

# Semantic Query for Engineering Knowledge Graphs Combining Rule Reasoning

Yaohui Wang, Yingzhong Zhang\*, Xiaofang Luo

School of Mechanical Engineering, Dalian University of Technology, Dalian Liaoning  
Email: \*zhangyz@dlut.edu.cn

Received: May 11<sup>th</sup>, 2019; accepted: May 23<sup>rd</sup>, 2019; published: May 30<sup>th</sup>, 2019

---

## Abstract

The knowledge graph is composed of fact nodes and directed edges. Ontology provides abstract semantic information supports for these facts and directed edges. It can provide formal representation for the knowledge that is difficult to be structured in the engineering field. An efficient semantic query for knowledge graphs is very necessary. Aiming at the actual requirements of engineering knowledge representation and semantic information query, this paper designs and implements a hybrid open world consumption and closed world consumption semantic query method for knowledge graphs, which combines OWL ontology and SWRL rules, employs object-oriented technology, and bases on backward chained reasoning. This method adopts SQWRL query style and extends its semantic query limitation. Finally, the feasibility of semantic query is verified by an example of automatic recognition for machining features.

## Keywords

Engineering Semantic Information, Ontology, Knowledge Graph, Query, Knowledge Reasoning, Rules

---

## 融合规则推理的工程知识图语义查询

王耀辉, 张应中\*, 罗晓芳

大连理工大学机械工程学院, 辽宁 大连  
Email: \*zhangyz@dlut.edu.cn

收稿日期: 2019年5月11日; 录用日期: 2019年5月23日; 发布日期: 2019年5月30日

---

## 摘要

知识图由事实节点和有向边组成的三元组相互连接而成, 本体为这些事实和有向边提供抽象层面的语义  
\*通讯作者。

信息支持,能够对工程领域难以结构化的知识提供形式化的表达,高效的工程领域知识图语义查询是十分必要的。本文针对工程知识表示和语义信息查询实际需求,采用面向对象技术,设计和实现一个混合开放世界假设和封闭世界假设,融合OWL本体和SWRL规则,基于逆向推理的知识图语义查询方法。该方法采用SQWRL查询风格,扩展了其语义查询限制。最后通过加工特征自动识别实例,验证语义查询的可行性。

## 关键词

工程语义信息,本体,知识图,查询,知识推理,规则

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

知识系统一直是强人工智能系统的核心,它以大量形式化的知识作为基础,通过计算机进行推理,满足各种应用对知识的需求。本体作为一种形式化的、对于共享概念明确的显式表示,能更好地表示各种术语、概念所隐含的知识[1]。随着语义 Web 技术的发展,基于本体的知识表示方法已经得到广泛的应用。OWL (Web Ontology Language, OWL)是由国际万维网组织 W3C 推荐和使用的基于本体的语义 Web 建模语言[2],集成语义 Web 规则语言 SWRL (Semantics Web rule language, SWRL)构成目前强大的知识表示工具[3],使得知识表示具有更强大表示能力和推理能力。知识图(Knowledge Graph)是 Google 公司 2012 年提出的一种表示知识的数据结构[4],旨在实现更智能的搜索引擎;知识图本质上是基于语义网络的知识库,其核心是采用 RDF (Resource Description Framework, RDF)三元组图模式描述知识实例(或者事实)及它们之间的语义关系。

然而,知识系统的一个重要的要求是能对知识表示的语义信息查询。W3C 推荐的 OWL 本体标准查询语言为 SPARQL (Simple Protocol and RDF Query Language, SPARQL) [5],它是建立在图模式匹配上的查询语言;基于描述逻辑查询语义信息,但缺乏与基于 Horn 逻辑的 SWRL 规则的集成,推理查询能力有限。SQWRL (Semantic Query-enhanced Web Rule Language, SQWRL) [6]是基于 SWRL 规则的查询语言,通过定义一组用于查询的运算符及融合 SWRL 规则实现查询,可以很好地将 Horn 规则与 OWL 本体融合,作为一个插件应用在 Protégé 本体编辑平台上,也提供 Java API。Jena 是惠普实验室使用 Java 语言开发的开源语义 Web 框架,它内置了支持 OWL 本体语言的推理机[7],但需要将实例和规则执行繁琐的转化步骤;此外,应用较广泛的推理机还有 Pellet、FaCT++、RACER、Hermit 等[8]。

另一方面,上述基于 OWL 本体的知识表示和推理技术都是基于开放世界假设(Open-world Assumption, OWA) [9],即假设任何一个命题都是真的,无论事实是否是真,都不能作否定的回答,面对 Internet 这样一个无限知识资源场景,OWA 假设有其合理性,这也与一阶逻辑的单调性密切相关。但是在工程应用领域,对一个具体的应用场景,信息是完整和封闭的,符合封闭世界假设(Closed-world Assumption, CWA)对一个查询命题需要做出“真”和“假”的回答,完全采用 OWA 假设的推理机制会限制很多查询功能,有时还会产生不正确的结果[9]。

本文针对工程知识表示和语义信息查询实际需求,采用面向对象技术,设计和实现一个混合 OWA 和 CWA,融合 OWL 本体和 SWRL 规则逆向推理的知识图语义查询方法,开发一个 C++API 动态库接口;

通过实例验证, 为工程信息知识表示和语义查询提供一个有效的工具。

## 2. 相关工作基础与总体方案

### 2.1. 工程领域知识的知识图表示

如上面所叙, 知识图为非结构化知识提供了一种基于语义本体表示的数据结构, 在知识问答系统、自然语言理解等方面得到广泛的应用。在工程领域中, 大量的设计、制造和检测知识很难结构化为规则知识, 知识图表达模式非常适合工程信息知识表示。同时 OWL 是 W3C 组织推荐和使用的语义 Web 本体建模语言, 为知识事实关联的概念提供一个显式、可共享的定义, 从而为语义查询和机器理解打下基础。

按照 Google 的知识图框架, 工程领域一个主题知识可以表示为一个知识图  $G = \langle V, E \rangle$ , 式中  $V$  表示该领域工程知识的事实节点集合,  $E$  表示连接两个实例的有向边集合。领域知识的实例事实通过本体或者资源进行标记, 标记的事实关联到其概念本体。一个工程领域知识库可以表示为一组独立的知识图集合,  $GB = \{G_1, G_2, \dots, G_n\}$ , 为后续的语义查询和推理打下基础。图 1 显示一部分冲压模具设计知识图。

在图 1 中, 三个事实节点: “冲压模具 1”、“钣金工件 1”和“圆孔 1”分别被本体概念类标记, 矩形框节点表示数据事实, 节点用有向的关系边(例如: “冲压”关系、“加工特征关系”等)连接起来。

从知识图概念和图 1 示例, 可以看出知识图表示可以将难以结构化的领域知识给出一个形式化的表示, 可以记录由知识产生的难以结构化的事实。

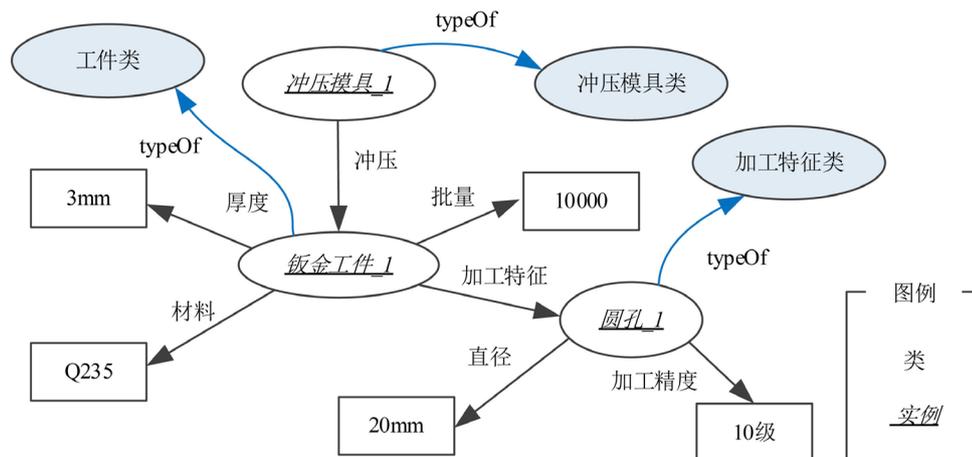


Figure 1. An example of an engineering field (stamping die design) knowledge graph  
图 1. 一个工程领域(冲压模具设计)知识图示例

### 2.2. SWRL 知识规则

上述基于本体构建的知识图难以表示“IF-THEN”等条件关系, 产生式规则则比较容易描述这种知识。SWRL 是常用的规则建立语言, 用于定义产生式规则知识。SWRL 规则基于 Horn 逻辑, 是一阶逻辑的可判定子集, 它可以采用和 OWL 相同的语言或者直接采用人类可读的语法进行表示。SWRL 规则由隐含规则(Imp)、原子(Atom)、变量(Variable)和内置逻辑(Build-in)等部分组成。

一个典型的 SWRL 规则如下所示:

$$Q_1 \wedge Q_2 \wedge Q_3 \wedge \dots \wedge Q_{n-2} \wedge Q_{n-1} \wedge Q_n \rightarrow P_x$$

在这条 SWRL 规则中,  $Q_1, Q_2, \dots, Q_{n-1}, Q_n$  和  $P_x$  表示原子, 每个原子中包含变量, 其中  $Q_1, Q_2, \dots, Q_{n-1}, Q_n$  通过交运算符连接, 表示规则的体,  $P_x$  表示规则的头, 蕴含的意义是如果  $Q_1, Q_2, \dots, Q_{n-1}, Q_n$  全部发生, 则

$P_x$  发生。

Build-in 为内置逻辑运算符集，例如在加工特征自动识别领域，需要使用规则定义加工特征，深孔的定义为长径比  $L$  满足  $5 \leq L \leq 10$ ，用 SWRL 规则描述为：

$$\text{Hole}(?f) \wedge \text{hasDiameter}(?f, ?d) \wedge \text{hasDepth}(?f, ?l) \wedge \text{swrlb:divide}(?q, ?l, ?d) \wedge \text{swrlb:lessThan}(?q, 10) \wedge \text{swrlb:greaterThan}(?q, 5) \rightarrow \text{DeepHole}(?f)$$

其中 ? 后连接变量， $f$  为孔特征类的一个实例变量； $\text{hasDiameter}$  和  $\text{hasDepth}$  表示孔直径数据关系和孔深度数据关系； $\text{divide}$ 、 $\text{lessThan}$  和  $\text{greaterThan}$  为内置逻辑运算，分别表示算术商运算、逻辑小于和逻辑大于。隐含规则为当孔  $f$  的深度  $l$  和直径  $d$  的比值  $q$  满足大于 5 小于 10，则孔  $f$  为深孔。通过这样 SWRL 规则，可以从现有概念和数据导出新的知识概念。

### 2.3. 工程知识图语义查询方案

本文针对工程知识表示和语义信息查询实际需求，采用面向对象技术，设计和实现一个混合 OWA 和 CWA，融合 OWL 本体和 SWRL 规则逆向推理的工程领域知识图语义查询方法，查询框架如图 2 所示。

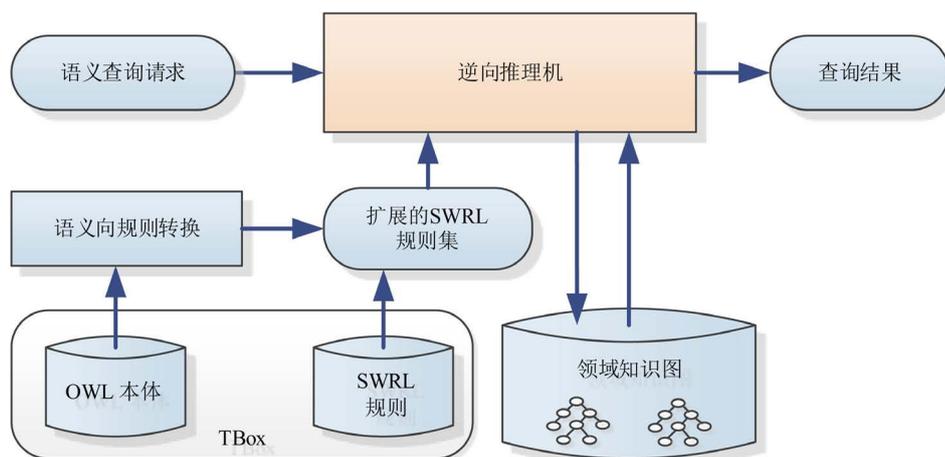


Figure 2. Engineering domain knowledge graph semantic query framework  
图 2. 工程领域知识图语义查询框架

工程语义信息查询框架分为知识库层和推理层。知识库层可以由本系统构建或者加载由 Protégé 构建的 OWL 领域本体和 SWRL 规则。建立完成的领域知识图本体和规则文件，通常为 RDF/XML 格式的 OWL 文档，然后将其中的 TBox、ABox 和 RBox 知识解析为面向对象表示框架中的概念类、属性类、规则类、实例类以及它们的对象知识集合，并将解析后的概念类和属性类知识做规则化处理，与解析后的规则类组成新的规则类集合。通过面向对象知识集合中形式化的实例类和规则类以及它们的对象，构成语义查询的知识集合。

推理层能够支持扩展后的 SQWRL 语义查询语言，通过对查询语言进行解析，与实例类、规则类以及它们的对象知识集合进行满足性和包含性检查，通过反向链式推理算法构造 Protégé 搜索树，完成工程语义信息的检索。

## 3. 融合知识规则的语义查询

### 3.1. 语义布尔查询设计

SQWRL 是在 SWRL 规则语言的基础上对 Build-in 内置逻辑运算进行扩展形成的语义检索语言，它

的核心运算符是 `sqwrl:select`。例如要从图 1 表示的知识图中，检索冲压板厚为 3 mm 的冲压模具实例：

$$\text{StampingDie}(?x) \wedge \text{punch}(?x, ?y) \wedge \text{SheetWorkpiece}(?y) \wedge \text{hasThickness}(?y, ?t) \wedge \text{swrlb:equal}(?t, 3) \rightarrow \text{sqwrl:select}(?x)$$

从上式可以看出，一个语义查询请求，就是一个 SQWRL 陈述。一个 SQWRL 陈述包括两部分：右边部分是一个核心 SQWRL 运算符：`sqwrl:select`，可以取一个或者多个变量，用这些变量值构成一个查询结果表的各个列。左边部分像一个标准的 SWRL 规则前提，将概念原子和特性关系通过布尔与运算联系起来。上述例子中 *StampingDie*、*SheetWorkpiece* 是原子概念类，*punch*、*hasThickness* 是关系，使用逻辑运算符 $\wedge$ 连接。所谓布尔查询，是指在词项上通过布尔逻辑运算符与( $\wedge$ )或( $\vee$ )非( $\neg$ )构建出布尔查询表达式。SQWRL 只支持 $\wedge$ 运算，因为语义查询推理机一般以开放世界假设为前提，认为知识是无限集合，对于 $\neg$ 的运算没有全集  $U$ ，所以任何 $\neg$ 的运算返回的仍然是一个不确定的无限集合。

但是在工程应用领域，知识具有一定的封闭性和完整性，同时对于一个给定的查询命题，也需要高效的给出确定的结论，所以我们以封闭世界假设为前提，对基于布尔的 SQWRL 语义查询进行扩展，允许 $\vee$ 和 $\neg$ 逻辑运算，定义如下：

$\neg$  运算：设概念集合为  $U_1$ ， $A$  为概念集中的概念，则有  $\neg A = U_1 - A$ ；

设某概念的关系集合为  $U_2$ ， $a$  为关系集中的一个关系，则有  $\neg a = U_2 - a$ ；

$\vee$  运算：设概念集合为  $U_1$ ， $A$ 、 $B$  为概念集中的概念，则有  $A \vee B = A \cup B$ ；

设某概念关系集合为  $U_2$ ， $a$ 、 $b$  为关系集中的关系，则有  $a \vee b = a \cup b$ ；

三者的优先级为非( $\neg$ ) > 与( $\wedge$ ) > 或( $\vee$ )，对于它们的推理支持将在下文中详细讲述。

### 3.2. 融合 OWL 本体语义与 SWRL 规则

OWL 本体基于描述逻辑表示概念和概念之间的关系，可以表示丰富的层次语义关系及约束语义关系，具有较强的知识描述能力，但在表达因果关系知识能力不足。在一般语义 Web 技术中，采用 OWL 本体与 SWRL 规则相结合的方式表示知识，由 OWL 定义原子概念，SWRL 定义因果关系的导出概念。因此要求语义查询不仅能查询由 OWL 本体定义的概念实例和关系，而且还能够查询 SWRL 导出的概念实例和关系。

根据本文设计的语义查询方案，采用反向推理的语义查询方法。反向推理算法需要统一格式的规则，需要将描述逻辑术语集合 TBox 规约到 Horn 逻辑的范围内。基于描述逻辑的 OWL 本体对应的描述逻辑子语言是 SHOIN，SHOIN 允许的逻辑运算集合如表 1 所示。SWRL 格式为了兼容 OWL，对于非和传递性等运算符进行限制，本文对 SWRL 原有的运算符集合进行了扩展，能够支持所有 SHOIN 允许的逻辑运算，将基于描述逻辑的 TBox 知识转化为 SWRL 格式的规则，与原本的 SWRL 规则组成新的规则库。注意，这种转化并不是等价，因为 TBox 中知识是双向的，但是规则中是单向的，它只是对于推理的一种支持，并且不会修改本体中的内容。

在表 1 中， $A$ 、 $C$ 、 $D$  表示本体中的类， $R$ 、 $S$  表示本体中的属性， $a$  表示个体实例， $n$  代表值。一些扩展运算符解释： $(?x, a_1, \dots, a_n)$  表示  $x$  的值只能为  $a_1, \dots, a_n$  中的某一个值； $R(?x, a)$  表示  $x$  和  $a$  具有属性  $R$  的属性断言； $R(?x, \geq n)$  表示和  $x$  具有关系  $R$  的属性断言最少为  $n$  个； $\vee$  运算符表示规则体中类的对象全部是规则头中类的对象； $\forall(R(?x, ?y) \wedge C(?y)) \rightarrow A(?x)$  表示只和  $C$  的对象拥有关系  $R$  断言的对象属于类  $A$ 。

### 3.3. 基于逆向推理的知识图语义查询

本文所讨论的语义查询，是一个类 SQWRL 语义查询。为了获得一个查询结果，从查询目标问题出

**Table 1.** From OWL class axioms to SWRL rules  
**表 1.** 从 OWL 类公理到 SWRL 规则

项目	命名	描述逻辑	OWL 本体构造符	SWRL 规则
	一般公理	概念层次	rdfs:subClassOf	$B(?x) \rightarrow A(?x)$
	<i>O</i>	Nominals	owl:oneOf owl:hasValue	$(?x, a_1, \dots, a_n) \rightarrow A(?x)$ $R(?x, a) \rightarrow A(?x)$
	<i>N</i>	数量限制	owl:minCardinality owl:maxCardinality	$R(?x, \geq n) \rightarrow A(?x)$ $R(?x, \leq n) \rightarrow A(?x)$
概念构造器公理		$\cap$	owl:intersectionOf	$C(?x) \wedge D(?x) \rightarrow A(?x)$
		$\cup$	owl:unionOf	$C(?x) \vee D(?x) \rightarrow A(?x)$
	<i>S</i>	$\exists$	owl:someValueFrom	$\exists(R(?x, ?y) \wedge C(?y)) \rightarrow A(?x)$
		$\forall$	owl:allValueFrom	$\forall(R(?x, ?y) \wedge C(?y)) \rightarrow A(?x)$
		$\neg$	owl:complementOf	$\neg C(?x) \rightarrow A(?x)$
	<i>S</i>	关系传递	TransitiveProperty	不支持
角色构造器公理	<i>H</i>	关系层次	rdfs:subPropertyOf	$R(?x, ?y) \rightarrow S(?x, ?y)$
	<i>I</i>	关系反向	owl:inverseOf	$R(?x, ?y) \rightarrow S(?y, ?x)$

发, 如果能够从知识图中找到一组变量值(一个变量可以是一个实例或者一个数据类型值), 替换 SQWRL 陈述左侧中的所有变量符号, 使得陈述左侧布尔表达式为真, 该方案就是一个可行的方案, 该组变量值作为查询结果输出, 该方案称为满足性方案。

显然, 通过搜索和匹配知识图中的节点(实例节点和数据节点)找到满足性方案很关键。本文采用逆向推理策略进行推理, 因为目标已经确定, 只需从目标状态出发, 按照子目标组成的逻辑顺序逐级向初始状态递归, 直至满足现有条件。下面将介绍具体推理算法:

1) SQWRL 陈述左侧概念和关系递归替换

如上面所述, SQWRL 陈述左侧部分是一个标准的 SWRL 规则前提, 通过布尔运算符将概念原子和特性关系连接起来。因此首先要检查规则中的概念和特性关系是否是 OWL 已经定义的概念和关系, 如果没有定义, 需要从 SWRL 规则库中搜索, 是否是一个导出的概念, 如果还不是, 查询推理结束; 否则将该导出概念规则的规则体替换该导出概念, 同时对替换体做递归检查, 直至全部概念和关系已经定义。

2) 满足性方案搜索

实际上就是一个对知识图节点事实逆向推理过程, 更具体说就是为了 SQWRL 的查询主体从知识图中搜索满足方案, 能够满足查询主体的所有事实(或者实例)即为查询(或者推理)结果。

为了高效地搜索和匹配知识图中的实例, 本文采用著名的 Prolog 生成树作为搜索数据结构。每一个可行方案构成生成树的一个叶子。

- 搜索树节点数据结构

搜索树节点是变量的取值状态, 最初状态是每个变量的值域。取值状态是变化的, 变量个数也是不确定的, 因此首先需要设计节点的数据结构, 如下所示:

```

struct variableTable
{
    CString strVariableName;
    CString strClassName;
    std::vector<cstring> strValueTable;
};

struct variableNode
{
    std::vector< variableTable > variableTable;
}

```

上式中 *strVariableName* 表示变量名, *strClassName* 表示变量关联的类名, *strValueTable* 表示在特定状态(节点下)下该变量的值域表, 用一个 *vector* 动态数组存储。*variableTable* 表示树节点下的变量表, 用一个 *vector* 动态数组存储, 记录该节点下所有变量。

- 搜索推理步骤

在上述节点数据结构基础上, 搜索推理步骤如下:

**Step1:** 构建规则体原子关系表。根据查询规则体出现的原子关系(一般是二元关系), 构建原子关系表 *ARL*。原子关系表由一组关系组成, 关系包括关系名, 定义域和值域。

**Step2:** 构造 Prolog 生成搜索树根。根据查询规则体出现的变量(用? 符号标识的字符)及关联的概念类, 构建该变量的值域表, 在初始状态(树根状态), 值域表存放该变量值域的所有实例或者数据; 将变量值域表添加到树根节点变量表动态数组中, 形成 Prolog 生成搜索树的根节点。

**Step3:** 开始变量选择。在反向推理中, 推理总是从具有最少变量值域, 并且还没有完全绑定到一个实例值的变量开始, 可以使得搜索树的分岔数量最少。因此对每一层推理, 首先选择具有最少变量值域的变量作为推理的开始, 如果选定, 设为 *s*, 即对 *s* 值域中的每一个变量产生分岔操作。

**Step4:** 搜索树分岔。搜索树分岔操作就是搜索树增长, 即从现有树节点构造新的节点。每一次分岔, 绑定 *s* 值域中以前没有绑定过的一个值; 根据规则体原子关系表顺序, 依次选择关系, 获得其他变量相应取值, 变量取值受到变量值域和规则体关系约束, 取逻辑交集, 生成一个新的树节点。例如, 设变量 *x*, *y*, 它们的值域分别为  $\{x_1, x_2, x_3\}$ ,  $\{y_1, y_2, y_3\}$ , 规则体关系  $op_1\{(x_1, y_1), (x_1, y_2), (x_2, y_2)\}$ , 则如果 *s* 取(绑定)  $x_1$ , 执行  $op_1$  关系, 则当前变量 *y* 取值  $y_1$  和  $y_2$ 。

**Step5:** 重复上述步骤 Step3 和 Step4, 直到满足如下条件结束: 1) 该节点的变量表中的任一个变量的取值域为空, 表明该变量在该分岔是不可求解的, 不能继续生长新的节点; 2) 所有变量都已经被绑定到一个确定的值上, 表明该节点变量绑定的值就是一个可满足方案。如果不满足条件, 则从该节点开始, 重复上述步骤 Step3 和 Step4。

## 4. 应用实例

为了更好说明和验证本文提出的方法, 同时考虑论文篇幅, 选择图 3 所示 CAD 边界模型的加工特征自动识别作为应用实例。加工特征是加工零件上具有特定加工语义的一组几何形状[10]。通过本文提出的语义查询方法, 可以自动识别检索出需要查询的特征, 主要过程如下:

### 1) 边界模型的知识图表示

首先构建加工特征识别应用领域本体。本体构建方法较多, 本文采用 protégé 构建 OWL 格式的加工特征识别领域本体。下面只列出相关的本体类和关系:

原子类(classes): 几何面(GeomFace), 平面(Plane), 圆柱面(Cylinder), 外圆柱面(OuterCylinder), 内圆柱面(InnerCylinder), 圆锥面(Cone), 外圆锥面(OuterCone), 内圆锥面(InnerCone)。平面、圆柱面和圆锥面是几何面子类, 外圆柱面和内圆柱面是圆柱面子类。

对象特性(object properties): 凹邻接(concaveAdjoin), 凸邻接(convexAdjoin)。

图 3 所示边界模型就可以表示为如图 4 所示的知识图。图中节点是标注过的实例, 图 3 零件模型有如下本体标注实例:

Plane{F1, F2, F3, F4, F5, F6, F8, F9, F11, F12}; InnerCylinder{ F7, F10, F13}; InnerCone{ F14}; concaveAdjoin{(F7, F8), (F9, F12), (F10, F12), (F11, F12), (F13, F14)}; convexAdjoin{(F7, F6), (F9, F6), (F10, F6), (F11, F6), (F13, F2), (F12, F2)}....}

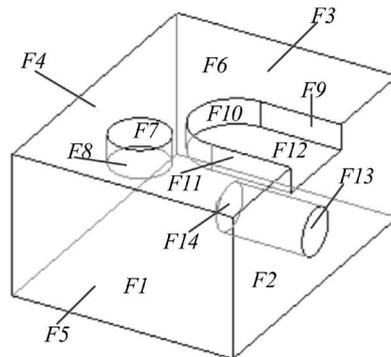


Figure 3. Part boundary model

图 3. 零件边界模型

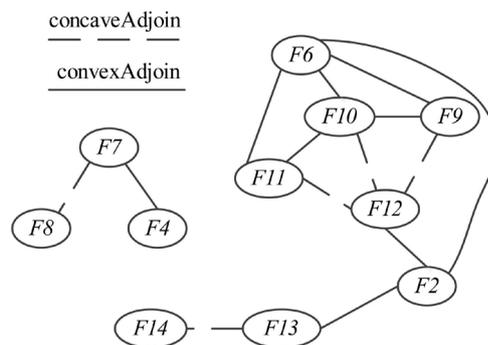


Figure 4. Face adjacency knowledge graph (part)

图 4. 面邻接知识图(部分)

2) 定义特定加工特征知识规则

OWL 本体仅定义概念的层次语义关系, 很多因果关系知识通过 SWRL 规则定义, 例如定义钻盲孔特征, 通过钻削加工形成的盲孔形状, 可以定义如下:

$InnerCylinder(?x) \wedge convexAdjoin(?x, ?y) \wedge concaveAdjoin(?x, ?z) \wedge InnerCone(?z) \rightarrow DrillingHole(?x)$

3) 通过语义查询实现加工特征识别

按照本文提出的语义查询方法, 就可以从面知识图中查询加工特征面。例如本实例, 要查询钻盲孔特征面, 执行如下查询语句:

$DrillingHole(?x) \wedge concaveAdjoin(?x, ?y) \rightarrow select(?x, ?y)$

从上述查询语句, 分析推理过程如下:

首先 *DrillingHole* 不是一个原子概念，但在 SWRL 规则中定义为导出概念，因此将定义 *DrillingHole* 的规则体部分替代 *DrillingHole*。Prolog 推理生成树如图 5 所示。从图中可以看出面 *F13*、*F14* 是查询结果。

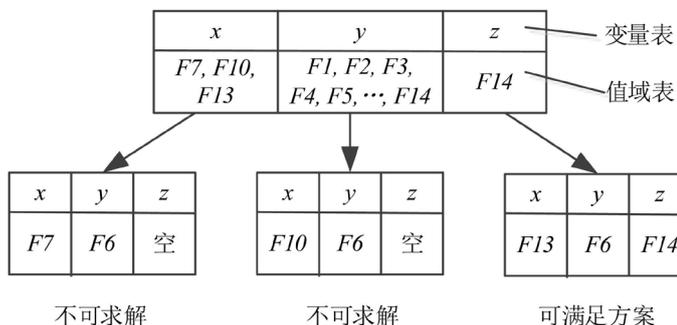


Figure 5. Query drilling blind hole inference spanning tree

图 5. 查询钻盲孔推理生成树

## 5. 结论

基于 OWL 本体构建的工程领域知识图可以被计算机处理，知识概念化、形式化且可共享，融合 SWRL 规则具有强大的知识表示能力，但是融合 OWL 本体与 SWRL 规则实现语义查询的工具还比较少。本文采用面向对象技术，将 OWL 本体处理为 SWRL 格式的规则，在封闭世界假设下集成 OWL 本体和 SWRL 规则，采用类 SQWRL 的查询语言，对工程知识图进行推理和语义查询，该方法非常适合处理工程语义信息，由于采用面向对象 Visual C++ 编程，查询效率更高。

下一步将对推理任务进行分割，将复杂的查询分解为一组简单查询的组合，同时对拥有复杂关系和限制的类提供更加高效的推理查询。

## 基金项目

本文获得国家自然科学基金(51775081)资助。

## 参考文献

- [1] Gruber, T.R. (1993) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5, 199-220. <https://doi.org/10.1006/knac.1993.1008>
- [2] McGuinness, D.L. and Harmelen, F.V. (2004) OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/>
- [3] Horrocks, I., Patel-Schneider, P.F., Boley, H., et al. (2004) SWRL: A Semantic Web Rule Language Combining OWL and RuleML. <http://www.w3.org/Submission/SWRL>
- [4] Singhal, A. (2012) Introducing the Knowledge Graph: Things, Not Strings. <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>
- [5] W3C (2008) SPARQL Query Language for RDF. <https://www.w3.org/TR/rdf-sparql-query/>
- [6] O'Connor, M. and Das, A. (2009) SQWRL: A Query Language for OWL. *Proceedings of the 6th International Conference on OWL: Experiences and Directions*, Chantilly, 208-215.
- [7] The HP Semantic Web Team (2008) Jena Semantic Web Framework. <http://www.jena.sourceforge.net/index.html>
- [8] Abburu, S. (2012) A Survey on Ontology Reasoners and Comparison. *International Journal of Computer Applications*, 57, 33-39.
- [9] Wang, Q. and Yu, X. (2011) Reasoning over OWL/SWRL Ontologies under CWA and UNA for Industrial Applications. *Proceedings of the 24th International Conference on Advances in Artificial Intelligence*, Springer, Berlin, Heidelberg, 789-798. [https://doi.org/10.1007/978-3-642-25832-9\\_80](https://doi.org/10.1007/978-3-642-25832-9_80)

- 
- [10] Zhang, Y., Luo, X., Zhang, B., *et al.* (2017) Semantic Approach to the Automatic Recognition of Machining Features. *The International Journal of Advanced Manufacturing Technology*, **89**, 417-437.  
<https://doi.org/10.1007/s00170-016-9056-8>

**知网检索的两种方式：**

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>  
下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 <http://cnki.net/>  
左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：[csa@hanspub.org](mailto:csa@hanspub.org)