(1)应用

1.RDF(S) 文件 -> RDF Graph

SPARQL Query (BGP, OPTIONAL, UNION, ORDER BY, FILTER) -> answer

2. English -> RDF(S) Graph(Turtle synatx PRIFEX will be given)

3.SPARQL semantic, operator,solution mappings -> result

4.RDFS file -> RFDS Reasoning (new triples how and why)

-> SPARQL query

5.R2RML file , Database instance -> RDF Gragh

6.English <-> OWL/DL Synatx

7.Protege Screen Shot -> Axioms, Reasoning

8.input Query ，Tbox -> rewritten Query

9.Axioms in DL -> select those in DL-Lite\_R

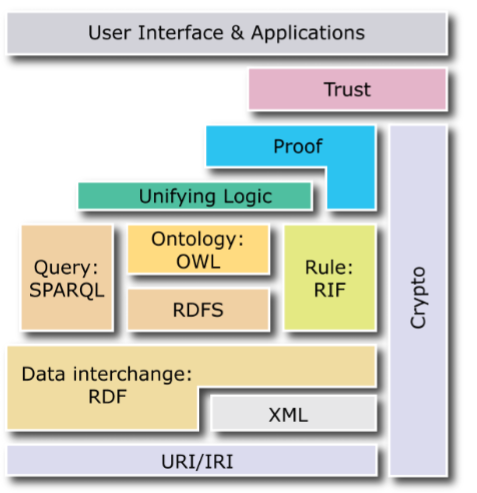
(2)问答解释

1.What is SW?

语义网是一种智能网络，它不但能够理解词语和概念，而且还能够理解它们之间的逻辑关系，可以使交流变得更有效率和价值。

语义网的核心是：通过给万维网上的文档如HTML添加能够被计算机所理解的语义“元数据”，从而使整个互联网成为一个通用的信息交换媒介。

1. Semantic web stack (W3C standard)



3.What is Jena? What can you do with Jena?

jena是一个java 的API,用于语义网中的应用程序开发

a)       以RDF/XML、三元组形式读写RDF

b)       RDFS，OWL，DAML+OIL等本体的操作

c)       利用数据库保存数据

d)       查询模型

e)       基于规则的推理

4.SPARQL Query types?

SELECT

Returns all, or a subset of, the variables bound in a query pattern match.

CONSTRUCT

Returns an RDF graph constructed by substituting variables in a set of triple templates.

ASK

Returns a boolean indicating whether a query pattern matches or not.

DESCRIBE

Returns an RDF graph that describes the resources found.

1. New features in SPARQL 1.1?

和SPARQL 1.0相比，SPARQL 1.1增加了一组新的查询功能，包括子查询（subqueries），赋值（value assignment），路径表达式（path expressions）和聚合查询（aggregates），如上面例子中用到的 COUNT 查询。

6.Constructs in RDFS?

1. DM vs R2RML?

The direct mapping defines an [RDF Graph](http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/" \l "dfn-rdf-graph) [[RDF-concepts]](http://www.w3.org/TR/rdb-direct-mapping/" \l "RDF-concepts) representation of the data in a relational database. The direct mapping takes as input a relational database (data and schema), and generates an RDF graph that is called the direct graph. The algorithms in this document compose a graph of relative IRIs which must be resolved against a base IRI [[RFC3987]](http://www.w3.org/TR/rdb-direct-mapping/" \l "RFC3987) to form an RDF graph.

RDB to RDF Mapping Language (R2RML) 是一种提供关系数据库和 RDF 数据集之间映射的语言。R2RML 规范使用 RDF/Turtle 规范提供这些映射。它的中心思想是尽可能实现直接的、简单的映射。例如，这种形式的映射允许用户使用 SPARQL 查询关系数据库，过去只能通过使用结构化查询语言 (SQL) 实现。它还允许跨 Linked Data Web 在行级别链接关系数据库。这一概念的功能非常强大，因为它消除了手动创建的中间层，并且动态地将关系数据公开为 RDF。R2RML 使您可以轻松地将所有封闭的关系数据库转换为 Web 可访问的模式。

DM：

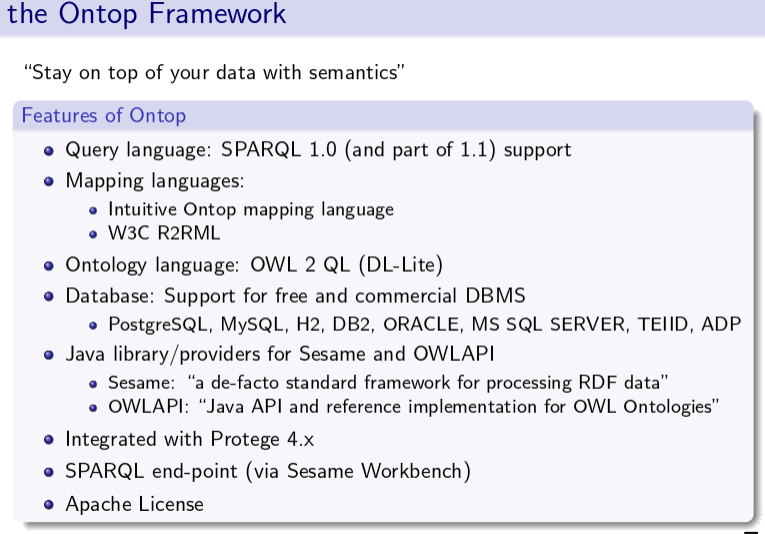
DB -> rdf gragh

不受人干预

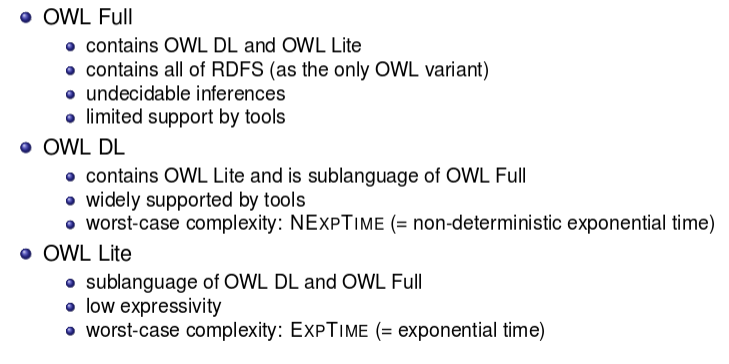
DB,.. -> rdf gragh

8) Features in Ontop?

Ontop is a framework for ontology based data access (OBDA). It supports SPARQL over virtual RDF graphs defined through mappings to RDBMS.



9) OWL 1 Variants? 本体描述语言



OWL Lite：表达能力最弱，OWL DL的一个子集，它通过降低了OWL DL中的公理约束来达到高效的推理。它支持的基数约束只有0和1.

OWL DL：将可判定推理能力和较强表达能力最为首要目标。它包含了OWL语言的所有成分，但使用时须收到一定的约束。

OWL Full：OWL Full包含OWL的全部语言成分并取消了OWL DL中的限制，它将RDFS扩展为一个完备的本体语言，支持那些不需要可计算性保证（no computational guarantees）但需要最强表达能力和完全自由的RDFS用户。

10) Relations between RDF/RDFS/OWL ??

RDF 通过类、属性和值来描述资源。   
此外，RDF 还需要一种定义应用程序专业的类和属性的方法。应用程序专用的类和属性必须使用对 RDF 的扩展来定义。   
RDF Schema 就是这样一种扩展。   
RDF Schema (RDFS)RDF Schema 不提供实际的应用程序专用的类和属性，而是提供了描述应用程序专用的类和属性的框架。   
RDF Schema 中的类与面向对象编程语言中的类非常相似。这就使得资源能够作为类的实例和类的子类来被定义。   
OWL 被设计为由计算机应用程序来读取（而不是被人类），较之 RDF， OWL 是一门具有更强机器解释能力的更强大的语言，OWL 拥有更大的词汇表以及更强大的语言。

1. DL Knowledge Base??

12) DL Syntax

DL：描述逻辑

一个描述逻辑系统包括四个基本的组成部分：

（1）表示概念和关系的构造集；

（2）Tbox术语集（概念术语的断言集合）；

（3）Abox断言集（个体的断言集合）；

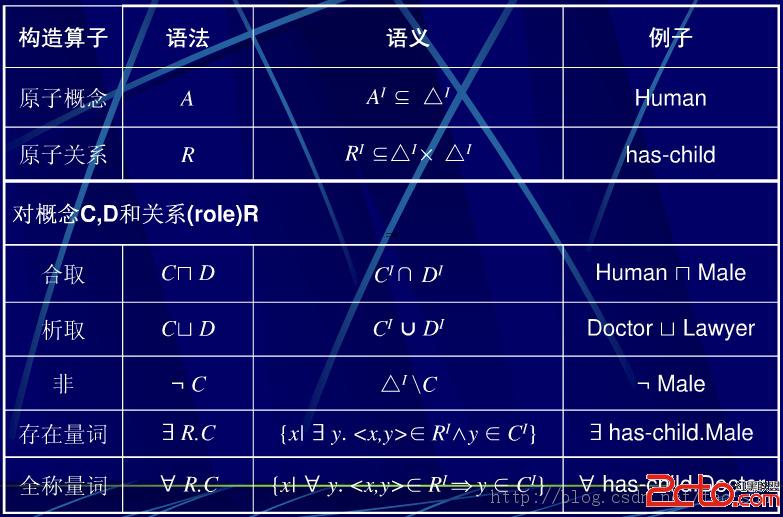
（4）Tbox和Abox上的推理机制

什么是描述逻辑（DL）？  
一种基于对象的知识表示的形式化。  
  
建立在概念和关系（Role）之上。  
概念：对象的集合  
关系：对象之间的二元关系  
  
是一阶逻辑FOL的一个可判定的子集  
  
特点：  
1. 具有很强的表达能力  
2. 是可判定的，总能保证推理算法终止

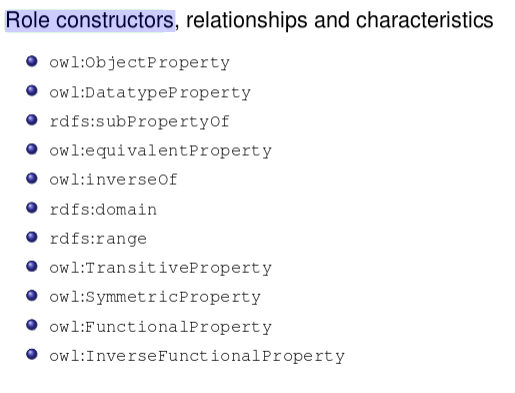
DL的体系结构：  
1. 表示概念和关系（Role）的构造集  
2. TBox(Terminology Box)：描述领域结构的公理集，包含概念定义及公理  
##可以理解为对类别的定义  
3. ABox(Assertional Box)：具体个体的公理集，包含概念断言和关系断言  
##可以理解为对个体的定义，以及具体的个体间的关系  
4. TBox和ABox上的推理机制：一个基于DL的知识库就是K=TBox+ABox，简写为 KB(T,A) ##KB即Knowledge Base

DL的基本元素：概念和关系  
概念：一个领域的子集。如学生、孩子、哺乳动物等概念  
{x|Student(x)},{x|Children(x)}  
关系（Role）:属性，二元关系。如朋友，夫妻  
{<x,y>|Friend(x,y)},{<x,y>|Couple(x,y)}

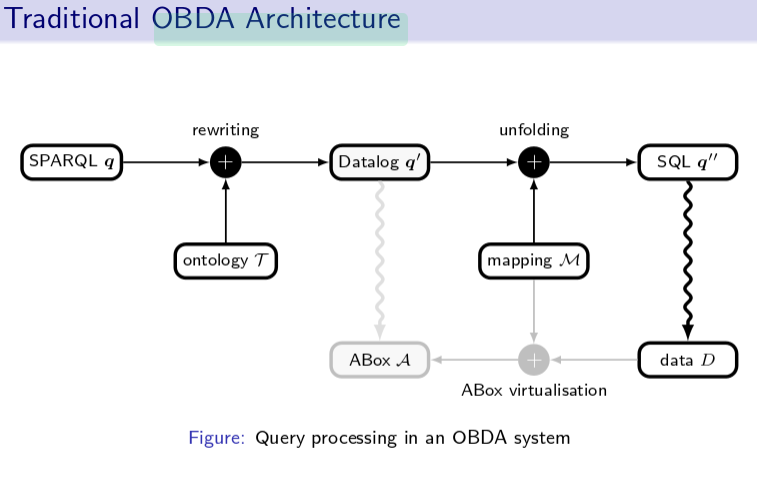
**TBox:描述领域结构的公理的集合**  
1. 引入概念的名称，表示类（一元谓词）  
{x"Student(x)}  
2. 声明包含关系的公理（属性，二元谓词）  
{<x,y>|Friend(x,y)}  
（如图1）  
  
一个解释I满足TBox T iff 它满足T中的每个公理（I entails T ）  
## 这里蕴含符号打不出来，使用 entails 代替  
## 逻辑符号表可参见：http://en.wikipedia.org/wiki/Table\_of\_logic\_symbols  
  
**ABox：断言部分，是描述具体清晰的公理的结合**  
1. 概念断言：表示一个对象是否属于某个概念  
a:C or C(a)  
例如：Student(Tom) 表示Tom是一个学生，也可以用Tom:Student表示  
  
2. 关系断言：表示两个对象是否满足一定的关系  
<a,b>:R or R(a,b)  
例如：hasChild(John,Mary) 表示John有个孩子叫Mary  
  
一个解释I满足ABox A iff 它满足A中的每个公理，记为 I entails A  
## I 被称作一个解释（Interpretation），实质上就是一个模型。  
  
一个解释I满足知识库∑=<T,A> iff 它满足T和A，记为 I entails ∑



13) Role Constructors in OWL?



14) OBDA Architecture



15) Right languages in OBDA?

DL

1. Virtual ABox ?

17) Query Rewriting ?

1the data is not manipulated

2the axioms are embedded into the query

3the rewritten query is translated to SQL of strange shape and quite big size in the worst case

4the translated query is answered by the DBMS

1. Constructors in DL-Lite\_R ?

一般地，描述逻辑依据提供的构造算子，在简单的 概念和关系上构造出复杂的概念和关系。 通常DL至少包含以下算子： ◆ 合取( )，吸取( )，非( ) ◆ 量词约束：存在量词( )，全称量词() 最基本的DL称之为ALC 例如，ALC中概念Happy-father定义为： Man has-child.Male has-child.Female has-child.(Doctor Lawyer)

DL中的其它算子

构造算子 语法 ≥n R . C 数量约束 语义 {x| | {y|<x,y>∈ RI ,y ∈ CI} | ≥n} 例子 ≥3 has-child .Male

≤ n R . C {x| | {y|<x,y>∈ RI ,y ∈ CI} | ≤ n} ≤ 3 has-child .Male RR\* {<y,x>|<x,y>∈ RI } (RI )\* has-childhas-child\*

逆 传递闭包

另外，有两个类似于FOL中的全集(true)和空集(false)的算子

top Bottom T ⊥ △I Male Male Man Man

参考资料：

描述逻辑与本体描述语言：<http://wenku.baidu.com/link?url=hySwXO9QrTYQs7KpHAG0H6RanXqG_Fm-HYS3LxgpqSBAD6zvNVKCxVvzG9gCe-bPLqyyOF1mes7QLKK8oF9nwVBqwmM4R28LlsP6eEd2DE3>

RDF 概念上的更新和一些本体论知识：

<http://www.ibm.com/developerworks/cn/xml/x-rdfconcepts/>

SPARQL 1.1概述 (中文翻译)：

<http://www.chinaw3c.org/REC-sparql11-overview-20130321-cn.html>

# 描述逻辑ABOX和TBOX：

<http://wenku.baidu.com/link?url=Edb9f96ceLK-Ppt7OsuegS-F1GYT-TrlrrKz49iRsjUgF0ae8ZxQ_fZ6ADFLt5BOvGkr3uZly7K435N2bm_nlCEDtIGdpJkt7aqfM1vLcye>