Лабораторная работа №3

**Правила семантических сетей**

Для определения пользовательских правил логического вывода предназначены синтаксисы: SWRL, SPIN и SHACL. Правила SWRL исторически возникли первыми, наиболее наглядны, а создавать и отлаживать такие правила можно в редакторе Protege.

В Protégé 4 и 5 для выполнения правил SWRL используется машина логического вывода Pellet, напрямую работающая с содержимым онтологий.

Синтаксис правил SWRL довольно прост. Каждое правило состоит из двух частей – условия и вывода, который формируется, если условие выполнено. И условие, и вывод могут состоять из нескольких атомов – элементарных логических выражений. Условие выполняется, если истинны все составляющие его атомы; если условие выполняется, то «сработают» все атомы, составляющие вывод. Отрицание и дизъюнкция в атомах не поддерживаются.

Каждый атом представляет собой предикат – утверждение о каких-либо объектах онтологии. Синтаксис записи предикатов соответствует принятому в логике. Например, если известно, что Петр является сыном Ивана, это может быть записано как предикат: **являетсяСыном ( Петр, Иван)**. То, что указано в скобках – аргументы предиката. При составлении правил SWRL в качестве аргументов могут выступать переменные или конкретные объекты.

Предикаты могут представлять собой не только утверждения о связях; рассмотрим все варианты предикатов, которые могут использоваться в правилах SWRL.

1. Утверждения о принадлежности к классу. Записывается в следующей форме: Человек( Петр ). Аргументом является индивидуальный объект. Минимальное SWRL-правило может выглядеть так:

Мужчина ( ?x ) -> Человек ( ?x )

Это правило утверждает, что любой экземпляр класса **Мужчина** является и экземпляром класса **Человек**. Как мы видим, переменные в SWRL, как и в SPARQL, предваряются знаком вопроса.

1. Утверждения о существовании связи. Примером является выражение

являетсяСыном ( Петр, ?x )

Заметим, что оно будет иметь практический смысл в правиле только в том случае, если хотя бы один аргумент будет переменной.

3. Утверждения о значении свойства-литерала. При помощи способа, аналогичного приведенному в предыдущем пункте, можно получать в переменные значения свойств-литералов, и затем сравнивать их с различными

выражениями при помощи встроенных функций. Пример:

имеетВозраст ( ?x , ?y ) ∧ swrlb:greaterThanOrEqual ( ?y, 16 ).

Это условие проверяет, что объект **?x** имеет возраст, больший или равный 16. Как мы видим из этого примера, знак ∧ используется для перечисления нескольких условий – это и некоторые другие обозначения соответствуют общепринятой нотации для записи логических выражений. Заметим также, что для проверки равенства значения свойства какой-либо величине можно обойтись более коротким выражением: имеетИмя ( ?x, "Иван").

4. Условие раздельности двух индивидуальных объектов:

differentFrom ( ?x, ?y )

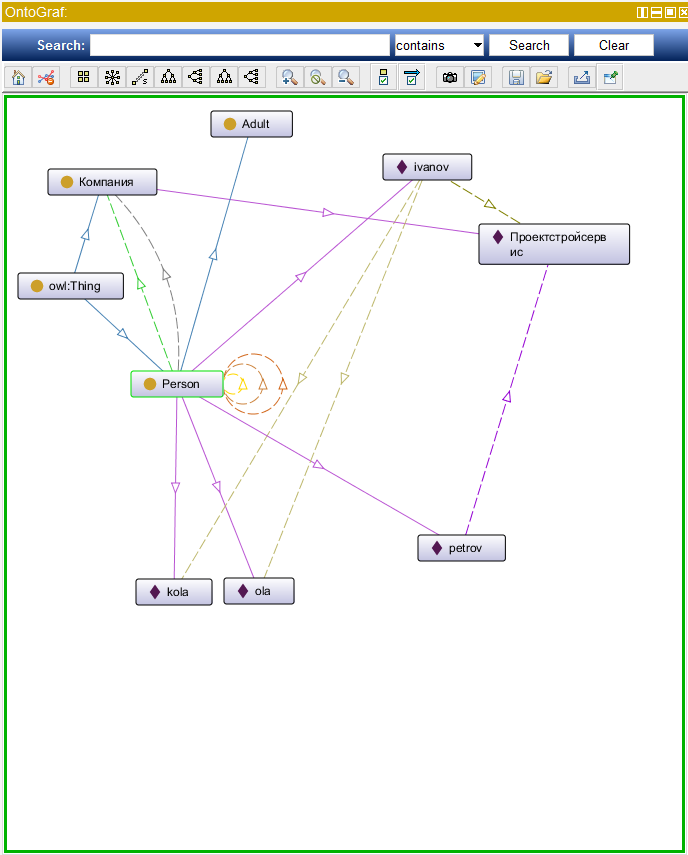
5. Условие совпадения двух объектов: sameAs ( ?x, ?y ). Условие будет истинно, если переменные ?x и ?y оказались равны одному и тому же объекту.

6. Условие о принадлежности значения переменной определенному типу данных. Например: xsd:int ( ?x )

7. Так называемые встроенные условия (built-ins), одно из которых мы уже видели в пункте 3: swrlb:greaterThanOrEqual. Имеется обширный набор таких функций, а также возможность определять собственные.

Результат формулируется при помощи атомов в такой же форме, как и условие.

Рассмотрим процесс создания и выполнения правил SWRL в редакторе Protégé на примере следующей онтологии.



Между классами Person и Компания в данной онтологии присутствует два типа связей:

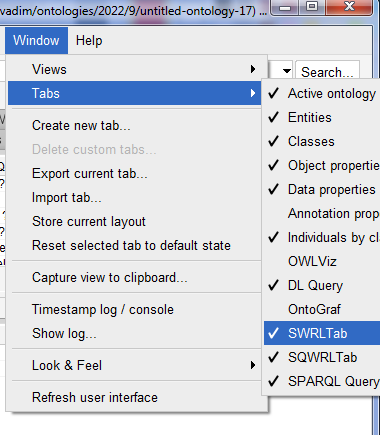
* директор (isDirector),
* работник (isWorker).

Объекты класса Person могут быть связаны друг с другом связями:

* “имеет ребенка”,
* “начальник”,
* “поздравляет с праздником“

Класс Person имеет подкласс Adult (взрослый).

Создаются правила на вкладке **SWRLTab:**



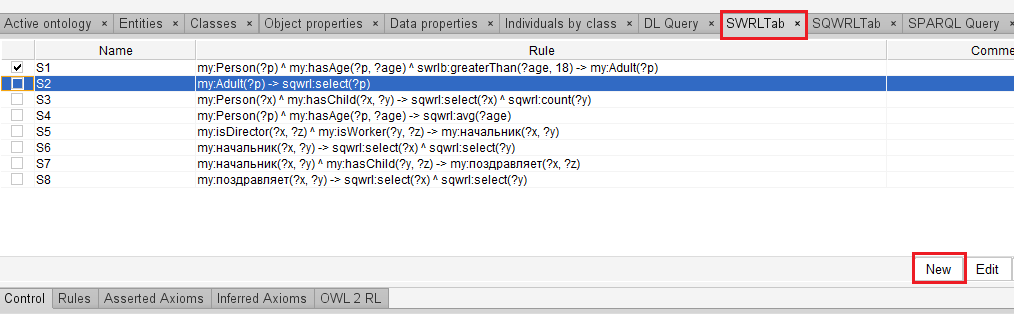
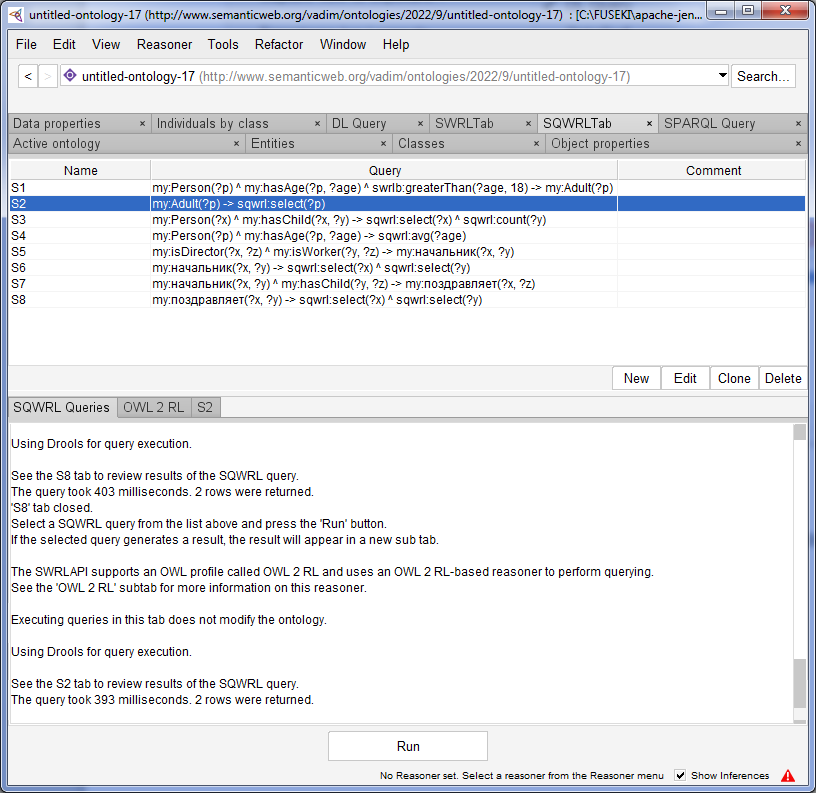


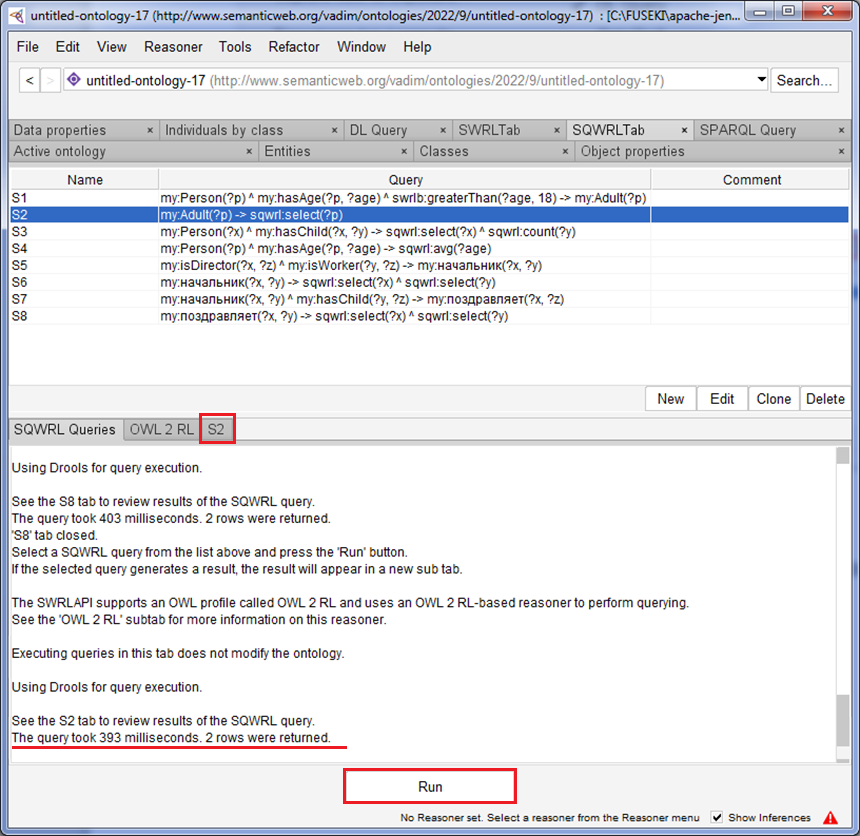
Рис. 1. Добавление новых правил

Допустим, хотим определить всех взрослых людей. Тогда выделяем галочкой правило S1 и переходим на вкладку SQWRL:

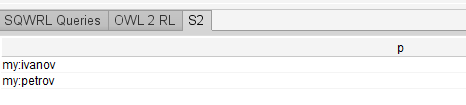


Здесь выделяем правило **S2**. Данное правило осуществляет вывод всех полученных в правиле **S1** экземпляров класса **Adult** (взрослый).

Для запуска вывода нажимаем кнопку **Run.**



Если запрос успешно выполняется, то его результат отображается на вкладке с именем, совпадающим с именем выполняемого правила.



Описание правил:

**S3**: для каждого человека определить количество детей.

**S4**: определение среднего возраста.

**S5, S7**: директора компаний поздравляют детей своих сотрудников с праздником.

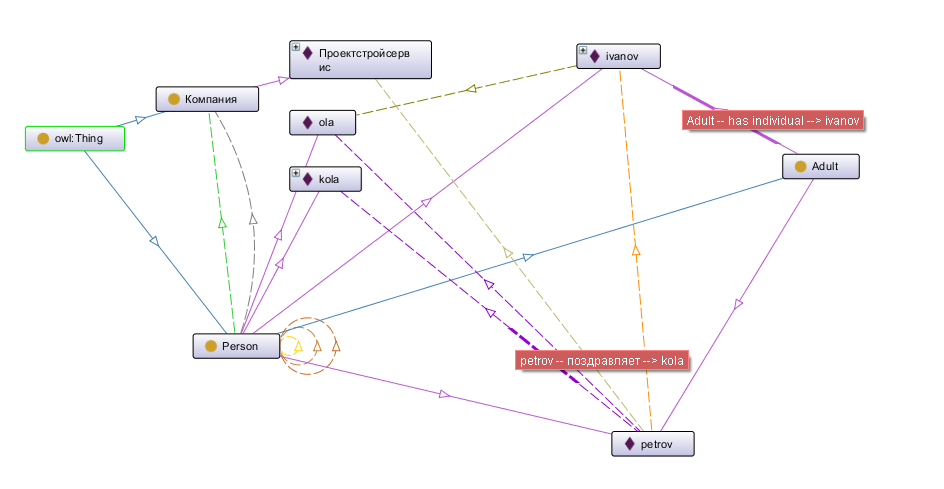
Знания, полученные в результате логического вывода, могут быть автоматически добавлены в онтологию. Для этого на вкладке SWRL последовательно выполните следующие действия:

1. Перенести знания из онтологии и правила в механизм вывода Drools.
2. Выполнить логический вывод
3. Поместить результаты вывода в онтологию.



В результате срабатывания правила S1 экземпляры класса People Иванов и Петров стали также представителями подкласса Adult (взрослый).

В результате срабатывания правил (S5,S7) в онтологии появились связи “поздравляет” между директором компании “Проектстройсервис” Петровым и детьми сотрудников этой компании.



**Задание**

* Изучите рассмотренные примеры.
* Создайте несколько правил для своей семантической сети.