## Постановка задачи

На вход ЕЯ-интерфейса поступает запрос. Будем рассматривать только запросы пользователя, представимые в виде

Фрагмент1 Сущ1 Предлог (возможно, пустой)   
Фрагмент2 Сущ2 Фрагмент3,

где Фрагмент 1 является либо пустой цепочкой, либо последовательностью прилагательных,

Сущ1 – существительное,

Фрагмент 2 является либо пустой цепочкой, либо последовательностью прилагательных,

Сущ2 – существительное,

Фрагмент3 является либо пустой цепочкой, либо выражением в кавычках (будем называть такие выражения искусственными именами).

Примеры входных запросов:

Запрос1 = "одноместные многоцелевые боевые самолеты российского производства",

Запрос2 = "экспериментальные летательные аппараты Китая",

Запрос3 = "широкофюзеляжные самолеты компании Airbus",

Запрос4 = " планета с самым большим радиусом",

Запрос5 = "частные аэропорты Германии".

Форму семантического представления (СП) входного запроса дает теория К-представлений В.А. Фомичева [5].

Возможным К-представлением Запроса3 является выражение

*самолёт(Тип-фюзеляжа,Широкофюзеляжный)(Производитель,Airbus).*

## Описания вспомогательных алгоритмов

## Описание функции *Dictionary-form*

**Вход:** Rm – морфологическое представление входного запроса Req, Rc - классифицирующее представление (КлП) запроса Req, p – целое – позиция какой-то словоформы в КлП Rc, т.е. 1≤ p ≤ lentext, где lentext = Длина(Т) - количество заполненных строк двумерного массива Rc.

**Значение:** лексема (базовая форма) слова в позиции p массива Rc, т.е. слова Rc[p, unit].

**Пример.** Пусть Req = Запрос6 = "недорогое вращающееся кресло для школьника", p = 5. Тогда Rc[p, unit] = «школьника», Dictionary-form(Rm, Rc, p) = «школьник».

## Алгоритм

**Описание функции *Input-line***

**Вход:** base - лексема (базовая форма) некоторого слова, Arls – двумерный массив – проекция лексико-смантического словаря Lsdic на входной запрос. **Значение:** целое – наименьший номер k такой строки масива Arls, что Arls[k, lec] = base, т.е. лексема base расположена в массиве Arls на пересечении строки с номером k и столбца с индексом lec.

## Алгоритм

**Описание функции *Modif-form***

**Вход:** строка вида R(z, b) или вида (H(z) = d), где R - бинарный реляционный символ, т.е. имя отношения с двумя атрибутами, и – имя функции с одним аргументом.

**Значение:** строка вида (R, b) в первом случае и строка вида (H, d) во втором случае.

**Пример.** *Modif-form(Вес(z, 3/тонна)) = (Вес, 3/тонна), Change- form(Целевое-место-использования (z, нек кухня)) = Modif-form((Целевое- место-использования, нек кухня))ю*

**Описание алгоритма *Construct-sem-image***

**Вход:** Rm – морфологическое представление входного запроса Req, Rc - классифицирующее представление (КлП) запроса Req, j – целое – позиция какого-то прилагательного в КлП Rc, m – целое – позиция некоторого прилагательного в КлП Rc, причем 1≤ j ≤ m, и если j < m, то в позициях от j до m расположены только прилагательные.

**Выход:** output – строка.

## Алгоритм

Начало

Цикл по k от j до m

Начало attribute := Dictionary-form(Rm, Rc, k) q := Input-line(Arls, attribute)

sem-item := Arls[q, sem]

Если k = j то Output := Modif-form(sem-item) Иначе Output := Output + Modif-form(sem-item)

{Здесь + - обозначение операции конкатенации (или сцепления) строк} кесли

конец

**Пример.** Пусть Запрос1 = "заднеприводные внедорожники", Запрос2 = "заднеприводные японские внедорожники". Используя алгоритм *Build- description* при обработке первого запроса, устанавливаем для j и m значение 1 и получаем значение выходной переменной Output = (Тип-привода, задний). Если же алгоритм *Construct-sem-image* используется при обработке второго запроса, то устанавливаем для j значение 1 и для m значение 2 и получаем значение выходной переменной Output = (Тип-привода, задний)(Страна- отношение, нек страна \* (Название, «Япония»)).

**Описание алгоритма *Discover-conc-relat***

**Вход:** Rm – морфологическое представление входного запроса Req, Rc - классифицирующее представление (КлП) запроса Req, position1 – целое – позиция первого существительного Сущ1, position2 – целое – позиция второго существительного Сущ2, prep – строка – предлог, относящийся к Сущ2 (возможно, пустой предлог nil).

**Выход:** semrel – строка – обозначение семантического отнощения, реализующегося в сочетании (Сущ1, prep, Сущ2); conc-noun1 – строка - обозначение cемантической единицы, ассоциированной с Сущ1 в рассматриваемом запросе; conc-noun2 – строка - обозначение cемантической единицы, ассоциированной с Сущ2 в рассматриваемом запросе.

## Алгоритм

base1:= Dictionary-form(Rm, Rc, position1)

{base1 - базовая форма (лексема) первого существительного} base2:= Dictionary-form(Rm, Rc, position2)

{base2 - базовая форма (лексема) второго существительного}

Пусть *narfrp –* количество строк в массиве *Arfrp.* Двумерный массив *Arfrp* строится по словарю предложных семантико-

синтаксических фреймов *Frp*. Массив *Arfrp.*содержит все шаблоны (или фреймы) из *Frp, «*привязанные» к предлогу prep.

Например, по запросу «НЕДОРОГОЙ КРЕМ ОТ КОМАРОВ» может быть построен следующий массив *Arfrp* с количеством строк *narfrp= 2:*

*(1, ‘от’, вещество, болезнь, 2, Против1, ‘таблетки от гриппа’); (2, ‘от’, вещество, дин.физ.об, 2, Против2, ‘мазь от комаров’);*

*(3, ‘от’, физическое явление, физ.об, 2, Эффект1, ‘тень от дома’). (4, ‘от’, изделие, модельер, 2, Дизайн, ‘сумка от Валентино’).*

В цикле по номеру *m* строки массива *Arfrp* (*m* изменяется от 1 до *narvfr*), где строка m является записью упорядоченного набора вида

***(m, prep, sort1, sort2, grc, rel, expl )*** , выполняются следующие действия:

Действие 1:

В цикле по строкам двумерного массива Arls (проекции лексико- семантического словаря *Lsdic* на входной запрос), состоящего из записей упорядоченных наборов вида

*(i, lec, pt, sem, st1,…,stk, commelentext)* ,

находится очередная строка с номером line1, для которой base1 = lec.

Затем проверяется следующее Условие 1: среди сортов*,* ***st1,…, stk*** найдется сорт, являющийся конкретизацией сорта ***sort1*** (в частности, такой сорт может совпадать с ***sort1***).

Действие 2: Аналогичное Условие 2 проверяется для base2 (т.е. для базовой формы второго существительного) во вложенном цикле по строкам двумерного массива Arls

Действие 3 (Выполняется в случае, когда Условие 1 истинно и Условие 2 истинно для некоторой строки массива Arls c номером line2):

Проверяется, может ли Сущ2 быть в грамматическом падеже с числовым кодом ***grc.***

Если ДА, то в выходную строку semrel рассматриваемого адгоритма добавляется обозначение смыслового отношения ***rel*** из рассматриваемой строки с номером *m* массива *Arfrp.*

Кроме того, выполняются присваивания

conc-noun1 := Arls[line1, sem], conc-noun2 := Arls[line2, sem],

**Пример 1.** По входному запросу пользователя рекомендательной системы НЕДОРОГОЙ КРЕМ ОТ КОМАРОВ может быть найдено смысловое отношение semrel = ПРОТИВ2.

**Пример 2.** По входному запросу пользователя другой рекомендательной системы ТЕМНО-КОРИЧНЕВЫЕ ТУФЛИ ОТ ГУЧЧИ может быть найдено смысловое отношение semrel = ДИЗАЙН.

## Описание головного модуля целевого алгоритма

**Вход:** T – входной запрос к РекС, Lingb – лингвистический базис

**Выход:** Semrepres – возможное К-представление входного запроса.

## Алгоритм SemParsing

Начало

Построить компонентно-морфологическое представление входного запроса, т.е. построить Rm – морфологическое представление входного запроса и Rc - классифицирующее представление (КлП) входного запроса

lentext := Длина (T) – количество элементарных значащих единиц входного текста (совпадает с количеством заполненных строк КлП Rc).

Построить двумерные массивы Arls, Arfrp – проекции на входной запрос Т соответственно лексико-семантического словаря Lsdic и словаря предложных сеантико-синтаксических фреймов Frp.

В цикде по k от 1 до lentext построить одномерный массив wordnouns, состоящий из существительных входного запроса, и сформировать значение целочисленной переменной numbwordnouns – количество существительных во входном запросе.

{Комментарий. Тогда wordnouns [1] – позиция первого существительного в запросе, wordnouns [2] - позиция второго существительного в запросе и т.д.} Если (numbwordnouns = 0) или (numbwordnouns > 2)

То Вывод («Неправильный запрос») Иначе

Если (numbwordnouns = 1) то вызвать алгоритм AttributesNounParsing Иначе вызвать алгоритм TwoWordnounsRequestsParsing

Кесли конец

## Описание основных подсистем целевого алгоритма Алгоритм AttributesNounParsing (Rm, Rc, wordnouns, Semrepres)

**Условие вызова:** numbwordnouns = 1 Начало

Position1 := wordnouns[1] Если position1 > 1

То Construct-sem-image(Rm, Rc, 1, position1-1, Characteristics1) Кесли

Base1 := Dictionary-form(Rm, Rc, position1) k1 := Input-line(Arls, base1) conc-noun1 := Arls[k1, sem]

Если position1 = 1

То description1 := conc-noun1

Иначе description1 := conc-noun1 \* Characteristics1

Кесли

Semrepres := ‘Запрос(S1, Качеств-состав’, + description1 + ‘)’

{Здесь + - обозначение операции конкатенации (или сцепления) строк} Конец

**Пример.** Пусть Запрос1 = "заднеприводные внедорожники", Запрос2 = "заднеприводные японские внедорожники". Тогда в результате обработки алгоритмом первого запроса переменная Semrepres получит значение Запрос(S1, Качеств-состав (S1, внедорожник \* (Тип-привода, задний))),

F в результате обработки алгоритмом второго запроса переменная Semrepres получит значение

Запрос(S1, Качеств-состав (S1, внедорожник \* (Тип-привода, задний) (Страна-отношение, нек страна1 \* (Название, «Япония»)))).

,

## Алгоритм NounOneNoulentextwoConnection(Rm, Rc, wordnouns, Semrepres)

**Условие вызова:** numbwordnouns = 2 Начало

position1 := wordnouns[1], position2 := wordnouns[2]

{Комментарий. Здесь position1 и position2 – позиции во входном запросе соответственно первого и второго сущетвительных}

posprep := 1 + position1

если Rc[posprep, tclass] ≠ предлог то prep := ‘nil’

иначе prep := Rc[posprep, unit]

{Комментарий. Таким образом, значением переменной prep является предлог, относящийся ко второму существительному. Если же такого предлога нет, то prep присваивается значение nil (пустой предлог)}. Find-sem-role(position1, position2, prep, role, conc-noun1, conc-noun2)

Если position1 = 1 {Т.е. нет прилагательных перед первым сущетвительным} То description1 := conc-noun1

Иначе

Construct-sem-image(Rm, Rc, position1 – 1, Characteristics1) Description1 := conc-noun1 \* Characteristics1

Кесли

Semrepres := ‘Запрос(S1, Качеств-состав’, + description1 + ‘)’

{Здесь + - обозначение операции конкатенации (или сцепления) строк} Base2 := Dictionary-form(Rm, Rc, position2)

**если** base2 - существительное собственное

**то** description2:= conc-noun2

## конец-если

{Пример: Для Запроса7 = "вместительный внедорожник из Японии" будет выполнен оператор присваивания

description2:= 'нек страна1 \* (Назв, "Япония")'} **если** base2 - существительное нарицательное **то** description2:= 'нек' + conc-noun2

## конец-если

Semrepres := Semrepres + (role, description2

{Пример 1: Для Запроса7 = "вместительный внедорожник из Японии" будет выполнен оператор присваивания

Semrepres := 'Запрос(S1, Качеств-состав(S1, внедорожник1 \* (Вместимость, высокая)(Страна-отношение, нек страна1 \* (Назв, "Япония")'}

{Пример 2: Для Запроса9 = "недорогие внедорожники фирмы "Тойота" будет выполнен оператор присваивания

Semrepres := 'Запрос(S1, Качеств-состав(S1, внедорожник1 \* (Цена, низкая)(Производитель, нек компания1'}

**если** после существительного в позиции position2 следует искусственное имя artif-name ("Опель", "Тойота" и т.д.)

**то** Semrepres := Semrepres + '\* (Назв,' + artif-name +')'}

## конец-если

{Пример: Для Запроса9 = "недорогие внедорожники фирмы "Тойота" будет выполнен оператор присваивания

Semrepres := 'Запрос(S1, Качеств-состав(S1, внедорожник1 \* (Цена, низкая) (Производитель, нек компания1 \* (Назв, "Тойота")'}

Semrepres := Semrepres + ')))'

{Закрыли три левые скобки}

{Пример: Для Запроса2 = "недорогие внедорожники фирмы "Тойота" будет выполнен оператор присваивания

Semrepres := 'Запрос(S1, Качеств-состав(S1, внедорожник1 \* (Цена, низкая)(Производитель, нек компания1 \* (Назв, "Тойота"))))'}

**конец**