Редактор расширенных сетей переходов с графическим интерфейсом пользователя

The editor of the augmented transition networks with graphical user interface

Лебедев А. С. (andremoniy@gmail.com)

Московский государственный институт электроники и математики

В докладе описывается разработанный визуальный редактор расширенных сетей переходов, позволяющий облегчить работу эксперта-программиста лингвистического процессора, основанного на использовании таких сетей. Возможности редактора иллюстрируются на простых примерах.

1. Введение

Центральной задачей при создании лингвистического процессора является выбор модели и разработка семантического анализатора. В данной работе рассматривается модель на основе расширенных сетей переходов.

Расширенная сеть переходов, также известная под названием «ATN-сеть (augmented transition network)» — давно известный инструмент семантического анализа [1]. Технологии, основанные на использовании расширенных сетей переходов, используются, например, таким крупным и известным разработчиком программного обеспечения, как компания «ПРОМТ»[2]. Расширенная сеть переходов в описываемой реализации представляет собой однонаправленный граф без циклов, где дугам графа поставлены в соответствие наборы морфологических свойств лексем, а узлам — наборы семантических ролей, которые можно выделить на данном этапе прохождения по графу. Алгоритм работы предлагаемого ATN-анализатора будет рассмотрен ниже. Расширенные сети представляют собой, таким образом, один из механизмов, с помощью которых эксперты предметной области и эксперты-лингвисты могут принимать непосредственное участие в программировании лингвистического процессора.

Такая модель использует парадигму, когда знания конструируются человеком, т. е. человек-эксперт постоянно сопровождает систему, внося в нее по мере необходимости новые знания. Для этой цели важно иметь хороший инструментарий для разработчика. [1]

В данной работе ставится задача создания приложения с удобным пользовательским интерфейсом и набором необходимого инструментария, посредством работы с которым можно было бы облегчить процесс программирования лингвистического процессора за счет привлечения экспертов-лингвистов предметной области. Данная задача была решена путем создания **визуального редактора расширенных сетей переходов**, работа с которым не требует знаний в области программирования.

2. Модель предлагаемого ATN-анализатора

Для более глубокого понимания поставленной задачи и способа ее решения опишем модель разработанного анализатора на основе сети переходов.

База знаний в разработанном ATN-анализаторе представлена в виде набора расширенных сетей переходов. Каждая сеть в данной модели соответствует одной части речи, с которой начинается разбор предложения. Дуги в таких сетях помечены частью речи и набором морфологических признаков. Все пути в такой сети, ведущие от начального состояния к конечному, соответствуют некоторому правилу для разбора предложения. В узлах сети могут находиться некоторые правила и команды, управляющие работой ATN-анализатора. Правила представляют собой набор семантических связей, выделенных на данном этапе разбора. Команды позволяют изменять состояние внутренних переменных анализатора или использовать в качестве семантической

связи неопределенные формы глаголов (в случае, если требуется выражение семантики с помощью конкретной глагольной формы).

Разбор предложения происходит следующим образом. Морфологический анализатор[1] получает на входе словоформу, а на выходе выдает данную словоформу и набор ее морфологических признаков, ключевым из которых является часть речи. В случае если рассматриваемому слову соответствует несколько разных лексем (с разными частями речи), то на выходе морфологический анализатор выдает массив соответствующих наборов морфологических признаков для каждой возможной лексемы. В соответствии с частью речи первого слова предложения ATN-анализатор открывает соответствующую расширенную сеть переходов. Если найдено несколько частей речи, то для каждой создается копия семантического представления и открывается соответствующая расширенная сеть переходов.

Для иллюстрации рассмотрим простейший случай, когда каждой словоформе соответствует одна часть речи (т. е. отсутствуют морфологические многозначности по признаку части речи). Пример предложения:

(1) Старушка вынула из рабочего ящика нательный золотой крестик Наташи (Достоевский).

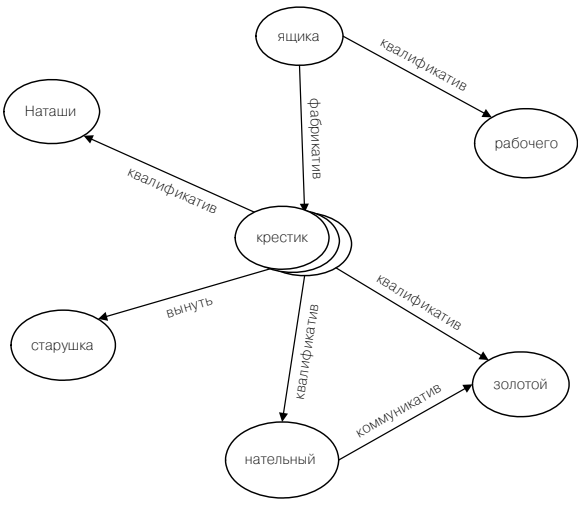

Рис. 1. Семантическое представление примера (1)

Семантическое представление (СП) предложения (1), полученное с помощью разработанного лингвистического процессора представлено на рис. 1. Для удобства разработчиков и экспертов система способна отображать схемы полученных семантических представлений в виде звезды. Ближе к центру рисунка располагаются лексемы, имеющие наибольшее число семантических и синтаксических связей.

При построении данного семантического представления ATN-анализатором использовался следующий фрагмент расширенной сети переходов — сеть1 (рис. 2). Числа в вершинах графа — это уникальные идентификаторы вершин.

Как видно на рис. 2. сеть перегружена неопределенными формами слов, и представляет собой синтаксически-семантический разбор конкретного примера (1) и поэтому не применима для других предложений. В качестве универсализации можно ослабить морфологические свойства ребер и исключить из них неопределенные словоформы. Тогда данная сеть (рис. 3) приобретет более облегченный вид, что позволит использовать ее для анализа целого класса предложений, имеющих такую же структуру.

Неопределенная форма у глагола оставлена, так как глагол определяет в данном контексте лингвистическое отношение между подлежащим *старушка* и дополнением *крестик*. На рис. 3 также представлен список морфологических признаков, назначенных ребру *180-18001*, которое описывает переход в случае, если следующее слово из входного потока является именем собственным, находящимся в родительном или именительном падеже, женского рода. Данная расширенная сеть уже более универсальна, и с помощью нее можно разобрать такие предложения, как:

(2) Девочка вынула из потертого рюкзака большой тяжелый учебник Маши.

(3) Гувернантка вынула из старого сундука белое свадебное платье Анны.

Следующим шагом универсализации сети является удаление из фреймов таких морфологических свойств, как «род», «число» и т.п. Таким образом, мы получим сеть3, которая позволяет разбирать уже более широкий класс предложений, в том числе и таких как:

(4) Мальчик вынул из синего моря ветхую рыбацкую сеть Петра.

Итак, для программирования ATN-анализатора требуется редактор, в возможности которого входят следующие основные функции:

- внесение в базу знаний новых расширенных сетей переходов;
- редактирование имеющихся сетей: добавление, удаление и изменение узлов сети, редактирование фреймов, назначаемых ребрам и узлам сети;
- поиск по имеющимся сетям.

---

[1] В работе применялись для исследований два морфологических анализатора:

а) ABBYY Retrieval & Morphology 4.0 Engine — инструментарий разработчика программного обеспечения;

б) Парсер mystem (http://company.yandex.ru/technology/mystem/) [3].
Наиболее положительный результат морфологического анализа показал парсер mystem.

Рис. 2. Фрагмент варианта расширенной сети переходов для примера (1) (Сеть1)

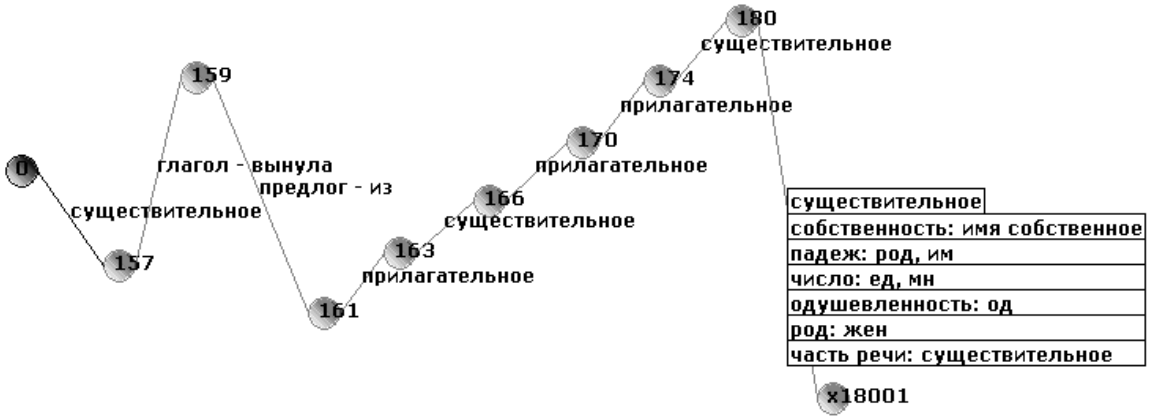

Рис. 3. Фрагмент расширенной сети переходов без словоформ для примера (1) (Сеть2).

Рис. 4. Сеть3

3. Визуальный редактор расширенных сетей переходов

Расширенные сети переходов представляют собой *семантическую память* (СП) [4]. В приведенном выше примере разработки расширенной сети переходов рассматривался простой случай, где не учитывалась многозначность слов и словоформ[2]. Кроме того, в узких предметных областях только эксперт предметной области может грамотно выделить те ключевые термины, которые особенно важно обработать исключительным образом при разборе предложения, т.е. термины, исключительным образом влияющие на семантику текста (и предложения в частности). В данном докладе описывается именно такой инструментарий, представленный в виде визуальной среды для редактирования расширенных сетей переходов (в модели, описанной в п. 2).

Редактор представляет собой оконное приложение, написанное на языке Java 1.5. В качестве морфологического анализатора теоретически может использоваться любой, практически же один из рассмотренных выше (программа mystem или морфологический анализатор Abbyy). Практическая заменимость морфологического анализатора достигается за счет его реализации в отдельном пакете.

Интерфейс редактора представляет собой окно, позволяющие отобразить в отдельных вкладках несколько расширенных сетей в виде графов (см. рис. 5).

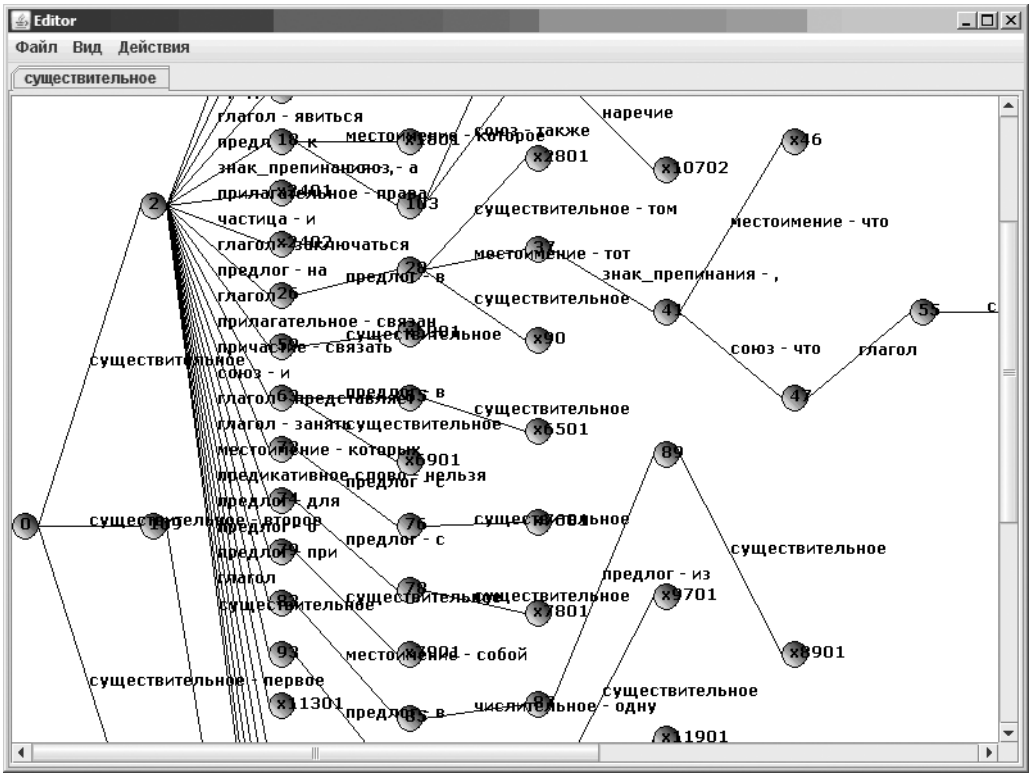

Рис. 5. Экран визуального редактора расширенных сетей переходов

Конструирование сети производится с помощью мыши. Правым щелчком мыши у выбранного узла сети создаются новые ребра. Нумерация узлов производится автоматически. Признаком конца сети является латинская буква «x» в начале имени узла. Морфологические свойства, которыми обладают ребра, могут быть заданы путем выбора нужного ребра, двойным щелчком мыши на выбранном ребре и последующим редактированием окна ввода свойств ребра (см. рис. 6). Во вкладке «Собственные свойства» задаются парные морфологические свойства в виде «свойство — значение», например: «падеж — вин».

Аналогичным образом осуществляется редактирование свойств узла (см. рис. 7). Под свойствами узла также понимаются парные совокупности, типа «свойство — значение». Название свойств не регламентировано, что позволяет при разработке ATN-анализатора не привязываться к фиксированным именам-константам: например, в качестве имени свойства, описывающего семантическое отношение в использованной реализации ATN-анализатора, выбрано имя «*role*». Семантическая роль описывается как заключенная в круглые скобки последовательность числовых значений, выражающих смещение

[2] Например, морфологический анализатор воспримет словоформу «рабочий» как существительное и прилагательное одновременно, выдавая на выходе описание этих двух лексем. Следовательно, и расширенную сеть переходов необходимо конструировать с учетом многозначности слов.

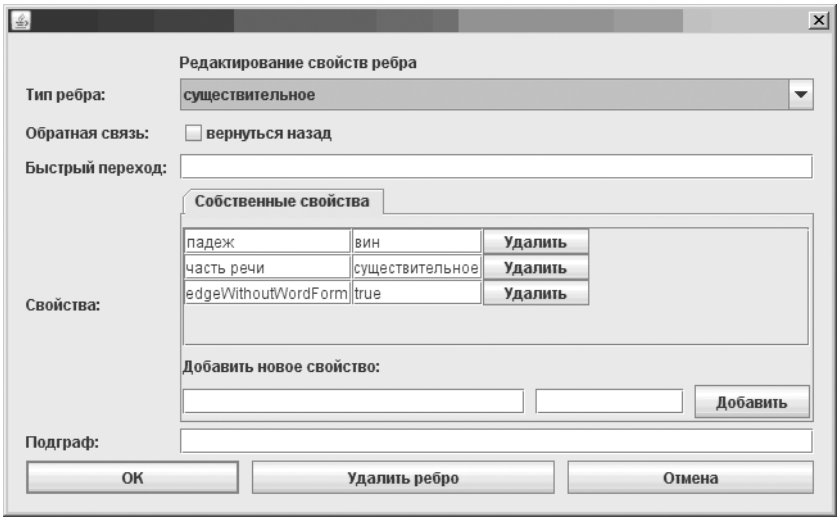

Рис. 6. Экран редактирования свойств ребра

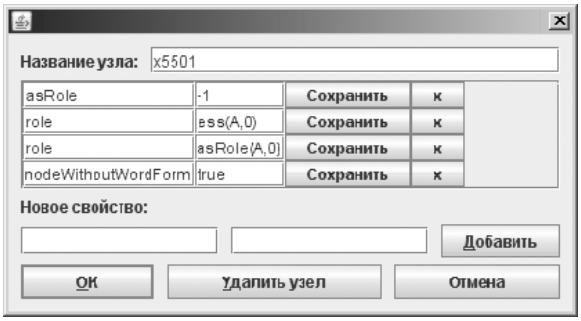

Рис. 7. Экран редактирования свойств узла

влево относительно текущего узла сети. Например, если в семантическое соотношение нужно включить слово, связанное с текущим ребром (т.е. ребром, «правым» концом которого является текущий узел), то его смещение будет равно «0», предыдущее ребро — «−1» и т.д. Порядок указания смещений соответствует порядку в описании семантической роли. Пользователь может выбрать узлы, участвующие в создаваемой роли, путем щелчка мыши по нужным узлам в соответствующем порядке. Затем достаточно выбрать узел для создания роли и указать ее название.

Редактор предоставляет пользователю широкие возможности по визуальному конфигурированию графа на экране, в том числе автоматическое выстраивание узлов сети, обеспечивающие удобное зрительное восприятие картинки, а именно, исключает пересечения ребер графа и представляет его в виде древа. В редакторе реализована функция прокрутки экрана, что позволяет редактировать сети практически не ограниченного объема[3].

Для ускорения конструирования сетей в редакторе представлена возможность автоматического построения ветвей графа по введенному предложению. Данный процесс выглядит следующим образом: пользователь вводит предложение, которое разбивается на отдельные лексические единицы (слова, знаки препинания, цифры), и каждая единица обрабатывается встроенным морфологическим анализатором. В строгой последовательности, в соответствии с введенным предложением, система строит ветвь сети, где каждому ребру приписывается набор морфологических признаков, взятых у соответствующего слова. Пользователю редактора остается только удалить лишние морфологические свойства и добавить нужным узлам сети свойства или семантические роли.

При большом числе вершин графа картинка может оказаться весьма громоздкой и трудной для восприятия. Для устранения этого эффекта предлагается использование фильтров, в частности:

- «только со словоформами» (т. е. отображаются только те дуги, чьи фреймовые структуры содержат конкретные словоформы)
- «только универсальные» (т. е. отображаются только те дуги, чьи фреймовые структуры не содержат конкретных словоформ).

В функционал программы включен также поиск по сетям. Экран окна поиска представлен на рис. 8.

Поиск можно производить в обычном режиме, а также в комбинации режимов «строгий поиск со словоформами» и «полное совпадение морфологических признаков».

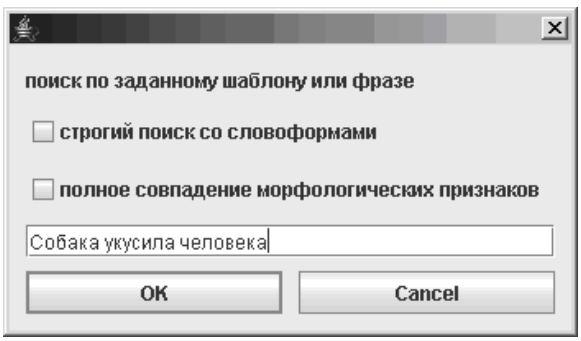

Рис. 8. Экран окна поиска

[3] Объемы ограничены только аппаратными характеристиками ЭВМ, на которой эксплуатируется редактор.

Функция поиска также использует морфологический анализатор. В обычном режиме будут найдены все пути графа, позволяющие произвести разбор данного предложения. В режиме строго поиска отобразятся те пути графа для разбора данного предложения, дуги которых содержат точное совпадение по словоформам введенной строки поиска. В режиме полного совпадения морфологических признаков отобразятся те пути графа, чьи дуги содержат морфологические признаки, которые совпадают с соответствующими признаками слов строки поиска, однако совпадение словоформ в данном режиме не обязательно.

В системе также реализован механизм внутреннего поиска, который запускается каждый раз при вызове автоматической перестройки графа. Данный алгоритм позволяет автоматически отслеживать одинаковые фрагменты сети, расположенные в разных частях графах и при возможности их объединять. Под двумя одинаковыми фрагментами в данном случае понимаются такие, которые имеют началом либо корневой узел графа, либо оканчиваются терминальными узлами (помеченными префиксом «x»), и имеют совпадающий по длине и соответствующим свойствам ребер и узлов некоторый путь. В частности, данный механизм удобен в случае использования автоматического построения «скелета» сети по шаблону-строке, когда после окончания редактирования свойств ребер требуется перестроить граф, объединив одинаковые фрагменты.

Сконструированные сети сохраняются в виде XML файлов, что упрощает применение созданных файлов в стороннем программном обеспечении благодаря легкости обработки языка XML. Классы, описывающие объекты графов, предоставляются в открытом доступе; таким образом, их можно использовать непосредственно при программировании на языке Java для использования сконструированных расширенных сетей переходов. Ниже приведен фрагмент XML файла, описывающий один узел сети:

```
<void property="node1">
  <object id="GraphXNode0" class="g.GraphXNode">
    <void property="name">
      <string>0</string>
    </void>
    <void property="parent">
      <object idref="GraphX0"/>
    </void>
    <void property="properties">
      <void method="add">
        <object class="g.GraphXProperty">
          <void property="name">
            <string>editor_x</string>
          </void>
          <void property="value">
            <string>0</string>
          </void>
        </object>
      </void>
      <void method="add">
        <object class="g.GraphXProperty">
          <void property="name">
            <string>editor_y</string>
          </void>
          <void property="value">
            <string>320</string>
          </void>
        </object>
      </void>
    </void>
  </object>
</void>
```

4. Сравнение с известными графическими редакторами графов

Было произведено сравнение созданного визуального редактора с несколькими известными бесплатными и общедоступными графическими редакторами, применяемых специалистами для создания и редактирования ATN-сетей и Синтаксических деревьев.

1) Augmented Syntax Diagram (ASD) Editor and Parser — редактор Расширенных Синтаксических Диаграмм, предложенный профессором Джеймсом А. Масоном (James A. Mason). Сравнение ASD и ATN сетей произведено проф. Д. Масоном в статье [5]. Отметим схожие черты ASD сетей, их редактора и предложенной в данной статье модели ATN сетей и описываемого визуального редактора:

- использование числовых индексов, или номеров этапов, в узловых метках для различения между разными узлами, которые помечены такими же словарными элементами;
- графический интерфейс, позволяющий конструировать графы на экране, а также изменять местоположение элементов с помощью мыши.

К недостаткам данного ASD-редактора можно отнести:

- недостаточно понятный интерфейс визуального отображения сетей и функций их редактирования;
- отсутствие функций поиска и автоматического построения заготовок сети;
- система адаптирована большей частью для обработки английского языка.

В целом, можно отметить, что данный ASD-редактор отражает в большей степени видение модели автором (проф. Д. Масоном), что накладывает некоторые ограничения на его широкое применение.

2) Linguistic Tree Constructor – визуальный редактор для анализа текстов с помощью синтакси-

ческих деревьев [6]. Программа имеет платформонезависимую реализацию, поддерживает любые языки для описания синтаксических деревьев.

К недостаткам системы можно отнести:

- слишком упрощенную модель отображения деревьев;
- обилие англоязычных сокращений-терминов;
- неудобный графический инструментарий;
- собственные форматы файлов данных, что затрудняет использование их в стороннем ПО.

3) TreeForm Syntax Tree Drawing Software — мощный инструмент для создания и редактирования синтаксических деревьев [7]. Программа имеет платформо-независимую реализацию, поддерживает различные языки при конструировании деревьев.

Можно выделить следующие преимущества системы:

- оригинальный и удобный интерфейс для построения синтаксических деревьев;
- возможность настройки цветовых схем интерфейса пользователя;
- экспорт деревьев в графических форматах (JPEG и PNG);

К недостаткам системы TreeForm можно отнести следующее:

- обилие англоязычных сокращений-терминов;
- отсутствие функции поиска;
- отсутствие нумерации узлов сети.

5. Заключение

Созданный визуальный редактор расширенных сетей переходов позволяет облегчить работу эксперта-программиста лингвистического процессора, основанного на использовании ATN-анализатора. Он включает в себя все необходимые функции, связанные с обработкой графов, назначением свойств ребер и вершин. Включение в состав редактора морфологического анализатора позволяет строить «скелеты» сетей по заданному предложению-шаблону, а алгоритм внутреннего поиска отслеживает наличие подобных структур в уже имеющейся базе знаний, что упрощает создание и расширение существующих сетей.

На использовании морфологического анализатора также основана функция поиска, которая позволяет эксперту быстро находить нужные фрагменты сетей для дальнейшего анализа или редактирования.

Фильтрация отображения позволяет отделить на визуальном уровне универсальные части сети от частей, предназначенных для обработки определенных особых словоформ и конструкций.

В комплексе с данным редактором разработан ATN анализатор, использующий файлы с описанием сетей, созданные с помощью редактора.

В настоящий момент разработанный редактор в совокупности представляемых им набором функций и удобным пользовательским интерфейсом является уникальным инструментом подобного типа.

Литература

1. *Люгер, Джордж, Ф.* Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание. Пер. с англ. — М.: Издательский дом «Вильяме», 2003. — 864 с.
2. *Технологии* компании ПРОМТ. [Электронный ресурс] — Режим доступа: http://www.promt.ru/company/technology/promt/ — Загл. с экрана.
3. *A fast* morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. [Электронный ресурс] — Режим доступа: http://company.yandex.ru/articles/iseg-las-vegas.xml — Загл. с экрана.
4. *Кузнецов И. П.* Семантические представления. М.: Наука, 1986. — 296 с.
5. *James A. Mason.* Augmented Syntax Diagram Grammars. [Электронный ресурс] — Режим доступа: http://www.yorku.ca/jmason/asdgram.htm — Загл. с экрана.
6. *Linguistic* Tree Constructor: About LTC. [Электронный ресурс] — Режим доступа: http://ltc.sourceforge.net/about.html — Загл. с экрана.
7. *TreeForm* Syntax Tree Drawing Software, Version 1.0.3. [Электронный ресурс] — Режим доступа: http://www.ece.ubc.ca/~donaldd/treeform.htm — Загл. с экрана.