**姓名: 王立敏**

**学号: 2017E8018661153**

**Q1. 应用公平Kripke模型非空问题算法验证A是否是K的公平可达性质**

**A1.**

bool EmpChecking(K)  
{

G:=(S,R);  
scclist:=scctarjan(G);  
for (each e in scclist) if (fairscc(K,e)) w=w+e;  
return ReachbilityAnalysis(K,w)=false;

}

这是模型的空性检查算法。

由最后一句“return ReachbilityAnalysis(K,w)=false;”可知，若K非空，则w可达，即I中一定存在元素属于公平强连通分量。

bool FairReachability(K,A)  
{

w:=I;  
repeat until w={};  
s:=w.getElement();  
if (s in A) and (FairState(K,s)=true) return true;  
visited[s]:=true;  
for each (s’ in R(s)), if (visited[s’]=false) w.putElement(s’);  
w.removeElement(s);  
return false;

}

这是公平可达检测算法，与可达检测算法ReachbilityAnalysis()相比较多了一条“and (FairState(K,s)=true)”公平状态判定条件，即需要I中存在某一元素s，使其既存在于A中又是公平强联通分量。由上述分析可知，当K满足非空时，后者肯定满足，但是我们仍需检测它是否存在于A中，但是原先的空性检查算法并没有检查这个条件，因此并不能判断A是否是K的公平可达性质，我们改造空性检查算法，得到如下新的公平可达性检测函数

bool FairReachability(K,A)

{

G:=(S,R);  
scclist:=scctarjan(G);  
for (each e in scclist) if (fairscc(K,e)) and (e in A) w=w+e;  
return ReachbilityAnalysis(K,w)=false;

}

**Q2. 应用公平Kripke模型非空问题算法验证A是否是K的公平可免性质**

**A2.**

bool FairAvoidability(K,A)  
{

S’:=S-A; R’=R|S’; I’:=I-A; F’=F|S’;  
G:=(S’,R’); K’:=(S’,R’,I’); K’’:=(S’,R’,I’,F’);  
scclist:=scctarjan(G);  
w:={}; for (each e in scclist) if (fairscc(K’’,e)) w:=(w+e);  
return ReachbilityAnalysis(K’,w);

}

这是公平可免检测函数。通过分析可知，只要存在某一元素属于I’:=I-A并且是公平强联通的，即存在公平强连通分量属于I，但是不属于A，则A就是公平可免的。由空性检查算法可知，存在元素于I中且是公平强联通分量，但是原空性检查算法并没有检查它是否存在于A中，因此无法判定A是否是公平可免性质，通过修改即可满足要求

bool FairAvoidability(K,A)  
{

G:=(S,R);  
scclist:=scctarjan(G);  
for (each e in scclist) if (fairscc(K,e)) and (e not in A) w=w+e;  
return ReachbilityAnalysis(K,w)=false;

}