

Московский государственный технический университет им.  
Баумана

Отчёт  
по лабораторной работе  
«Семантические сети»  
по курсу «Интеллектуальные системы»

Выполнил: Миляев Н.А.,  
Группа ИУ5-81.

Москва  
2010

# 1. Аннотация

В данной работе представлена программа, выполняющая функции анализа семантической сети, состоящей из сущностей и отношений между ними. В ходе анализа программа производит вывод новых отношений на основе отношений, заданных пользователем.

## 2. Используемые технологии

Программа написана на языке PHP и построена на основе Web-технологий. Для её работы требуется следующее ПО:

- На сервере — ОС Linux, Web-сервер Apache, PHP 5-й версии (возможна работа с другими ОС и Web-серверами, тестирование не проводилось).
- На клиенте — любой Web-браузер (на клиентской стороне используется только HTML).

Так как программа построена на основе Web-технологий, она может быть доступной через сеть Интернет.

## 3. Алгоритм решения задачи

В начале работы программа анализирует файл БЗ (путь к которому задан переменной \$file программы) и считывает данные БЗ во внутренние структуры (массивы сущностей, отношений и запросов пользователя). Если число полей в строке файла не соответствует ожидаемым, или задано неизвестное отношение, выдается сообщение об ошибке и переход на следующий этап не выполняется. Если в программе не заданы хотя бы две сущности и одно отношение, выдается сообщение об ошибке. Также сообщение об ошибке выдаётся, если сущность находится в отношении с самой собой.

Далее программа принимает и анализирует команды, полученные от пользователя, аналогичным образом. Если команда задана неправильно, выдается сообщение об ошибке. Если команда не содержит ни одного запроса, также выдается сообщение об ошибке.

Далее программа производит вывод новых отношений на семантической сети. Семантическая сеть поддерживает 4 типа отношений:

IS - «есть». A IS B — сущность A относится к классу сущностей B. Например, «кошка есть домашнее животное».

ISAPART - «есть часть». A ISAPART B — сущность A есть часть B, входит в B. Например, «голова есть часть туловища».

HAS - «частный признак». A HAS B — B является частным признаком A. Если A входит в какую-нибудь другую сущность C, признак B не передаётся C. Например, «волосатость есть признак головы, голова есть часть тела — отсюда не следует, что всё тело покрыто волосами».

PROP - «общее свойство». A PROP B — B является общим свойством A, которое переносится на все сущности, в которых входит A. Например, «голова покрыта кожей, голова есть часть тела, и тело также покрыто кожей».

В ходе вывода проверяются все пары отношений, находятся такие, из которых следует новое отношение, и вывод из пары отношений добавляется в массив отношений.

Действуют следующие правила вывода:

$A \text{ IS } B, B \text{ IS } C \rightarrow A \text{ IS } C$  (транзитивность отношения «есть»)

$A \text{ ISAPART } B, B \text{ ISAPART } C \rightarrow A \text{ ISAPART } C$  (транзитивность отношения «есть часть»).

$A \text{ IS } B, B \text{ HAS } C \rightarrow A \text{ HAS } C$  (переносимость признака класса на элемент этого класса).

$A \text{ IS } B, B \text{ PROP } C \rightarrow A \text{ PROP } C$  (переносимость свойства класса на элемент этого класса).

$A \text{ ISAPART } B, B \text{ PROP } C \rightarrow A \text{ PROP } C$  (переносимость свойства целого на части).

$A \text{ ISAPART } B, C \text{ IS } B \rightarrow A \text{ ISAPART } C$  (переносимость отношения «есть часть» на элемент класса).

Если в ходе вывода появляется отношение сущности с самой собой, выводится сообщение об ошибке.

Далее программа производит анализ запросов пользователя и выполняет их, выводя сущности, связанные с сущностями, указанными в запросе.

## 4. Описание языка БЗ и команд пользователя.

Для представления отношений между сущностями семантической сети и запросов пользователя в файле БЗ используется простой язык, который также применяется для задания команд пользователя (команда может содержать отношения, дополняющие семантическую сеть БЗ, а также запросы). Запросы могут входить и в файл БЗ.

Представление отношения:

<сущность 1> <тип отношения> <сущность 2>

Именами сущностей могут быть любые последовательности печатных символов, в которые не входят символы пробела, табуляции, новой строки и точки). Типом отношения может быть IS, ISAPART, PROP или HAS. Разделителем сущностей и отношений является пробел. Допустимо указывать несколько пробелов подряд.

Представление запроса замены: <новое имя сущности>=<имя существующей сущности>; заменяет вхождения одного имени сущности на другое.

Представление запроса сущности:

? <имя сущности>

В ходе выполнения запроса происходит вывод на экран сущностей, связанных с заданной.

Представление запроса отношения:

? <имя сущности 1>.<имя сущности 2>[.<имя сущности 3>.<имя сущности 4>...]

В ходе выполнения запроса происходит проверка отношений между сущностями, разделёнными точкой. Сущности, связанные с последней сущностью, выводятся на экран.

Строки, начинающиеся с #, игнорируются (считаются комментариями). Пустые строки также игнорируются.

## 5. Пример работы программы.

Пусть в файле БЗ хранится следующая информация:

MURKA IS CAT

#Мурка - кошка

CAT IS DOMESTIC

#Кошка — домашнее животное

DOMESTIC IS ANIMAL

#Домашние животные относятся к животным

ANIMAL PROP LIFE

#Животные обладают тем свойством, что они живые

ANIMAL HAS EYES

#У животных есть глаза

PAW ISAPART CAT

#Лапа есть часть кошки

CLAW ISAPART PAW

#Коготь есть часть лапы

Граф семантической сети представлен на рис. 1.

Если ввести запрос на сущность, например

? MURKA

можно узнать всю информацию о Мурке, которая была получена с использованием данных об отношениях и правил вывода:

MURKA

Сущность MURKA:

является сущностью класса CAT

является сущностью класса DOMESTIC

является сущностью класса ANIMAL

имеет общее свойство LIFE

имеет признак EYES

Число связей сущности MURKA составляет 5.

В состав сущности MURKA входят сущности:

PAW

CLAW

В переводе на естественный язык это будет означать следующее: Мурка — кошка, которая относится также к домашним животным и к животным вообще. Она живая, и у неё есть глаза, лапы и когти. Интересно заметить, что одинаковые по звучанию на естественном языке отношения в БЗ выражены по-разному (есть глаза и есть лапы). Это зависит от желания инженера по знаниям выразить то или иное отношение предметной области именно таким образом.

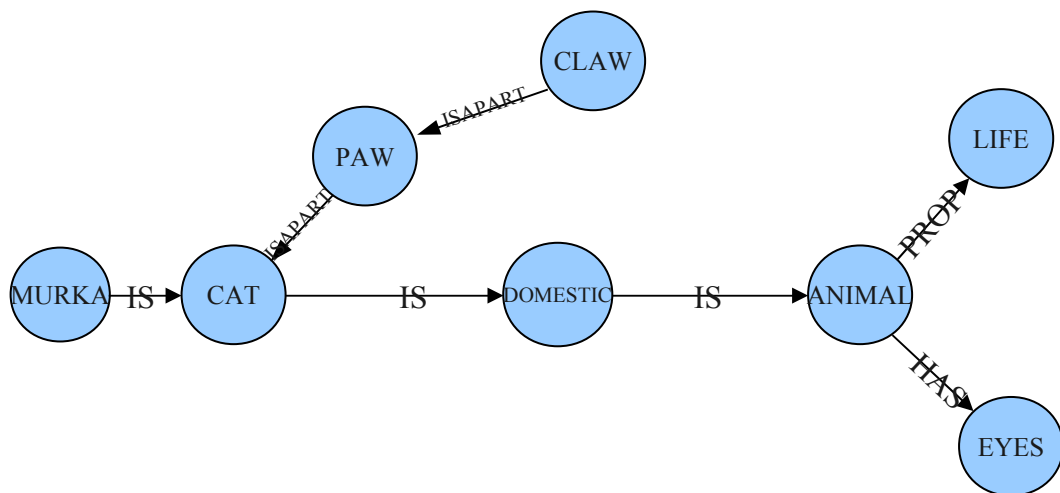


Рис. 1. Схема графа семантической сети.

Воспользуемся возможностью проверки связей между сущностями:

? CLAW.MURKA

#Есть ли когти у Мурки?

? CLAW.LIFE

#Является ли коготь живой материей?

? CLAW.EYES

#Есть ли у когтя глаза?

Получается следующий результат:

CLAW.MURKA

Сущность CLAW есть часть сущности MURKA.

Сущность MURKA:

...

CLAW.LIFE

Сущность LIFE есть общее свойство сущности CLAW.

Сущность LIFE:

LIFE не имеет связанных сущностей.

CLAW.EYES

Сущности EYES и CLAW не связаны.

На данном примере можно увидеть различие общих и частных признаков (свойств). Свойство жизни, присущее организму, можно считать присущим его частям, а наличие глаз — нет.

Отношение ISAPART проверяется в обе стороны: на запросы

? MURKA.CLAW

? CLAW.MURKA

программа выводит, что сущность CLAW есть часть сущности MURKA.

На запрос о связях нескольких сущностей программа проверяет каждую связь, например, на запрос:

? MURKA.CLAW.PAW.LIFE

#Является ли живой материей лапа, на которой находится коготь Мурки?

программа выведет:

MURKA.CLAW.PAW.LIFE

Сущность CLAW есть часть сущности MURKA.

Сущность CLAW есть часть сущности PAW.

Сущность LIFE есть общее свойство сущности PAW.

Сущность LIFE:

LIFE не имеет связанных сущностей.

Программа проверяет корректность задания отношений. Так как все отношения в сети антирефлексивны, для проверки корректности можно просмотреть список отношений после вывода, и поискать там отношения сущности с самой собой. Если такие отношения имеются, семантическая сеть построена некорректно. Пример:

A IS B

C ISAPART B

A ISAPART D

D ISAPART C

? A

Из первых двух отношений следует, что C ISAPART A. С учетом третьего отношения получаем C ISAPART D, с учетом последнего — C ISAPART C, что недопустимо. Программа выдает сообщение об ошибке «БЗ содержит кольцевые зависимости на сущности C!».

В командах пользователя допускается определение новых сущностей, отношений, а также подстановок, например:

```
DOG IS DOMESTIC
#собака - домашнее животное
DOG HAS BARK
#собака лает
PET=DOG
#мой питомец - собака
? PET.BARK
#лает ли мой питомец?
PET=CAT
#а если это кошка?
? PET.BARK
```

Вывод программы:

```
PET = DOG
Произведена подстановка DOG в PET.
PET.BARK
Сущность BARK есть признак сущности DOG.
Сущность BARK:
BARK не имеет связанных сущностей.
PET = CAT
Произведена подстановка CAT в PET.
PET.BARK
Сущности BARK и CAT не связаны.
```

В данном случае PET не является отдельной сущностью, а представляет собой альтернативное имя сущности, смысл которого может изменяться.

## 6. Выводы.

В ходе выполнения данной работы представлена программа, способная выводить новые отношения в семантической сети, пользуясь правилами вывода, и отвечать на запросы пользователя. Программа использует простой язык представления отношений и пользовательских запросов, предоставляет пользователю добавлять (но не удалять и изменять) отношения в заложенную в БЗ семантическую сеть, имеет возможность проверки отношений на корректность. Недостатком программы является фиксированное число и вид отношений (отсутствие возможности задания отношений и правил вывода непосредственно в БЗ).