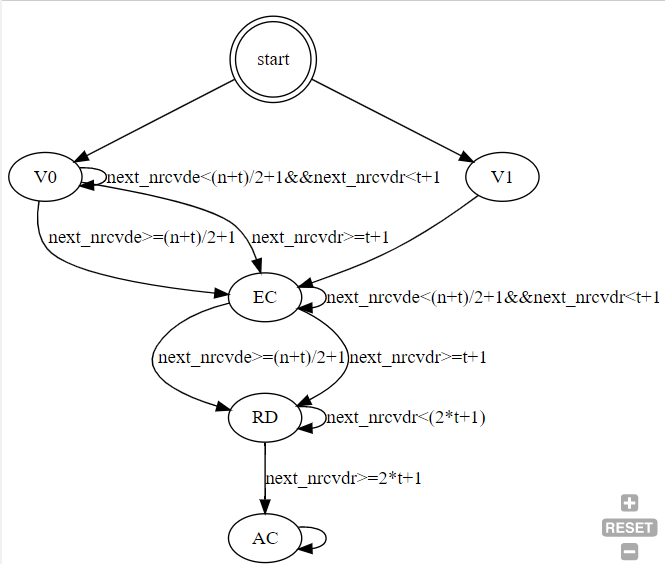
|  |  |  |  |
| --- | --- | --- | --- |
| 协议名称 | 错误类型 | 协议假设 | 协议性质 |
| aba-asyn-byzagreement0 | Byzantine | N > 3，T >= 1，N > 3 \* T, T >= F >= 0 | unforgeability, correctness, agreement |

1. 协议假设
2. n>3：节点个数大于3
3. f>=0：错误节点个数大于等于0
4. n>3\*f：总节点个数大于3倍错误节点个数
5. 如果发送器正确，则所有正确的进程都决定其值
6. 如果发送器是恶意的，则要么没有正确的进程来确定，要么它们都确定相同的值。
7. 工作流程

协议中有三种类型的消息：initial、echo和ready。协议从发送器发送initial消息开始。然后进程通过echo消息相互报告接收到的值。如果一个进程接收到超过（n+k）/2条echo消息，它将用ready消息宣布它。如果进程接收到2k+1条值一致的ready消息，它将确定该值。

1. 协议模式



1. 协议性质
2. Agreement：所有节点最终都会确定相同的值

ex\_acc：存在做出达到AC状态的节点

all\_acc：任意节点都达到AC状态

[](ex\_acc -> <>(all\_acc))：在任何时刻，如果存在达到AC状态的节点，那么最终任意节点都会达到AC状态

1. Correctness：

prec\_corr：所有节点初始状态都为V1

ex\_acc：存在达到AC状态的节点

prec\_corr -> <>(ex\_acc)：如果所有节点初始状态都为V1，那么最终会存在达到AC状态的节点

1. unforgeability：

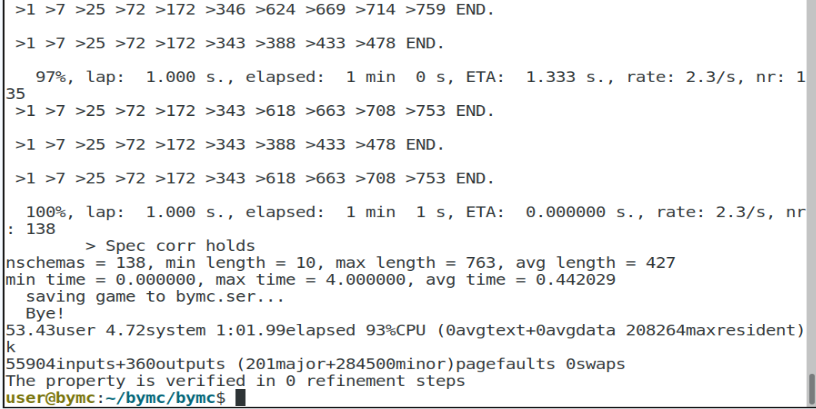
prec\_unforg：所有节点初始状态都为V0

(prec\_unforg -> []!ex\_acc)：如果所有节点初始状态都为V0，那么在任意时刻都不会有节点达到AC状态

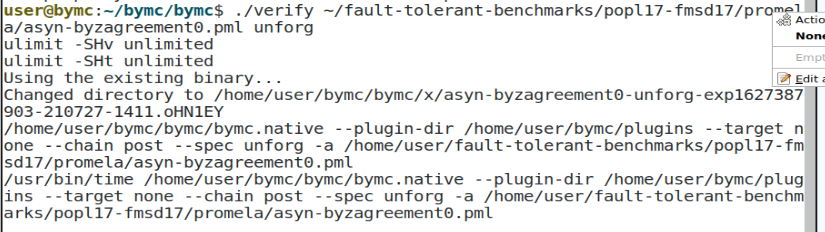
1. 验证结果

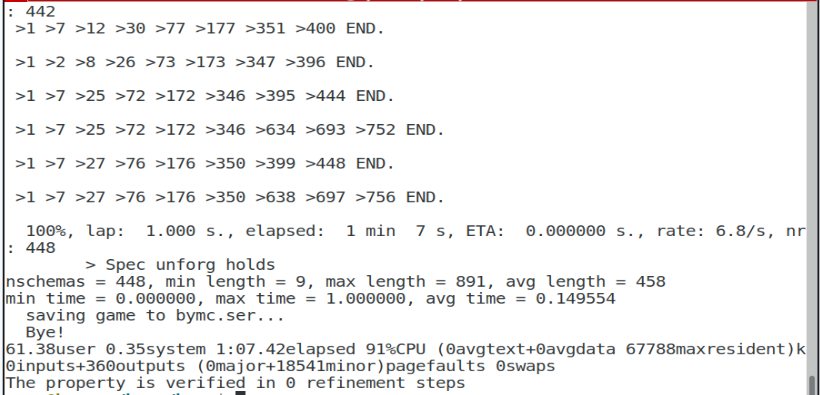
correctness：验证成功





unforgeability：验证成功





agreement：验证成功

