《软件分析与验证》第二次书面作业讲解

谢兴宇

清华大学

2020 年 4 月

Contents

Contents

第一题

题目

用下划线标出下列公式中所有自由的“变元出现”。

$$\forall x.(f(x) \wedge (\exists y.g(x, y, z))) \wedge (\exists z.g(x, y, z))$$

第一题

题目

用下划线标出下列公式中所有自由的“变元出现”。

$$\forall x.(f(x) \wedge (\exists y.g(x, y, z))) \wedge (\exists z.g(x, y, z))$$

出现：被置于某个公式的某个位置。
一阶逻辑连接词优先级：
$“\neg” > “\wedge” > “\vee” > “\rightarrow” > “\leftrightarrow” > “\forall” = “\exists”$

第一题

题目

用下划线标出下列公式中所有自由的“变元出现”。

$$\forall x.(f(x) \land (\exists y.g(x, y, z))) \land (\exists z.g(x, y, z))$$

出现：被置于某个公式的某个位置。
一阶逻辑连接词优先级：
“$\neg$” $>$ “ $\land$ ” $>$ “ $\lor$ ” $>$ “ $\rightarrow$ ” $>$ “ $\leftrightarrow$ ” $>$ “$\forall$” $=$ “$\exists$”
答案：

$$\forall x.(f(x) \land (\exists y.g(x, y, \underline{z}))) \land (\exists z.g(x, \underline{y}, z))$$

第二题

题目

下列哪些问题或理论是可判定的？

1. 命题逻辑公式的有效性
2. 一阶逻辑公式的有效性
3. T_{E}
4. $T_{\mathbb{N}}$
5. T_{A} 的无量词片段

第二题

有效性的判定

1. 命题逻辑：枚举每一个命题变元所取真值
2. 一阶逻辑：Turing & Church
3. T_{E}：任一 T_{E} 中的公式自是一阶逻辑；任一 FOL 中的公式，将 $=$ 换成另一个新谓词，便可得到一个等价的 T_E 中的公式。故（T_{E} 可判定）当且仅当（FOL 可判定）。
4. $T_{\mathbb{N}}$：Presburger
5. T_{A} 的无量词片段：枚举公式中数组、下标和取值均有限，故枚举（数组，下标）的值即可

可判定性：总结

Theory	Description	Full	QFF
FOL	一阶逻辑	no	yes
T_E	带等词的一阶逻辑	no	yes
T_{PA}	Peano 算术	no	no
$T_{\mathbb{N}}$	Presburger 算术	yes	yes
$T_{\mathbb{Z}}$	线性整数	yes	yes
T_A	数组	no	yes

表: 理论和其无量词片段的可判定性

第三题

题目

请举出一个不是同余关系（congruence relation）的等价关系

第三题

题目

请举出一个不是同余关系（congruence relation）的等价关系

定义：给定集合 S 和 S 上的函数集 F，若二元关系 R 满足：对于函数集 F 中的任意函数 f，记 f 的元数为 n，在 S 中任取 $s_1, \cdots s_n, t_1, \cdots, t_n$，

$$\text{若 } s_1 R t_1, \cdots, s_n R t_n \text{，则 } f(s_1, \cdots, s_n) = f(t_1, \cdots, t_n)$$

，则称 R 是一个**同余关系**。

第三题

题目

请举出一个不是同余关系（congruence relation）的等价关系

定义：给定集合 S 和 S 上的函数集 F，若二元关系 R 满足：对于函数集 F 中的任意函数 f，记 f 的元数为 n，在 S 中任取 $s_1, \cdots s_n, t_1, \cdots, t_n$，

$$\text{若 } s_1 R t_1, \cdots, s_n R t_n \text{，则 } f(s_1, \cdots, s_n) = f(t_1, \cdots, t_n)$$

，则称 R 是一个**同余关系**。
参考答案：考虑 $\mathbb{Z}$ 和其上的二元关系 $\equiv_2$

$$m \equiv_2 n \text{ iff } m \equiv n \pmod 2$$

和函数 $f(m) := \lfloor \frac{m}{2} \rfloor$，则 $\equiv_2$ 对于函数集 $\{f\}$ 而言不是同余关系。

第四题

题目

找到两个不同的等价关系，使得其中一个 refine 另一个。

第四题

题目

找到两个不同的等价关系，使得其中一个 refine 另一个。

给定两个集合 S 上的二元关系 R_1 和 R_2，对于任意的 $s_1, s_2 \in S$ 使得 $s_1 R_1 s_2$，都有 $s_1 R_2 s_2$，我们便称 R_1 *refines* R_2。

第四题

题目

找到两个不同的等价关系，使得其中一个 refine 另一个。

给定两个集合 S 上的二元关系 R_1 和 R_2，对于任意的 $s_1, s_2 \in S$ 使得 $s_1 R_1 s_2$，都有 $s_1 R_2 s_2$，我们便称 R_1 *refines* R_2。

参考答案：取 $S := \{0, 1\}$，S 上的两个二元关系 $R_1 := \{\langle 0, 0\rangle, \langle 1, 1\rangle\}$，$R_2 := \{\langle 0, 0\rangle, \langle 0, 1\rangle, \langle 1, 0\rangle, \langle 1, 1\rangle\}$，则 R_1 refines R_2。

第五题

Problem

In the congruence closure algorithm, subterms of a formula are represented by DAGs. Which term does the figure represents? Write it out in a formulaic way.

题目

在同余闭包算法中，一个公式的所有子项可以被表示为一个有向无环图。请写出右图表示的项。

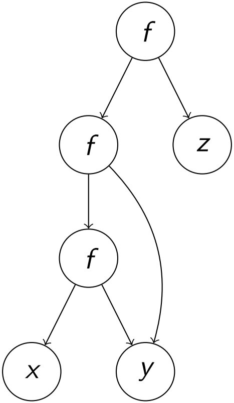


图: Subterms of a term

第五题

子项关系可以用一个特殊的 DAG 来表示：

- 每一个节点上都有一个标记，零出度节点标有一个变元、常元或零元函数，非零出度节点标有一个非零元函数。
- 同一个节点的出边是有序的。
- 不同零出度节点的标记不同；不同的非零出度节点，或者标记不同，或者出边的数量不同，或者第 i 条出边的终点不同。

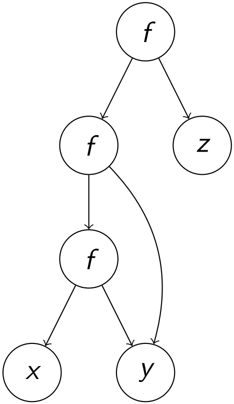


图: Subterms of a term

第五题

每一个节点与一个子项一一对应，按逆拓扑序来定义：

- 标有变元或常元的节点所对应的子项便是其自身的标记。
- 标有函数符的节点所对应的子项是，将其被标记的函数（依序作用于其每一条出边的终点对应的子项）得到的子项。

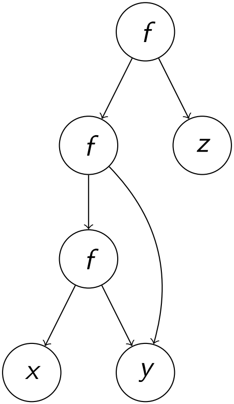


图: Subterms of a term

第五题

$$
\begin{array}{ll}
6: & z \\
5: & y \\
4: & x \\
3: & f(x, y) \\
2: & f(f(x, y), y) \\
1: & f(f(f(x, y), y), z)
\end{array}
$$

答案: $f(f(f(x, y), y), z)$

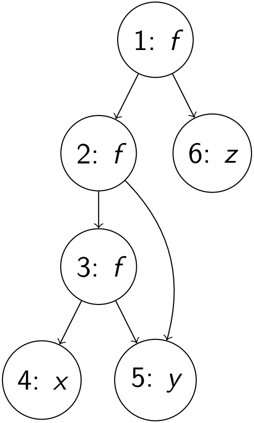
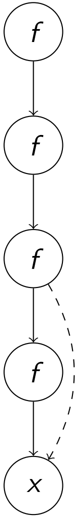


图: Subterms of a term

第六题

题目

右图是一个同余闭包算法执行过程中的 DAG，虚线表示一次合并操作。从图中你能推断出哪些同余类？

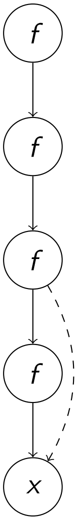


同余闭包算法

Step 1. 找到 CNF F 中的所有子项，记其为 S_F。
Step 2. 初始时，每个子项都属于一个仅包含其自身的同余类，s 所在的同余类记为 $[s]$。
Step 3. 据 F 中形如 $s = t$ 的原子公式合并 $[s]$ 和 $[t]$。
Step 4. **(function congruence)** 检查公式中出现的所有函数符 f，若以下条件成立：

- $f(s_1, \cdots, s_n) = f(t_1, \cdots, t_n)$
- $[f(s_1, \cdots, s_n)] = [f(t_1, \cdots, t_n)]$
- $[s_1] = [t_1], \cdots, [s_n] = [t_n]$
- $f(s_1, \cdots, s_n), f(t_1, \cdots, t_n) \in S_F$

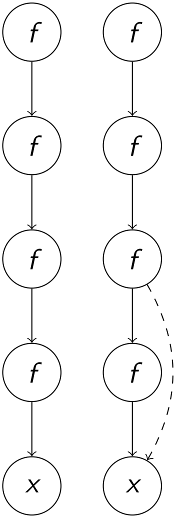
，则将 $[f(s_1, \cdots, s_n)]$ 与 $[f(t_1, \cdots, t_n)]$ 合并。不断重复此步骤直到找不到新的同余类合并。
Step 5. 若 F 中有形如 $\neg s = t$ 的原子公式，但 $[s] \neq [t]$，说明 F 不可满足。



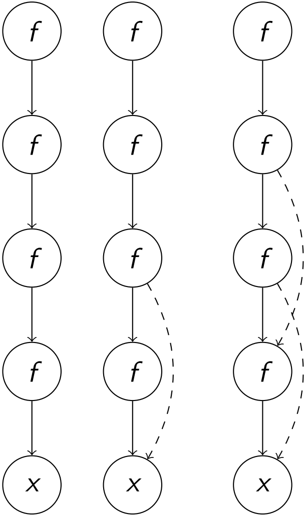
第六题

$$\begin{tikzcd}
f \arrow[d] \\
f \arrow[d] \\
f \arrow[d] \\
f \arrow[d] \\
x
\end{tikzcd}$$

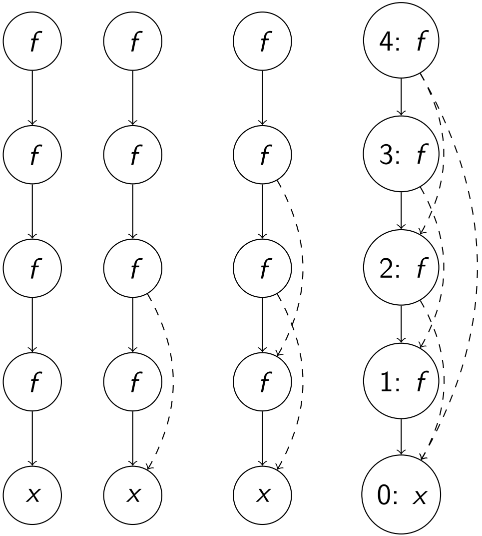
第六题



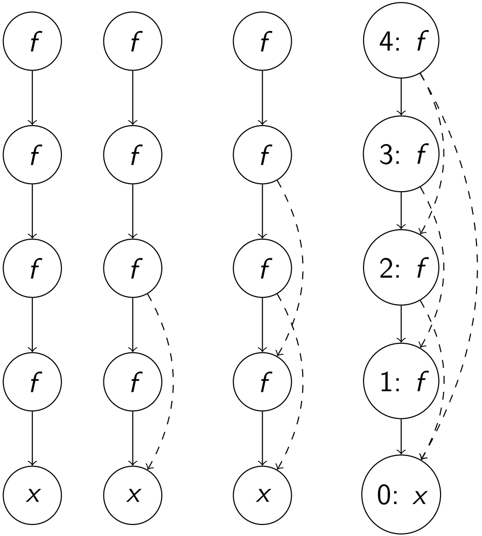
第六题



第六题



第六题



最终，我们找到了两个同余类：

$\{x, f(x), f(f(f(f(x))))\}$

$\{f(x), f(f(f(x)))\}$

Contents

题目大意

用 Dafny 证明定义

```
datatype Nat = Zero | Succ(n: Nat)
```

满足 Peano 算术中除归纳公理外的其他公理。

Peano 算术

历史：1860 年代 Hermann Grassmann 首次发现了公理化自然数的巨大潜能，1889 年 Dedekind 和 Peano 完成了后续工作。

Peano 算术

历史：1860 年代 Hermann Grassmann 首次发现了公理化自然数的巨大潜能，1889 年 Dedekind 和 Peano 完成了后续工作。

$\Sigma_{\mathsf{PA}}: \{0, 1, +, \times, =\}$

- *Zero:* $\forall x.\ \neg(x+1=0)$
- *Additive identity:* $\forall x.\ x+0=x$
- *Times zero:* $\forall x.\ x\times 0=0$
- *Successor:* $\forall x, y.\ (x+1=y+1)\to x=y$
- *Plus successor:* $\forall x, y.\ x+(y+1)=(x+y)+1$
- *Times successor:* $\forall x, y.\ x\times(y+1)=x\times y+x$
- *Induction:* $\forall F.(F[0]\wedge(\forall x.F[x]\to F[x+1]))\to\forall x.F[x]$

Peano 算术

历史：1860 年代 Hermann Grassmann 首次发现了公理化自然数的巨大潜能，1889 年 Dedekind 和 Peano 完成了后续工作。

$\Sigma_{\mathsf{PA}} : \{0, 1, +, \times, =\}$

- *Zero:* $\forall x.\ \neg(x + 1 = 0)$
- *Additive identity:* $\forall x.\ x + 0 = x$
- *Times zero:* $\forall x.\ x \times 0 = 0$
- *Successor:* $\forall x, y.\ (x + 1 = y + 1) \to x = y$
- *Plus successor:* $\forall x, y.\ x + (y + 1) = (x + y) + 1$
- *Times successor:* $\forall x, y.\ x \times (y + 1) = x \times y + x$
- *Induction:* $\forall F.(F[0] \land (\forall x.F[x] \to F[x + 1])) \to \forall x.F[x]$

然而有“非标准模型”满足上述公理，如果想真正刻画“自然数”，还需要：

$$\forall x.x = 0 \lor (\bigvee_{n \in \mathrm{dom}(PA)} x = \underbrace{1 + 1 + \cdots + 1}_{n \text{ many}})$$

Dafny

```
datatype Nat = Zero | Succ(n: Nat)
function one(): Nat { Succ(Zero) }
function add(x: Nat, y: Nat): Nat {
  match(x) {
    case Zero => y
    case Succ(n) => Succ(add(n, y))
  }
}
function mult(x: Nat, y: Nat): Nat {
  match(x) {
    case Zero => Zero
    case Succ(n) => add(mult(n, y), y)
  }
}
```

一个证明思路

Zero: $\forall x.\ \neg(x+1=0)$, Trivial
Additive identity: $\forall x.\ x+0=x$, **对 x 归纳**
Times zero: $\forall x.\ x\times 0=0$, **对 x 归纳**
Successor: $\forall x,y.\ (x+1=y+1)\rightarrow x=y$, Trivial
Plus successor: $\forall x,y.\ x+(y+1)=(x+y)+1$, **对 x 归纳**
Times successor: $\forall x,y.\ x\times(y+1)=x\times y+x$
先证明

$$\forall x,y.x+y=y+x$$

和

$$\forall x,y,z.(x+y)+z=(x+z)+y$$

Contents

答疑环节

欢迎提问！