

# 基于 SWRL 规则的本体推理研究

周 亮,黄志球,倪 川

(南京航空航天大学 计算机科学与技术学院,江苏 南京 210016)

**摘 要:**本体具有较强的知识表达能力,目前已经成为计算机学科的一个研究热点。本体在知识推理方面的能力比较弱,已成为 OWL 技术推广应用的主要瓶颈。将语义 Web 规则语言(Semantic Web Rule Language, SWRL)引入到本体中,能大大改善本体的推理能力,从而挖掘出许多新的隐含知识。文中将本体引入到故障树领域中,对如何构建故障树本体及相应的 SWRL 规则进行了研究。首先采用 OWL 语言构建故障树领域本体,然后将故障树中事件之间的逻辑关系转化成 SWRL 规则语言,最后将故障树领域本体和 SWRL 规则放入 JESS 推理机中进行推理,能挖掘出故障树中的隐含知识,从而解决系统故障的快速定位。通过实验证明了文中提出方法的可行性和有效性。

**关键词:**本体;故障树;语义 Web 规则语言;推理规则

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2015)10-0067-04

doi:10.3969/j.issn.1673-629X.2015.10.014

## Research on Ontology Reasoning Based on SWRL Rules

ZHOU Liang, HUANG Zhi-qiu, NI Chuan

(School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,  
Nanjing 210016, China)

**Abstract:** Ontology has a strong ability to express knowledge and has become a hot research topic in computer science currently. But ontology has a weak ability in knowledge reasoning, which makes it a main bottleneck in spreading ontology technology. It will greatly improve the ability of reasoning by introducing Semantic Web Rule Language (SWRL) into ontology, which can gain implicit knowledge. In this paper, introduce ontology into fault tree domain and study how to construct a FT domain ontology and SWRL. First, construct a FT domain ontology by Web Ontology Language. Second, transform the logical relationship among events of FT into SWRL. Finally, put these SWRL rules and the FT ontology into an inference engine JESS. Then new knowledge is produced and it is exploited for the rapid location of system faults. Through the experiment, it proves the correctness and effectiveness of proposed method.

**Key words:** ontology; fault tree; SWRL; reasoning rule

## 0 引 言

本体是对概念以及概念之间关系的精确描述<sup>[1]</sup>,在知识的重用和共享方面优势明显,得到了国内外许多科研人员的关注,并在计算机的许多领域(语义 Web、知识库智能系统等)得到了广泛应用。但本体在知识推理方面的能力很有限,推理效果较差,并且知识库维护比较困难。因此,知识推理已成为本体技术推广应用的主要瓶颈,亟待解决。

语义网规则语言(SWRL)<sup>[2]</sup>是万维网联盟(W3C)于 2004 年提出,用于描述推理规则。它是以 OWL 子语言 OWL DL<sup>[3]</sup>、OWL Lite 和 RuleML 为基

础的规则描述语言。其目的是为了让规则可与 OWL 产生结合,从而提高本体的推理能力。

## 1 本体描述语言

本体是对某个领域(资源)中的相关概念以及概念之间关系的精确描述<sup>[4]</sup>。OWL 是一种本体描述语言,是对本体进行形式化描述,使得计算机能够理解本体描述的知识。总而言之,本体是可形式化的知识,是概念化的明确说明<sup>[5]</sup>。本体理论及相关技术近年来也得到了不断的完善与发展,在许多领域已经得到了广泛的应用。目前,国外建立的本体非常多,已经实现的

收稿日期:2014-11-28

修回日期:2015-03-04

网络出版时间:2015-09-23

基金项目:国家自然科学基金资助项目(61100034,61170043);中央高校基本科研业务费专项资金(CXZZ11\_0218)

作者简介:周 亮(1983-),男,硕士研究生,研究方向为基于本体的软件工程、语义 Web;黄志球,博士,博士生导师,CCF 高级会员,研究方向为软件工程、软件度量。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150923.1509.072.html>

本体主要有:Cyc<sup>[6]</sup>、WordNet<sup>[7]</sup>、UMLS<sup>[8]</sup>等。Cyc 项目是由 DouglasLenat 在 1984 年设立的,由 Cycorp 公司开发并维护,其目标是建立一个庞大的人类常识知识库,使人工智能的应用能够以类似人类推理的方式工作。WordNet 是 Princeton 大学的心理学家、语言学家和计算机工程师联合设计的一种基于认知语言学的英语词典。它不是光把单词以字母顺序排列,并且按照单词的意义组成一个“单词的网络”,并支持自动的文本分析以及人工智能应用。UMLS 又称为统一医学语言系统,是对生物医学科学领域内许多受控词表的一部纲目式汇编。UMLS 提供的是一种位于这些词表之间的映射结构,目的是使这些不同的术语系统之间能够彼此转换。

OWL 语言是 W3C 组织于 2004 年批准的本体描述语言推荐标准<sup>[9]</sup>。OWL 包含三个子语言:OWL Lite、OWL DL 和 OWL Full。OWL Full 的表达能力最强,OWL Lite 的表达能力最弱。在表达能力和包含的语言构成上,每一种子语言都是前一种语言的扩展。OWL Lite 一般用于只需要一个分类层次和简单约束的场合。OWL DL 具有较强的语义表达能力并且支持高效推理。OWL Full 包含了所有的 OWL 词汇和 RDF(S)提供的原语,知识表达能力在 OWL 三种子语言中最强大。但是 OWL Full 取消了基数限制中对传递型属性的约束,因此推理变得不可判定。

2 SWRL 相关理论

本体属于数据层,用来描述领域资源<sup>[10]</sup>。虽然本体支持推理,但因为没有提供用于推理的规则,很难表达类属性之间的关系,其作用非常有限,而且使用起来非常麻烦。

SWRL 主要的表示方式有两种:XML 和 RDF。其中,XML 方式以 RuleML+OWLX 描述,RDF 方式以 OWL+RDF 描述<sup>[11]</sup>,如图 1 所示。SWRL 的两种表示方式的语法结构一致,功能相同,只是表达方式不同,目的都是为了能够更方便地将推理规则和本体结合在一起,使本体表达的内容更丰富,更具有实用性<sup>[11]</sup>。

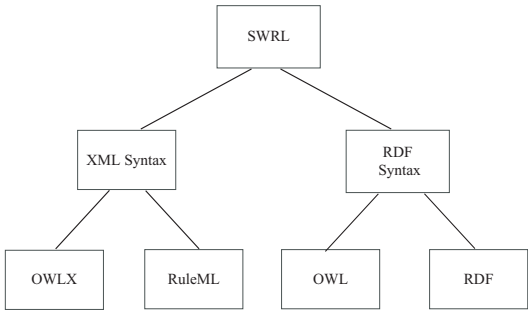


图 1 SWRL 表示方式

SWRL 规则的基本形式是表示前提和结论的推导

关系。前提和结论都可以包括单个或者多个基本命题,基本命题之间是逻辑与的关系<sup>[12]</sup>。形式为:Antecedent→Consequent。SWRL 由 Imp、Atom、Variable 和 Building 四个部分组成<sup>[13]</sup>。Atom 用于定义条件判断的限制式;Imp 是用于定义规则的,Imp 中的限制式是由 Atom 提供;Building 用于定义各种逻辑比较关系,分别用于数值比较、布尔运算、字符串操作等。在本体中 SWRL 规则主要使用两个限制式:

(1)  $C(x)$ : $x$  可以是变量或本体的实例; $C$  是类。说明  $x$  是  $C$  类的一个实例,如  $Parent(?x)$ ,说明  $Parent$  是一个类。

(2)  $P(x,y)$ : $x,y$  可以是变量或本体的实例, $P$  是对象属性。如  $hasBrother(?x,?y)$ ,说明  $y$  是  $x$  的兄弟。

举例说明如下:

$hasFather(?x,?y) \wedge hasBrother(?y,?z) \rightarrow hasUncle(?x,?z)$

在上述 SWRL 规则中, $x,y,z$  为类  $Person$  的 3 个不同的实例, $hasParent$ 、 $hasBrother$ 、 $hasUncle$  为 3 个不同的对象属性。该规则的前提是:如果  $y$  是  $x$  的父亲,并且  $z$  是  $y$  兄弟。结论是: $z$  是  $x$  的叔叔。

3 实验分析

3.1 故障树本体的构建

故障树是一种定性的逻辑关系因果图,主要包含两个要素:事件和逻辑门<sup>[14]</sup>。逻辑门用于表达事件与事件之间的逻辑关系。事件可分为:输入事件和输出事件。文中利用 OWL 语言构建故障树本体,构建平台为 Protégé。故障树本体的类信息如表 1 所示。

表 1 故障树本体的类信息表

Class	Class
Fault_tree_name	Event_species
Event	Logic_gates
Event_status	Logic_gate_species

表 1 中,Fault\_tree\_name、Event、Event\_status、Event\_species、Logic\_gates、Logic\_gate\_species 是六个平行的基本类。分别代表故障树的名称、事件、事件状态(发生和不发生)、事件的种类(输入事件和输出事件)、逻辑门和逻辑门的种类。部分对象属性的信息如表 2 所示。

最后,根据具体的故障树,选择 Protégé 工具的 Individuals 标签页,就可以对本体创建实例和添加实例的属性值。通过上述步骤,故障树本体初步创建完成。本体构建完成后,还需要对其进行一致性验证,确保本体中包含的所有知识不存在矛盾。文中采用 Protégé

工具自带的 pellet 推理机对构建的故障树本体进行检查,是满足一致性要求的。

表 2 故障树本体的部分对象属性信息表

属性名	定义域	值域	属性说明
Fault_tree_name_is	Event、Logic_gate	Fault_tree_name	故障树名称是
Event_species_is	Event	Event_species	事件的种类是
Status_is	Event	Event_status	事件的状态是
condition_is	Logic_gate	Event	禁门和优先与门中输出事件发生的条件是
Logic_gate_species_is	Logic_gate	Logic_gate_species	逻辑门的种类是
output_event_is	Logic_gate	Event	逻辑门的输出事件是
input_event_1_is	Logic_gate	Event	逻辑门的第一个输入事件是

3.2 构建 SWRL 规则

在故障树中,事件之间的逻辑关系是用逻辑门来表示的,分别为与门、或门、非门、优先与门、表决门、异或门和禁门七种逻辑门。

(1)与门(And\_gate):与门的输入事件全部发生,输出事件才会发生。逻辑与门表示的是事件的交集。

假设逻辑门  $G$  为与门,  $A$  是与门  $G$  的输出事件,  $B$  和  $C$  是与门  $G$  的输入事件。如果输出事件  $A$  发生,由与门的定义,可知:输入事件  $B$  和输入事件  $C$  一定会同时发生。

SWRL 规则如 Rule-1 所示:  
Rule-1:  $\text{Logic\_gate}(?x) \wedge \text{Logic\_gate\_species\_is}(?x, \text{与门}) \wedge \text{output\_event\_is}(?x, ?a) \wedge \text{input\_event\_1\_is}(?x, ?y) \wedge \text{input\_event\_2\_is}(?x, ?z) \wedge \text{Status\_is}(?a, \text{发生}) \rightarrow \text{Status\_is}(?y, \text{发生}) \wedge \text{Status\_is}(?z, \text{发生})$

在规则 1 中:  $\text{Logic\_gate}$ 、 $\text{Logic\_gate\_species\_is}$ 、 $\text{output\_event\_is}$ 、 $\text{input\_event\_1\_is}$ 、 $\text{input\_event\_2\_is}$ 、 $\text{Status\_is}$  来源于 3.1 节构建的故障树本体中的类和对象属性。

对规则 1 解析如下:  $\text{Logic\_gate}(?x)$  表示  $x$  是  $\text{Logic\_gate}$  类的一个实例;  $\text{Logic\_gate\_species\_is}(?x, \text{与门})$  表示  $x$  为与门;  $\text{output\_event\_is}(?x, ?a)$  表示实例  $x$  的  $\text{output\_event\_is}$  为实例“ $a$ ”;  $\text{input\_event\_1\_is}(?x, ?y)$  和  $\text{input\_event\_2\_is}(?x, ?z)$  表示实例  $x$  的  $\text{input\_event\_1\_is}$ 、 $\text{input\_event\_2\_is}$  分别为实例“ $y$ ”和实例“ $z$ ”;  $\text{Status\_is}(?a, \text{发生})$  表示实例  $a$  的  $\text{Status\_is}$  为实例“发生”;  $\text{Status\_is}(?y, \text{发生})$  和  $\text{Status\_is}(?z, \text{发生})$  表示实例  $y$  和  $z$  的  $\text{Status\_is}$  都为实例“发生”。文中之后的 SWRL 规则,和规则 1 类似,限于篇幅,不再做详细解析。

(2)或门(Or\_gate):一个或者多个输入事件发生时,输出事件即可发生。逻辑或门表示的是事件的并集。

假设逻辑门  $G$  为或门,  $A$  是或门  $G$  的输出事件,  $B$  和  $C$  是或门  $G$  的输入事件。如果输出事件  $A$  发生,由

或门的定义,可知输入事件会有 3 种可能情况:

- ①  $B$  事件发生,  $C$  事件不发生;
- ②  $B$  事件不发生,  $C$  事件发生;
- ③  $B$  事件和  $C$  事件都发生。

当逻辑门为或门,在不知道输入事件是否发生的时候,文中默认为第一个输入事件发生,其余的输入事件不发生(根据事件发生的概率,按照从大到小的原则,依次为第一个输入事件、第二个输入事件等。这样可以使故障判定更加准确),将推理结果和实际情况进行比较。如果二者不一致,则第二个输入事件发生,其余不发生。依次类推,来建立 SWRL 规则。异或门和表决门构建 SWRL 规则的思路,和或门相同。

SWRL 规则如 Rule-2、Rule-3 所示:  
Rule-2:  $\text{Logic\_gate}(?x) \wedge \text{Logic\_gate\_species\_is}(?x, \text{或门}) \wedge \text{output\_event\_is}(?x, ?y) \wedge \text{input\_event\_1\_is}(?x, ?z) \wedge \text{input\_event\_2\_is}(?x, ?a) \wedge \text{Status\_is}(?y, \text{发生}) \wedge \text{Status\_is}(?a, \text{不发生}) \rightarrow \text{Status\_is}(?z, \text{发生})$   
Rule-3:  $\text{Logic\_gate}(?x) \wedge \text{Logic\_gate\_species\_is}(?x, \text{或门}) \wedge \text{output\_event\_is}(?x, ?y) \wedge \text{input\_event\_1\_is}(?x, ?z) \wedge \text{input\_event\_2\_is}(?x, ?a) \wedge \text{Status\_is}(?y, \text{发生}) \wedge \text{Status\_is}(?z, \text{不发生}) \rightarrow \text{Status\_is}(?a, \text{发生})$

(3)非门(Not\_gate):非门的输出事件是将输入事件取反后得到的对立事件。

假设逻辑门  $G$  为非门,  $A$  是非门  $G$  的输出事件,  $B$  是非门  $G$  的输入事件。如果输出事件  $A$  发生,由非门的定义,可知:输入事件  $B$  不发生。

SWRL 规则如 Rule-4 所示:  
Rule-4:  $\text{Logic\_gate}(?x) \wedge \text{Logic\_gate\_species\_is}(?x, \text{非门}) \wedge \text{output\_event\_is}(?x, ?y) \wedge \text{input\_event\_1\_is}(?x, ?z) \wedge \text{Status\_is}(?y, \text{发生}) \rightarrow \text{Status\_is}(?z, \text{不发生})$

优先与门、表决门、异或门和禁门可以由与门、或门和非门组成而成<sup>[15]</sup>,对应的 SWRL 规则在这不做详

细介绍。

图 2 是某一待分析的包含七种不同逻辑门的故障树。首先将图 2 所示的故障树,在 3.1 节中建立的本体中进行实例化(创建本体中的实例);然后根据 3.2 节中对逻辑门的分析及 SWRL 规则的语法,构建出相应的 SWRL 规则;最后将本体和 SWRL 规则,放入 JESS 推理引擎<sup>[16]</sup>中进行推理,就挖掘出故障树中的隐含信息。详细的 SWRL 规则和实验结果如图 3 所示。

在图 3 的 SWRL 规则中,Rule-1 表达的意思为:如果逻辑门为与门,逻辑门的输出事件发生,那么逻辑门的输入事件就会发生。Rule-2 和 Rule-3 表达的意思为:如果逻辑门为或门,逻辑门的输出事件发生,那么逻辑门的输入事件有一个或者多个发生。Rule-4 表达的意思为:如果逻辑门为非门,逻辑门的输出事件

发生,那么逻辑门的输入事件就不会发生。其他的规则分别为优先与门、表决门、异或门和禁门。

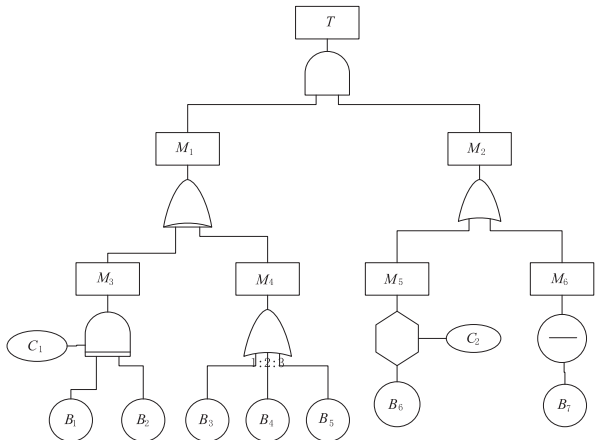


图 2 某一待分析的故障树

Name	Expression
Rule-1	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 与门) ∧ output_event_is(?x, ?a) ∧ input_event_1_is(?x, ?y) ∧ input_event_2_is(?x, ?z) ∧ Status_is(?a, 发生) → Status_is(?y, 发生) ∧ Status_is(?z, 发生)
Rule-10	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 表决门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ input_event_3_is(?x, ?b) ∧ Status_is(?y, 发生) ∧ Status_is(?a, 不发生) → Status_is(?z, 发生) ∧ Stat...
Rule-11	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 表决门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ input_event_3_is(?x, ?b) ∧ Status_is(?y, 发生) ∧ Status_is(?a, 不发生) → Status_is(?z, 发生) ∧ Stat...
Rule-2	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 或门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ Status_is(?y, 发生) ∧ Status_is(?a, 不发生) → Status_is(?z, 发生)
Rule-3	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 或门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ Status_is(?y, 发生) ∧ Status_is(?a, 不发生) → Status_is(?z, 发生)
Rule-4	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 非门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ Status_is(?y, 发生) → Status_is(?z, 不发生)
Rule-5	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 优先与门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ condition_is(?x, ?b) ∧ Status_is(?y, 发生) → Status_is(?z, 发生) ∧ Status_is(?a, 发生) ∧ Status_is...
Rule-6	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 异或门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ Status_is(?y, 发生) ∧ Status_is(?a, 不发生) → Status_is(?z, 发生)
Rule-7	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 异或门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ Status_is(?y, 发生) ∧ Status_is(?a, 不发生) → Status_is(?z, 发生)
Rule-8	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 禁门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ condition_is(?x, ?a) ∧ Status_is(?y, 发生) → Status_is(?a, 发生) ∧ Status_is(?z, 发生)
Rule-9	Logic_gate(?x) ∧ Logic_gate_species_is(?x, 表决门) ∧ output_event_is(?x, ?y) ∧ input_event_1_is(?x, ?z) ∧ input_event_2_is(?x, ?a) ∧ input_event_3_is(?x, ?b) ∧ Status_is(?y, 发生) ∧ Status_is(?z, 不发生) → Status_is(?a, 发生) ∧ Stat...

SWRLJESSTabRulesAsserted AxiomsInferred AxiomsOWL 2 RLL

Inferred Axioms

Status\_is(M6, 发生)  
Status\_is(M2, 发生)  
Status\_is(M1, 发生)  
Status\_is(B2, 发生)  
Status\_is(B1, 发生)  
Status\_is(B7, 不发生)  
Status\_is(C1, 发生)  
Status\_is(M3, 发生)

图 3 SWRL 规则和实验结果

图 3 的实验结果表明,如果  $T$  事件和  $M_3$  事件发生,  $B_6$  不发生,那么通过推理得出,基本事件  $B_1$ 、 $B_2$  和  $B_7$  发生。这和故障树分析法得出的结论是一致的。

## 4 结束语

文中将 SWRL 规则与本体结合起来,能大大改善本体的推理能力,从而挖掘出许多新的隐含知识。通过实验证明,用 OWL 来构建故障树本体,用 SWRL 规则建立推理规则,将两者放入 JESS 推理机中进行推理,推理出的结果与使用故障树分析法得出的结论是一致的。由于本体能精确描述出概念以及概念之间的关系,因此,能够解决故障树重复构建问题。

但是,将本体和 SWRL 规则引入到故障树领域,还处于探索阶段,构建更为普遍适用的故障树本体及相应的 SWRL 规则,还需要进行大量工作。例如,如何利用本体和 SWRL 规则来解决故障树的定量分析;如何解决动态故障树的问题等。这需要进行更深入的研究,也是下一步研究的方向。

## 参考文献:

[1] 葛 强,沈国华,黄志球,等. Web 服务中支持本体推理的

隐私保护研究[J]. 计算机科学与探索,2013,7(6):536-544.

[2] Horrocks I, Patel-Schneider P F, Boley H, et al. SWRL: a semantic web rule language combining OWL and RuleML[R]. [s. l.]: W3C Member Submission, 2004.

[3] W3C. OWL Web ontology language semantics and abstract syntax[DB/OL]. (2004) [2014]. <http://www.w3.org/TR/2004/REG-owl-semantics-20040210.html>.

[4] 杜小勇,李 曼,王 珊. 本体学习研究综述[J]. 软件学报,2006,17(9):1837-1847.

[5] 邓志鸿,唐世渭,张 铭,等. Ontology 研究综述[J]. 北京大学学报:自然科学版,2002,38(5):730-738.

[6] Weikum G, Theobald M. From information to knowledge: harvesting entities and relationships form web sources[C]//Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on principles of database system. Indiana: ACM Press, 2010:65-76.

[7] Navigli R, Ponzetto S P. BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network[J]. Artificial Intelligence, 2012, 193: 217-250.

[8] Cimino J J. High-quality, standard, controlled healthcare terminologies come of age[J]. Methods of Information in Medi-



表 2 协议特征字段与检测结果

协议	特征字段	错误率/%	漏判率/%
HTTP	47 45 54 202f	0.011	0
FTP	76 69 63 65	0.021	0
SMTP	52 53 45 54 0d0a	0.016	0.12
PPLIVE	76 02 46 543f 42	0.019	0.15

由表 2 可知,FTP 协议识别的错误率要比其他协议略高,HTTP 协议和 FTP 协议的漏判率为 0,即它们都能被准确识别,而 SMTP 协议和 PPLIVE 协议有少量数据不能识别。文中算法提取的协议特征在检测的过程中具有较低的错误率和误判率。

4 结束语

文中提出了一种适用于协议特征提取的多级 T+序列树挖掘算法。该方法通过创建多级 T+序列树、裁剪和构建投影 T+序列树等操作得到协议特征序列。选择了 4 种协议的数据流进行了有效测试,实验结果表明,该算法取得了较好的效果。下一步的工作重点是进一步提高文中算法的运行效率及对加密协议特征的挖掘分析。

参考文献:

[1] 杨丰瑞,吴 辉,张治中. 基于 DPI 技术 LTE-SI 接口流量识别系统的设计与实现[J]. 重庆邮电大学学报:自然科学版,2014,26(5):622-625.

[2] 杜瑞颖,杨 勇,陈 晶,等. 一种基于相似度的高效网络流量识别方案[J]. 山东大学学报:理学版,2014,49(9):109-114.

[3] 赵国锋,吉朝明,徐 川. Internet 流量识别技术研究[J]. 小型微型计算机系统,2010,31(8):1514-1520.

[4] Yoon Sung-Ho, Park Jun-Sang, Kim Myung-Sup. Signature maintenance for Internet application traffic identification using header signatures[C]//Proceedings of 2012 international conference on network operations and management. Maui:IEEE, 2012:1151-1158.

(上接第 70 页)

cine, 2011, 50(2):101-104.

[9] 贾 哲,黄志球,王珊珊,等. 支持本体推理的 P3P 隐私策略冲突检测研究[J]. 计算机科学与探索,2013,7(1):74-82.

[10] Guarino N, Masolo C, Vetere G. Ontoseek: content-based access to the web[J]. IEEE Intelligent Systems and Their Applications, 1999, 14(3):70-80.

[11] 郭文英. 基于 SWRL 推理的语义关联发现及其在本体映射与集成中的应用[D]. 杭州:浙江大学,2006.

[12] 王海林. SWRL 推理规则在平面几何证明中的应用[J]. 计

[5] Zhang Wen, Wang Heng. Identification of peer-to-peer traffic based on process fingerprint[C]//Proceedings of 2011 international conference on mechatronic science, electric engineering and computer. Jilin:IEEE, 2011:1559-1562.

[6] Du Jiang, Long Tao. P2P traffic identification research based on the SVM[C]//Proceedings of 2013 international conference on wireless and optical communication. Chongqing:IEEE, 2013:683-686.

[7] 张晓初,杨瑞君,吴伟航,等. 互联网流量采集分析系统设计与实现[J]. 计算机工程,2012,38(3):82-84.

[8] Park B, Won Y J, Kim M, et al. Towards automated application signature generation for traffic identification[C]//Proc of NOMS 2008. Salvador:IEEE, 2008:160-167.

[9] 龙 文,马 坤,辛 阳,等. 适用于协议特征提取的关联规则改进算法[J]. 电子科技大学学报,2010,39(2):302-305.

[10] Lin Guanzhou, Xin Yang, Yang Yixian. An application-level features mining algorithm based on PrefixSpan[C]//Proceedings of 2010 international conference on computer engineering and technology. Chengdu:IEEE, 2010:461-465.

[11] Jian Ding, Meng Han. Sequential pattern mining on highly similar and dense dataset[C]//Proceedings of 2013 international conference on computational and information sciences. Shiyang:IEEE, 2013:762-765.

[12] Liu Peiyu, Gong Wei, Jia Xian. An improved PrefixSpan algorithm research for sequential pattern mining[C]//Proceedings of 2011 international conference on IT in medicine and education. Cuangzhou:IEEE, 2011:103-108.

[13] 张 魏,刘 峰,腾少华. 改进的 PrefixSpan 算法及其在序列模式挖掘中的应用[J]. 广东工业大学学报,2013,30(4):49-54.

[14] 郭 超,李 坤,王永炎,等. 多核处理器环境下内存数据库索引性能分析[J]. 计算机学报,2010,33(8):1512-1522.

[15] 刘 勇,奚建清,黄东平,等. 图形处理器上内存数据库索引 T-树的研究[J]. 华南理工大学学报:自然科学版, 2013,41(3):22-28.

算机技术与发展,2010,20(9):218-221.

[13] 钱 凌. 一个基于本体和规则推理的查询系统的设计与实现[D]. 南京:东南大学,2006.

[14] Arnold F, Belinfante A, van der Berg F, et al. Computer safety, reliability, and security[M]. Berlin:Springer, 2013.

[15] 黄 凤. 基于描述逻辑的访问控制策略冲突检测方法研究[D]. 南京:南京航空航天大学,2010.

[16] Forgy C L. Rete: a fast algorithm for the many pattern/many object pattern match problem[J]. Artificial Intelligence, 1982, 19(1):17-37.

基于SWRL规则的本体推理研究

作者：[周亮](#)，[黄志球](#)，[倪川](#)，[ZHOU Liang](#)，[HUANG Zhi-qiu](#)，[NI Chuan](#)

作者单位：[南京航空航天大学 计算机科学与技术学院, 江苏 南京, 210016](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015(10)

引用本文格式：[周亮](#).[黄志球](#).[倪川](#).[ZHOU Liang](#).[HUANG Zhi-qiu](#).[NI Chuan](#) [基于SWRL规则的本体推理研究](#)[期刊论文]  
]-[计算机技术与发展](#) 2015(10)