

2023-2024 学年 知识工程（双语）实验报告

任课教师：吴天星

院 系 人工智能学院

专 业 人工智能

姓 名 蒋雨初

任 务 Knowledge Reasoning

1 实验三

1.1 实验任务

金融领域知识推理

1. 阅读程序 Demo_finance.java 源码，分别注释本体和规则的部分，观察推理结果的变化;
2. 撰写 Datalog 规则进行推理，观察新的推理结果 (对应 ” 实验课代码/src/main/resources/data/finance_rule.txt”):
 - (a) 如果 A 是 B 的子类，B 是 C 的子类，那么 A 是 C 的子类 (对应 inance data.nt 中的谓词”subClassOf”)
 - (b) 如果 A 的类型是PublicCompany，那么PublicCompany的任意父类也是 A 的类型 (对应 finance_data.nt 中的谓词 “type”)。

法律领域知识推理

1. 撰写 Datalog 规则进行推理，观察新的推理结果 (对应 ” 实验课代码/src/main/resources/data/legal_rule.txt”) :
 - (a) 如果案件 A 关联事件 B, 事件 B 的发生时间是案件 A 的关键节点 (对应 legal_data.nt 中的谓词”Relate” 和”Time”)

1.2 金融领域知识推理

1.2.1 Task 1

实验代码中对应于导入本体和规则的代码位于以下代码片段:

```
30 // 导入本体、实例数据与自定义规则
31 OWLOntologyManager manager = OWLManager.createOWLOntologyManager();
32 OWLOntology ontology =
    manager.loadOntologyFromOntologyDocument(ontologyFile);
33 store.importOntology(ontology);
34 store.importFiles(new File[] {dataFile});
35 store.importFiles(new File[] {ruleFile});
```

Code 1: Demo_finance.java 中对应导入规则和本体的代码片段

在注释掉本体与规则前，推理结果如下:

原有的三元组总数量为10个
推理过后的三元组总数量为24个
推理出来的三元组总数量为14个

Code 2: 同时导入本体和规则的推理结果

分别注释掉Code 1中第 33、35 行后，重新推理得到Code 3和Code 4。

原有的三元组总数量为10个
推理过后的三元组总数量为18个
推理出来的三元组总数量为8个

原有的三元组总数量为10个
推理过后的三元组总数量为16个
推理出来的三元组总数量为6个

Code 3: 不导入本体时的推理结果

Code 4: 不导入规则时的推理结果

对比注释导入本体前后的推理结果Code 2和Code 3，少了如下 6 条三元组：

```
<http://www.example.org/kse/finance#万达集团> rdf:type
  <http://www.example.org/kse/Company> .

<http://www.example.org/kse/finance#王健林> rdf:type
  <http://www.example.org/kse/Person> .
... 省略四条三元组
```

这是因为在本体文件 finance_onto.owl 中定义了 Object Property control。而 control 的 range 是 Company，domain 是 Person。control 的缺失使得与 Company 和 Person 相关的类型无法被推理出。

对比注释导入规则前后的推理结果Code 2和Code 4，少了如下 8 条三元组：

```
<http://www.example.org/kse/finance#融创中国>
  <http://www.example.org/kse/finance#conn_trans>
  <http://www.example.org/kse/finance#乐视网> .

<http://www.example.org/kse/finance#乐视网>
  <http://www.example.org/kse/finance#conn_trans>
  <http://www.example.org/kse/finance#融创中国> .

<http://www.example.org/kse/finance#孙宏斌>
  <http://www.example.org/kse/finance#hold_share>
  <http://www.example.org/kse/finance#融创中国> .
... 省略五条三元组
```

这是因为在规则文件 finance_rule.txt 中定义了两条规则 p:hold_share(?X,?Y):- p:control(?X,?Y). 和 p:conn_trans(?Y,?Z):- p:hold_share(?X,?Y), p:hold_share(?X,?Z).. 前者的

含义是若X control Y,则能推断出X hold_share Y。后者的含义是若X hold_share Y和X hold_share Z同时成立,那么可推断出Y conn_trans Z。例如对于孙宏斌control 融创中国,在第一条规则下可以推出孙宏斌hold_share 融创中国。再由孙宏斌hold_share 融创中国 孙宏斌hold_share 乐视网,可以用第二条规则推出 融创中国con_trans 乐视网。

1.2.2 Task 2

根据要求编写规则:

```
1 PREFIX p: <http://www.example.org/kse/finance#>
2 p:subClassOf(?X,?Z):- p:subClassOf(?X,?Y), p:subClassOf(?Y,?Z) .
```

推理结果如Code 5所示。

原有的三元组总数量为10个
推理过后的三元组总数量为25个
推理出来的三元组总数量为15个

原有的三元组总数量为10个
推理过后的三元组总数量为27个
推理出来的三元组总数量为17个

Code 5: 谓词 subClassOf 的 Datalog 代码

Code 6: 谓词 type 的 Datalog 代码

对比Code 2,多推理出了一条三元组。这是因为由PublicCompany SubClassOf Company和 Company SubClassOf Organization 推出了PublicCompany SubClassOf Organization。

如果 A 的类型是PublicCompany, 那么PublicCompany的任意父类也是 A 的类型, 实现如下:

```
1 PREFIX p: <http://www.example.org/kse/finance#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 rdf:type(?X,?Y):- rdf:type(?X,p:PublicCompany),
    p:subClassOf(p:PublicCompany,?Y) .
```

推理结果如Code 6所示。使用新的规则,从融创中国 type PublicCompany 与 PublicCompany SubClassOf Organization PublicCompany SubClassOf Company可以推出融创中国 type Company与融创中国 type Organization。

1.3 法律领域知识推理

在未引入所要求的规则时, 得到推理结果Code 7。

原有的三元组总数量为16个
推理过后的三元组总数量为23个
推理出来的三元组总数量为7个

原有的三元组总数量为16个
推理过后的三元组总数量为25个
推理出来的三元组总数量为9个

Code 7: 原始的法律领域推理结果

Code 8: 引入 Keypoints 后的推理结果

再编写如下规则, 推理出Code 8, 新增了两条三元组。

```
1 PREFIX p: <http://www.reason/legal#>
2 p:KeyPoints(?Z,?X) :- p:Relate(?X,?Y), p:Time(?Y,?Z) .
```

2 实验四

2.1 实验任务

1. 撰写规则，观察新的推理结果（对应“实验课代码二/src/main/resources/data/diagnosis_rule.txt”）：
 - (a) 已知 Pa 转换为 Kpa 的转换公式（1KPa=1000Pa），求设备的进出口压差为多少 Kpa？（对应 diagnosis_data.nt 中的谓词“进出口压差（Pa）”）
 - (b) 某冷凝设备进出口压差大于 20KPa，该冷凝设备存在“冷凝设备压差过大”故障。（对应 diagnosis_data.nt 中的谓词“进出口压差（KPa）”和“type”）
2. 撰写规则，观察新的推理结果（对应“实验课代码二/src/main/resources/data/finance_rule.txt”）：
 - (a) 一家类型不是劣迹企业的公司，并且四月份的股票价格高于三月份，则该公司是一家优质企业（注意，这道题的数据也需要仿照之前格式自己创造！对应 finance_data.nt）

2.2 Task 1

根据要求，如果已知某个实体 X 的进出口压差以 Pa 表示，那么可以通过将这个值除以 1000 来得到同一压差的 KPa。使用如下代码：

```
1 PREFIX p: <http://www.example.org/kse/diagnosis#>
2 p:进出口压差 (KPa) (?X,?Z) :- p:进出口压差 (Pa) (?X,?Y), BIND (?Y / 1000
AS ?Z) .
```

这里的 BIND 是一个操作符，用于创建新的变量 ?Z，其值为 ?Y / 1000，即将 Pa 转换为 KPa。

由Code 9可见多推出了两条三元组，即由冷凝器1 进出口压差(Pa)X 与冷凝器2 进出口压差(Pa)X 推出了冷凝器1 进出口压差(KPa)X与冷凝器2 进出口压差(KPa)X。

原有的三元组总数量为6个
推理过后的三元组总数量为11个
推理出来的三元组总数量为5个

原有的三元组总数量为6个
推理过后的三元组总数量为12个
推理出来的三元组总数量为6个

Code 9: 实现谓词“进出口压差 (KPa)”
后推理结果

Code 10: 实现谓词“故障”和“冷凝设备
压差过大”后的推理结果

要实现“某冷凝设备进出口压差大于 20KPa, 该冷凝设备存在‘冷凝设备压差过大’故障”, 只需要在之前的基础上再新增一行, 完整代码如下:

```
1 PREFIX p: <http://www.example.org/kse/diagnosis#>
2 p:进出口压差 (KPa) (?X,?Z) :- p:进出口压差 (Pa) (?X,?Y), BIND (?Y / 1000
   AS ?Z) .
3 p:故障(?X, p:冷凝设备压差过大) :- p:类型(?X,p:冷凝设备),
   p:进出口压差 (KPa) (?X,?Z), FILTER(?Z > 20) .
```

这里的 **FILTER** 函数用于应用条件, 确保只有当压差超过特定阈值时, 才会诊断出特定的故障类型, 最终得到Code 10。

新增的三元组冷凝器2 故障 冷凝设备压差过大是由
冷凝器2 进出口压差(KPa)>"30"^^xsd::integer推出的。

2.3 Task 3

根据要求, 需要创造数据, 撰写规则: 一家类型不是劣迹企业的公司, 并且四月份
的股票价格高于三月份, 则该公司是一家优质企业。设计本体结构如图 1所示, 并设定
各企业三、四月股价如图 2, 完整的文件内容如下所示:

```
<http://www.example.org/kse/finance#劣迹企业>
<http://www.example.org/kse/finance#subClassOf>
<http://www.example.org/kse/finance#企业> .
<http://www.example.org/kse/finance#优质企业>
  <http://www.example.org/kse/finance#subClassOf>
  <http://www.example.org/kse/finance#企业> .

<http://www.example.org/kse/finance#巴里阿阿>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://www.example.org/kse/finance#企业> .
<http://www.example.org/kse/finance#菊厂>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://www.example.org/kse/finance#企业> .
<http://www.example.org/kse/finance#菊厂>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://www.example.org/kse/finance#劣迹企业> .
```

```

<http://www.example.org/kse/finance#CloseAI>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://www.example.org/kse/finance#企业> .

<http://www.example.org/kse/finance#CloseAI>
  <http://www.example.org/kse/finance#三月份股价>
    "9"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://www.example.org/kse/finance#CloseAI>
  <http://www.example.org/kse/finance#四月份股价>
    "810"^^<http://www.w3.org/2001/XMLSchema#integer> .

<http://www.example.org/kse/finance#菊厂>
  <http://www.example.org/kse/finance#三月份股价>
    "41"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://www.example.org/kse/finance#菊厂>
  <http://www.example.org/kse/finance#四月份股价>
    "91"^^<http://www.w3.org/2001/XMLSchema#integer> .

<http://www.example.org/kse/finance#巴里阿阿>
  <http://www.example.org/kse/finance#三月份股价>
    "114"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://www.example.org/kse/finance#巴里阿阿>
  <http://www.example.org/kse/finance#四月份股价>
    "51"^^<http://www.w3.org/2001/XMLSchema#integer> .

```

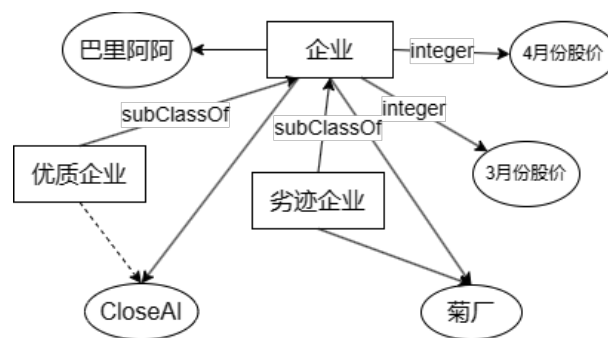


图 1: 简单的公司本体。其中圆圈代表实例或属性、矩形代表类。实线箭头代表已存在的关系，虚线箭头代表需要推出的关系。

推理结果如Code 11，可见正确地推出了预期的“CloseAI”是“优质企业”这一结论。

企业	三月份股价	四月份股价
巴里阿阿	114	51
菊厂	41	91
CloseAI	9	810

图 2: 为三家企业设定的三、四月份股价

原有的三元组总数量为**12**个
推理过后的三元组总数量为**13**个
推理出来的三元组总数量为**1**个

Code 11: 自定义数据的推理结果