形式化方法期中考试（1）2020 秋

期中试题及参考答案

1. **证明：**

$$(P \wedge Q) \vdash P \to Q$$

证明：

```
---------------------- (var)
  (P/\Q), P ⊢ P/\Q
---------------------- (/\ E2)
   (P/\Q), P ⊢ Q
---------------------- (→ I)
   (P/\Q) ⊢ P→Q
```

2. **证明：**

$$\vdash (P \vee Q) \to (\neg Q \to P)$$

证明：

```
                                                               --------------------- (var) -
                                                                 (P\/Q), ~Q, Q ⊢ Q
                                                               -----------------------------
                                                                                (P\/Q), ~Q,
---------------------- (var)  ----------------------- (var) -----------------------------
 (P\/Q), ~Q ⊢ P\/Q               (P\/Q), ~Q, P ⊢ P                                (P\/Q), ~(
-------------------------------------------------------------------------------------------
                                          (P\/Q), ~Q ⊢ P
-------------------------------------------------------------------------------------------
```

(P\/Q) ⊢ (~Q→P)

- -

⊢ (P\/Q) → (~Q→P)

3. **证明：存在无理数 p，q 使得 p^q 是有理数**

证明：令 $p = q = \sqrt{2}$，则 $p^q = (\sqrt{2})^{\sqrt{2}}$

考虑下面两种情况：

1. $(\sqrt{2})^{\sqrt{2}}$ 是有理数，证毕。
2. $(\sqrt{2})^{\sqrt{2}}$ 是无理数，令 $p = (\sqrt{2})^{\sqrt{2}}, q = \sqrt{2}$，则 $p^q = ((\sqrt{2})^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^2 = 2$，证毕。

这是按照构造主义来证明的吗？谈谈你的看法

不是，构造主义要求必须找出具体的值来，而证明并没有构造一个值。

4. **存在命题**

$$F = \neg(p \to (q \land (\neg p \to q)))$$

消除蕴含式

$$\begin{aligned} C(F) &= C(\neg(p \to (q \land (\neg p \to q)))) \\ &= \neg(C(p \to (q \land (\neg p \to q)))) \\ &= \neg(\neg C(p) \lor C(q \land (\neg p \to q))) \\ &= \neg(\neg p \lor (C(q) \land C(\neg p \to q))) \\ &= \neg(\neg p \lor (q \land (\neg C(\neg p) \lor C(q)))) \\ &= \neg(\neg p \lor (q \land (\neg\neg p \lor q))) \end{aligned}$$

转换为 NNF

$$F = \neg(\neg p \lor (q \land (\neg\neg p \lor q)))$$

$$\begin{aligned} C(F) &= C(\neg(\neg p \lor (q \land (\neg\neg p \lor q)))) \\ &= C(\neg(\neg p)) \land C(\neg(q \land (\neg\neg p \lor q))) \\ &= p \land (C(\neg q) \lor C(\neg(\neg\neg p \lor q))) \\ &= p \land (\neg q \lor (C(\neg\neg\neg p) \land C(\neg q))) \\ &= p \land (\neg q \lor (\neg p \land \neg q)) \end{aligned}$$

转换为 CNF

$$F = p \land (\neg q \lor (\neg p \land \neg q))$$

$$\begin{aligned} C(F) &= C(p \land (\neg q \lor (\neg p \land \neg q))) \\ &= C(p) \land C(\neg q \lor (\neg p \land \neg q)) \\ &= p \land (D(C(\neg q), C(\neg p \land \neg q))) \\ &= p \land (D(\neg q, C(\neg p \land \neg q))) \\ &= p \land (D(\neg q, \neg p \land \neg q)) \\ &= p \land (D(\neg q, \neg p) \land D(\neg q, \neg q)) \\ &= p \land (\neg q \lor \neg p) \land (\neg q \lor \neg q) \\ &= p \land (\neg q \lor \neg p) \land \neg q \end{aligned}$$

5.

```
1  DPLL(F) {
2      newF = BCP(F)
3      if(newF == TRUE)
4          return sat;
5      if(newF == FALSE)
6          return unsat;
7
8      x = select_var(newF);
9      if(DPLL(newF[x -> TRUE]))
10         return sat;
11     return DPLL(newF[x -> FALSE])
12 }
```

The `BCP()` method stands for **Boolean Constraint Propagation**, which is based on unit resolution. Unit resolution deals with one unit clause, which must be p or ~p, and one clause contains the negation of the unit clause.

Suppose we call the function `DPLL()` with the following proposition `F`:

$$F = (\neg p_1 \lor p_3) \land (\neg p_2 \lor p_3 \lor p_4) \land (p_1 \lor \neg p_3 \lor \neg p_4) \land (p_1)$$

For the first recursive call, on line 2 of `DPLL()`*, what's the value for* `newF`*?*

$$newF = p_3 \land (\neg p_2 \lor p_3 \lor p_4)$$

For the first recursive call, which variable you'll choose at line 8 of the `DPLL()` *function?*

> 选择了 p_1，因为 p_1 在整个命题的最后子命题中作为原子命题存在。

What's the final result for the function `DPLL()`? Is the proposition `F` satisfiable or not?

> `DPLL` 的最终结果代表 F 是否有使其能够成立的取值。对于本题而言 F 是可以满足的。

6.

- Alice 必须坐在一个凳子上：

$$A = (A_1 \wedge \neg A_2 \wedge \neg A_3) \vee (\neg A_1 \wedge A_2 \wedge \neg A_3) \vee (\neg A_1 \wedge \neg A_2 \wedge A_3)$$

- Bob 必须坐在一个凳子上：

$$B = (B_1 \wedge \neg B_2 \wedge \neg B_3) \vee (\neg B_1 \wedge B_2 \wedge \neg B_3) \vee (\neg B_1 \wedge \neg B_2 \wedge B_3)$$

- Carol 必须坐在一个凳子上：

$$C = (C_1 \wedge \neg C_2 \wedge \neg C_3) \vee (\neg C_1 \wedge C_2 \wedge \neg C_3) \vee (\neg C_1 \wedge \neg C_2 \wedge C_3)$$

- 第一个椅子上只能坐一个人：

$$F_1 = (A_1 \wedge \neg B_1 \wedge \neg C_1) \vee (\neg A_1 \wedge B_1 \wedge \neg C_1) \vee (\neg A_1 \wedge \neg B_1 \wedge C_1)$$

- 第二个椅子上坐一个人：

$$F_2 = (A_2 \wedge \neg B_2 \wedge \neg C_2) \vee (\neg A_2 \wedge B_2 \wedge \neg C_2) \vee (\neg A_2 \wedge \neg B_2 \wedge C_2)$$

- 第三个椅子上只能坐一个人：

$$F_3 = (A_3 \wedge \neg B_3 \wedge \neg C_3) \vee (\neg A_3 \wedge B_3 \wedge \neg C_3) \vee (\neg A_3 \wedge \neg B_3 \wedge C_3)$$

- Alice 不能坐在 Carol 旁边：

$$F_4 = (A_1 \rightarrow \neg C_2) \wedge (A_2 \rightarrow \neg(C_1 \vee C_3)) \wedge (A_3 \rightarrow \neg C_2)$$

> **注：** 本答案假设1在最左边，3在最右边。（左）1 ---- 2 ---- 3 （右）

- Bob不能坐在 Alice 右边：

$$F_5 = (B_1 \rightarrow \neg A_2) \wedge (B_1 \rightarrow \neg A_3) \wedge (B_2 \rightarrow \neg A_3)$$

则有约束条件如下：

$$F = A \wedge B \wedge C \wedge F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge F_5$$

7.

证明：

$$\forall x.\,(P(x) \to Q(x)) \vdash \forall x.\,(P(x)) \to \forall x.\,(Q(x))$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.\,(P(x)\to Q(x)), \forall x.\,(P(x)),\ x \vdash \forall x.\,(P(x)\to Q(x)).}\,(var)}{\forall x.\,(P(x)\to Q(x)), \forall x.\,(P(x)),\ x \vdash P(x)\to Q(x).}\,(\forall E) \quad \dfrac{\dfrac{}{\forall x.\,(P(x)\to Q(x)), \forall x.\,P(x),\ x \vdash \forall x.\,P(x).}\,(var)}{\forall x.\,(P(x)\to Q(x)), \forall x.\,(P(x)),\ x \vdash P(x).}}{\dfrac{\forall x.\,(P(x)\to Q(x)), \forall x.\,(P(x)),\ x \vdash Q(x).}{\forall x.\,(P(x)\to Q(x)), \forall x.\,(P(x)) \vdash \forall x.\,(Q(x)).}\,(\forall I)}\,(\to E)}{\forall x.\,(P(x)\to Q(x)) \vdash \forall x.\,(P(x)) \to \forall x.\,(Q(x)).}\,(\to I)$$

8.

$$F = \exists x.\,(P(y,x) \wedge \forall y.\,(\neg Q(y,x)) \vee P(y,z))$$

- 自由变量： $\{y,\ z\}$
- 绑定变量： $\{x,\ y\}$

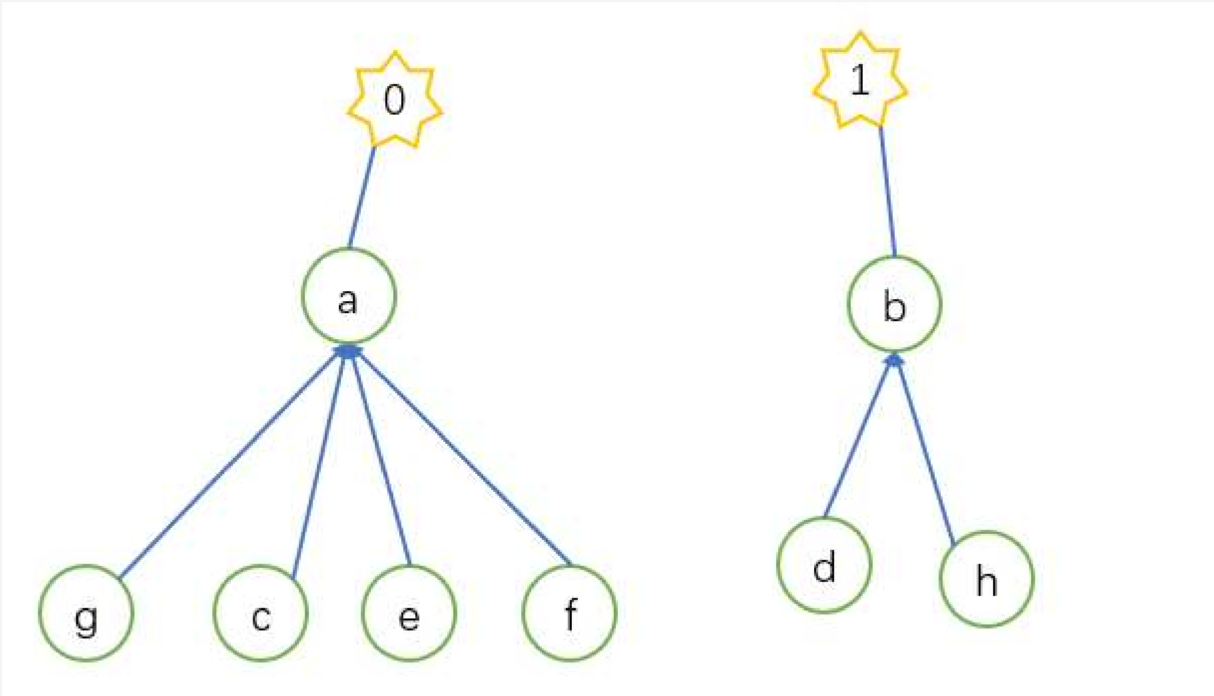
$$\begin{aligned} F[y \mapsto x] &= \exists x.\,(P(y,x) \wedge \forall y.\,(\neg Q(y,x)) \vee P(y,z))[y \mapsto x] \\ &= \exists t.\,(P(y,t) \wedge \forall y.\,(\neg Q(y,t)) \vee P(y,z))[y \mapsto x] \\ &= \exists t.\,(P(y,t) \wedge \forall s.\,(\neg Q(s,t)) \vee P(y,z))[y \mapsto x] \\ &= \exists t.\,(P(x,t) \wedge \forall s.\,(\neg Q(s,t)) \vee P(x,z)) \end{aligned}$$

考虑到 x 是绑定变量，因此 $F[x \mapsto R(y,z)]$ 的结果应该保持不变。

9.

$$a = c, b = d, a = e, d = h, f = g, g = e$$

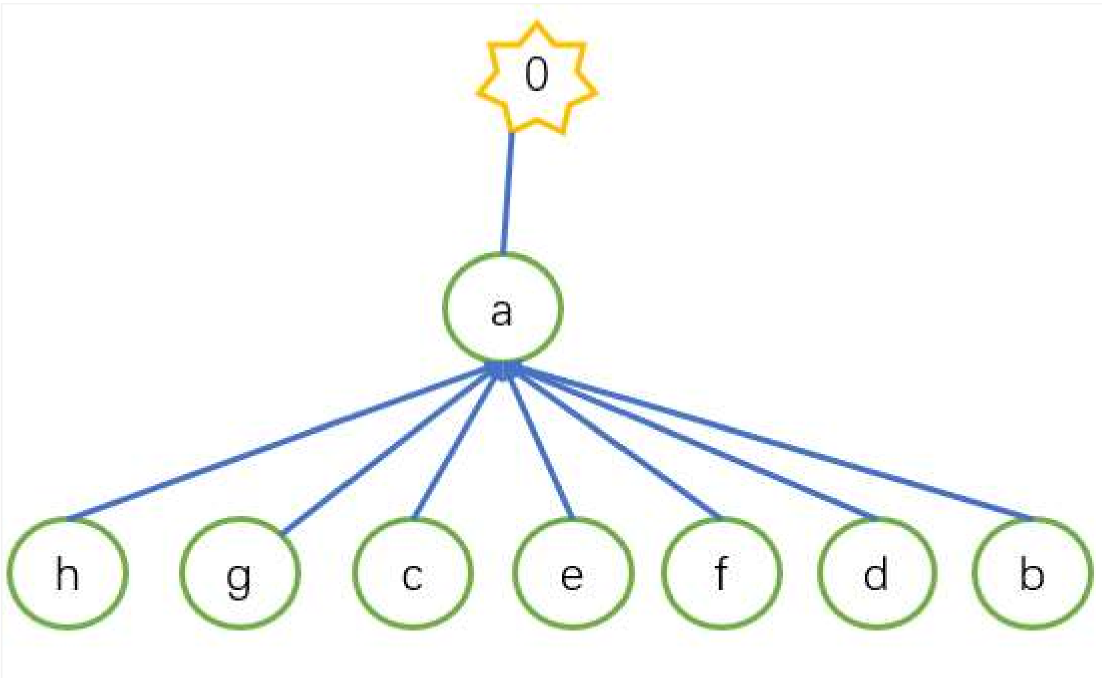
If we use the Union-Find algorithm to solve these qualities, what does the data structure look like? Please draw some sketches. (You don't need to write down the algorithm details.)



Suppose we add another equality

$$e = d$$

into this group, what does the data structure look like, after we use union-find algorithm to solve these equalities?



10.

```
int calculate_a(int in_1, int in_2) {
```

```
    int out_a_1 = in_1 * in_2;
    int out_a_2 = in_1 + in_2;
    int out_a = out_a_1 - out_a_2;
    return out_a;
}

int calculate_b(int in_1, int in_2) {
    int out_b = (in_1 * in_2) - (in_1 + in_2);
    return out_b;
}
```

Please describe the basic idea to prove these two algorithms are equivalent, by using `EUF` *theory. Please write down the logical proposition* `F` *you need to prove.*

基本原则就是：证明对于任意的相同输入，两者产生相同的输出。由于输入的范围过大，无法构造证明，因此，只需要证明其反命题是不可满足的即可。

$$
\begin{aligned}
F ::= \quad & (outa1 = f(in1, in2) \\
\wedge \; & outa2 = h(in1, in2) \\
\wedge \; & outa = g(outa1, outa2) \\
\wedge \; & g(f(in1, in2), h(in1, in2)) = outb) \rightarrow (outa = outb)
\end{aligned}
$$

> f 代表乘法操作，h 代表加法操作，g 代表减法操作。

Bob wants to prove the above proposition `F`*, by using the Z3 solver, the code he wrote looks like:*

```
solver = Solver()
solver.add(F)
print(solver.check())
```

这个代码无法证明两者等价。正确代码如下：

```
solver = Solver()
solver.add(Not(F))
print(solver.check())
```