



# 人工智能导论

## 基于知识的人工智能

郭兰哲

南京大学 智能科学与技术学院

Homepage: [www.lamda.nju.edu.cn/guolz](http://www.lamda.nju.edu.cn/guolz)

Email: [guolz@nju.edu.cn](mailto:guolz@nju.edu.cn)

# 回顾：命题逻辑

- 语法、语义、逻辑连接词
- 逻辑等价 -> 命题形式转换
- 归结与归结证明
- 确定子句，前向推理、后向推理
- 命题逻辑的不足：
  - 表达能力有限，没有办法简单的描述具有多个对象的环境（不像自然语言）
  - 例：没办法描述“与无底洞相邻的方格有微风”

# 从命题逻辑到谓词逻辑

□ 在命题逻辑中，每个陈述句是最基本的单位(即原子命题)，无法对原子命题进行分解，因此在命题逻辑中，**不能表达局部与整体、一般与个别的关系**

□ 例如，对于苏格拉底论断

- ✓  $\alpha$ : 所有的人总是要死的
- ✓  $\beta$ : 苏格拉底是人
- ✓  $\gamma$  : 所以苏格拉底是要死的

无法在命题逻辑基础上推导出： $\alpha \wedge \beta \rightarrow \gamma$

虽然是正确的，但是无法通过命题逻辑来进行推理判断

需要引入更加强大的逻辑表示方法，谓词逻辑

# 谓词逻辑

---

在谓词逻辑中，将原子命题进一步细化，分解出**个体**、**谓词**和**量词**，来表达个体与总体的内在联系和数量关系

谓词逻辑中三个核心概念：

- 个体、谓词 (predicate) 和量词 (quantifier)

# 谓词逻辑

□ 个体：个体是指所研究领域中可以独立存在的具体或抽象的概念

具体或特定的个体：“张三” “李四” “1” “2” “3”

抽象或泛指的个体：x, y, z

所有个体对应的集合为个体域

# 谓词逻辑

□ 谓词：谓词是用来刻画个体属性或者描述个体之间关系存在性的元素，其值为真或为假

- 包含一个参数的谓词称为一元谓词，表示一元关系，通常是用来刻画个体是否包含特定的属性：如 $p(x)$ :  $x$ 是质数，表示某个数字是否是质数
- 包含多个参数的谓词称为多元谓词，表示个体间的多元关系，通常用于描述个体之间是否存在特定的关联，如 $\text{Father}(x, y)$ : 表示 $x$ 是 $y$ 的父亲

# 谓词逻辑：谓词与函数的区别

- 函数中个体变元用个体常量（来自定义域）代入后结果仍然是**个体（值域）**，如定义函数 $f(x) = x + 10$ ，则 $f(2) = 12$
- 谓词中个体变元用个体常量带入后就变成了**命题**，如 $p(x)$ : $x$ 是质数，带入 $x = 3$ ， $p(3)$ 是命题
- 函数是从定义域到值域的映射；谓词是从定义域到{Ture, False}的映射

# 谓词逻辑：事实符号化

例：将下列客观事实符号化

1. Richard是国王。
2. Lucy和Lily是姐妹。
3. 北京是中国的首都。

(1) *King(Richard)*。其中，Richard是一个个体常量，*King*是一个描述“国王”这个一元关系的谓词。

(2) *Sister(Lucy, Lily)*。其中，Lucy和Lily是两个个体常量，*Sister*是一个描述“姐妹”这个二元关系的谓词。

(3) *Capital(北京, 中国)*。其中，北京和中国是两个个体常量，*Capital*是一个描述“首都”这个二元关系的谓词。

# 谓词逻辑：量词

## □ 全称量词 (universal quantifier, $\forall$ )

全称量词用符号 $\forall$ 表示，表示一切的、凡是的、所有的、每一个等。

$\forall x$ 表示定义域中的所有个体， $\forall xP(x)$ 表示定义域中的所有个体具有性质 $P$

## □ 存在量词(existential quantifier, $\exists$ )

存在量词用符号 $\exists$ 表示，表示存在、有一个、某些等。

$\exists x$ 表示定义域中存在一个或若干个个体，

$\exists xP(x)$ 表示定义域中存在一个个体或若干个体具有性质 $P$

## □ 全称量词和存在量词统称为量词

# 谓词逻辑

---

## □ 全称量词与存在量词之间的关系

- $\forall x P(x) \equiv \neg \exists x \neg P(x)$
- $\forall x \neg P(x) \equiv \neg \exists x P(x)$
- $\neg \forall x P(x) \equiv \exists x \neg P(x)$
- $\exists x P(x) \equiv \neg \forall x \neg P(x)$

# 谓词逻辑

- 约束变元：在全称量词或存在量词约束条件下的变量称为约束变元
- 自由变元：不受全称量词或存在量词约束的变量称为自由变元

自由变元即可以存在于量词的约束范围之内，也可以存在于量词的约束范围之外

$$(\forall x)(A(x) \vee B) \equiv (\forall x)A(x) \vee B$$

$$(\forall x)(A(x) \wedge B) \equiv (\forall x)A(x) \wedge B$$

$$(\exists x)(A(x) \vee B) \equiv (\exists x)A(x) \vee B$$

$$(\exists x)(A(x) \wedge B) \equiv (\exists x)A(x) \wedge B$$

# 谓词逻辑

□ 在约束变元相同的情况下，全称量词对合取满足分配律，存在量词对析取满足分配律

$$(\forall x)(A(x) \wedge B(x)) \equiv (\forall x)A(x) \wedge (\forall x)B(x)$$

$$(\exists x)(A(x) \vee B(x)) \equiv (\exists x)A(x) \vee (\exists x)B(x)$$

$$(\forall x)(A(x) \vee B(x)) \equiv (\forall x)A(x) \vee (\forall x)B(x) \quad \text{(不成立)}$$

$$(\exists x)(A(x) \wedge B(x)) \equiv (\exists x)A(x) \wedge (\exists x)B(x) \quad \text{(不成立)}$$

# 谓词逻辑

例：使用全称量词或存在量词描述下列事实。

- (1) 所有的国王都是人。
- (2) 所有的国王都头戴皇冠。

(1) “对于所有的 $x$ ，如果 $x$ 是国王，那么 $x$ 是人”， 其符号化表示为 $(\forall x)(King(x) \rightarrow Person(x))$ 。

其中 $x$ 是变量符号，由于 $x$ 受到全称量词的约束，因此 $x$ 是约束变元； $King(x)$ 是一个一元谓词，表示 $x$ 是国王， $Person(x)$ 是一个一元谓词，表示 $x$ 是人。

(2) “对于所有的 $x$ ，如果 $x$ 是国王，那么 $x$ 头戴皇冠”， 符号化表示为 $(\forall x)(King(x) \rightarrow Head\_On(Crown, x))$ 。

其中 $x$ 是变量符号，由于 $x$ 受到全称量词的约束，因此 $x$ 是约束变元； $Crown$ 是一个常量符号，表示皇冠； $King(x)$ 是一个一元谓词，表示 $x$ 是国王， $Head\_On(Crown, x)$ 是一个二元谓词，表示 $x$ 头戴皇冠。

# 谓词逻辑

□ 当公式中存在多个量词时：

- ✓ 若多个量词都是全称量词或者都是存在量词，则量词的位置可以互换
- ✓ 若多个量词中既有全称量词又有存在量词，则量词的位置不可以互换

设  $A(x, y)$  是包含变元  $x, y$  的谓词公式，则如下关系成立：

$$(\forall x)(\forall y)A(x, y) \Leftrightarrow (\forall y)(\forall x)A(x, y)$$

$$(\exists x)(\exists y)A(x, y) \Leftrightarrow (\exists y)(\exists x)A(x, y)$$

$$(\forall x)(\forall y)A(x, y) \Rightarrow (\exists y)(\forall x)A(x, y)$$

$$(\forall x)(\forall y)A(x, y) \Rightarrow (\exists x)(\forall y)A(x, y)$$

$$(\exists y)(\forall x)A(x, y) \Rightarrow (\forall x)(\exists y)A(x, y)$$

$$(\exists x)(\forall y)A(x, y) \Rightarrow (\forall y)(\exists x)A(x, y)$$

$$(\forall x)(\exists y)A(x, y) \Rightarrow (\exists y)(\exists x)A(x, y)$$

$$(\forall y)(\exists x)A(x, y) \Rightarrow (\exists x)(\exists y)A(x, y)$$

# Wumpus世界

定义时间为 $t$ , agent所处的位置为 $x$

$$\forall x, t \ At(\text{Agent}, x, t) \wedge \text{Smelt}(t) \Rightarrow \text{Smelly}(x)$$

$$\forall x, t \ At(\text{Agent}, x, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(x)$$

$$\forall y \ Breezy(y) \Rightarrow \exists x \ Pit(x) \wedge \text{Adjacent}(x, y)$$

$$\forall x, y \ Pit(x) \wedge \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y)$$

$$\forall y \ Breezy(y) \Leftrightarrow [\exists x \ Pit(x) \wedge \text{Adjacent}(x, y)]$$

# 谓词逻辑：推理规则

设 $A(x)$ 是谓词公式， $x$ 和 $y$ 是变元， $a$ 是常量符号，则存在如下谓词逻辑中的推理规则：

- 全称量词消去 (Universal Instantiation, UI) :  $(\forall x)A(x) \rightarrow A(y)$
- 全称量词引入 (Universal Generalization, UG) :  $A(y) \rightarrow (\forall x)A(x)$
- 存在量词消去 (Existential Instantiation, EI) :  $(\exists x)A(x) \rightarrow A(c)$
- 存在量词引入 (Existential Generalization, EG) :  $A(a) \rightarrow (\exists x)A(x)$

# 谓词逻辑：推理规则

例：已知

- $(\forall x)(P(x) \rightarrow Q(x))$
- $(\forall x)(Q(x) \rightarrow R(x))$

证明： $(\forall x)(P(x) \rightarrow R(x))$

证明过程：

- $(\forall x)(P(x) \rightarrow Q(x))$
- $P(x) \rightarrow Q(x)$  (消去全称量词)
- $(\forall x)(Q(x) \rightarrow R(x))$
- $Q(x) \rightarrow R(x)$  (消去全称量词)
- $P(x) \rightarrow R(x)$
- $(\forall x)(P(x) \rightarrow R(x))$  (引入 $x$ )

# 谓词逻辑：推理规则

例：已知

- $(\forall x)(F(x) \rightarrow (G(x) \wedge H(x)))$
- $(\exists x)(F(x) \wedge P(x))$

试证明： $(\exists x)(P(x) \wedge H(x))$

证明过程：

1.  $(\forall x)(F(x) \rightarrow (G(x) \wedge H(x)))$  (已知)
2.  $(\exists x)(F(x) \wedge P(x))$  (已知)
3.  $F(a) \wedge P(a)$  (2的EI)
4.  $F(a) \rightarrow (G(a) \wedge H(a))$  (1的UI)
5.  $F(a)$  (由3知)
6.  $G(a) \wedge H(a)$  (4和5的假言推理)
7.  $P(a)$  (由3知)
8.  $H(a)$  (由6知)
9.  $P(a) \wedge H(a)$  (7和8的合取)
10.  $(\exists x)(P(x) \wedge H(x))$  (9的EG)

# 自然语言的形式化

为了对自然语言所描述的内容进行精确表达，可通过**逻辑语言**

## 对自然语言进行形式化

例：每一个奇数均存在一个大于它的奇数：

- $\text{odd}(x)$ :  $x$ 是奇数
- $\text{Great}(x, y)$ :  $x$  大于 $y$
- $(\forall x) \left( \text{odd}(x) \rightarrow (\exists y)(\text{odd}(y) \wedge \text{Great}(y, x)) \right)$

# 自然语言的形式化

例：证明苏格拉底三段论 “所有人都是要死的， 苏格拉底是人， 所以苏格拉底是要死的”。

设 $F(x)$ :  $x$ 是人,  $G(x)$ :  $x$ 是要死的,  $a$ 是苏格拉底

前提:  $(\forall x)(F(x) \rightarrow G(x))$ ,  $F(a)$ , 结论:  $G(a)$

证明:

(1)  $(\forall x)(F(x) \rightarrow G(x))$

(2)  $F(a) \rightarrow G(a)$  (1的全称量词消去)

(3)  $F(a)$

(4)  $G(a)$  (2和3的假言推理)

# 自然语言的形式化

例：前提：1) 每驾飞机或者停在地面或者飞在天空；2) 并非每驾飞机都飞在天空

结论：有些飞机停在地面

形式化： $\text{plane}(x)$ ： $x$ 是飞机；  $\text{in\_ground}(x)$ ： $x$ 停在地面；  $\text{on\_fly}(x)$ ： $x$ 飞在天空

已知： $(\forall x) (\text{plane}(x) \rightarrow \text{in\_ground}(x) \vee \text{on\_fly}(x))$ ,  $(\neg \forall x) (\text{plane}(x) \rightarrow \text{on\_fly}(x))$

请证明： $(\exists x) (\text{plane}(x) \wedge \text{in\_ground}(x))$

证明：

1.  $(\neg \forall x) (\text{plane}(x) \rightarrow \text{on\_fly}(x))$  (已知)
2.  $(\exists x) \neg (\text{plane}(x) \rightarrow \text{on\_fly}(x))$
3.  $(\exists x) \neg (\neg \text{plane}(x) \vee \text{on\_fly}(x))$
4.  $(\exists x) (\text{plane}(x) \wedge \neg \text{on\_fly}(x))$

# 自然语言的形式化

1.  $(\neg \forall x)(plane(x) \rightarrow on\_fly(x))$
2.  $(\exists x)\neg(plane(x) \rightarrow on\_fly(x))$
3.  $(\exists x)\neg(\neg plane(x) \vee on\_fly(x))$
4.  $(\exists x)(plane(x) \wedge \neg on\_fly(x))$
5.  $plane(a) \wedge \neg on\_fly(a)$
6.  $plane(a)$
7.  $\neg on\_fly(a)$
8.  $(\forall x)(plane(x) \rightarrow in\_ground(x) \vee on\_fly(x))$
9.  $plane(a) \rightarrow in\_ground(a) \vee on\_fly(a)$
10.  $in\_ground(a) \vee on\_fly(a)$
11.  $in\_ground(a)$
12.  $plane(a) \wedge in\_ground(a)$
13.  $(\exists x)(plane(x) \wedge in\_ground(x))$

# 逻辑

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief
Fuzzy logic	facts + degree of truth	known interval value

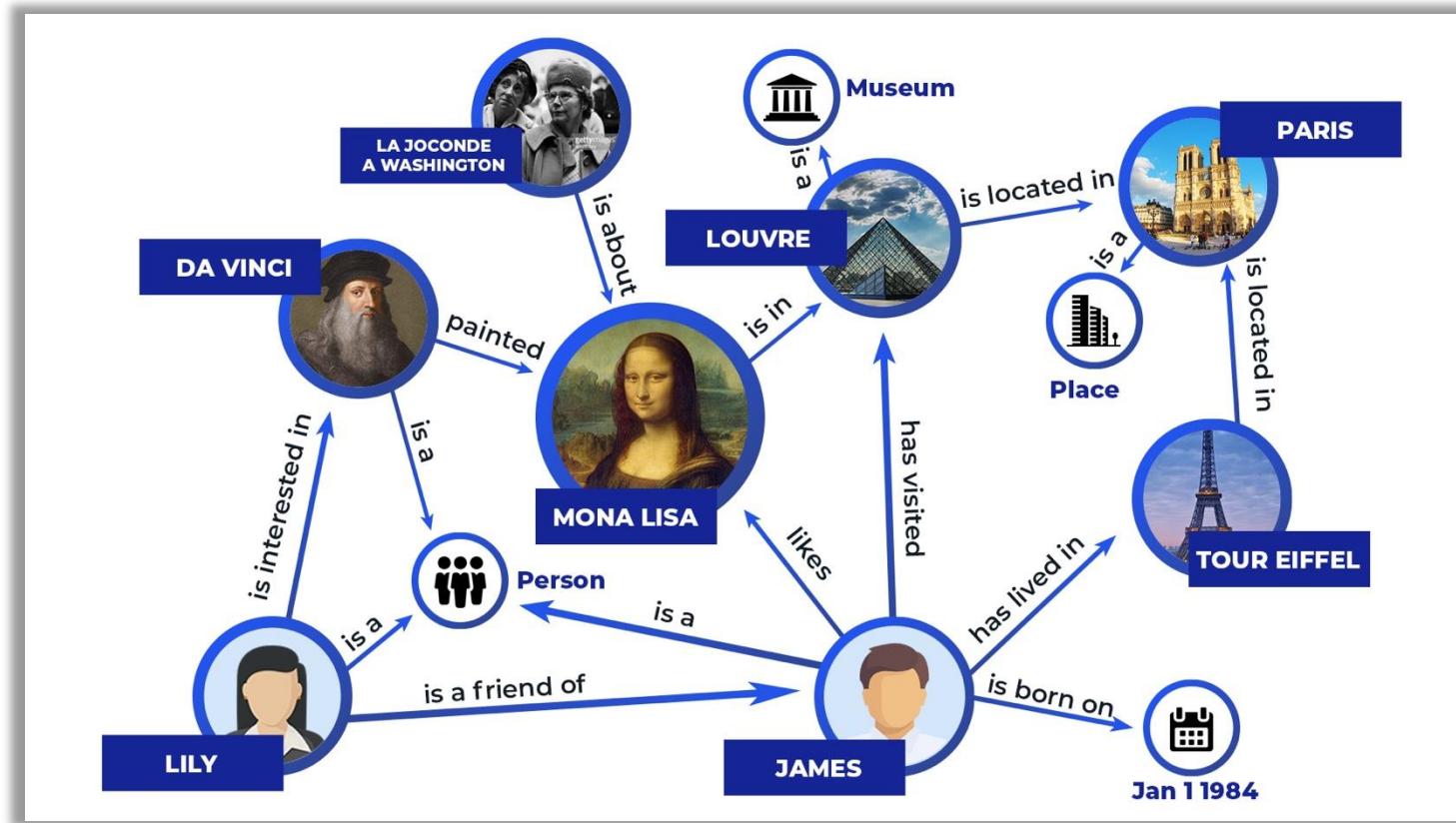
# 大纲

---

## 知识图谱

# 知识图谱

知识图谱是一种以图结构来描述知识和建模事物之间关联关系的技术方法



知识图谱于 2012 年由 Google 提出，其初衷是利用网络多源数据构建的知识库来增强语义搜索，提升搜索质量和体验

知识图谱已被广泛应用于智能搜索、智能问答、个性化推荐、内容分发等领域

# 从语义搜索认识知识图谱

## 李奧納多·達文西

译者：



 [更多图片](#)

李奧納多·達文西，又譯達芬奇，全名李奧納多·迪·塞爾·皮耶羅·達文西，是義大利文藝復興時期佛羅倫斯共和國的博學者：在繪畫、音樂、建築、數學、幾何學、解剖學、生理學、動物學、植物學、天文學、氣象學、地質學、地理學、物理學、光學、力學、發明、土木工程等領域都有顯著的成就。[维基百科](#)

出生信息：1452年4月15日，義大利Anchiano

逝世于：1519年5月2日，法國昂布瓦斯克勞斯·呂斯城堡

系列作品：[聖母像](#), [麗達與天鵝](#)

建筑作品：[Leonardo's Rivellino](#)

画风：[文藝復興](#), [義大利文藝復興](#), [文藝復興全盛期](#)

祖先：[瑟皮耶羅·達文西](#), [卡特琳娜·達文西](#), [安東尼奧·達溫奇](#), [Lucia di ser Piero di Zoso da Bacchereto](#)

## 艺术作品



蒙娜麗莎  
1503 年

最後的晚餐  
1498 年

救世主  
1500 年

維特魯威人

## 展出地点



Royal Collection...  
Museo Galileo  
羅浮宮  
國家藝廊  
佛羅倫斯  
巴黎  
華盛頓哥倫比亞特區

## 用户还搜索了



米開朗基羅  
拉斐爾  
李奧納多·狄卡皮歐  
文森·梵谷



[查看照片](#)

## 南京大學

網站 行車路線 儲存 致電

中国南京市的大学

南京大學，簡稱南大，位於中國江蘇省南京市，該校歷史或可追溯至三國吳永安元年，歷史上曾歷經多次變遷，亦是中國第一所集教學和研究於一體的現代大學。中華民國政府撤離南京後，中華人民共和國成立前夕由「國立中央大學」易名「國立南京大學」，翌年逕稱「南京大學」，沿用至今。[维基百科](#)

地址：中国江苏省南京市鼓楼区 邮政编码: 210093

电话号码：+86 25 8359 3186

总裁：[Chen Jun](#)

招生人数：35,434 (2007 年)

创立于：1902 年

子公司：[南京大學物理學院](#), [更多](#)

省份：[江苏省](#)

[免责声明](#)

## 近期活动

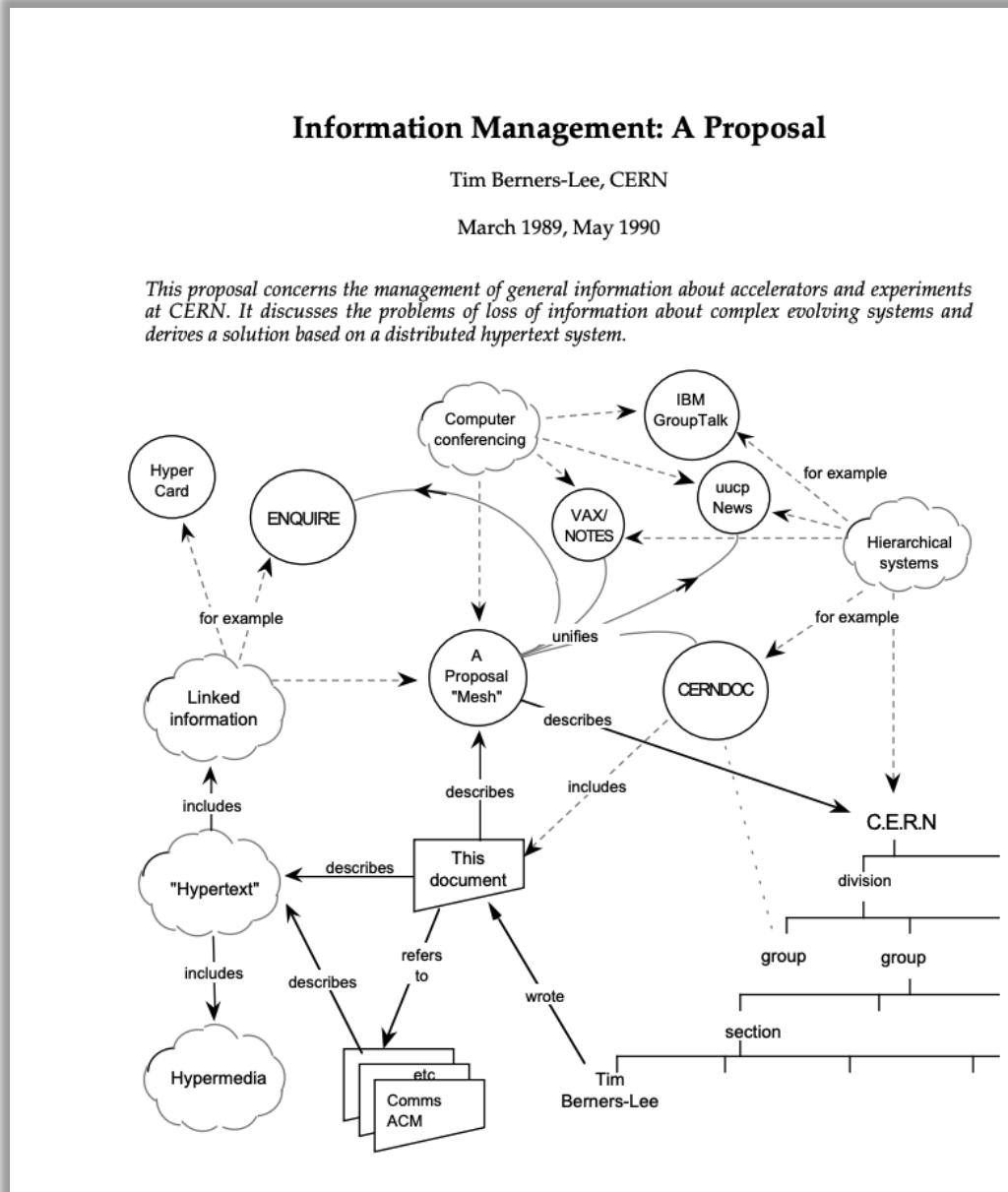
10月20日 週五 International Conference on Integrated Circ...

## 知名校友

[趙忠堯](#) [李國鼎](#) [張翔](#) [Jingguang Chen](#)

[查看更多項目 \(超過 45 項\)](#)

# 万维网



万维网 (world wide web) 是以链接为中心的信息系统  
Linked information system

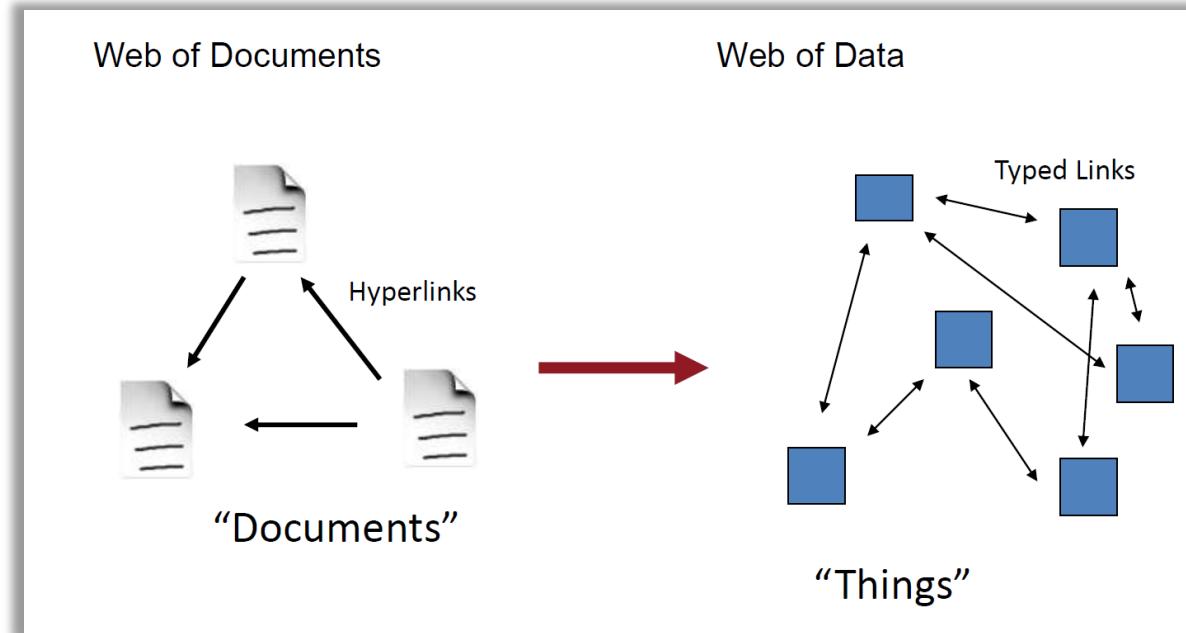
万维网以**文本**和**链接**描述信息，用户通过超链接浏览互联网上的各类资源，也可以通过互联网将自己的信息发布出去

Tim Berners-Lee  
2016年图灵奖



# 语义万维网

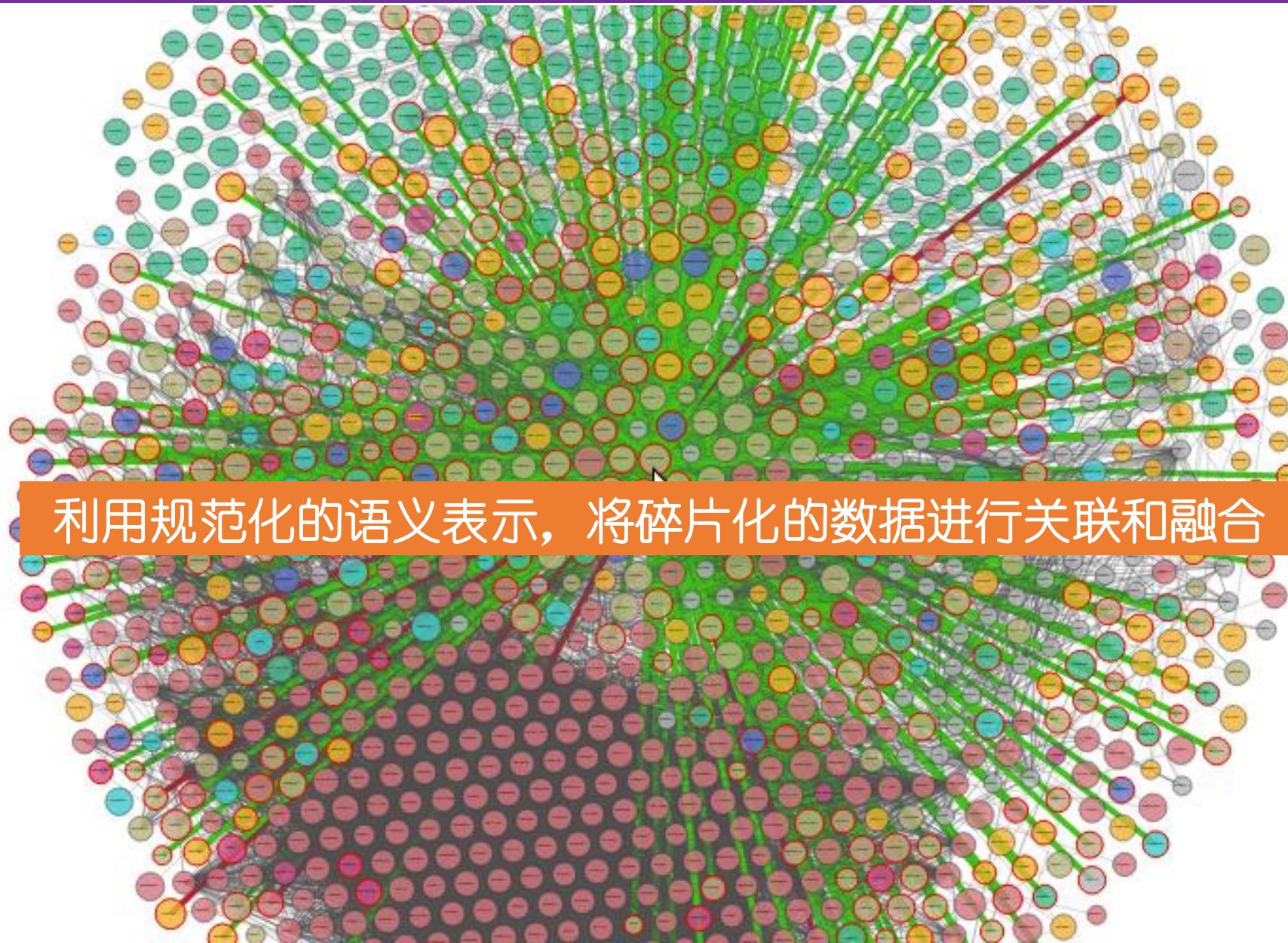
语义万维网 (semantic web) 中，信息内容具有良好的语义定义，计算机可以理解并自动存取语义信息



## 语义网与万维网的主要区别

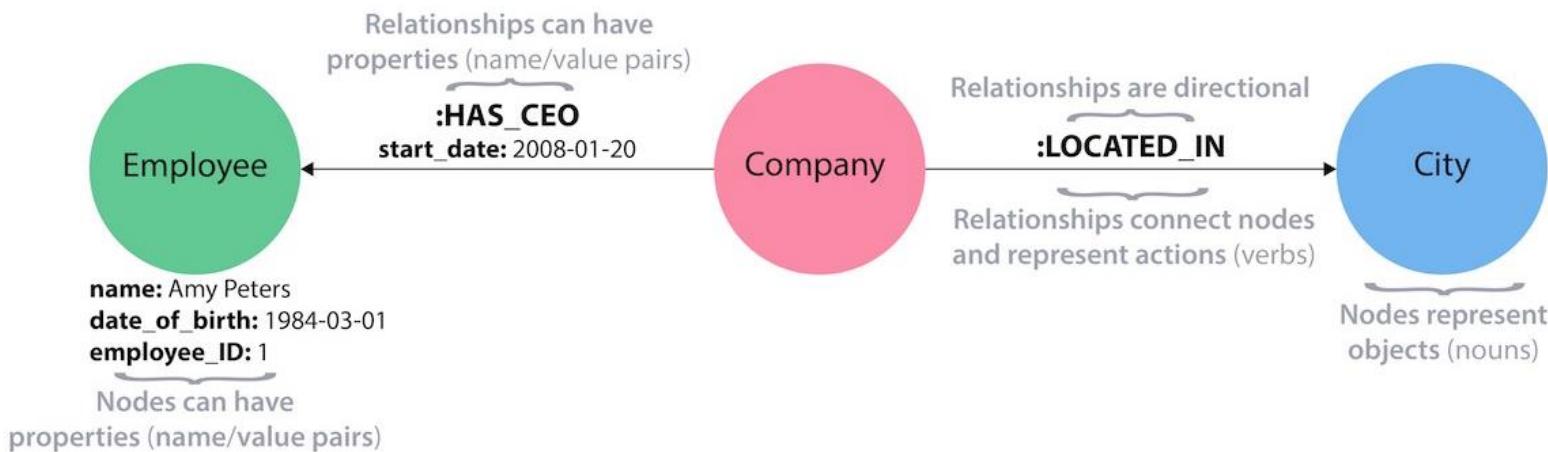
- ✓ 万维网是关于网页链接的结构，节点是网页，节点之间通过超链接连接
- ✓ 语义网是关于语义内容结构化表示的图结构，节点是语义信息

# Linked Data



# 属性图(property graph)

- 属性图由节点(Nodes)、关系(Relationships)、属性(Property)和标签(Labels)组成
- 节点上包含属性，属性可以以任何键值形式存在
- 关系连接节点，每个关系都有拥有一个方向、一个标签、一个开始节点和结束节点
- 关系也可以有属性，即边属性，可以通过在关系上增加属性提供有关边的元信息，如创建时间

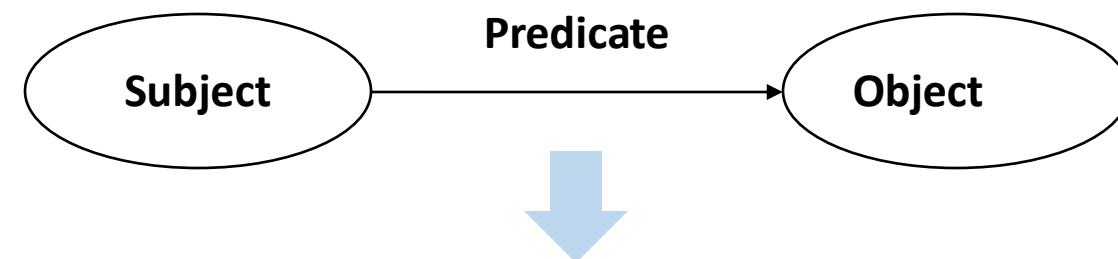


# 资源描述框架RDF

RDF表示 Resource Description Framework (资源描述框架)

国际万维网联盟W3C推动的面向Web的语义标准

RDF的基本组成单位是三元组 SPO

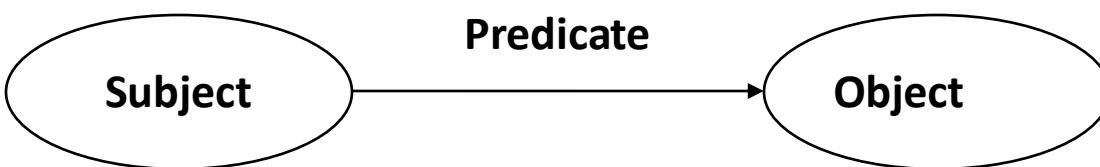


(subject (主) , predicate (谓词) , object (宾语) )

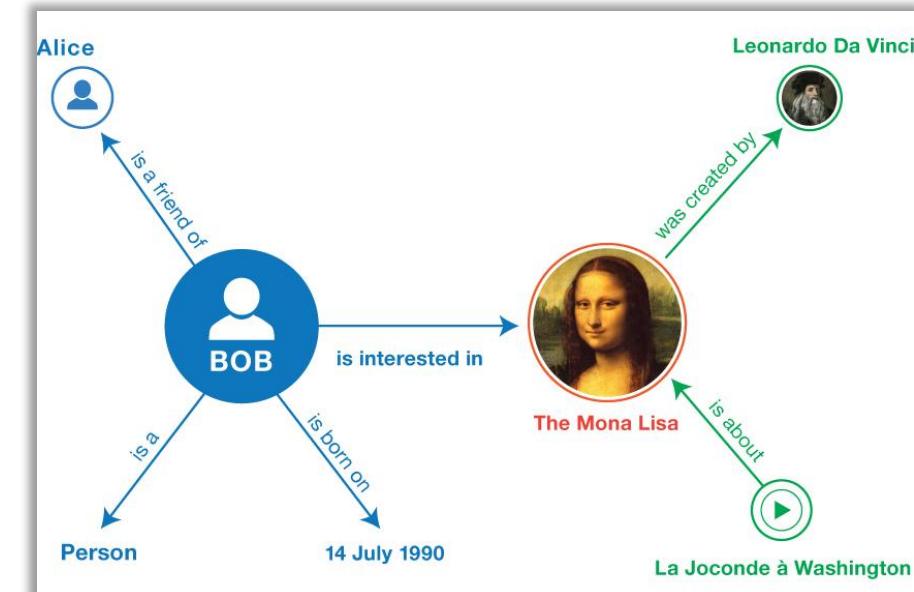


(subject (南京大学) , predicate (位于) , object (江苏) )

## 基本数据模型：有向标记图



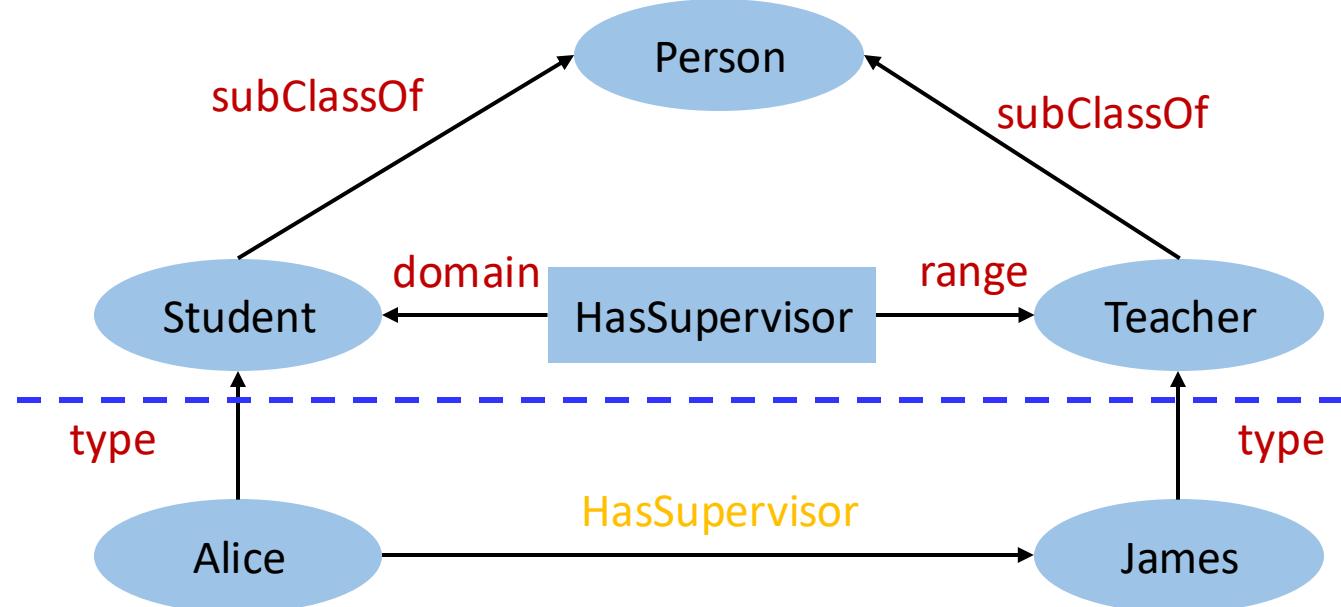
<Bob> <is a> <person>  
<Bob> <is a friend of> <Alice>  
<Bob> <is born on> <the 4th of July 1990>  
<Bob> <is interested in> <the Mona Lisa>  
<the Mona Lisa> <was created by> <Leonardo da Vinci>



# RDF Schema

RDFS(Resource Description Framework Schema)是用来定义RDF中的类和属性语义的描述性语言，定义了资源间的继承关系及属性约束等

1. rdfs:Class. 用于定义类
2. rdfs:domain. 用于表示该属性属于哪个类别
3. rdfs:range. 用于描述该属性的取值类型
4. rdfs:subClassOf. 用于描述该类的父类
5. rdfs:subProperty. 用于描述该属性的父属性



# 基于RDFS的简单推理



# OWL网络本体语言

## □ 本体 (Ontology)

本体是研究“存在”的科学，即试图解释存在是什么、世间所有存在的共同特征是什么

## □ 计算机领域的本体

在计算机领域，本体是指一种“形式化的，对于共享概念体系的明确且详细的说明”

## □ 网络本体语言 (Web Ontology Language, OWL)

OWL是对RDFS关于描述资源词汇的一个扩展，OWL中添加了额外的预定于词汇来描述资源，具备更好的语义表达能力

# OWL vs. RDF Schema

RDF Scheme: Class, subclass, Property, subProperty.....



- **Complex Classes:** intersection, union and complement
- **Property Restrictions:** existential quantification, universal quantification, hasValue
- **Cardinality Restrictions:** MaxQualifiedCardinality, minQualifiedCardinality
- **Property Characteristics:** inverseOf, SymmetricProperty, AsymmetricProperty, propertyDisjointWith, ReflexiveProperty, FunctionalProperty
- ...

# OWL网络本体语言

## □ 等价性声明

exp: 运动员 owl:equivalentClass exp: 体育选手

exp: 获得 owl:equivalentProperty exp: 取得

exp: 运动员A owl:sameIndividualAs exp: 小明

exp是命名空间 <http://www.example.org>的别称

## □ 传递性声明

exp: ancestor rdf:type owl: TransitiveProperty

exp: ancestor是一个传递关系

例如： exp: 小明 exp: ancestor exp: 小林；

exp: 小林 exp: ancestor exp: 小红

推理得出: exp: 小明 exp: ancestor exp: 小红

# OWL网络本体语言

## □ 声明属性的对称性

exp: friend rdf:type owl:SymmetricProperty

exp: friend具有对称性

例如： exp: 小明 exp: friend exp: 小林，

推理得出： exp: 小林 exp: friend exp: 小明

## □ 声明属性的函数性

exp: hasMother rdf: type owl: FunctionalProperty

exp: hasMother是一个具有函数性的属性，

因为每个人只有一个母亲，作为约束作用到知识库

# OWL网络本体语言

## □ 声明两个属性互反

exp: ancestor **owl: inverseOf** exp: descendant

exp: ancestor 和 exp: descendant是互反关系

例如： exp: 小明 exp: ancestor exp: 小林，

推理得出： exp: 小林 exp: descendant exp: 小明

## □ 声明属性的局部约束：

### 全称限定

exp: Person **owl: allValuesFrom** exp: Women

exp: Person **owl: onProperty** exp: hasMother

例如： exp:hasMother在主语属于exp:Person这个类

时， 宾语的取值只能来自exp:Women这个类

# OWL网络本体语言

## □ 声明属性的局部约束：

存在限定

```
exp:SemanticWebPaper owl:someValuesFrom exp:AAAI  
exp:SemanticWebPaper owl:onProperty exp:publishedIn
```

例如：exp:publishedIn在主语属于exp:SemanticWebPaper类时，宾语的取值部分来自exp:AAAI这个类。即，关于语义网的论文部分发表在AAAI中

## □ 声明属性的局部约束：

基数限定

```
exp:Person owl:cardinality "1"  
exp:Person owl:onProperty exp:hasMother
```

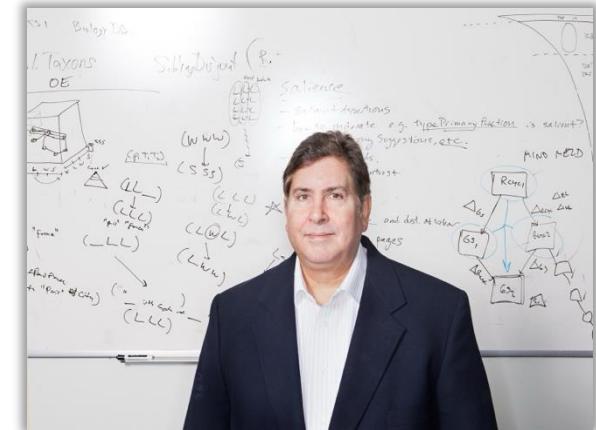
例如：exp:hasMother在主语属于exp:Person这个类时，宾语的取值只能有一个，本质上属于属性的局部约束

# OWL网络本体语言

OWL中的其他词汇	描述
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，声明所属类必有一个取值

# 典型知识图谱项目：CYC

- 取自英文单词“百科全书”(encyclopedia)，目标是把人类的常识编码，建成知识库
  - 典型的常识知识如“Every tree is a Plant”，“People die eventually”等
  - Cyc知识库主要由术语Terms和断言Assertions组成
  - Cyc目前有两个版本，企业版和研究版，研究版对研究人员开放



雷纳特 (1950-)

# 典型知识图谱项目：WordNet

- WordNet是著名的词典知识库，主要用于词义消歧，由普林斯顿大学认知科学实验室从1985年开始研发
- WordNet主要定义了名词、动词、形容词和副词之间的语义关系
- 例如：
  - ✓ 名词之间的上下位关系（如：“犬科动物”是“狗”的上位名词）
  - ✓ 动词之间的蕴含关系，（如：“打鼾”蕴含着“睡眠”）
- WordNet3.0已经包含超过15万个词和20万个语义关系



<https://wordnet.princeton.edu/>

# 典型知识图谱项目：ConceptNet

- ConceptNet是常识知识库。最早源于MIT媒体实验室的Open Mind Common Sense (OMCS) 项目，OMCS项目是由著名人工智能专家Marvin Minsky于1999年提议创立
- ConceptNet主要依靠互联网众包、专家创建来构建，新版本也导入了大量开放的结构化数据，如Dbpedia、Wikinary、Wordnet等
- ConceptNet以三元组形式的关系型知识组成，ConceptNet5版本中已经包含由2800万关系描述
- 与谷歌知识图谱比，ConceptNet侧重词与词之间的关系，更加接近于WordNet，但又比WordNet包含的关系类型更多
- ConceptNet完全免费开放、支持多种语言



# 典型知识图谱项目：Freebase



截至2014年年底，Freebase共包含6800万个实体、10亿条关系、超过24亿事实三元组信息

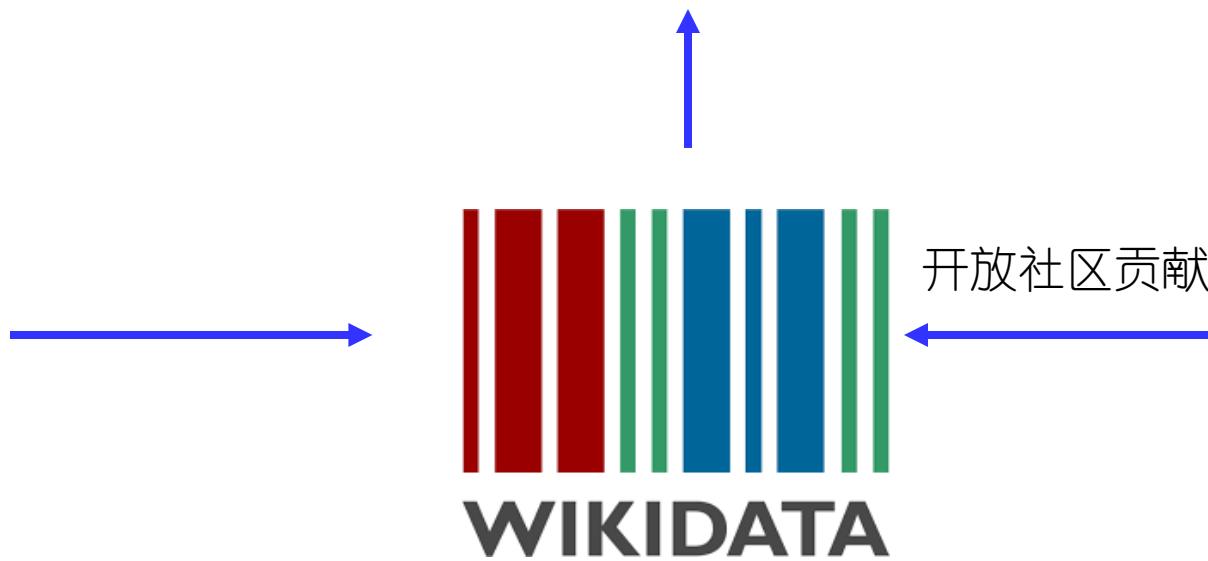
2016年谷歌对Freebase停止更新，把所有数据捐给WikiData

# 典型知识图谱项目：WikiData

目标是构建全世界最大的免费知识库，采用CC0完全自由许可协议



维基基金会



[https://www.wikidata.org/wiki/Wikidata:Main\\_Page](https://www.wikidata.org/wiki/Wikidata:Main_Page)

# 典型知识图谱项目：BabelNet

- BabelNet是类似于WordNet的多语言词典知识库，目标是解决WordNet在非英语语种中数据缺乏的问题，采用的方法是将WordNet词典与Wikipedia百科集成
- 首先建立WordNet中的词与Wikipedia的页面标题的映射，然后利用Wikipedia中的多语言链接，再辅以机器翻译技术，来给WordNet增加多种语言的词汇
- BabelNet包含了271中语言，1400万同义词组，36.4万词语关系和3.8亿从Wikipedia中抽取的链接关系，集成了WordNet在词语关系上的优势和Wikipedia在多语言方面的优势，是目前最大的多语言词典知识库

The screenshot shows a BabelNet entity page for 'Nanjing University'. At the top, there's a search bar with '南京大学' and dropdown menus for 'Chinese' and '翻译为...'. Below the header, the entity ID 'bn:02439362n' is shown along with labels '名词' and '命名实体'. A small thumbnail image of a building is followed by a brief description in Chinese: '南京大學，簡稱南大，是位於中國南京市的一個普通高等學校，該校歷史或可追溯至三國吳永安元年（258年），歷史上曾歷經多次變遷，亦是中國第一所集教學和研究於一體的現代大學。' with a link to 'Wikipedia'. Below this, tabs for '定义', '关联', and '来源' are visible, along with a language selector 'Chinese > 更多语言'. Further down, there are sections for 'Disambiguation' and another link to 'Wikidata'. A large sidebar on the right contains a circular network graph icon with the text 'EXPLORE NETWORK'.

# 典型知识图谱项目：NELL

Never-Ending Language Learner (NELL)

永不停歇的语言学习者



- NELL是卡内基梅隆大学开发的知识库，采用互联网挖掘的方法从Web自动抽取三元组知识
- 基本理念：给定一个初始的本体（少量类和关系的定义）和少量样本，让机器能够通过自学习的方式不断的从Web学习和抽取新的知识
- 目前NELL已经抽取了500多万条三元组知识

# 典型知识图谱项目：YAGO

- YAGO是德国马克斯-普朗克研究所 (Max Planck Institute, MPI) 构建的大型多语言知识库。
- YAGO主要继承了Wikipedia、WordNet和GeoNames三个来源的数据。YAGO将WordNet的词汇定义与Wikipedia的分类体系进行了融合，使其具有更加丰富的实体分类体系
- YAGO考虑了时间和空间知识，为很多知识条目增加了时间和空间维度的属性描述



<https://yago-knowledge.org/>

# 中文领域开放知识图谱

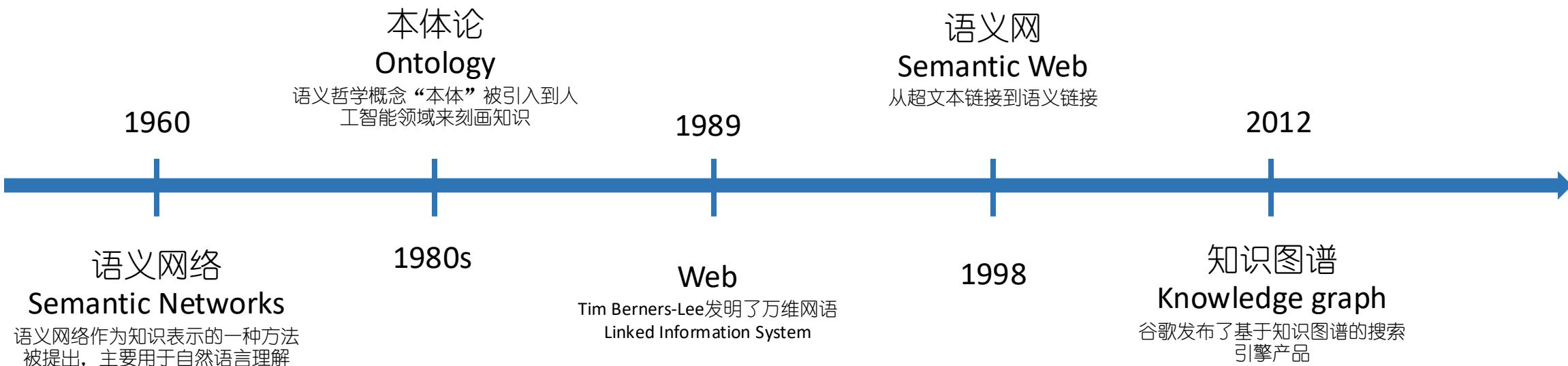
<http://www.openkg.cn/>



## 资源分类

常识	城市	金融	农业	地理
气象	社交	物联网	医疗	娱乐
生活	商业	出行	科教	...
其他				

# 知识图谱相关概念的演变



# 大纲

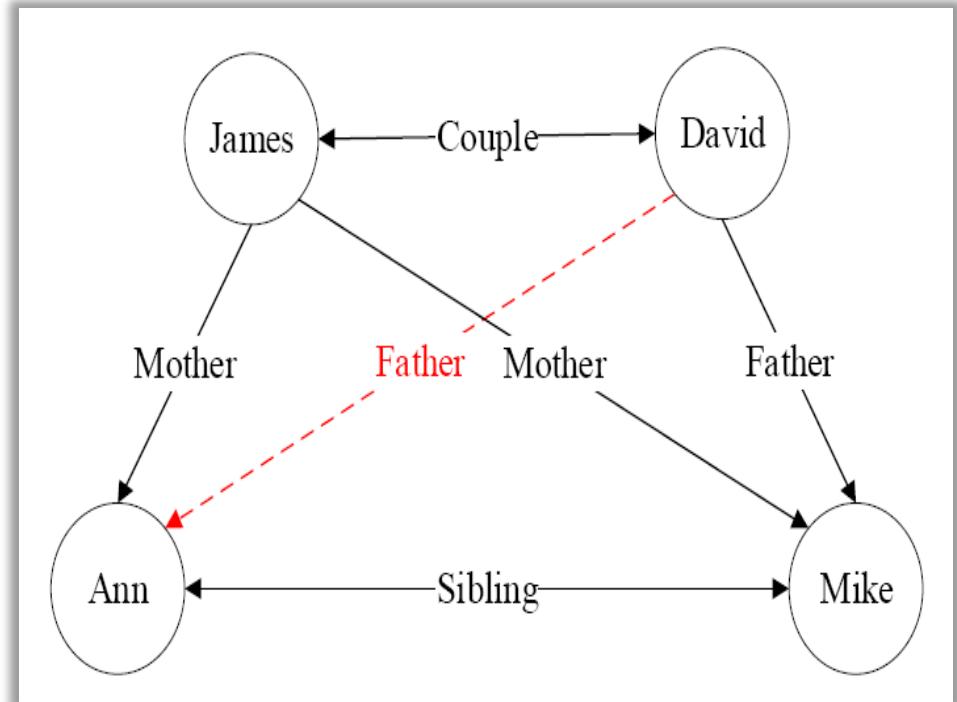
---

## 知识图谱推理

# 知识图谱推理

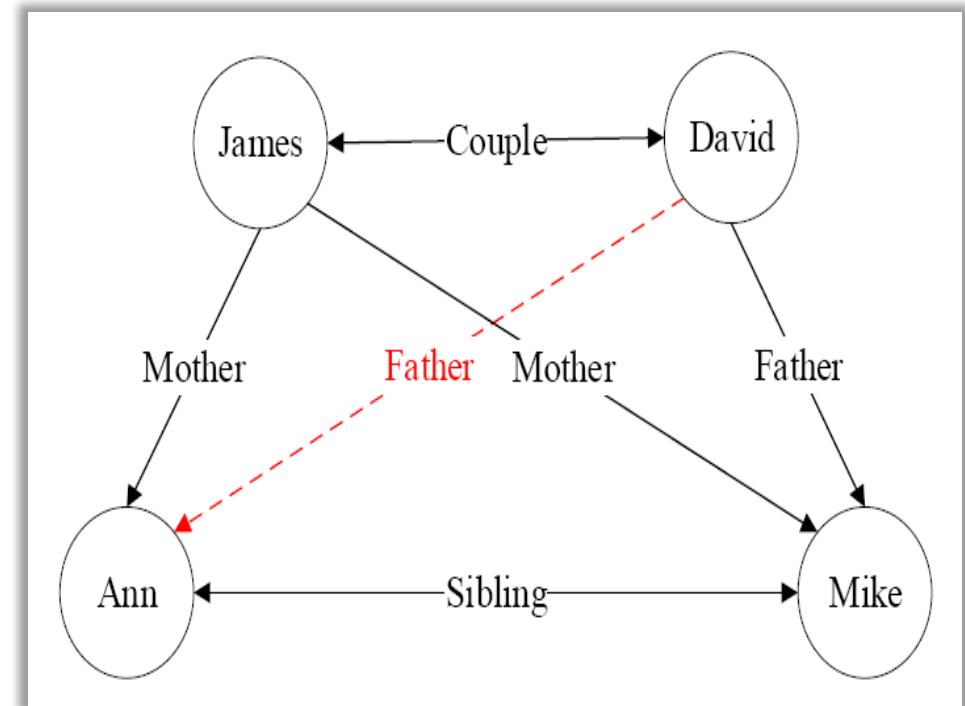
□ 知识图谱由有向图构成，每个节点是一个实体（如人名、地名、事件和活动等），任意两个节点之间的边表示这两个节点之间存在的关系，**知识图谱推理一般指的是关系推理。**

□ 一般而言，可将知识图谱中任意两个相连节点及其连接边表示成一个三元组 (*triplet*)，即  
*<left\_node, relation, right\_node>*，  
例如：*<David, Father, Mike>*



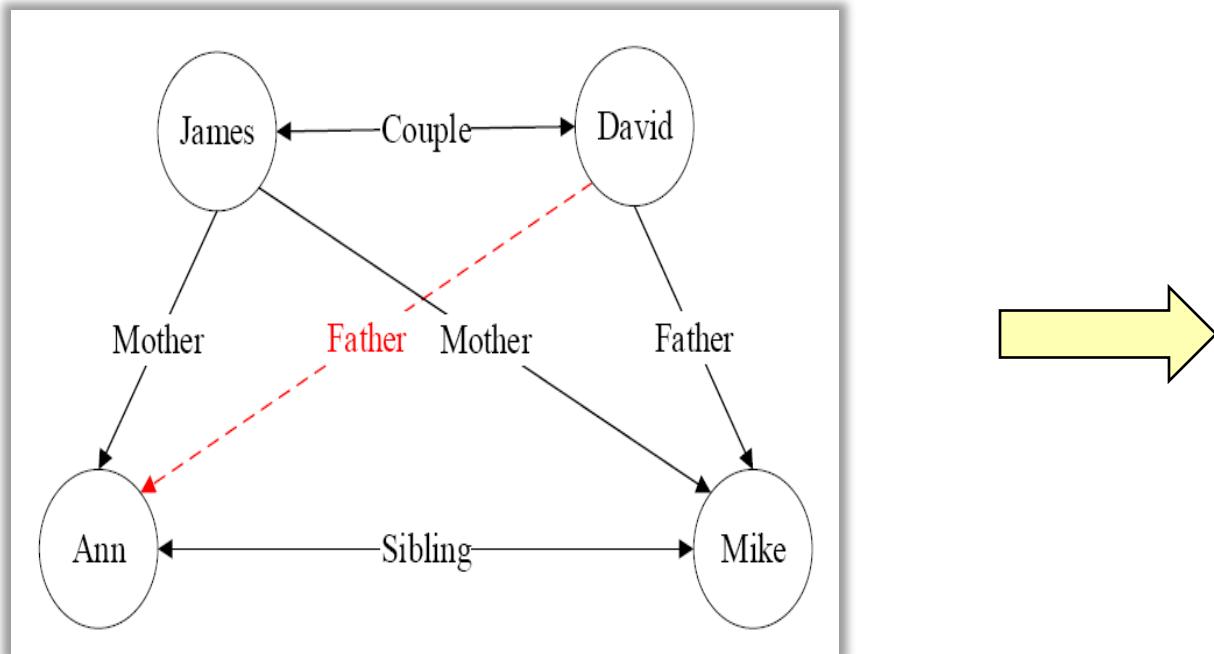
# 知识图谱推理

- 知识图谱中节点之间存在的关系可以用一阶逻辑 (first order logic, FOL) 的形式来描述
- 如图中形如 $\langle James, Couple, David \rangle$ 的关系可用  $Couple(x, y)$  来描述, 即  $Couple(James, David)$
- $Couple(x, y)$ 是一阶谓词,  $Couple$ 表示图中实体之间具有的关系,  $x$ 和 $y$ 是谓词变量



# 知识图谱推理

**关系推理：**如何基于实体中已有的关系推断出未知的关系，从现有知识中发现新的知识，从而扩充和丰富现有知识库



*father(David, Ann)*

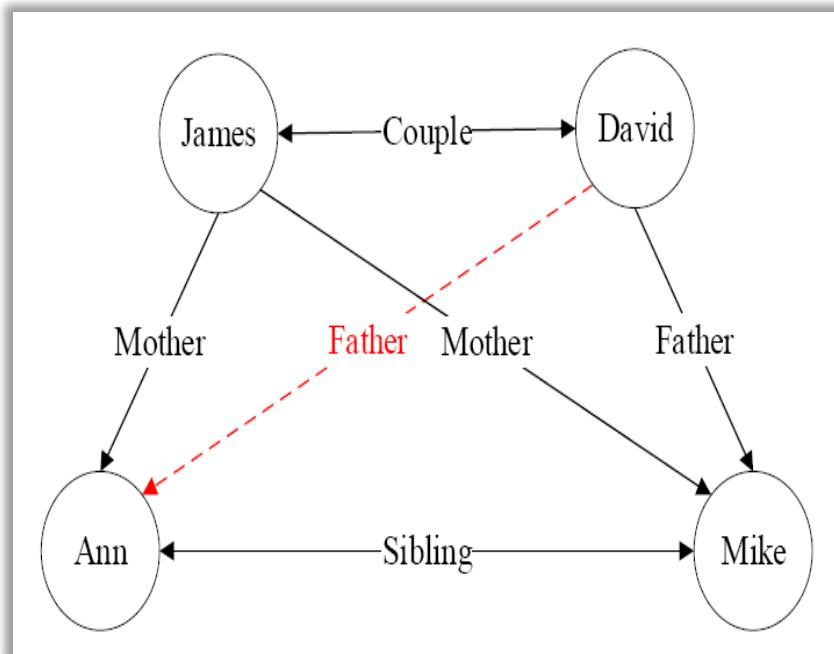
如何从已有知识中推理得到？

# 知识图谱推理：归纳学习

- 归纳逻辑程序设计 (Inductive Logic Programming, ILP)是机器学习和逻辑程序设计交叉领域的研究内容
- ILP使用一阶谓词逻辑进行知识表示，通过修改和扩充逻辑表达式对现有知识归纳，完成推理任务
- 作为ILP的代表性方法，FOIL (First Order Inductive Learner) 通过序贯覆盖 (sequential covering) 实现规则推理
  - 序贯覆盖：即逐条归纳，在训练集上每学到一条规则，就将该规则覆盖的训练样本去除，然后以剩下的样本重复上述规程

# FOIL (First Order Inductive Learner)

- FOIL: 给定目标谓词 $P$ 、目标谓词 $P$ 的训练样例（正例 $E^+$ 和反例 $E^-$ ），以及其他背景知识样例，得到可推理出目标谓词 $P$ 这一结论的推理规则



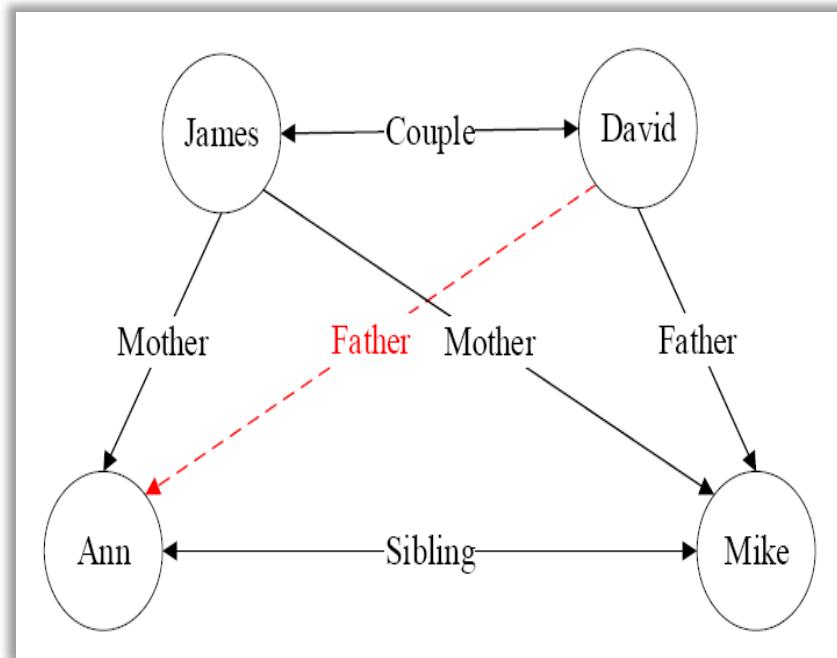
假设目标谓词 $P$ 是 $Father(x, y)$

目标谓词是需要推断规则的结论，也称为规则头

在给定推理结论后，FOIL算法学习得到使得结论满足的前提条件，即目标谓词作为结论的推理规则

# FOIL (First Order Inductive Learner)

□ FOIL：给定目标谓词 $P$ 、目标谓词 $P$ 的训练样例（正例 $E^+$ 和反例 $E^-$ ），以及其他背景知识样例，得到可推理出目标谓词 $P$ 这一结论的推理规则

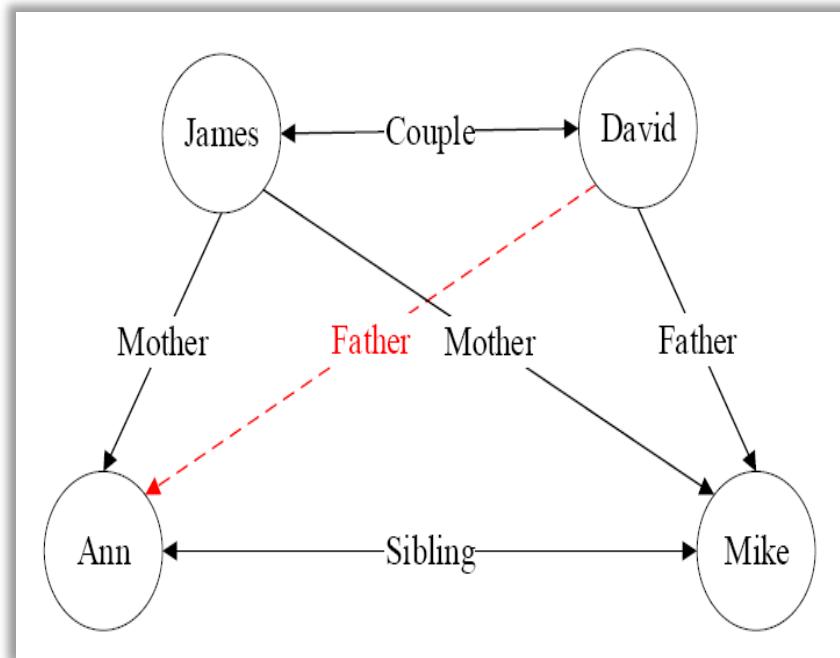


## 训练样例

- 目标谓词 $Father(x, y)$ 只有一个正例 $Father(David, Mike)$
- 反例一般不会显式给出，但可从知识图谱中构造出来，如：从知识图谱中已经知道 $Couple(David, James)$ 成立，则 $Father(David, James)$ 可作为目标谓词 $P$ 的一个反例，记为 $\neg Father(David, James)$

# FOIL (First Order Inductive Learner)

- FOIL：给定目标谓词 $P$ 、 $P$ 的训练样例（正例 $E^+$ 和反例 $E^-$ ），以及其他背景知识样例，得到可推理出目标谓词 $P$ 这一结论的推理规则



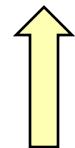
## 背景知识样例

- 背景知识：知识图谱中目标谓词以外的其他谓词实例化结果
- 如： $Sibling(Ann, Mike)$ 、 $Couple(David, James)$ 等

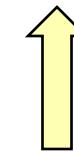
# FOIL (First Order Inductive Learner)

- FOIL：给定目标谓词  $P$ 、目标谓词  $P$  的训练样例（正例  $E^+$  和反例  $E^-$ ），以及其他背景知识样例，得到可推理出目标谓词  $P$  这一结论的推理规则

$$(\forall x)(\forall y)(\forall z)(\text{Mother}(z, y) \wedge \text{Couple}(x, z) \rightarrow \text{Father}(x, y))$$



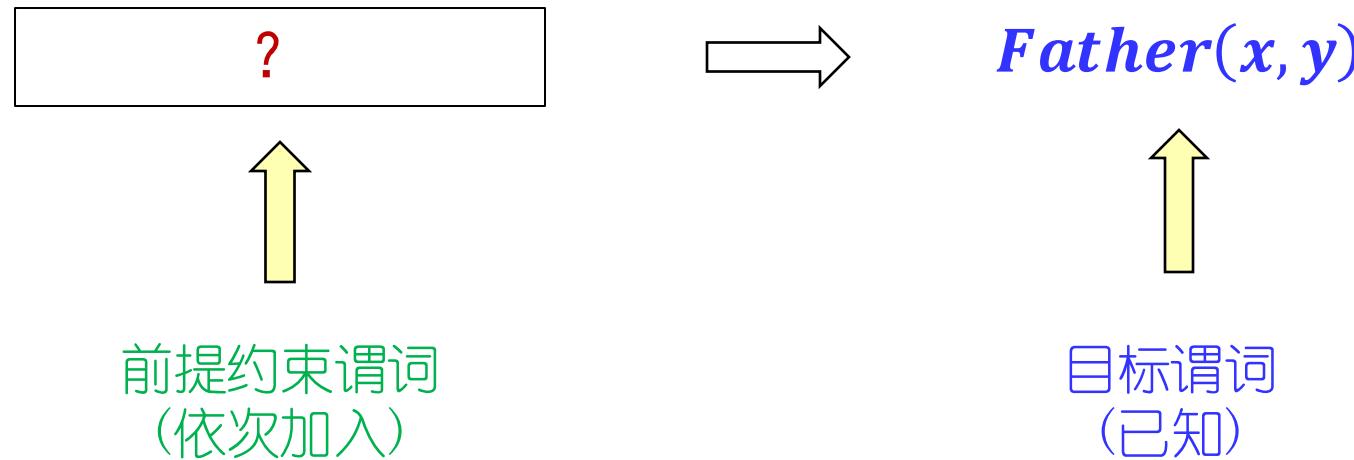
前提约束谓词  
(学习得到)



目标谓词  
(已知)

positive examples + negative examples + background knowledge examples  $\Rightarrow$  hypothesis

# FOIL (First Order Inductive Learner)



如何选择谓词？哪些谓词  
作为目标谓词的前提约束  
谓词更好？

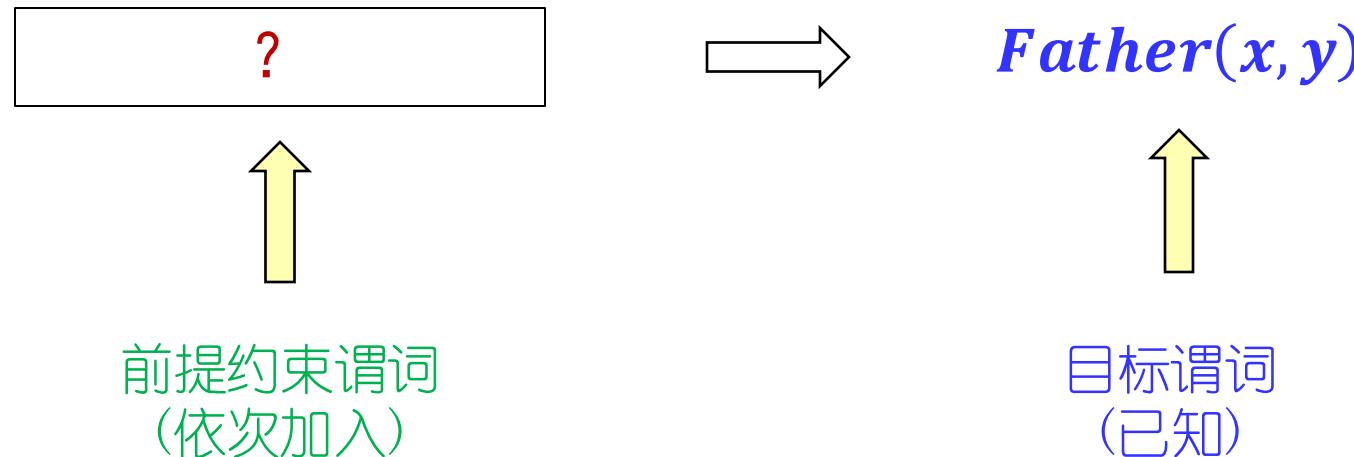
**FOIL中信息增益值  
(information gain)**

FOIL信息增益值计算方法如下：

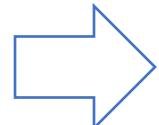
$$FOIL\_Gain = \widehat{m}_+ \cdot \left( \log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

其中， $\widehat{m}_+$ 和 $\widehat{m}_-$ 是增加前提约束谓词后所得新推理规则覆盖的正例和反例的数量， $m_+$ 和 $m_-$ 是原推理规则所覆盖的正例和反例数量。

# FOIL (First Order Inductive Learner)



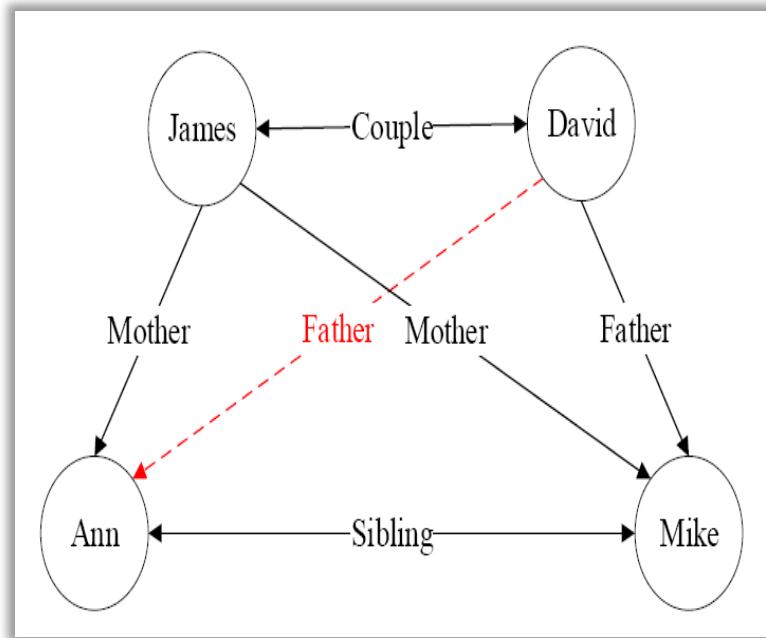
- $Mother(\cdot, \cdot)$
- $Sibling(\cdot, \cdot)$
- $Couple(\cdot, \cdot)$
- ...



- 依次将谓词加入到推理规则中作为前提约束谓词
- 计算所得到新推理规则的FOIL增益值
- 基于计算所得FOIL增益值来选择最佳前提约束谓词

# FOIL (First Order Inductive Learner)

目标谓词：**Father(x, y)**



## 目标谓词训练样例集合

Father(David, Mike)  
¬Father(David, James)  
¬Father(James, Ann)  
¬Father(James, Mike)  
¬Father(Ann, Mike)

## 背景知识样例集合

Sibling(Ann, Mike)  
Couple(David, James)  
Mother(James, Ann)  
Mother(James, Mike)

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集			

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	

给定目标谓词，此时推理规则只有目标谓词，因此推理规则所覆盖的正例和反例的样本数分别是训练样本中正例和反例的数量，即1和4，因此， $m_+ = 1$ ,  $m_- = 4$ 。

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$			

- 将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则： $\textcolor{blue}{Mother(x, y) \rightarrow Father(x, y)}$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$			

- 将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则： $\textcolor{blue}{Mother(x, y) \rightarrow Father(x, y)}$
- 在背景知识中， $Mother(x, y)$ 有两个实例： $Mother(\text{James}, \text{Ann})$ ,  $Mother(\text{James}, \text{Mike})$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$			

- 将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则： $\textcolor{blue}{Mother(x, y) \rightarrow Father(x, y)}$
- 在背景知识中， $Mother(x, y)$ 有两个实例： $Mother(\text{James}, \text{Ann})$ ,  $Mother(\text{James}, \text{Mike})$
- 对于 $Mother(\text{James}, \text{Ann})$ 这一实例， $x = \text{James}$ ,  $y = \text{Ann}$ , 将 $x$ 和 $y$ 代入 $Father(x, y)$ 得到 $Father(\text{James}, \text{Ann})$ , 可知在训练样本中 $Father(\text{James}, \text{Ann})$ 是一个反例

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$			

- 将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则： $\textcolor{blue}{Mother(x, y) \rightarrow Father(x, y)}$
- 在背景知识中， $Mother(x, y)$ 有两个实例： $Mother(\text{James}, \text{Ann})$ ,  $Mother(\text{James}, \text{Mike})$
- 对于 $Mother(\text{James}, \text{Mike})$ 这一实例， $x = \text{James}$ ,  $y = \text{Mike}$ , 将 $x$ 和 $y$ 代入 $Father(x, y)$ 得到 $Father(\text{James}, \text{Mike})$ ，可知在训练样本中 $Father(\text{James}, \text{Mike})$ 是一个反例

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA

- 将 $Mother(x, y)$ 作为前提约束谓词加入，可得到推理规则： $\textcolor{blue}{Mother(x, y) \rightarrow Father(x, y)}$
- 在背景知识中， $Mother(x, y)$ 有两个实例： $Mother(\text{James}, \text{Ann})$ ,  $Mother(\text{James}, \text{Mike})$
- 对于 $Mother(\text{James}, \text{Mike})$ 这一实例， $x = \text{James}$ ,  $y = \text{Mike}$ , 将 $x$ 和 $y$ 代入 $Father(x, y)$ 得到 $Father(\text{James}, \text{Mike})$ ，可知在训练样本中 $Father(\text{James}, \text{Mike})$ 是一个反例

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(x, z)$			

- 如果将 $Couple(x, z)$ 作为前提约束谓词加入，可得到如下推理规则  $Couple(x, z) \rightarrow Father(x, y)$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(x, z)$			

- 如果将 $Couple(x, z)$ 作为前提约束谓词加入，可得到如下推理规则 $Couple(x, z) \rightarrow Father(x, y)$
- 在背景知识中， $Couple(x, z)$ 只有一个实例 $Couple(David, James)$ ，即 $x = David$ ,  $z = James$ ，将其代入 $Father(x, y)$ 得到 $Father(David, y)$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James) $\neg$ Father(James, Ann) $\neg$ Father(James, Mike) $\neg$ Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(x, z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	

- 在训练样本中存在正例 $Father(David, Mike)$ 以及反例 $\neg Father(David, James)$ , 即 $Couple(x, z) \rightarrow Father(x, y)$ 覆盖正例和反例数量分别为1和1。

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) ¬Father(David, James) ¬Father(James, Ann) ¬Father(James, Mike) ¬Father(Ann, Mike)
--------------	---	----------------	---

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(x, z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	1.32

- 信息增益值为：

$$\widehat{m}_+ \cdot \left( \log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

$$= 1 \cdot \left( \log_2 \frac{1}{1+1} - \log_2 \frac{1}{1+4} \right) = 1.32$$

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x, y) \leftarrow$	$Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 3$	0.32
	$Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Sibling(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 2$	0.74
	$Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$Couple(x, z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	1.32
	$Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 2$	NA
	$Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA

$Couple(x, z)$  加入后信息增益最大，将  $Couple(x, z)$  加入推理规则，  
得到新推理规则： **Couple(x, z) → Father(x, y)**

目标谓词	Father(David, Mike)
训练样例集合	$\neg$ Father(David, James)
	$\neg$ Father(James, Ann)
	$\neg$ Father(James, Mike)
	$\neg$ Father(Ann, Mike)

- 将训练样例中与该推理规则不符的样例去掉
- 这里不符指当  $Couple(x, z)$  中  $x$  取值为 David 时，与  $Father(David, )$  或  $\neg Father(David, )$  无法匹配的实例

$Couple(x, z)$  加入后信息增益最大，将  $Couple(x, z)$  加入推理规则，  
得到新推理规则： **Couple(x, z) → Father(x, y)**

目标谓词	$Father(David, Mike)$
训练样例集合	$\neg Father(David, James)$
	$\neg Father(James, Ann)$
	$\neg Father(James, Mike)$
	$\neg Father(Ann, Mike)$

- 将训练样例中与该推理规则不符的样例去掉
- 这里不符指当  $Couple(x, z)$  中  $x$  取值为 David 时，与  $Father(David, )$  或  $\neg Father(David, )$  无法匹配的实例

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James)
--------------	---	----------------	--

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow Couple(x, z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x, y)$ $\leftarrow Couple(x, z)$	$\wedge Mother(z, y)$			

- $\wedge Mother(z, y)$ 作为前提约束谓词加入，可得到推理规则

$$\textcolor{blue}{Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y)}$$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James)
--------------	---	----------------	--

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow Couple(x, z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x, y)$ $\leftarrow Couple(x, z)$	$\wedge Mother(z, y)$			

- 在背景知识中， $Mother(z, y) \wedge Couple(x, z)$ 有2个实例：
  - ✓  $Mother(James, Ann) \wedge Couple(David, James)$ , 即 $x = David, y = Ann, z = James$ , 将其代入 $Father(x, y)$ 得到 $Father(David, Ann)$
  - ✓  $Mother(James, Mike) \wedge Couple(David, James)$ , 即 $x = David, y = Mike, z = James$ , 将其代入 $Father(x, y)$ 得到 $Father(David, Mike)$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James)
--------------	---	----------------	--

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow Couple(x, z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x, y)$ $\leftarrow Couple(x, z)$	$\wedge Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 0$	

- 在背景知识中， $Mother(z, y) \wedge Couple(x, z)$ 有2个实例：
  - ✓  $Mother(James, Ann) \wedge Couple(David, James)$ , 即 $x = David, y = Ann, z = James$ , 将其代入 $Father(x, y)$ 得到 $Father(David, Ann)$
  - ✓  $Mother(James, Mike) \wedge Couple(David, James)$ , 即 $x = David, y = Mike, z = James$ , 将其代入 $Father(x, y)$ 得到 $Father(David, Mike)$

# FOIL

背景知识 样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词 训练样例集合	Father(David, Mike) $\neg$ Father(David, James)
--------------	---	----------------	--

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x, y) \leftarrow Couple(x, z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x, y)$ $\leftarrow Couple(x, z)$	$\wedge Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 0$	1

- 信息增益值为：

$$\widehat{m}_+ \cdot \left( \log_2 \frac{\widehat{m}_+}{\widehat{m}_+ + \widehat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-} \right)$$

$$= 1 \cdot \left( \log_2 \frac{1}{1+0} - \log_2 \frac{1}{1+1} \right) = 1$$

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
现有规则	拟加入前提约束谓词	正例	反例	信息增益值
	$Father(x, y) \leftarrow Couple(x, z)$	$m_+ = 1$	$m_- = 1$	1.32
$Father(x, y) \leftarrow Couple(x, z)$	$\wedge Mother(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Mother(z, y)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 0$	1
	$\wedge Sibling(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(x, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Sibling(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(x, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 1$	NA
	$\wedge Couple(x, z)$	$\widehat{m}_+ = 1$	$\widehat{m}_- = 1$	0
	$\wedge Couple(y, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(y, z)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(z, x)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA
	$\wedge Couple(z, y)$	$\widehat{m}_+ = 0$	$\widehat{m}_- = 0$	NA

# FOIL

$Mother(z, y)$ 加入信息增益最大，将 $Mother(z, y)$ 加入推理规则  
得到新推理规则： **$Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y)$**

目标谓词	$Father(David, Mike)$
训练样例集合	$\neg Father(David, James)$

将训练样例中与该推理规则不符的样例去掉  
(x=David、y=Mike、z =James)

# FOIL

$Mother(z, y)$ 加入信息增益最大，将 $Mother(z, y)$ 加入推理规则  
得到新推理规则： **$Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y)$**

目标谓词	$Father(David, Mike)$
训练样例集合	$\neg Father(David, James)$

将训练样例中与该推理规则不符的样例  
去掉 ( $x=David$ 、  $y=Mike$ 、  $z =James$ )

# FOIL

$Mother(z, y)$ 加入信息增益最大，将 $Mother(z, y)$ 加入推理规则  
得到新推理规则： **$Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y)$**

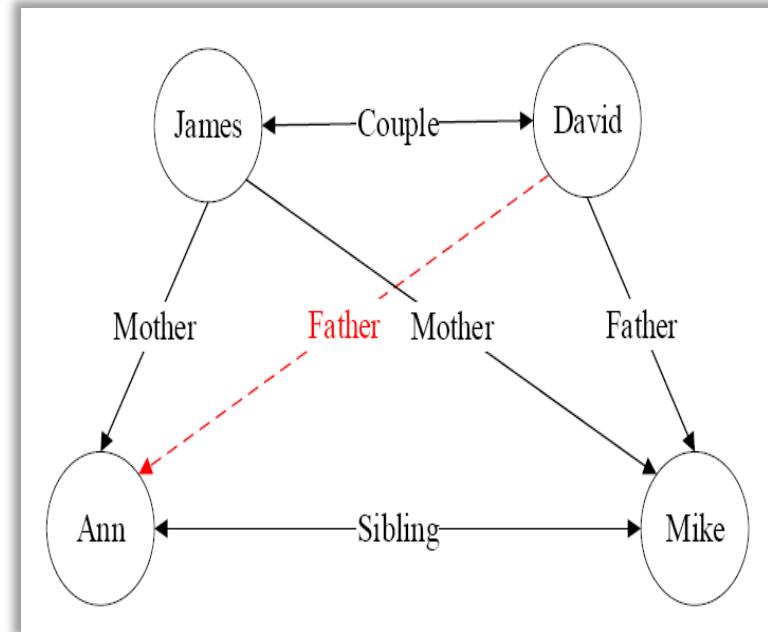
目标谓词	$Father(David, Mike)$
训练样例集合	$\neg Father(David, James)$

该推理规则覆盖训练样本集中正例  
 $Father(David, Mike)$ 且不覆盖任意反例  
因此算法学习结束

$Mother(z, y)$ 加入信息增益最大，将 $Mother(z, y)$ 加入推理规则

得到新推理规则： **$Mother(z, y) \wedge Couple(x, z) \rightarrow Father(x, y)$**

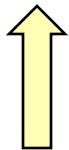
**$Mother(z, y) \wedge Couple(x, z)$**   
 $\rightarrow Father(x, y)$



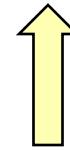
已知： $Mother(James, Ann)$ ,  $Couple(David, James)$   
 推理得到： $Father(David, Ann)$

# FOIL

$$(\forall x)(\forall y)(\forall z)(\text{Mother}(z, y) \wedge \text{Couple}(x, z) \rightarrow \text{Father}(x, y))$$



前提约束谓词  
(学习得到)



目标谓词  
(已知)

推理手段：positive examples + negative examples + background knowledge examples  $\Rightarrow$  hypothesis

给定目标谓词，FOIL算法从实例（正例、反例、背景样例）出发，不断测试所得到推理规则是否还包含反例，一旦不包含反例，则学习结束，展示了“**归纳学习**”能力，在学得推理规则后，再给推理规则中的变量赋予具体例子，经过“**演绎**”得到新的知识

## FOIL算法

- 输入：目标谓词 $P$ 、目标谓词 $P$ 的训练样例（正例集合 $E^+$ 和反例集合 $E^-$ ）以及其他背景知识样例
- 输出：可得到目标谓词 $P$ 这一结论的推理规则
  - (1) 将目标谓词作为所学习推理规则的结论；
  - (2) 将其他谓词逐一作为前提约束谓词加入推理规则，计算所得到推理规则的FOIL信息增益值，选取可带来最大信息增益值的前提约束谓词加入原来的推理规则，得到新的推理规则，并将训练样例集合中与该推理规则不符的样例去掉；
  - (3) 重复 (2) 过程，直到所得的推理规则不覆盖任何反例。

# 知识图谱推理

## 知识图谱推理的主要方法

- 基于逻辑规则的推理（如：FOIL、AMIE）
- 基于图结构和统计学习的推理（如：PRA）
- 基于分布式表示学习的推理（如：TransE）
- 基于神经网络的推理（如：R-GCN）
- ...