

Компьютерная лингвистика и интеллектуальные технологии:
по материалам международной конференции «Диалог 2017»

Москва, 31 мая — 3 июня 2017

РУССКОЯЗЫЧНАЯ ВОПРОСНО- ОТВЕТНАЯ СИСТЕМА КАК СИСТЕМА ФОРМАЛЬНОГО ВЫВОДА

Медовый В. С. (vmedoviy@mail.ru)

Медицинские компьютерные системы
(МЕКОС), Москва, Россия

В статье представлена демоверсия текстовой русскоязычной вопросно-ответной системы. Повествовательные предложения из заданной текстовой базы рассматриваются как набор аксиом, а текст вопроса — как формула, подлежащая доказательству последовательным выводом из аксиом с помощью некоторого набора правил вывода. Ответ на вопрос формируется при наличии и с использованием цепочки вывода. Предлагается конкретный набор правил вывода.

Ключевые слова: вопросно-ответная система, формальная система, правила вывода, процедура доказательства, порождающая грамматика

RUSSIAN LANGUAGE QUESTION ANSWERING SYSTEM AS A FORMAL SYSTEM

Medovyy V. S. (vmedoviy@mail.ru)

MEdical COmputer Systems (MECOS), Moscow, Russia

The paper presents a demo version of Russian language question answering system. Declarative sentences of a given text database are treated as a set of axioms. A question is treated as a formula to be proof by a procedure which deduces the question formula from the axioms by means of a set of inference rules. An answer is formed on a base of the proof chain. A specific set of inference rules is proposed.

Key words: question answering, formal system, inference rules, proof procedure, generative grammar

1. Постановка задачи

Не претендуя на оригинальность постановки и идеи решения (см. далее, п. 2), рассмотрим следующую задачу обработки текстов.

Имеется базовый русскоязычный текст, состоящий из повествовательных предложений. Считается, что базовый текст содержит некоторую истинную информацию на заданную тему.

Кроме того, имеется текст, содержащий повествовательные предложения с информацией общего характера, не привязанной к определенной теме. Назовем этот текст метабазой. При поступлении вопроса в виде текста вопросительного предложения надо автоматически сформировать реакцию на вопрос — текстовый ответ, адекватный в некотором смысле (см. ниже) поступившему вопросу, базовому тексту и метабазе.

Например, для базового текста с новостями на футбольную тему (взято из [1]):

- (B1) *Подопечные Слуцкого обыграли мадридский «Атлетико».*
- (B2) *Уже на второй минуте вратарь достал мяч из ворот.*
- (B3) *Мюнхенская «Бавария» обыграла мадридский «Реал» в первом полуфинальном матче футбольной Лиги чемпионов.*

и текста метабазы (также из [1]):

- (M1) *Подопечные Слуцкого — команда ЦСКА.*
- (M2) *Мадридский Атлетико — чемпион Испании.*
- (M3) *Если вратарь достал мяч из ворот, то был забит гол.*
- (M4) *Если NN победил, то NN нанес поражение.*
- (M5) *Если NN1 победил NN2, то NN2 не победил NN1.*

адекватными ответами на вопросы:

- (Q1) *Какая команда нанесла поражение чемпиону Испании?*
- (Q2) *На какой минуте забили гол?*
- (Q3) *Кого обыграла «Бавария»?*
- (Q4) *«Бавария» победила «Реал Мадрид»?*
- (Q5) *Победил ли мадридский «Реал» «Баварию»?*

являются соответственно тексты:

- (A1) *Подопечные Слуцкого ЦСКА*
- (A2) *второй*
- (A3) *мадридский Реал в первом полуфинальном матче футбольной Лиги чемпионов*
- (A4) *ДА*
- (A5) *НЕТ*

Определение адекватного ответа зависит от типа вопросительного предложения. Ограничимся вопросами, образованными на основе невопросительных предложений ([2], § 2594). Вопросы со специфическими синтаксическими образцами ([2], §2606) пока не рассматриваем.

- а) Вопросы типа да/нет — вопросы без вопросительных местоименных слов ([2], § 2595).

Это вопросы, текст которых **Q** преобразуется в повествовательное предложение **PQ** после удаления вопросительной интонации (знака вопроса) и вопросительных частиц (в вышеприведенных примерах это (Q4) и (Q5)). Кроме предложения **PQ**, построим еще и сопряженное с ним повествовательное предложение-отрицание **PQNOT**, добавив отрицательную частицу «не» к сказуемому **PQ**. Например, для вопроса (Q5) получим предложение

(PQNOT5) *Не победил мадридский «Реал» «Баварию».*

Адекватным ответом на вопрос **Q** будем считать текст «ДА», если преобразованный текст **PQ** логически вытекает (выводится) из текстов базы и метабазы, либо текст «НЕТ», если из этих текстов вытекает истинность предложения **PQNOT**.

- б) Вопросы с вопросительными местоименными словами ([2], § 2603) типа «какой», «что», «кто», «где», «когда» и т. п.

Адекватным ответом является такой текст, который, будучи подставлен в вопросительное предложение вместо вопросительного местоименного слова, образует повествовательное предложение, истинность которого логически выводится из текстов базы и метабазы.

Очевидно, что при таком определении может быть несколько адекватных ответов на вопрос.

При обосновании выводимости ответа на конкретный вопрос используются, вообще говоря, не все предложения базового текста и метабазы. Те предложения базы и метабазы, которые используются при логическом выводе ответа, назовем обоснованиями ответа и будем их считать неотъемлемой частью ответа. Так, для приведенных выше примеров ответы должны будут выглядеть примерно так:

ВОПРОС:

Какая команда нанесла поражение чемпиону Испании?

==ОТВЕТ N1

Подопечные Слуцкого ЦСКА.

====ОБОСНОВАНИЯ

Подопечные Слуцкого обыграли мадридский «Атлетико». Если NN победил, то NN нанес поражение. Подопечные Слуцкого — команда ЦСКА. Мадридский Атлетико — чемпион Испании.

ВОПРОС:

На какой минуте забили гол?

==ОТВЕТ N1

Второй.

=====ОБОСНОВАНИЯ

Уже на второй минуте вратарь достал мяч из ворот. Если вратарь достал мяч из ворот, то был забит гол.

ВОПРОС:

Кого обыграла «Бавария»?

==ОТВЕТ N1

Мадридский Реал в первом полуфинальном матче футбольной Лиги чемпионов.

=====ОБОСНОВАНИЯ

Мюнхенская «Бавария» обыграла мадридский «Реал» в первом полуфинальном матче футбольной Лиги чемпионов.

ВОПРОС:

«Бавария» победила «Реал Мадрид»?

==ОТВЕТ N1

ДА.

=====ОБОСНОВАНИЯ

Мюнхенская «Бавария» обыграла мадридский «Реал» в первом полуфинальном матче футбольной Лиги чемпионов.

ВОПРОС:

Мадридский «Реал» победил «Баварию»?

==ОТВЕТ N1

НЕТ.

=====ОБОСНОВАНИЯ

Мюнхенская «Бавария» обыграла мадридский «Реал» в первом полуфинальном матче футбольной Лиги чемпионов. Если NN1 победил NN2, то NN2 не победил NN1.

2. Вопросно-ответная система как формальная система

В соответствии с определением адекватного ответа для поиска ответа достаточно иметь процедуру проверки логической выводимости из базовых предложений некоторого текста, производного от текста предложения вопроса.

Действительно, для вопросов типа да/нет это следует непосредственно из определения.

Для вопросов с вопросительными местоименными словами преобразуем текст вопроса **Q** в повествовательное предложение **PQ**, убрав вопросительную интонацию и добавив «-то» к вопросительному местоименному слову («какой» → «какой-то», «что» → «что-то», и т. п.). Для приведенных выше примеров получим:

(PQ1) *Какая-то команда нанесла поражение чемпиону Испании.*

(PQ2) *На какой-то минуте забили гол.*

(PQ3) *Кого-то обыграла «Бавария».*

Если ответ на вопрос Q существует, то и предложение PQ , как утверждение о существовании некоторого объекта, логически следует из базового текста. Допустим обратное: каким-то образом доказана логическая выводимость PQ из базового текста как утверждение о существовании. Тогда по цепочке вывода можно найти и конкретный ответ.

Тем самым задача построения вопросно-ответной системы сводится к задаче проверки логической выводимости некоторого текста из набора заданных предложений.

Такая задача соответствует схеме построения *формальной системы* [3], если предложения базы и метабазы считать аксиомами, а производные от вопроса тексты PQ (и $PQNOT$ для вопросов типа да/нет) рассматривать как формулы, подлежащие доказательству (выводу из аксиом) с помощью последовательного применения некоторых заранее определенных *правил вывода*. Правило вывода является отношением, т. е. подмножеством множества упорядоченных наборов $\{(t_1, \dots, t_N, t')\}$ из $N+1$ текстов. В наборе (t_1, \dots, t_N, t') тексты t_1, \dots, t_N будем называть *посылками*, а текст t' — *непосредственным следствием*. Содержательно правила вывода определяются так, что если посылки выражают истинные в некотором смысле высказывания, то и непосредственное следствие также выражает истинное высказывание. При количестве посылок $N=1$ правило вывода называется *унарным*, при $N=2$ — *бинарным*.

Для двухместного отношения унарного правила вывода $DR = \{(t, t')\}$ можно построить *обратное* отношение, в которое включим все наборы с переставленными посылками и непосредственным следствием: $DR_{обр} = \{(t', t)\}$.

Если для унарного правила вывода обратное отношение также включено в множество правил вывода, то такое правило вывода будем называть *правилом перевода*. Содержательно это означает, что в правиле перевода посылка и непосредственное следствие выражают логически равносильные высказывания.

3. Правила вывода

Опишем конкретный набор правил вывода, который используется в демонстрационной версии русскоязычной вопросно-ответной системы **QUANDEMO** (см. ниже, п.5). Этот набор состоит из двух частей с условными названиями:

- 1) синтаксические правила вывода
- 2) реляционные правила вывода.

3.1. Синтаксические правила вывода

Все эти правила — унарные (с единственной посылкой), определяются отношением для упорядоченных наборов из двух (t, t') текстов, где t — посылка, t' — непосредственное следствие.

Общим условием для синтаксических правил является присутствие в тексте посылки пары открывающей и закрывающей скобок — специальных символов «[» и «]», т. е. текст посылки должен быть представлен в виде $t = \mathbf{a} + [+X+] + \mathbf{b}$, где \mathbf{a} , X , \mathbf{b} — некоторые тексты, а плюсом обозначена операция конкатенации текстов.

Общая схема построения синтаксического правила: пара (t, t') принадлежит отношению правила, если текст t' можно представить в виде $a+f(X)+b$, где a, X, b представляют посылку t , $a f(X)$ является результатом преобразования текста X в соответствии с некоторым шаблоном, задающим специфику синтаксического правила (см. ниже).

Заметим, что в тексте $Y=f(X)$, в свою очередь, могут присутствовать парные скобки $[\]$. Если такие скобки отсутствуют, то правило вывода назовем терминальным.

Все преобразования текста $Y=f(X)$, используемые в синтаксических правилах, относятся к одному и тому же классу. При определении этого класса предполагается, что:

- 1) текст состоит из *терминальных* символов — символов русскоязычной раскладки клавиатуры (букв, цифр, знаков препинания и т. д.), за исключением служебных символов: пар скобок $\{ \}$ и $[\]$;
- 2) задан набор *нетерминальных* символов $\{NT_i\}$; для их обозначений используем произвольные тексты, заключенные в фигурные скобки, например, $\{\text{существительное}\}$;
- 3) для любого нетерминального символа NT , любого текста t и целочисленных $Pstart \geq 1$, $Len \geq 0$ определена *контекстная характеристическая функция* $H(NT, t, Pstart, Len)$, принимающая значения *false* либо *true*.

Тем самым для нетерминального символа NT , текста t и позиции $Pstart$ определено множество отрезков текста, начинающихся с позиции $Pstart$ текста t , для которых значение характеристической функции равно *true* при некотором $Len > 0$. Обозначим это множество $Segm(NT, t, Pstart)$.

Допустим, имеется цепочка терминальных и/или нетерминальных символов $Z = z_1 \dots z_k$, а текст t можно разбить на k непересекающихся отрезков t_1, \dots, t_k , начинающихся в позициях текста t со значениями P_1, \dots, P_k , $P_1 < P_2 < \dots < P_k$. Если для всех $i = 1, \dots, k$ в случае терминального символа z_i $t_i = z_i$, а для нетерминального символа z_i t_i принадлежит множеству $Segm(z_i, t, P_i)$, то упорядоченное множество отрезков t_1, \dots, t_k назовем *разбиением* текста t в соответствии с цепочкой Z , а сами отрезки — *составляющими разбиения*.

Будем задавать преобразование текста следующими параметрами:

- а) условие применения преобразования: цепочка произвольных символов $ZCond$ и признаком ДА/НЕТ; может отсутствовать;
- б) левая часть схемы преобразования $Zleft$ — цепочка произвольных символов, причем если в $Zleft$ есть повторяющиеся нетерминальные символы, то они дополнительно индексируются;
- в) правая часть схемы преобразования $ZRight$ — цепочка произвольных символов, причем все нетерминальные символы цепочки $ZRight$ должны присутствовать в цепочке $Zleft$.

Эти параметры записываются в виде **шаблона** преобразования:

Zleft=>ZRight при отсутствии условия,

IF: ZCond :ENDIF Zleft=>ZRight при наличии условия **ZCond** с признаком ДА,

IFNOT: ZCond :ENDIFNOT Zleft=>ZRight при условии **ZCond** с признаком НЕТ.

Преобразование с заданным шаблоном по определению применимо к тексту **X**, имеющему разбиение t_1, \dots, t_k в соответствии с цепочкой **Zleft** и удовлетворяющему условию преобразования, а именно, при условии типа ДА, текст должен иметь хотя бы одно разбиение в соответствии с **ZCond**, а при условии типа НЕТ такие разбиения должны отсутствовать.

Имея разбиение t_1, \dots, t_k , результат преобразования $Y=f(X)$ строим по цепочке **ZRight** путем замены каждого входящего в цепочку **ZRight** нетерминального символа на составляющую t_i , соответствующую этому же нетерминальному символу в **ZLeft**.

При наличии в правой части схемы специального терминального символа **#** этот символ заменяется на уникальный идентификатор. Этот идентификатор состоит из префикса и целочисленного глобального счетчика. Префикс «S» устанавливается для исходных текстов базы, префикс «M» — для метабазы и префикс «Q» — для исходных текстов вопросов.

Содержательно описанные преобразования текста можно определить как выделение в исходном тексте некоторых составляющих и формирование из них комбинации путем перестановки, дублирования и вставки текстовых констант.

Поясним на примере: рассмотрим синтаксическое правило вывода, заданное шаблоном без условия применимости:

{ЭлемГрПодлежN1Mн} {ГруппаСказуемоеVfMн}.

=>

[XnMн# есть {ЭлемГрПодлежN1Mн}.][XnMн#

{ГруппаСказуемоеVfMн}.]

где *{ЭлемГрПодлежN1Mн}* и *{ГруппаСказуемоеVfMн}* —

нетерминальные символы.

Применим это правило к тексту [*Подопечные Слуцкого обыграли мадридский «Атлетико».*]

Находим разбиение текста в скобках [] (об алгоритме поиска разбиения см. ниже) в соответствии с левой частью шаблона «*{ЭлемГрПодлежN1Mн} {ГруппаСказуемоеVfMн}.*»:

t_1 = *Подопечные Слуцкого* (соответствует нетерминалу *{ЭлемГрПодлежN1Mн}*),

t_2 = ' ' (пробел),

t_3 = *обыграли мадридский «Атлетико»* (соответствует нетерминалу *{ГруппаСказуемоеVfMн}*),

t_4 = '.' (точка).

Используя правую часть шаблона, как результат получим текст:

[XnMнS1_ есть Подопечные Слуцкого.][XnMнS1_ обыграли

мадридский Атлетико.]

Здесь символ # в правой части шаблона порождает уникальный идентификатор «S1».

Набор шаблонов синтаксических правил вывода, применяемых в демоверсии, можно посмотреть по ссылке <https://cloud.mail.ru/public/FXEs/HuC7RxuFU>, файл Rules_Deduct.

В этом файле синтаксические правила вывода разбиты на группы с целью введения дополнительной априорной информации, сокращающей перебор правил, применимых к данному тексту. А именно, каждая группа правил имеет свой уникальный идентификатор. Запись группы состоит из списка шаблонов правил и/или ссылок на некоторые другие группы правил. В правой части шаблона правила при наличии парных скобок [] указывается идентификатор группы правил, применимых к тексту в этих парных скобках **после** применения преобразования в соответствии с заданным шаблоном.

Для реализации преобразования по заданному шаблону нужен алгоритм поиска всех разбиений заданного текста t в соответствии с заданной цепочкой символов $Z=z_1\dots z_k$.

В основе алгоритма лежит некоторая реализация контекстной характеристической функции $H(NT,t,Pstart,Len)$, использующая идеи порождающих грамматик Хомского [4].

А именно, будем считать, что нетерминальные символы относятся к одной из следующих категорий:

1) Символы **порожденных классов**.

для каждого из таких символов имеется группа порождающих правил вида $NT:=z_1\dots z_k$, где $z_1\dots z_k$ — произвольные (терминальные и/или нетерминальные) символы.

Эти правила записаны в специальном файле Rules_Grammar, который можно посмотреть по ссылке <https://cloud.mail.ru/public/91SN/HRGZdMwvL>.

2) **Словарные** символы — обозначения некоторых множеств цепочек терминальных символов, совпадающие с именами текстовых файлов, в которых эти цепочки собраны в алфавитном порядке, каждая цепочка в отдельной строке.

Словарные файлы содержат основы словоформ различных морфологических типов. Примеры словарных файлов (расширение .voc) можно посмотреть по ссылке <https://cloud.mail.ru/public/CSZK/ZLiBnNRtA>

3) **Функциональные** символы — некоторые специальные символы.

Тогда характеристическая функция строится следующим образом.

1) Для символа порожденного класса NT функция $H(NT,t,Pstart,Len)$ рекурсивно вызывает саму себя с нетерминалами из правой части порождающего правила.

2) Для словарного символа алгоритм характеристической функции несложен: соответствующий словарный файл просматривается на предмет поиска строки, входящей в текст t с позиции $Pstart$.

- 3) Для каждого функционального символа имеется своя специальная реализация характеристической функции.

Например, для функционального символа $NT=\{\$AnyStringAtEnd\}$ (произвольная цепочка символов до конца текста t) характеристическая функция $H(NT,t,Pstart,Len)$ возвращает **true**, если $(Pstart+Len-1)$ не превышает общей длины текста t .

Обратим внимание, что для некоторых функциональных символов характеристическая функция может принимать значение **true** при $Len=0$. Такие функциональные символы используются в порождающих правилах для проверки контекстной выполнимости некоторых условий.

3.2. Реляционные правила вывода

Эти правила применимы к текстам специального вида — *реляциям и формулам*.

Реляция имеет общие атрибуты, в частности, текст *сказуемого*, и несколько *кортежей*. Более чем один кортеж формируется для предложений с однородными членами предложения. Содержательно реляции выражают отношения между несколькими объектами, о которых идет речь в тексте.

Отдельный кортеж реляции с несколькими кортежами можно рассматривать и как самостоятельную реляцию с теми же атрибутами и этим единственным кортежем — *выборку из* исходной реляции.

Кортеж — это упорядоченный набор *элементов кортежа*.

Элемент кортежа — это набор некоторых атрибутов элемента и ссылка на уникальный идентификатор некоторой *переменной* или идентификатор некоторой другой, подчиненной реляции. Ссылки на идентификаторы могут отсутствовать.

Переменная характеризуется набором *свойств*.

Свойство — это текст свойства и набор атрибутов свойства, содержательно выражает одноместное отношение (принадлежность к некоторому подмножеству объектов).

Отрицанием данной реляции будем называть реляцию, полученную из данной путем добавления к сказуемому отрицательной частицы '**не**', если такой частицы в сказуемом нет, и исключения частицы '**не**', если таковая есть.

Формулы — это реляции, соединенные связками логических операций. Например, если $R1$ и $R2$ — реляции, то текст

ОПЕРАЦИЯ если то

R1

R2

является формулой.

Будем записывать реляции и формулы в двух формах: первичной, в формате XML и вторичной, в специальном реляционном формате REL.

Приведем примеры реляций.

1) Для исходного текста

Подопечные Слуцкого обыграли мадридский «Атлетико».

реляции в формате XML:

```
<СВ ИД="ХпМнS1 _ " ТИП="ПРИЛ" ВОС="$ADJ11">Подопечн|ые
</СВ>
<СВ ИД="ХопS2 _ " ПАДЕЖ="Род"/>
<СВ ИД="ХопS2 _ " ТИП="ПРИЛ" ВОС="$ADJ41">Слуцк|ого
</СВ>
<СВ ИД="ХпМнS1 _ " ПРИТЯЖ="ХопS2 _ "/>
<СВ ИД="ХдS3 _ " ПАДЕЖ="Вин"/>
<СВ ИД="ХдS3 _ " ТИП="ПРИЛ" ВОС="$ADJ41">мадридск|ий
</СВ>
<СВ ИД="ХдS3 _ " ТИП="СУЩ" ВОС="$NNOFLM" РОД="МУЖ" ЧСЛ="ЕД">Атлетико
</СВ>
<N1VF ЗАЛОГ="АКТ" ВОС="$VPast">обыгра|ли
<ПОДЛЕЖ ИД="ХпМнS1 _ " ТИП="ПРОСТ"/>
<ДОП ИД="ХдS3 _ " ТИП="ПРОСТ"/>
</N1VF>
```

Здесь представлены несколько свойств переменных (теги СВ) и двуместная реляция (тег N1VF) с единственным кортежем, выражающая отношение между переменными ХпМнS1 и ХдS3.

Те же самые свойства и реляции, записанные в специальном формате REL:

```
Ид перем=ХпМнS1 _
Падеж=
Род=
Число= Мн
Одуш=
Колич свойств=2
    Свойство=Подопечн|ые
        Словарь=$ADJ11
        ЧастьРечи=ПРИЛ
        Уровень=0
        ?ДобКмест=FALSE
    Свойство=Слуцк|ого
        Словарь=$ADJ41
        ЧастьРечи=ПРИЛ
        Уровень=1
        ?ДобКмест=FALSE
=====
Ид перем=ХдS3 _
Падеж=Вин
Род=МУЖ
Число=ЕД
Одуш=
Колич свойств=2
    Свойство=мадридск|ий
        Словарь=$ADJ41
```

```

ЧастьРечи=ПРИЛ
Уровень=0
?ДобКмест=FALSE
Свойство=Атлетико
Словарь=$NNOFLM
ЧастьРечи=СУЩ
Уровень=0
?ДобКмест=FALSE

```

=====

```

ОТНОШ Id=N1VFS1
сказуемое=обыгра|ли
словарь сказ=$VPast
залог=АКТ

```

==Число кортежей=1

```

Cortege N=1

```

===Число элементов=2

```

Вид эл=ПОДЛЕЖ

```

```

Мод эл=

```

```

Текст эл=

```

```

Падеж эл=

```

```

Перем эл=ХпМнS1 _

```

```

Отнош эл=

```

```

ГрОдн=

```

=====

```

Вид эл=ДОП

```

```

Мод эл=

```

```

Текст эл=

```

```

Падеж эл=Вин

```

```

Перем эл=ХдS3 _

```

```

Отнош эл=

```

```

ГрОдн=

```

Здесь идентификаторы переменных сведены в отдельный список и для каждой переменной приводится список свойств, которыми обладает эта переменная. Для самой реляции используется обозначение 'ОТНОШ'.

2) Приведем примеры реляции с двумя кортежами для текста с однородными членами предложения:

ЦСКА обыграл «Атлетико» и «Баварию».

В формате XML (тег N1VF):

```

СВ ИД="ХпЕдS1 _" ТИП="СУЩ" ВОС="$NNOFLM" РОД="МУЖ" ЧСЛ="ЕД">ЦСКА
</СВ>
<N1VF ЗАЛОГ="АКТ" ВОС="$VPast">обыгра|л
<ПОДЛЕЖ ИД="ХпЕдS1 _" ТИП="ПРОСТ"/>
<ДОП ТИП="ОДНОР" ГРОДН="ГрОднВинS2 _">
<СВ ИД="ХдS3 _" ПАДЕЖ="Вин"/>
<СВ ИД="ХдS3 _" ТИП="СУЩ" ВОС="$NNOFLM" РОД="МУЖ" ЧСЛ="ЕД">Атлетико
</СВ>
<ДОП ИД="ХдS3 _" ТИП="ПРОСТ"/>
<СВ ИД="ХдS4 _" ПАДЕЖ="Вин"/>
<СВ ИД="ХдS4 _" ТИП="СУЩ" ВОС="$NWFJAF2" РОД="ЖЕН" ЧСЛ="ЕД">Баварию

```

```
</СВ>  
<ДОП ИД="ХдS4 _ " ТИП="ПРОСТ"/>  
</ДОП>  
</N1VF>
```

В формате REL:

```
Ид перем=ХпЕдS1 _  
Падеж=  
Род=МУЖ  
Число=ЕД  
Одуш=  
Колич свойств=1  
Свойство=ЦСКА  
Словарь=$NNOFLM  
ЧастьРечи=СУЩ  
Уровень=0  
?ДобКмест=FALSE
```

=====

```
Ид перем=ХдS3 _  
Падеж=Вин  
Род=МУЖ  
Число=ЕД  
Одуш=  
Колич свойств=1  
Свойство=Атлетико  
Словарь=$NNOFLM  
ЧастьРечи=СУЩ  
Уровень=0  
?ДобКмест=FALSE
```

=====

```
Ид перем=ХдS4 _  
Падеж=Вин  
Род=ЖЕН  
Число=ЕД  
Одуш=  
Колич свойств=1  
Свойство=Бавари|ю  
Словарь=$NWFJAF2  
ЧастьРечи=СУЩ  
Уровень=0  
?ДобКмест=FALSE
```

=====

```
ОТНОШ Id=N1VFS1  
сказуемое=обыгра|л  
словарь сказ=$VPast  
залог=АКТ  
==Число кортежей=2  
Cortege N=1  
===Число элементов=2  
Вид эл=ПОДЛЕЖ  
Мод эл=
```

```

Текст эл=
Падеж эл=
Перем эл=ХпЕдS1 _
Отнош эл=
ГрОдн=
=====
Вид эл=ДОП
Мод эл=
Текст эл=
Падеж эл=Вин
Перем эл=ХдS3 _
Отнош эл=
ГрОдн=РдоднВинS2 _
=====
Cortege N=2
===Число элементов=2
Вид эл=ПОДЛЕЖ
Мод эл=
Текст эл=
Падеж эл=
Перем эл=ХпЕдS1 _
Отнош эл=
ГрОдн=
=====
Вид эл=ДОП
Мод эл=
Текст эл=
Падеж эл=Вин
Перем эл=ХдS4 _
Отнош эл=
ГрОдн=РдоднВинS2 _

```

Итак, будем считать, что определены тексты специального вида — реляции и формулы.

Перечислим реляционные правила вывода.

3.2.1. Правило *если то*

Правило бинарное, для применения нужны две посылки: реляция **R1** и формула с операцией «**если то**» над входящими в нее реляциями **R2** и **R3**.

Если реляция **R2** выводима из **R1**, то реляция **R3** по определению является непосредственным следствием.

3.2.2. Правило *N1-N1*

Правило бинарное, применяется к двум посылкам.

Первая посылка должен быть реляцией **R1** со специальным сказуемым «**есть**», в единственном кортеже которой есть элемент с атрибутами **ПОДЛЕЖ** и ссылкой на переменную **X1**. Такие реляции формируются для предложений с определениями типа

(M1) *Подопечные Слуцкого — команда ЦСКА.*

Вторая посылка — произвольная реляция **R2**.

Непосредственным следствием является реляция **R3**, полученная из реляции **R2** путем замены на переменную **X1** тех переменных из элементов кортежей **R2**, у которых список свойств непуст и полностью включен в список свойств переменной **X1**. Проверка включения текста в список выполняется с учетом синонимии: текст свойства либо совпадает с текстом свойства из списка, либо синонимичен ему. Информация о синонимии текстов свойств размещается в словарных файлах в виде дополнительных указаний в строках файлов.

3.2.3. Правило синонимии сказуемых

Правило унарное. Посылка — произвольная реляция **R** со значением атрибута **сказуемое Pred1**.

Непосредственным следствием является реляция, полученная из **R** путем замены значения атрибута **сказуемое Pred1** на любое другое значение **Pred2**, в некотором смысле синонимичное значению **Pred1**. Например, в вышеприведенном примере реляции с идентификатором **Id=N1VFS1** сказуемое **обыгра|л** можно заменить на **победи|л**.

Информация о синонимии сказуемых размещается в соответствующих словарных файлах в виде дополнительных указаний в строках файлов.

3.2.4. Правило выборки

Правило унарное. Посылка должна быть реляцией **R** с **N** кортежами, **N>1**.

Непосредственным следствием является реляция, полученная из **R** путем исключения из реляции любого кортежа.

3.2.5. Правило проекции

Правило унарное. Посылка должна быть реляцией **R** с единственным кортежем из **n** элементов, **n>1**.

Непосредственным следствием является реляция, полученная из **R** путем исключения из кортежа любого элемента.

3.2.6. Правило перестановки элементов

Правило унарное. Посылка должна быть реляцией **R** с единственным кортежем из **n** элементов, **n>1**.

Непосредственным следствием является реляция, полученная из **R** путем перестановки в кортеже любых двух элементов.

3.2.7. Правило смены залога *актив* → *пассив*

Правило унарное. Посылка должна быть реляцией **R** с единственным кортежем из **n** элементов, **n>1**. Атрибут реляции **залог** должен иметь значение **акт**.

В кортеже должны присутствовать элемент-подлежащее и элемент-дополнение, т. е. элементы со значениями атрибута **вид элемента** соответственно **ПОДЛЕЖ** и **ДОП**. Кроме того, элемент-дополнение должен быть без предлога (пустое значение атрибута **Текст эл**) и в винительном падеже (значение **ВИН** атрибута **Падеж эл**).

Непосредственным следствием является реляция, полученная из **R** путем смены значения атрибута **вид элемента** у элемента-подлежащего на **ДОП**, а у элемента-дополнения на **ПОДЛЕЖ**, то есть элементы подлежащего и дополнения меняются ролями.

3.2.8. Правило смены залога *пассив* → *актив*

Правило аналогично предыдущему, только атрибут реляции **залог** должен иметь значение **пас**, а элемент-дополнение должен быть в творительном падеже.

3.2.9. Правило попарного соответствия элементов кортежей

Правило унарное. Посылка должна быть реляцией **R1** с единственным кортежем из **n** элементов, $\{\mathbf{Elem1}(i)\}$, $i=1, \dots, n$.

Непосредственным следствием посылки **R1** в соответствии с данным правилом является реляция **R2**, также с единственным кортежем из **n** элементов, $\{\mathbf{Elem2}(i)\}$, $i=1, \dots, n$, для которой выполняются следующие условия:

- 1) значения атрибута **сказуемое** реляций **R1** и **R2** совпадают,
- 2) каждая пара элементов **Elem1(i)**, **Elem2(i)**, $i=1, \dots, n$, удовлетворяет специальному условию соответствия элементов кортежей, зависящему от сочетания атрибутов этих элементов.

Рассмотрим несколько примеров, в каких ситуациях условие соответствия считается выполненным. Пусть **Elem1** и **Elem2** — элементы кортежей.

- 1) Оба элемента представлены переменными (имеют непустые значения атрибута **Перем эл** — ссылки на переменную)

Тогда:

должны совпадать атрибуты вида элементов;

если переменная **Elem2** имеет свойство с вопросительным местоименным словом, то у элементов должны совпадать значения атрибута **падеж**;

если в обоих элементах атрибут **вид элемента** имеет значение «ДОПОЛНЕНИЕ», то должны совпадать и предлоги (из атрибута **текст элемента**);

если у переменных элементов значения атрибута одушевленности непусты, то эти значения должны совпадать,

все свойства переменной **Elem2**, за исключением свойства с вопросительным местоименным словом, должны входить в список свойств переменной **Elem1** с учетом возможной синонимии.

- 2) Оба элемента представлены подчиненными реляциями **r1** и **r2** соответственно (имеют непустые значения атрибута **Отнош эл**).

Тогда реляция **r2** должна быть выводима из **r1** путем последовательного применения реляционных правил вывода.

- 3) Оба элемента не имеют ссылок на переменные и подчиненные отношения.

Если у элемента **Elem2** нет свойства с вопросительным местоименным словом, то

должны совпадать или быть синонимичны атрибуты текста элемента. Кроме того,

должны совпадать значения атрибутов вида и моды элементов.

4. Алгоритм поиска ответа

Вернемся к постановке задачи:

- имеются базовый текст, текст метабазы и текст вопроса;
- заданы синтаксические и реляционные правила вывода.

Кроме того, примем следующее, достаточно сильное общее допущение, что заданные синтаксические правила являются правилами перевода (см. п. 2).

Допустим, имеется текст t и набор правил вывода $DR = \{DR_i\}$.

Назовем текст t_{end} *конечным следствием* текста t по набору DR ,

если t_{end} выводим из t путем последовательного применения правил набора DR ,

к тексту t_{end} неприменимо ни одно из правил набора DR .

Процесс построения ответа на вопрос разобьем на несколько шагов.

Шаг 1. Каждое предложение базового текста заключим в квадратные скобки и для полученного текста выполним построение всех конечных следствий по синтаксическим правилам вывода.

Синтаксические правила вывода подобраны таким образом, что каждое конечное следствие оказывается текстом, состоящим из свойств переменных, реляций и формул в формате XML. Тексты конечных следствий для всех предложений данной базы сохраняются в отдельном файле с расширением .xml. С помощью специального алгоритма преобразуем тексты конечных следствий в реляционный формат и сохраняем результаты в отдельном файле с расширением .rel.

Шаг 2. Аналогичен шагу 1 для предложений метабазы.

Шаг 3. Для полученных на шагах 1 и 2 текстов реляционных форматов выполняем построение всех конечных следствий по бинарным реляционным правилам вывода «если то» и «N1-N1».

Шаг 4. Исключение кортежей конечных следствий, противоречащих хотя бы одному кортежу реляций метабазы, полученных на шаге 2. Для этого рассматриваем попарно кортежи как выборки из соответствующих реляций и проверяем выводимость отрицания кортежа метабазы из кортежа конечного следствия. При проверке выводимости используются только унарные реляционные правила (см. п. 3.). Если такая выводимость имеет место, то кортеж конечного следствия исключается.

Шаг 5. Полученные конечные следствия сохраняем в отдельном файле с именем $\langle \dots \rangle Deduct.rel$.

В результате после выполнения шагов 1-5 для предложений базы и метабазы формируются 2 файла в формате XML и 3 файла реляционного формата.

Примеры этих файлов для вышеприведенной футбольной темы см. по ссылке <https://cloud.mail.ru/public/66ha/WLecqhUgq>

Шаг 6. Предложение вопроса Q преобразуем в повествовательную форму PQ , (см. п. 1), PQ заключим в квадратные скобки и для полученного текста $[PQ]$ выполняем построение всех конечных следствий по синтаксическим правилам вывода. Преобразуем тексты полученных конечных следствий в реляционный формат.

Шаг 7. Для каждой вопросительной реляции, полученной из предложения вопроса на шаге 6, выполняем процедуру проверки выводимости этой реляции из реляций базы и метабазы, сформированных на шагах 1 — 5 и сохраненных в файле `<...>Deduct.rel`. Для проверки выводимости на этом шаге используются только унарные реляционные правила вывода (см. п. 3.2).

Допустим, выводимость вопросительной реляции установлена. С учетом сделанного выше допущения, что синтаксические правила являются правилами перевода, тем самым установлена и выводимость текста PQ .

Если выводимость вопросительной реляции не установлена, то для вопроса типа да/нет формируем отрицание вопросительной реляции и повторяем процедуру проверки выводимости для отрицания вопросительной реляции.

Если установлена выводимость текста PQ , то выполняется просмотр цепочки вывода с целью получения списка обоснований ответа. Кроме того, для вопросов с вопросительными местоименными словами по цепочке вывода строится текст ответа.

Так как цепочек вывода может быть несколько (процедура проверки выводимости ищет все цепочки, насколько это возможно), то ответ может оказаться неединственным.

Если выводимость текста PQ (и $PQNOT$ для вопросов да/нет) не установлена, формируем реакцию системы «**ответ не найден**».

Шаг 8. Контроль списка ответов. В результате выполнения шага 7 получаем список ответов на входной вопрос. Если ответ не является единственным, пытаемся выявить ошибочные (неадекватные) ответы. В некоторых случаях это можно сделать только с помощью интерактивной процедуры с участием пользователя, в результате которой метабаза пополняется утверждениями об истинности/ложности некоторых высказываний, причем пополнение происходит сразу в реляционном формате. Ошибочные ответы исключаем. Кроме того, из двух ответов с одним и тем же обоснованием исключаем тот, текст которого является подстрокой другого ответа с точностью до перестановки слов-форм, т. е. оставляем более «полный» ответ.

Обратим внимание, что шаги 1–5 зависят только от базы и метабазы. Поэтому, сохранив результаты выполнения этих шагов в специальных файлах, при поступлении нового вопроса по той же самой теме поиск ответа можно начинать прямо с шага 6. Если в ходе контроля ответа на шаге 8 метабаза была пополнена, то поиск ответа для нового вопроса начинается с шага 4. Для пояснения приведем математический аналог: при решении задач используются, как правило, ранее доказанные утверждения — теоремы, а не исходные аксиомы математической теории.

5. Демоверсия вопросно-ответной системы

Алгоритм поиска ответов из п. 4 реализован в программном комплексе QUANDEMO, дистрибутив который можно свободно скачать по ссылке <https://cloud.mail.ru/public/4S9s/w8vKhjgPF>.

Программный комплекс предназначен для работы под управлением ОС MS Windows (любая версия).

Процедура инсталляции не предусмотрена. Для установки достаточно разархивировать дистрибутив в отдельную папку. В папке находится исполняемый файл **quan.exe**, который надо запустить любым доступным способом.

После запуска появляется окно диалога с пользователем, в котором надо выбрать базовую тему из предложенного списка и ввести текст вопроса в специальном поле.

Предварительно можно просмотреть текст базовой темы, нажав кнопку **Просмотр**.

Для поиска ответа надо нажать кнопку **Ответить**.

Если поле текста вопроса оставить пустым и нажать **Ответить**, то вопросы считаются введенными по умолчанию из специального списка.

Тексты тем по большей части взяты из школьных пособий по русскому языку.

Время формирования ответа при первом выборе темы может быть достаточно большим. При повторном выборе темы время реакции резко сокращается.

При работе программного комплекса в каталог размещения записываются вспомогательные файлы с расширениями `.xml`, `.rel`, а также файлы ответов с расширением `.ans`. Чтобы добавить новую базовую тему, надо записать в каталог размещения текстовый файл с расширением `.th`. Вопросы по умолчанию записываются в файл с тем же именем, что у файла базовой темы, но с расширением `.qu`.

Литература

1. *Igor Boguslavsky, Vyacheslav Dikonov, Leonid Iomdin, Alexander Lazursky, Victor Sizov, Svetlana Timoshenko. Semantic Analysis and Question Answering: a System Under Development. International Conference Dialogue-2015 Proceedings, Issue, Part 1*
2. Русская грамматика, Том 2 Синтаксис М.: Наука, 1980
3. *Клини С. К. Введение в метаматематику. — М.: ИЛ, 1957.*
4. *Гладкий А. В. Формальные грамматики и языки. — М.: Наука, 1973.*