第2次实验

实验目的

这次实验的目的是使用NuSMV实现互斥的First Attempt

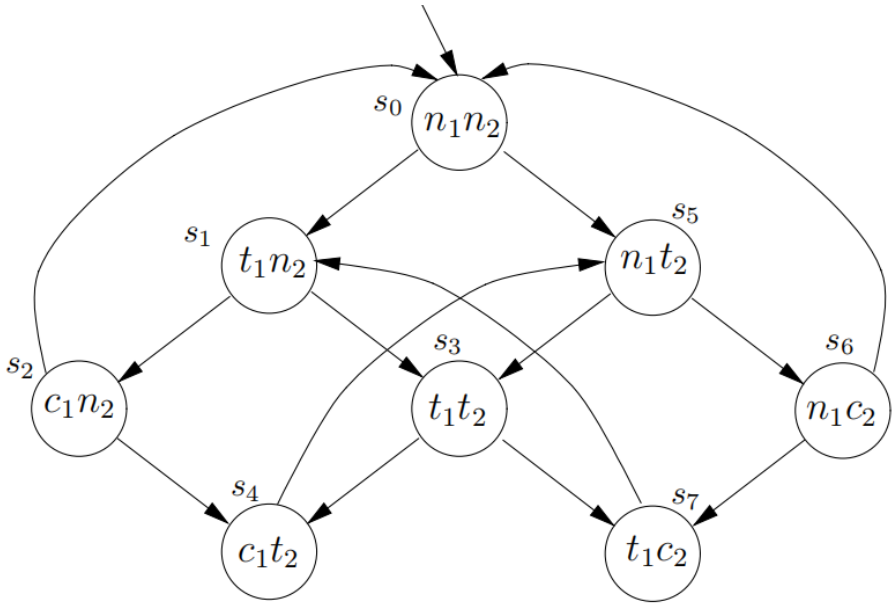

Figure 3.7. A first-attempt model for mutual exclusion.

分析

First Attempt和后两次的Attempt的最大区别就是turn，即不区分不同的t_1t_2
这样会导致pr1=t时，可能会有pr2 n->t->c->n的循环，不满足Liveness

程序设计

与Attempt3不同点主要在于没有turn，只要满足$st = t \wedge other! = c$，st就可以变为c
还有几个细微的不同点：

- c_1t_2节点没有指向自己的圈，说明当状态为c时，下一个状态一定是n，而不会一直停留在c
- 正是由于上面的原因，我们不再需要FAIRNESS !(st = c)

prc设计如下：

```
MODULE prc(other-st)
VAR
st: {n, t, c};
ASSIGN
init(st) :=n;
next(st) :=
case
    (st = n): t;
    (st = t) & (other-st != c) :c;
    (st = c): n;
    TRUE :st;
esac;

FAIRNESS running
```

CTL验证

non-blocking

即对于每个满足n_1的状态，存在后继状态，满足t_1。
用CTL表示：
$AG(n_1 \rightarrow (EX\ t_1))$
用NuSMV表示：
`CTLSPEC AG( (pr1.st = n) -> (EX pr1.st = t) )`

no strict sequencing

即存在一条路径，具有满足c_1的两个不同状态，这两个状态之间不存在其他状态满足这个性
用CTL表示：
$EF(c_1 \wedge E[c_1\ U\ (\neg c_1 \wedge E[\neg c_2\ U\ c_1])])$
用NusMV表示：
`CTLSPEC EF ( pr1.st = c & E[(pr1.st = c) U (!(pr1.st = c) & E [!(pr2.st = c) U (pr1.st = c)])] )`

实验结果

```
-- specification AG (pr1.st = n -> EX pr1.st = t)  is true
-- specification AG (pr2.st = n -> EX pr2.st = t)  is true
-- specification EF (pr1.st = c & E [ pr1.st = c U (!(pr1.st = c) & E [ !(pr2.st = c) U pr1.st = c ] ) ] )  is true
-- specification EF (pr1.st = c & E [ pr1.st = c U (!(pr1.st = c) & E [ !(pr2.st = c) U pr1.st = c ] ) ] )  is true
-- specification  G !(pr1.st = c & pr2.st = c)  is true
-- specification  G (pr1.st = t ->  F pr1.st = c)  is false
```

- 1,2为non-blocking，满足
- 3,4为no strict sequencing，满足
- 5为safety，满足
- 6为Liveness，不满足
  综上，只有Liveness不满足