

hw9

李晨昊 2017011466

2019-11-21

目录

1	Exercise 12.4.1	1
2	Exercise 12.5.1	2
2.1	2
2.2	2
3	Exercise 12.7.1	3

1 Exercise 12.4.1

h 和 g 用于表示新创建对象的标号，它们不是代码的一部分，你可以假设类型为 T 的对象有一个字段 f 。使用本节中的 Datalog 规则来推导出所有可能的 pts 和 $hpts$ 事实。

```
h: T a = new T();
g: T b = new T();
T c = a;
a.f = b;
b.f = c;
T d = c.f;
```

$pts(a, h) :- \text{“}h: T a = new T()\text{”}$

$pts(b, g) :- \text{“}g: T b = new T()\text{”}$

$pts(c, h) :- \text{“}c = a\text{”} \ \& \ pts(a, h)$

$hpts(h, f, g) :- \text{“}a.f = b\text{”} \ \& \ pts(a, h) \ \& \ pts(b, g)$

$hpts(g, f, h) :- \text{“}b.f = c\text{”} \ \& \ pts(b, g) \ \& \ pts(c, h)$

$\text{pts}(d, g) \text{ :- "d = c.f" \& pts(c, h) \& hpts(h, f, g)}$

2 Exercise 12.5.1

1. 构造 EDB 关系 actual, formal 和 cha
2. 推导出所有可能的 pts 和 hpts 事实

```
class t {  
1)g:  t n() { return new r(); }  
}  
class s extends t {  
2)h:  t n() { return new s(); }  
}  
class r extends s {  
3)i:  t n() { return new r(); }  
}  
main() {  
4)j:  t a = new t();  
5)  a = a.n();  
}
```

以下用字母编号代表对象编号，数字编号代表程序点和方法。以下约定隐藏的 `this` 是第零个参数。

2.1

$\text{actual}(5, 0, a)$

$\text{formal}(1, 0, \text{this}@1); \text{formal}(2, 0, \text{this}@2); \text{formal}(3, 0, \text{this}@3)$

$\text{cha}(t, n, 1); \text{cha}(s, n, 2); \text{cha}(r, n, 3)$

2.2

$\text{pts}(\text{return}@1, g) \text{ :- } g: \text{return new r()}"$

$\text{pts}(\text{return}@2, h) \text{ :- } \text{"h: return new s()}"$

$\text{pts}(\text{return}@3, i) \text{ :- } \text{"i: return new r()}"$

$\text{pts}(a, j) \text{ :- } \text{"j: t a = new t()}"$

$\text{invokes}(5, 1) \text{ :- } \text{"5) a.n()"} \& \text{pts}(a, j) \& \text{hType}(j, t) \& \text{cha}(t, n, 1)$

```

pts(this@1, j) :- invokes(5, 1) & formal(1, 0, this@1) & actual(5, 0, a) & pts(a, j)

// 本条推理采用的规则并不存在。虽然书中说返回值可以用赋值来体现，但是没有给出对应的推理规则，这是我自己设想的一种比较合理的规则

pts(a, g) :- “5) a = a.n()” & invokes(5, 1) & pts(return@1, g)

pts(this@1, g) :- invokes(5, 1) & formal(1, 0, this@1) & actual(5, 0, a) & pts(a, g)

invokes(5, 3) :- “5) a.n()” & pts(a, g) & hType(g, r) & cha(r, n, 3)

pts(this@3, g) :- invokes(5, 3) & formal(3, 0, this@3) & actual(5, 0, a) & pts(a, g)

pts(this@3, j) :- invokes(5, 3) & formal(3, 0, this@3) & actual(5, 0, a) & pts(a, j)

pts(a, i) :- “5) a = a.n()” & invokes(5, 3) & pts(return@3, i)

pts(this@1, i) :- invokes(5, 1) & formal(1, 0, this@1) & actual(5, 0, a) & pts(a, i)

pts(this@3, i) :- invokes(5, 1) & formal(1, 0, this@1) & actual(5, 0, a) & pts(a, j)

```

3 Exercise 12.7.1

使用例子 12.28 的符号编码方式，生成一个 BDD 来表示由元组 (b, b), (c, a) 和 (b, a) 组成的关系。你可以用任意的方式对布尔变量进行排序，以获取最简洁的 BDD。

设 a 的编码为 00，b 的编码为 01，**c 的编码为 11**。使用布尔变量 wx 对第一个分量进行编码，使用布尔变量 yz 对第二个分量进行编码，则关系可以表示为：

y	x	w	z
0	1	0	1
0	1	1	0
0	1	0	0

