

Методы и алгоритмы управления ресурсами на основе онтологий в среде Грид

Студент группы ДА-61м

Валерий Гончаренко

Постановка задачи

- Среда грид имеет гетерогенную, неоднородную структуру
- Огромные массивы данных
- При переходе на SOA, появляется задача подбора подходящего сервиса
- В основном применяется поиск ресурсов на основании ключевых слов
- Проблематично вводить новые понятия

Семантический поиск

- Не просто совпадение символов, а их суть.
- Развивается в концепции SemanticWeb и уже применяется на практике
- Возможности по выделению новых знаний
- Набор стандартизированных технологий (RDF(S), OWL, SPARQL, SWRL...)
- Набор программных библиотек и интерфейсов (KAON, Jena, OWL API, Virtuoso, Joseki ...)

Рассмотренные подходы к использованию онтологий

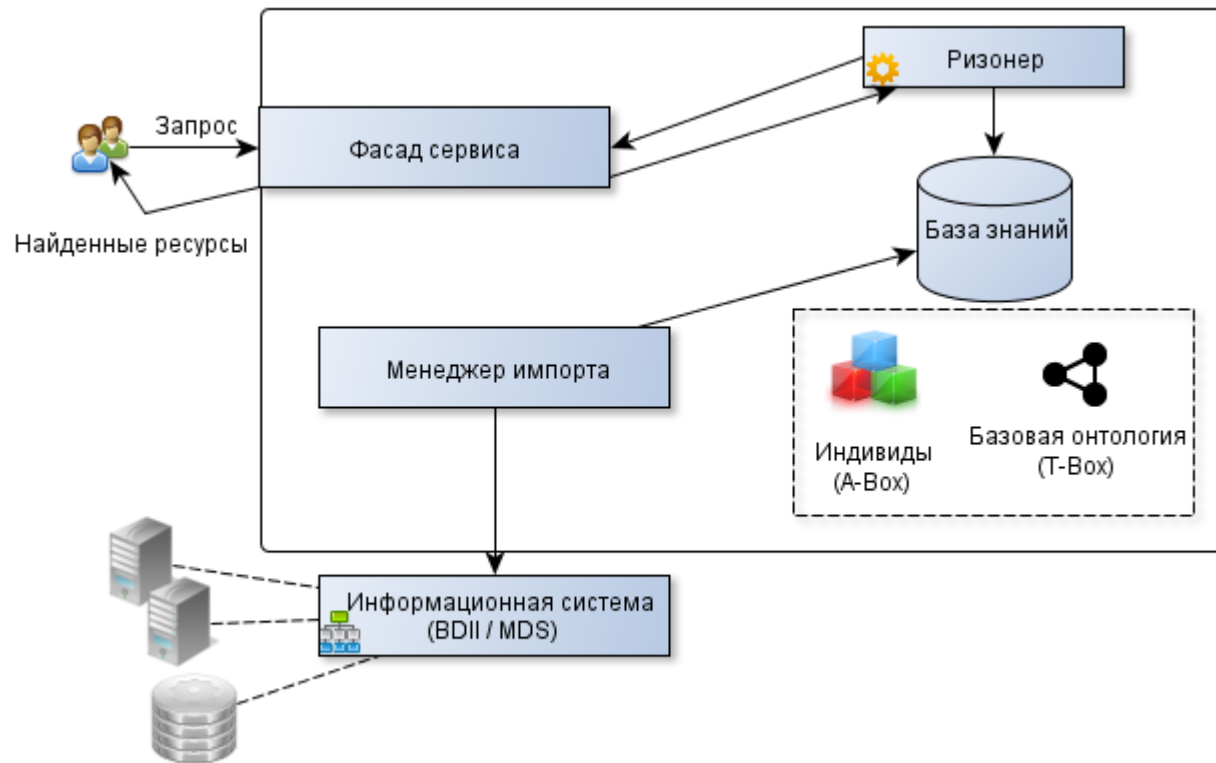
- Thamarai Selvi Somasundaram. Semantic-based Grid Resource Discovery and its Integration with the Grid Service Broker (Интеграция семантического слоя в GridBus)
- Said M. S-MDS: Semantic monitoring and discovery system (Реализация семантической MDS с использованием SPARQL)
- Balachandar R. Amarnath. Ontology-based Grid resource management (Использование правил на основе Algernon)

Рассмотренные подходы к использованию онтологий

- Neela Narayanan V. Resource Matchmaking in Grid Semantically (Двухэтапный поиск: Algernon + H-Match)
- Devis Bianchini: Hybrid Ontology-based Matchmaking for Service Discovery (Двухэтапный поиск)
- Felix Heine. Towards Ontology-Driven P2P Grid Resource Discovery (Организация одноранговой сети на основе объединения онтологий)

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

Общая архитектура системы подбора ресурсов



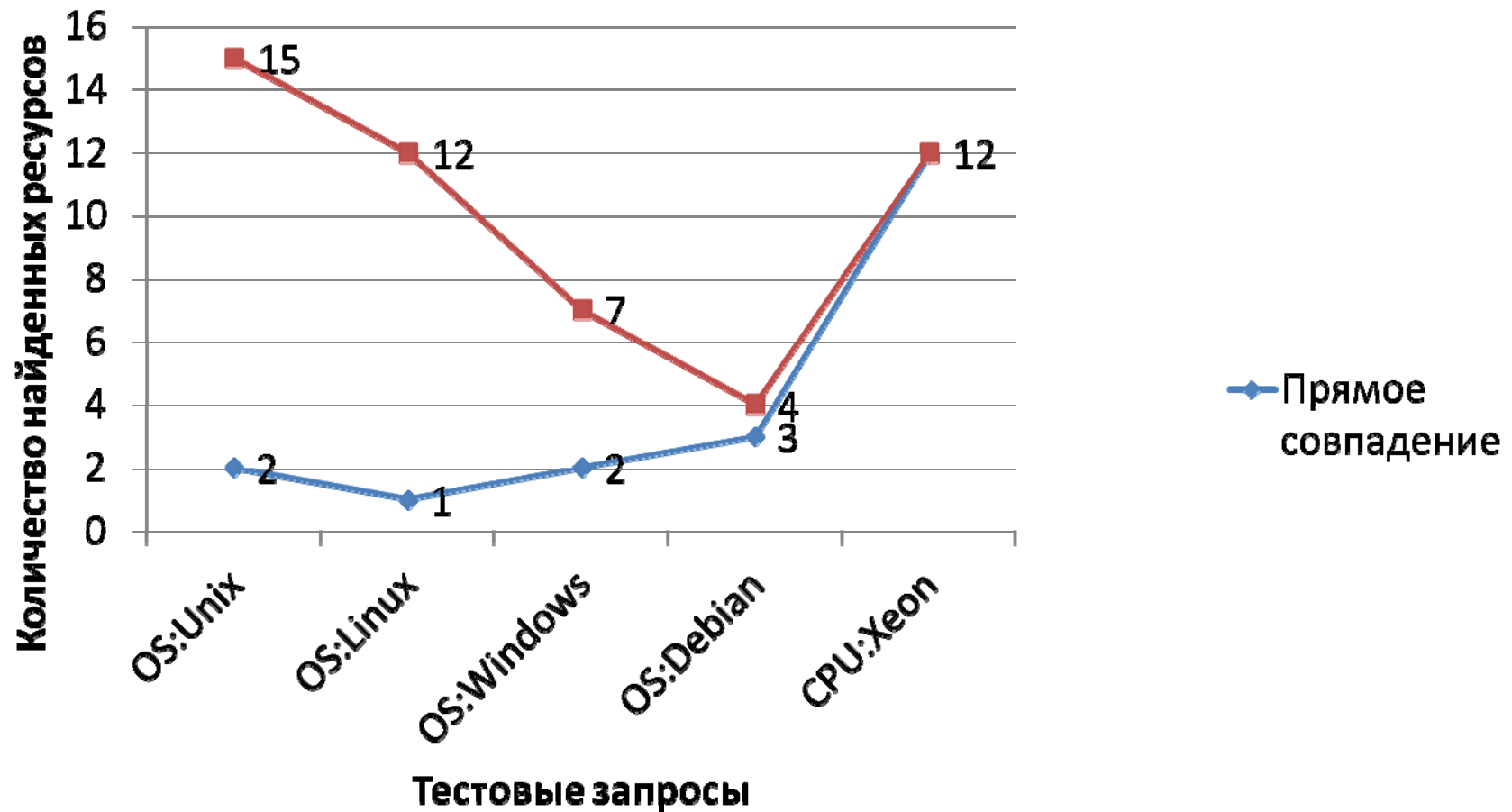
Рассматриваемая онтология



Проведенные исследования

1. Исследование эффективности подбора ресурсов с использованием традиционных и семантических методов
2. Исследование производительности систем на основе запросов Class expression и SPARQL

Эффективность подбора ресурсов с использованием традиционных и семантических методов

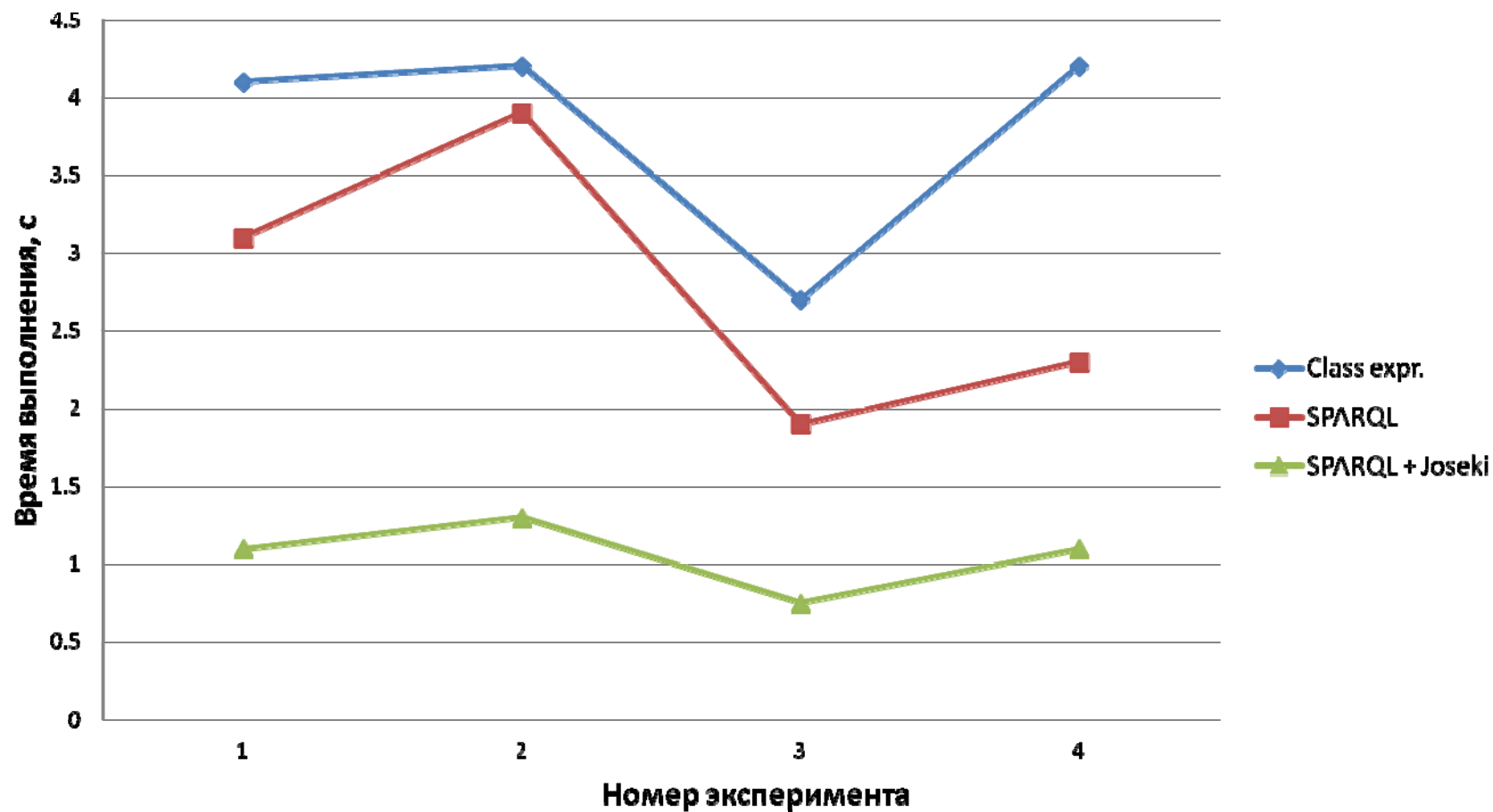


`SubCluster and describedBy some(Host and describedBy some Linux)`

Исследование производительности

- Загружено данные 1000 подкластеров Грид через LDAP (objectClass: GlueSubCluster) (~60000 триплетов)
- Машина для тестов: процессор Intel Pentium T4200 и 4 ГБ оперативной памяти.
- СЛВ: RacerPro
- Тестовые запросы:
 1. Получить экземпляры SubCluster, на которых установлены процессоры фирмы Intel.
 2. Получить экземпляры SubCluster с процессорами Intel в количестве больше 100 шт.
 3. Получить экземпляры SubCluster у которых больше 100 логических процессоров.
 4. Получить экземпляры SubCluster у которых стоит операционная система Linux.

Время выполнения запросов при выполнении class expression и SPARQL



Общие проблемы подходов

- Необходимость начальной классификации
 - Для данных 1000 подкластеров, время классификации: 80 – 100 с.
 - В случае загрузки всех данных с информационных сервисов: > 2-3 часов.
- Использование табличных (tableaux) систем логического вывода (СЛВ)
- Использование высокой степени экспрессивности онтологии

Направления дальнейших исследований

- Сделать переход к использованию OWL EL/RL
- Использовать СЛВ на основе правил
- Упростить онтологию
- Использовать механизм разделения онтологий (ontology modularity)
- Переход на использование SPARQL DL

Основные результаты работы

- Проанализированы существующие методы подбора ресурсов на основе онтологий.
- Разработана программная реализация системы подбора ресурсов, основанной на онтологии.
- Экспериментально исследована способность реализованной системы к подбору ресурсов Грид, а также ее производительность

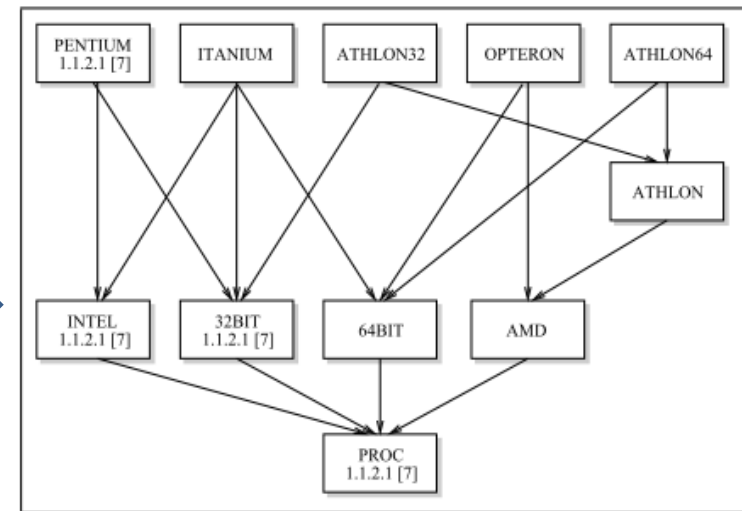
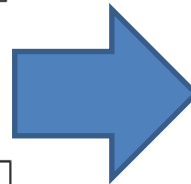
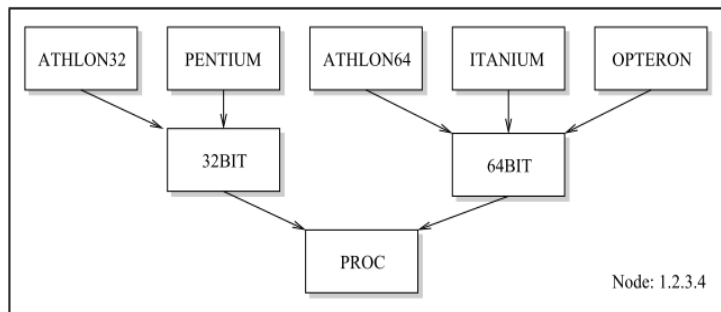
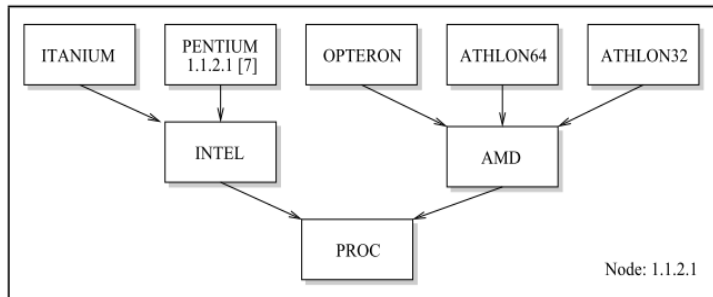
Спасибо за внимание!

ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

Пример описания установленного ПО на кластере

GlueHostApplicationSoftwareRunTimeEnvironment	LCG-2_4_0
GlueHostApplicationSoftwareRunTimeEnvironment	LCG-2_5_0
GlueHostApplicationSoftwareRunTimeEnvironment	LCG-2_6_0
GlueHostApplicationSoftwareRunTimeEnvironment	LCG-2_7_0
GlueHostApplicationSoftwareRunTimeEnvironment	MPI-START
GlueHostApplicationSoftwareRunTimeEnvironment	MPICH
GlueHostApplicationSoftwareRunTimeEnvironment	MPICH-1.2.7
GlueHostApplicationSoftwareRunTimeEnvironment	MPICH2
GlueHostApplicationSoftwareRunTimeEnvironment	MPICH2-1.0.4
GlueHostApplicationSoftwareRunTimeEnvironment	MPI_SHARED_HOME
GlueHostApplicationSoftwareRunTimeEnvironment	OPENMPI
GlueHostApplicationSoftwareRunTimeEnvironment	OPENMPI-1.2.8
GlueHostApplicationSoftwareRunTimeEnvironment	R-GMA
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_2_2_13
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_2_2_9
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_3_11_1_remove_scheduled
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_3_3_6
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_3_8_7
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_3_8_7_patch2
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_3_9_7
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_5
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_6
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_6_patch1
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_7
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_7_patch1
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_7_patch2
GlueHostApplicationSoftwareRunTimeEnvironment	VO-cms-CMSSW_4_1_7_patch3

Объединение знаний в подходе, основанном на P2P



Виды СЛВ

- 1. Табличные DL-процессоры. Имеют низкую производительность, но способны производить умозаключения на очень сложных онтологиях с множеством нетривиальных конструкций. (Pellet, RacerPro, FaCT++, Hermit, SHER).
- 2. Дизъюнктивные Datalog-процессоры. Трансформируют онтологию в дизъюнктивную Datalog программу и используют методику дедуктивных баз данных и правило резолюций. Удовлетворительное быстродействие, однако не поддерживают определенные OWL конструкции, в частности кардинальные ограничения и номиналы. (KAON2, не развивается).
- 3. Процессоры правил. Используют системы обработки правил для умозаключений в онтологиях. Высокая производительность, но поддержка лишь простых онтологий. (Sesame/OWLIM, Jena, OWLJessKB).

Примеры запросов SPARQL-DL

- Получить все классы:

```
SELECT ?c WHERE { Class(?c) }
```

- Узнать, является A подклассом B:

```
PREFIX ex: <http://example.com#>  
ASK { DirectSubClassOf(ex:A, ex:B) }
```

- Получить все транзитивные свойства индивида:

```
SELECT ?x  
WHERE { Transitive(?p),  
PropertyValue(<http://example.com#myClass>, ?p, ?x) }
```

- Получить иерархию классов, выводя пары родитель-подкласс:

```
SELECT ?a ?b  
WHERE { DirectSubClassOf(?a, ?b) }
```

- Объединить индивидов двух классов:

```
PREFIX wine: <http://www.w3.org/TR/2003/PR-owl-guide20031209/wine#>  
SELECT ?i  
WHERE { Type(?i, wine:PinotBlanc) }  
OR WHERE { Type(?i, wine:DryRedWine) }
```

OWL 2 QL

SubClassOf только между простыми выражениями
(класс или **ObjectSomeValuesFrom** с **owl:Thing**)

- + **ObjectSomeValuesFrom** в правом аргументе
- + **DisjointClasses** и **ObjectComplementOf** в правом аргументе
- + **DataSomeValuesFrom**

обратные свойства **InverseProperties** и включения **SubObjectProperty**

- × цепи **ObjectPropertyChain** и **SubDataPropertyOf**
- + области определения **ObjectPropertyDomain/Data...**
- + дизъюнкность **DisjointObjectProperties/Data...**
- + (а)симметричность, (ир)рефлексивность **SymmetricObject**
- × номиналы **ObjectHasValue/ObjectOneOf/Data...**
- × универсальные ограничения **ObjectAllValuesFrom/Data...**
- × объединение **ObjectUnionOf/DisjointUnion/DataUnionOf**

OWL 2 EL

кванторы существования **ObjectSomeValuesFrom/Data...**

✗ универсальные кванторы **ObjectAllValuesFrom/Data...**

✗ считающие кванторы **ObjectMaxCardinality ...**

пересечение **ObjectIntersectionOf/DataIntersectionOf**

✗ объединение **ObjectUnionOf/DisjointUnion/DataUnionOf**

✗ дополнение **ObjectComplementOf**

номиналы **ObjectOneOf/DataOneOf/ObjectHasValue/DataHasValue**

✗ длинные перечисления **ObjectOneOf/DataOneOf**

свойства **SubObjectPropertyOf/SubDataPropertyOf**

✗ обратные свойства **InverseObjectProperties**

✗ дизъюнктивность **DisjointObjectProperties/DisjointDataProperti**

области определения **ObjectPropertyDomain/ObjectPropertyRange/.**

OWL 2 RL

SubClassOf только между следующими выражениями $L \sqsubseteq R$

- + класс (кроме **owl:Thing**) и пересечения (**ObjectIntersectionOf**)
- L объединения (**ObjectUnionOf**)
- L перечисления (**ObjectOneOf**)
- R дополнение (**ObjectComplementOf**)
- L кванторы существования (**ObjectSomeValuesFrom/Data**)
- R универсальные кванторы (**ObjectAllValuesFrom/Data**)
- + кванторы с индивидом (**ObjectHasValue/Data**)
- R макс кардинальность 0/1 (**ObjectMaxCardinality/Data**)

все типы аксиом кроме **DisjointUnion** и **ReflexiveObjectProperty**