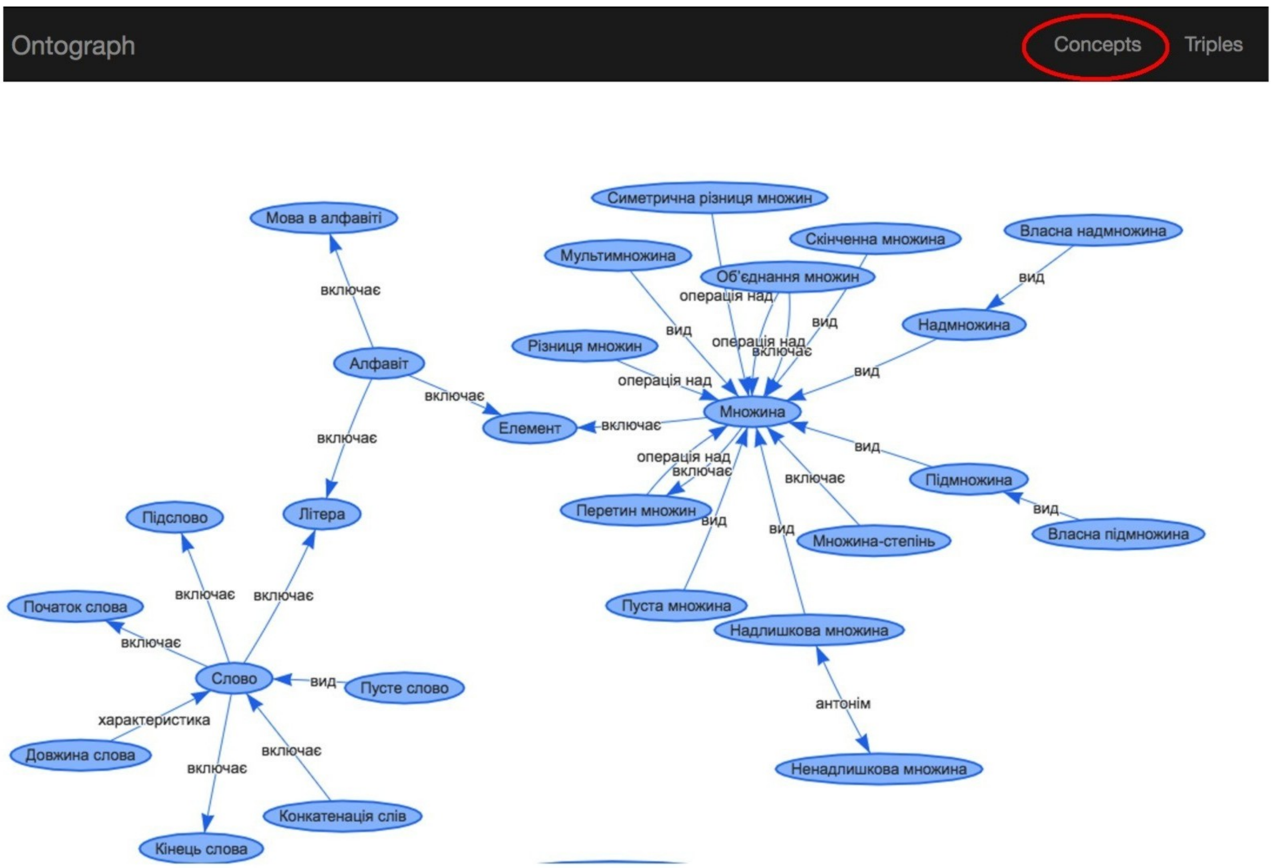

Рис. 1: Пример перехода на вкладку с концептами

После чего появляется окошко:

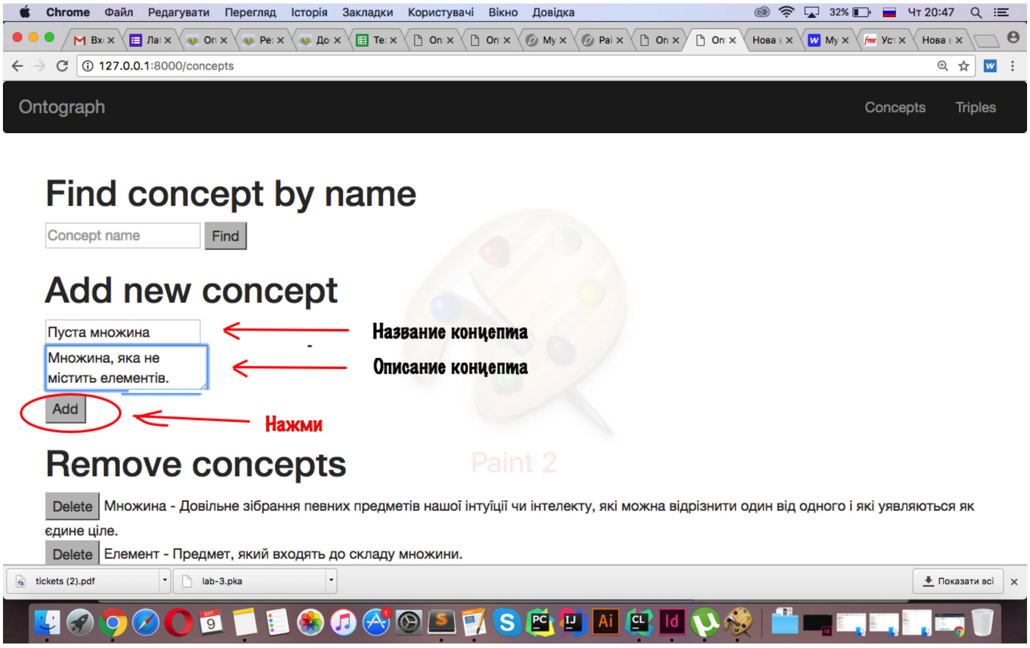

Рис. 2: Пример добавления концепта

Для того что б найти концепт нужно ввести его название в поле и нажать на кнопку <Find>:

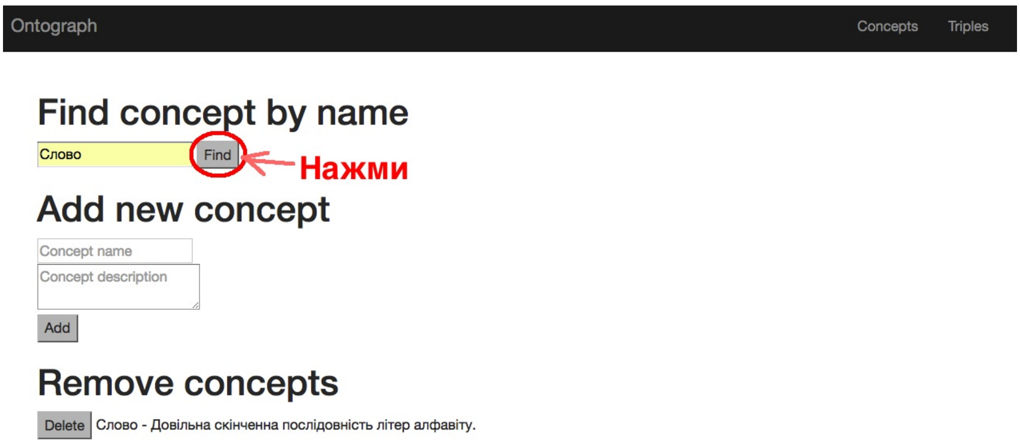

Рис. 3: Пример поиска концепта

Для удаления концепта нужно нажать клавишу <Delete> возле концепта который нужно удалить:

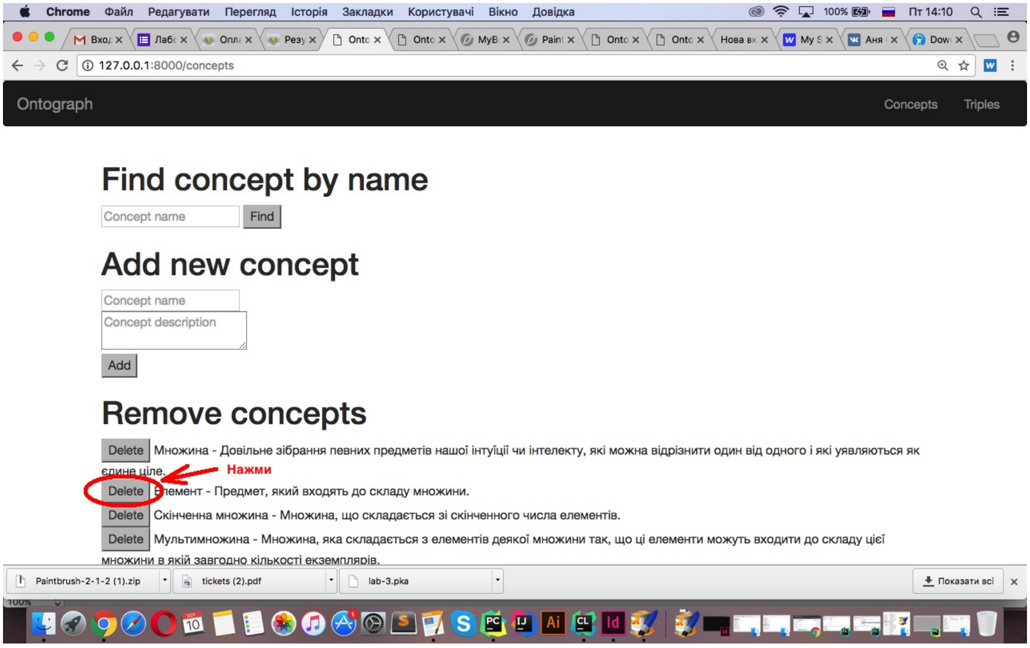

Рис. 4: Пример удаления концепта

Когда нужно установить связь между двумя концептами, то для этого выбираем вкладку <Triples>

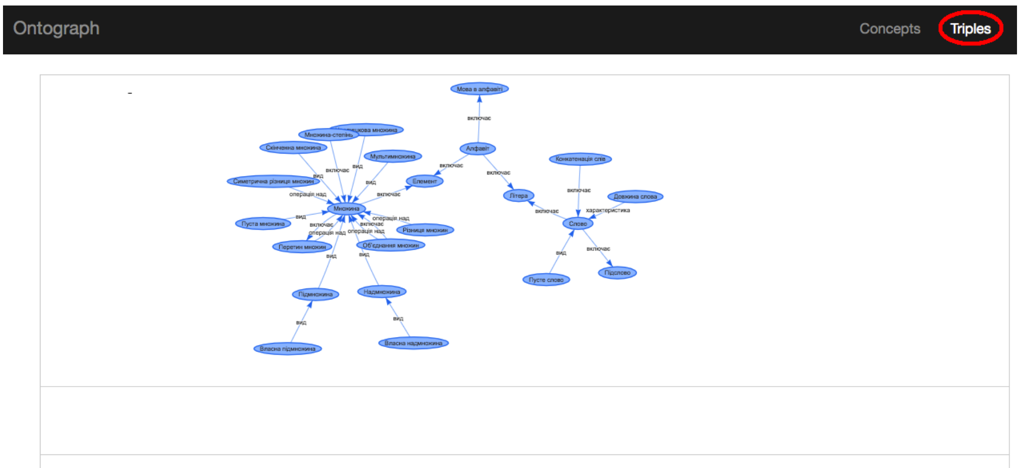

Рис. 5: Пример перехода на вкладку со связями

Затем выбирается первый концепт из списка, вводится имя связи, выбирается второй концепт и нажимается клавиша Add.

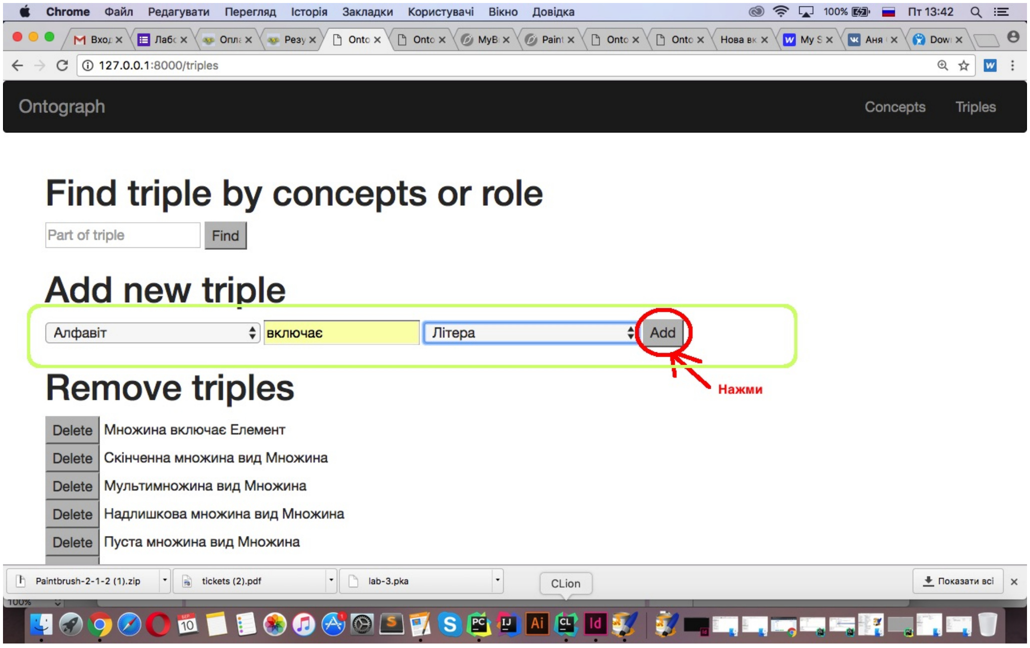

Рис. 6: Пример добавления связи

Имеется возможность найти все концепты по связи. Для этого нужно ввести название связи и нажать клавишу <Find>:

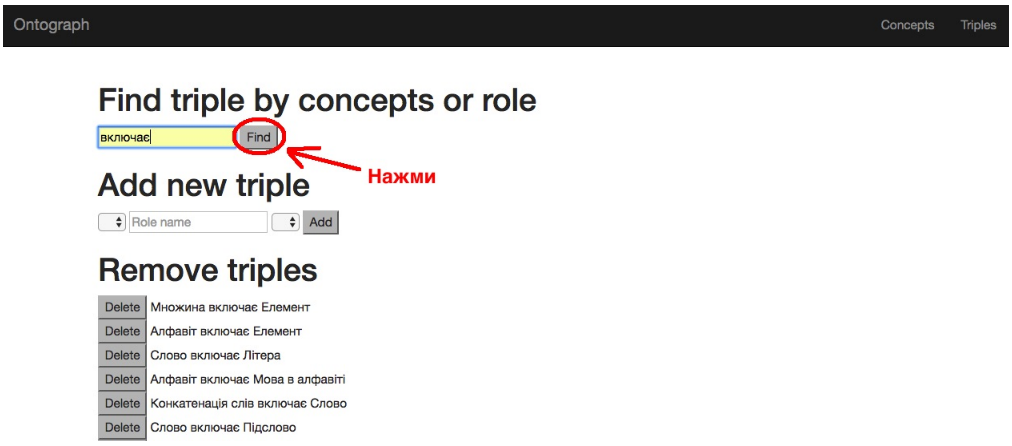

Рис. 7: Пример поиска связи

Для удаления связи нужно нажать клавишу <Delete> возле связи, которую нужно удалить.

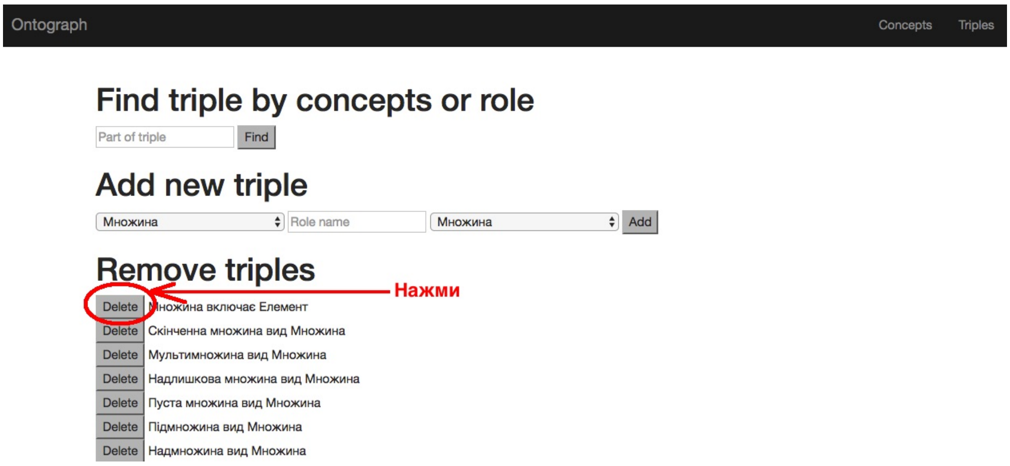

Рис. 8: Пример удаления связи

Для просмотра определения концепта нужно дважды нажать на него левой кнопкой мыши и определение появится под онтографом:

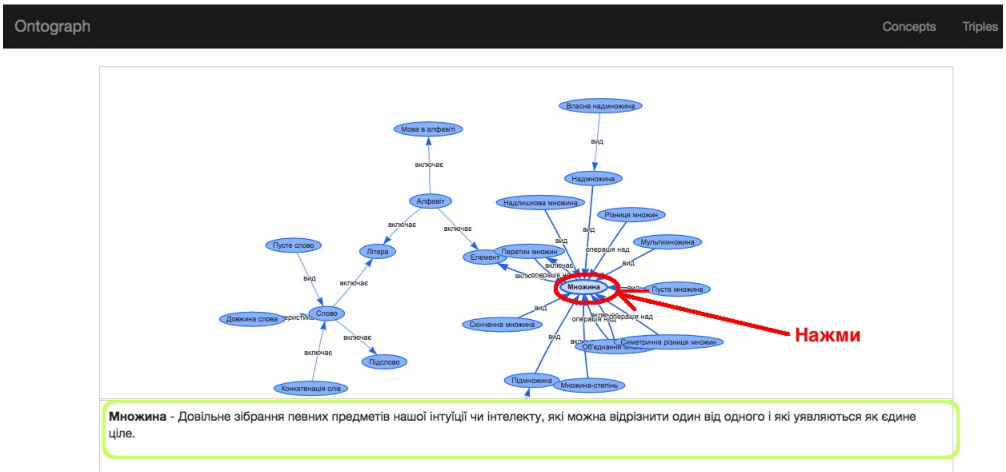

Рис. 9: Пример показа определения

Для просмотра синонимов концепта нужно один раз нажать на него левой кнопкой мыши и синонимы появятся возле данного концепта:

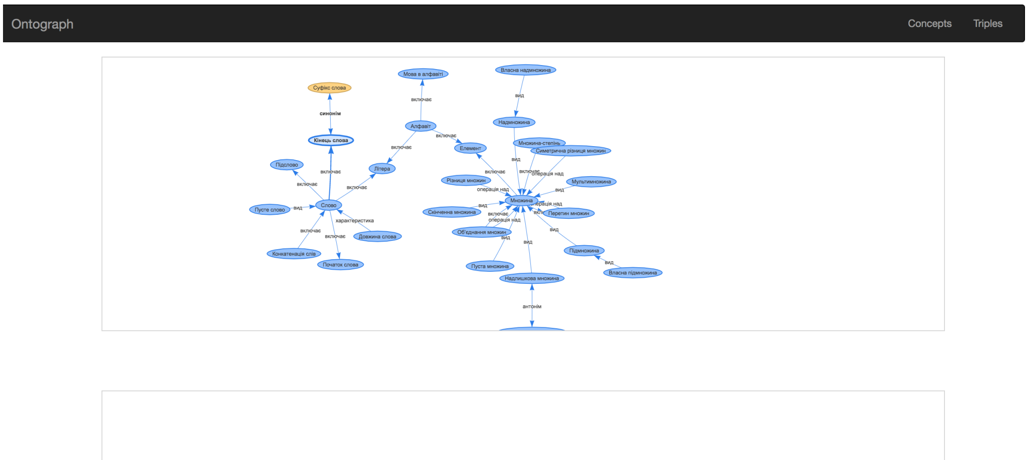

Рис. 10: Пример показа синонимов

Список литературы

[1] А.В. Палагин, С.Л. Крывый, Н.Г. Петренко Знание-ориентированные информационные системы с обработкой естественноязыковых объектов: основы методологии и архитектурно-структурная организация. – УСиМ. – 2009. – № 3. – С. 42 – 55.

[2] Chomsky N. Studies on Semantics in Generative Grammar. The Hague: Mouton, 1972.– Limited preview available in English. – Vista previa limitada disponible en castellano.

[3] Chomsky N. Topics in the Theory of Generative Grammar. Paris: Mouton, 1972. – Limited preview available in English. – Vista previa limitada disponible en castellano.

[4] Fillmore C. J. Frame Semantics . – Linguistics in the morning calm: Selected papers from the SICOL. Seoulю – 1982. – PP. 111 – 137.

[5] Шенк Р. Обработка концептуальной информации. – М.: Энергия, 1980. – 361с.

[6] Шенк Р., Бирнбаум Л., Мей Дж. Новое в зарубежной лингвистике. Компьютерная лингвистика. – М.: Прогресс, 1989. – Вып. 14.

[7] Мельчук И. А. Опыт теории лингвистических моделей “Смысл <=> Текст”. М. – 1974 (2-е изд.). – 1999.

[8] Дарчук Н. П. Комп'ютерне анотування українського тексту: результати і перспективи. К: Освіта України. – 2013. – 544 с.

[9] Леонтьева Н. Н. К теории автоматического понимания естественного языка. М.: Изд.-во Моск. ун-та. ч.2. Семантические словари: состав, структура, методика создания. – 2001. – 40 с.

[10] Baader F., Calvanese D., McGuinness D.L. and other. The Description Logic Handbook. Cambridge: University Press. – 2007. – 601 p.

[11] Пименова М.В. Введение в когнитивную лингвистику. Кемеровою – 2004.

[12] Рубашкин В. Ш. Представление и анализ смысла в интеллектуальных информационных системах. – М.: Наукаю – 1989. – 192 с.

[13] Кулик Б. А. Логика естественных рассуждений. – С.-Петербург: Невский диалект. – 2001. – 127 с.

[14] Brachman R.J. What's in a concept: structural foundations for semantic networks. – In Journal of Man-Machine Studies. – 1977. – 9(2). – PP.127 – 152.

[15] Brachman R.J. Structured inheritance networks. – Research in Natural Language Understanding. – Quarterly Progress Report. – N. 1 – BBN. – Report No. 3742.

[16] Schmidt-Schauβ M. Subsumption in KL-ONE is undesidable. – Proc. of the 1st Int. Conf. on Princieples of Knowledge Representation and Reasoningю – 1989 (KR'89). – PP. 421 – 431.

[17] Nebel B. Termonological reasoning is inherently intractable. – Artificial Inteligence. – 1990. – N. 43 – PP. 235 – 249.

[18] Schild K. A correspondence theory for terminological logics. – Preliminary report. In Proc. of the 12h Int. Joint Conf. on Artificial Intelligence (IJCA'91). – 1991. PP. 466 – 471.

[19] Tarski A. A lattise-theoretical fixpoint theorem and its applications. – Pacific Journal of Mathematics. – 1955. – N. 5 – PP. 285 – 309.

[20] Сергієнко І.В., Кривий С.Л., Провотар О.І. Алгебраїчні аспекти інформаційних технологій. – К., Наукова думка. – 2011. – 399 с.

[21] Lutz C. Complexity of termonological reasoning revisited. – In Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99). Lecture Notes in Artificial Intelligence. - v.1705 – 1999. – PP. 181 – 200.

[22] Calvanese D., Giacomo G., Lancerini M. Conjunctive query containment in Description Logics with n-ary relations. – In Proc. of the 1997 Description Logic Workshop (DL'97). – 1997. – PP. 5 – 9.

[23] Calvanese D., Giacomo G., Lancerini M. On the decidability of query containment under constraints. – In Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98). – 1998. – PP. 149 – 158.

Сведения об авторах:

Крывый Сергей Лукьянович, доктор физ.-мат. наук, профессор, профессор кафедры информационных систем факультета компьютерных наук и кибернетики Киевского национального университета имени Тараса Шевченка.

Дарчук Наталия Петровна, доктор филологических наук, профессор, профессор кафедры украинского языка и прикладной лингвистики Института филологии Киевского национального университета имени Тараса Шевченка.

Ясенова Ирина Сергеевна, кандидат технических наук, доцент кафедры компьютерных наук факультета информационных технологий Киевского национального университета биоресурсов и природопользования.

Головина Александра Леонидовна, студентка 4-го курса факультета компьютерных наук и кибернетики Киевского национального университета имени Тараса Шевченка.

Соляр Анна Сергеевна, студентка 4-го курса факультета компьютерных наук и кибернетики Киевского национального университета имени Тараса Шевченка.

METHODS AND TOOLS OF KNOWLEDGE REPRESENTATION SYSTEMS

S.L.Kryvyy, N.P.Darchuk, I.S.Yasenova, A.L.Golovina, A.S.Solyar

Abstract: Short review of systems analisys, tools and knowledge representations, building of ontologies by using natural languages text processing are considred. In spacial case presented review on Descriptions Logics, which is now active development. An example representation of concrete objective area by using ontograph with set of operations over ontologies are described.

ACM Classification Keywords: Systems of knowledge representation, Description Logics, ontograph, operations over ontologies