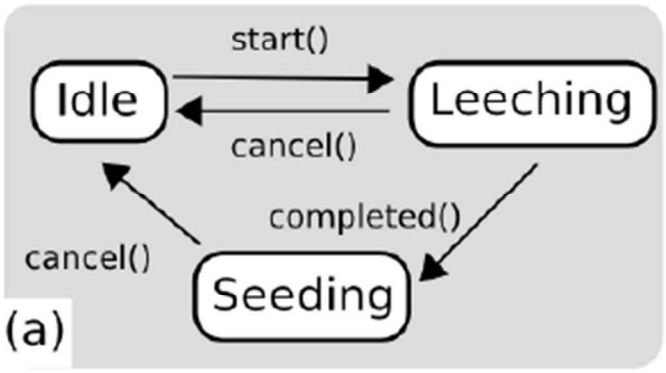
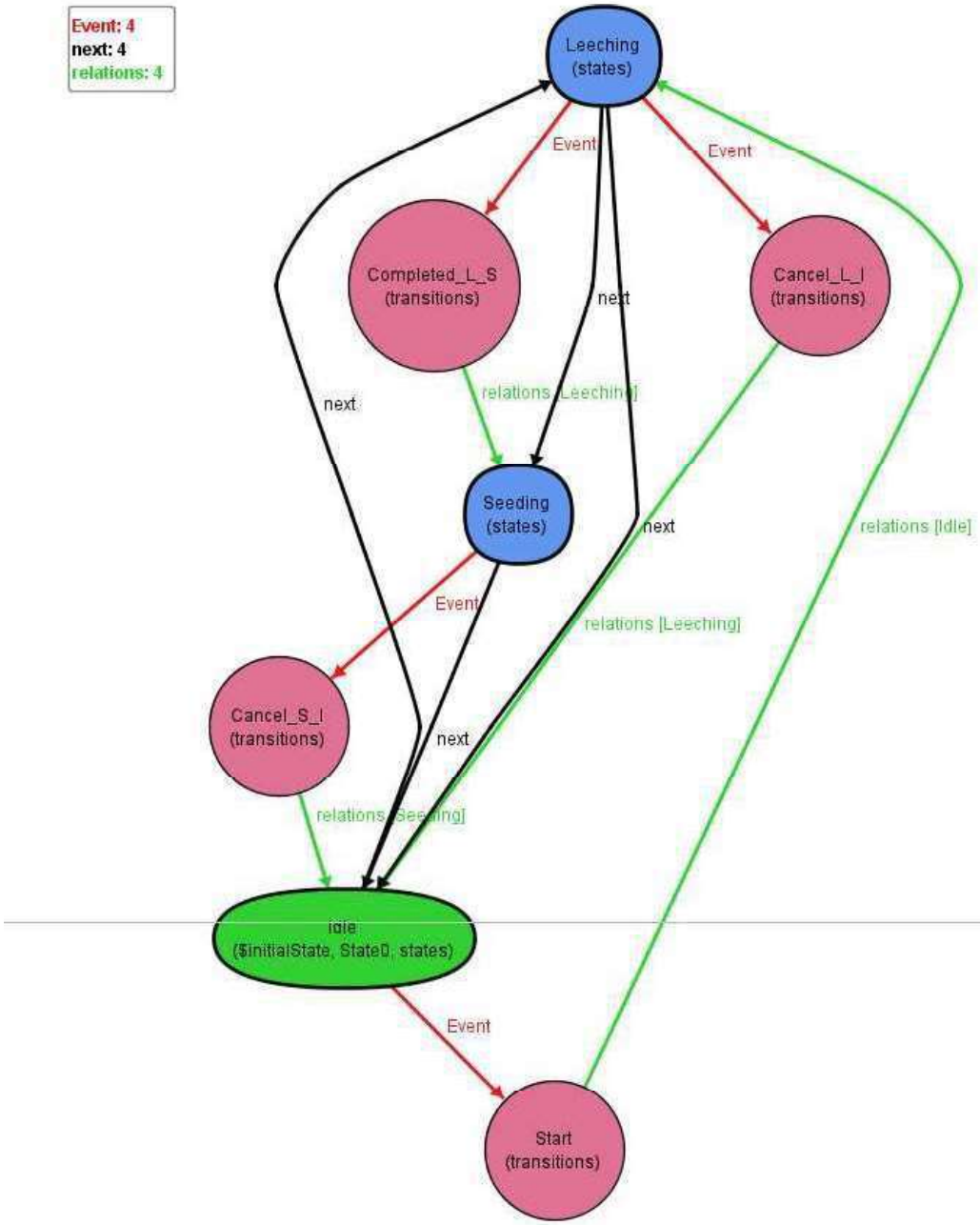
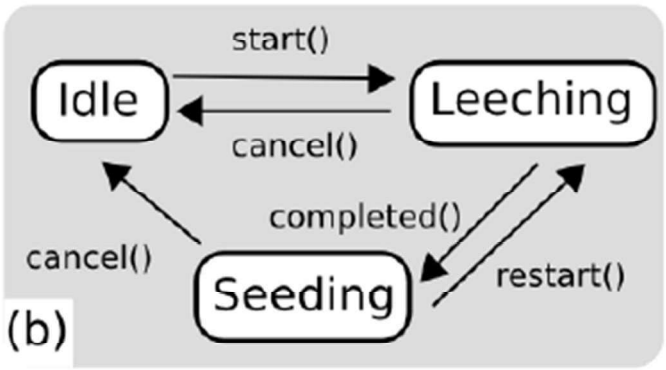
Executing showExamples will return the 6 possible scenarios of our peer-to-peer file sharing system:

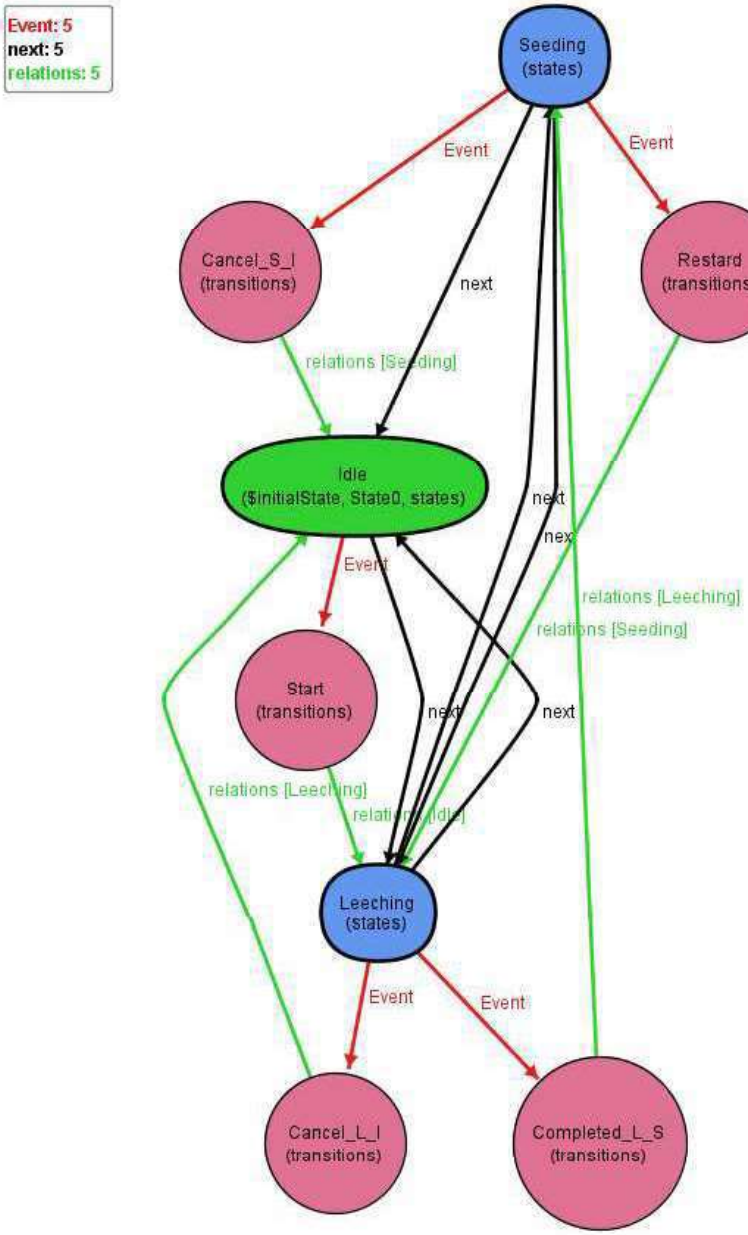
- *Scenario 1.1*



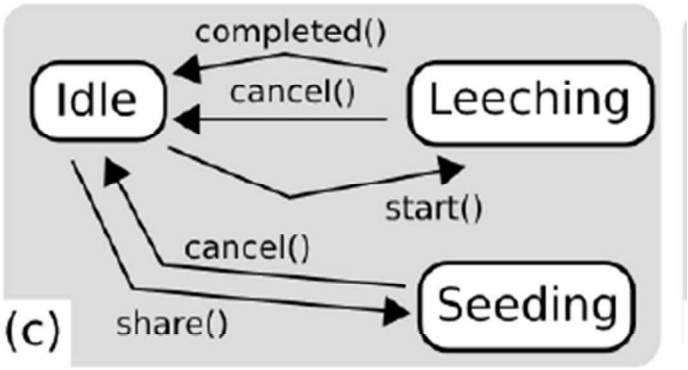


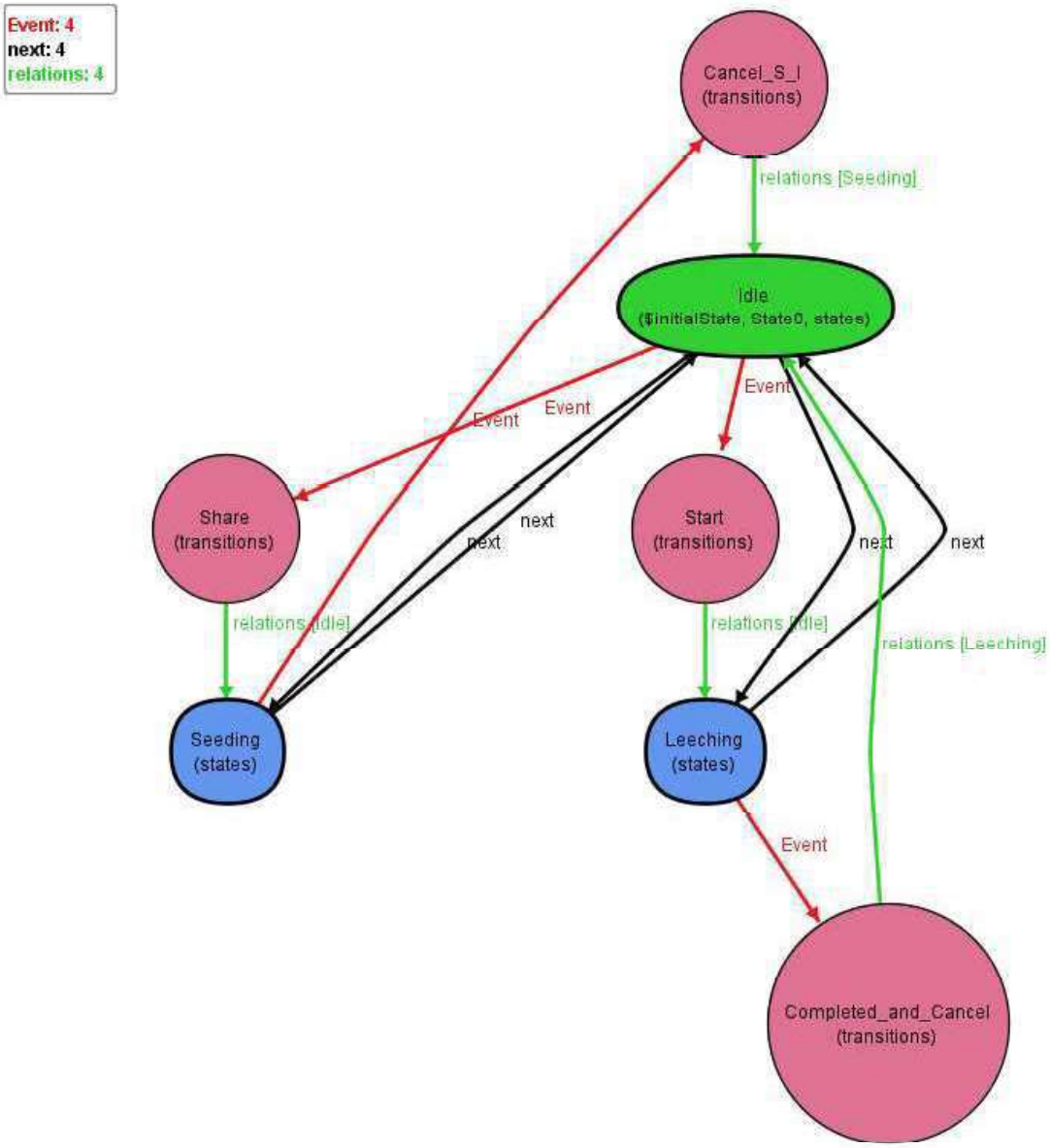
- *Scenario 12*


start()
Idle
Leeching
cancel()
completed()
cancel()
Seeding
restart()
(b)

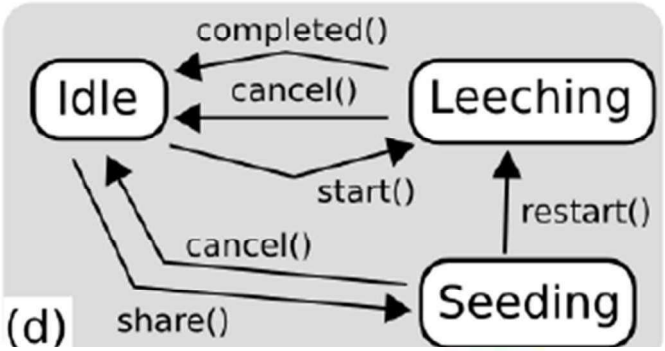

Event: 5
next: 5
relations: 5
Seeding (states)
Event
Event
Cancel_S_I (transitions)
next
Restard (transitions)
relations [Seeding]
Idle ($initialState, State0, states)
next
next
Event
relations [Leeching]
relations [Seeding]
Start (transitions)
next
next
relations [Leeching]
relations [Idle]
Leeching (states)
Event
Event
Cancel_L_I (transitions)
Completed_L_S (transitions)

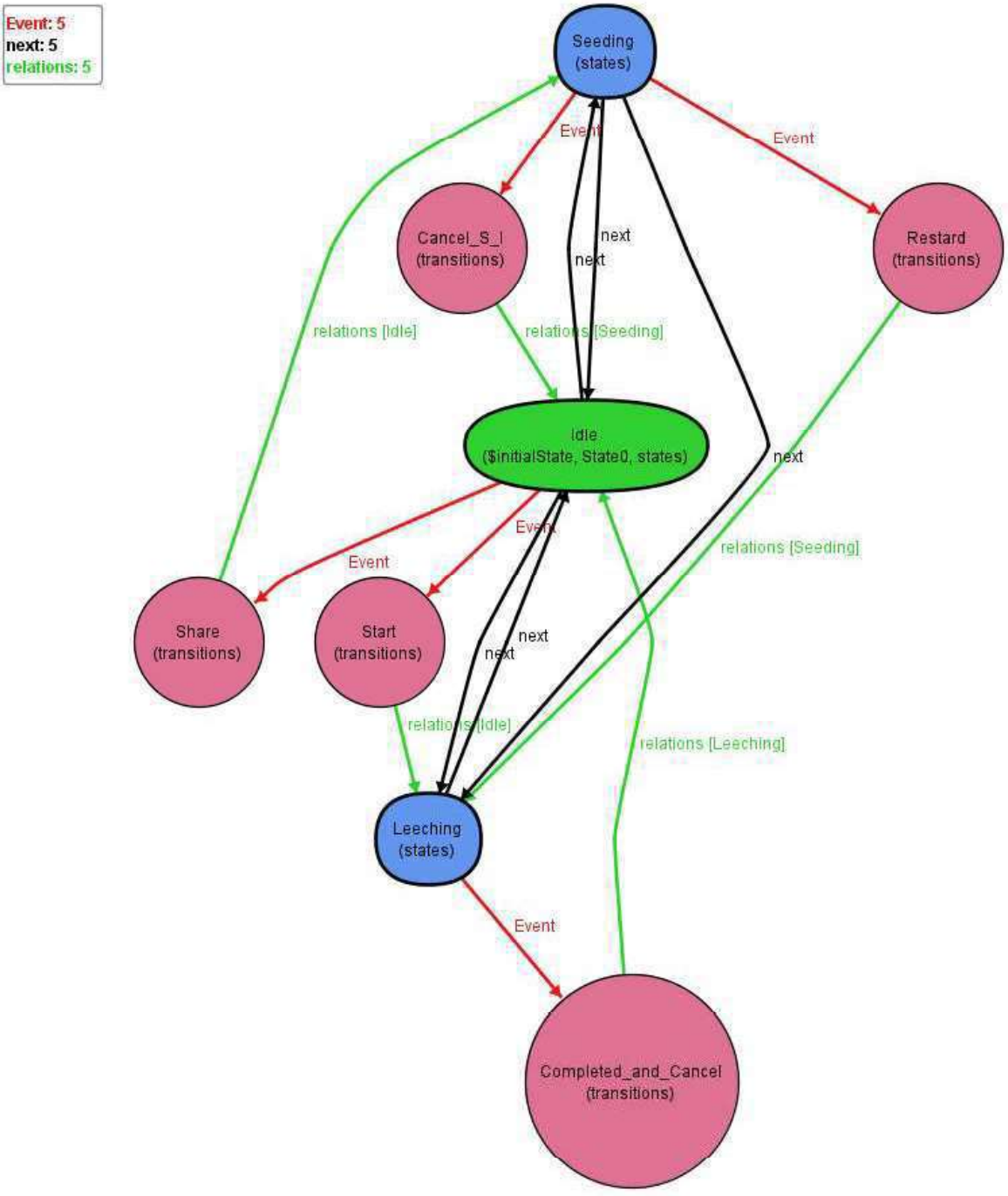
- *Scenario 21*


completed()
Idle
cancel()
Leeching
start()
cancel()
Seeding
(c)
share()

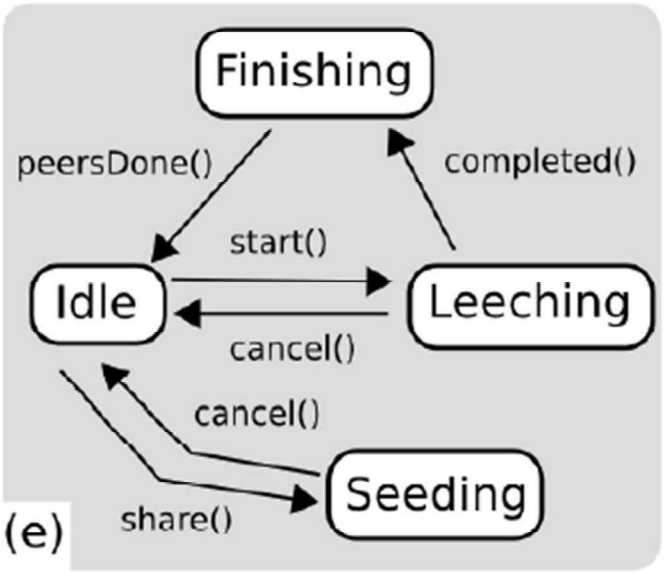

Event: 4
next: 4
relations: 4
Cancel_S_I
(transitions)
relations [Seeding]
Idle
($initialState, State0, states)
Event
Event
Event
Share
(transitions)
next
next
Start
(transitions)
next
next
relations [Idle]
relations [Idle]
relations [Leeching]
Seeding
(states)
Leeching
(states)
Event
Completed_and_Cancel
(transitions)

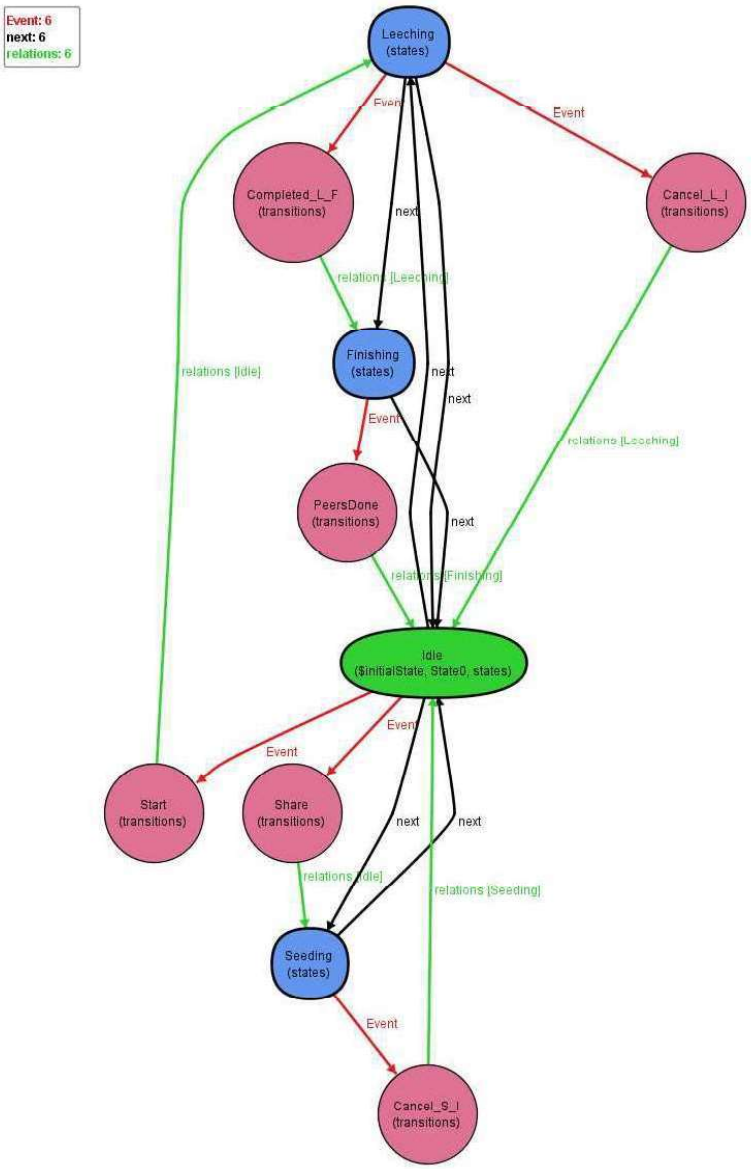
- *Scenario 22*


completed()
cancel()
Idle
Leeching
start()
restart()
cancel()
Seeding
(d)
share()

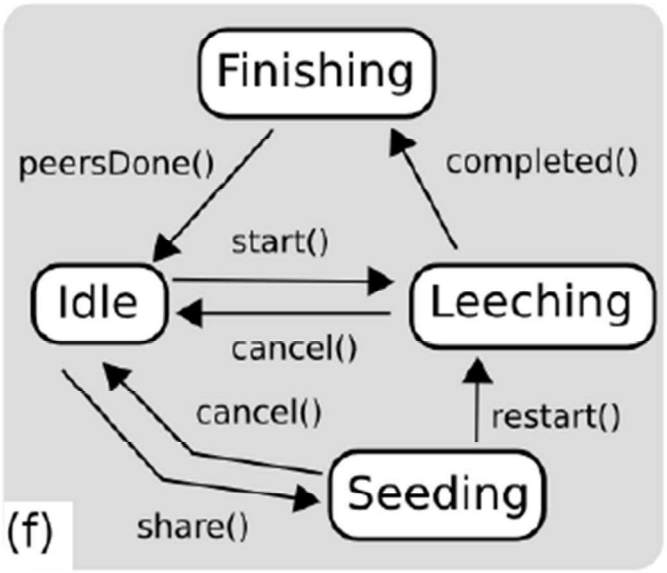

Event: 5
next: 5
relations: 5
Seeding (states)
Event
Event
Cancel_S_I (transitions)
next
next
Restard (transitions)
relations [Idle]
relations [Seeding]
Idle ($initialState, State0, states)
next
Event
Event
relations [Seeding]
Share (transitions)
Start (transitions)
next
next
relations [Idle]
relations [Leeching]
Leeching (states)
Event
Completed_and_Cancel (transitions)

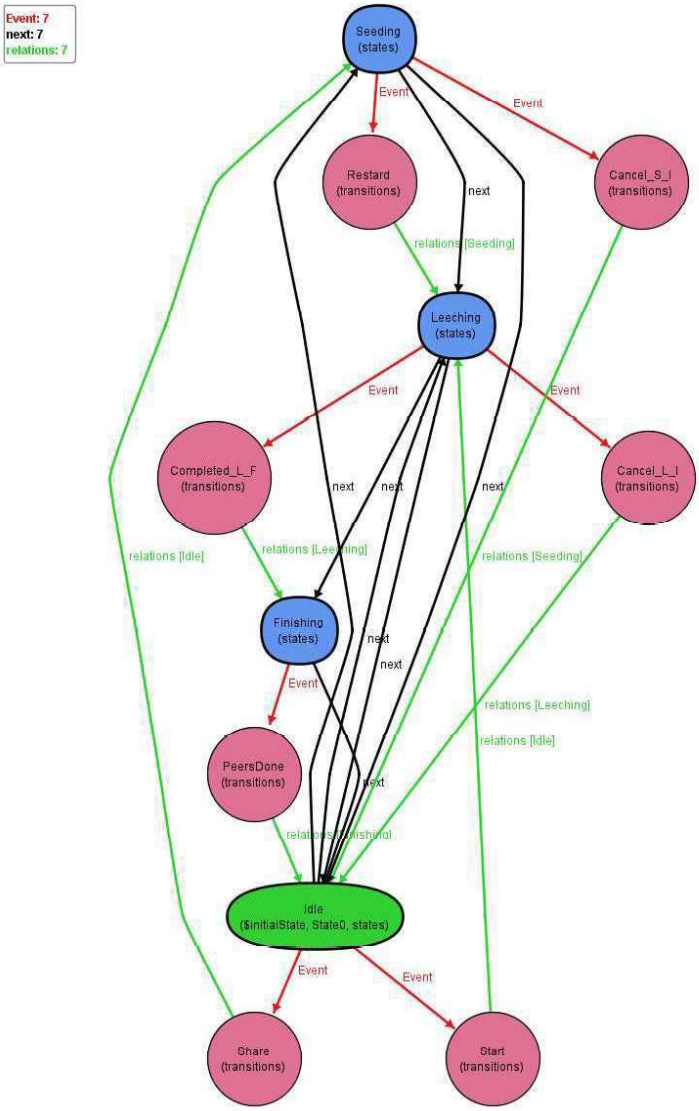
- *Scenario 31*


Finishing
peersDone()
completed()
start()
Idle
Leeching
cancel()
cancel()
Seeding
share()
(e)


Event: 6
next: 6
relations: 6
Leeching (states)
Event
Event
Completed_L_F (transitions)
next
Cancel_L_I (transitions)
relations [Leeching]
Finishing (states)
next
next
relations [Idle]
Event
relations [Leeching]
PeersDone (transitions)
next
relations [Finishing]
Idle ($initialState, State0, states)
Event
Event
Start (transitions)
Share (transitions)
next
next
relations [Idle]
relations [Seeding]
Seeding (states)
Event
Cancel_S_I (transitions)

- *Scenario 33*

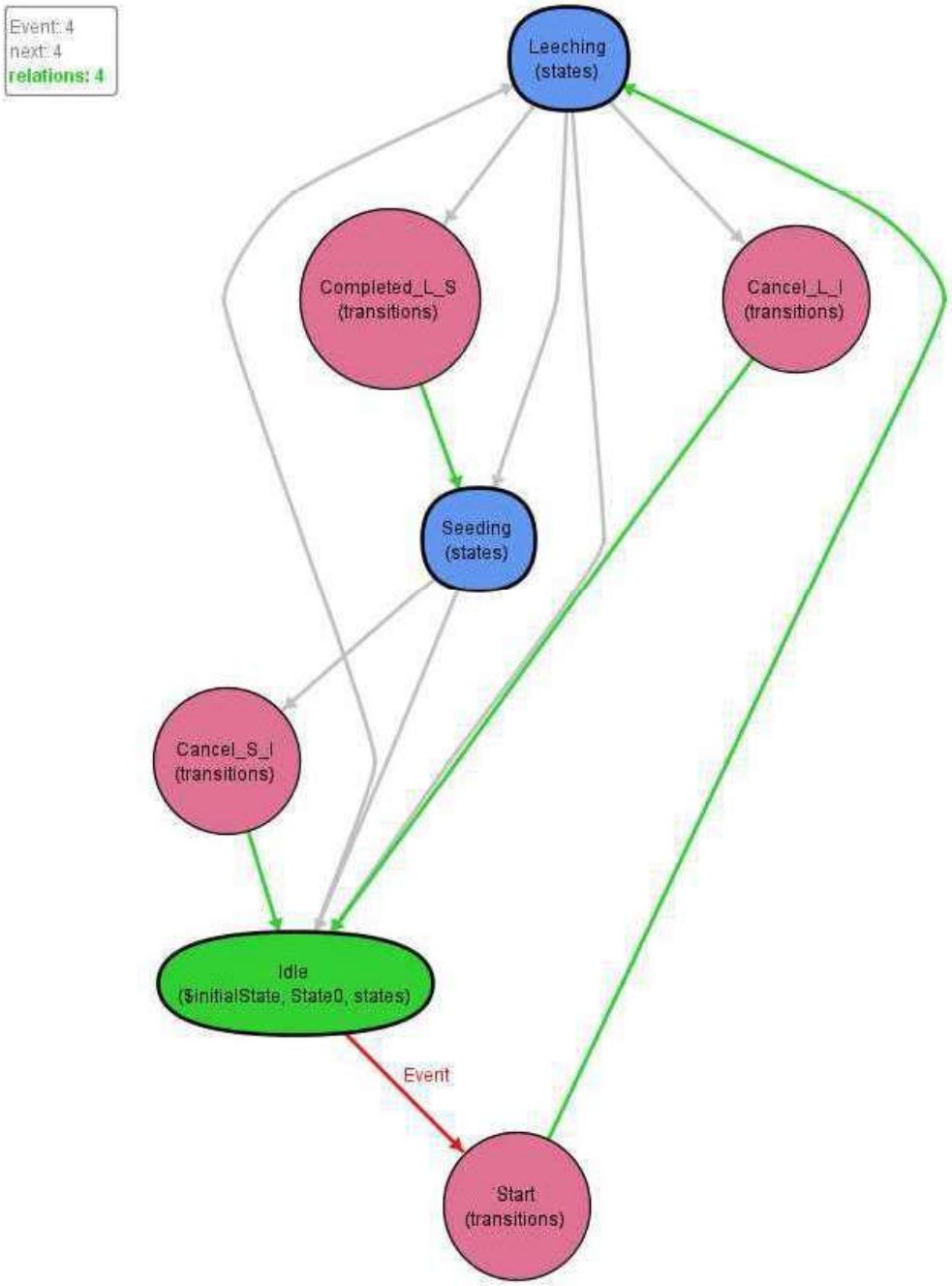

Finishing
peersDone()
completed()
start()
Idle
Leeching
cancel()
cancel()
restart()
Seeding
share()
(f)

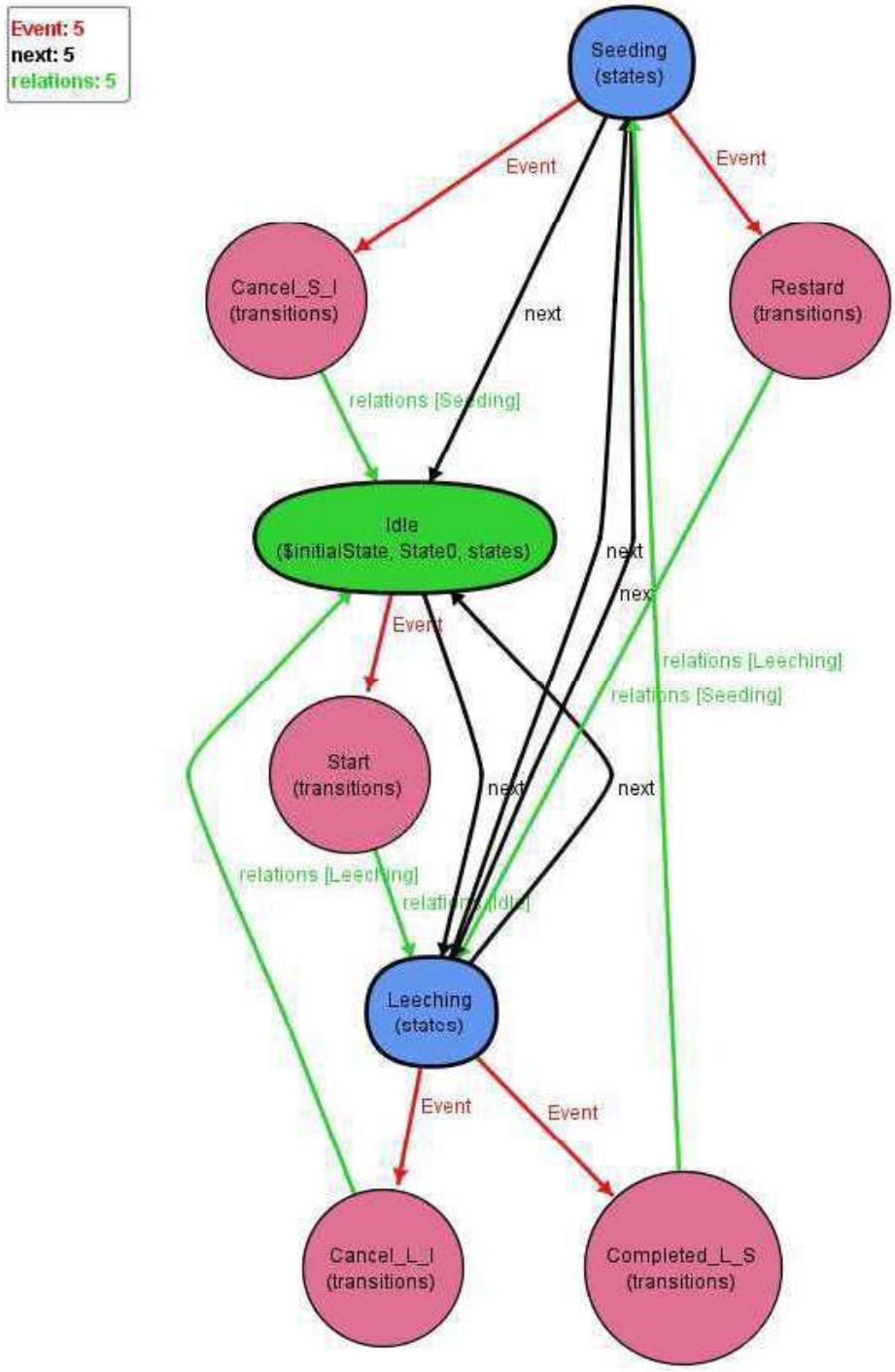

Event: 7
next: 7
relations: 7
Seeding (states)
Event
Event
Restard (transitions)
next
Cancel_S_I (transitions)
relations [Seeding]
Leeching (states)
Event
Event
Completed_L_F (transitions)
next
next
next
Cancel_L_I (transitions)
relations [Idle]
relations [Leeching]
relations [Seeding]
Finishing (states)
next
next
Event
relations [Leeching]
relations [Idle]
PeersDone (transitions)
next
Idle ($initialState, State0, states)
Event
Event
Share (transitions)
Start (transitions)

As an example, if we want to check the constraint SC1 with CTL, we can just write our specification in the section 'assert CTL_Formula':

```
// ****************************************** CTL checking ************************
assert CTL_Formula{ // here you can write your CTL spec
   ctl_mc[ ag[{
      s:TS.states | s in Idle implies Seeding in s.(TS.next) // this is the SC1
   }] ]
}
```

Our program will return all counter examples when Seeding is not directly reachable from Idle (2 cases):





We can construct the partial models' table of truth, we take the HC1 as example. In the first execution, we execute the HC1 predicate with #Completed_and_Cancel = 1, and we will find some instances in the output, which means: $\boldsymbol{\Phi_M \wedge \Phi_P = SAT}$, and then we run HC1 with #Completed_and_Cancel = 0 (negation), we will find some instances in the output, which means: $\boldsymbol{\Phi_M \wedge not\ \Phi_P = SAT}$, so we have the truth value ***Maybe,*** we do the same thing to fill the truth table with the other cases.