

А.А. ЗУЕНКО, П.А. ЛОМОВ, А.Г. ОЛЕЙНИК
**ПРИМЕНЕНИЕ МЕТОДОВ РАСПРОСТРАНЕНИЯ
ОГРАНИЧЕНИЙ ДЛЯ УСКОРЕНИЯ ОБРАБОТКИ ЗАПРОСОВ
К ОНТОЛОГИЯМ**

Зуенко А.А., Ломов П.А., Олейник А.Г. **Применение методов распространения ограничений для ускорения обработки запросов к онтологиям.**

Аннотация. Предлагается рассматривать задачу обработки SPARQL-запросов к онтологии как задачу удовлетворения ограничений. Кратко представлен формальный аппарат, служащий для описания задач удовлетворения ограничений с помощью специализированных матрицеподобных структур. Применение данного аппарата позволяет более эффективно по сравнению с табличным представлением описывать, хранить и обрабатывать нечисловые ограничения предметной области. Для ускорения обработки запросов к онтологиям большого объема предлагается применить ранее разработанный авторами метод распространения нечисловых ограничений. По сравнению с традиционным подходом к обработке SPARQL-запросов, базирующимся на динамическом программировании, предлагаемый метод позволяет ускорить их выполнение за счет «компактного» представления онтологии, а также применения оригинальных авторских правил редукции пространства поиска. На конкретном примере показано использование этого математического аппарата для снижения размерности пространства поиска при выполнении запроса к онтологии междисциплинарных знаний.

Ключевые слова: запрос к онтологии, задача удовлетворения ограничений, матричное представление ограничений, распространение ограничений.

1. Введение. Инструменты для работы со структурами онтологических моделей позволяют сопоставлять и связывать различные онтологии, выбирать компоненты существующих онтологий для создания новых, что способствует унификации онтологических моделей и расширяет возможности их совместного использования. Важным аспектом при совместном использовании онтологий различных областей знания является обеспечение их семантической интероперабельности, основанной на одинаковой интерпретации различными онтологиями информационных объектов. Упростить решение проблемы семантической интероперабельности прикладных онтологий, которые предполагается интегрировать, на этапе их разработки позволяет применение шаблонов онтологического проектирования (Ontology Design Patterns, ODP) [1]. ODP положены в основу разрабатываемой авторами онтологии интегрированного пространства знаний (ИПЗ) об Арктической зоне Российской Федерации (АЗРФ). В [2] обоснована актуальность задачи формирования ИПЗ АЗРФ и рассмотрены подходы к ее решению на примере создания единого пространства мультипредметных знаний Кольского научного центра РАН.

Существующие и развиваемые технологии и средства решения проблем технологической и семантической интероперабельности предметных онтологий, объединяемых в ИПЗ АЗРФ, позволяют сформировать комплексную формализованную систему междисциплинарных знаний. Но пополнение этой системы новыми знаниями выводит на передний план еще одну проблему — проблему манипулирования представленными в этой системе знаниями, обусловленную снижением производительности запросов. Значимые задержки отклика системы на запросы проявляются, когда количество триплетов в онтологиях ИПЗ достигает 10 миллионов [3, 4].

В настоящей статье для ускорения обработки запросов к онтологиям большого объема предлагается применить ранее разработанные авторами методы распространения ограничений [5]. Кратко представлен формальный аппарат, служащий для описания с помощью матрицеподобных структур задач удовлетворения ограничений. Применение данного аппарата позволяет более эффективно, по сравнению с табличным представлением, описывать, хранить и обрабатывать нечисловые ограничения предметной области. На конкретном примере показано применение этого математического аппарата для снижения размерности пространства поиска при выполнении запроса к онтологии междисциплинарных знаний.

2. Онтология интегрированного пространства знаний. В качестве основы онтологии ИПЗ использована онтология информационных артефактов (Information Artifact Ontology, IAO) [6], которая, в свою очередь, является расширением одной из распространенных онтологий верхнего уровня — базовой формальной онтологии (Basic Formal Ontology, BFO) [7]. Онтология IAO ориентирована на представление процессов получения информации, их участников, а также информационных источников и носителей.

С целью упрощения применения онтологии ИПЗ было принято решение выделить целостные фрагменты понятийной системы IAO, представляющие отдельные концепты, дополнить их описание с учетом специфики проведения и документирования исследовательской деятельности в РФ и локализовать их в виде отдельных онтологических паттернов содержания (Ontology Content Design Patterns). Каждый такой паттерн реализован в виде мини-онтологии, которая отражает одну точку зрения на несколько связанных понятий предметной области. Использование паттернов для представления знаний в онтологии избавляет пользователя от манипуляции с большим количеством понятий и отношений [8], а также позволяет гарантировать качество такого представления ввиду того, что каждый паттерн представляет

доказавший свою эффективность вариант решения задачи онтологического моделирования. Общая структура онтологии ИПЗ представлена на рисунке 1.

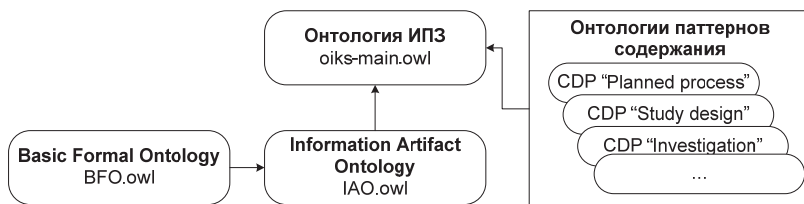


Рис. 1. Общая структура онтологии ИПЗ

В качестве примера рассмотрим один из паттернов разработанной онтологии ИПЗ, используемый для представления процессов исследований и соотносенных с ними объектов. UML-схема данного паттерна приводится на рисунке 2.

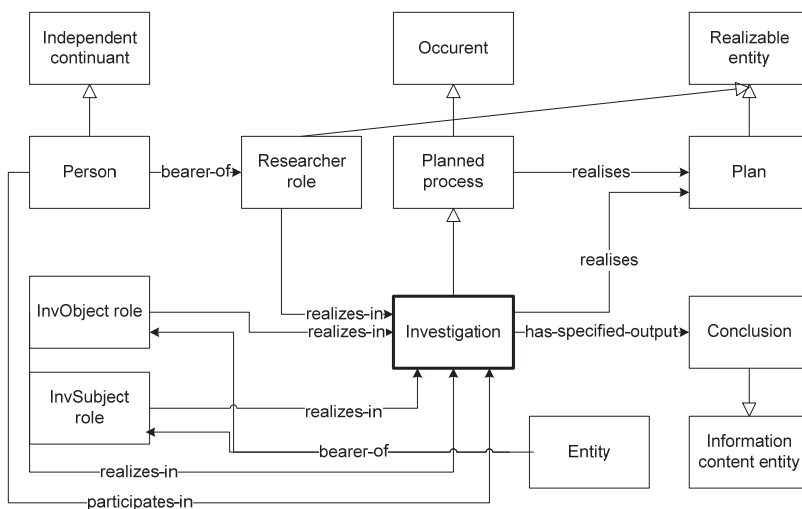


Рис. 2. UML-схема онтологического паттерна содержания «Исследование» (Investigation)

«Процесс исследования» (Investigation) включает в качестве участников (отношение participates-in) персон (Person), реализующих роль «Исследователь» (Researcher), и сущности, реализующие роли «Объект исследования» (Investigation object role) и «Предмет исследования» (Investigation subject role). В результате процесса

исследования продуцируется некоторое заключение (Conclusion), которое является «Информационным объектом» (Information content entity). Понятия «Независимая сущность» (Independent continuant), «Происходящее» (Occurrent), «Реализуемая сущность» (Realizable entity) принадлежат онтологии BFO.

В соответствии с технологией экстремального проектирования онтологий (eXtreme Design methodology, XD) [9, 10] для каждого паттерна содержания задается набор квалификационных вопросов (Competency questions), которые указывают, какую информацию можно получить с помощью онтологии, содержащей данный паттерн. Для рассматриваемого паттерна были определены следующие квалификационные вопросы:

- Кто участвовал в процессе исследования?
- Кто занимался исследованием данной сущности?
- Какие исследования проводились для данной сущности?
- С кем сотрудничал участник в процессе исследования?
- Какую роль выполнял участник исследования?
- Какой предмет в контексте данного объекта исследовался?
- Какие выводы были получены в результате исследования?

Рассмотренный паттерн содержания, а также полная онтология ИПЗ могут быть загружены из GIT-репозитория [11].

При работе с онтологией для каждого квалификационного вопроса формируется соответствующий запрос на специализированном языке SPARQL. Обеспечение приемлемой скорости исполнения таких запросов на больших объемах данных является одной из распространенных проблем [12-14]. В отличие от большинства работ-прототипов, например [4, 15], в настоящей статье предлагается задачу обработки запросов к онтологиям рассматривать не как задачу динамического программирования, а как задачу удовлетворения ограничений (программирования в ограничениях).

3. Задачи удовлетворения ограничений и матричное представление ограничений. Программирование в ограничениях сформировалось на стыке таких направлений, как искусственный интеллект, теория языков программирования, символьные вычисления и вычислительная логика. Технология программирования в ограничениях широко используется при решении многих прикладных задач, таких как календарное планирование, задачи назначения частоты, обработка изображений, тестирование сверхбольших интегральных схем, анализ языков программирования и ЗАДАЧИ понимания естественного языка и т.д. Система программирования в ограничениях стремится сократить перебор вариантов, для ускорения

вычислительных процедур используются методы распространения ограничений, эвристического поиска и т.п.

Дадим определения, необходимые для дальнейшего изложения.

Определение 1. Согласно [16, 17] *сеть ограничений* или *задача удовлетворения ограничений* (Constraint Satisfaction Problem — CSP) определена множеством переменных $V = \{x_1, x_2, \dots, x_n\}$ и множеством ограничений $C = \{C_1, C_2, \dots, C_m\}$. Каждая переменная x_i имеет непустую область определения D_i (домен). Каждое ограничение C_i включает некоторое подмножество переменных (диапазон отношения) и задает допустимые комбинации значений для этого подмножества.

Ограничением называется пара (R, S) , где R — отношение, определенное на диапазоне S . Решение задачи удовлетворения ограничений состоит в присваивании значения каждой переменной так, чтобы одновременно удовлетворялись все ограничения.

Ограничение может быть представлено либо в явной форме путем перечисления кортежей значений, удовлетворяющих заданному условию, либо в виде абстрактного отношения, для которого вводится две операции: проверка кортежа на принадлежность отношению и перечисление элементов отношения [18]. В первом случае ограничения могут задаваться с помощью таблиц, описывающих допустимые сочетания значений переменных [19]. При этом вычисления, необходимые для решения CSP, удобно выполнять с использованием операций в стиле реляционной алгебры. Часть из этих операций предполагают, что отношения сформированы в одной схеме (один диапазон), другие же операции служат для работы с отношениями, сформированными в разных схемах.

Пусть R_1, R_2 — два отношения с одинаковым диапазоном. Тогда к ним можно применить следующие операции: *пересечение* $R_1 \cap R_2$, *объединение* $R_1 \cup R_2$, *разность* $R_1 \setminus R_2$, *дополнение* $U \setminus R_1$ отношения R_1 относительно универсума U (универсального отношения).

Также введем операцию соединения многоместных отношений.

Определение 2. *Соединение* $R_S \bowtie R_T$ двух отношений R_S (с диапазоном S) и R_T (с диапазоном T) является отношением с диапазоном $S \cup T$ и множеством всех кортежей $(a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_k)$, получаемых слиянием кортежей $(a_1, \dots, a_n, b_1, \dots, b_m) \in R_S$ и $(b_1, \dots, b_m, c_1, \dots, c_k) \in R_T$, где компоненты b_1, \dots, b_m являются значениями переменных из множества $S \cap T$.

Качественные ограничения, такие как правила, логические формулы, многоместные отношения играют важную роль при моделировании многих предметных областей. Существующие на данный момент в теории удовлетворения ограничений методы, основанные на табличном представлении нечисловых ограничений, недостаточно эффективны. В настоящей работе для представления и обработки нечисловых ограничений предлагается использовать два типа матрицеподобных структур: C -системы и D -системы [20, 21].

Особенностью настоящих исследований является то, что эти структуры предлагается рассматривать как ограничения над конечными доменами, а рассуждения на данных структурах реализовывать в форме процедур удовлетворения ограничений.

С помощью C -систем удобно моделировать дизъюнктивные нормальные формы (ДНФ) конечных предикатов. Продемонстрируем это на примере. Пусть задан конечный предикат:

$$\varphi(x, y, z) = (x = a, b) \wedge (y = a, c) \vee (z = d).$$

Для простоты все переменные определены на множестве $\{a, b, c, d\}$. Здесь и далее будем использовать запись вида $(x = a, b)$ для обозначения выражения $(x = a) \vee (x = b)$. Учитывая, что область истинности одноместного предиката $(x = a, b)$ есть $\{a, b\}$, то область истинности предиката $\varphi(x, y, z)$ может быть представлена в виде следующей C -системы:

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

Атрибуты X , Y , Z отношения $R[XYZ]$ соответствуют переменным x , y , z формулы $\varphi(x, y, z)$. Заметим, что “*” — сокращенное обозначение всего домена атрибута. C -систему $R[XYZ]$ можно преобразовать в многоместное отношение следующим образом:

$$(\{a, b\} \times \{a, c\} \times \{a, b, c, d\}) \cup (\{a, b, c, d\} \times \{a, b, c, d\} \times \{d\}).$$

С помощью D -систем моделируются конъюнктивные нормальные формы (КНФ) конечных предикатов. D -система записывается как матрица компонент-множеств, которые заключены в перевернутые скобки. D -системы позволяют легко вычислять дополнение C -систем: берется дополнение каждой компоненты.

Например, $\neg\varphi \equiv (\neg(x = a, b) \vee \neg(y = a, c)) \wedge \neg(z = d)$, что с учетом конечных областей определения переменных равносильно $\neg\varphi \equiv ((x = c, d) \vee (y = b, d)) \wedge (z = a, b, c)$. Предикат $\neg\varphi$ может быть выражен как D -система $\bar{R}[XYZ]$:

$$\bar{R}[XYZ] = \begin{bmatrix} \{c, d\} & \{b, d\} & \emptyset \\ \emptyset & \emptyset & \{a, b, c\} \end{bmatrix}.$$

\emptyset — это фиктивная компонента, не содержащая значений.

В работах [5, 6] приводятся конкретные методы решения задач CSP, основанные на матричном представлении ограничений с конечными доменами в виде D -систем. Данные методы опираются на утверждения, позволяющие зачастую редуцировать пространство поиска без организации ветвления.

С однотипными матрицами ограничений, заданными внутри одного и того же диапазона (в одной схеме отношения), можно выполнять любые операции алгебры множеств. При этом нет необходимости преобразовывать их в совокупность элементарных кортежей — все операции выполняются с компонентами атрибутов или кортежами компонент. Это свойство позволяет существенно уменьшить трудоемкость вычислений. Операции (пересечение, объединение, дополнение) для матрицеподобных структур осуществляются на основании теорем, изложенных в [20]. Ввиду ограничений на объем публикации здесь приводится лишь теорема, непосредственно используемая в дальнейшем.

Теорема 1. Пересечение двух C -систем с одинаковыми схемами равно C -системе, содержащей все непустые пересечения каждой строки первой C -системы с каждой строкой второй C -системы.

Пусть заданы C -системы:

$$R_1[XYZ] = \begin{bmatrix} \{a, b, d\} & \{f, h\} & \{b\} \\ \{b, c\} & * & \{a, c\} \end{bmatrix},$$

$$R_2[XYZ] = \begin{bmatrix} \{a, d\} & * & \{b, c\} \\ \{b, d\} & \{f, h\} & \{a, c\} \\ \{b, c\} & \{g\} & \{b\} \end{bmatrix}.$$

Оценим количество операций сравнения элементов при пересечении отношений в двух случаях: 1) отношения представлены через элементарные кортежи; 2) отношения выражены в виде C -систем.

В первом случае имеем: количество операций = $18 \times 22 \times 3 = 1188$, поскольку отношение R_1 содержит 18 элементарных кортежей, а отношение R_2 — 22 кортежа, причем каждый из элементарных кортежей содержит по 3 элемента. Во втором случае пересечение отношений R_1 и R_2 вычисляется как результат пересечения всех пар C -кортежей, содержащихся в этих C -системах:

$$\begin{aligned} & [\{a, b, d\} \{f, h\} \{b\}] \cap [\{a, d\} * \{b, c\}] = [\{a, d\} \{f, h\} \{b\}]; \\ & [\{a, b, d\} \{f, h\} \{b\}] \cap [\{b, d\} \{f, h\} \{a, c\}] = \emptyset; \\ & [\{a, b, d\} \{f, h\} \{b\}] \cap [\{b, c\} \{g\} \{b\}] = \emptyset; \\ & [\{b, c\} * \{a, c\}] \cap [\{a, d\} * \{b, c\}] = \emptyset; \\ & [\{b, c\} * \{a, c\}] \cap [\{b, d\} \{f, h\} \{a, c\}] = [\{b\} \{f, h\} \{a, c\}]; \\ & [\{b, c\} * \{a, c\}] \cap [\{b, c\} \{g\} \{b\}] = \emptyset. \end{aligned}$$

Затем из непустых C -кортежей формируем итоговую C -систему:

$$R_1 \cap R_2 = \left[\begin{array}{ccc} \{a, d\} & \{f, h\} & \{b\} \\ \{b\} & \{f, h\} & \{a, c\} \end{array} \right].$$

В этом случае получим следующую оценку: количество операций = $(3 \times 2 + 2 \times 3 + 1 \times 2) + (3 \times 2 + 2 \times 2 + 1 \times 2) + (3 \times 2 + 2 \times 1 + 1 \times 1) + (2 \times 2 + 3 \times 3 + 2 \times 2) + (2 \times 2 + 3 \times 2 + 2 \times 2) + (2 \times 2 + 3 \times 1 + 2 \times 1) = 75$. Каждое слагаемое, заключенное в скобки, характеризует пересечение определенной пары C -кортежей и вычисляется как сумма произведений мощностей их соответствующих компонент. Видно, что представление отношений в виде матрицеподобных структур способно существенно ускорить выполнение операций с отношениями.

Как будет показано далее, при использовании матрицеподобных структур операция соединения отношений также упрощается подобно рассмотренной ранее операции пересечения, и ее результат можно вычислить без попарного сравнения всех элементарных кортежей.

Среди разработанных алгоритмов удовлетворения нечисловых ограничений особый интерес представляют алгоритмы распространения ограничений, которые позволяют преобразовать описание задачи CSP за полиномиальное время к более простому виду с сохранением всех решений. Для C - и D -систем это означает, что из доменов переменных удаляются «лишние» значения, могут быть

удалены некоторые значения из компонент, некоторые столбцы и/или строки *C*-систем и *D*-систем, и даже *C*- и *D*-системы целиком.

4. Вывод на совокупности *C*-систем. Приведем утверждения, позволяющие реализовывать эквивалентные преобразования системы ограничений, представленной в виде совокупности *C*-систем [22]. Ранее эти утверждения были сформулированы и применены одним из авторов настоящей статьи для качественного моделирования статических аспектов функционирования технических систем.

Утверждение 1 (У1). Если все строки (кортежи) *C*-системы пусты, то есть содержат хотя бы по одной пустой компоненте каждая, то *C*-система пуста (соответствующая задача CSP несовместна).

Утверждение 2 (У2). Если все компоненты некоторого атрибута (столбца *C*-системы) являются полными, то данный атрибут можно удалить из *C*-системы (удаляются все компоненты, стоящие в соответствующем столбце), а пара «удаляемый атрибут – его домен» сохраняется в векторе частичного решения.

Утверждение 3 (У3). Если домен некоторого атрибута *C*-системы содержит значения, не встречающиеся в соответствующем столбце, то эти значения удаляются из данного домена.

Утверждение 4 (У4). Если строка *C*-системы содержит хотя бы одну пустую компоненту (строка пуста), то строка удаляется.

Утверждение 5 (У5). Если компонента некоторого атрибута содержит значение, не принадлежащее соответствующему домену, то это значение удаляется из компоненты.

Утверждение 6 (У6). Если одна строка *C*-системы полностью доминирует (покомпонентно содержит) другую строку, то доминируемая строка удаляется из *C*-системы.

Рассмотрим применение описанных утверждений при выполнении SPARQL запросов к онтологии ИПЗ, фрагмент которой показан на рисунке 2.

5. Обработка SPARQL-запроса к онтологии с помощью матриц ограничений. При использовании SPARQL онтология рассматривается как RDF-документ, состоящий из набора триплетов вида: «субъект — свойство — объект». Например, «Иванов-В-И — участвует-в — Исследование№213», «Иванов-В-И — выполняет-роль — Координатор». Каждый триплет можно рассматривать как дугу некоторого графа, а весь RDF-документ представлять в виде графа.

Типовой SPARQL-запрос на извлечение данных представляет собой конъюнкцию и/или дизъюнкцию шаблонов триплетов, в соответствии с которыми из онтологии выбираются удовлетворяющие им наборы триплетов, являющиеся решениями (Solutions). Решения

представляют собой подграфы графа онтологии. Например, запрос, состоящий из конъюнкции шаблонов: « $?x$ — является-экземпляром — Персона», « $?x$ — участвует-в — Исследование № 213» и « $?x$ — выполняет-роль — Координатор», позволяет найти подграфы, в которых вершина ($?x$) связана дугами «является-экземпляром», «участвует-в» и «выполняет-роль» с вершинами «Персона», «Исследование № 213» и «Координатор» соответственно. Выполнение запросов осуществляется обработчиком запросов, который обычно является компонентом хранилища триплетов (RDF — triple store). Наличие в запросе длинной связки шаблонов триплетов негативно влияет на скорость его выполнения [23].

Управление большим RDF-хранилищем представляет собой ресурсоемкую задачу с точки зрения физической организации хранилища, проведения индексирования триплетов, а также выполнения запросов. Самым простым решением является хранение всего множества триплетов в одной таблице, но с возрастанием объема данных и сложности запросов, включающих конъюнкцию нескольких шаблонов, появляются проблемы с производительностью, так как для ответа на запрос перебирается большое число связок триплетов. По этой причине обычно используют более сложные способы группировки триплетов в таблицы (cluster-property table), соответствующие:

- 1) одному свойству (предикату);
- 2) свойству (предикату) для сущностей одного класса;
- 3) комбинациям субъектов, предикатов и объектов, которые часто встречаются в большинстве запросов.

Помимо организации структуры RDF-хранилища, возникают сложности с оптимизацией выполнения запросов. При их выполнении приходится оперировать не записями, как в реляционной базе данных, а связками (join) триплетов различной длины, включающих труднопредсказуемые комбинации предикатов. С целью повышения производительности обработчик запроса задействует оптимизатор, который выбирает наиболее дешевый план выполнения запроса на основе функции оценки стоимости. Обычно данная функция учитывает количество промежуточных результатов, порождаемых связкой шаблонов триплетов в том или ином плане выполнения.

Представление RDF-хранилища в рамках предлагаемого подхода способно сократить расход памяти компьютера для хранения RDF-триплетов, а также повысить эффективность обработки SPARQL-запросов за счет структуризации знаний в виде специализированных матрицеподобных структур.

Рассмотрим применение методов распространения ограничений, разработанных ранее одним из авторов статьи, для ускорения обработки

SPARQL-запросов к онтологиям на примере получения ответа на квалификационный вопрос онтологического паттерна содержания для понятия «Процесс исследования» (Investigation): «Кто занимался исследованием магнитного поля (magnetic field)?». В соответствии с UML-схемой (рисунок 2) паттерна «Исследование» (Investigation), необходимо найти персон (экземпляры класса «Person»), реализовавших (отношение «realizes-in») свои роли исследователей (экземпляры класса «Researcher-role») в процессе исследования (экземпляры класса «Investigation»), в котором также была реализована роль объекта исследования (экземпляры класса «InvObjectRole»), носителем которой (отношение «bearer-of») выступил объект «magnetic-field» — экземпляр класса «Entity».

Соответствующий SPARQL-запрос будет иметь вид (листинг 1):

```
PREFIX IKS: < http://www.iimm.ru/ontologies/oiks>
PREFIX INV: < http://www.iimm.ru/ontologies/cdp/investigation-process>
SELECT ?person
WHERE {
?person rdf:type IKS:Person.
?person IKS:bearer-of ?role.
?role rdf:type INV:Researcher-role.
?role IKS:realizes-in ?inv.
?inv IKS: realizes ?objRole.
?objRole IKS:bearer-of "magnetic-field".
}
```

Листинг 1. Пример SPARQL-запроса

Рассмотрим его выполнение на конкретном наборе данных (рисунок 3).

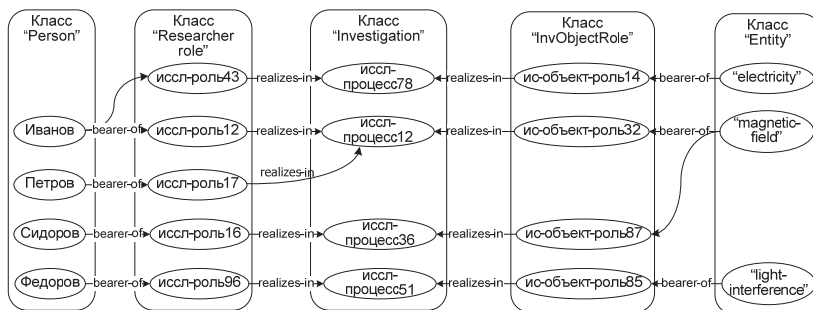


Рис. 3. Примерный фрагмент онтологии ИПЗ

Приведенный фрагмент онтологии включает набор экземпляров, связанных отношениями, используемыми для описания понятия

«Исследование» (Investigation). Для наглядности во фрагменте присутствуют только те классы и отношения, которые представлены в рассматриваемом SPARQL-запросе.

Для классов, экземпляров классов и связей показанного на рисунке 3 фрагмента онтологии введем специальные обозначения. Каждому классу фрагмента онтологии сопоставим определенную переменную (атрибут), а каждому экземпляру класса определенное значение данной переменной так, как это показано в таблице 1.

Таблица 1. Обозначение классов и экземпляров классов

Названия классов и экземпляров классов	Обозначение классов	Обозначение значений переменных (экземпляров классов)
КЛАСС «PERSON» «Иванов» «Петров» «Сидоров» «Федоров»	<i>X</i>	<i>a</i> ₁ <i>a</i> ₂ <i>a</i> ₃ <i>a</i> ₄
КЛАСС «RESEARCHER_ROLE» «иссл-роль43» «иссл-роль12» «иссл-роль17» «иссл-роль16» «иссл-роль96»	<i>Y</i>	<i>b</i> ₁ <i>b</i> ₂ <i>b</i> ₃ <i>b</i> ₄ <i>b</i> ₅
КЛАСС «INVESTIGATION» «иссл-процесс78» «иссл-процесс12» «иссл-процесс36» «иссл-процесс51»	<i>Z</i>	<i>c</i> ₁ <i>c</i> ₂ <i>c</i> ₃ <i>c</i> ₄
КЛАСС «INV_OBJ_ROLE» «исс-объ-роль14» «исс-объ-роль32» «исс-объ-роль87» «исс-объ-роль85»	<i>W</i>	<i>d</i> ₁ <i>d</i> ₂ <i>d</i> ₃ <i>d</i> ₄
КЛАСС «ENTITY» «electricity» «magnetic-field» «light-interference»	<i>L</i>	<i>e</i> ₁ <i>e</i> ₂ <i>e</i> ₃

Тогда в пределах рассматриваемого фрагмента получим следующие соответствия «атрибут — домен атрибута»:

$$X - \{a_1, a_2, a_3, a_4\}, Y - \{b_1, b_2, b_3, b_4, b_5\}, Z - \{c_1, c_2, c_3, c_4\}, W - \{d_1, d_2, d_3, d_4\}, L - \{e_1, e_2, e_3\}.$$

Теперь каждой связи онтологии, которая фигурирует в рассматриваемом SPARQL-запросе, сопоставим ограничение:

$R_1[XY]$ — отношение «bearer-of» между классами X («PERSON») и Y («RESEARCHER_ROLE»);

$R_2[YZ]$ — отношение «realizes-in» между классами Y («RESEARCHER_ROLE») и Z («INVESTIGATION»);

$R_3[WZ]$ — отношение «realizes-in» между классами W («INV_OBJ_ROLE») и Z («INVESTIGATION»);

$R_4[LW]$ — отношение «bearer-of» между классами L («ENTITY») и W («INV_OBJ_ROLE»);

Также для моделирования селектора SPARQL-запроса введем следующее унарное ограничение:

$R_5[L]$ — ограничение на переменную L (КЛАСС «ENTITY»), задающее условие фильтрации (селектор) в SPARQL-запросе.

На рисунке 3 перечисленные отношения $R_1 — R_4$ обозначены стрелками. В соответствии с рисунком 3, учитывая конкретные экземпляры классов, можно записать перечисленные выше отношения в виде предлагаемых матриц ограничений, а именно в виде C -систем:

Отношение $R_1[XY]$:

$$\begin{bmatrix} \{a_1\} & \{b_1, b_2\} \\ \{a_2\} & \{b_3\} \\ \{a_3\} & \{b_4\} \\ \{a_4\} & \{b_5\} \end{bmatrix}.$$

Отношение $R_2[YZ]$:

$$\begin{bmatrix} \{b_1\} & \{c_1\} \\ \{b_2, b_3\} & \{c_2\} \\ \{b_4\} & \{c_3\} \\ \{b_5\} & \{c_4\} \end{bmatrix}.$$

Отношение $R_3[WZ]$:

$$\begin{bmatrix} \{d_1\} & \{c_1\} \\ \{d_2\} & \{c_2\} \\ \{d_3\} & \{c_3\} \\ \{d_4\} & \{c_4\} \end{bmatrix}.$$

Отношение $R_4[LW]$:

$$\begin{bmatrix} \{e_1\} & \{d_1\} \\ \{e_2\} & \{d_2, d_3\} \\ \{e_3\} & \{d_4\} \end{bmatrix}.$$

На примере матрицы R_1 поясним, что обозначает подобная запись. Первая строка данной матрицы соответствует двум стрелкам (дугам) на рисунке 3, проведенным от экземпляра «Иванов» (значение a_1) класса «PERSON» (переменная X) к экземплярам «иссл-роль43» (значение b_1) и «иссл-роль12» (значение b_2) класса «RESEARCHER_ROLE» (переменная Y). По аналогии, каждой строке всех перечисленных C -систем соответствует некоторое множество дуг между экземплярами соответствующих классов онтологии, изображенных на рисунке 3.

Кроме того, известно, что на переменную L в SPARQL-запросе наложено еще одно *дополнительное ограничение* $R_5[L]$: $R_5[L] = [\{e_2\}]$.

За счет представления RDF-триплетов в виде совокупности матрицеподобных структур повышается эффективность обработки запросов к онтологии, как это будет продемонстрировано далее.

В настоящей работе предлагается задачу обработки запроса к онтологии рассматривать как задачу удовлетворения ограничений. Приводимый ранее пример запроса в терминах матриц ограничений будет выглядеть следующим образом:

$$Pr_X (R_1[XY] \bowtie R_2[YZ] \bowtie R_3[WZ] \bowtie R_4[LW] \bowtie R_5[L]).$$

Представим SPARQL-запрос в виде сети ограничений (рисунок 4):

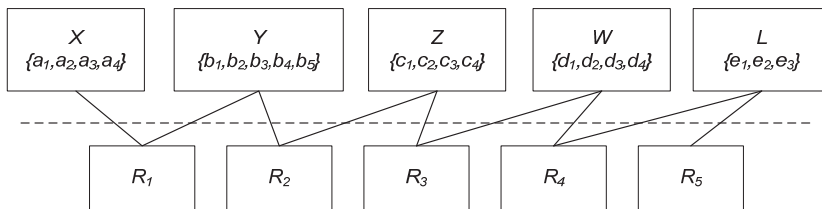


Рис. 4. Исходная задача CSP

Обработка запроса производится на основе приведенных ранее утверждений для совокупности S -систем.

Сеть ограничений является двудольным графом: одна из долей графа — это множество переменных и текущие области определения этих переменных (домены переменных), другая доля — сами ограничения. Ребро связывает две вершины (переменную и ограничение) тогда и только тогда, когда эта переменная формирует диапазон данного ограничения (входит в схему данного отношения).

Далее проиллюстрируем применение авторского метода распространения ограничений к обработке данного запроса, то есть для решения представленной задачи CSP.

При распространении ограничений обычно сначала обрабатывают унарные ограничения, поскольку с их помощью можно непосредственно «сузить» домены переменных. Затем уже активируются бинарные ограничения, поскольку их обработка немного более трудоемка. Алгоритмы удовлетворения унарных и бинарных ограничений выполняются за полиномиальное время, поэтому задачи с множественными отношениями часто сводят к совокупности унарных и бинарных отношений. Активация ограничений осуществляется по событиям. К таким событиям относится «усечение» домена. Вполне типична ситуация, когда на основе анализа определенного ограничения «усекается» домен некоторой переменной, а затем активируется другое ограничение, содержащее в схеме данную переменную. В ходе анализа вновь активированного ограничения может быть «усечен» домен другой переменной, состоящей в том же отношении.

Для нашего примера распишем данную схему вывода по шагам.

Сначала активируется унарное ограничение $R_5[L]$. Удаляем из домена переменной L все значения, кроме значения e_2 (применяем УЗ). После чего само ограничение $R_5[L]$ исключается из рассмотрения по У2, поскольку единственная компонента этой S -системы становится полной (совпадает с новым доменом переменной L).

Поскольку рассматриваемая задача CSP не содержит больше унарных ограничений (все унарные ограничения уже удовлетворены),

то следующим активируется одно из бинарных ограничений. На текущем шаге выбирается ограничение $R_4[LW]$, так как домен его атрибута L был изменен на предыдущем этапе вывода:

$$\begin{array}{cc} L & W \\ \{e_2\} & \{d_1, d_2, d_3, d_4\} \\ 1 \left[\begin{array}{cc} \{e_1\} & \{d_1\} \\ \{e_2\} & \{d_2, d_3\} \\ \{e_3\} & \{d_4\} \end{array} \right]. \end{array}$$

Для удобства объяснения здесь и далее в верхних двух строках S -системы записываются имена атрибутов и домены атрибутов, а строки S -системы пронумерованы.

Из компонент первого столбца данной S -системы, применяя У5, исключаем значения, не принадлежащие усеченному домену атрибута L . В результате в первом столбце в строках 1 и 3 образуются пустые компоненты, то есть компоненты, не содержащие ни одного значения. Далее, пользуясь У4, удаляем строки 1 и 3. В процессе упрощений ограничение $R_4[LW]$ преобразуется в ограничение:

$$\begin{array}{cc} L & W \\ \{e_2\} & \{d_1, d_2, d_3, d_4\} \\ 2 \left[\begin{array}{cc} * & \{d_2, d_3\} \end{array} \right]. \end{array}$$

Согласно У3, удаляем из домена атрибута W значения, отличные от d_2 и d_3 . После чего обе компоненты второй строки становятся полными и по У2 можно элиминировать оба столбца рассматриваемой S -системы. Значит, данное ограничение исключается из рассмотрения. Сеть ограничений будет выглядеть следующим образом (рисунок 5):

Частичное решение: $L \text{ — } \{e_2\}$.

Остаток:

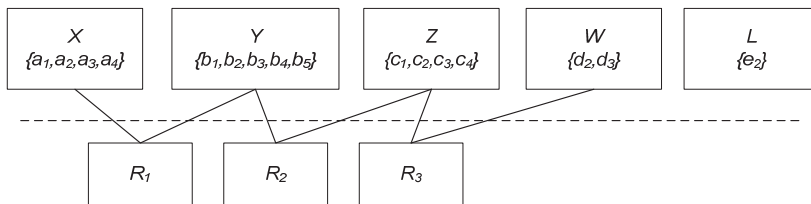


Рис. 5. Редуцированная сеть ограничений

Далее количество вершин и ребер в сети ограничений не изменяется, поэтому дальнейшие шаги процедуры распространения не будем представлять графически.

Следующим активируется ограничение $R_3[WZ]$, поскольку был изменен домен атрибута W , присутствующий в схеме данного отношения. Отношение $R_3[WZ]$ на текущем шаге выглядит так:

$$\begin{array}{cc}
 W & Z \\
 \{d_2, d_3\} & \{c_1, c_2, c_3, c_4\} \\
 1 \left[\begin{array}{cc} \{d_1\} & \{c_1\} \\ \{d_2\} & \{c_2\} \\ \{d_3\} & \{c_3\} \\ \{d_4\} & \{c_4\} \end{array} \right].
 \end{array}$$

Применяем У5 для того, чтобы исключить из первого столбца значения, не принадлежащие домену атрибута W . В первом столбце в строках 1 и 4 образуются пустые компоненты, что позволяет удалить данные строки из рассматриваемой C -системы, пользуясь У4.

C -система преобразуется к виду:

$$\begin{array}{cc}
 W & Z \\
 \{d_2, d_3\} & \{c_1, c_2, c_3, c_4\} \\
 2 \left[\begin{array}{cc} \{d_2\} & \{c_2\} \\ \{d_3\} & \{c_3\} \end{array} \right].
 \end{array}$$

Анализируя второй столбец, можно сузить домен переменной Z до множества $\{c_2, c_3\}$ (утверждение У3). Более никаких операций упрощения к данной C -системе применить нельзя.

Поскольку последние изменения коснулись домена атрибута Z , то следующим активируется ограничение $R_2[YZ]$, содержащее в схеме отношения данный атрибут:

$$\begin{array}{cc}
 Y & Z \\
 \{b_1, b_2, b_3, b_4, b_5\} & \{c_2, c_3\} \\
 1 \left[\begin{array}{cc} \{b_1\} & \{c_1\} \\ \{b_2, b_3\} & \{c_2\} \\ \{b_4\} & \{c_3\} \\ \{b_5\} & \{c_4\} \end{array} \right] .
 \end{array}$$

Пользуясь утверждением У5, исключаем из второго столбца матрицы ограничений значения, которые не принадлежат новому домену переменной Z . В результате строки 1 и 4 матрицы элиминируются. Анализ первых компонент двух оставшихся строк 2 и 3 позволяет на основании У3 сузить домен атрибута Y до множества $\{b_2, b_3, b_4\}$. Тогда ограничение $R_2[YZ]$ примет вид:

$$\begin{array}{cc}
 Y & Z \\
 \{b_2, b_3, b_4\} & \{c_2, c_3\} \\
 2 \left[\begin{array}{cc} \{b_2, b_3\} & \{c_2\} \\ \{b_4\} & \{c_3\} \end{array} \right] .
 \end{array}$$

На заключительном этапе активируется ограничение $R_1[XY]$. После применения правил редукции, ограничение $R_1[XY]$ примет вид:

$$\begin{array}{cc}
 X & Y \\
 \{a_1, a_2, a_3\} & \{b_2, b_3, b_4\} \\
 1 \left[\begin{array}{cc} \{a_1\} & \{b_2\} \\ \{a_2\} & \{b_3\} \\ \{a_3\} & \{b_4\} \end{array} \right] .
 \end{array}$$

В результате получим следующую упрощенную задачу CSP, к которой более не применить ни одного из представленных ранее утверждений (правил редукции), то есть процесс распространения ограничений останавливается, достигнув неподвижной точки:

Области определения переменных $X - \{a_1, a_2, a_3\}$, $Y - \{b_2, b_3, b_4\}$, $Z - \{c_2, c_3\}$, $W - \{d_1, d_2\}$, $L - \{e_2\}$.

Ограничения

Отношение $R_1[XY]$:

$$\begin{bmatrix} \{a_1\} & \{b_2\} \\ \{a_2\} & \{b_3\} \\ \{a_3\} & \{b_4\} \end{bmatrix}.$$

Отношение $R_2[YZ]$:

$$\begin{bmatrix} \{b_2, b_3\} & \{c_2\} \\ \{b_4\} & \{c_3\} \end{bmatrix}.$$

Отношение $R_3[WZ]$:

$$\begin{bmatrix} \{d_2\} & \{c_2\} \\ \{d_3\} & \{c_3\} \end{bmatrix}.$$

Таким образом, процедура распространения ограничений позволила получить более простую задачу CSP по сравнению с исходной: 1) количество ограничений уменьшилось с пяти до трех; 2) суммарное количество значений в доменах переменных уменьшилось с $4 + 5 + 4 + 4 + 3 = 20$ до $3 + 3 + 2 + 2 + 1 = 11$; 3) в каждом из оставшихся ограничений стало меньше строк; 4) некоторые из оставшихся строк содержат редуцированные компоненты, в частности вторая компонента первой строки первого отношения $R_1[XY]$.

При необходимости выполнить соединение упрощенных отношений $R_1[XY]$, $R_2[YZ]$ и $R_3[WZ]$ потребуется проспрягать всего $3 \times 2 \times 2 = 12$ строк C -систем в отличие от первоначальной задачи CSP, где требовалось $4 \times 4 \times 4 \times 3 \times 1 = 192$ подобных операций.

Представленный пример наглядно доказывает, что препроцессинг задачи на основе авторских методов удовлетворения ограничений позволяет существенно снизить размерность решаемой задачи, а иногда и получить ее окончательное решение. В частности, в рассмотренном нами примере SPARQL-запроса в процессе распространения ограничений получено окончательное решение: $X - \{a_1, a_2, a_3\}$. Интересующими нас исследователями магнитного поля являются: Иванов, Петров, Сидоров.

Численные результаты экспериментальной проверки подтвердили полученные в ходе теоретического исследования выводы. В настоящей работе они не приводятся ввиду ограничений на объем публикации.

6. Заключение. Использование онтологических паттернов для формирования онтологий различных областей знания, ориентированных на включение в интегрированное пространство междисциплинарных знаний, существенно облегчает решение задачи обеспечения семантической интероперабельности интегрируемых онтологий. Однако при этом в рамках ИПЗ ожидается быстрый рост числа представленных в онтологиях сущностей и связей, что влечет снижение производительности при обработке запросов к ИПЗ. По сравнению с традиционным подходом к обработке SPARQL-запросов, базирующимся на динамическом программировании, предлагаемый в статье метод обработки запросов позволяет ускорить их выполнение за счет использования специальных матрицеподобных структур для «компактного» представления онтологии, а также применения авторских методов удовлетворения ограничений для организации эффективного вывода на онтологии.

В дальнейшем предполагается модифицировать контекстно-ориентированный подход к управлению ограничениями [24, 25] на задачу интеграции разнородных знаний и данных с целью повышения релевантности извлекаемой информации и скорости обработки запросов конечного пользователя за счет активации только актуальных текущей ситуации компонентов онтологической модели.

Литература

1. *Gangemi A.* Ontology Design Patterns for Semantic Web Content // Proceedings of the Fourth International Semantic Web Conference. Galway. Ireland. Springer. 2005. pp. 262–276.
2. *Олейник А.Г., Ломов П.А.* О формировании единого пространства мультипредметных знаний Кольского научного центра РАН // Труды пятой международной конференции «Системный анализ и информационные технологии» (САИТ-2013). 2013. Т.1. С. 258–265.
3. *Nikolić N. et al.* RDF Stores Performance Test on Servers with Average Specification // Proceedings of the 5th International Conference on Information Science and Technology (ICIST). 2015. pp. 67–72.
4. *Neumann T., Weikum G.* RDF3X: a RISCstyle engine for RDF // Proceedings of the 34th International Conference on Very Large Data Bases (VLDB). Auckland. New Zealand. 2008. pp. 647–659.
5. *Зуенко А.А.* Вывод на ограничениях с применением матричного представления конечных предикатов // Искусственный интеллект и принятие решений. 2014. № 3. С. 21–31.
6. *Smith B. et al.* IAO-Intel: An Ontology of Information Artifacts in the Intelligence Domain // Proceedings of the Eighth International Conference on Semantic Technologies for Intelligence, Defense and Security (STIDS), (CEUR 2013). 2013. vol. 1097. pp. 33–40.
7. Basic Formal Ontology. URL: <http://ifomis.uni-saarland.de/bfo/> (дата обращения: 10.11.2016).
8. *Lomov P., Shishaev M.* Ad-hoc Synthesis of Composite Content Ontology Design Patterns // Proceedings of the 26th International Conference on Information Modelling and Knowledge Bases (EJC 2016). Tampere University of Technology. Pori. 2016. pp. 25–33.

9. *Blomqvist E., Hammar K., Presutti V.* Engineering Ontologies with Patterns: The eXtreme Design Methodology // *Ontology Engineering with Ontology Design Patterns*. IOS Press. 2016. vol. 25. pp. 23–50.
10. *Blomqvist E., Presutti V., Daga E., Gangemi A.* Experimenting with eXtreme Design // *Proceedings of EKAW 2010*. Springer. Berlin-Heidelberg-New York. 2010. LNCS 6317. pp. 120–134.
11. *Ontology of Integrated Knowledge Space*. URL: <https://github.com/palandlom/ontology-of-integrated-knowledge-space>. (дата обращения: 11.11.2016).
12. *Schmidt M., Meier M., Lausen G.* Foundations of SPARQL query optimization // *Proceedings of the 13th International Conference on Database Theory*. 2010. .pp. 4–33.
13. *Tsialiamanis P. et al.* Heuristics-based query optimisation for SPARQL // *Proceedings of the 15th International Conference on Extending Database Technology (EDBT '12)*. ACM. 2012. pp. 324–335.
14. *Aranda C., Arenas M., Corcho M.* Semantics and Optimization of the SPARQL 1.1 Federation Extension The Semantic Web: Research and Applications // *8th Extended Semantic Web Conference*. Springer Berlin Heidelberg. 2011. pp. 1–15.
15. *Gubichev A., Neumann T.* Exploiting the query structure for efficient join ordering in SPARQL queries // *Proceedings of the 17th international conference on Extending Database Technology*. 2014. pp. 439–450.
16. *Bartak R.* Constraint Programming: In Pursuit of the Holy Grail // *Proceedings of the Week of Doctoral Students (WDS99)*. 1999. Part IV. pp. 555–564.
17. *Ruttkay Zs.* Constraint satisfaction a survey // *CWI Quarterly*. 1998. vol. 11. pp. 163–214.
18. *Russel S., Norvig P.* Artificial Intelligence: A Modern Approach. 3rd edition // Prentice Hall. 2010. 1132 p.
19. *Щербина О.А.* Удовлетворение ограничений и программирование в ограничениях // *Интеллектуальные системы*. 2011. Т. 15. №. 1–4. С. 54–73.
20. *Kulik B.A., Zuenko A.A., Friedman A.Ya.* Deductive and Defeasible Reasoning on the Basis of a Unified Algebraic Approach // *Scientific and Technical Information Processing*. 2015. vol. 42. no. 6. pp. 8–16.
21. *Zakrevskij A.* Integrated Model of Inductive-Deductive Inference Based on Finite Predicates and Implicative Regularities // *Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems*. IGI Global. 2013. pp. 1–12.
22. *Зуенко А.А.* Качественное моделирование технических систем на основе методов распространения ограничений // *Открытые семантические технологии проектирования интеллектуальных систем: мат. VI Междунар. науч.-техн. конф.* Минск. БГУИР. 2016. С.573–578.
23. *Neumann T., Moerkotte G.* Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins // *ICDE*. 2011. pp. 984–994.
24. *Smirnov A.V., Levashova T.V., Shilov N.G., Krizhanovskiy A.A.* Knowledge Fusion in Context-Aware Decision Support: Ontology-Based Modeling and Patterns // *Studies in Fuzziness and Soft Computing*. 2014. vol. 314. pp. 35–51.
25. *Smirnov A., Levashova T., Shilov N.* Knowledge Fusion in Context-Aware Decision Support Systems // *KEOD 2014 — Proceedings of the International Conference on Knowledge Engineering and Ontology Development*. 2014. pp. 186–194.

Зуенко Александр Анатольевич — к-т техн. наук, старший научный сотрудник лаборатории информационных технологий управления промышленно-природными системами, Федеральное государственное бюджетное учреждение науки Институт информатики и математического моделирования технологических процессов Кольского научного центра Российской академии наук (ИИММ КНЦ РАН). Область научных интересов: представление и обработка знаний в слабо формализованных предметных областях, программирование в ограничениях. Число научных публикаций — 104.

zuenko@iimm.ru; ул. Ферсмана, 24а, Мурманская обл., Апатиты, 184209; р.т.: +78155574050, Факс: +78155574050.

Ломов Павел Андреевич — к-т техн. наук, научный сотрудник лаборатории региональных информационных систем, Федеральное государственное бюджетное учреждение науки Институт информатики и математического моделирования технологических процессов Кольского научного центра Российской академии наук (ИИММ КНЦ РАН). Область научных интересов: представление знаний, онтологическое моделирование, Semantic web, информационная безопасность. Число научных публикаций — 48. lomov@iimm.ru; ул. Ферсмана, 24а, Мурманская обл., Апатиты, 184209; р.т.: +79522967633, Факс: +7(81555)74050.

Олейник Андрей Григорьевич — д-р техн. наук, заместитель директора по научной работе, Федеральное государственное бюджетное учреждение науки Институт информатики и математического моделирования технологических процессов Кольского научного центра Российской академии наук (ИИММ КНЦ РАН). Область научных интересов: информационные технологии, компьютерное моделирование, базы данных, системы поддержки принятия решений. Число научных публикаций — 166. oleynik@iimm.ru, <http://www.iimm.ru/users/oleynik>; ул. Ферсмана, 24а, Мурманская обл., Апатиты, 184209; р.т.: +7155579220, Факс: +7155574050.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ (проекты №№ 15-07-03321а, 16-07-00377а, 16-07-00562а).

A.A. ZUENKO, P.A. LOMOV, A.G. OLEYNIK
**APPLICATION OF CONSTRAINT PROPAGATION TECHNIQUES
 TO SPEED UP PROCESSING OF QUERIES TO ONTOLOGIES**

Zuenko A.A., Lomov P.A., Oleynik A.G. Application of Constraint Propagation Techniques to Speed up Processing of Queries to Ontologies.

Abstract. We propose to regard a problem of processing of SPARQL queries to the ontology as a constraint satisfaction problem. The formal apparatus for formalization of constraint satisfaction problems by means of specialized matrix-like structures is briefly given. The use of this apparatus allows one to describe, store and process non-numeric constraints of the subject domain more efficiently compared with the table representation. To speed up processing of queries to big ontologies it is proposed to apply the non-numeric constraint propagation technique previously developed by the authors. Compared with the traditional approach to processing of the SPARQL queries, which is based on dynamic programming, the proposed technique allows one to accelerate their performance by "compact" representation of ontology, as well as by using the original rules of the search space reduction. Application of the mathematical apparatus for reducing the dimension of the search space at processing a query to the ontology of multidisciplinary knowledge is exemplified.

Keywords query to the ontology, constraint satisfaction problem, a matrix representation of constraint, constraint propagation.

Zuenko Alexander Anatolievich — Ph.D., senior researcher of information technologies for nature-industrial systems control laboratory, Institute for Informatics and Mathematical Modelling of Technological Processes of the Kola Science Center of Russian Academy of Sciences (IIMM KSC RAS). Research interests: knowledge representation and processing in poorly formalized subject domains, constraint programming. The number of publications — 104. zuenko@iimm.ru; 24A, Fersman st., 184209, Apatity, Murmansk Region, Russia; office phone: +78155574050, Fax: +78155574050.

Lomov Pavel Andreevich — Ph.D., researcher of regional information systems laboratory, Institute for Informatics and Mathematical Modelling of Technological Processes of the Kola Science Center of Russian Academy of Sciences (IIMM KSC RAS). Research interests: knowledge representation, ontological modeling, Semantic web. The number of publications — 48. lomov@iimm.ru; 24A, Fersman st., 184209, Apatity, Murmansk Region, Russia; office phone: +79522967633, Fax: +7(81555)74050.

Oleynik Andry Gigorievich — Ph.D., Dr. Sci., deputy director, Institute for Informatics and Mathematical Modelling of Technological Processes of the Kola Science Center of Russian Academy of Sciences (IIMM KSC RAS). Research interests: information technologies, computer modeling, databases, systems of decision support. The number of publications — 166. oleynik@iimm.ru, <http://www.iimm.ru/users/oleynik>; 24A, Fersman st., 184209, Apatity, Murmansk Region, Russia; office phone: +7155579220, Fax: +7155574050.

Acknowledgements. This research is supported by RFBR (grants №№ 15-07-03321a, 16-07-00377a, 16-07-00562a).

References

1. Gangemi A. Ontology Design Patterns for Semantic Web Content. Proceedings of the Fourth International Semantic Web Conference. Galway, Ireland. Springer. 2005. pp. 262–276.
2. Olejnik A.G. [About the formation of integrated multi-disciplinary knowledge space]. *Trudy pjatoj mezhdunarodnoj konferencii "Sistemnyj analiz i informacionnyje tehnologii" (SAIT-2013)* [Proceedings of the Fifth International Conference "System Analysis and Information Technologies" (SAIT 2013): Proceedings of the conference]. 2013. Issue 1. pp. 258–265. (In Russ.).
3. Nikolić N. et al. RDF Stores Performance Test on Servers with Average Specification. Proceedings of the 5th International Conference on Information Science and Technology (ICIST). 2015. pp. 67–72.
4. Neumann T., Weikum G. RDF3X: a RISCstyle engine for RDF. Proceedings of the 34th International Conference on Very Large Data Bases (VLDB). Auckland, New Zealand. 2008. pp. 647–659.
5. Zuenko A.A. [Constraint inference based on the matrix representation of finite predicates]. *Iskustvennui intellekt I prinyatie reshenii – Artificial intelligence and decision-making*. 2014. vol. 3. pp. 21–31. (In Russ.).
6. Smith B. et al. IAO-Intel: An Ontology of Information Artifacts in the Intelligence Domain. Proceedings of the Eighth International Conference on Semantic Technologies for Intelligence, Defense and Security (STIDS), (CEUR 2013). 2013. vol. 1097. pp. 33–40.
7. Basic Formal Ontology. Available at: <http://ifomis.uni-saarland.de/bfo/> (accessed 10.11.2016).
8. Lomov P., Shishaev M. Ad-hoc Synthesis of Composite Content Ontology Design Patterns. Proceedings of the 26th International Conference on Information Modelling and Knowledge Bases (EJC 2016). Tampere University of Technology. Pori. 2016. pp. 25–33.
9. Blomqvist E., Hammar K., Presutti V. Engineering Ontologies with Patterns: The eXtreme Design Methodology. In: *Ontology Engineering with Ontology Design Patterns*. IOS Press. 2016. vol. 25. pp. 23–50.
10. Blomqvist E., Presutti V., Daga E., Gangemi A. Experimenting with eXtreme Design. Proceedings of EKAW 2010. Springer. Berlin-Heidelberg-New York. 2010. LNCS 6317. pp. 120–134.
11. Ontology of Integrated Knowledge Space. Available at: <https://github.com/palandlom/ontology-of-integrated-knowledge-space>. (accessed 11.11.2016).
12. Schmidt M., Meier M., Lausen G. Foundations of SPARQL query optimization. Proceedings of the 13th International Conference on Database Theory. 2010. pp. 4–33.
13. Tsialiamanis P. et al. Heuristics-based query optimisation for SPARQL. Proceedings of the 15th International Conference on Extending Database Technology (EDBT '12). ACM. 2012. pp. 324–335.
14. Aranda C., Arenas M., Corcho O. Semantics and Optimization of the SPARQL 1.1 Federation Extension The Semantic Web: Research and Applications. 8th Extended Semantic Web Conference. Springer Berlin Heidelberg. 2011. pp. 1–15.
15. Gubichev A., Neumann T. Exploiting the query structure for efficient join ordering in SPARQL queries. Proceedings of the 17th international conference on Extending Database Technology. 2014. pp. 439–450.
16. Bartak R. Constraint Programming: In Pursuit of the Holy Grail. Proceedings of the Week of Doctoral Students (WDS99). 1999. Part IV. pp. 555–564.
17. Ruttkay Zs. Constraint satisfaction a survey. *CWI Quarterly*. 1998. vol. 11. pp. 163–214.
18. Russel S., Norvig P. *Artificial Intelligence: A Modern Approach*. 3rd edition. Prentice Hall. 2010. 1132 p.

19. Shherbina O. A. [Constraint propagation and constraint programming]. *Intellektual'nye sistemy – Intelligent systems*. 2011. vol. 15. no. 1–4. pp. 54–73. (In Russ.).
20. Kulik B.A., Zuenko A.A., Friedman A.Ya. Deductive and Defeasible Reasoning on the Basis of a Unified Algebraic Approach. *Scientific and Technical Information Processing*. 2015. vol. 42. no. 6. 2015. pp. 8–16.
21. Zakrevskij A. Integrated Model of Inductive-Deductive Inference Based on Finite Predicates and Implicative Regularities. In *Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems*. IGI Global. 2013. pp. 1–12.
22. Zuenko A.A. [Qualitative modeling of technical systems based on the constraint propagation techniques] *Otkrytye semanticheskie tehnologii proektirovaniya intellektual'nyh sistem (OSTIS-2016): mat. VI Mezhdunar. nauchn.-tehn. konf.* [Open Semantic Technologies for Intelligent Systems (OSTIS-2016): proceedings of the VI International scientific and technical conference]. Minsk: BGUIR. 2016. pp. 573–578. (In Russ.).
23. Neumann T., Moerkotte G. Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins. *ICDE*. 2011. pp. 984–994.
24. Smirnov A.V., Levashova T.V., Shilov N.G., Krizhanovsky A.A. Knowledge Fusion in Context-Aware Decision Support: Ontology-Based Modeling and Patterns. *Studies in Fuzziness and Soft Computing*. 2014. vol. 314. pp. 35–51.
25. Smirnov A., Levashova T., Shilov N. Knowledge Fusion in Context-Aware Decision Support Systems. *Proceedings of the International Conference on Knowledge Engineering and Ontology Development – 2014 (KEOD 2014)*. pp. 186–194.