1. 交通灯管控系统（可信软件）
2. 系统装备四个传感器：发信号（on/off）
    a. 交通灯两个状态：红、绿
    b. 在什么场景下工
    c. ……满足用户要求
3. 系统两个约束( FUN是功能需求 )
    a. 墙顶是什么东西（两个交通灯，交通灯的位置，司机遵守交通规则）： FUN-1
    b. 岛上、桥上的数量不能超过一定的数量 : FUN-2
    c. 桥只有一条路 : FUN-3
4. 系统的假设
    a. EQP-1：系统有两个交通灯
    b. EQP-2
    c. EQP-3：司机遵守交通灯的指示
    d. EQP-4：车开、车停
    e. EQP-5：四个传感器

1. The refinement strategy：建立模型的目的是证明是与需求对应的，建立模型是为了做精化
2. 逐步精化最后一个模型才符合需求
3. 两个功能，第一个功能在第一个模型中实现，越抽象越好
    a. 每次迭代覆盖一个功能
    b. 第二个模型就满足两个需求
    c. 逐步引入传感器和交通灯
    d. 证明系统满足需求
4. 今早发现问题，

1. 建立模型
2. 声明变量
    a. constant常量：上限（content d） axm:常量的性质

i. axm0_1：d 属于N(自然数) [0是第一版模型，1是第一个性质]

b. variable： n (inv：变量的性质)

i. inv0_1：n 属于 N

ii. inv0_2：n <= d

3. 声明事件

a. ML_out: 大陆到岛上

i. n := n + 1

b. ML-in：岛上到大陆

i. n := n - 1

4. 事件看成一个谓词

5. ML out / inv0 1 / INV

a. ML out / inv0_1 / INV

a. 事件、不变式、证明义务

6. 系统一直运行下去，检查模型没有死锁

a. 所有的事件都不能运行，即死锁

b. 只要有一个运行就不是死锁

7. 证明义务：模型，guard

8. 新的事件不能发散：新增加的事件不能没完没了的改变（must not diverge）

a. NAT：

b. VAR：V >= 0，V(n) > V(n')，需要自己找出V的存在（有时候找不到V的形式，但是可以证明V是存在的）

i. 2a + b

c. 严格单调减

9. 证明系统没有死锁

a. 精化之后相对没有死锁

b. 抽象模型下做到某一个点可以往下做，具体模型也可以继续往下做

1. 继续精化：添加交通灯

2. SIM：继续精化

a. 增加变量，保留原来的变量

b. 具体变量，抽象变量 V 与 W 不相交

3. 1

4. 1

5. 1

6.