

## 早期的知识表示方法

### 1.一阶谓词逻辑 (First-Order Logic)

Horn逻辑：一阶谓词逻辑的子集  $\text{has\_child}(a,b)$

描述逻辑：一阶谓词逻辑的可判定子集  $\mathcal{O} := \langle T, A \rangle$

### 2.产生式系统 IF ..THEN...CF(置信度, 知识强度)

### 3.框架

框架名:	小A抢劫杀人案
犯罪意图:	抢劫
犯罪结果:	杀人
被杀者:	小B
知情人:	小C
罪犯:	小A
条件一:	有小C指控小A
条件二:	小A招认

### 4.语义网络

利用节点和带标记的边结构的有向图描述事件、概念、状况、动作及客体之间的关系

## 基于语义网的知识表示框架

RDF:资源描述框架(Resource Description Framework, RDF)

R代表页面, 图片、视频等任何具有URI标识符

D标识属性、特征和资源之间的关系

F标识模型、语言和这些描述的语法。

在RDF中, 知识总是以**三元组**的形式出现, 即每一份知识都可以被分解为: (subject, predicate, object)。**主谓宾**

图模型: RDF三元组可以看做是图模型的边和顶点

在RDF中resource和properties是以URIs的形式表示的

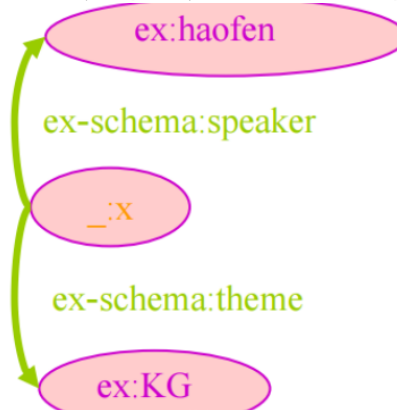




RDF空白结点:

RDF允许存在匿名的资源, 它可以作为连接其他非匿名资源的桥梁, 此时RDF中会出现空白结点, 因为这个资源不被URI表示, 所以可以表示为 `_:xyz`。

Haofen是某一次KG讲座的讲者。



开放世界假设。这表示意味着CCF ADL讲座至少有一位讲者  
分布式定义可以合并

带标注的RDF: 扩展RDF用来表示更多的信息, 如时间、不确定性、空间、信任等

语法形式为:  $(s,p,o):\lambda(s,p,o):\lambda$

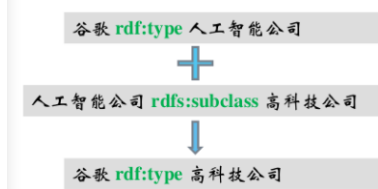
$\lambda$ 是一个标志, 可以是上述的扩展信息

例子: (特朗普, 就职, 总统): 2017年1月

RDF Schema (RDFS) : RDFS在RDF基础上提供了一个术语、概念等的定义方式, 以及哪些属性可以应用到哪些对象上。

Class、subClass、type、Property、subPropertyOf、Domain、Range

支持推理功能:



# OWL

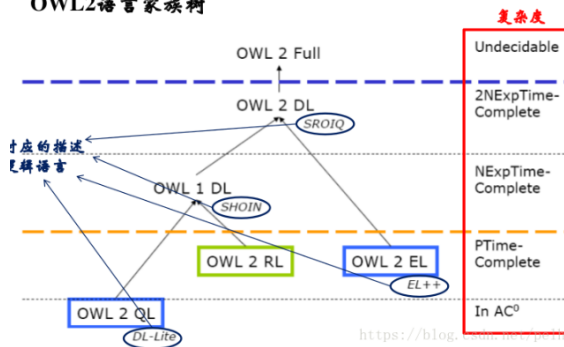
## Web本体语言Web Ontology Language

RDF(S)语义表达能力显得太弱，还缺少诸多常用的特征。

owl:equivalentClass	等价类
owl:equivalentProperty	等价属性
owl:sameIndividualAs	等价个体
owl:TransitiveProperty	属性传递
owl:inverseOf	属性互反
owl:FunctionalProperty	属性函数性
owl:SymmetricProperty	属性对称性
owl:allValuesFrom owl:onProperty owl:someValuesFrom owl:onProperty owl:cardinality owl:onProperty	属性约束
owl:intersectionOf	相交的类
owl:oneOf	声明枚举类型
owl:disjointWith	声明两个类不相交
owl:unionOf	声明类的并运算
owl:minCardinality owl:maxCardinality	最小最大的基数限定
owl:InverseFunctionalProperty	声明互反类具有函数属性
owl:hasValue	属性的局部约束时,声明所约束类必有一个取值

## OWL2:OWL的最新版本

### OWL2语言家族树



## OWL2 QL:query language,基于本体的查询

允许的核心词汇	对应的描述逻辑公理举例
rdfs:subClassOf	$Mother \sqsubseteq Person$
rdfs:subPropertyOf	$hasSon \sqsubseteq hasChild$
rdfs:domain	$\exists hasSon. T \sqsubseteq Person$
rdfs:range	$T \sqsubseteq \forall hasSon. Person$
owl:inverseOf	$hasChild \equiv hasParent^{-}$
owl:disjointWith	$Women \sqcap Man \sqsubseteq \perp$

基于QL的本体查询可以优化到多项式对数时间复杂度

OWL2 EL:概念术语描述,推理而设计

允许的核心词汇	对应的描述逻辑公理举例
rdfs:subClassOf	$Mother \sqsubseteq Person$
rdfs:subPropertyOf	$hasSon \sqsubseteq hasChild$
owl:someValuesOf	$\exists hasSon.Children \sqsubseteq Person$ $Parent \sqsubseteq \exists hasSon.Children$
owl:intersectionOf	$Star \sqcap Women \sqsubseteq Scandal$
owl:TransitiveProperty	$Tran( hasAncestor )$ <a href="https://">https://</a>

OWL2 RL:为高效推理设计的本体语言(推理针对的是实例数据)

允许的核心词汇
rdfs:subClassOf
rdfs:subPropertyOf
rdfs:domain
rdfs:range
owl:TransitiveProperty
owl:FunctionalProperty
owl:sameAs
owl:equivalentClass
owl:equivalentProperty
owl:someValuesFrom
owl:allValuesFrom

SPARQL:RDF查询语言

可以针对不同的数据集撰写复杂的连接，支持主流的图数据库。

与SQL很相似，定义了OPTIONAL、FILTER、UNION、FROM等关键字，同时它支持嵌套查询，基于规则的查询

可跨库查询

A SPARQL query comprises, in order:

- Prefix declarations, for abbreviating URIs
- Dataset definition, stating what RDF graph(s) are being queried
- A result clause, identifying what information to return from the query
- The query pattern, specifying what to query for in the underlying dataset
- Query modifiers, slicing, ordering, and otherwise rearranging query results

```
# prefix declarations
PREFIX foo: <http://example
...
# dataset definition
FROM ...
# result clause

SELECT ...
# query pattern
WHERE {
  ...
}
# query modifiers
ORDER BY ...
```

变量,RDF中的资源以 “?” 或者 “\$” 指示

查询学生姓名，选修课程，以及他们的年龄；如果有年龄，那么年龄必须大于25岁。

```
SELECT ?module ?name ?age
WHERE {
  ?student exp:studies ?module .
  ?student foaf:name ?name .
  OPTIONAL {
    ?student exp:age ?age .
  }
  FILTER (?age > 25)
}
```

finance:融创中国	rdf:type	finance:地产事业
finance:孙宝斌	finance:control	finance:融创中国
finance:贾跃亭	finance:control	finance:乐视网
finance:孙宝斌	finance:hold_share	finance:乐视网
finance:王健林	finance:control	finance:万达集团
finance:万达集团	finance:main_income	finance:地产事业
finance:融创中国	finance:acquire	finance:乐视网
finance:融创中国	finance:acquire	finance:万达集团

<b>SELECT ?P ?X</b> <b>WHERE {</b> ?P finance:control ?c . ?c finance:acquire ?X . <b>}</b>	<b>查询结果</b>						
	<table> <tr><th>?P</th><th>?X</th></tr> <tr><td>孙宝斌</td><td>乐视网</td></tr> <tr><td>孙宝斌</td><td>万达集团</td></tr> </table>	?P	?X	孙宝斌	乐视网	孙宝斌	万达集团
?P	?X						
孙宝斌	乐视网						
孙宝斌	万达集团						

## Json-LD: 基于JSON表示和传输互联数据

```
{
  "http://schema.org/name": "Manu Sporny",
  "http://schema.org/url": { "@id": "http://manu.sporny.org/" }, "http://schema.org/image":
  { "@id": "http://manu.sporny.org/images/manu.png" }
}
```

- JSON-LD通过引入规范术语表示，比如统一化表示“name”，“homepage”和“image”的URI，使得数据交换和机器理解成为基础。

## JSON-LD的语法和JSON兼容

## RDFa: (Resource Description Framework in attributes)网页标记语言

- 通过引入名字空间的方法在已有的标签中加入RDFa相应的属性来使得支持RDFa技术的浏览器或者搜索引擎可以解析到，从而达到优化的目的。

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/"
  about="http://www.example.com/books/wikinomics">
  <span property="dc:title">Wikinomics</span>
  <span property="dc:creator">Mr right</span>
  <span property="dc:date">2006-09-02</span>
</div>
```

- 上面的代码示例中用到了RDFa属性中的about属性和property属性，这段代码示例说明了一篇文章，然后描述了和这篇文章相关的信息，比如说标题，创建者和创建日期，而这些属性就可以使得支持RDFa的机器识别。

从机器可理解的层面优化搜索，提升访问性以及网页数据的关联性。

## HTML5 Microdata: 在网页标记语言嵌入机器可读的属性数据

```
<section itemscope itemtype="http://data-vocabulary.org/Person">
  <h1 itemprop="name">Andy</h1>
  <p></p>
  <a itemprop="url" href="http://www.example.com/blog">My Blog</a>
</section>
```

定义一个概念类型  
通过itemtype指示

实体Andy的两个属性，  
通过itemprop标签指示

在自定义命名空间  
下的实体Andy

浏览器可以很方便地从网页上提取微数据实体、属性以及属性值。

## RDF语义模型——关系显示定义

## 关系模型——关系隐式声明

