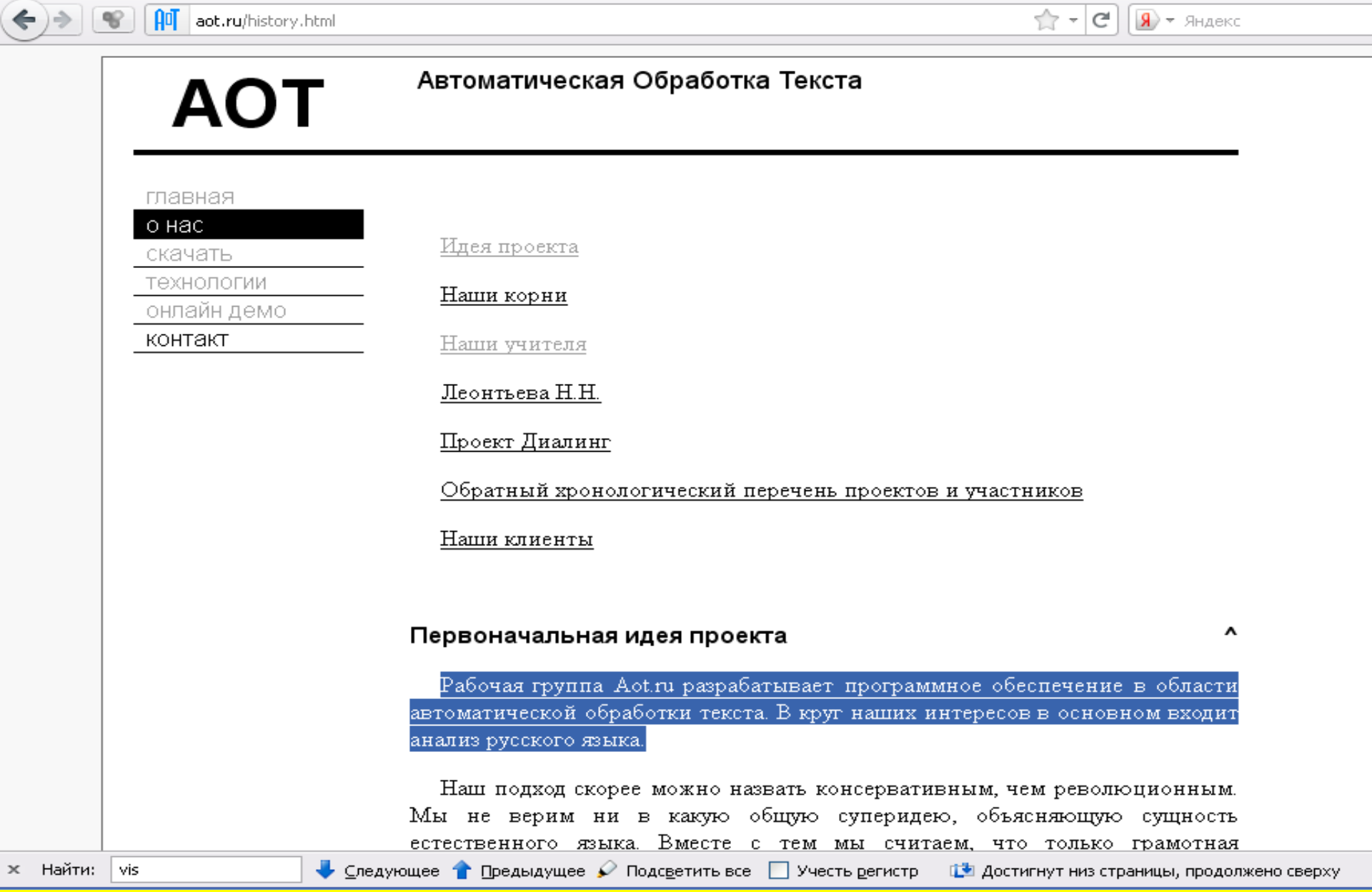
Проект АОТ (Автоматическая Обработка Текста, aot.ru) – рабочая группа и Интернет-сайт

Рабочая группа Aot.ru разрабатывает программное обеспечение в области автоматической обработки текста. В круг ее интересов в основном входит анализ русского языка.



Сайт aot.ru, раздел О НАС

Технологии АОТ базируются на многоуровневом представлении естественного языка, которое, в свою очередь, было заимствовано у системы ФРАП (Система французско-русского автоматического перевода была разработана коллективом лаборатории машинного перевода Всесоюзного центра переводов совместно с коллективом лаборатории машинного перевода МГПИИЯ им М. Тореза. 1976-1986 ГГ.)

Компоненты, составляющие языковую модель, - лингвистические процессоры, которые друг за другом обрабатывают входной текст. Вход одного процессора является выходом другого.

Выделяются следующие компоненты:

Графематический анализ. Выделение слов, цифровых комплексов, формул и т.д.

Морфологический анализ. Построение морфологической интерпретации слов входного текста

Синтаксический анализ. Построение дерева зависимостей всего предложения

Семантический анализ. Построение семантического графа текста

Для каждого уровня разрабатывался свой язык представления.

Язык представления, как полагается, состоит из *констант*

и

***правил их комбинирования*.**

На графематическом уровне константами были графематические дескрипторы (ЛЕ – лексема, ЦК – цифровой комплекс и т.д.)

На морфологическом уровне – граммемы (рд – родительный падеж, мн -множественное число).

На синтаксическом – названия отношений и групп (ПОДЛ – отношение между подлежащим и сказуемым, ПГ - предложная группа).

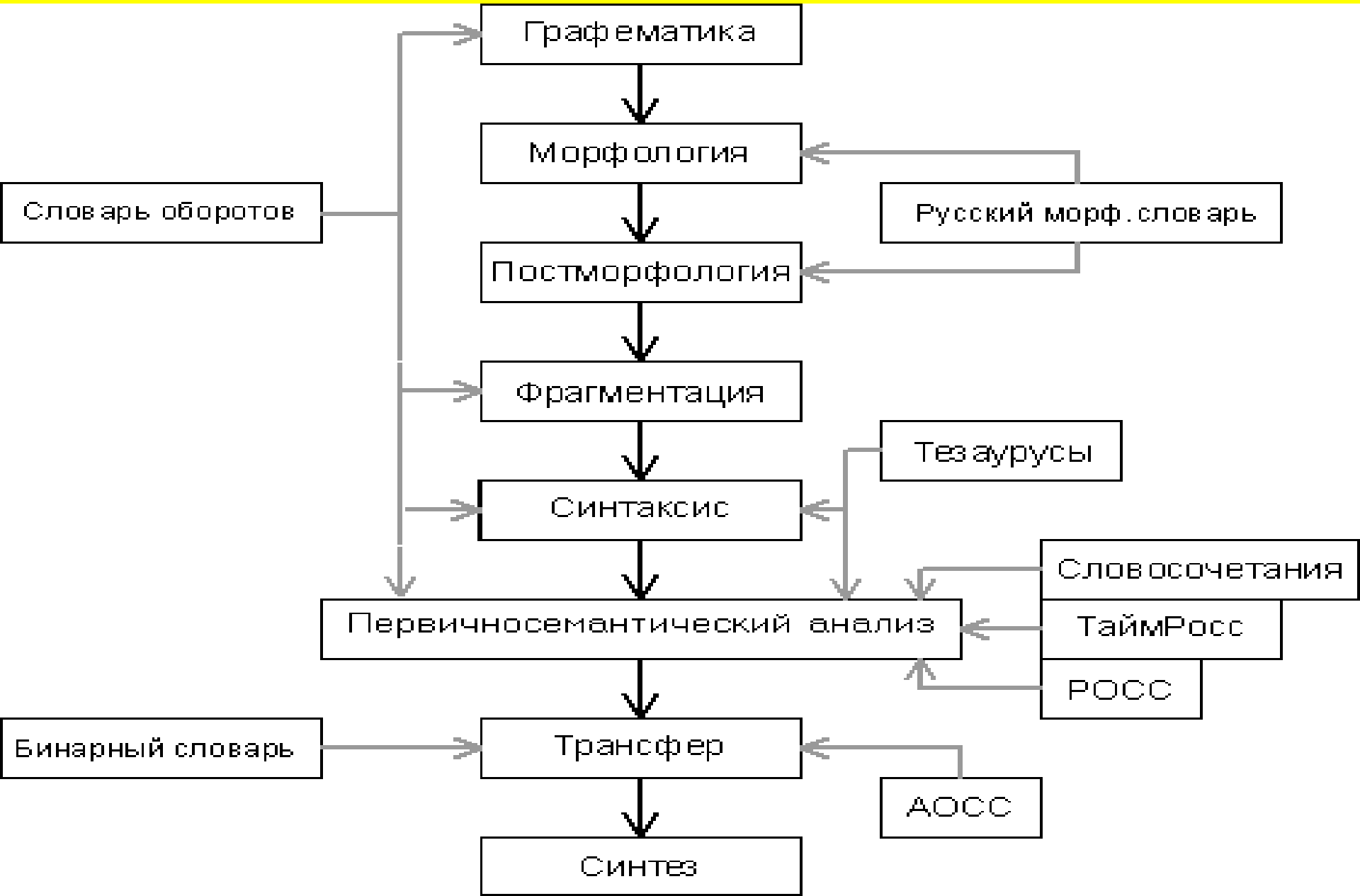
На семантическом – семантические категории и отношения.

С каждого уровня представления можно сделать переход к такому же представлению на другом естественном языке (трансфер), что позволяет осуществлять перевод, даже если «глубокий» (семантический) анализатор не смог обработать текст.

Основой для построения уровней служили результаты работы предыдущих этапов, однако последующие анализаторы также могли улучшить представление предыдущих.

Например, для какого-то предложения синтаксический анализатор не смог построить полного дерева зависимостей, тогда, возможно, семантический анализатор сможет спроектировать построенный им семантический граф (описание) на синтаксис.

Текст обрабатывается по следующей технологии



Графематика

Графематический анализ (ГрафАн) - это программа начального анализа естественного текста, представленного в виде цепочки ASCII символов, вырабатывающая информацию, необходимую для дальнейшей обработки Морфологическим и Синтаксическим процессорами.

ASCII (англ. *American Standard Code for Information Interchange*) — американская стандартная кодировочная таблица для печатных символов и некоторых специальных кодов.

В задачу графематического анализа входят:

- Разделение входного текста на слова, разделители и т.д.
- Сборка слов, написанных в разрядку;
- Выделение устойчивых оборотов, не имеющих словоизменительных вариантов;
- Выделение ФИО (фамилия, имя, отчество), когда имя и отчество написаны инициалами;
- Выделение электронных адресов и имен файлов;
- Выделение предложений из входного текста;
- Выделение абзацев, заголовков, примечаний.

Входные и выходные данные

На вход графематике подается файл plain-текста в Windows-кодировке. На выходе графематика строит таблицу, состоящую из двух столбцов.

- В первом столбце стоит некоторый кусок входного текста (выделенный по особым правилам).
- во втором столбце стоят графематические дескрипторы, характеризующие этот кусок текста.

Например, из текста «Иван спал» будет построена таблица из трех строк

Кусок входного текста	Графематические дескрипторы
Иван	RLE Aa NAM?
спал	RLE aa SENT_END

Графематические дескрипторы

Название	Русское название	Объяснение	Примеры
RLE	ЛЕ	русская лексема, присваивается последовательностям, состоящим из кириллицы	Иван
LLE	ИЛЕ	иностранная лексема, присваивается последовательностям из латиницы	John
DEL	РЗД	разделитель.	"*', '=', '_'
PUN	ЗПР	знак препинания, присваивается последовательностям, состоящим из одинаковых знаков препинания	".", '[', ']', '(', ')', '-', ':', ';'

Название	Русское название	Объяснение	Примеры
DC	ЦК	цифровой комплекс, присваивается последовательностям, состоящим из цифр	1234
DSC	ЦБК	цифро-буквенный комплекс, присваивается последовательностям, состоящим из цифр и букв	34h
GRAUNK		сложный узел, присваивается последовательностям, не обладающим вышеперечисленными признаками	

Разновидности дескриптора DEL

SPC	ПРБ	строка пробелов или табуляций
EOLN	КСТ	признак конца строки
PAR_SYM		символ параграфа
EMSYM		нулевой символ

Разновидности дескриптора PUN:

OPN	открывающая скобка	'{', '[', '('
CLS	закрывающая скобка	'}', ']', ')'
HYP	дефис	-

Разновидности дескриптора ЗПР и РЗД:

DPUN	последовательность одинаковых символов, длина которой больше 20
PLP	последовательность одинаковых символов, длина которой больше 1

Разновидности дескриптора ЛЕ и ИЛЕ:

aa	признак того, что все символы лексемы - малые	мам а
Aa	признак того, что первый символ лексемы - большой;	Мам а
AA	признак того, что все символы лексемы - большие	МА МА

Контекстные дескрипторы

BEG	ставится на начале текста (входного файла), т.е. всегда стоит на нулевой строке таблице. Причем, важно сказать, что нулевая строка таблицы используется как служебная (содержимое первого столбца нулевой строки не входит во входной текст)	
EOP	ставится на конце фразы.Концом фразы считается только ";".	
SENT_END	конец предложения	
NAM?	признак того, что лексема, возможно, является частью имени собственного. Присваивается лексеме, начинающейся с большой буквы и не имеющей перед собой символа конца предложения.	
BUL	ставится на начале пункта перечисления	
INDENT	ставится на начале абзаца	

EXPR1	ставится на начале оборота	типа "во взаимодействии с"
EXPR2	ставится на конце оборота	
FAM1	ставится на начале ФИО	типа "Иванов И.И."
FAM2	ставится на конце ФИО	
FILE1	начало имени файла	c:\test.txt
FILE1	конец имени файла	

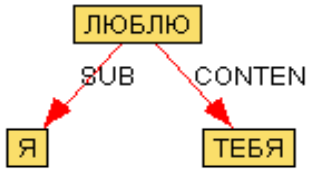
ABB1	начало сокращения	и т.п.
ABB2	конец сокращение	
KEY1	начало последовательности обозначений клавиш	Ctrl-F
KEY2	начало последовательности обозначений клавиш	
EA	электронный адрес	www.aot.ru sokirko@yandex.ru

Dialing Graph Builder

Введите одно предложение по-русски (не более 150 символов):

Я тебя люблю

Submit Request



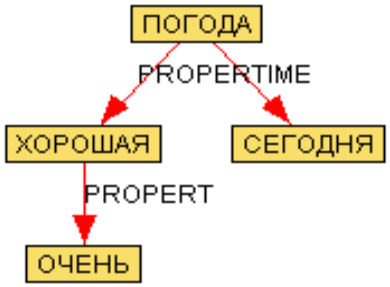
Построение семантического графа фразы Я ТЕБЯ ЛЮБЛЮ

Dialing Graph Builder

Введите одно предложение по-русски (не более 150 символов):

Сегодня очень хорошая погода

Submit Request



Построение семантического графа предложения СЕГОДНЯ ОЧЕНЬ ХОРОШАЯ ПОГОДА

Часто используются дескрипторы, относящиеся к **макросинтаксическому** анализу (анализу **расположения абзацев**, **заголовков**).

В **макросинтаксическом** анализе абзацы, заголовки и т.д. называются ***условно предложениями*** (УП). Макросинтаксические дескрипторы ставятся на конце УП в зависимости от типа УП.

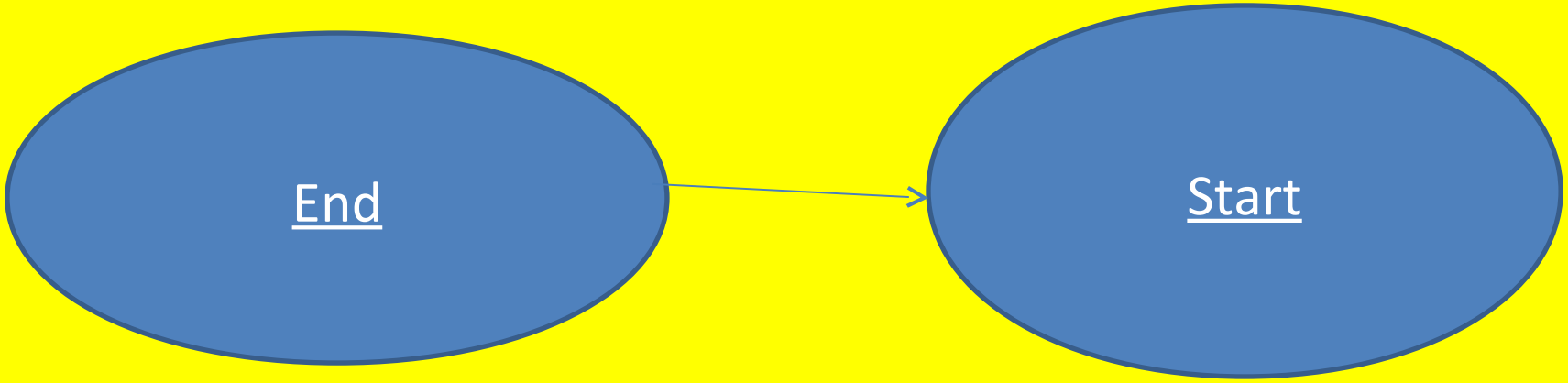
CS?	ставится на конце УП, тип которого не определен
CS	ставится на конце простого УП
HDNG	ставится на конце заголовка
CS_AUX	ставится на конце УП, заключенного в скобки
CS_PRNT	ставится на конце УП, заканчивающегося на двоеточие
DOC	ставится на нулевой строке графематической таблицы

Деление на предложения

Данный алгоритм включен в модуль графематики, так как на результаты его работы опирается макросинтаксический анализ.

На вход алгоритма подаются два числа StartPos и EndPos, которые обозначают первую и последнюю строки входного текста.

Программа ищет конец предложения, а потом после него ищет начало предложения.



Основные постулаты алгоритма

Начало текста совпадает с началом первого предложения, конец текста – с концом последнего.

Предложение всегда начинается с большой буквы.

Предложение не бывает больше одного абзаца.

Предложение не может состоять *только* из знаков препинания.

Определяется вспомогательный примитив IsSentenceEndMark (**обозначение конца предложения**).

На вход подается номер строки.

На выходе примитив возвращает истину, если эта строка содержит символ "?", "!", "." или многоточие.

Определяется вспомогательный примитив IsSentenceEndSeq (**обозначение последовательности конца предложения**).

На вход подается номер строки.

Примитив возвращает истину в двух следующих случаях:

- Если для этой строки верна функция IsSentenceEndMark и если контактно справа нет закрывающей кавычки (если предложение закавычено, закрывающая кавычка входит в это предложение);

- Если строка является закрывающей кавычкой, а контактно слева стоит строка, для которой верно IsSentenceEndMark.

Программу поиска предложений можно приблизительно описать следующим образом:

- Пусть i – текущая строка между StartPos и EndPos.

- Если на строке i стоит помета **начала абзаца**, тогда нужно пройти назад все пробелы и длинные разделители (PLP) и дойти до конца предыдущего абзаца.

Если в конце абзаца (до первого слова) стоит строка, которая удовлетворяет IsSentenceEndSeq, тогда нужно поставить SENT_END в этой строке, иначе нужно поставить SENT_END на конец предыдущего абзаца.

•Если на строке i стоит макросинтаксическая помета **УП**, тогда нужно сделать то же самое, что и в пункте 2, только надо учесть, что помета **УП ставится на конце абзаца**, а не начале (как в пункте 2).

•Если до начала текущего предложений стояла **открывающая скобка(кавычка)**, и текущая строка указывает на слово до соотв. закрывающей скобки(кавычки), тогда нужно поставить SENT_END на закрывающую скобку(кавычку), а i сместить на ближайшее после закрывающей кавычки (скобки) слово.

Если текущая строка удовлетворяет функции IsSentenceEndSeq и не стоит внутри графематических групп (FIO1-FIO2 и т.д.), тогда проходятся все знаки препинания от текущей строки.

Проверяется, что знак препинания, который заканчивает предложение, не должен стоять в самом начале строки. Далее ищется первое слово от текущей строки и считается началом нового предложения.