



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

МЕТОД ПРЕОБРАЗОВАНИЯ КЛАССОВ СЕМАНТИЧЕСКИХ СЕТЕЙ

Тимченко В.А.

*Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук,
г. Владивосток, Россия*

rakot2k@mail.ru

В работе представлен метод преобразования классов семантических сетей, определяющий, как по заданной спецификации преобразования на основе исходного текстового или структурного представления семантической информации получить целевое текстовое или структурное представление этой информации, удовлетворяющее этой спецификации. Данный метод можно представить как совокупность трех базовых методов преобразования. Спецификации преобразований описываются в терминах моделей классов семантических сетей и их преобразований, являющихся инвариантными по отношению к технологическим пространствам.

Ключевые слова: методы преобразования семантических сетей, представление информации семантическими сетями, семантические сети, структурные проекции.

ВВЕДЕНИЕ

В настоящее время семантические сети успешно используются в качестве наглядной и универсальной структуры для представления информации (преимущественно декларативной) общего вида в разных предметных областях. Разные виды информации, независимо от конкретного синтаксиса языка их представления, в абстрактном синтаксисе представляют собой в общем случае (мульти-) графы, возможно, типизированные [Agrawal A. et al., 2005] [Ehrig K. et al., 2005]. Поэтому многие задачи преобразования семантической информации, возникающие в различных областях профессиональной деятельности, технологических пространствах [Kurtev, 2002] [Bezivin, 2005], а также на стыке разных дисциплин и технологических пространств, можно сформулировать в терминах преобразования семантических сетей. К таким задачам среди прочих можно отнести: задачу отображения онтологий, преобразования баз знаний, преобразование программ, преобразование моделей в моделью ориентированном подходе к инженерии информационных систем. Задача преобразования семантических сетей в общем случае может быть сформулирована следующим образом: по исходной семантической сети и спецификации преобразования сформировать новую семантическую сеть, удовлетворяющую этой спецификации.

Вместе с тем, в разных технологических пространствах исторически независимо сформировались свои эффективные модели и основанные на них методы преобразования информации; разработаны программные системы их поддержки. Однако, как показывает практика и опыт использования, методы и формализмы, применяемые в одном технологическом пространстве, зачастую по многим причинам напрямую не могут быть “спроецированы” на другое [Czarnecki, 2006].

Основным недостатком существующих методов преобразования разных видов семантической информации, таким образом, является их ориентированность на некоторое технологическое пространство или один вид преобразований – эндогенных или экзогенных, либо вовсе на конкретную задачу или класс задач; а также то, что они рассчитаны на круг специалистов, обладающих необходимыми знаниями в конкретной области.

В данной работе представлен метод преобразования классов семантических сетей, основанный на модели классов семантических сетей и их преобразований, которые являются инвариантными по отношению к технологическим пространствам [Тимченко, 2012]. Данный метод можно представить как совокупность трех базовых методов преобразования: “текст – семантическая сеть”, “семантическая сеть – текст” и “семантическая сеть – семантическая сеть”.

Причина, по которой должна обеспечиваться

возможность выполнять преобразования вида “текст – семантическая сеть” и “семантическая сеть – текст”, состоит в том, что во многих предметных областях текстовое представление информации (представление в конкретном синтаксисе) зачастую более привычно и понятно человеку, нежели структурное, в то время как последнее (представление в абстрактном синтаксисе), как правило, существенно удобнее для обработки программными системами. Таким образом, исходная информация может иметь как текстовое, так и структурное представление (в виде семантической сети). Целевую информацию также может потребоваться представить как в текстовом, так и в структурном виде.

Еще один аргумент состоит в том, что часто бывает быстрее и проще сформировать представление информации в виде сети понятий автоматически по текстовому представлению, чем вручную, с помощью средства редактирования порождать такую сеть под управлением описания класса сетей понятий для этой информации. Примером является формирование программы на языке программирования в виде дерева абстрактного синтаксиса по описанию грамматики этого языка.

Спецификацией для метода преобразования “семантическая сеть – семантическая сеть” является описание структурной проекции на языке описания структурных проекций. Спецификацией для методов преобразования “текст – семантическая сеть” и “семантическая сеть – текст” является описание синтаксических ограничений (в терминах модели синтаксических ограничений) [Тимченко, 2012].

1. Метод преобразования “текст – семантическая сеть”

Существует множество разнообразных алгоритмов для анализа текста КС-языков разных классов. Наибольший интерес с точки зрения универсальности представляют общие методы анализа языков, не ограниченных требованиями к наличию тех или иных свойств у грамматики, их описывающей ($LL(k)$, $LR(k)$, наличие определенного порядка (грамматики предшествования), линейность и т.д.), то есть языков, являющихся контекстно-свободными в самом широком смысле [Хомский, 1966]. Такие языки описываются КС-грамматиками, правила которых имеют вид $A \rightarrow \alpha$, где A – понятие языка, или нетерминал, а α – строка символов из объединенного словаря терминалов и нетерминалов, либо пустая строка. В рамках данной работы решается задача анализа текста, который может быть описан на произвольном КС-языке.

Для решения задачи анализа выбран алгоритм Эрли, который подходит для осуществления разбора текста на КС-языке, задаваемом КС-грамматикой без ограничений. Алгоритм Эрли считается одним

из самых эффективных универсальным методом синтаксического анализа текстов, записанных на КС-языках. И хотя известен алгоритм Томиты и его новейшие модификации, которые претендуют на еще лучшие показатели эффективности КС анализа, отмечается, что реализация этих алгоритмов в десять, а то и более, раз сложнее, чем реализация алгоритма Эрли. Описание алгоритма Эрли можно найти, например, в [Ахо, 1978].

Вход:

- текстовое представление информации (*Text*);
- описание класса семантических сетей информации (*S*).

Выход:

- представление информации в виде семантической сети (S_{inst}).

Описание:

Этап 1. Формирование КС-грамматики по описанию синтаксических ограничений и описанию класса сетей понятий (S_{sr} и S_{nc} из S). Данный этап включает в себя:

– преобразование в соответствии с семантикой метасимволов синтаксических ограничений, содержащих факультативные элементы, а также ограничений, представляющих альтернативы и перечисления, в множество правил грамматики вида $A_i \rightarrow \alpha_i$, где A_i – нетерминальное понятие из описания класса сетей понятий, а α_i – последовательность, состоящая, в общем случае, из нетерминальных и терминальных понятий класса сетей понятий, а также элементов конкретного синтаксиса;

– формирование таблицы элементов конкретного синтаксиса, а также таблицы, содержащей ограничения на вид лексем.

Этап 2. Анализ текстового представления (*Text*) по сформированной КС-грамматике и построение последовательности списков ситуаций Эрли. На данном этапе используется алгоритм Эрли [Ахо, 1978].

Этап 3. Формирование семантической сети (S_{inst}), соответствующей описанию класса сетей понятий (S_{nc}), на основе полученной последовательности списков ситуаций и КС-грамматики.

Основная идея этапа состоит в том, чтобы, двигаясь от вершины сети (начального понятия) рекурсивно достраивать сеть понятий в соответствии с правилами грамматики и построенной последовательностью списков ситуаций Эрли.

Данный этап представлен рекурсивной функцией (*BuildNetwork*), входными параметрами которой являются: указатель на ситуацию Эрли, обрабатываемую на очередном шаге рекурсии (*state*); номер списка ситуации (*number*); указатель на вершину в формируемой S_{inst} , в которой находится вычислительный процесс при

выполнении очередного шага рекурсии (*concept*). Результат выполнения функции (*result*) содержит указатель на ситуацию Эрли, которую нужно обработать для корректной обработки перечислимых правил, а также – номер списка ситуаций, в котором эта ситуация находится, либо *пусто* – если правило, соответствующее текущей обрабатываемой ситуации, не является перечислимым (т.е. вида $\{;\dots\}$). При первом вызове аргументами функции *BuildNetwork* являются:

- указатель на ситуацию, являющуюся допускающей для входной строки символов. Допускающей является ситуация, присутствующая в списке ситуаций (из последовательности списков ситуаций) с наибольшим номером и содержащая правило грамматики, в котором нетерминал в левой части есть начальный нетерминал грамматики, точка в правой части – крайняя справа, и номер ситуации есть 0;

- длина входной строки КС-текста (*Text*) в лексемах (в качестве номера списка ситуаций);

- начальной вершиной формируемого S_{inst} является левая часть правила, содержащегося в допускающей ситуации.

Шаг рекурсии состоит в следующем:

1. Из текущей ситуации Эрли (переданной в параметре вызова *BuildNetwork*) *state* берется правило грамматики *rule*.

2. В цикле, пока не достигнут первый его элемент из правой части ($k \geq 1$), выполняются следующие действия:

- 2.1. Берется k -ый элемент из правой части правила *rule* – *grammarElem* (перед циклом k равен номеру позиции последнего элемента правой части правила *rule*);

- 2.2. если *grammarElem* является терминалом, то:

- 2.2.1. создать терминальное понятие в S_{inst} и сделать его потомком *concept* – понятия S_{inst} , переданного в параметре вызова *BuildNetwork*: взять из таблицы терминалов тот, который был считан при формировании данного списка ситуаций, и добавить его в качестве понятия-потомка. При этом если для данного терминала уже было создано понятие в S_{inst} , то к нему создается только отношение от *concept*, иначе – создается новое понятие в S_{inst} и отношение к нему от *concept*;

- 2.2.2. перейти к предыдущему элементу из правой части правила ($k = k - 1$), а также к предыдущему списку ситуаций Эрли;

- 2.3. если *grammarElem* является нетерминалом, то:

- 2.3.1. создать нетерминальное понятие в S_{inst} (*childConcept*), присвоить ему в качестве значения значение нетерминала грамматики *grammarElem* и сделать его потомком понятия *concept*;

- 2.3.2. получить подходящую ситуацию Эрли для следующего шага рекурсии (*nextState*) по текущему номеру списка ситуаций Эрли и рассматриваемому элементу правила грамматики *grammarElem*;

- 2.3.3. если текущее правило является перечислимым, то необходимо сформировать *result* по текущему номеру списка ситуаций Эрли и текущей ситуации Эрли (*state*), чтобы следующий вызов *BuildNetwork* относился к этому же нетерминалу *grammarElem*, но соответствовал предыдущей итерации для перечислимого правила;

- 2.3.4. рекурсивно вызывается функция *BuildNetwork* с аргументами *nextState*, текущий номер списка ситуаций Эрли и *childConcept*. Далее для обработки перечислимого правила в цикле, пока результат вызова *BuildNetwork* (обозначим его *res*) не *пусто*, данная функция вызывается рекурсивно с аргументами: значение из *res*, содержащее указатель на ситуацию Эрли, значение из *res*, содержащее номер списка ситуаций, в котором эта ситуация находится, и *childConcept*;

- 2.3.5. перейти к предыдущему элементу из правой части правила ($k = k - 1$), а текущим номером списка ситуаций Эрли становится номер списка, в котором находится родительская ситуация ситуации *nextState*.

3. Выполняется выход из *BuildNetwork* с возвратом результата *result*.

2. Метод преобразования “семантическая сеть – текст”

Задача отображения некоторой информации, представленной в структурном виде, в текстовое представление является задачей синтеза текста. В рамках данной работы решается задача синтеза текста, который также может быть описан на произвольном КС-языке. Вариантом решения задачи синтеза является также недоопределенное текстовое представление, в котором не все слова (лексемы) являются элементами конкретного синтаксиса или значениями терминальных понятий. Некоторые понятия из семантической сети, представляющей описание класса сетей понятий, могут быть недоопределены в сети понятий, являющейся ее экземпляром. В этом случае в результирующем тексте должны быть синтезированы имена этих недоопределенных конструкций.

Вход:

- представление информации в виде семантической сети (S_{inst});
- описание класса семантических сетей информации (S).

Выход:

- текстовое представление информации (*Text*).

Описание:

Этап 1. Формирование КС-грамматики по описанию синтаксических ограничений и описанию класса сетей понятий (S_{sr} и S_{nc} из S). Данный этап

полностью совпадает с аналогичным этапом метода преобразования “текст – семантическая сеть”, который описан в предыдущем разделе.

Этап 2. Рекурсивный обход S_{inst} в глубину с параллельным просмотром правил сформированной КС-грамматики и синтезом фрагментов результирующего текста. Данный этап представлен рекурсивной функцией (*SynthesiseText*), входными параметрами которой являются: указатель на элемент (нетерминал) сформированной КС-грамматики (*grammarElem*); указатель на вершину в S_{inst} (*concept*). При первом вызове аргументами функции *SynthesiseText* являются аксиома грамматики и корневая вершина S_{inst} . Шаг рекурсии состоит в следующем:

1. Если *concept* является терминальным понятием, то к *Text* добавляется строковое представление значения понятия (при этом, если тип значения понятия *concept* есть “строковое”, то перед и после строкового представления значения ставится символ-ограничитель строковых констант, заданный в описании синтаксических ограничений).

2. Если *concept* является нетерминальным понятием, то:

2.1. получить правило грамматики (*rule*), у которого в левой части стоит *grammarElem* (если таких правил нет, то синтез текста не может быть выполнен);

2.2. если таких правил больше одного, то это одно из альтернативных правил. В этом случае выбирается нужная альтернатива (по имени метапонятия для понятия-потомка *concept*) и рекурсивно вызывается функция *SynthesiseText* с аргументами: правая часть альтернативного правила, понятие-потомок понятия *concept*;

2.3. если это не альтернативное правило, то в цикле по элементам правой части правила *rule* (одновременно выполняется проход по потомкам понятия *concept*):

2.3.1. элемент конкретного синтаксиса добавляется к *Text*;

2.3.2. если в S_{inst} нет понятий, соответствующих текущему элементу грамматики, то синтез текста не может быть выполнен;

2.3.3. если текущий потомок понятия *concept* (*child*) соответствует текущему элементу грамматики (*grElem*) (т.е. их типы совпадают, и имя метапонятия *child* совпадает с *grElem*), то рекурсивно вызывается функция *SynthesiseText* с аргументами: *grElem*, *child*; в противном случае синтез текста не может быть выполнен;

2.3.4. если *rule* является перечислимым правилом, то в цикле перебираются только потомки понятия *concept*, а текущий элемент грамматики *grElem* остается неизменным; к *Text* при этом добавляется символ-разделитель (заданный в описании синтаксических ограничений), которым в

тексте разделены повторяющиеся элементы; выполняется п. 2.3.3.

3. Метод преобразования “семантическая сеть – семантическая сеть”

Данный метод основан на операционной семантике языка описания структурных проекций, разработанного для формирования пользователем множества правил в терминах модели описания структурных проекций [Тимченко, 2012].

Вход:

- представление исходной информации в виде семантической сети (SS_{inst});
- описание класса сетей понятий целевой информации (TS_{nc});
- описание структурной проекции (μ).

Выход:

- представление целевой информации в виде семантической сети (TS_{inst}).

Описание:

Этап 1. Кэширование описания структурной проекции (μ) в хэш-структуре (*Mapping*) с целью ускорения доступа к правилам структурной проекции. При этом левые части правил проекции становятся ключами в хэш-структуре. Правая часть каждого правила проекции сохраняется в специальном абстрактном типе данных (АТД), после чего для доступа к нужным значениям правой части правила используются операции этого АТД. Экземпляры АТД сопоставляются ключам в хэш-структуре.

Этап 2. Выполняется рекурсивный обход исходной семантической сети (SS_{inst}) в глубину, на каждом шаге которого:

для вершины-прототипа текущей вершины из описания проекции (*Mapping*) выбирается применимое правило (если такое есть). В соответствии с описанием правой части выбранного правила в целевой семантической сети (TS_{inst}) достраивается подсеть.

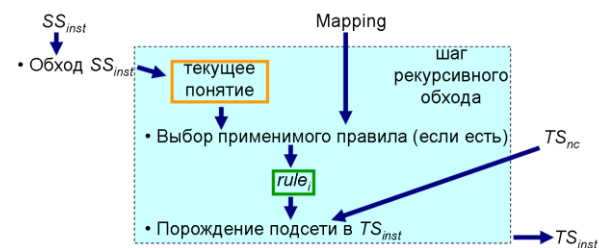


Рисунок 1 – Общая схема метода преобразования “семантическая сеть – семантическая сеть”

При этом создается новая подсеть, являющаяся экземпляром некоторой подсети из описания класса сетей понятий целевой информации (TS_{nc}), и связывается с уже сформированной на данном этапе (если такая есть) семантической сетью. После применения очередного правила формируемая целевая сеть всегда соответствует своей

метаинформации (рис.1). Порождение подсети TS_{inst} определяется через порождение множеств понятий и отношений, в нее входящих, по их прототипам из TS_{nc} .

Понятие в TS_{inst} порождается следующим образом: создается нетерминальное или терминальное понятие соответствующего типа (эта информация берется на основе информации о понятии-прототипе), а в качестве значения понятию может назначаться (в соответствии с правой частью применяемого правила) (ниже приведены альтернативные варианты):

- имя/значение понятия-прототипа;
 - имя/значение понятия из SS_{inst} или его терминального понятия-потомка;
 - значение из хеш-таблицы по заданному псевдониму (которое было туда предварительно помещено);
 - явно заданное в правой части правила значение, либо это значение вычисляется по заданному в правиле выражению (в которое может входить произвольное количество переменных, инициализируемых значениями из SS_{inst} , либо из хеш-таблицы).
- Направленное отношение между двумя понятиями в TS_{inst} порождается следующим образом:
- начальное и конечное понятия берутся из хеш-таблицы по заданному псевдониму, либо ищутся в уже сформированной на данном этапе TS_{inst} (в этом случае в правиле явно указано имя/значение понятия (или его понятия-прототипа, по которому оно порождено));
 - отношение добавляется к множеству отношений, выходящих из понятия-начала, либо в соответствие с заданным порядковым номером, либо последним, если порядковый номер в правиле не задан.

ЗАКЛЮЧЕНИЕ

Разработаны методы преобразований “текст – семантическая сеть”, “семантическая сеть – текст” и “семантическая сеть – семантическая сеть”, определяющие, как по соответствующей спецификации преобразования на основе входной информации получить выходную, удовлетворяющую этой спецификации. Метод преобразования “текст – семантическая сеть” позволяет анализировать текст на языке, задаваемом КС-грамматикой без ограничений, в то время как большинство широко распространенных средств реализуют метод разбора, требующий, чтобы входная грамматика удовлетворяла определенным свойствам.

Следует отметить, что в зависимости от решаемой задачи преобразования, эти методы можно комбинировать между собой в разных

сочетаниях. Всего допустимо три комбинации, в зависимости от представления входной информации (текстовое или в виде семантической сети) и требуемого представления выходной информации (текстовое или в виде семантической сети). Возможность варьировать порядок применения методов позволяет существенно расширить спектр прикладного применения результатов работы.

Работа выполнена при финансовой поддержке РФФИ, проект 12-07-00179-а и ДВО РАН по Программе ОНИТ, проект 12-И-ОНИТ-04.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Agrawal et al., 2005] Agrawal, A. Reusable Idioms and Patterns in Graph Transformation Languages / A. Agrawal, A. Vizhanyo, Z. Kalmar, F. Shi, A. Narayanan, G. Karsai // Journal Electronic Notes in Theoretical Computer Science (ENTCS). – Vol. 127. – Issue 1. – 2005. – Pp. 181 – 192.
- [Ehrig et al., 2005] Ehrig, K. Model Transformation by Graph Transformation: A Comparative Study / K. Ehrig, E. Guerra, J. de Lara, L. Lengyel, T. Levendovszky, U. Prange, G. Taentzer, D. Varro, S. Varro-Gyapay // Proceedings of Model Transformations in Practice Workshop, MoDELS Conference. – Jamaica. – 2005. [Electronic resource]. URL: <http://www.inf.mit.bme.hu/FTSRG/Publications/varro/2005/mtip05.pdf> (дата обращения: 26.11.2012).
- [Kurtev, 2002] Kurtev, I. Technological Spaces: An Initial Appraisal / I. Kurtev, J. Bezivin, M. Aksit // Int. Federated Conf. (DOA, ODBASE, CoopIS), Industrial track, Irvine. – 2002.
- [Bezivin, 2005] Bezivin, J. Model-based Technology Integration with the Technical Space Concept / J. Bezivin, I. Kurtev // In Proceedings of the Metainformatics Symposium, Springer-Verlag. – 2005.
- [Czarnecki, 2006] Czarnecki, K. Feature-based survey of model transformation approaches / K. Czarnecki, S. Helsen // IBM Systems Journal, special issue on Model-Driven Software Development. 45(3). – 2006. – Pp. 621 – 645.
- [Тимченко, 2012] Тимченко, В. А. Модель классов семантических сетей и их преобразований / В. А. Тимченко // Материалы международной научно-технической конференции OSTIS-2012. – Минск БГУИР. – 2012. – С. 63 – 70.
- [Хомский, 1966] Хомский, Н. Формальные свойства грамматик / Н. Хомский // Кибернетический сборник, новая серия, вып. 2, изд-во “Мир”, Москва. – 1966. – С. 121 – 230.
- [Ахо, 1978] Ахо, А. Теория синтаксического анализа, перевода и компиляции / А. Ахо, Дж. Ульман // Москва: Мир. 1978. В двух томах.

THE METHOD OF SEMANTIC NETWORK CLASSES TRANSFORMATION

Timchenko V.A.

*Federal State Budget Institution of Science
Institute of Automation and Control Processes Far
Eastern Branch of the Russian Academy of
Sciences, Vladivostok, Russia*

rakot2k@mail.ru

The paper presents a method for semantic network classes transformation. This method defines how to generate the target textual or structural representation of semantic information from source textual or structural representation of semantic information in concordance with transformation specification. This method can be represented as a combination of three basic methods of transformation. Specifications of transformations are described in terms of the models of semantic network

classes and their transformations. Both models are invariant to the different technological spaces.

INTRODUCTION

Currently, semantic networks have been successfully used as a visual and universal structure for representation of common information (mostly declarative) in different domains. Different types of information, regardless of the concrete syntax of their representation, in the abstract syntax are, in general, the (multi-) typed graphs [Agrawal A. et al., 2005] [Ehrig K. et al., 2005]. Therefore many problems of information transformation appearing in different areas of professional activity, technological spaces [Kurtev, 2002] [Bezivin, 2005], as well as at their joint, can be formulated in terms of the semantic networks transformation. These problems include: the problem of ontology mapping, knowledge bases transformation, programs transformation, models transformations in model-driven engineering (MDE). The problem of semantic networks transformation generally can be formulated as follows: on basis of initial semantic network and the transformation specification to generate the new semantic network that meets this specification.

At the same time, in different technological spaces the efficient models and methods as well as software systems for information transformation were developed independently. However, as practice and experience find out, the methods and formalisms applied in a technological space often cannot be directly “mapped” to another one [Czarnecki, 2006].

The main drawback of the existing methods for different types of semantic information transformation is their orientation on a special technological space or one type of transformation – endogenous or exogenous, or even on a specific class of problems. The other lack of the existing methods is that they are designed for group of professionals who have special knowledge and skills in a particular domain or technological space.

This paper presents a method for semantic network classes transformation, based on the models of semantic network classes and their transformations which are invariant to the different technological spaces [Timchenko, 2012]. This method can be represented as a set of three basic methods of transformation: “text-to-semantic network”, “semantic network-to-text” and “semantic network-to-semantic network”.

The reason that it should be possible to perform the “text-to-semantic network” and “semantic network-to-text” transformations is that in many domains the textual representation of the information (representation in a concrete syntax) are often more familiar to people than structural representation, while the latter (the representation in the abstract syntax) is usually much more convenient for software systems processing. Thus, the source information can have both textual and structural representation (in the form of a semantic network). Target information can also be required to present both in textual and in the structural form.

Another argument is that it is often faster and easier to create a representation of information in a network of concepts automatically based on textual representation than manually. The latter means using some editing tools to generate such a network of concepts using some semantic network class as meta information. An example is the creation of the program in a programming language in the form of an abstract syntax tree in concordance with the grammar of this language.

The specification for the “semantic network-to-semantic network” transformation method is a description of the structure mapping in the terms of language for structure mappings description. The specification for the “text-to-semantic network” and “semantic network-to-text” transformation methods is the description of syntax restrictions (in terms of the syntax restrictions model) [Timchenko, 2012].

MAIN PART

The first section presents the description of the main ideas and the key steps of the “text-to-semantic network” transformation method.

The second section presents the description of the main ideas and the key steps of the “semantic network-to-text” transformation method.

The third section presents the description of the main ideas and the key steps of the “semantic network - to-semantic network” transformation method.

CONCLUSION

There have been developed “text-to-semantic network”, “semantic network-to-text” and “semantic network-to-semantic network” methods of transformations defining how on the basis of the appropriate transformation specification and the input information to generate the output one, that meets this specification. “Text-to-semantic network” transformation method allows you to analyze the text in the language specified by a CF-grammar without restrictions, while most common implemented methods of analysis require that the input grammar satisfies certain properties.

It should be noted that, depending on the problem of transformation, these methods can be combined together in different ways. In total three combinations are allowed, depending on the representation of input information (text or in the form of a semantic network) and the required representation of output information (text or in the form of a semantic network). The possibility to vary the order of the methods application can significantly expand the range of applications of the results of the work.

The paper was written with financial support from RFBR, project 12-07-00179-a and from FEB RAS under Program of the RAS branch “Branch of nanotechnologies and information technologies”, project 12-I-OHIT-04.