И.М. Ножов

Морфологическая и синтаксическая обработка текста (модели и программы) [1]

Научный руководитель -
доктор технических наук,
профессор Д.Г. Лахути

Научный консультант -
Т.Ю. Кобзарева.

Москва - 2003

[1] Internet-публикация содержит исправления и сокращения оригинального текста диссертации, а также изменено первоначальное название *«Реализация автоматической синтаксической сегментации русского предложения»*.

ВВЕДЕНИЕ

Синтаксический анализ является одним из наиболее исследованных направлений в теории computer science. Синтаксические анализаторы широко применяются в таких областях как создание компиляторов, проектирование интерфейсов баз данных, искусственный интеллект (ИИ), автоматическая обработка текстов (АОТ), в том числе для автоматизированных информационно-поисковых систем (АИПС, или «поисковых машин»),

машинный перевод (МП), анализ химических формул и распознавание хромосом. Синтаксическим анализом (parsing) называется процесс структурирования линейной репрезентации в соответствии с заданной грамматикой [D.Grune, C.Jacobs, 1990]. Такое определение, являясь наиболее общим и абстрактным, позволяет охватить весь спектр приложений синтаксических методов. Техникой parsing называется вся совокупность существующих алгоритмов для решения задач синтаксического анализа. Техника parsing берет свое начало в формальных синтаксических теориях естественного языка (ЕЯ), моделирующих механизм распознавания человеком языковых структур. Несмотря на это, именно применение техники parsing в задачах автоматической обработки текста далеко не всегда бывает эффективным и дает положительный результат. Так, например, контекстно-свободные грамматики (context-free grammars) и аппарат конечных автоматов (finite-state automata) широко используются в системах морфологического анализа, снятия омонимии и выделения именных групп внутри предложения, но теряют свое прикладное значение в задачах сегментационного, полного синтаксического и семантического анализа, особенно для языков с относительно свободным порядком слов, каким является русский. Формальные математические модели и их программные динамические реализации не способны охватить всю сложность и многообразие языковой системы. Применение формализма для структурирования предложения естественного языка зачастую приводит к потере правильного синтаксического представления или комбинаторному взрыву, когда программа оказывается не в состоянии просчитать все возможные варианты структур. Лингвистически мотивированные причины такого "провала" - явление омонимии, длина связи между словами, сочинительные конструкции, нарушающие древесность графа, и сложность сегментной структуры предложения. Сфера действия методов распознавания и классификации объектов в лингвистических процессорах тоже сильно ограничена: скрытые модели Маркова удается применить только в узких контекстно-ограниченных задачах снятия морфологической омонимии [Xerox, 1999], нейронные сети используются в системах автоматического распознавания речи [С. Гладунов, О. Федяев, 2002], - такие модели, построенные на обучении и являющие собой альтернативный технике parsing подход, не имеют достаточной силы для отражения способности предложения

естественного языка к неограниченному усложнению. Все эти обстоятельства позволили прикладной (компьютерной) лингвистике выделиться в отдельную область исследования и стать самостоятельно развивающейся ветвью искусственного интеллекта.

Далее в работе мы будем использовать понятие синтаксического анализа только применительно к предложению естественного языка.

Взаимодействие между лингвистикой и computer science началось еще полвека назад с возникновением теории Н. Хомского, развитием генеративизма и появлением электронно-вычислительных машин. Многие лингвистические идеи и концепции на протяжении последних десятилетий были заимствованы и воплощены в программировании, теоретической информатике и информационных системах. Наиболее яркими примерами такого заимствования могут служить базисный компонент порождающей грамматики Н. Хомского, который стал прототипом первых компиляторов искусственных языков, или выдвинутая М. Мински, исследователем в области ИИ, теория фреймов для представления реальных объектов в системах распознавания образов и естественных языков [Г. Буч, 2000], которая сыграла свою роль как в становлении объектно-ориентированного подхода в программировании, так и в семантических исследованиях языка, а наследование и полиморфизм - фундаментальные принципы объектно-ориентированного программирования - стали применяться в проектировании лексиконов [I. Sag, T. Wasow, 1999].

Существует и удивительная связь между естественными и искусственными языками, которая заключается в закономерности эволюции языков. Первый опыт программирования в машинных кодах или на языках низкого уровня, к которым относится ассемблер, характеризуется скорее командным (императивным) стилем, где только упорядоченная последовательность операторов (команд) образует осмысленное действие, подобно тому как в языках с развитым словообразованием последовательная конкатенация грамматических аффиксов порождает слово, обладающее новым значением. С развитием таких языков как ALGOL-60 или COBOL усложняются синтаксические конструкции языка, появляется блочная структура программ. В следующем поколении языков, Pascal и C, текст программы становится похож на многопролетные лестницы, возможность описывать логику действий развернутыми синтаксическими конструкциями задает "ступенчатую" форму

текста. Последнее поколение объектно-ориентированных языков (CLOS, Object Pascal, C++ и Java) стремятся к описанию ключевых абстракций предметной области; абстракции объединяются в библиотеки классов, а программы оперируют объектами этих классов, вызывая методы классов и используя свойства классов, тем самым упрощая синтаксические конструкции, но усложняя структуру объектов и семантические зависимости между ними; текст современный программы напоминает набор коротких четверостиший или деклараций, где каждая строка - обращение к объекту со своим значением и сложной семантикой. Нечто подобное наблюдается и в процессе эволюции естественных языков, когда постепенное вырождение словоизменительной парадигмы в морфологии приводит к ужесточению порядка слов в предложении и фиксации жестких синтаксических конструкций, а последующее усложнение семантики, за счет насыщения языка идиомами и фраземами, за счет появления более абстрактных понятий или новых значений старых слов или за счет пополнения общеупотребительной лексики из научных метаязыков, приводит к упрощению синтаксиса. Конечно, такой сценарий развития не является обязательным и предопределенным для многих языковых групп и семей, но такой путь эволюции до некоторой степени справедлив для италийской группы индоевропейских языков - от латыни к современному итальянскому и французскому - и для группы германских языков.

Разумеется, что такое сравнение программных и естественных языков является во многом условным, но одно можно утверждать с полной уверенностью: "изменчивость - глубинное и универсальное свойство" [С. Бурлак, С. Старостин, 2001] как естественных, так и искусственных языков. Очевидно то, что направления векторов развития систем естественного и искусственного языков совпадают, как и то, что история человеческого языка насчитывает тысячелетия, а искусственных пять десятилетий. Возможно, именно глобальность задачи и разнообразие явлений синтаксиса предложения помноженное на число существующих на земле языков с развитой письменностью оправдывает разработку новых моделей и алгоритмов, отличных от общепризнанных техник parsing или математических моделей, успешно используемых в других областях человеческого знания.

Теоретическая лингвистика и типологический опыт исследования языков создали необходимый описательный аппарат для компьютерного

моделирования автоматического анализа текстов. Множество теоретических подходов можно разделить на два основных направления: формализм и функционализм. Формализм утверждает, что язык есть врожденная компонента человеческого мышления, которая может быть представлена в виде абстрактной модели на метаязыке формальной грамматики и не зависит от способов использования языка, а функционализм напротив полагает, что строение языка определяется его использованием [Я. Тестелец, 2001]. Исследования в формальной лингвистике можно тоже условно разделить на два подхода: построение универсальной грамматики, верной для всех существующих языков мира, и построение формальной модели, наиболее полно охватывающей все множество грамматических явлений конкретного языка. Н. Хомский стал родоначальником первого подхода и основателем школы генеративистов, самым ярким представителем второго подхода является И. Мельчук, автор модели "Смысл ⇔ Текст".

В задачах автоматической обработки текста (АОТ), как правило, используются концепции, разработанные в рамках формализма. Совмещая два подхода формальной лингвистики, программные модели являются лишь частичной реализацией теоретических исследований.

Работы по созданию синтаксического модуля велись еще в конце 60-ых годов, но вычислительная мощность компьютеров не позволяла реализовать сложные алгоритмы анализа в полном объеме. Упрощение алгоритмов и отказ от перебора омонимичных вариантов - компромисс, который приводил к низкой точности синтаксического анализа предложения. Сегодня, по-прежнему, задача автоматизированного анализа синтаксиса ЕЯ сводится к двум параметрам: качеству, определяемому парой «точность (уровень ошибок в построенных синтаксических структурах предложений), полнота (степень покрытия текста синтаксическими связями, или связность графа предложения)», и скорости, пока что недостаточной для ряда прикладных задач.

Ниже будут введены несколько определений понятий, связанных с синтаксическим анализом естественного языка, которые позже получат более точные формулировки. Линейной репрезентацией предложения естественного языка называется цепочка элементов, где каждый элемент является минимальной синтаксической единицей. Минимальная синтаксическая единица может быть словоформой или оператором с определенным набором

характеристик. Оператором называется знак препинания или сочинительный союз. Обязательной составляющей такого набора у словоформы является ее морфологическая репрезентация, обычно состоящая из значения части речи и граммем, а у знака препинания или сочинительного союза - тип оператора (значение, выполняемой им грамматической функции). Таким образом, можно представить линейную репрезентацию предложения в виде цепочки морфологических репрезентаций словоформ и типов операторов.

Процессом структурирования линейной репрезентации предложения называется построение ориентированного графа синтагм и ориентированного графа сегментов.

Синтагма определяет бинарное синтаксическое отношение вида R(A, B), где A и B - словоформы, а R - тип синтаксического отношения, который соответствует имени синтагмы; A является хозяином, B - слугой, т. е. A управляет B. Таким образом, узлами графа синтагм являются терминальные единицы. Связанность не является обязательным условием такого графа, так как синтагмы опираются только на морфологические репрезентации словоформы, линейный порядок предложения и, в некоторых случаях, на примитивную модель управления. На этом уровне анализа связи, для построения которых необходимо использовать сложную модель управления (предикатно-аргументную структуру) или семантическую информацию, могут не фиксироваться в графе синтагм.

Интуитивно сегмент можно определить как часть предложения (в частном случае целиком простое предложение), выделенную на письме знаками пунктуации и описывающую отдельную ситуацию; каждый такой сегмент имеет в качестве вершины явный предикат, выраженный в большинстве случаев финитной формой глагола, или «скрытый» предикат, который может быть выражен либо деепричастием, либо причастием, либо именем с семантической характеристикой действия; каждый такой предикат и задает ситуацию. Близкие по значению понятия в теоретической лингвистике - "предикация" и "элементарное предложение". В западной лингвистической традиции понятие сегмент эквивалентно термину клауза: "клаузой называется любая группа , в том числе и не предикативная, вершиной которой является глагол, а при отсутствии полнозначного глагола - связка или грамматический элемент, играющий роль связки" [Тестелец, 2001]. Например, любое придаточное

предложение (или причастный и деепричастный обороты) в составе сложного является сегментом, равно как и простое предложение в составе сложного образует отдельный сегмент. Сегмент, в терминах системы составляющих, является фразовой категорией (подобно NP, VP, PP, etc. [I. Sag, T. Wasow, 1999]) или нетерминальной единицей. Таким образом, узлами графа сегментов являются нетерминальные единицы.

Морфология, лексема, основа, окончание - понятия и термины, в последние годы ставшие общеупотребительными. Любой грамотный пользователь глобальной сети сможет "на пальцах" объяснить значение этих слов и преимущества поиска информации с использованием морфологии. На сегодняшний день только для русского языка существует несколько десятков известных систем морфологического анализа, число же программ английской морфологии в несколько раз больше. Следующим этапом в развитии направления искусственного интеллекта, занимающегося автоматической обработкой текста, является создание промышленной системы синтаксического анализа естественного языка.

Задача сегментации предложения является первой и, возможно, самой сложной компонентой полного синтаксического анализа. Целью сегментации является выделение и классификация сегментов в составе сложного предложения. Вторая компонента - построение внутрисегментных связей (графа синтагм) - исследована намного глубже и имеет успешные решения, экспериментально подтвержденные на анализе простых (односегментных) предложений. Основной упор в представляемой работе делается на разработку стратегий и методов автоматической системы сегментационного анализа предложения, хотя и предлагается ряд решений, связанных с внутрисегментным анализом терминальных единиц, а также методы моделирования морфологического анализа и снятия омонимии.

В последние десятилетия в странах Западной Европы, США и России проводятся чрезвычайно интересные и перспективные исследования по созданию систем автоматического синтаксического анализа для многих индоевропейских языков. Все попытки моделирования таких систем, как правило, происходят без предварительной сегментации предложения, что приводит к порождению в ходе анализа большого числа ложных синтаксических связей внутри сложного предложения и значительному

снижению скорости анализа. Отсутствие в моделях отдельного сегментационного компонента можно считать одной из основных причин того, что до сих пор не создано эффективных систем синтаксического анализа для русского языка (РЯ) [Т. Кобзарева и др., 2000]. Сегментационный компонент может быть использован и в качестве самостоятельной системы при решении многих прикладных задач автоматической обработки текстов (ИПС, автоматическое реферирование, машинный перевод, etc.). Сегментация предложения, наряду с морфологическим анализом, должна стать базисной составляющей любой полной системы АОТ. Таким образом, создание компонента сегментации русского предложения является чрезвычайно актуальной задачей.

Синтаксический анализ - задача приближения. Любая синтаксическая теория должна обладать описательной и объяснительной силой. Это утверждение с некоторыми оговорками и дополнениями остается справедливым и для прикладных моделей. Описательная сила модели формулируется как максимально возможное покрытие грамматических явлений рассматриваемого языка. Объяснение в теоретической лингвистике заключается в рассмотрении вопроса о существовании в языке именно данных наблюдаемых фактов, а не других [Я. Тестелец, 2001]. В данной работе объяснение понимается в контексте ИИ: любая интеллектуальная система должна уметь обосновать каждый шаг принятых ею в ходе анализа решений [M. Boden, 1990]. Такой критерий подразумевает, что количество эвристик и вероятностно-статистических распределений в системе синтаксического анализа должно быть сведено к минимуму. Существует и третий, не менее важный критерий прикладной модели - эмулирующий принцип построения алгоритмов, - который заключается в способности лингвистического процессора к воспроизведению интуиции и схемы рассуждений человека в процессе изучения и восприятия языка.

Идеальная модель лингвистического процессора состоит из четырех основных анализаторов: графематического (внешнее представление текста), морфологического, синтаксического и семантического. В данном случае мы ограничимся рассмотрением трехсоставного процессора без семантического анализатора.

Целью настоящей работы было создать экспериментальную систему автоматической сегментации русского предложения, демонстрирующую возможность эффективного – с точки зрения качества и скорости анализа – решения этой задачи как ключевого этапа полного автоматического синтаксического анализа русского текста. Основной решаемой проблемой была при этом разработка методов автоматической сегментации предложения и способов программирования, позволяющих минимальными силами решить поставленную задачу применительно к текстам произвольной (или почти произвольной) синтаксической сложности, а также построение прикладной модели лингвистического процессора, удовлетворяющего описательному, объяснительному и эмулирующему принципам.

Предметом исследования является структура сложного предложения русского языка и законы ее построения.

Работа построена на описании и сравнении решений и результатов двух систем синтаксического анализа, использующих компонент сегментации русского предложения.

Синтаксический процессор группы ДИАЛИНГ был создан в рамках проекта русско-английского машинного перевода (1999-2001). Фундаментом для исследований группы ДИАЛИНГ послужила система французско-русского автоматического перевода (ФРАП), разработанная в ВЦП совместно с МГПИИЯ им. М. Тореза в 1976-86 гг., и система анализа политических текстов (ПОЛИТЕКСТ), разработанная в Центре информационных исследований совместно с ВЦ ИСК РАН в 1991-97 гг [Н. Леонтьева, 1995].

Синтаксический анализатор научный группы Отделения интеллектуальных систем (ОИС) Института Лингвистики РГГУ (Д.Г. Лахути, Т.Ю. Кобзарева, И.М. Ножов) был создан в 1999-2003 гг. Предлагаемый проект продолжает развиваться и содержит наиболее полную реализацию идей сегментации русского предложения. Базисом для проводимых исследований послужила модель автоматического поверхностно-синтаксического анализа русского предложения, разработка которой была начата еще в 1971 г. в Информэлектро в секторе (затем отделе) Д.Г.Лахути группой лингвистов под руководством Г.А.Лесскиса .

Также в работе предложены альтернативные подходы к проектированию некоторых составляющих лингвистического процессора, разработанные

автором диссертации в НТЦ "Система" (1997-1998 гг.) и в исследовательском отделе компании Inxight, Software Inc. (2002-2003 гг.).

Методы исследования:

- Создание и пополнение лексиконов, содержащих необходимую для анализа морфологическую и грамматическую информацию;
- Разработка лингвистических стратегий и правил, отвечающих синтаксическим законам языка; изучение множества грамматических явлений, характерных для русского языка; поиск (с использованием конкорданса) случаев применения описываемых грамматических конструкций в корпусе текстов;
- Проектирование общей схемы лингвистического процессора и прикладной модели синтаксического анализа;
- Разработка алгоритмов порождения и перебора структурных вариантов предложения, связанных с явлением морфологической и синтаксической омонимии естественного языка;
- Создание динамических структур данных для представления и хранения синтаксической информации и программное моделирование процесса анализа на ЭВМ;
- Создание отладочного массива предложений, охватывающего все множество отраженных в модели грамматических явлений, и тестирование системы на пространстве реальных текстов.
- Оценка эффективности применения предложенных методов в системах АОТ или МТ.

Научная новизна работы состоит в том, что предложенные алгоритмы порождения структурных вариантов предложения позволили создать успешную модель лингвистического процессора и отказаться от декартова произведения омонимов, проверить работоспособность оригинальных грамматических стратегий анализа и реализовать методы автоматической сегментации без искусственного ограничения на перебор структурных вариантов, обусловленных морфологической и синтаксической омонимией, и без ограничения на глубину рекурсии сегментов и длину предложения.

Практическая значимость работы определяется программными реализациями анализаторов, созданных на базе разработанных методов и

стратегий и получивших практическое применение в различных системах автоматической обработки информации. В диссертации приведены примеры внедрения программ.

В процессе работы над диссертацией автором были получены следующие научные результаты:

1. Разработаны два метода автоматического синтаксического анализа предложения: метод активизации омонимов и рекурсивный метод монтажа разрывных сегментов.
2. Построена прикладная модель синтаксического анализатора, удовлетворяющего описательному, объяснительному и эмулирующему принципам, и позволяющая вести анализ параллельно: "снизу вверх" и "сверху вниз".
3. Отлажены грамматические стратегии сегментации и доказана их работоспособность.
4. Программно реализованы, совместно с другими разработчиками, две системы: промышленный синтаксический процессор группы "Диалинг" и экспериментальная сегментационный анализатор группы ОИС под руководством Д.Г. Лахути.
5. В процессе проводимых исследований и изучения существующих подходов к проектированию лингвистических процессоров автором, совместно с другими исполнителями, были разработаны и внедрены следующие прикладные модули: бессловарный морфологический анализ (НТЦ "Система") и Russian LinguistX Platform 3.5 (Inxight, Software Inc.), включающая в себя tokenizer, stemmer, tagger и np-grouper русского языка.

Апробация работы. Основные выводы и научные результаты диссертационной работы докладывались на международных конференциях Диалог в 2000 и 2001 гг., на национальных конференциях по искусственному интеллекту КИИ в 2000 и 2002 гг. и на научно-технической конференции ВИНИТИ в 2000 г. По теме диссертации автором опубликовано 6 работ. Сдана в печать одна статья.

Структура и объем работы: Диссертация состоит из введения, четырех глав, заключения, списка литературы из 53 наименований и двух приложений. Общий объем работы - 148 страниц, основной текст – 131 страница.

В первой главе приводятся аналогии с химическим строением сложного вещества, шахматной игрой и монтажом фильма, существенные для понимания изложенного в работе подхода к построению модели синтаксической сегментации; рассматриваются современные представления об искусственном интеллекте и его взаимосвязях с естественным языком в аналитической философии; вводятся определения лингвистических понятий релевантных для прикладных моделей; содержится изложение фундаментальных концепций синтаксической теории Head-driven Phrase Structure Grammar (HPSG) и описание ее приложений; рассматриваются синтаксические процессоры английского (LinkParser) и немецкого (STP) языков.

Во второй главе дается описание составляющих лингвистического процессора, которые предшествуют синтаксическому анализатору; рассматриваются различные решения и подходы к проектированию системы морфологического анализа, модуля снятия омонимии и задачи выделения из текста именных групп (NP).

В третьей главе дается описание синтаксического процессора ДИАЛИНГ: системы сегментационных и синтаксических правил, вершины сегментов и синтаксические группы, тезаурусы, элементарные аналитические формы и группы с разрывными союзами; содержится описание сегментационного анализатора группы ОИС: грамматические стратегии сегментации Т.Ю. Кобзаревой и модульность анализа, два типа омонимии (морфологическая и синтаксическая), граф синтагм и граф сегментов, общая схема и прикладная модель сегментации, рекурсивный метод монтажа разрывных сегментов и метод активизации омонимов; приводится сравнительный анализ двух систем.

В четвертой главе диссертации обсуждаются примеры использования и внедрения синтаксических процессоров ЕЯ и их составляющих: бессловарный морфологический анализ в системах автоматического построения словарей и поиска в правовой базе данных НТЦ "Система", технологии Inxight LinguistX Platform в системах АОТ (Murax, Categorizer и Smart Discovery), синтаксический процессор в системе машинного переводчика ДИАЛИНГ, экспериментальные и обучающие возможности сегментационного анализатора группы ОИС.

Создание сегментационного анализатора группы ОИС стало возможным в первую очередь благодаря лингвистико-алгоритмическому аппарату,

разработанному Т.Ю. Кобзаревой, и руководителю проекта д.т.н., профессору Д.Г. Лахути.

Разработка синтаксического процессора группы ДИАЛИНГ - результат коллективного творчества. В разное время в проекте принимали участие следующие специалисты:

1. А. Сокирко (руководитель проекта);
2. Д. Панкратов (русский синтаксис и сегментация, программная реализация);
3. Л. Гершензон (система синтаксических и сегментационных правил);
4. Т. Кобзарева (русский синтаксис и сегментация);
5. И. Ножов (русский синтаксис и сегментация, программная реализация).

Всем участникам проекта ДИАЛИНГ автор выражает свою благодарность.

За техническую поддержку в реализации проекта бессловарного морфологического анализа автор благодарит А.Н. Кудрина (руководителя отдела разработки НТЦ "Система").

Также автор выражает благодарность исследователям компании Inxight, Software Inc. за оказанную техническую поддержку, научные консультации и обсуждения, проводившиеся при создании русской версии LinguistX Platform (tokenizer, stemmer, tagger и np-grouper):

1. Masayo Iida (руководитель отдела лингвистических исследований Inxight, Санта Клара, Калифорния, США);
2. David van den Akker (руководитель департамента разработки Inxight, Антверпен, Бельгия);
3. Carolina Rubio de Hita (ведущий специалист Inxight, Антверпен, Бельгия).

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ И ПРИКЛАДНЫЕ СИСТЕМЫ

I. Синтаксические аналогии

В современной теоретической лингвистике часто используются аналогии, связанные с другими научными дисциплинами и областями человеческого знания, которые помогают наглядно представить и продемонстрировать структурные задачи и подходы к моделированию языковых процессов.

Так, для задачи реконструкции праязыков в компаративистике распространено сопоставление понятия "генетического дрейфа" в биологии и законов распределения фонетических соответствий в языках. Самая популярная и распространенная аналогия в синтаксических теориях связана с химией: строение молекулы и явление изомерии [И. Мельчук, 1999].

В этом разделе будут приведены три аналогии, которые могут быть полезны для понимания задачи сегментации сложного предложения, - химическое строение сложного вещества, шахматная игра и монтаж фильма.

Следуя аналогии Мельчука, попытаемся представить "объемное" предложение с включенными в него придаточными молекулой сложного вещества в органике, состоящей из атомов двух и более видов, где каждый отдельный сегмент играет роль такого атома. Сегмент, в свою очередь, также состоит из конечного множества иерархически организованных элементов, т.е. имеет свою внутреннюю независимую от общей структуру. Как соединения атомов в молекуле образует разные вещества, так и по-разному связанные сегменты образуют сложные предложения, отличающиеся по смыслу. Рассмотрим для наглядности следующий пример: 'Иван, который оставался в городе, сказал, что видел Петра'. Это сложное предложение состоит из трех разнородных простых сегментов: (1) 'Иван сказал', (2) 'который оставался в городе' и (3) 'что видел Петра'. Соединение сегментов '2←1→3' задает смысл приведенного примера, в то время как тип соединения '1→3→2' соответствует предложению с другим общим смыслом: ' Иван сказал, что видел Петра, который оставался в городе', а тип соединения '2→3→1' порождает бессмыслицу. Разумеется, разные типы соединения обусловлены не только

внешними условиями, но и составляющими внутри каждого сегмента, равно как и устойчивые связи между атомами в молекуле зависят не только от физических условий, но и от самих химических элементов. Конечно, аналогия с химическим строением сложного вещества весьма субъективна, но позволяет продемонстрировать тот факт, что в предложении существует некоторая макроструктура, живущая по своим законам и отличающаяся от принятой (состоящей из слов).

Первым ученым, который заметил аналогию между шахматной партией и системой языка, был швейцарский лингвист Фердинанд де Соссюр. Для него шахматы служили удачной метафорой для противопоставления диахронии и синхронии в языке: каждое передвижение фигуры в течение партии изменяет позицию и дальнейшее развитие на доске, причем последствия одного хода могут быть незначительными, а могут иметь необратимые последствия; передвижение фигур во время игры аналогично языковым изменениям в диахронии, а каждая позиция на доске между ходами игроков сравнима с синхронным срезом языка во времени [Ф. де Соссюр, 1999]. Существуют другие, придуманные после Соссюра и не менее интересные шахматные аналогии для естественного языка. Тот факт, что на одной клетке шахматной доски ни в какой момент игры не могут одновременно стоять две фигуры сравнивается с гипотезой единственности заполнения грамматической позиции в предложении, когда, например, не может быть двух подлежащих или двух сказуемых в одном простом предложении [Я. Тестелец, 2001]. Но для представления процесса сегментации нас будет интересовать совсем другое свойство шахматной игры, а точнее способность шахматиста. Способность шахматиста заключается в его интуиции, которая позволяет даже человеку с минимальным опытом игры выбирать фокусное пространство на шахматной доске, т.е. из миллиарда возможных ходов и комбинаций в каждой позиции безошибочно выбирать десяток единственно правильных и осмысленных, не просчитывая остальные. Таких фокусных пространств или ключевых узлов в шахматной позиции может быть несколько, и человек сосредоточивается на выборе одной, самой выгодной на его взгляд, комбинации из десятка возможных, пытаясь просчитать изменение позиции на несколько шагов вперед и предсказать ответы противника. Выбор такого фокусного пространства стал ключевой задачей для программистов и специалистов в области ИИ,

создававших шахматные программы. Оперативной памяти даже самых мощных на сегодняшний день компьютеров не хватает, чтобы просчитать все комбинаторно возможные комбинации на несколько шагов вперед и выбрать из них лучшую. Задача эмуляции такой способности шахматиста была решена через обучающие алгоритмы, но даже сейчас можно легко "повесить" среднюю шахматную программу, сделав в начале партии непредсказуемый, бессмысленный ход, не заложенный в память (схему) программы, который заставит ее потерять фокусное пространство и приступить к перебору всего множества вариантов. Способность шахматиста и существование фокусного пространства на шахматной доске подводит нас к гипотезе, что процесс восприятия и понимания человеком сложного предложения состоит отнюдь не в попытке построения всех связей между словами в этом предложении. Очевидно, что человек запоминает и строит только самые необходимые, базовые связи, вырывая нужные куски-ситуации и забывая остальное, а потом, при необходимости, достраивая и предсказывая подробности. Возможно, поэтому человек всегда пересказывает полученную им информацию "своими словами", добавляя иногда не существующие в реальности подробности. Такой принцип выбора фокусного пространства, безусловно, связан и с явлением языковой избыточности, позволяющей человеку вычислять смысл незнакомого слова в тексте из контекста. Теперь попробуем применить этот принцип к построению сегментов. Рассмотрим следующий пример: 'Девочка, решив уже, когда ее позвали, задачу, засмеялась'. Чтобы собрать сегменты: (1) ' девочка засмеялась', (2) 'решив уже задачу' и (3) 'когда ее позвали', - абсолютно необязательно знать, что 'решив' управляет 'задачу' или 'позвали' управляет 'ее', или 'девочка' зависит от 'засмеялась'. Достаточно знать структурные законы и последовательность действий, которые позволят собрать разрозненные блоки-отрезки ('решив уже' и 'задачу') в один сегмент ('решив уже задачу'), и структурные ограничения, которые не позволят объединить два не относящихся друг к другу отрезка ('девочка' и 'задачу' или 'когда ее позвали' и 'засмеялась') в одно целое ('девочка задачу' или 'когда ее позвали засмеялась'). Таким образом, подобная шахматная аналогия подводит нас к гипотезе о том, что, имея правильную грамматическую стратегию, анализ существующей макроструктуры предложения и сборку разрывных сегментов возможно осуществить без предварительного построения большинства синтаксических связей между словами, т.е. провести анализ

"сверху вниз". Важен и другой вывод: такой способ восприятия предложения является более естественным для человека.

Фильм состоит из эпизодов, а каждый эпизод есть последовательность кадров, выбранная и определенная режиссером из порой огромного объема отснятого материала. Такая смонтированная последовательность кадров должна наилучшим образом выражать все грани и оттенки смысла, чувств и переживаний, столь неуловимых категорий, которыми оперирует художник, и которые он пытается донести до зрителя в этом эпизоде. Основная цель монтажа попытаться расположить кадры фильма в таком линейном порядке, чтобы их конечная непрерывная цепочка возымела максимальное воздействие на сознание зрителя. Сергей Эйзенштейн считается родоначальником русской школы кинематографии и одним из первых теоретиков монтажа. Эйзенштейн определяет монтаж как столкновение и конфликт кусков [С. Эйзенштейн, 2000], оппонируя Пудовкину, для которого монтаж - последовательное сцепление кадров, отражающее логическую последовательность действия в эпизоде. Хороший монтаж различает чередование общего и крупного планов, где в батальных сценах можно противопоставить трагедию общественную (поле боя) и трагедию личную (лицо солдата), или ряд сделанных в эпизод вставок кадров из других временных и пространственных сцен и т.д. Именно такого рода чередования и вставки определяют Эйзенштейновские принципы столкновения и конфликта, делают монтаж интеллектуальным. Техника монтажа, опирающаяся на эти два принципа, повышает экспрессивную (выразительную) силу кинематографа (особенно, это было актуально для немого кино). Эйзенштейн напрочь отметает возможность сцепления - "кирпичики", рядами излагающие мысль. Многие лингвисты склонны объяснять избыточность грамматики естественного языка выразительной силой последнего. Одну и ту же мысль одними и теми же словами можно выразить в пределах одного предложения множеством разных синтаксических конструкций и грамматических построений. Зачем понадобилось в приведенном выше примере ("Девочка, решив уже, когда ее позвали, задачу, засмеялась.") разрывать два цельных сегмента, и почему грамматика русского языка допускает подобные построения, когда ту же "историю" можно было разложить на три простых предложения ("Девочка уже решила задачу. Ее позвали. В этот момент она засмеялась.") или расположить последовательно ситуации ("Решив уже задачу,

девочка засмеялась, когда ее позвали"). Синтаксис языка позволяет вкладывать ситуации друг в друга и чередовать их, разбивая тем самым цельные фрагменты и создавая "матрешечную" структуру предложения. Свойство рекурсивности сегмента, т.е. возможность включать в себя теоретически бесконечное число других сегментов, наглядно демонстрирует экспрессивную (выразительную) силу языка. Необходимо признать и то, что оригинальное построение примера отличается от его двух вариантов неуловимым оттенком или "интонацией" смысла. Человек оперирует сегментами: разбивает, склеивает, чередует их, - выбирая, в конечном счете, оптимальную форму для выражения своей мысли. С точки зрения человека работа автоматического сегментационного анализатора - "демонтирование" составленной им формы изложения, с точки зрения программы процесс сегментации - монтаж логической последовательности «кадров».

Абстрактная структура, в первую очередь, является хорошим инструментом для представления полученных знаний об объекте и удобным средством для описания рассматриваемого объекта, но и сам по себе результат процесса структурирования обладает несколькими важными свойствами [D.Grune, C.Jacobs, 1990]: абстрактная форма позволяет (а) увидеть наиболее общие закономерности, скрытые в линейной репрезентации объекта, (б) продолжить анализ на основе накопленных знаний в структуре, (в) распознать ошибки, допущенные в строении объекта. Все три свойства верны и для синтаксической структуры предложения, которая позволяет (а) увидеть возможные пути трансформации или оценить проективность предложения, (б) перейти на следующий уровень первичного семантического анализа [А. Сокирко, 2001], (в) проверить грамматическую правильность анализируемого предложения. Но синтаксическая структура обладает еще одним уникальным свойством, она физически материализует то, что раньше считалось прерогативой гуманитарного знания, литературоведения или пространных рассуждений эстетов, - это стиль автора. Достаточно беглого взгляда, чтобы увидеть стилистические различия В. Набокова и А. Чехова, представленные структурами фрагментов их произведений на рис. 1 и рис. 2. Обе структуры построены машиной - автоматическим синтаксическим процессором группы ДИАЛИНГ.

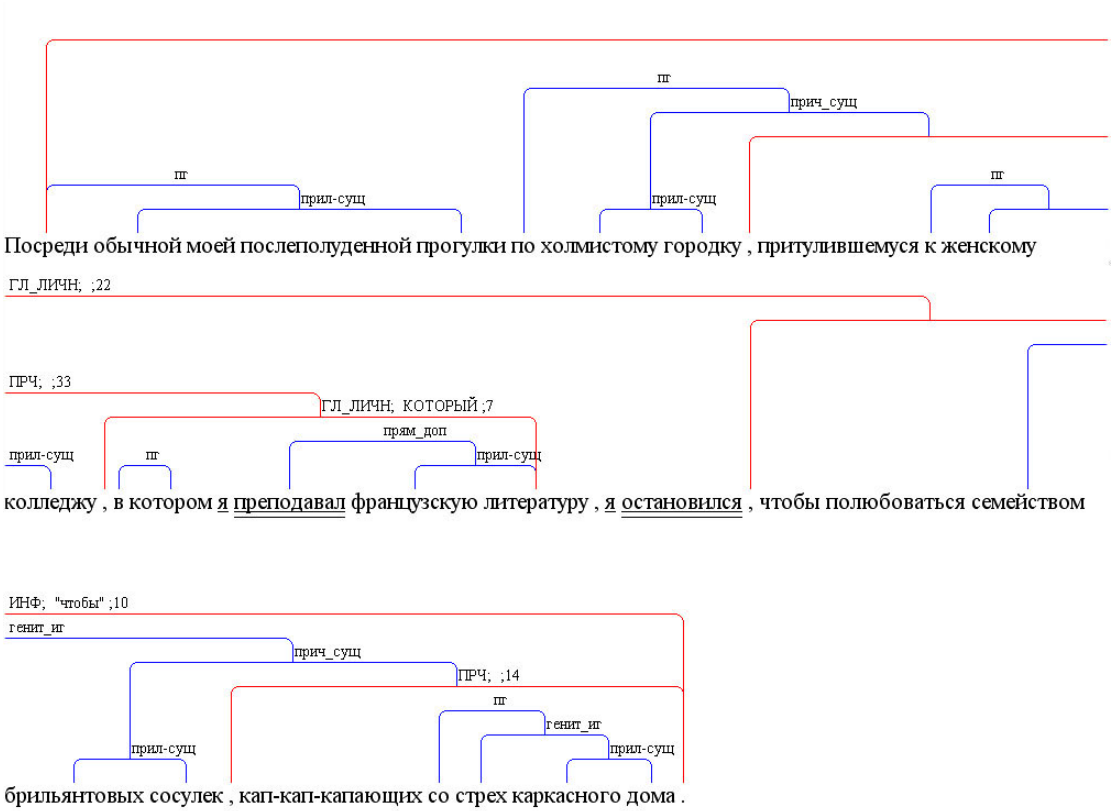


Рис.1 В. Набокова «Сестры Вэйн»

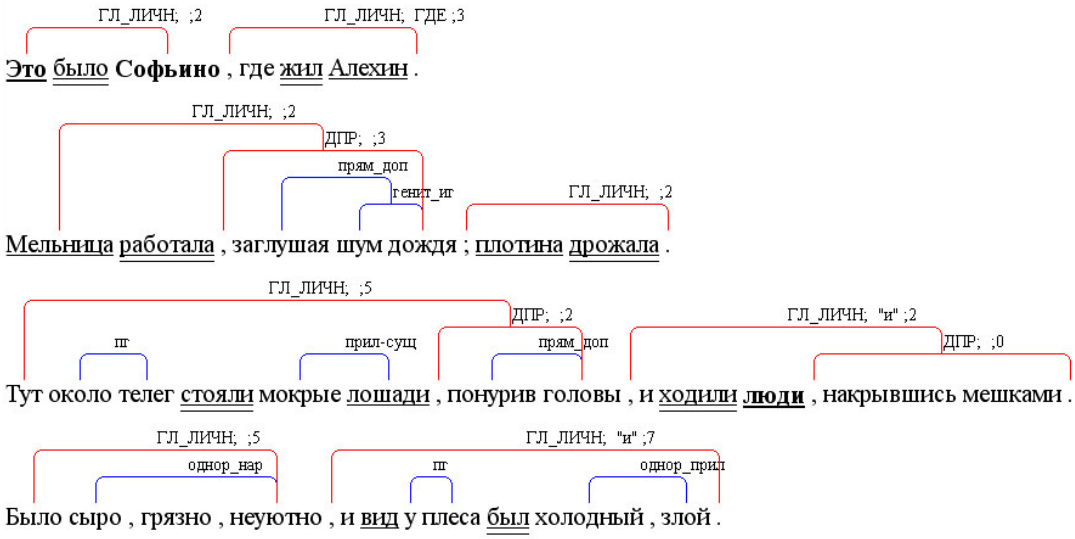


Рис.2 А. Чехова «Крыжовник»

II. Фундамент синтаксического анализа.

В этом разделе обсуждаются основные грамматические средства и понятия (явления), которыми оперирует процессор в ходе автоматического синтаксического анализа. Рассматриваемые явления могут быть как внутренними, относящимися к терминальным единицам и связям между ними, так и внешними, т.е. универсальными структурными законами. Существующий набор явлений в современной теоретической лингвистике намного шире, но, к сожалению, далеко не все из них хорошо формализуемы и могут быть использованы в прикладных моделях. Определения некоторых лингвистических понятий изменены и формулируются только в контексте прикладных моделей, коими являются синтаксический процессор Диалинг и сегментационный анализатор группы ОИС под руководством Д.Г. Лахути.

Все языковые средства, которыми располагает система для определения синтаксических понятий, являются либо свойствами самого объекта, т.е. предложения естественного языка, либо свойствами его элементов, т.е. словоформ и знаков пунктуации (операторов). Синтаксические понятия, по существу, представляют собой функции, где параметрами служат языковые средства, а сами функции используются в условиях грамматических стратегий или правил. Ниже приведены пять языковых средств синтаксического анализа:

1. *Словоизменительные морфологические средства.* Для языков с развитой морфологией, каким является русский, - это основной способ материализации синтаксических связей. Словоформа w_1 морфологически зависит от словоформы w_2 по морфологической категории C, если граммема (значение грамматической категории) g категории C, характеризующей w_1 , выбирается в зависимости от некоторого свойства f словоформы w_2. Словоформа w_2 называется контролером морфологической зависимости, а w_1 - ее мишенью [Я. Тестелец, 2001]. Другими словами, один элемент предложения подстраивается под другой, т.е. принимает грамматическую форму продиктованную вторым элементом. Показателем морфологической зависимости в русском служит флексия, т.к. граммемы в русском обычно приписаны флексии, что позволяет в некоторых случаях обнаружить зависимость между двумя словоформами, отсутствующими в словаре, (например, "глок-ая куздр-а"). Если категория C, по которой наблюдается

морфологическая зависимость, выражается в вершине, налицо вершинное маркирование, если же эта категория выражается в зависимой словоформе - зависимостное маркирование [Я. Тестелец, 2001]. В русском языке граммемы многих форм омонимичны ('ночи' = [[рд., дт., пр., ед.], [им., вн., мн.]] - омонимия числа и падежа), что создает определенные трудности в процессе анализа. Неоднозначность граммем в ходе автоматического синтаксического анализа иногда приводит к возникновению синтаксической омонимии и построению альтернативного синтаксического варианта (графа синтагм). Падежная омонимия с номинативом часто приводит к неоднозначному определению правой границы сегмента и, как следствие, к построению альтернативной структуры сегментации (графа сегментов). Парадокс или скорее взаимовлияние двух уровней анализа морфологического и синтаксического состоит в том, что граммема, являясь эффективным средством поиска морфологической зависимости, которая служит одним из способов реализации синтаксического отношения, может быть однозначно проинтерпретирована только вследствие фиксации этого отношения.

2. *Селективные признаки:* Классифицирующие (селективные) признаки приписываются лексемам в грамматическом словаре, в отличие от граммем, которые вычисляются, исходя из парадигматического класса, для каждой словоформы на этапе морфологического анализа. Наиболее важной для синтаксиса является классификация лексем по категориальным (частеречным) признакам: существительное, глагол, прилагательное, etc. Существует и более дробное деление на субкатегориальные признаки внутри частей речи, так существительные можно разбить на два класса: одушевленные и неодушевленные. Категориальные признаки задают потенциальных участников синтаксической связи и определяют в большинстве случаев вершину в структуре, а также предопределяют понятия управления и согласования. Одушевленность и неодушевленность в русском языке служит контролером согласования для некоторых форм мужского рода или во множественном числе - 'вижу большого [мр., ед., вн.] кролика [мр., ед., вн.] (*большой [мр., ед., вн.] кролика [мр., ед., вн.])' ~ 'вижу большой [мр., ед., вн.] стол [мр., ед., вн.] (*большого [мр., ед., вн.] стол [мр.,

ед., вн.])' или 'вижу четкие фотомодели' ~ 'вижу красивых фотомоделей' (пример Е. Ножовой).

3. *Служебные слова*: предлоги, союзы и союзные слова, вспомогательные компоненты аналитических форм, частицы и т.д. Средства, которые служат в качестве опорных точек анализа. Так, союз может быть использован для определения поверхностного типа сегмента, или вспомогательный компонент аналитической формы содержит недостающие предикату граммемы, или предлог оформляет актант глагола.
4. *Знаки препинания (операторы)*: запятая, тире, точка, вопросительный знак, etc. Это средство не выделяется в теоретических описаниях, так как теоретический синтаксис имеет дело больше с устным языком, чем с письменным, к тому же не все письменные языки, в отличие от русского, имеют жесткие правила расстановки знаков препинания. В первую очередь, операторы определяют границы как сегментов, так и всего предложения. Тире является выражением эллиптированного элемента предложения и часто используется в стратегиях поиска неморфологического предиката. Анализ бифункциональности оператора (когда, например, оператор является одновременно и правой границей сегмента, и оператором сочинения слов) - одна из самых трудных задач сегментации, которая и задает рекурсивный характер как грамматических стратегий анализа, так и методов программной реализации. В теоретических работах принято выделять интонацию как средство синтаксического анализа. Действительно, операторы в письменном тексте являются частичным выражением подмножества синтаксических случаев, характеризующихся интонацией в устном языке. В примере А. Кибрика предложение "В этой гимназии учился впоследствии всемирно известный киноартист", произнесенное с падением интонации на 'впоследствии' имеет синтаксическую связь 'учился → впоследствии', а при отсутствии падения тона - 'известный → впоследствии' [А. Кибрик, 2001]. Такие случаи применения интонации для различения синтаксических связей не фиксируются операторами в письменной форме, поэтому идеальный синтаксический процессор должен решить эту проблему через понятие синтаксической омонимии, построив две равноправных синтаксических структуры предложения.

5. *Порядок слов:* Линейное расположение слов в предложении играет особую роль в изолирующих языках (китайский) и является основным средством для выражения синтаксических отношений в этих языках. Наряду с селективными признаками порядок слов имеет доминирующее значение в проектировании синтаксических анализаторов языков с бедной морфологией (английский). Во многих системах английского синтаксиса порядок слов задает направление поиска хозяина или слуги для каждого класса лексем и типа связи [D. Sleator, D. Temperley, 1991]. Для русского языка это средство анализа имеет второстепенное значение, хотя и применяется в ряде случаев для установление синтаксических связей или оценки омонимичных структур предложения. Несмотря на свободный порядок слов в русском, некоторые синтаксические зависимости имеют обязательным критерием выделения жесткий линейный порядок: генитивное определение должно следовать за определяемым словом ('ножка стол-а', 'сын отц-а'); предлог предшествует существительному ('на стол-е', 'у отц-а'); в 90% случаев определение, выраженное прилагательным или местоименным прилагательным, стоит до существительного (['большой красивый стол', 'седой отец'] ~ ['впечатление необычное']). Порой статистическое расположение синтаксических вершин и их зависимых позволяет разделить все типы синтаксических отношений на три типа: левоветвящиеся (прилагательное существительное: 90%), правоветвящиеся (генитивное определение: 100%) и смешанные (слабые актанты глагола: 50%/50%). Подобные эмпирические распределения могут эффективно использоваться в прикладных моделях. В лингвистической типологии эмпирически установлена универсальная классификация языков мира: языки левого (японский) и правого ветвления (русский и английский). Правда, эта классификация, в основном, строится на статистическом распределении фразовых категорий в линейном порядке предложения, к которым относятся именные (NP), предложные группы (PP) и клаузы (некоторые виды сегментов: придаточные определительные, причастные обороты, etc.). Другая синтаксическая классификация оперирует линейным порядком основных членов предложения: подлежащее (subject), сказуемое (verb) и дополнение (object). Английский относится к языкам Subject Verb Object (SVO) порядка, для русского SVO порядок является статистически доминирующим и наиболее естественным, но грамматически не

единственно возможным. В английском предложении ‘The farmer kills the duckling’ ‘Фермер убивает утенка’ (пример Э. Сепира [Э. Сепир, 1993]) любое изменение порядка слов ведет к изменению смысла всего высказывания ('The duckling kills the farmer' 'Утенок убил фермера.') или к потере грамматической правильности ('* The farmer the duckling kills.'Фермер утенка убил.'), то в русском переводном эквиваленте (‘Фермер убивает утенка’) возможен 3! перестановок, сохраняющих как общий смысл высказывания, так и грамматическую правильность, т.е. в русском варианте данного предложения возможны любые комбинаторные порядки: SVO, SOV, OVS, etc. Таким образом, линейный порядок предложения в автоматическом синтаксическом анализе используется как указатель наиболее вероятного направления поиска слуги или хозяина, и только в редких случаях как обязательный критерий установления синтаксической зависимости.

Ниже будут определены понятия, оперирующие изложенными языковыми средствами, только в рамках их приложения в синтаксическом анализаторе:

1. Согласованием называется пересечение векторов граммем двух словоформ, где ожидаемый результат пересечения определяется категориальными признаками словоформ. Согласование может быть полным или частичным.

Полное согласование:

(а) $V_A \cap V_N$ = [c, Sg, g] || [c, Pl], где V_A - вектор граммем полного прилагательного, причастия или местоименного прилагательного; V_N - вектор граммем существительного; $c \in C$ = [им., рд., вн., дт., тв., пр.] - значение падежа; Sg (ед. ч.) и Pl (мн. ч.) - значения грамматического числа; $g \in G$ = [мр., жр., ср.] - значение грамматического рода.

(б) $V_{Snom} \cap V_P$ = [p≠∅, n] || [g] || [p=∅, Pl], где V_{Snom} - вектор граммем подлежащего, выраженного существительным или местоимением в именительном падеже; V_P - вектор граммем сказуемого, выраженного финитной формой глагола или краткой формой прилагательного или причастия; $p \in P$ = [∅, 1л., 2л., 3л.] - значение грамматического лица; $n \in N$ = [Sg, Pl].

Частичное согласование:

(а) $V_A \cap V_N$ = [c], такой тип согласования используется в дуальных конструкциях (например, "красные стол и стул" или "синий и красный мячи"), в тех случаях

когда еще не построены сочинительные группы. Применение частичного согласования в этих конструкциях зависит полностью от грамматического описания, принятого в прикладной модели. Альтернативный вариант анализа дуальных конструкций состоит в предварительном поиске сочинительных групп, вычисления граммем группы и сведения проверки согласования при последующем установлении атрибутивной связи (именной группы) к полному согласованию типа (а).

(б) $V_{A1} \cap V_{A2} = [c]$, $V_{N1} \cap V_{N2} = [c]$, $V_{P1} \cap V_{P2} = [p \neq \varnothing, n] \parallel [Imptv, n] \parallel [Inf] \parallel [g] \parallel [p=\varnothing, Pl]$, где Imptv - императив, Inf - инфинитив. Подобного рода согласование используется для определения сочинительных конструкций в русском языке.

2. Примитивной моделью управления называется вектор M, определенный в словаре для каждой лексемы L, способной управлять словоформой X. Вектор M лексемы L содержит значения селективных признаков и/или граммемы словоформы X. Вектор $M \subset M| =$ [предлог, подчинительный союз, инфинитив, им., рд., вн., дт., тв., пр.]. Управлением называется пресечение вектора M лексемы L с вектором граммем словоформы X или с значением селективных признаков словоформы X. Явление примыкания и конгруэнтности, а также более сложные случаи управления, не используются в предлагаемых моделях синтаксических анализаторов и считаются прерогативой этапа первичного семантического анализа [А. Сокирко, 2001].
3. Грамматические понятия, построенные на объединении значений селективных признаков в более крупные единицы, используются в синтаксических моделях. Предикат в предложении может быть выражен словоформой с значением части речи $ps \in PS =$ [финитная ф. гл., кр. прил., кр. прич., предикатив]. При построении атрибутивной связи AN A может быть выражено словоформой с значением части речи $a \in A =$ [полное прилагательное, полное причастие, местоименное прилагаельное], а N может быть выражено словоформой с значением части речи $n \in N =$ [существительное, местоимение, субстантивированное прилагательное].

В синтаксических анализаторах изложенные выше понятия обычно оформляются в виде программных функций, которые служат для проверки и установления возможного синтаксического отношения. Таким образом,

изложенные понятия объединяются в более крупных модулях анализа, каковыми являются грамматические правила и стратегии:

1. Каждое грамматическое правило устанавливает один тип синтаксического отношения R(A, B) между двумя единицами анализа и однозначно задает вершину. Число используемых типов отношений, а также их названия, зависит от прикладной модели и конкретной системы, набор универсальных синтаксических отношений для русского языка приведен во многих теоретических работах [Я. Тестелец, 2001]: отпредложное (предлог и управляемое им существительное), определительное (существительное и его согласованное определение), посессивное (существительное и его несогласованное опрделение), субъектное (сказуемое и подлежащее), etc. В роли единиц анализа, на месте A и B, где A - вершина, а B - зависимое, могут выступать как отдельные словоформы, так и целые группы (фразовые составляющие); заполнение A и B во многом зависит от синтаксического аппарата, принятого в анализаторе для описания структуры. Идеальное грамматическое правило в автоматическом синтаксическом анализе характеризуется следующими критериями: (а) описывает только один тип синтаксического отношения; (б) однонаправленность анализа, т.е. зависимое B может находиться только слева или только справа от вершины A; (в) не содержит рекурсивных вызовов или вызовов других правил; (г) обрабатывает только контактно расположенные единицы анализа; (д) результат не зависит от порядка применения правил. Использование грамматических правил задает прозрачность архитектуры процессора и обеспечивает устойчивость системы к изменениям. Перечисленные критерии не являются строгими, но приближают правило к его идеальной форме.
2. Грамматические стратегии, наравне с правилами, используются во всех системах автоматического синтаксического анализа. Типичным примером компонента процессора, построенного на стратегии, является анализ сочинения. Сложность анализа сочинительных конструкций состоит в том, что в процессе построения связи одновременно могут рассматриваться больше чем две единицы анализа; учитываются операторы (знаки препинания и сочинительные союзы) внутри конструкции; нарушается древесность графа, т.к. каждый элемент множества узлов, образующих

сочинительную связь, попарно связан со всеми остальными элементами множества и одновременно является как слугой, так и хозяином узлов, принадлежащих множеству сочинения. Грамматическое сочинение проецируется на все уровни анализа и типы синтаксических единиц, терминальные и нетерминальные: сочинительная конструкция может состоять из теоретически неограниченного числа сочиненных словоформ или именных групп, или предложных групп, или отдельных сегментов (сочиненные придаточные внутри сложного предложения или причастные обороты и т.д.). Стратегии позволяют эффективно организовывать процесс сегментационного анализа. Грамматическая стратегия в прикладных моделях характеризуется следующими критериями: (а) двунаправленность анализа, т.е. зависимое В может находиться как слева, так и справа от вершины А; (б) учитывает единицы, стоящие между потенциальным зависимым и хозяином, в процессе анализа; (в) позволяет строить связи между разрывными составляющими; (г) ищет варианты синтаксической связи для анализируемой единицы, принимая во внимание возможность синтаксической омонимии; (д) может содержать рекурсивные вызовы, оперировать грамматическими правилами и использовать другие стратегии в качестве подпрограмм; (е) оперирует общими структурными ограничениями. Стратегии представляют определенную сложность для программной реализации и гораздо более чувствительны к изменениям в системе, чем правила, но использование стратегий повышает точность анализа, обеспечивает модульность системы и позволяет проектировать сложные схемы взаимодействия компонент модели (см. гл. 3).

Перечислим общие структурные ограничения в прикладных моделях анализа:

1. Проективность. А. Е. Кибрик: Линейная структура предложения проективна, если между каждой парой слов, связанных подчинительной связью, находятся только слова, зависящие (непосредственно или опосредованно) от одного из этих слов [А. Кибрик, 2001]. Я. Г. Тестелец: Предложение называется проективным, если, при том, что все стрелки зависимостей проведены по одну сторону от прямой, на которой записано предложение: (а) ни одна из стрелок не пересекает никакую другую стрелку (принцип непересечения стрелок); (б) никакая стрелка не накрывает корневой узел (принцип необрамления стрелок) [Я. Тестелец, 2001]. Предложения, в

которых нарушается принцип необрамления стрелок, называются слабо проективными, но являются грамматически допустимыми. В реальных системах ограничение на проективность служит для проверки грамматической правильности построенных подструктур в предложении, при этом используется только принцип непересечения стрелок и, как правило, для именных групп (определительная связь) и предложных групп (отпредложная связь), т.к. уже на уровне глагольных групп ограничение на проективность не является строгим и может нарушаться в ряде случаев (в устной речи, художественной литературе или бюрократически-деловых текстах). Структура сегментов предложения является строго проективной, и этот принцип кладется в основу сегментационного анализа (подробнее см. гл. 3). На рис. 1 приведен пример проективной структуры именной группы с несогласованным определением, на рис. 2 показана непроективная и грамматически недопустимая структура именной группы.

Рис. 1

Рис. 2

2. Любая синтаксическая единица (терминальная или нетерминальная) в структуре предложения может непосредственно зависеть только от одной вершины, кроме случая сочинения. В сочинительных конструкциях вершина, входящих в нее единиц, не определена, хотя в некоторых моделях такой вершиной объявляется сочинительный союз, что является формальным допущением, сохраняющим единообразность структурного представления.
3. Простой сегмент предложения содержит только один субъект (подлежащее), кроме случая сочинения субъектов.
4. Простой сегмент предложения содержит только один предикат (сказуемое), кроме случая сочинения предикатов.

Общие структурные ограничения применяются как в ходе синтаксического анализа, так и на этапе оценки равноправных синтаксических представлений, полученных как следствие морфологической или синтаксической омонимии.

III. Гипотеза глубины.

Еще в 1961 году американским ученым В. Ингве была выдвинута гипотеза глубины [В. Ингве, 1965] для синтаксиса естественного языка. Ингве опирался в своих исследованиях на работы в области психологии, где доказывалась, что человек имеет ограничение на объем непосредственной памяти равное семи единицам. Так, человек в среднем способен запомнить с первого раза и правильно воспроизвести около семи десятичных цифр или несвязанных между собой слов.

Ингве использовал в своей работе порождающие грамматики Хомского и компьютерную модель синтеза английского предложения для демонстрации принципа глубины. Линейная последовательность терминальных единиц (с естественным для английского порядком слов - слева направо) порождаемого предложения появляется итерационно, т.е. путем пошагового применения формальных правил разворачивается сверху вниз структура составляющих. При этом, на каждом шаге расширения структура раздваивается на левую составляющую, к которой применяется правило на следующем шаге работы программы, и на правую, хранящуюся в оперативной памяти машины до тех пор, пока левая ветвь синтезируемой структуры не получит интерпретацию на уровне терминальных единиц. Таким образом, чем больше глубина вложения стоящих слева от вершины зависимых, тем больше число промежуточных единиц, хранящихся в оперативной памяти и ожидающих своей интерпретации. На примере английской глагольной группы p_1(p_2(p_3(p_4(very clearly) projected)) pictures) appeared) показана глубина вложения первого элемента ‘very’ в линейной последовательности словосочетания, т.е. в ходе синтеза такого словосочетания, в момент появления терминального узла ‘very’ в порождаемой структуре, в памяти машины будет храниться 4 нетерминальных символа (V, N, A, Adv), соответствующих возможной порождающей грамматике такой группы: VP -> NP + V; NP -> AP + N; AP -> AdvP + A; AdvP -> SecAdv + Adv; V -> ‘appeared’; N -> ‘pictures’; A -> ‘projected’; Adv -> ‘clearly’; SecAdv -> ‘very’.

Анализируя данный пример, Ингве приходит к выводу, что система частей речи обеспечивает способ автоматического подсчета шагов вниз по ветви левостороннего вложения составляющих и прекращения расширения конструкции прежде, чем она пересечет придел глубины. Вершина может быть расширена влево и сентенциальным дополнением, например в английском придаточным предложением с союзом ‘that’: That is true is obvious (“то, что это справедливо, очевидно”). Подобные структуры с расширением вершины влево с n-глубиной вложений называются регрессивными. Регрессивные структуры требуют, чтобы запоминался дополнительный нетерминальный символ для каждого шага развертки вниз. Человек, воспринимающий предложения с регрессивной структурой, также вынужден запоминать слова или группы слов, расположенные до их смысловой вершины. Прогрессивная структура с n-глубиной вложений последовательно расширяется вправо от вершины, сохраняя в оперативной памяти на каждом шаге только один нетерминальный символ. Прогрессивная структура не ограничена объемом памяти, и следовательно может расширяться бесконечно. Примером такой прогрессии служит английское предложение с придаточным ‘that’ в постпозиции к глаголу s(John said cl_1(that Paul said cl_2(that Bill said…))). Синтаксическая регрессия, в отличие от прогрессии, имеет ограничение на глубину вложения, обусловленное объемом непосредственной памяти человека, где максимальное значение n статистически равно приблизительно пяти составляющим. Ингве утверждает, что грамматика естественного языка располагает механизмами для ограничения глубины регрессивных структур, так в английском невозможно вложение придаточных-подлежащих с союзом ‘that’ (That is true is obvious ~ *That that it is true is obvious isn’t clear ~ It isn’t clear that it’s obvious that it’s true), т.е. грамматика языка строится таким образом, что в ней исключаются слишком глубокие конструкции, а взамен их вводятся конструкции меньшей глубины. Ингве отмечает внутреннюю асимметричность структуры английского языка, вызванную накладываемыми ограничениями на возможности ветвления влево, по сравнению с ветвлением направо.

Самым ярким примером регрессивной структуры в русском языке можно считать частотное явление сегментных “матрешек” [Т. Кобзарева, 2002], а именно свойство рекурсивности сегмента. Возвращаясь к примеру,

приведенному в разделе «Синтаксические аналогии», s(девочка, cl_1(решив уже, cl_2(когда ее позвали), задачу), засмеялась) не трудно заметить регрессивный характер этого предложения, но, в отличие от приведенных в работе Ингве английских конструкций, данная русская регрессивная структура не имеет грамматически мотивированного ограничения на глубину и может расширяться, путем вложения новых сегментов, теоретически бесконечно, формально не нарушая грамматической правильности предложения: s(девочка, cl_1(решив уже, cl_2(когда, cl_3(чтобы продолжить, cl_4(…), начатый разговор), ее позвали), задачу), засмеялась). Единственным и естественным ограничением сегментной матрешки ($cl_1(cl_2(\ldots(cl_n())\ldots))$) служит компетенция носителя языка: объем непосредственной памяти, который не позволяет человеку воспринимать предложения с вложенными друг в друга сегментами, где превышается некоторая допустимая глубина. Любопытно совпадение, что в первых компиляторах языка C++ существовало ограничение на глубину вложения шаблонов (templates) [Б. Страуструп, 1999], которое программист вынужден был соблюдать при написании программ.

Впоследствии гипотеза глубины послужила основанием для создания (уже в терминах вершинных грамматик) универсальной типологической классификации, разделившей языки мира на языки левого и правого ветвлений. В терминологии Ингве, языки левого ветвления (аварский, японский) тяготеют к регрессивной структуре предложения, а языки с правым (английский, русский) – к прогрессивной структуре. Нельзя полностью принять утверждение ученого о том, что каждый язык располагает способами ограничения регрессии и способами, дающими возможность обойти ограничения объема памяти. Существование языков левого ветвления и явление сегментной матрешки в русском демонстрируют факультативность таких грамматических механизмов ограничения глубины (хотя попытки решения загадки лево- и правоветвящихся языков предпринимались Дж. Хокинсом через понятие, определенное им, как сфера идентификации составляющих (constituent recognition domain) [Я. Тестелец, 2001]). Ингве приходит к неоспоримому выводу, что глубина есть фактор, влияющий на развитие языка.

Существует три единственно возможных линейных расположения зависящего сегмента от сегмента-хозяина:

1. сегмент-слуга стоит слева от сегмента-хозяина (инверсное распределение);

2. сегмент-слуга стоит справа от сегмента-хозяина (последовательное распределение);
3. зависимый сегмент разрывает главный сегмент, то есть зависимый сегмент вложен в подчиняющий сегмент (гнездование).

При этом, первый и третий типы расположения задают регрессивную структуру предложения, а второй – прогрессивную.

Стоит сделать предположение, что возможно провести аналогичную левому и правому ветвлениям классификацию естественных языков по приведенным выше трем типам распределения сегментов в предложении с использованием информации о грамматических классах сегментов (см. гл. 3). Также интересна оценка грамматически допустимой глубины гнездования в разных языках.

Гипотеза глубины и свойство рекурсивности сегмента являются базисом для понимания структуры сложного предложения и для разработки подхода к задаче автоматической сегментации.

IV. **Head-driven Phrase Structure Grammar (HPSG).**

> В лингвистике гораздо легче создать свою новую теорию, чем разобраться в уже существующей чужой…

В начале 90-х годов в американкой математической лингвистике активно разрабатывался новый класс грамматик для анализа синтаксической структуры естественного языка, основанный на лексическом подходе и критике контекстно-свободных грамматик (CFG). Одним из первых вариантов такой теории явилась грамматика GPSG (Generalized Phrase Structure Grammar), разработанная еще в конце 80-х гг., которая достаточно быстро эволюционировала в грамматику управляемых вершинами фразовых категорий - Head-driven Phrase Structure Grammar (HPSG). Главными идеологами HPSG стали ученые Стэндфордского Университета И. Саг и Т. Васоу, создавшие компьютерную лабораторию для экспериментальных исследований прикладных возможностей HPSG. Данный класс грамматик отличает два тезиса:

- Построение иерархической структуры свойств (feature structure) каждой лексической единицы языка, содержащей грамматическую и семантическую информацию, и проектирование лексикона с иерархической организацией

типов свойств, где каждый тип-потомок может наследовать и переопределять свойства предка (такая система организации лексикона во многом следует объектно-ориентированной модели программирования).

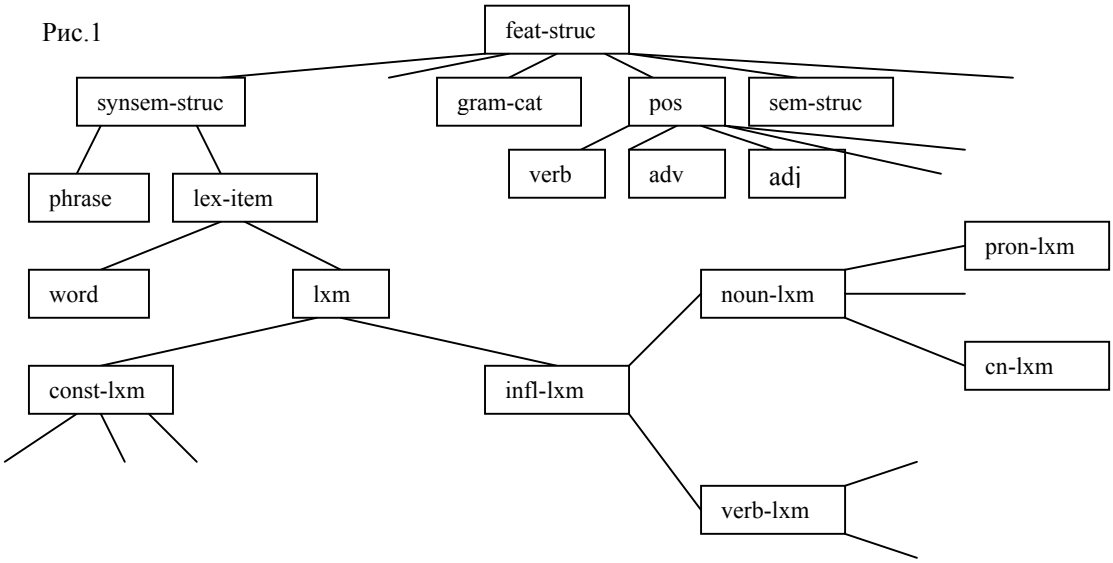
- Унификация – как базовый механизм построения синтаксической структуры.

Многие теоретические постулаты HPSG заимствованы из теории принципов и параметров, позднего варианта порождающей грамматики (ПГ) Хомского, а именно, из ее базового модуля X'-теории. Сторонниками ПГ в X'-теории признается необходимость определения вершины структуры фразовых категорий и производится отказ от базового компонента, т.е. формальных правил генерации предложения [Я. Тестелец, 2001]. В X'-теории доминирующим становится лексический подход (через словарь) к построению синтаксической структуры. Грамматика HPSG не использует понятие проекции составляющей и сохраняет базовый компонент, в качестве дополнительного инструмента механизма унификации, сводя количество правил грамматики к минимуму и делая их максимально общими. Сохраняя правила, HPSG, в отличие от X'-теории, теряет универсальность грамматики, но приобретает практическую значимость для программных реализаций: разработанный механизм унификации позволяет проектировать эффективные прикладные системы синтаксического анализа.

Критика CFG сторонниками лексикализма состоит в том, что (а) контекстно-свободные грамматики произвольны (отсутствие у фразовых категорий вершины и ее свойств); (б) CFG избыточны (простейший случай, когда возникает избыточность, отсутствие возможности проверки согласования).

На рис.1 приведен фрагмент иерархии типов для английского языка, принятый в структуре лексикона HPSG, и таблица свойств/ограничений по умолчанию, присвоенных каждому типу.

Рис.1



Фрагмент таблицы общих типов от базового предка feat-struc (структура свойств) к потомку cn-lxm (нарицательные существительные):

Тип	**Свойства/ Ограничения**	**Комментарии**
feat-struc (структура свойств)		базовый абстрактный тип
Synsem-struc (синтактико-семантическая структура)	[SYN gram-cat; SEM sem-struc]	SYN – свойство, описывающее грамматический компонент лексемы, которое задается структурами свойств, определенных в типе gram-cat; SEM - свойство, описывающее семантический компонент лексемы, которое задается структурами свойств, определенных в типе sem-struc.
gram-cat (грамматическая категория)	[HEAD pos; COMPS list(synsem-struc); SPR list(synsem-struc)]	HEAD – структура свойств вершины; COMPS – список возможных комплементов, заданных значениями структурного типа synsem-struc; SPR - список возможных спецификаторов, заданных значениями типа synsem-struc.
lex-item (лексическая единица)	[ARG-ST list(synsem-struc)]	ARG-ST – аргументная (актантная) структура (argument structure), заданная списком synsem-struc.
lxm (лексема)	[SEM [MODE / none]]	MODE – модальность, одно из “подсвойств” свойства SEM, принимает по умолчанию пустое значение и может быть

		переопределено в значении типа-потомка.
infl-lxm		Абстрактный тип
noun-lxm	[SYN [HEAD [noun: AGR [PER / 3rd]; ANA / -]]; ARG-ST / <>; SEM [MODE / ref]]	Свойство HEAD состоит из значения типа noun (существительное): AGR (согласование) имеет в своем составе свойство PER (лицо) со значением по умолчанию 3, ANA (анафор) с отрицательным значением по умолчанию (которое может быть переопределено в типе-потомке для референциальных местоимений); ARG-ST по умолчанию задается пустым списком. MODE в SEM присваивается значение по умолчанию ‘референция’.
cn-lxm (common noun lexeme)	[SYN [HEAD [AGR (1)]; SPR <[]>]; ARG-ST / <[DetP; AGR (1)]>]	Значение согласования AGR сворачивается до идентификатора (1), SPR – свойство вершины присоединять спецификатор, заданный по умолчанию списком, состоящим из одного элемента; ARG-ST состоит из одного элемента, выраженного детерминатором и ограничением AGR, которое должно совпадать по идентификатору с аналогичным свойством вершины.

Пример словарного входа лексемы ‘book’:

<book, [cn-lxm: ARG-ST <[COUNT +]>]; SEM […]>, где положительное значение свойства COUNT (исчисляемость) – ограничение на значение аргумента. Здесь мы опускаем значение семантического компонента, т.к. нас интересует, в первую очередь, устройство синтаксической структуры (семантический компонент HPSG описывает ситуацию, используя смысловые отношения, и позволяет вычислять смысл всего предложения путем конкатенации значений семантических свойств его составляющих).

Применив принцип наследования от типа-предка для лексемы ‘book’, мы получим полную структуру свойств:

<book, [cn-lxm: SYN [HEAD [noun: AGR (1); ANA / -; SPR<[]>]]; ARG-ST <[DetP; AGR (1); COUNT +]>]; SEM [MODE / ref; …]>

В лексиконе HPSG существует множество лексических правил (Lexical Rules), позволяющих построить словоформу и ее свойства от данной лексемы. Простейшими примерами таких правил могут служить правило для

единственного числа существительного (Singular Noun Lexical Rule): <(1), [noun-lxm]> => <(1), [word; SYN [HEAD [AGR [NUM sg]]]]> или правило множественного числа (Plural Noun Lexical Rule): <(1), [noun-lxm, AGR-ST <[COUNT +]>]> => <F_{NPL}(1), [word; SYN [HEAD [AGR [NUM pl]]]]>, где F_{NPL} – морфологическая функция, присоединяющая флексию для формы множественного числа: (book) => F_{NPL}(book) = books.

Основной недостаток такого лексикона для прикладных моделей анализа – трудоемкость разработки. Очевидно, что для русского языка число типов и лексических правил сильно возрастет. Отсутствие разделения анализа на уровни и словари (морфологический, синтаксический и семантический) лишает архитектуру лексикона прозрачности. Для языков с развитой морфологией намного эффективней задавать ограничения (constraints) по согласованию процедурно. Алгоритмический подход к синтаксическому анализу позволяет сводить к минимуму использование статических данных, тогда как лексикализм и успешность работы грамматик, построенных на унификации, целиком зависят от полноты лексикона.

Унификацией называется наиболее общий метод, позволяющий двум совместимым дескрипциям структуры свойств соединять информацию, которую они содержат, в одну (обычно большую) дескрипцию. Две дескрипции являются совместимыми в том случае, если они не содержат в своих структурах конфликтующих типов или разных атомарных значений одних и тех же свойств. Если дескрипция D_1 определена множеством структур свойств σ_1 и D_2 определена множеством σ_2, тогда унификация D_1 и D_2 определена пересечением σ_1 и σ_2. Допустим, существует частное грамматическое правило для построения фразовой структуры с учетом согласования, типа [phrase: POS (1); NUM (2)] → [word: POS (1); NUM (2)] + NP, где POS – свойство селективного признака и NUM – свойство категории числа. Тогда два вхождения идентификатора (1) и два вхождения идентификатора (2) означают, что значение свойства POS и значение свойства NUM материнского узла в левой части правила и соответствующие значения первого дочернего узла в правой части правила должны быть унифицированы. В лексиконе HPSG разные структуры свойств могут быть вложены одна в другую, что позволяет создавать сложные иерархические структуры, а значение селективного признака определяется

лексическим типом (pos → verb, adj, noun, etc. см. рис.1). Таким образом, свойство вершины HEAD для лексической единицы можно определять через тип, соответствующий значению селективного признака, где каждому такому типу приписано свойство согласования AGR: например, [HEAD [noun: AGR [PER 3rd; NUM pl]]], свойство HEAD является сложной иерархической структурой. В этом случае можно утверждать, что два элемента согласованы, если унифицированы спецификации их свойств AGR. Теперь можно переписать приведенное выше частное правило в более общем виде: [phrase] → H[word] + NP, где 'H' маркирует вершинный дочерний узел, который содержит идентифицируемую с материнским узлом структуру свойств HEAD.

В HPSG вводится два универсальных синтаксических принципа:

- Принцип вершины HFP (Head Feature Principle)

  Для любой фразовой категории, где определена вершина, значение свойства HEAD материнского узла и значение свойства HEAD дочернего узла должны быть унифицированы.
- Принцип модели управления (The Valence Principle)

  Значения свойств SPR (спецификатор) и COMPS (комплементы) материнского узла идентичны значениям аналогичных свойств вершинного дочернего узла.

Аналогичным образом метод унификации используется и при построении семантической структуры (свойство SEM), для этого в грамматике определяются дополнительные принципы.

Базовый компонент грамматики HPSG в упрощенном виде состоит из четырех максимально общих синтаксических правил [I. Sag, T. Wasow, 1999]:

1. Правило комплемента вершины (Head-Complement Rule)

   [phrase: COMPS <>] → H[word: COMPS <(1),…,(n)>] (1) … (n) , где n – идентификатор комплемента.

   Фразовая категория может состоять из лексической вершины и следующих за ней комплементов; в частном случае список комплементов пуст.
2. Правило спецификатора вершины (Head-Specifier Rule)

   [phrase: SPR <>] → (1) H[phrase: SPR <(1)>]

   Фразовая категория может состоять из фразовой вершины и предшествующего ей спецификатора.

3. Правило модификатора вершины (Head-Modifier Rule)

   [phrase] → H(1)[phrase] [phrase: MOD (1)]

   Фразовая категория может состоять из фразовой вершины и следующего за ней совместимого фразового модификатора.

4. Правило сочинения (Coordination Rule)

   [SYN (0); IND s_0] → [SYN (0); IND s_1] … [SYN (0); IND s_{n-1}] [HEAD conj; IND s_0] [SYN (0); IND s_n], где семантическое свойство IND - индекс некоторой ситуации.

   Любое число вхождений элементов с одинаковой синтаксической структурой (свойство SYN) могут быть соединены в один сочинительный элемент той же структуры.

Приведенный базовый компонент грамматических правил обладает тремя недостатками: (а) жесткий линейный порядок составляющих в правой части правила, что не позволяет использовать такого рода правила в языках с относительно свободным порядком синтаксических составляющих, каким является русский (то же относится и к структурным свойствам лексикона HPSG, где строго определен порядок следования комплементов лексемы, так [COMPS <NP, PP>] означает, что в линейной цепочке предложения именная группа, управляемая данной лексемой, должна стоять перед предложной); (б) правила не способны анализировать слабо проективные структуры, грамматически допустимые во многих языках; (в) абсолютная зависимость синтаксических правил от правильности и полноты структур свойств отдельно взятого словарного входа лексикона.

Рассмотрим пример синтаксического анализа грамматикой HPSG английского предложения ‘They sent us a letter’ (‘Они послали нам письмо’) без учета семантической структуры:

<they, [word: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]>	<sent, [word: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS < NP_j[CASE acc], NP_k[CASE acc]>]]>	<us, [word: SYN [HEAD [noun: CASE acc; AGR [PER 1rd; NUM pl]]; SPR <>; COMPS <>]]>	<a, [word: SYN [HEAD [det: COUNT +; AGR [3sing]]]]>	<letter, [word: SYN [HEAD [noun: AGR [3sing; GEND neut]]; SPR <D[AGR [3sing; GEND neut]; COUNT +]>; COMPS <(PP)>]]> *круглые скобки комплимента означают его факультативность,*

				в данном случае предложной группы PP
<they, [word: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]>	<sent, [word: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS < NP_j[CASE acc], NP_k[CASE acc]>]]>	<us, [word: SYN [HEAD [noun: CASE acc; AGR [PER 1rd; NUM pl]]; SPR <>; COMPS <>]]>	<a, [word: SYN [HEAD [det: COUNT +; AGR [3sing]]]]>	<NP, [phrase: SYN [HEAD [noun: AGR (1)]; SPR <D[AGR (1); COUNT +]>; COMPS <>]]> *обнуляется список комплементов в соответствии с правилом комплемента вершины и идентифицируются значения свойств HEAD и SPR в соответствии с HFP*

<they, [word: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]>	<sent, [word: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS < NP_j[CASE acc], NP_k[CASE acc]>]]>	<NP, [phrase: SYN [HEAD [noun: CASE acc; AGR [PER 1rd; NUM pl]]; SPR <>; COMPS <>]]> *правило комплемента вершины и HFP*	<NP, [phrase: SYN [HEAD [noun: AGR [3sing; GEND neut]]; SPR <>; COMPS <>]]> *унификация произошла в соответствии с правилом спецификатора вершины и отвечает принципу HFP*

<NP, [phrase: SYN [HEAD [noun: CASE nom; AGR [PER 3rd; NUM pl]]; SPR <>; COMPS <>]]> *правило комплемента вершины и HFP*	<VP, [phrase: SYN [HEAD [verb]; SPR <NP_i[CASE nom]>; COMPS <>]]> *принцип модели управления, правило комплемента вершины и HFP*
<VP, [phrase: SYN [HEAD [verb]; SPR <>; COMPS <>]]> *принцип модели управления, правило спецификатора вершины и HFP*	

Порядок применения синтаксических правил в HPSG – свободный.

Успешными унификациями называется цепочка унификаций, которая приводит к построению связного синтаксического дерева, т.е. структура предложения сворачивается до уровня одной вершины с единой структурой свойств. Одними из факторов, влияющих на количество ложных унификаций в ходе анализа, являются факультативные (слабые) комплементы и морфологическая омонимия, опущенная в рассмотренном выше примере (‘letter’

имеет значение как существительного, так и глагола в английском). Так же очевидно, что в языке со свободным порядком составляющих, с высоким коэффициентом глубины вложения и возможностью прерывания составляющих, число ложных унификаций сильно увеличится, а значит, и уменьшится скорость анализа.

На основе грамматики HPSG в Стэндфордской лаборатории создается система автоматического синтаксического анализа английского предложения, программная реализация процессора осуществляется на функциональном языке программирования LISP [S. Oepen, J. Caroll, 2000]. Пока что объем лексикона и скорость процессора не позволяют проводить анализ сложных предложений. Для отладки работы и развития анализатора используется приложение тестового обеспечения для естественно-языковых процессов TSNLP (Test Suites for Natural Language Processing), которое содержит базу данных тестовых примеров и результатов анализа [S. Oepen, K. Netter, 1997]. Синтаксический процессор HPSG контролируется системой, осуществляющей наблюдение (profiling) и оценку скорости работы отдельных функций анализатора, что позволяет вести протокол эволюции программной модели с учетом вносимых в нее изменений. Такой инструмент profiling имеет около сотни параметров оценки (таких как число ложных и успешных унификаций, время работы процессора ЭВМ для отдельно взятой операции, объем выделенной динамической памяти, etc.) и дает возможность выявлять критические зоны для скорости работы синтаксического анализа [S. Oepen, J. Caroll, 2000].

Несмотря на указанные недостатки подхода лексикализма и недостатки базового компонента унифицирующей грамматики, необходимо признать большой экспериментальный потенциал построенной на HPSG модели для исследователей в области ИИ.

V. Link Grammar Parser (LinkParser).

Противоположный HPSG подход к синтаксическому анализу английского языка был разработан группой американских исследователей (Daniel Sleator, Davy Temperley и др.) в самом начале 90-х гг., этот проект получил название Грамматика Соединений (Link Grammar). Базовое отличие Link Grammar состоит в том, что используемая модель анализа является

контекстно-свободной грамматикой CFG [D. Sleator, D. Temperley, 1991]. В отличие от HPSG, абстрактной и универсальной синтаксической теории ЕЯ, Link Grammar с самого начала создавалась как аппарат для автоматической системы анализа предложения, что позволило авторам отойти от академических представлений, принятых в лингвистической традиции.

Каждая единица словаря грамматики описывается формулой, состоящей из соединителей (коннекторов connector). Коннектор состоит из имени типа связи (например, S – субъект, O – объект, CL – сегмент и т.д.), в которую может вступать рассматриваемая единица анализа, и суффикса, определяющего вектор направления соединения (‘+’ право-направленный коннектор и ‘-’ лево-направленный коннектор). Лево-напрпавленный и право-направленный коннекторы одного типа образуют связь (соединение link). Так, два слова W_1 и W_2, имеющие словарные входы W_1: A^- и W_2: A^+, образуют соединение A в линейной последовательности W_2W_1, но не связаны в цепочке W_1W_2.

Язык формул, оперирующий коннекторами, использует четыре связки: оператор конъюнкции &, оператор дизъюнкции or, фигурные скобки {} для обозначения факультативности и неограниченность повторения @ (эквивалент оператора + Клини). Так, в формуле слова W: D^- & {@A^-} выражение ‘@A^-’ означает, что должна быть реализована связь с дескриптором A слева от W хотя бы один раз, и может повторяться неограниченное число раз; выражение ‘{@A^-}’ означает, что связь A факультативна. Конъюнкция несимметрична для однонаправленных коннекторов и задает строгий порядок слов в предложении: в формуле W: A^+ & B^+ слово, реализующее соединение A, должно находится ближе к W в линейной последовательности предложения, чем слово, реализующее соединение B, в той же последовательности. Для разнонаправленных коннекторов конъюнкция симметрична: формулы W: A^- & B^+ и W: B^+ & A^- эквивалентны.

Проблема избыточности словаря решается в системе LinkParser путем разбиения слов английского языка на 23 класса, где каждому такому классу приписывается своя формула. Разумеется, существует слова и подмножества слов-исключений, которые получают отдельную от основных классов формульную интерпретацию (к ним относятся, например, описание модальных глаголов или референциальных местоимений). Слова обобщаются в классы по селективным и субкатегориальным признакам. В ходе анализа словам в системе

приписываются значения их базовых классов – селективных признаков (‘cat.n ran.v’).

Тип коннектора задается именем, где начальные заглавные буквы являются базовым дескриптором, а нижний составной индекс, как правило, задает значение граммемы, что позволяет косвенно проверять согласование или необходимое управление при установлении связи (например, ‘S^{+}’ – существительное, ‘dogs ideas: Sp^{+}’ – существительное во множественном числе, ‘dog idea: Ss^{+}’ - существительное в единственном числе). Таким образом, могут соединяться либо равные коннекторы, либо два коннектора, один из которых выше уровнем: ‘Spa^{+}’ может соединяться с ‘S^{-}’, ‘Sp^{-}’ или ‘Spa^{-}’, но не с ‘Ss^{-}’ или ‘Spb^{-}’. В анализаторе LinkParser используется около ста различных коннекторов, различающихся преимущественно нижнем индексом, число базовых дескрипторов - сравнительно небольшое.

В LinkParser вводятся общие структурные ограничения:

- Проективность: связи между словами в предложении не пересекаются.
- Полнота связей: все слова в линейной последовательности должны быть соединены между собой.
- Порядок: в линейной цепочке предложения должен выполняться порядок реализаций соединений, заданный в формуле несимметричной конъюнкцией для однонаправленных коннекторов.
- Исключение: для одной пары слов не может быть проведено больше одной связи.

Рассмотрим пример анализа простого предложения ‘The cat chased a snake’ (‘Кошка преследовала змею’).

Фрагмент словаря:

Словоформа	Формула
the a	D^{+}
cat snake	D^{-} & (O^{-} or S^{+})
Chased	S^{-} & O^{+}

Результат анализа:

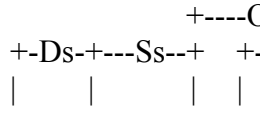


рис. 1

Нетрадиционность модели Link Grammar состоит и в том, что разработчики отказались от системы составляющих, столь популярной для

представления синтаксической структуры английского языка, и используют формализм, идеологически близкий к теории зависимостей, описанной в работах И. Мельчука. В отличие от деревьев зависимостей, бинарные связи, строящиеся LinkParser, не содержат вершины и не имеют направления. Используя информацию о селективных дескрипторах, приписанную терминальным единицам предложения, и тип коннекторов, маркирующих соединения, можно транслировать построенную LinkParser проективную структуру (linkage) в классическое дерево зависимостей, такая же трансляция возможна, рассматривая вложения соединений, и в систему непосредственных составляющих, определенных в выходной структуре анализатора.

Чтобы получить для каждого слова множество его однозначных интерпретаций (т.е. последовательностей лево-направленных и право-направленных коннекторов), формула, приписанная каждому слову в предложении, приводится к ее дизъюнктивной форме. Дизъюнктивной формой называется конечное множество дизъюнктов формулы. Дизъюнкт имеет вид $((L_1, L_2, \ldots, L_m) (R_n, R_{n-1}, \ldots, R_1))$, где $L_1, L_2, \ldots, L_m$ лево-направленные коннекторы, а $R_1, R_2, \ldots, R_n$ право-направленные. В стандартной форме дизъюнкт можно представить в виде формулы, использующей только оператор конъюнкции: $(L_1 \& L_2 \& \ldots \& L_m \& R_1 \& R_2 \& \ldots \& R_n)$. Тогда формулу

$(A^- \text{ or } ()) \& D^- \& (B^+ \text{ or } ()) \& (O^- \text{ or } S^+)$

можно представить в дизъюнктивной форме как множество из восьми дизъюнктов:

((A, D) (S, B))
((A, D, O) (B))
((A, D) (S))
((A, D, O) ())
((D) (S, B))
((D, O) (B))
((D) (S))
((D, O) ())

Дизъюнкты используются в алгоритме автоматического синтаксического анализа на основе Link Grammar.

Морфологическая омонимия задается при анализе дизъюнкцией формул двух омонимов (например, терминальной единице ‘letter’ в предложении соответствуют два словарных входа ‘letter.n: f_1’ и ‘letter.v: f_2’, тогда ‘letter: f_1 or

f_2’). Синтаксическая омонимия – множество всех построенных структур одного предложения, отвечающих выше перечисленным структурным ограничениям.

Для решения задачи сегментации в грамматику LinkParser введены специальные типы коннекторов: TH, CL, CO, X, B, etc. Глаголы, способные в качестве дополнения присоединять сегмент (clause), содержат в своей формуле TH^+ (придаточное с союзом ‘that’) или CL^+. Глагол с коннектором TH^+ образует связь TH с подчинительным союзом ‘that’, а союз ‘that’ с субъектом придаточного сегмента. В тех случаях, когда придаточный сегмент занимает позицию перед главным, используется $\{CO^-\}$ для субъекта главного сегмента и CO^+ для придаточного союза. Для определения правой границы вложенного сегмента служит факультативный коннектор $\{Xc^+\}$ в формуле подчинительного союза и Xc^- у запятой (‘,:Xc^-‘). На рис.2 продемонстрирован результат анализа сложного предложения, содержащего придаточный сегмент с союзом ‘that’ и предложный сегмент, где ‘after: (CL^+ or J^+) & ($\{Xc^+\}$ & CO^+)’ выступает одновременно в роли союза и предлога:

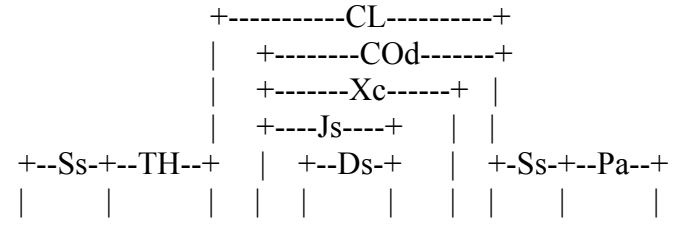


рис. 2

Для вложенных относительных придаточных, где возможна ситуация опущения союза (‘who’, ‘which’, etc.), используется коннектор $\{B^+\}$ для существительных (потенциальный субъект главного сегмента) и B^- в формуле транзитивных (переходных) глаголов (потенциальный предикат относительного придаточного). На рис. 3 показан результат анализа вложенного в главный сегмент относительного придаточного:

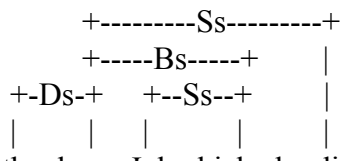


рис. 3

В системе LinkParser существует постпроцессор, предназначенный для работы с уже построенными альтернативными структурами предложения. Основная концепция постпроцессора заключается в разделении структуры на домены (domains) по одному или нескольким определенным типам связи.

Доменами (областями определения) называются полученные в результате деления независимые фрагменты предложения. Принципы деления на домены, как правило, определяются для каждого отдельного типа связи. В большинстве случаев используются сегментные связи (CL, CO, B, etc.) для нахождения доменов. Так, в предложении ‘John thinks there might be a problem’ выделяется два домена, соответствующие делению сложного предложения на простые сегменты: ‘John thinks’ и ‘there might be a problem’. Группой называется все множество связей определенных в пределах одного домена. На группах определены множества правил типа: группа, которой принадлежит связь X, должна содержать либо связь Y, либо Z. Основная цель такого постпроцессора - создать дополнительные ограничения (фильтр, реализующийся в системе как правила группы домена) на уже построенные синтаксические варианты, отвечающие общим структурным ограничениям.

Алгоритм синтаксического анализа в процессоре LinkParser основан на методе динамического программирования [D. Grinberg, J. Lafferty, 1995], т.е. в ходе анализа предложения все множество синтаксических единиц, входящих в предложение S, разбивается на перекрывающиеся подмножества (подзадачи) с сохранением исходного линейного порядка, где каждое такое подмножество является (в случае успешного построения связей между его элементами) поддеревом полного графа S и называется частичным решением (partial solution). Пусть S состоит из конечного множества упорядоченных словоформ $S = [W_1, W_2, \ldots, W_n]$, тогда процедура синтаксического анализа P порождает для S некоторое первоначальное множество M пар регионов (regions), где $M = [([W_1 .. W_2] [W_2 .. W_n]), ([W_1 .. W_3] [W_3 .. W_n]), \ldots, ([W_1 .. W_i] [W_i .. W_n]), \ldots, ([W_1 .. W_{n-1}] [W_{n-1} .. W_n])]$. Регионом называется отрезок предложения $S' = [W_i .. W_j]$, где i и j – границы региона, а все словоформы, входящие в отрезок, включая и границы, - единицы региона. Рекурсивный вызов процедуры P для региона S’ порождает новое множество $M' = [([W_i .. W_{i+1}] [W_{i+1} .. W_j]), ([W_i .. W_{i+2}] [W_{i+2} .. W_j]), \ldots, ([W_i .. W_{j-1}] [W_{j-1} .. W_j])]$ и т.д. Рекурсивный вызов процедуры P для региона S’ определен только в том случае, если между W_i и W_j удалось установить связь X, т.е. дизъюнкт W_i содержит коннектор X^+, а дизъюнкт W_j X^-. Очевидно, что процедура P в процессе анализа может неоднократно вызываться для одного и то же региона R, для того, чтобы избежать повторного вычисления R, используется прием memoization[Т. Кормен и др., 2001, стр. 298]: когда в

процессе выполнения алгоритма регион встречается в первый раз, его решение заносится в хеш-таблицу, и в дальнейшем решение для этого региона берется непосредственно из таблицы. Каждая словоформа состоит из множества дизъюнктов $W_j = [D_1, D_2, \ldots, D_m]$, поэтому полное определение возможного региона выглядит как $S' = [W_{ik} .. W_{jl}]$, где i и j – индексы словоформ, а k и l – индексы их дизъюнктов. Так, в упрощенном виде, выглядит основной алгоритм построения структуры в LinkParser, благодаря ограничению на проективность, являющийся модификацией алгоритма оптимальной триангуляции выпуклого многоугольника [Т. Кормен и др., 2001, стр. 306]. Скорость рекурсивного алгоритма синтаксического анализа экспоненциально зависит от количества слов в предложении, но применение memoization позволяет решить задачу построения синтаксической структуры в Link Parser за время[2] $O(n^3)$, где n – количество слов в предложении. Число различных подзадач (как и в задаче оптимальной триангуляции многоугольника [Т. Кормен и др., 2001, стр. 309]) в основном процессе вычисления синтаксической структуры составляет $\Theta(n^2)$. На скорость выполнения алгоритма сильно влияет общее число дизъюнктов в последовательности словоформ. Например, из формулы существительного ‘time’ порождается 770 дизъюнктов.

Для ускорения работы алгоритма синтаксического анализа в LinkParser предложен ряд решений, в том числе и эмпирических. Перед началом анализа устанавливается фильтр, удаляющий все дизъюнкты, содержащие «непарные» коннекторы: если для некоторого коннектора X^- дизъюнкта D, принадлежащего словоформе W, слева в линейной последовательности S не найдено X^+, то D будет удален, аналогично для право-направленного коннектора X^+. Другой метод ускорения вводит эмпирическое ограничение на длину возможного соединения в зависимости от типа связи. Несмотря на применяемые методы оптимизации, тестирование системы показывает, что в большинстве случаев анализ сложных предложений, длина которых превышает 25-30 слов, приводит к комбинаторному взрыву, и результатом работы анализатора становится “панический” граф, как правило, случайный вариант синтаксической структуры, зачастую несвязанной.

[2] Уточнение относительно времени работы основного алгоритма Link Parser было внесено Сергеем Протасовым.

К сожалению, использование грамматики LinkParser для русского языка представляется невозможным по ряду причин:

- Основная идея грамматики - использование лево- и право-ветвящихся коннекторов – теряет свою силу для языка с относительно свободным направлением связей (особенно для глагольных групп).
- Если предположить, что каждое возможное направление связи можно маркировать отдельным типом коннектора, то в этом случае резко возрастет как число базовых коннекторов, так и число дизъюнктов словоформ, что негативно скажется на скорости работы процессора.
- Избыточность и значительно возрастающий объем словаря, которые возникают в силу морфологической развитости флективного языка: каждая морфологическая форма описывается отдельной формулой, где нижний индекс входящего в нее коннектора должен будет обеспечить процедуру согласования, что приведет к усложнению составления коннекторов и к увеличению их общего числа в грамматике.

Тем не менее, LinkParser по праву считается одним из самых элегантных и детально проработанных решений задачи синтаксического анализа английского языка, а лингвистическая прозрачность грамматики и программная реализация алгоритмов на языке С придают процессору полную завершенность.

VII. Сегментационный анализатор немецкого предложения (STP).

Немецкими учеными из Исследовательского Центра Искусственного Интеллекта в Saarbruecken в конце 90-х гг. был создан Поверхностный Текстовый Процессор (Shallow Text Processor) [G. Neumann, J. Piskorski, 2001]. STP, как и лингвистические процессоры русского языка, относится к классу модульных систем (vs. HPSG и Link Grammar), характеризующихся разделением на функционально независимые компоненты, каждый из которых соответствует одному из уровней лингвистического анализа. STP первоначально разрабатывался для немецкого языка, хотя сейчас предпринимаются попытки перенести технологию анализатора на материал английского и японского языков. Архитектура процессора делится на две составляющих: лексический уровень и сегментный уровень. Графематический анализ (разбиение предложения на слова, выделение знаков препинания,

аббревиатур, etc.), морфологический анализ (лемматизация входных словоформ), модуль снятия частеречной омонимии, выделение шаблонных групп (темпоральные группы в тексте (время и даты), организации, персоналии, географические имена) относятся к лексическому уровню системы. Сегментный уровень состоит из трех модулей: сегментация предложения, построение именных (NP) и предложных (PP) групп внутри сегментов, установление грамматических функций (поиск комплементов глагола с использованием глагольной модели управления).

Принципиальным решением в процессоре называется отказ от традиционного анализа «снизу вверх» и применение принципа «разделяй и властвуй» [Т. Кормен и др., 2001, стр. 26] для вычисления синтаксической структуры предложения. В первой фазе синтаксического анализа определяется топологическая структура (topological structure) предложения (т.е. выделение глагольных групп (VP) и сегментов), во второй фазе происходит выделение фразовых категорий в пределах, определенных границами сегментов. Таким образом, в первой фазе анализ предложения проводится «сверху вниз», во второй – «снизу вверх», но на фрагментах меньше длины предложения. Нужно сказать, что идея необходимости разделения сегментационного и непосредственно синтаксического (в смысле установление связей между отдельными словами) анализа – параллельное построение сверху и снизу структуры предложения – существовала в московской прикладной лингвистике еще в 70-ые годы. Такая стратегия позволяет значительно снизить стоимость вычислений. Процессор не ставит своей целью построения полного графа предложения, поэтому результатом синтаксического разбора является частично связанное дерево зависимостей. При таком подходе сегментационный анализ становится центральным идеологическим компонентом архитектуры системы.

Большая часть алгоритмов в системе, включая определение топологической структуры предложения, реализована на машинах конечных состояний (FSM - finite-state machine) [Т. Кормен и др., 2001, стр.788], что значительно повышает скорость вычислений и эффективность распределения памяти. Большая часть грамматик фразовых категорий (NP и PP) представлена на регулярных выражениях [Дж. Фридл, 2001], преобразующихся в конечные автоматы. В STP используется два основных типа FSM: простые (FST - finite-state transducer) и весовые (WFST) трансдюсеры. FST называется автомат, в

котором каждый переход между состояниями в сети имеет выходную помету в дополнение к входной [XRCE MLTT, 1995]. Например, FST служит для представления контекстных правил в модуле снятия частеречной омонимии. WFST позволяет присваивать веса своим переходам и состояниям в сети.

Топологической структурой предложения называется разбиение всего предложения, равно как и его отдельных сегментов, на определенные зоны-поля (fields), границы которых определены глагольной группой и подчинительным союзом (союзным словом) в случае придаточного. Для топологической структуры немецкого языка использовано свойство аналитической глагольной группы, которая может быть разделена на две части (“haette ueberredet werden muessen”: “haette” и “ueberredet werden muessen”): левую (LVP) и правую (RVP). Как следствие такого деления, возникают поля структуры: фронтальное (FF), LVP, среднее (MF), RVP и остаток (RF). Для простого предложения “Er haette gestern ueberredet werden muessen” (“Он должен бы был быть предупрежден вчера”) структура выглядит следующим образом:

FF	LVP	MF	RVP	RF
Er	Haette	gestern	ueberredet werden muessen	ПУСТОЕ

То же верно и для придаточного предложения только лишь с той разницей, что LVP будет либо пустым, либо занято подчинительным союзом (союзным словом), а RVP займет полная (неразрывная) глагольная группа. Каждое отдельное поле может быть произвольно сложным. Структура вложенного в главное придаточного определительного предложения “Der Mann, der gestern haette ueberredet werden muessen, lief nach Hause” (“Человек, который должен бы был быть предупрежден вчера, уехал домой”):

FF	LVP	MF	RVP	RF
ПУСТОЕ	Der	gestern	haette ueberredet werden muessen	ПУСТОЕ

С помощью такого рода топологической структуры изящно обыгрывается принцип полноты (наличие предиката, союза и их окружение) всего предложения, а вместе с тем и его отдельных сегментов. Алгоритм, осуществляющий сегментацию, начинает с топологической структуры вложенных сегментов и, сворачивая подчинительные до уровня нетерминальных единиц, заканчивает структурой простого предложения в составе сложного. Алгоритм сегментации – рекурсивный и состоит из четырех ступеней анализа, каждой из которых соответствует своя стратегия (грамматика):

1. идентификация глагольных групп (VG);
2. идентификация базовых фрагментов (BC);
3. комбинирование последовательностей базовых фрагментов (CC) с целью формирования расширенных единиц (полных сегментов); если расширенная единица не идентифицирована, то перейти на шаг 4, иначе вернутся на шаг 2;
4. идентификация главных (простых) сегментов (MC);

VG выделяет как отдельные глаголы, так и цепочки глагольных форм в линейной последовательности предложения. Каждому глаголу приписывается его морфологическое значение, а в случае грамматической омонимии значения глагольных форм связаны оператором дизъюнкции.

BC членит исходное предложение по знакам пунктуации на отдельные фрагменты, присваивая каждому фрагменту свой тип, исходя из наличия/отсутствия подчинительного союза или глагольной формы. Фрагменты с подчинительными союзами называются базовыми.

CC анализирует рекурсивные вложения базовых фрагментов на основе их топологической структуры. Выделяются два возможных типа рекурсивных вложений: (а) среднего поля (MF-рекурсии), когда вложенный сегмент заключен между левой LVP и правой RVP частями глагольной группы подчиняющего сегмента; (б) остатка (RF-рекурсии), когда вложенный сегмент следует за RVP подчиняющего сегмента. Через операцию вложения подчинительных сегментов CC расширяет базовые фрагменты, рекурсивно доводя их до полных сегментов.

MC осуществляет анализ топологической структуры простого сегмента в составе сложного предложения и определяет сочинение сегментов.

Рассмотрим последовательность действий для сегментации предложения “Weil die Siemens GmbH, die vom Export lebt, Verluste erlitt, musste sie Aktien verkaufen” (“Потому что фирма Siemens GmbH, которая зависит от экспорта, понесла убытки, они вынуждены были продавать акции”):

Weil die [*company* Siemens GmbH], die … [Verb-Fin], V. [Verb-Fin], [Modv-Fin] sie A. [FV-Inf]	Шаг 1. (VG)
Weil die [*company* Siemens GmbH] [Rel-Cl], V. [Verb-Fin], [Modv-Fin] sie A. [FV-Inf]	Шаг 2. (BC)
[Subconj-CL], [Modv-Fin] sie A. [FV-Inf]	Шаг 3. (CC) MF-рекурсия и возврат

	на шаг 2.
[Subconj-CL], [Modv-Fin] sie A. [FV-Inf]	Шаг 3. (CC) (без изменений) переход на шаг 4
[clause]	Шаг 4 (MC)

Результат сегментации в скобочной записи: [*MAIN-CL* [*SUB-CL* Weil die [*company* Siemens GmbH] [*REL-CL* , die vom Export lebt], Verluste erlitt], musste sie Aktien verkaufen.]

После завершения работы сегментации проводится построение NP и PP групп внутри сегментов и установление грамматических функций на основе лексикона, где хранится около 12 тысяч глаголов с возможными моделями управления (subcategorization frames).

В описание процессора не включена информация о построении или разрешении синтаксической омонимии на уровне сегментов, т.е. возможность рассмотрения структурных вариантов сегментации предложения с разными границами сегментов. Также нет упоминания о сочинении предикатов – важной составляющей анализа для правильного определения границ сегментов.

Программная реализация системы первоначально выполнена на языке LISP, а затем переведена на C++. Тестирование STP немецкого языка демонстрирует высокую точность и скорость анализа.