

<<语义Web技术基础>>

图书基本信息

书名：<<语义Web技术基础>>

13位ISBN编号：9787302289692

10位ISBN编号：7302289697

出版时间：2012-12

出版时间：清华大学出版社

作者：[德] 希茨利尔

页数：294

字数：491000

译者：俞勇

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<语义Web技术基础>>

### 内容概要

《世界著名计算机教材精选：语义Web技术基础》主要介绍了语义万维网基础技术。

《世界著名计算机教材精选

：语义Web技术基础》从实用的角度，从语义Web的发展开始，介绍了当今流行的几种语义Web技术，是国内外第一本关于语义Web技术的图书。

全书包括9章，分别介绍了语义技术的起源，RDF的基础知识，RDF和RDFS的形式语义，Web本体语言OWL，OWL的形式语义，本体和规则，本体的查询语言，本体工程，以及本体的一些应用。

## <<语义Web技术基础>>

### 作者简介

作者：（德国）希茨利尔（Pascal Hitzler）（德国）Markus Krotzsch（德国）Sebastian Rudolph（德国）York Sure 译者：俞勇 希茨利尔是AIFB的助理教授，2009年9月作为教学人员加入俄亥俄州德顿的莱特州立大学Kno.e.sis中心。

2001年，他在科克大学和爱尔兰国立大学获得数学博士学位。

从2001年到2004年，他在德国的德累斯顿技术大学人工智能学院当博士后研究员。

在2003年，他花费了三个月的时间在俄亥俄州克利夫兰的凯斯西储大学做助研。

他多样的研究兴趣覆盖语义万维网、知识表示和推理、神经—符号集成和计算机科学的形式基础。

## <<语义Web技术基础>>

### 书籍目录

#### 第1章 语义的探求

- 1.1 建立模型
- 1.2 用知识计算
- 1.3 交流信息
- 1.4 语义万维网技术
- 1.5 进一步阅读

#### 第2章 用rdf和rdfs模式表示的简单本体

- 2.1 rdf简介
  - 2.1.1 图，而不是树
  - 2.1.2 rdf中的名称：uri
  - 2.1.3 rdf中的数值：文字
- 2.2 rdf语法
  - 2.2.1 从图到三元组
  - 2.2.2 简单的三元组语法：n3、n-triple和turtle
  - 2.2.3 rdf的xml序列化
  - 2.2.4 在xml中的rdf：uri与其他问题
  - 2.2.5 更短的uri：xml实体与相对uri
  - 2.2.6 uri从哪里来，它们的含义是什么
- 2.3 高级特性
  - 2.3.1 rdf中的数据类型
  - 2.3.2 语言设置和数据类型
  - 2.3.3 多值关系
  - 2.3.4 空白结点
- 2.4 rdf模式中简单本体
  - 2.4.1 类与实例
  - 2.4.2 子类和类结构
  - 2.4.3 属性
  - 2.4.4 子属性与属性层次
  - 2.4.5 属性限制
  - 2.4.6 rdfs中的附加信息
- 2.5 编码特别的数据结构
  - 2.5.1 rdf中的列表
  - 2.5.2 关于命题的命题：物化
- 2.6 一个例子
- 2.7 本章总结
  - 2.7.1 rdf(s)语言构成的概述
- 2.8 练习
- 2.9 进一步阅读

#### 第3章 rdf形式语义

- 3.1 为什么要有语义
- 3.2 rdf(s)的模型论语义
  - 3.2.1 简单解释
  - 3.2.2 rdf-解释
  - 3.2.3 rdfs解释
  - 3.2.4 数据类型的解释

## <<语义Web技术基础>>

- 3.2.5 例子
- 3.3 根据推理规则的句法推理
  - 3.3.1 简单推导的演绎规则
  - 3.3.2 rdf-推导的演绎规则
  - 3.3.3 rdfs-推导的演绎规则
  - 3.3.4 针对数据类型的附加规则
  - 3.3.5 rdfs演绎的例子
- 3.4 rdf ( s ) 的语义局限
- 3.5 本章总结
- 3.6 练习
- 3.7 进一步阅读
- 第4章 网络本体语言owl
  - 4.1 用owl表示的本体
    - 4.1.1 owl本体的头
    - 4.1.2 类、角色和个体
    - 4.1.3 简单的类关系
    - 4.1.4 个体间的关系
    - 4.1.5 封闭式类
    - 4.1.6 布尔类构造器
    - 4.1.7 角色限制
    - 4.1.8 角色关系
    - 4.1.9 角色特性
    - 4.1.10 推理类型
  - 4.2 owl的种类
    - 4.2.1 owl full
    - 4.2.2 owl dl
    - 4.2.3 owl lite
  - 4.3 即将推出的owl 2标准
    - 4.3.1 owl 2 dl
    - 4.3.2 owl 2概图
    - 4.3.3 owl 2 full
  - 4.4 本章总结
    - 4.4.1 owl 1语言构造子概述
    - 4.4.2 owl 2语言附加构造子概述
  - 4.5 练习
  - 4.6 扩展阅读
- 第5章 owl形式语义
  - 5.1 描述逻辑
    - 5.1.1 描述逻辑alc
    - 5.1.2 作为描述逻辑的owl dl
    - 5.1.3 命名描述逻辑以及它们与owl子语言之间的关系
    - 5.1.4 sroiq的形式语法
  - 5.2 owl的模型论语义
    - 5.2.1 sroiq的外延语义
    - 5.2.2 通过谓词逻辑定义sroiq语义
  - 5.3 owl自动推理
    - 5.3.1 推理问题

## &lt;&lt;语义Web技术基础&gt;&gt;

- 5.3.2 否定范式
- 5.3.3 alc的tableaux算法
- 5.3.4 shiq的tableaux算法
- 5.3.5 计算复杂性
- 5.4 本章总结
- 5.5 练习
- 5.6 进一步阅读
- 第6章 本体和规则
- 6.1 什么是规则
- 6.2 一阶规则语言datalog
  - 6.2.1 datalog导论
  - 6.2.2 datalog语义
- 6.3 规则和owl dl的结合
  - 6.3.1 合成语义：datalog和描述逻辑
  - 6.3.2 计算结论
  - 6.3.3 描述逻辑规则
  - 6.3.4 dl-safe规则
- 6.4 规则内部交换格式rif
  - 6.4.1 rif-core
  - 6.4.2 面向对象的数据结构：rif中的框架
  - 6.4.3 rif-core语义
  - 6.4.4 对于rif-core的xml句法
  - 6.4.5 rif和owl dl的结合
  - 6.4.6 rif和rdf(s)的结合
  - 6.4.7 rif-core和rif-bld的更多特性
- 6.5 本章总结
- 6.6 练习
- 6.7 进一步阅读
- 第7章 查询语言
- 7.1 sparql：rdf的查询语言
  - 7.1.1 简单的sparql查询
  - 7.1.2 简单图模式：三元组和变量
  - 7.1.3 sparql中的空白结点
  - 7.1.4 复杂图模式：组、可选和可替代
  - 7.1.5 对数据值的查询
  - 7.1.6 过滤器
  - 7.1.7 结果格式
  - 7.1.8 修饰符
  - 7.1.9 sparql语义和sparql代数
    - 7.1.10 sparql的其他表达特性
- 7.2 owl dl的合取查询
  - 7.2.1 owl的局限
  - 7.2.2 合取查询的介绍
  - 7.2.3 非区分变量
  - 7.2.4 合取查询和规则
  - 7.2.5 合取查询与sparql
- 7.3 本章总结

## <<语义Web技术基础>>

7.4 练习

7.5 进一步阅读

第8章 本体工程

8.1 需求分析

8.2 本体创建--你的知识在哪里

8.2.1 在你的大脑中：人类知识源

8.2.2 在你的书中：非结构化源

8.2.3 在互联网中：半结构化源

8.2.4 在数据库中：结构化源

8.3 本体的质量保证

8.3.1 本体评估：怎样改进本体

8.3.2 如何（不）正确地建模

8.3.3 本体精炼：如何改进本体

8.4 模块化本体：分而治之

8.5 软件工具

8.5.1 本体编辑器

8.5.2 rdf存储

8.5.3 owl dl推理引擎

8.5.4 owl 2概要的推理引擎

8.5.5 quonto

8.5.6 数据日志和规则引擎

8.5.7 更多系统

8.6 本章总结

8.7 进一步阅读

第9章 应用

9.1 万维网数据交换与聚合

9.1.1 用元数据为万维网数据添砖加瓦

9.1.2 词汇表

9.2 语义维基

9.2.1 语义媒体维基

9.2.2 应用

9.3 语义门户

9.4 数据格式中的语义元数据

9.5 生命科学中的语义web

9.6 本体用于标准化

9.7 rif应用

9.8 面向未来的应用

9.9 本章总结

9.10 进一步阅读

附录a 可扩展标记语言xml

a.1 简述xml

a.2 xml语法

a.3 xml模式

a.3.1 元素、属性和数据类型

a.3.2 用户自定义类型

附录b 集合论

b.1 基本概念

<<语义Web技术基础>>

b.2 集合操作

b.3 关系和函数

附录c 逻辑

c.1 语法

c.2 语义

c.3 证明理论和可判定性

附录d 练习答案

d.

第2章 解答

d.

第3章 答案

d.

第4章 答案

d.

第5章 答案

d.

第6章 答案

d.

第7章 答案

参考文献



## 章节摘录

版权页：插图：（1）无值（即无约束的变量）。

（2）空结点。

（3）URI。

（4）RDF字面量。

但是一般情况下，RDF规范中并没有给出不同类型的字面量是如何比较的。

只有当xsd:string类型的字面量和一个无类型的字面量比较并且具有相同的字面值的时候，无类型的字面量被定义为更小。

关于两个无约束的变量、两个空结点和两个未知类型的字面量之间的序也是没有定义的，根据具体的实现不同而不同，在这些具体实现中可能还支持其他SPARQL标准没有定义序的类型。

此外，我们还可以根据多个变量的值来对结果排序。

声明出现的顺序决定了这些变量的相关度：当且仅当第一个变量的两个不同约束值相同的时候，第二个变量才会用来排序，以此类推。

对于每一个变量，其顺序都是可以单独给定的，例如在ORDER BY DESC（？price）？

title语句中，它按价格降序而当两个物品价格相等的情况下则依照他们标题的字典（升）序。

几乎所有的查询语言都会有一个关键的特性，即可以选取结果序列中的某一段。

在SPARQL中，它可以通过LIMIT和OFFSET来实现。

通过这两个关键字，我们可以选中一段结果，它的结果数上限为LIMIT，且开始的位置为OFFSET。

比如说，这个查询显示从第25号开始的5个三元组，根据主语的元素（假定都是URI）排序。

这样，切分参数可以用来分段获取结果，这对于许多搜索引擎来说是相当常用的。

至于第25个元素具体是什么，当然依赖于我们采用的序。

如果不定义这样的序，那么使用LIMIT和OFFSET的结果一般来说是不可预测的。

甚至，它的序对于每一个查询都不一样（与实现的细节有关），这样一来获取结果将不可靠，因此选取一段结果的查询必需使用具体的ORDER BY定义。

也因为这个原因，ORDER BY对于CONSTRUCT、ASK和DESCRIBE也是有用的。

最后一个使大的结果集更容易处理的选择是去除结果列表中不必要的重复。

确切地说，保证没有两行结果是完全一样的十分有用的。

这可以通过DISTINCT来实现，而其只允许在SELECT后面出现。

事实上，这个特性对于其他的结果格式来说是没有很大用途的（虽然，在这点上值得讨论，它还是会有点作用，参见练习7.6）。

所有变量的约束都相同的行将会合并为一行。

但是值得注意的是，即便如此并非结果表中的所有冗余都消除了。

例如，可能会存在两行除了空结点的标识符以外其余部分都是相同的。

如同在7.1.3节中讨论的，结果中一个空白结点的标识符可能是相关的，但这只会发生在同一个标识符在其他的三元组中也使用到的情况。

要试图除去所有形式的冗余是相当困难的事情，所以DISTINCT只处理最显然的情况。

本节中所有讨论到的参数都可以相互组合，因此决定它们起作用的顺序是相当重要的。

SPARQL定义了如下的处理步骤：（1）基于ORDER BY对所有结果排序。

（2）消除结果集中所有未被选中的变量（投影）。

（3）如果有要求，则消除重复的结果。

（4）按照OFFSET指引，删除开头的结果。

（5）删除所有LIMIT地址之后的结果。

基于这个处理顺序，我们可以发现结果有可能按照没有被选中的变量排序，然而DISTINCT只基于选中的变量集。

OFFSET和LIMIT的处理指的是可能已经由DISTINCT缩小过的结果集。



## <<语义Web技术基础>>

### 编辑推荐

在附录中我们提供对于背景知识的简要的说明，这对理解《世界著名计算机教材精选:语义Web技术基础》中的内容是需要或有帮助的。

我们建议仅在需要时参考相应内容，如一个对于XML或一阶逻辑的简单介绍。

附录A讲述了XML与XML模式。

附录B列出了集合论中的基本符号。

附录C中重述了一阶谓词逻辑基础。

附录D提供了对于书中相应章节的所有练习的解答。

<<语义Web技术基础>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>