АЛГЕБРАИЧЕСКИЙ ПОДХОД К ТРАНСЛЯЦИИ ОНТОЛОГИЙ

Ермолаев В. А., к.ф.-м.н., доцент, Кеберле Н. Г., ассистент,
Маляр Е. Н., студент, Плаксин С. Л., аспирант

Запорожский государственный университет

В статье предлагается подход к переводу текстов онтологий с одного формализованного языка на другой, с сохранением смысла концепций онтологии. Для построения перевода в подходе используется формальный аппарат общей алгебры. Применимость предложенного алгебраического подхода иллюстрируется на примере перевода онтологий, записанных на языке L_{acm} описания «легковесных» онтологий предметной области «Публикации в области Computer Science», в онтологии на целевом языке древовидных структур L_{tree}, который используется в проекте RACING. В статье определяются формальные грамматики обоих языков, а также отношение перевода между ними. В статье показана применимость данного подхода для построения одного из множества возможных переводов между формализованными языками для описания «легковесных» онтологий.

Ключевые слова: онтология, классификация, формальная трансляция, семантика, алгебраический подход, ACM, RACING

Ермолаев В. А, Кеберле Н. Г., Маляр Е. Н., Плаксин С. Л.,АЛГЕБРАІЧНИЙ ПІДХІД ДО ТРАНСЛЯЦІЇ ОНТОЛОГІЙ/ Запорізький державний університет

У статті пропонується підхід до перекладу текстів онтологій з однієї формалізованої мови на іншу, зі збереженням змісту концепцій онтології. Для побудови перекладу в підході використовується формальний апарат загальної алгебри. Застосовність запропонованого алгебраїчного підходу ілюструється на прикладі перекладу онтологій, записаних мовою L_{acm} опису «легковагих» онтологій предметної області «Публікації в галузі Computer Science», в онтології цільовою мовою деревоподібних структур L_{tree}, що використовується в проекті RACING. У статті визначаються формальні граматики обох мов, а також відношення перекладу між ними. У статті показана застосовність даного підходу для побудови одного з множини можливих перекладів між формалізованими мовами для опису «легковагих» онтологій.

Ключові слова: онтологія, класифікація, формальна трансляция, семантика, алгебраічний підхід, ACM, RACING

Ermolayev, V. A., Keberle, N. G., Malyar, E. N., Plaksin. S. L. ALGEBRAIC APPROACH TO ONTOLOGY TRANSLATION/ Zaporozhye Stete University

Proposed is the approach to ontology translation from one formal ontology specification language to another preserving the semantics of ontology elements. Formal means of general algebra are used for translation mapping construction. Applicability of the proposed approach is illustrated by the example translation of ontologies formalized by means of L_{acm} language into the target L_{tree} language. L_{acm} is the DAML namespace for specification of the lightweight ontologies in “Publications in Computer Science” domain. Tree structures specification language L_{tree} is used in RACING project. Formal grammars for both languages and the translation mapping are defined in the paper. It is also shown that the proposed approach is applicable for the building of one of the possible translation mappings (from the set of the possible translations) between formal languages for lightweight ontologies specification.

Keywords: ontology, classification, formal translation, semantics, algebraic approach, ACM, RACING.

Эффективность извлечения необходимой информации из всемирной сети существенным образом ограничивается несовершенством используемых сегодня механизмов поиска, основанных на ключевых словах. Данные механизмы позволяют учитывать только синтаксические различия между критерием поиска и соответствующей информацией в документе. Смысловые или семантические несоответствия между понятиями, которые вкладывает пользователь в ключевые слова своего критерия поиска, и

понятиями, заложенными в слова документа его создателем, при этом не рассматриваются. Именно семантические несоответствия приводят порой к результатам поиска, совершенно неадекватным желанию пользователя. Факт наличия смысловых различий синтаксически схожих терминов приобретает еще большее значение для реально распределенных информационных систем, в которых информационные ресурсы сопровождаются независимыми владельцами, не имеющими возможности и не стремящимися согласовывать свою терминологию. Важным фактором также является наличие различных групп пользователей, для представителей которых синтаксически сходные термины имеют абсолютно различный смысл. Так, например, для специалистов, занятых в сфере информационных технологий, термин «агент» означает компьютерную программу, имеющую некоторые интеллектуальные свойства. Тогда как для служащих в рекламной сфере этот же термин обозначает специалиста, умеющего продать гражданам товар, который им абсолютно не нужен.

Информационные ресурсы одного владельца редко дают полную информацию о состоянии исследуемой проблемы. Так, например, для подготовки данной статьи потребовались источники нескольких авторов (см. список использованной литературы). Поэтому, одной из важных задач информационного поиска является проблема интеграции информационных ресурсов. Из предыдущих замечаний непосредственно вытекает, что такая интеграция должна выполняться на семантическом уровне, обеспечивая общий набор понятий или общую онтологию для этих ресурсов. Основными проблемами интеграции информационных ресурсов, таким образом, являются различия в моделях данных хранения ресурсов, в семантической классификации терминов, используемых в данных ресурсах, в форматах хранения, в синтаксисе языков запросов.

Одним из способов подготовки ресурса к семантической интеграции является аннотирование его документов в терминах локальной онтологии домена. Эта онтология отражает систему понятий владельца данного ресурса о предметной области и обычно записывается на одном из формальных языков спецификации знаний [1]. Аннотации используются затем для семантического поиска и классификации документов информационных ресурсов. Очевидно, что локальные онтологии различных ресурс-провайдеров могут иметь отличия даже в рамках одной предметной области. Для решения задач семантической интеграции ресурсов и последующего семантического поиска документов в системе интегрированных ресурсов необходимо обеспечить «выравнивание» понятий этих локальных онтологий на базе общей онтологии медиаторной системы, обеспечивающей интеграцию. Одной из подзадач проблемы «выравнивания» онтологий является задача трансляции онтологий с различных языков описания знаний во внутренний язык медиаторной системы, на котором представлена общая онтология.

В проекте RACING[1] в качестве такого внутреннего языка выбран язык представления древовидных структур L_{tree}, представленный в данной статье.

В статье представлен подход к трансляции онтологий в язык L_{tree}. Подход иллюстрируется демонстрацией механизма трансляции онтологий, записанных на языке классификации публикаций ACM[2], построенном на базе языка DAML+OIL[3].

Статья организована следующим образом. В разделе 2 дается обоснование применимости и описание этапов применения алгебраического подхода к задаче трансляции онтологий, выраженных на разных формализованных языках. В разделах 3 и 4 определяются формальные грамматики входного языка классификации электронных публикаций ACM и целевого языка древовидных структур. В разделе 5 определяется транслирующее отображение.

2 ПРИМЕНЕНИЕ АЛГЕБРАИЧЕСКОГО ПОДХОДА К ТРАНСЛЯЦИИ ОНТОЛОГИЙ

Трансляция текста с одного языка на другой необходимо должна сохранять смысл данного текста. Большие сложности возникают при трансляции естественно-языковых текстов. Однако, в некоторых случае, при работе с текстами на формализованном языке, например, текстами программ на языке программирования, можно добиться семантически эквивалентной трансляции (см.[2], [3]).Для этого используется теория формальных грамматик и общая алгебра.

---

[1] RACING: Rational Agent Coalitions for Intelligent Mediation of Information Retrieval on the Net. Проект финансируется Министерством образования и науки Украины. http://www.zsu.zp.ua/racing/

[2] ACM – Association for Computing Machinery. http://www.acm.org/

[3] DAML+OIL: DAML – DARPA Agent Markup Language, OIL – Ontology Inference layer http://www.w3.org/TR/daml+oil-reference

Наличие формальных порождающих грамматик некоторых языков L_1 и L_2 позволяет применять затем аппарат общей алгебры при поиске отображения перевода (в виде (квази-)гомоморфизма) конструкций языка L_1 в конструкции языка L_2.

А именно, пусть G_1, G_2 - кс-грамматики кс-языков L_1 и L_2 с конечными множествами свободных образующих соответственно, M_1, M_2 - их порождающие алгебры (здесь и далее используется терминология из [2], [3].), M_1^{synt}, M_2^{synt} - их синтаксические алгебры, M_1^{sem}, M_2^{sem} - их семантические алгебры. Пусть также известны гомоморфизмы g_1^{synt}, g_2^{synt}, отображающие синтаксические алгебры языков на соответствующие порождающие алгебры, и гомоморфизмы g_1^{sem}, g_2^{sem}, отображающие синтаксические алгебры обоих языков в их семантические алгебры.

Тогда искомое отношение перевода τ с языка L_1 на язык L_2 существует, если существует отношение η между соответствующими синтаксическими алгебрами обоих языков (см. Рис.1). При этом требуется, чтобы семантические алгебры обоих языков совпадали. Если же последнее условие не выполняется, то отношение η ищут в виде квазигомоморфизма [3].

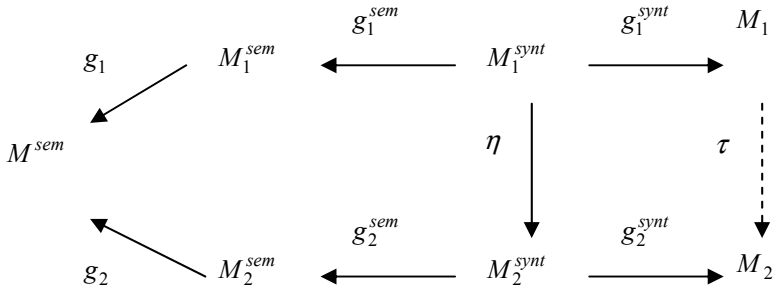

Рисунок 1 – Диаграмма соотношения между порождающими, синтаксическими и семантическими алгебрами формальных языков

В данной статье алгебраический подход, предложенный в [3] для построения перевода между двумя формальными языками, применяется для трансляции текстов *онтологий* с одного языка описания онтологий на другой. Данный подход может быть применен к любым языкам, обладающим контекстно-свободной грамматикой и имеющим конечное множество свободных образующих. В результате применения данного подхода получают *один из* возможных переводов с языка на язык, а при наличии дополнительных требований (существование изоморфизмов между соответствующими алгебрами) – единственный возможный перевод.

В качестве примера, демонстрирующего применение алгебраического подхода, была взята онтология предметной области “Публикации в области Computer Science”[4], используемая в сообществе ACM для классификации электронных публикаций. Данная онтология является «легковесной»[4], т.к. содержит только описания концепций предметной области, упорядоченных отношением иерархии «подкласс-суперкласс». Разработанная в ACM классификация публикаций в области Computer Science является одной из самых подробных контролируемых классификаций в этой области. Дерево каталога ACM классификации на данный момент насчитывает 1286 разделов.

Язык описания данной онтологии является надстройкой над языком XML. Язык XML (eXtensible Markup Language) получил большое распространение как язык разметки текстов, в котором допускается расширение основных разметочных конструкций (тэгов) конструкциями, соответствующие понятиям, присущим только конкретной предметной области. Такие дополнительные тэги определяются с помощью специальной конструкции XML - пространства имен (XML namespace), и используются наравне со стандартными тэгами разметки документов.

Существуют общепринятые расширения языка XML для разметки текстов на основании знаний о предметной области с большей или меньшей степенью детализации. В качестве таких языков используются специализированные языки, такие как RDF[5], RDF-S[6], DAML+OIL[7]. Каждый из перечисленных языков реализуется как надстройка над языком XML, и определяется с помощью соответствующего пространства имен (rdf, rdfs, daml, oil).

[4] Онтология доступна на http://daml.umbc.edu/ontologies/classification.daml

Язык классификации публикаций ACM представлен в виде пространства имен с идентификатором acm и использует конструкции языков RDF, RDFS, DAML+OIL.

Целевым языком представления онтологии был выбран язык древовидных структур, разработанный в рамках проекта RACING (см. раздел 4).

3. ВХОДНОЙ ЯЗЫК КЛАССИФИКАЦИИ ПУБЛИКАЦИЙ ACM

Представим грамматику входного языка классификации публикаций ACM, L_{acm} в нормальной форме Бэкуса (БНФ).

<*acmClassification*>::=<*acmClassificationName*><*acmTopic*>+
<*acmClassificationName*>::=<*acmLiteral*>
<*acmTopic*> ::=”<**acm:Topic rdf:ID=””**<*acmTopicPath*>**””>”**
”**<rdfs:label>”**<*acmTopicName*>**”</rdfs:label>”**
[<*acmComment*>]
<*acmSeeAlso*>+
<*acmSubTopit*>+
[<*acmSuperTopic*>+] **“/acm:Topic”**

<*acmTopicPath*>::=<*acmTopicName*><*acmDashedTopicList*>
<*acmDashedTopicList*> ::=”/”<*acmTopicName*><*acmDashedTopicList* >
<*acmComment*> ::=”**<rdfs:comment>”**<*string*>**”</rdfs:comment>”**
<*acmSeeAlso*> ::=**“<rdfs:seeAlso rdfs:resource= ““#”**<*acmTopicPath*>””**/>”**
<*acmSubTopic*> ::=**“<acm:SubTopic rdfs:resource=””#”** <*acmTopicPath*>””**/>”**+

<*acmSuperTopic*>::=**“<acm:SuperTopic rdfs:resource=””#”**<*acmTopicPath*>**””/>”**
<*acmTopicName*>::=<*acmLiteral*>
<*acmLiteral*> ::=<*string*>

Для облегчения анализа данной грамматики преобразуем нотации БНФ <*nonTerminalList*>+ и {*nonTerminalList*} в отдельные нетерминальные символы типа <*nonTerminalList*>. Получим такие дополнительные нетерминалы:

<*acmTopicList*> ::=<*acmTopic*><*acmTopicList*>
<*acmSeeAlsoList*> ::=<*acmSeeAlso*><*acmSeeAlsoList*>
<*acmSubTopicList*> ::=<*acmSubTopic*><*acmSubTopicList*>
<*acmSuperTopicList*> ::=<*acmSuperTopic*><*acmSuperTopicList*>

Тогда, формально, порождающая грамматика языка L_{acm} имеет вид:

$$G_{acm} = (A_{acm}, V_{n\_acm}, \sigma_{acm}, P_{acm}),$$

где множество терминалов целевого языка

A_{acm} = {<acm:Topic rdf:ID=”, ”>,</acm:Topic>, <rdfs:label>,</rdfs:label>, /, <rdfs:comment>, </rdfs:comment>, <rdfs:SeeAlso rdfs:resource=#”>, “/>,<acm:SubTopic rdfs:resource=#”, <acm:SubTopic rdfs:resource=#”};

множество нетерминалов целевого языка

V_{n_acm} = {<*acmClassification*>, <*acmClassificationName*>, <*acmTopicPath*>, <*acmComment*>, <*acmSeeAlso*>, <*acmSubTopic*>, <*acmSuperTopic*>, <*acmLiteral*>, <*string*>, <*acmTopicList*>, <*acmSeeAlsoList*>, <*acmSubTopicList*>, <*acmSuperTopicList*>, <*acmDashedTopicList*>};

аксиома целевого языка

σ_{acm}=<*classification*>

схема грамматики, определяющая множество правил порождения, представлена выше в нормальной форме Бэкуса.

Порождающая алгебра $MA_{acm} =< M(G_{acm}), \Omega_{acm} >$ будет контекстно-свободной, т.к. в левых частях правил из P_{acm} стоят только нетерминальные символы, а во-вторых многоосновной.

Множество операций Ω_{acm} .

$$\omega_{acmClassification} : A_{acmClassificationName} \times A_{acmTopicList} \rightarrow A_{acmClassification}$$

$\omega_{acmClassificationName} : A_{acmLiteral} \rightarrow A_{acmClassificationName}$

$\omega_{acmTopicList} : A_{acmTopic} \times A_{acmTopicList} \rightarrow A_{acmTopicList}$ $\omega_{acmTopic} :$
$A_{acmTopicName} \times A_{acmTopicPath} \times A_{acmComment} \times A_{acmSeeAlso} \times A_{acmSubTopic} \times A_{acmSuperTopic} \rightarrow A_{acmTopic}$

$\omega_{acmTopicPath} : A_{acmTopicName} \times A_{acmDashedTopicList} \rightarrow A_{acmTopicPath}$

$\omega_{acmDashedTopic} : A_{acmTopicName} \times A_{acmDashedTopicList} \rightarrow A_{acmDashedTopicList}$

$\omega_{acmComment} : A_{acmString} \rightarrow A_{acmComment}$

$\omega_{acmSeeAlsoList} : A_{acmSeeAlso} \times A_{acmSeeAlsoList} \rightarrow A_{acmSeeAlsoList}$

$\omega_{acmSeeAlso} : A_{acmTopicName} \rightarrow A_{acmSeeAlso}$

$\omega_{acmSubTopicList} : A_{acmSubTopic} \times A_{acmSubTopicList} \rightarrow A_{acmSubTopic}$

$\omega_{acmSubTopic} : A_{acmTopicPath} \rightarrow A_{acmSubTopic}$

$\omega_{acmSuperTopicList} : A_{acmSuperTopic} \times A_{acmSuperTopicList} \rightarrow A_{acmSuperTopicList}$

$\omega_{acmSuperTopic} : A_{acmSuperTopicName} \rightarrow A_{acmSuperTopic}$

$\omega_{acmTopicName} : A_{acmString} \rightarrow A_{acmTopicName}$

$\omega_{acmLiteral} : A_{acmString} \rightarrow A_{acmLiteral}$

Носителем алгебры является набор множеств
$M(G_{acm}) = \{A_{acmClassification}, A_{acmClassificationName}, A_{acmTopicList}, A_{acmTopic}, A_{acmTopicPath}, A_{acmDashedTopicList}, A_{acmComment}, A_{acmLabel}, A_{acmSeeAlsoList}, A_{acmSeeAlso}, A_{acmSubTopicList}, A_{acmSubTopic}, A_{acmSuperTopicList}, A_{acmSuperTopic}, A_{acmTopicName}, A_{acmLiteral}, A_{acmString}\}$

4 ЯЗЫК ДРЕВОВИДНЫХ СТРУКТУР L_{tree}

Грамматика целевого языка L_{tree}, предназначенного для представления классификации ACM в виде иерархии рубрик, в нормальной форме Бэкуса имеет вид:

<*treeClassification*>	::=	<*treeTopic*>+
<*treeTopic*>	::=	**<k>**<*treeCode*>**</k>** <*treeTopicName*><*treeTopicPath*><*treeTopicSuper*> <*treeSeeAlso*>+][<*treeTopicComment*>]
<*treeCode*>	::=	<*integer*>
<*treeTopicName*>	::=	**"<n>"**<*string*>**"</n>"**
<*treeTopicPath*>	::=	<*treeTopicName*>"/"+ <*treeTopicName*>
<*treeTopicSuper*>	::=	**"<superclass>"**<*treeCode*>**"</superclass>"**
<*treeTopicComment*>	::=	**"<com>"**<*string*>**"</com>"**
<*treeTopicSeeAlso*>	::=	**"<s_a>"**<*treeCode*>**"</s_a>"**

Аналогично, преобразуем нотации БНФ < *nonTerminalList* >+ и {*nonTerminalList*} в отдельные нетерминальные символы типа <*nonTerminalList*>. Получим такие дополнительные нетерминалы:

<*treeTopicList*> ::=< *treeTopic*><*treeTopicList*>
<*treeTopicSeeAlsoList*> ::=< *treeTopicSeeAlso*><*treeTopicSeeAlsoList*>

Формально, язык L_{tree} является контекстно-свободным, с порождающей грамматикой вида:

$$G_{tree} = (A_{tree}, V_{n\_tree}, \sigma_{tree}, P_{tree}),$$

где множество терминалов целевого языка

A_{tree} = {<k>, </k>, <n>, </n>, <superclass>, </superclass>, <com>, </com>, <s_a>, </s_a>, /};

множество нетерминалов целевого языка

V_{n_tree} = {<*treeClassification*>, <*treetopic*>, <*treetopicList*>, <*treecode*>, <*treetopicName*>, <*treetopicPath*>, <*treetopicSuper*>, <*treetopicSeeAlso*>, <*treetopicComment*>, <*treetopicSeeAlsoList*>};

аксиома целевого языка

σ_{tree} =<classification>;

схема грамматики, определяющая множество правил порождения, представлена выше в нормальной форме Бэкуса.

Порождающая алгебра языка будет такой:

$$MA_{tree} =< M(G_{tree}), \Omega_{tree} >,$$

где $M(G_{tree}) = \{A_{treeClassification}, A_{treeClassificationName}, A_{treeTopicList}, A_{treeTopic}, A_{treeLiteral}, A_{treeTopicName}, A_{treeCode}, A_{treeTopicPath}, A_{treeDashedTopicList}, A_{treeTopicSuper}, A_{treeSeeAlso}, A_{treeComment}, A_{treeInteger}, A_{treeString}\}$

Язык является контекстно-свободным.

Множество операций Ω_{tree}

$v_{treeClassification} : A_{treeClassificationName} \times A_{treeTopicList} \to A_{treeClassification}$

$v_{treeClassificationName} : A_{treeLiteral} \to A_{treeClassificationName}$

$v_{treeTopicList} : A_{treeTopic} \times A_{treeTopicList} \to A_{treeTopicList}$

$v_{treeTopic} : A_{treeTopicName} \times A_{treeCode} \times A_{treeTopicPath} \times A_{treeTopicSuper} \times A_{treeSeeAlso} \to A_{treeTopic}$

$v_{treeTopicPath} : A_{treeTopicName} \times A_{treeDashedTopicList} \to A_{treeTopicPath}$

$v_{treeDashedTopicList} : A_{treeTopicName} \times A_{treeDashedTopicList} \to A_{treeDashedTopicList}$

$v_{treeCode} : A_{treeInteger} \to A_{treeCode}$

$v_{treeTopicName} : A_{treeString} \to A_{treeTopicName}$

$v_{treeTopicSuper} : A_{treeCode} \to A_{treeTopicSuper}$

$v_{treeComment} : A_{treeString} \to A_{treeComment}$

$v_{treeSeeAlso} : A_{treeCode} \to A_{treeSeeAlso}$

$v_{treeLiteral} : A_{treeString} \to A_{treeLiterat}$

5. ТРАНСЛЯЦИЯ С ЯЗЫКА L_{acm} В ЯЗЫК L_{tree}

Поскольку языки L_{acm} и L_{tree} являются контекстно-свободными, с конечными множествами свободных образующих, методы, предложенные в [3] для нахождения отношения перевода, применимы к этим языкам.

Построим перевод языка L_{acm} в L_{tree}.

Синтаксическая алгебра языка L_{acm} имеет вид:

$$MA_{acm}^{synt} =< A_{acmString}, \Omega_{acm} >$$

Синтаксическая алгебра языка L_{tree} имеет вид:

$$MA_{tree}^{synt} =< \{B_{\text{treeInteger}}, B_{treeString}\}, \Omega_{tree} >$$

Сравним синтаксические алгебры для данных языков L_{acm} и L_{tree}. Для языка L_{acm} система свободных образующих состоит из единственного множества $A_{acmString}$, а для языка L_{tree} система свободных образующих состоит из множеств $B_{\text{treeInteger}}, B_{treeString}$.

Таким образом, для данных языков L_{acm} и L_{tree}, ни системы свободных образующих, ни сигнатуры операций соответствующих алгебр не совпадают. Поэтому, согласно [3] отображение между порождающими алгебрами этих двух языков будем искать в виде квазигомоморфизма.

Предварительно потребуется выравнивание систем свободных образующих.

Будем полагать, что в алгебре MA_{acm}^{synt} система свободных образующих состоит из множеств $A_{acmString}, A_{acnInteger}$, причем $A_{acmInteger} \equiv \varnothing$. Везде, где необходимо, элементу $a \in A_{acmString}$ будем сопоставлять пару (a,b), где $b \in B_{acmInteger}$.

Обозначим $A_{string} = A_{acmstring} \cup B_{treeString}$, $A_{\text{integer}} = A_{acmInteger} \cup B_{treeInteger}$.

Будем также предполагать наличие в A_{string} элемента NULL, обозначающего пустую строку. Воздействие любой операции из Ω_{acm} или Ω_{tree}, с сигнатурой из множества A_{string}, на элемент NULL, возвращает элемент NULL соответствующего основного множества. Таким образом, порождаются пустые элементы основных множеств.

Введем в рассмотрение семантическую алгебру $MA^{sem} =< A_{string}, A_{\text{integer}}; \Omega_{acm} \cup \Omega_{tree} >$. Эта алгебра является семантической алгеброй для обоих языков, следовательно, соотношение между порождающими алгебрами обоих языков и их синтаксическими алгебрами будет таким, как показано на рисунке 2.

Искомый перевод τ между двумя порождающими алгебрами A_{acm} и A_{tree} существует, т.к. существует квазигомоморфизм η между синтаксическими алгебрами MA^{synt}_{acm} и MA^{synt}_{tree}

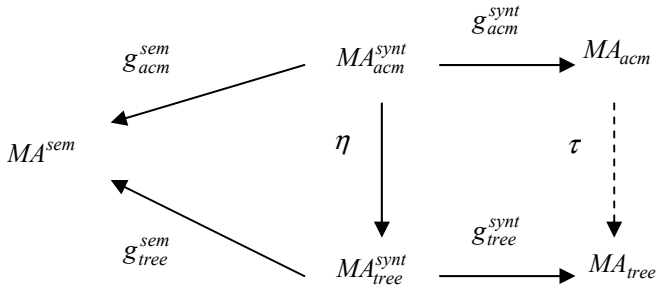

Рисунок 2 – Диаграмма отношений между алгебрами языков L_{acm} и L_{tree}

Построим квазигомоморфизм η.

Квазигомоморфизм η задается системой равенств, сопоставляющих операциям из синтаксической алгебры MA^{synt}_{acm} производные операции из синтаксической алгебры MA^{synt}_{acm}.

η_1: $\omega_{acmLiteral}(s) = v_{treeLiteral}(s),\ s \in A_{string}$

η_2: $\omega_{acmInteger}(i) = v_{treeInteger}(i),\ i \in A_{integer}$

η_3: $\omega_{acmTopicName}(\omega_{acmLiteral}(s)) = v_{treeLiteral}(s),\ s \in A_{string}$

η_4: $\omega_{acmComment}(s) = v_{treeComment}(s),\ s \in A_{string}$

η_5: $\omega_{acmDashedTopicList}(\omega_{acmTopicName}(\omega_{acmLiteral}, (s_1)),\ \omega_{acmDashedTopicList}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_2))),\ \ldots) = v_{treeDashedTopicList}(v_{treeTopicName}(s_1),\ v_{treeDashedTopicList}(v_{treeTopicName}(s_2),\ \ldots)),\ s_1, s_2, \ldots \in A_{string}$

η_6: $\omega_{acmTopicName}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_1)),\ \omega_{acmDashedTopicList}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_2)),\ \ldots) = v_{treeTopicPath}(v_{treeTopicName}(s_1),\ v_{treeDashedTopicList}(v_{treeTopicName}(s_2),\ \ldots)),\ s_1, s_2, \ldots \in A_{string}$

Пополним синтаксическую алгебру S_{tree} операцией соответствия между элементами множества $A_{treeTopicPath}$ и элементами множества $A_{treeCode}$.

V_1: $A_{treeTopicPath} \rightarrow A_{treeCode}$:

$\forall a \in A_{treeTopicPath}\ \exists! c \in A_{treeCode}$

Операции V_1 нет соответствия в A_{tree}

η_7: $\omega_{acmSeeAlso}(\omega_{acmTopicPath}(\omega_{acmTopicName}(\omega_{acmLiteral}(s)), \ldots) = v_{treeSeeAlso}(v_1(v_{treeTopicPath}(s),\ \ldots)))$

η_8: $\omega_{acmSuperTopic}[\omega_{acmTopicPath}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_1)), \ldots NULL] \ldots \omega_{acmDashedTopicList}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_k)), NULL))] = v_{treeTopicSuper}(v_1(v_{treeTopicPath}(v_{treeTopicName}(s_1), \ldots v_{treeDashedTopicList}(v_{treeTopicName}(s_{k-1}),\ \ldots \underbrace{))))))))}_{k-1 скобка}$

$$\eta_9\colon\ \omega_{acmSubTopic}(\omega_{acmTopicPath}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_1)),\ldots\ \omega_{acmDashedTopicList}(\omega_{acmTopicName}(\omega_{acmLiteral}(s_k)),NULL)) = v_{treeSubTopic}(v_{treeTopicPath}(v_{treeTopicName}(s_1),\ldots v_{treeDashedTopicList}(v_{treeTopicName}(s_k))))$$

Выравниваем сигнатуры порожденных и синтаксических алгебр, добавляя к $M(G_{tree})$ множество $A_{treeSubTopic} \equiv \varnothing$ и операцию $v_{treeSubTopic}\colon A_{treeTopicPath} \rightarrow v_{treeSubTopic}$

$$\eta_{10}\colon\ \omega_{acmTopic}(\omega_{acmTopicPath}(s_{1,\ldots,}s_{k1})),\ \omega_{acmTopicName}(\ldots s_{k1+1})),\ \omega_{acmComment}(s_{k1+2}),\ \omega_{acmSeeAlso}(\ldots s_{k1+3},\ldots,s_{k2}),\ \omega_{acmSubTopic}(\ldots,\ s_{k2+1},\ldots,s_{k3}),\ \omega_{acmSuperTopic}(\ldots s_{k3+1},\ \ldots,s_{k4}) =$$

$$v_{treeTopic}(v_{treeTopicName}(s_{k1+1})v_{treeTopicPath}(\ldots s_1\ldots s_{k1}),\ v_1(v_{treeTopicPath}(\ldots s_1\ldots s_{k1})),\ v_{treeComment}(s_{k1+2}),\ v_{treeSeeAlso}(\ldots,\ s_{k1+3},\ \ldots,s_{k2}),\ v_{treeTopicSuper}(\ldots,\ s_{k3+1},\ldots,\ s_{k4}))$$

$$\eta_{11}\colon\ \omega_{acmClassificationName}(\omega_{acmLiteral}(s)) = v_{treeClassificationName}(v_{treeLiteral}(s))$$

$$\eta_{12}\colon\ \omega_{acmClassificationName}(\omega_{acmClassificationName},\ \omega_{acmTopicList}) = v_{treeClassification}(v_{treeClassificationName}, v_{treeTopicList})$$

Полученный квазигомоморфизм индуцирует искомое отношение перевода – квазигомоморфизм τ.

6. ВЫВОДЫ

В статье рассматривается вопрос применения алгебраического подхода к построению перевода текстов *онтологий* с одного формализованного языка на другой, с сохранением смысла концепций онтологии. На примере языка L_{acm} - языка описания «легковесной» онтологии предметной области «Публикации в области Computer Science» и целевого языка древовидных структур L_{tree}, используемого в проекте RACING, демонстрируется применимость указанного алгебраического подхода. В статье также даны формальные грамматики языков L_{acm} и L_{tree}.

ЛИТЕРАТУРА

1. Ontology Language Standardisation Efforts. Sean Bechhofer (Ed.) OntoWeb Deliverable 4.0, Ver. 01 29.01.2002, http://www.ontoweb.org/download/deliverables/d4.0.pdf – доступ проверялся 19.06.2002г.
2. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. Киев, Наукова думка, 1989.-376 с.
3. Летичевский А.А. Синтаксис и семантика формальных языков.//Кибернетика.-1968. №4, с.1-10.
4. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб:ПИТЕР, 2000.
5. RDF. http://www.w3.org/RDF – доступ проверялся 19.06.2002г.
6. RDF-S. http://www.w3.org/TR/rdf-schema – доступ проверялся 19.06.2002г.
7. DAML+OIL. http://www.w3.org/TR/daml+oil-reference – доступ проверялся 19.06.2002г.