Analysis of State charts
+
Sequence Diagrams

through mapping on
Place/Transition Petri Nets

Wisam Al Abed

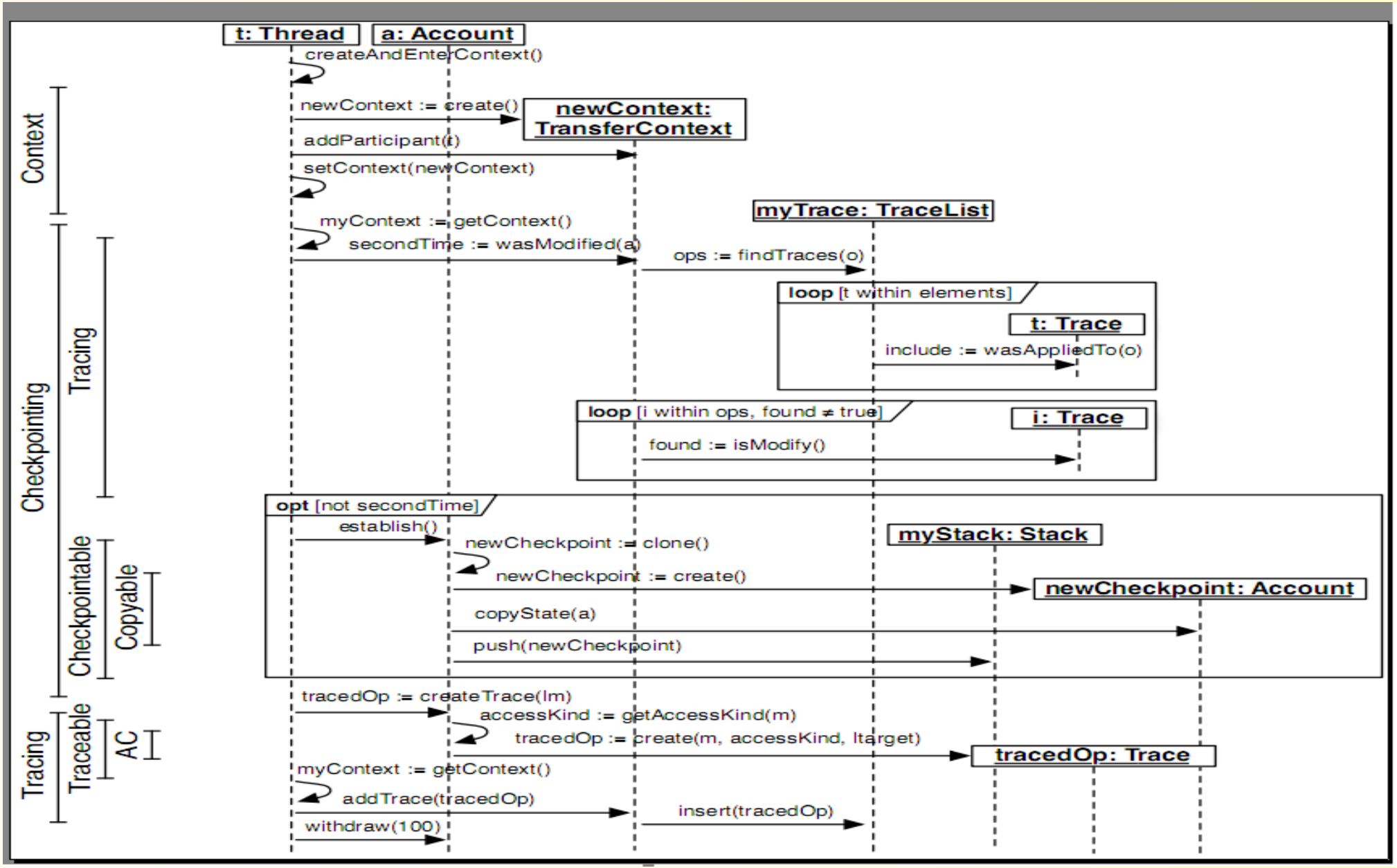
Outline

- Motivation
- What I have done so far
- Conclusion/Future Work
- References

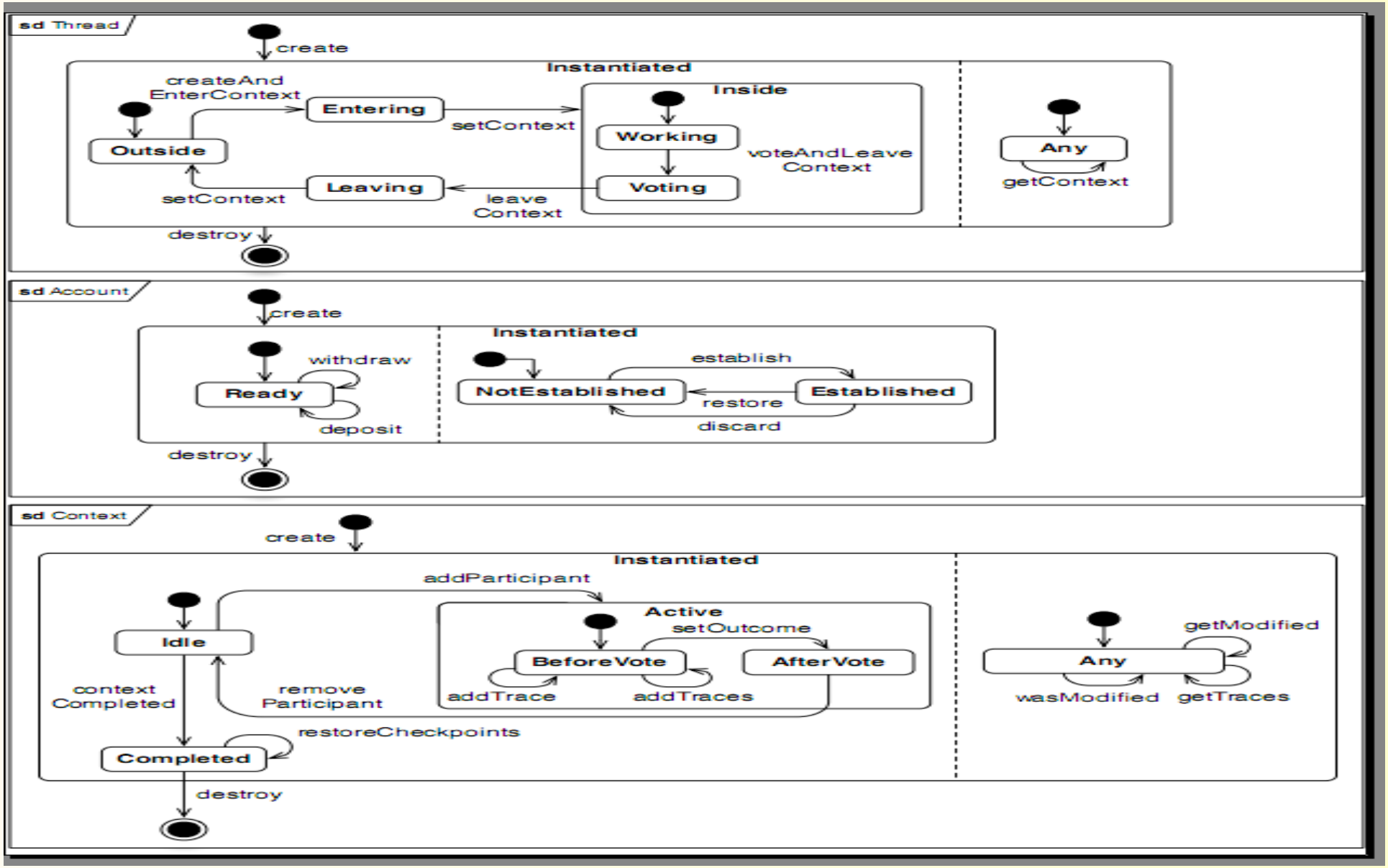
Motivation

- In RAM we model behavior of our aspects using UML sequence and state diagrams. Both these formalisms, while different, will essentially allow us to model the same behavior but depict it from two different perspectives.
- The challenge is to ensure that both these views do indeed correspond to the same behavior.

Motivation

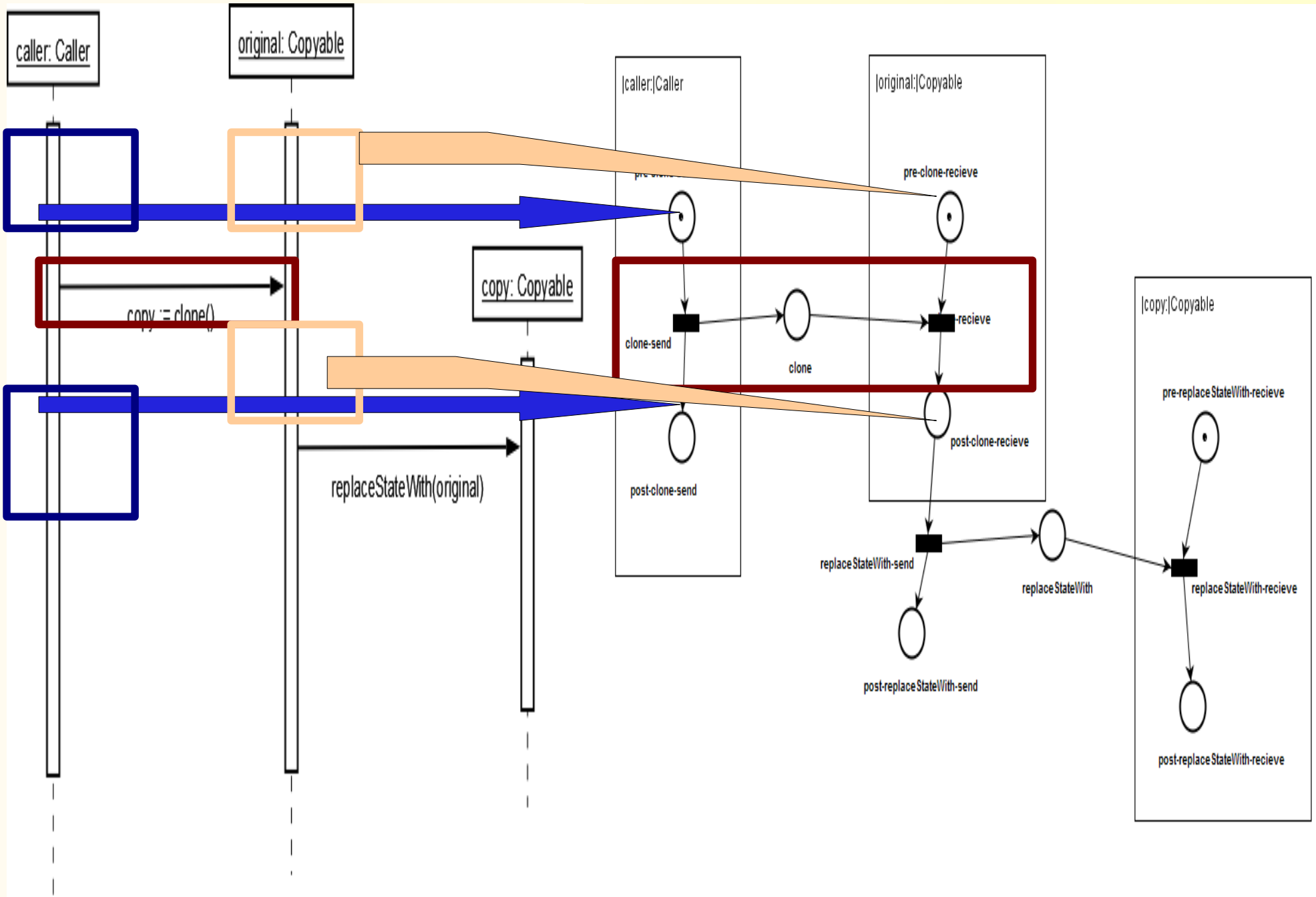

t: Thread
a: Account
createAndEnterContext()
newContext := create()
newContext: TransferContext
addParticipant(t)
setContext(newContext)
myContext := getContext()
secondTime := wasModified(a)
myTrace: TraceList
ops := findTraces(o)
loop [t within elements]
t: Trace
include := wasAppliedTo(o)
loop [i within ops, found ≠ true]
i: Trace
found := isModify()
opt [not secondTime]
establish()
myStack: Stack
newCheckpoint := clone()
newCheckpoint := create()
newCheckpoint: Account
copyState(a)
push(newCheckpoint)
tracedOp := createTrace(lm)
accessKind := getAccessKind(m)
tracedOp := create(m, accessKind, ltarget)
tracedOp: Trace
myContext := getContext()
addTrace(tracedOp)
insert(tracedOp)
withdraw(100)
Context
Checkpointing
Tracing
Checkpointable
Copyable
Tracing
Traceable
AC

Motivation


sd Thread
create
Instantiated
createAnd EnterContext
Entering
setContext
Inside
Working
voteAndLeave Context
Voting
Outside
Leaving
setContext
leave Context
destroy
Any
getContext
sd Account
create
Instantiated
withdraw
Ready
deposit
destroy
establish
NotEstablished
restore
Established
discard
sd Context
create
Instantiated
addParticipant
Idle
context Completed
remove Participant
Active
setOutcome
BeforeVote
AfterVote
addTrace
addTraces
restoreCheckpoints
Completed
destroy
Any
getModified
wasModified
getTraces

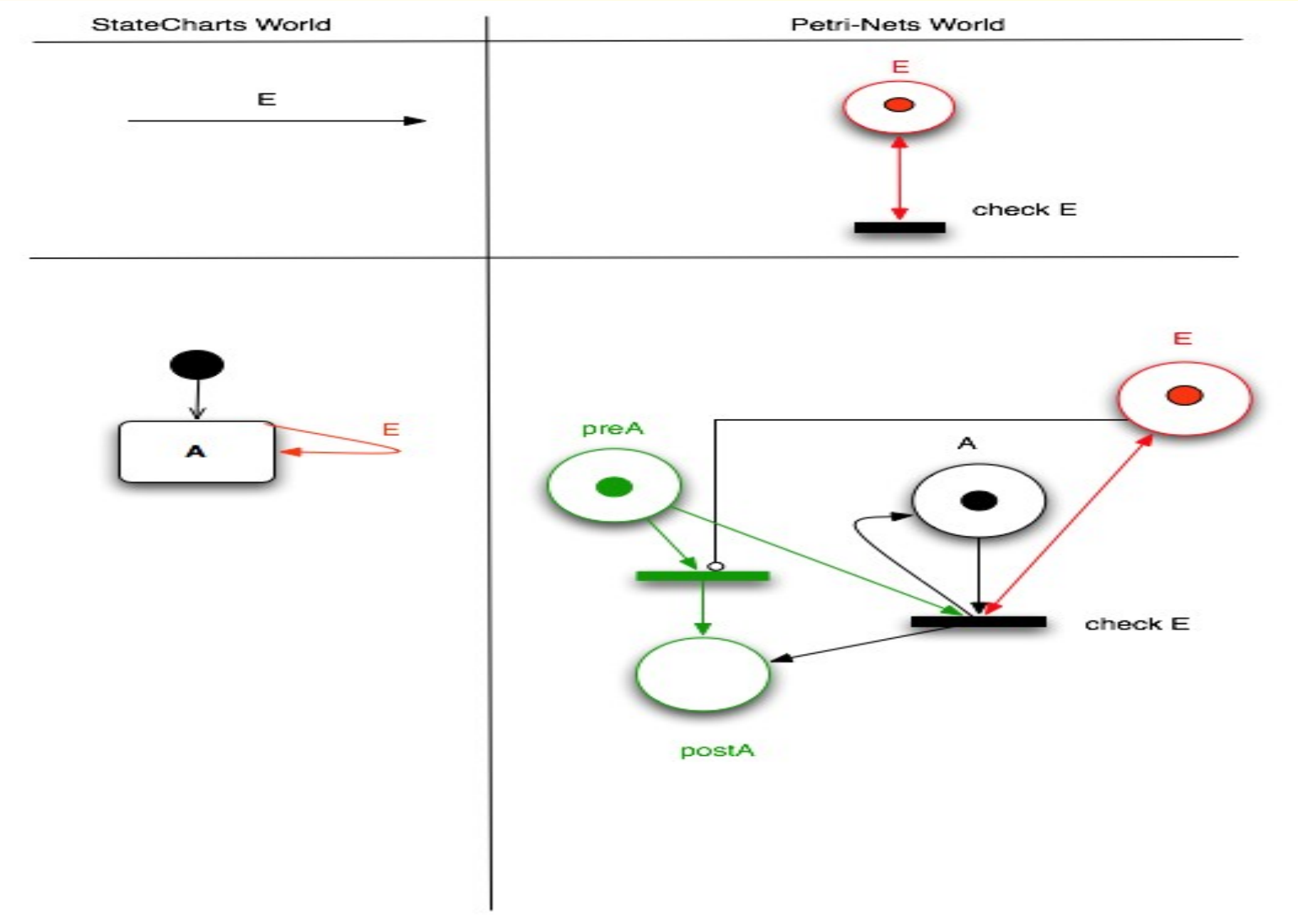
Chapter 1

Sequence Diagram To Petri-net

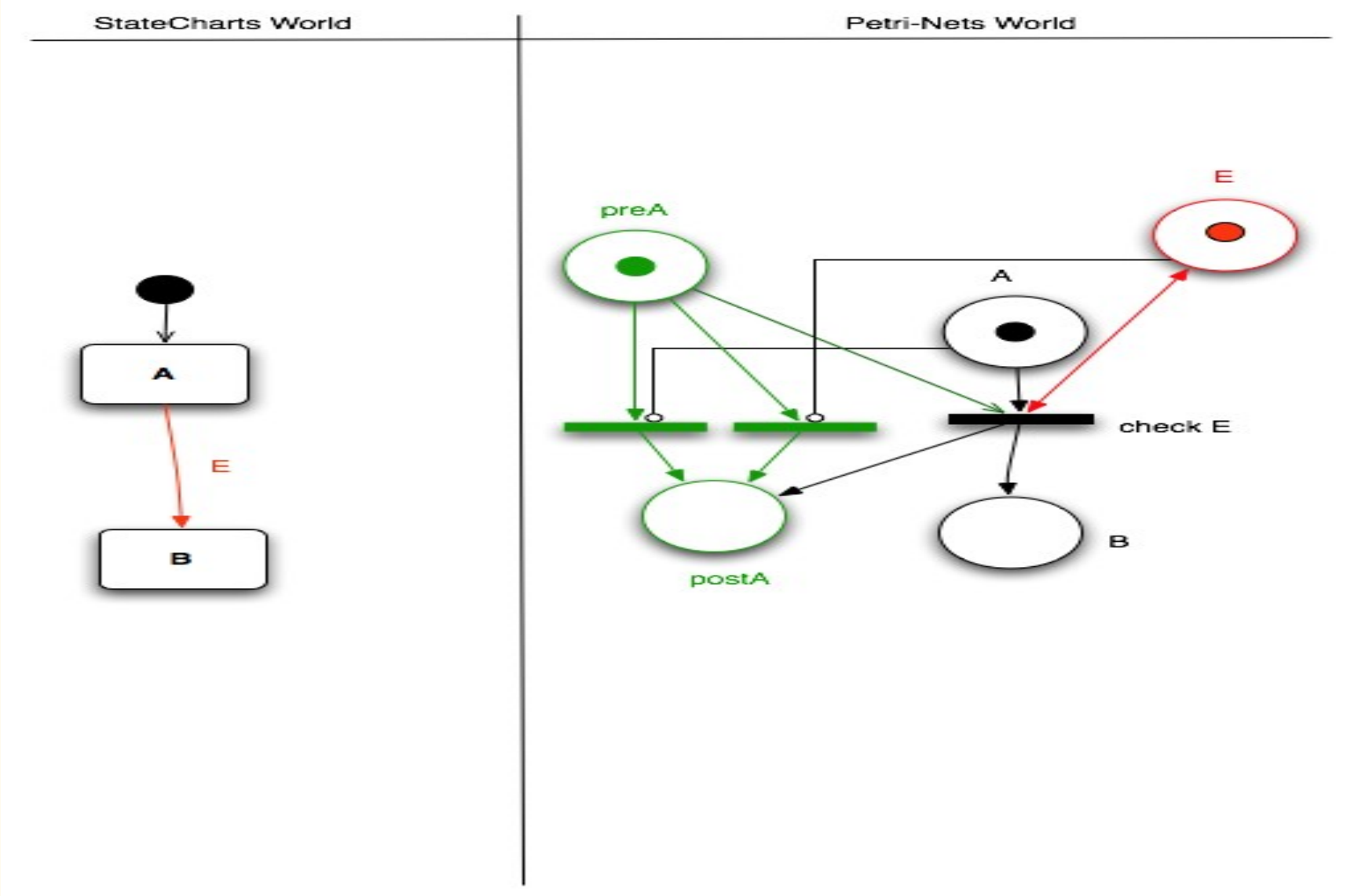


Chapter 2

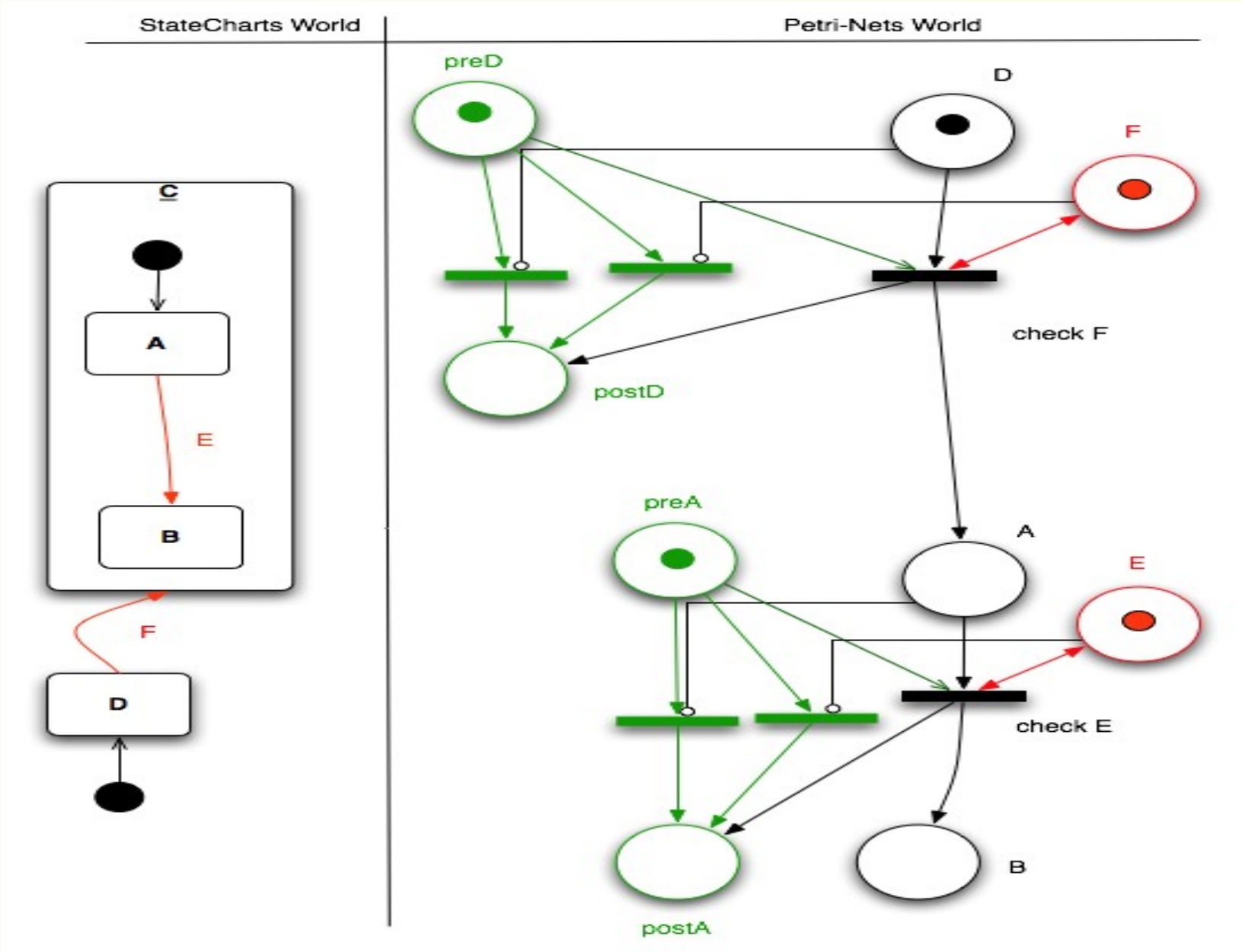
State Charts To Petri Nets



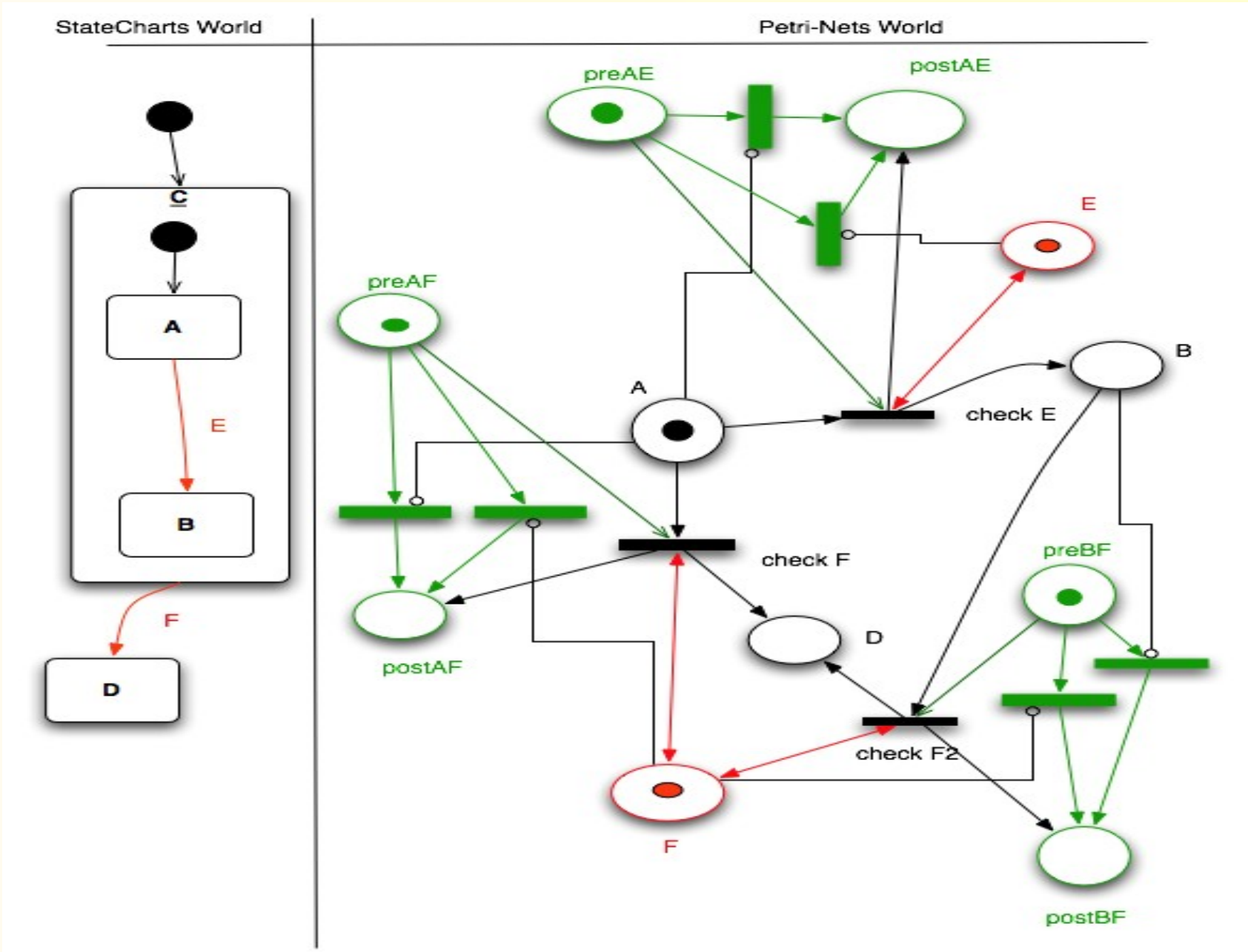
State Charts To Petri Nets



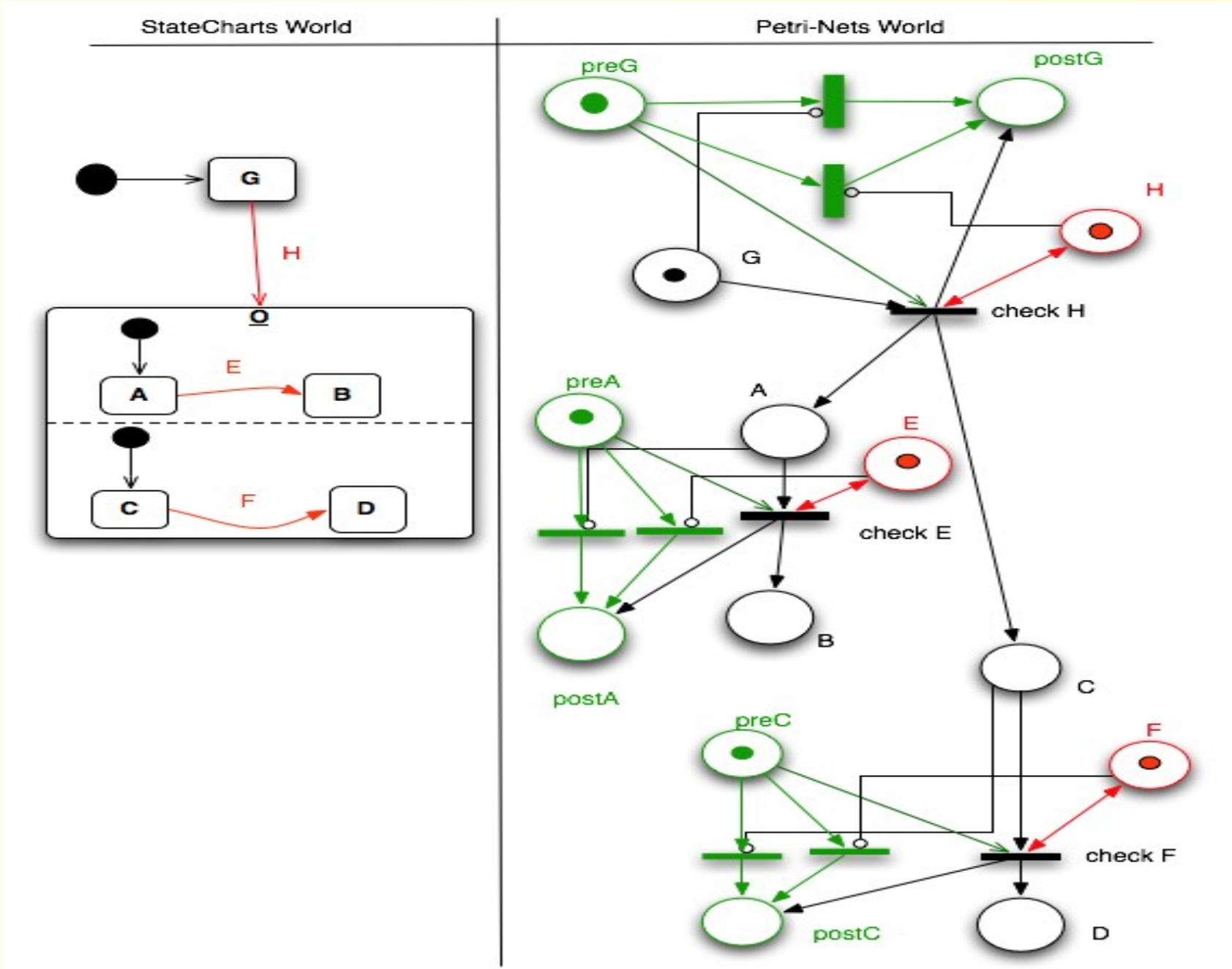
State Charts To Petri Nets



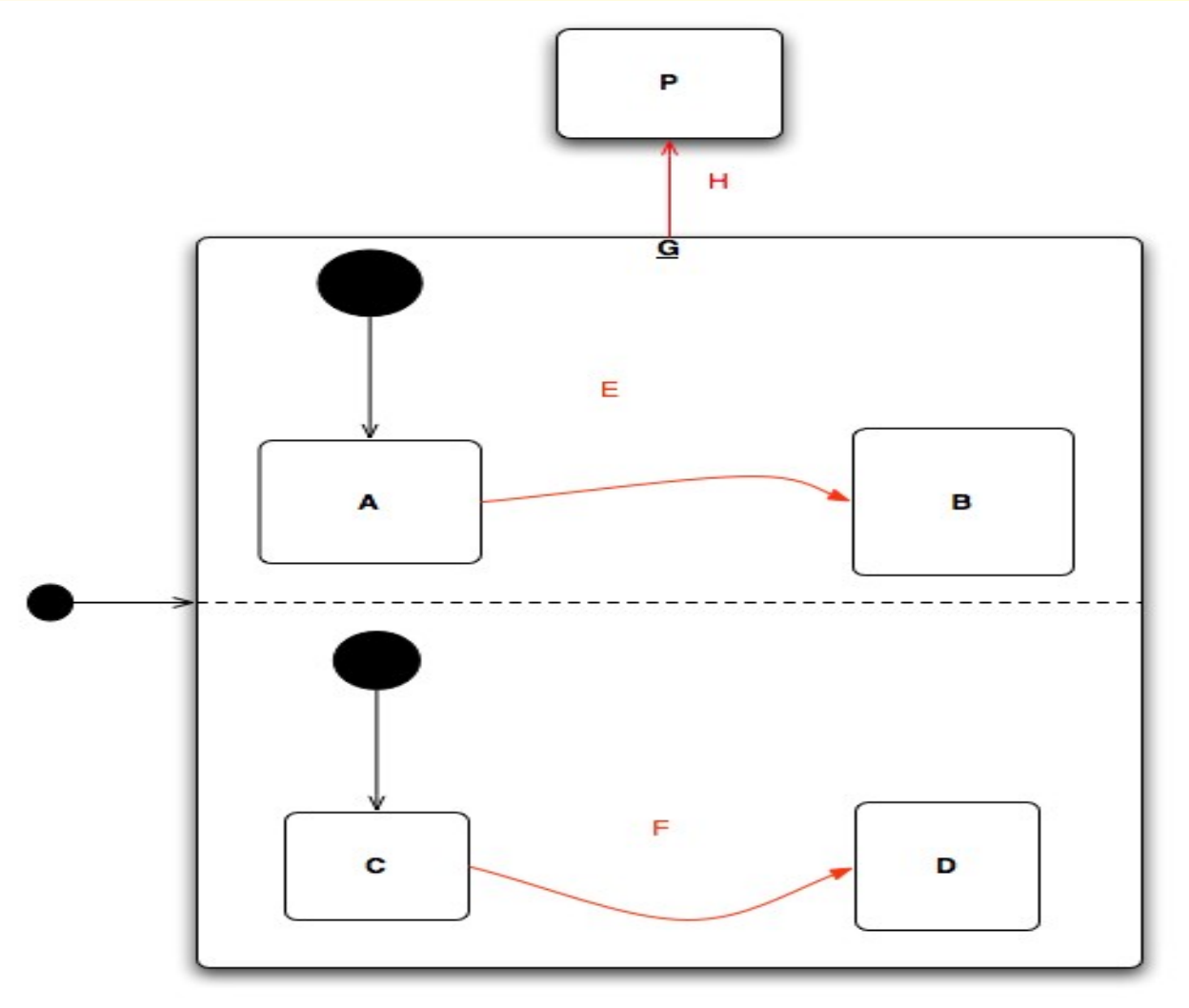
State Charts To Petri Nets



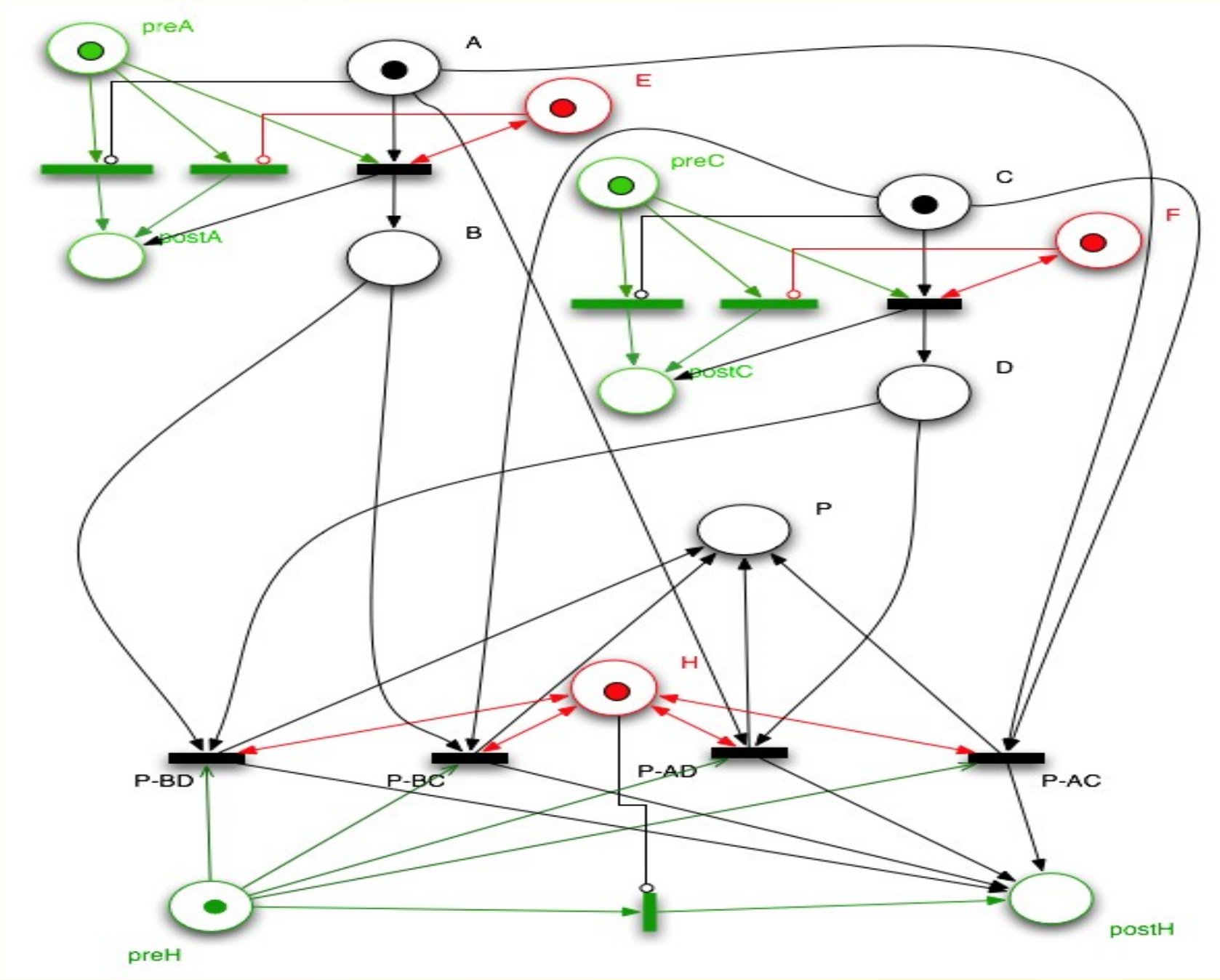
State Charts To Petri Nets



State Charts To Petri Nets

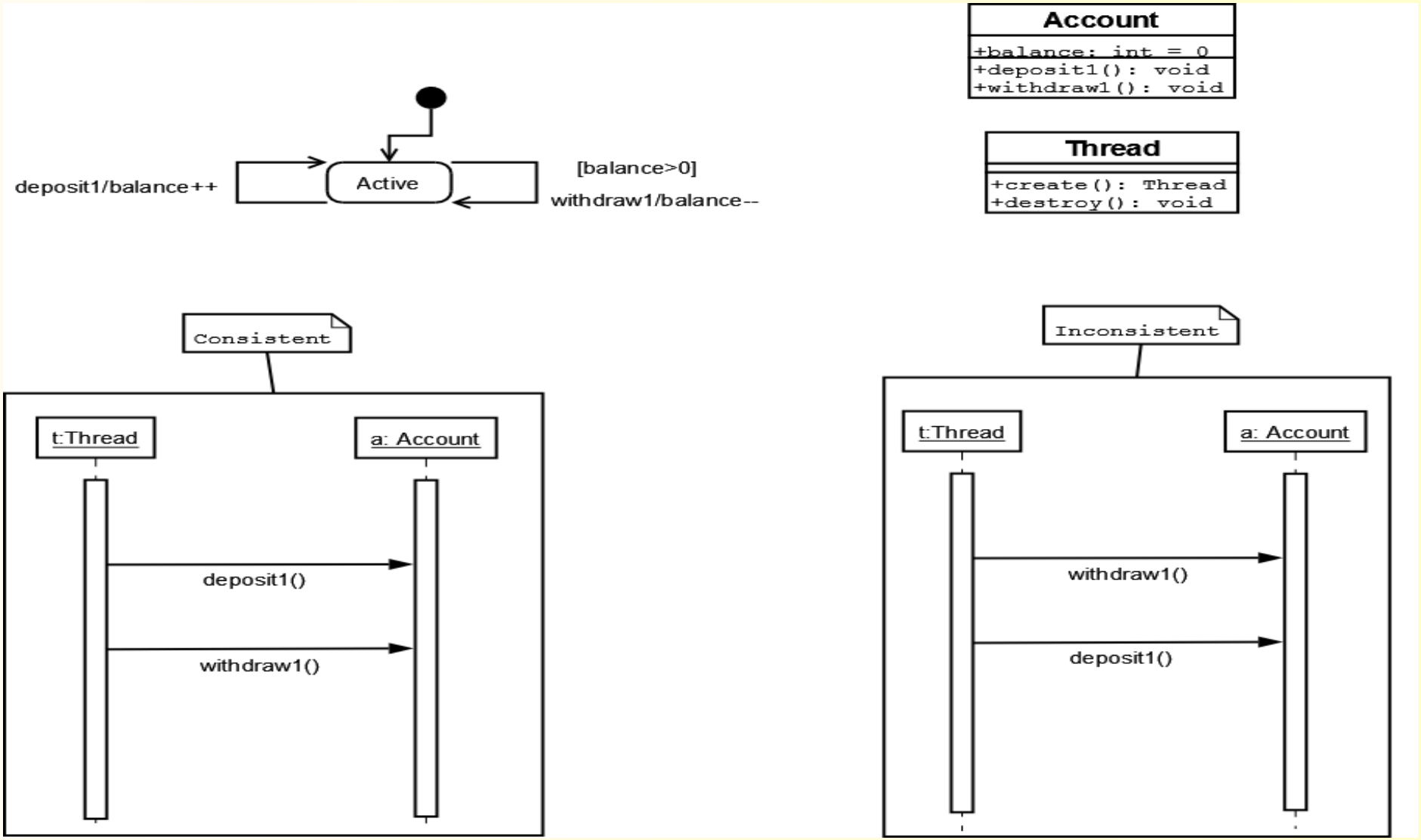


State Charts To Petri Nets

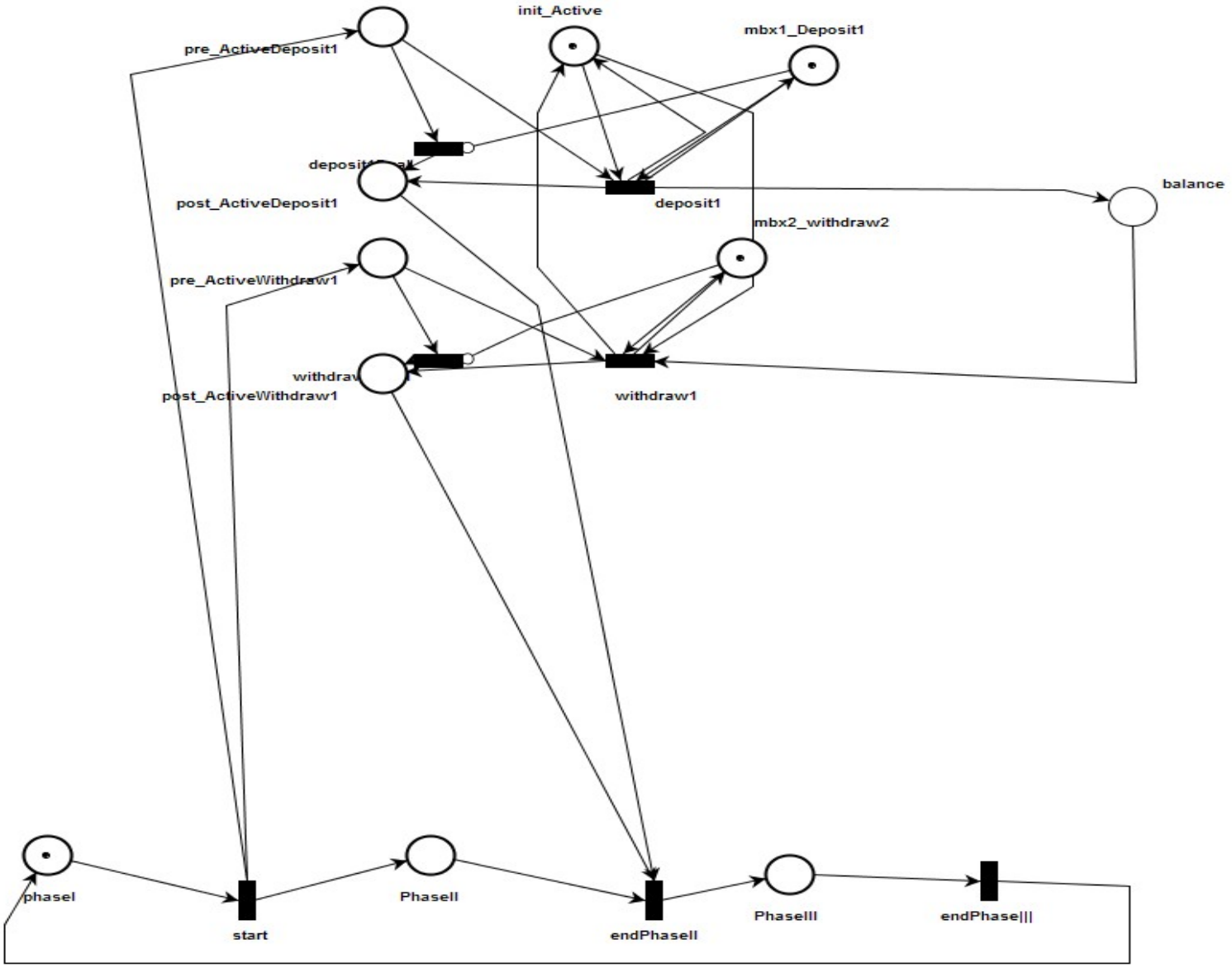


Chapter 3

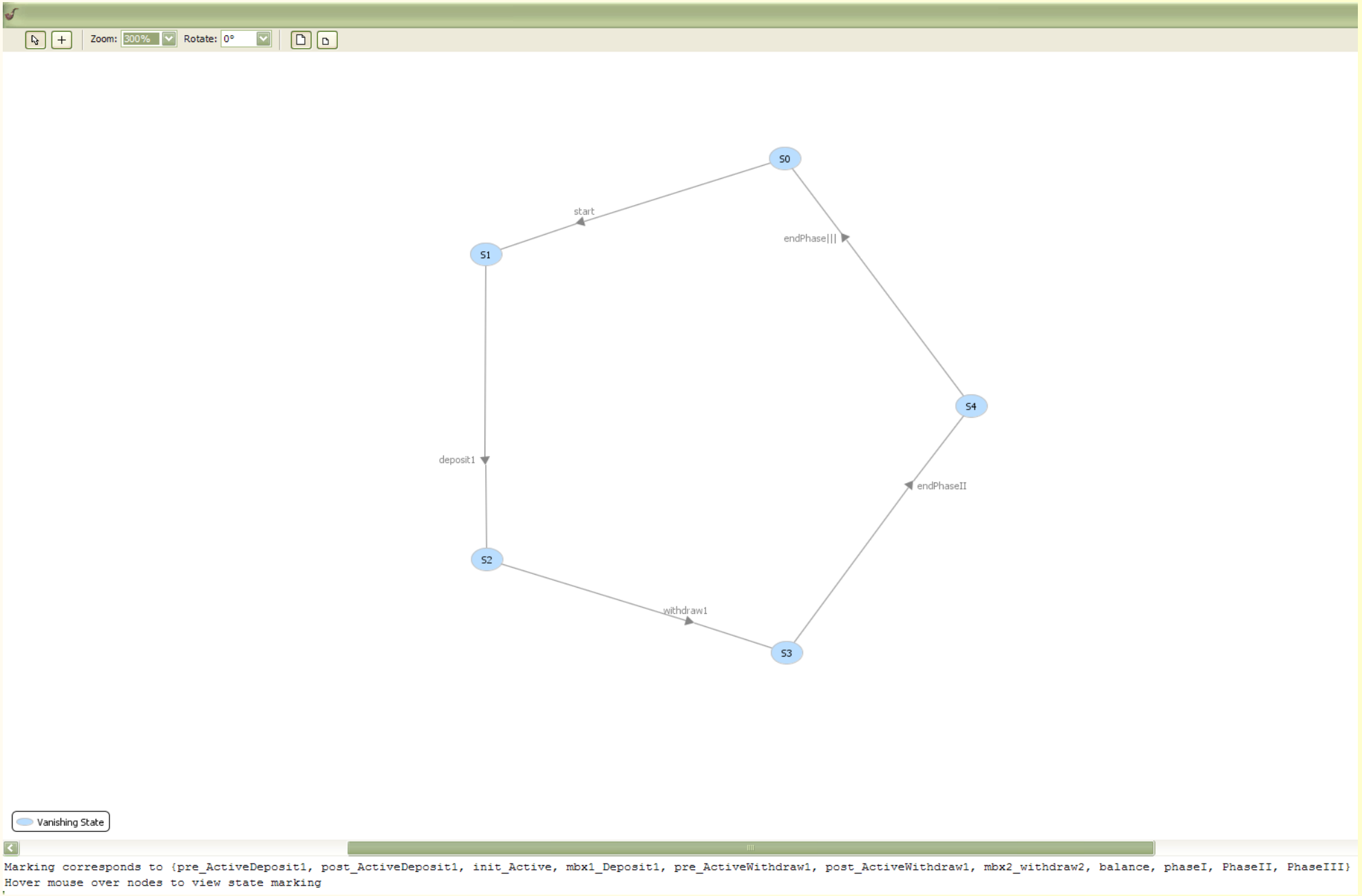
The Example


Account
+balance: int = 0
+deposit1(): void
+withdraw1(): void
Thread
+create(): Thread
+destroy(): void
deposit1/balance++
Active
[balance>0]
withdraw1/balance--
Consistent
t:Thread
a: Account
deposit1()
withdraw1()
Inconsistent
t:Thread
a: Account
withdraw1()
deposit1()

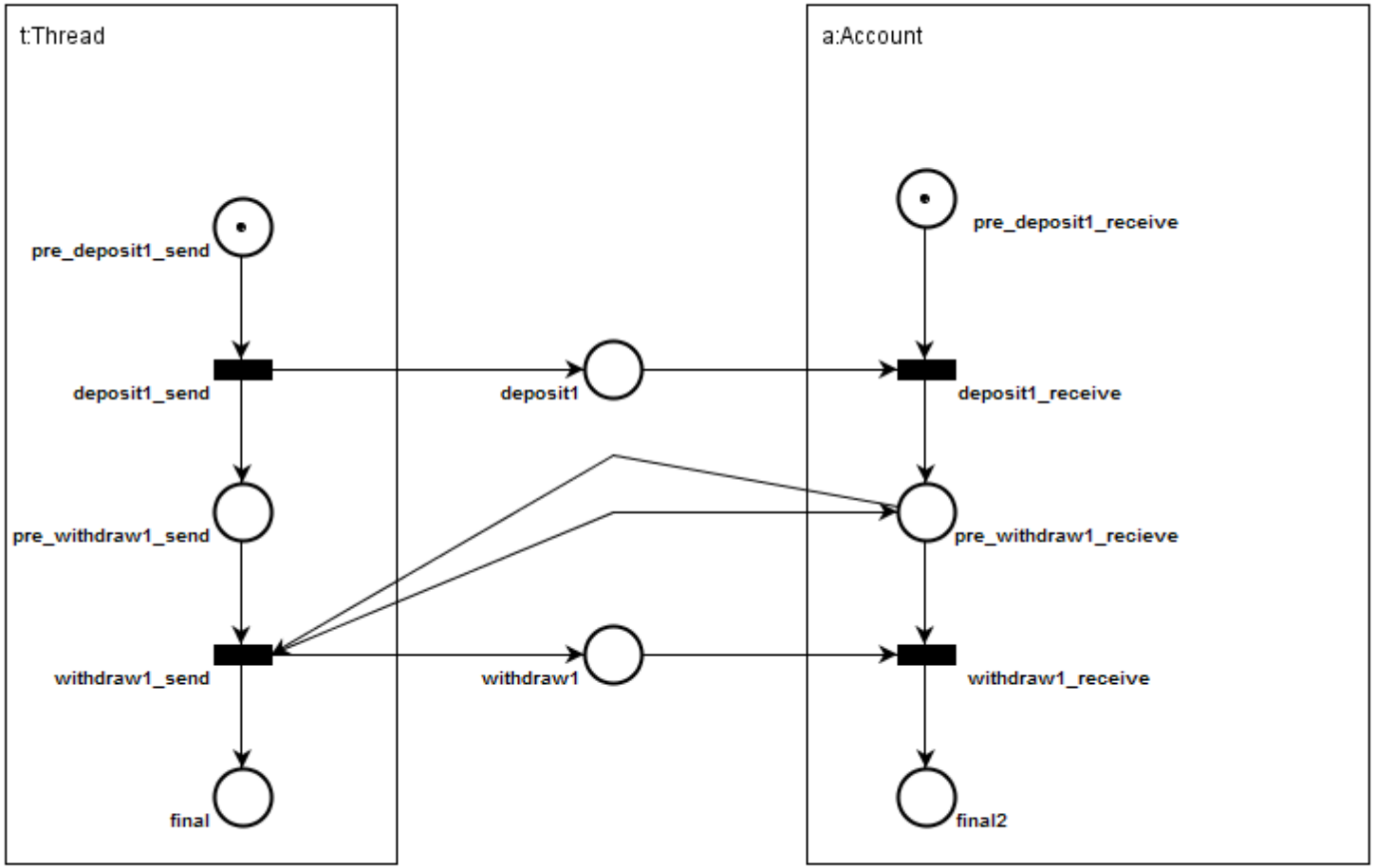
Transformed State Chart



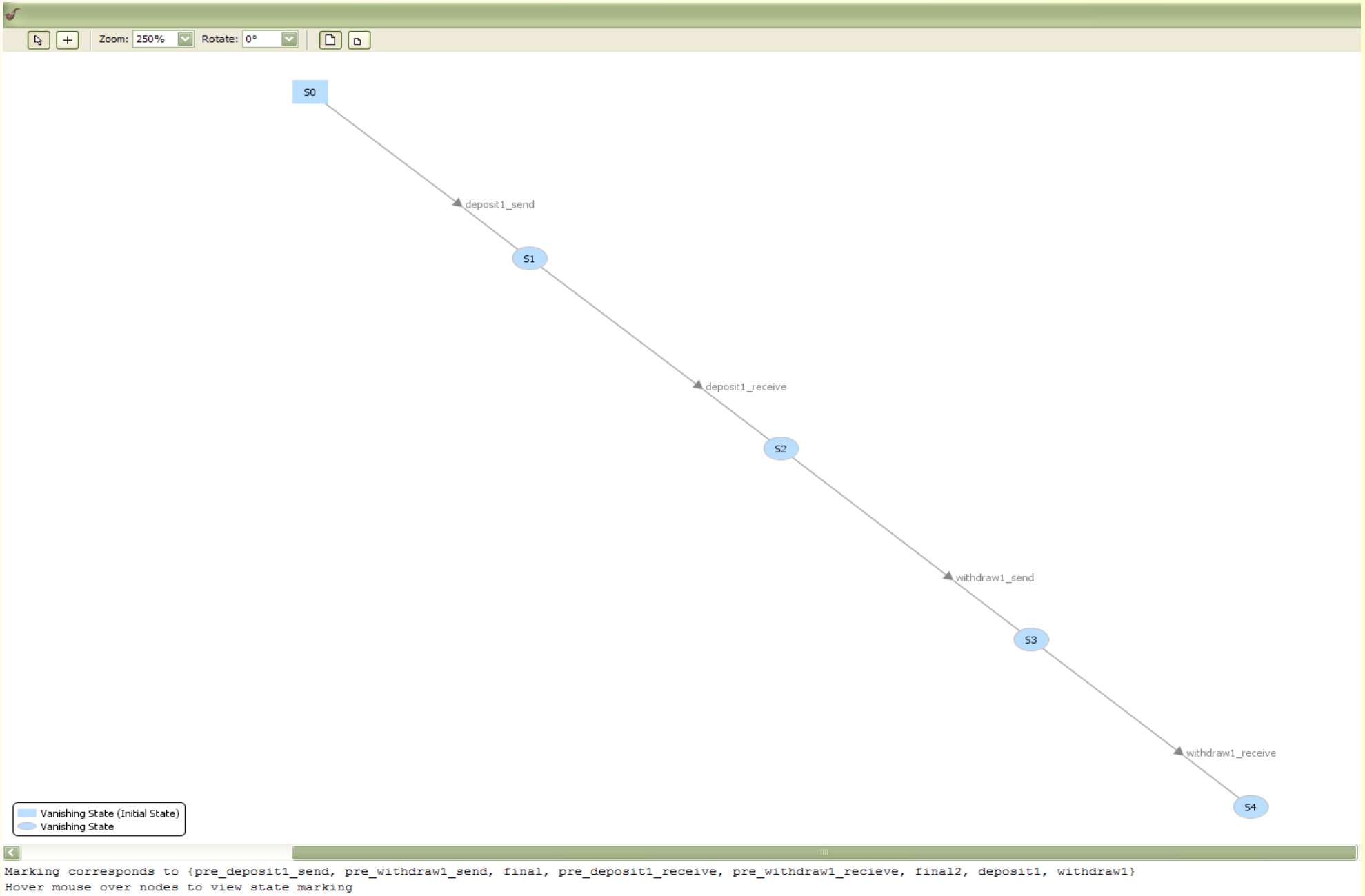
Reachability



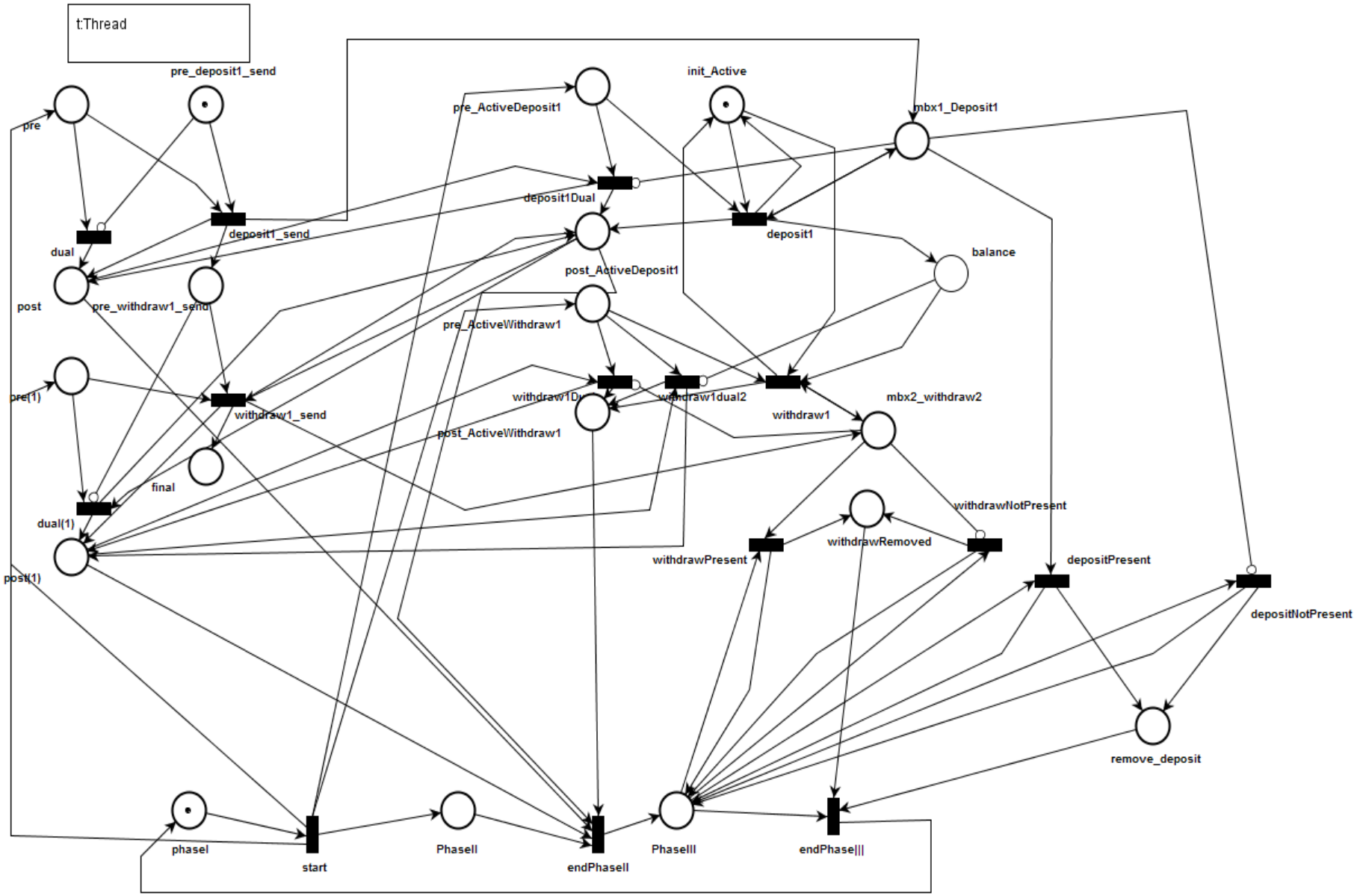
Consistent Case



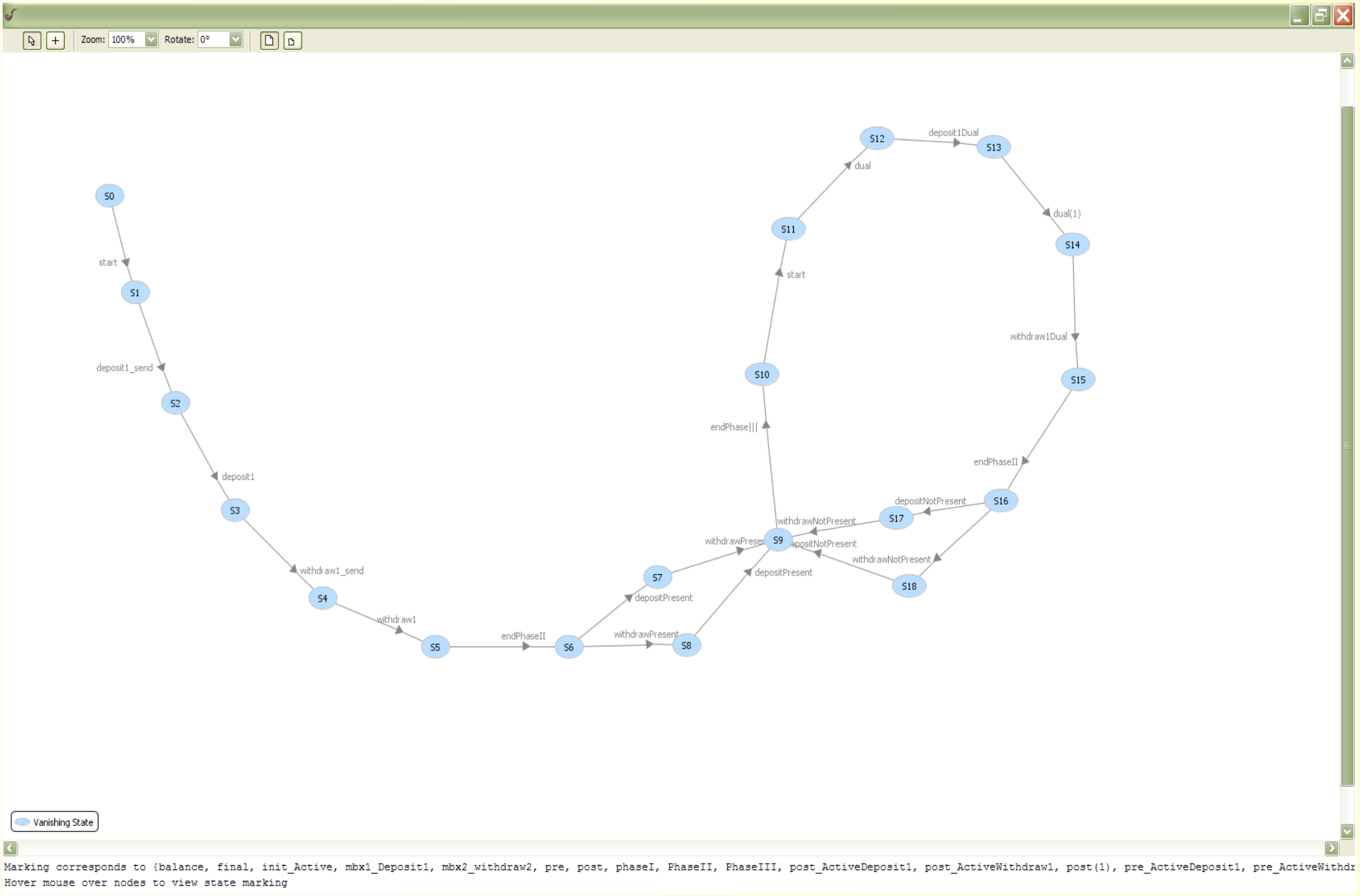
Reachability Graph

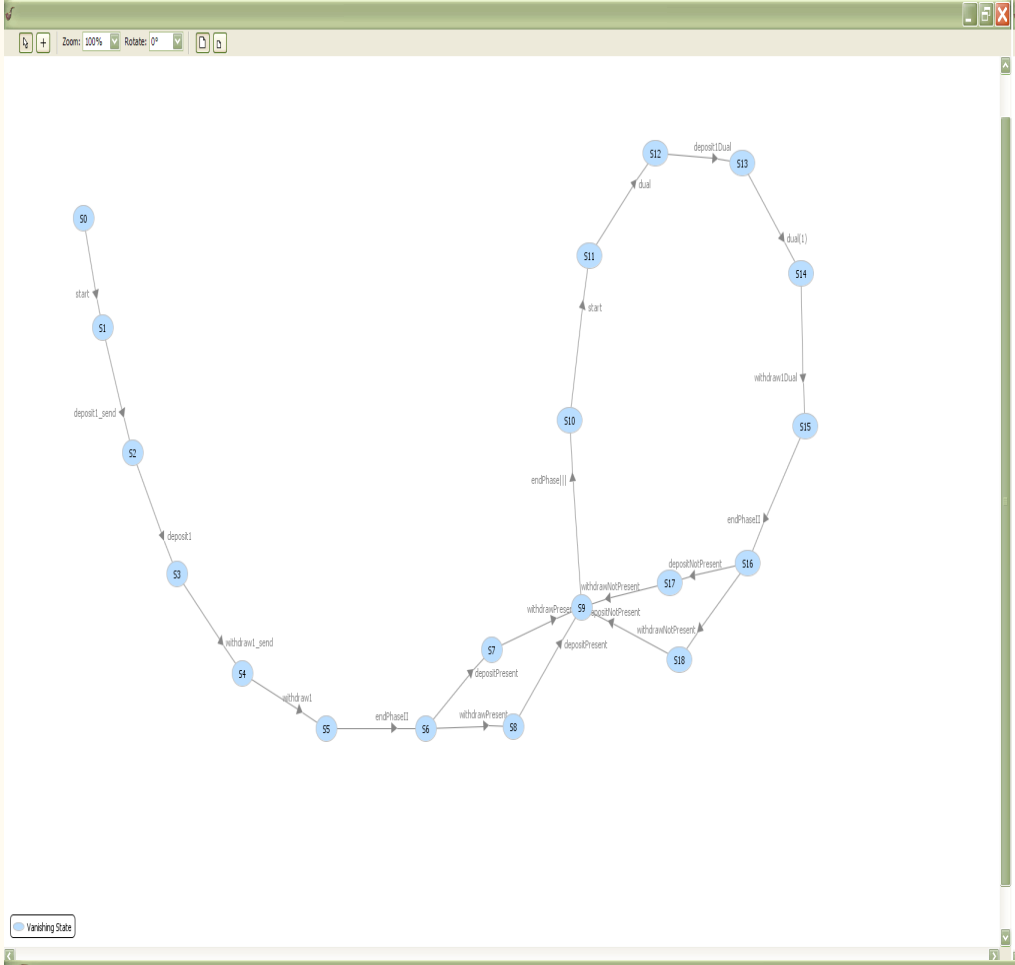


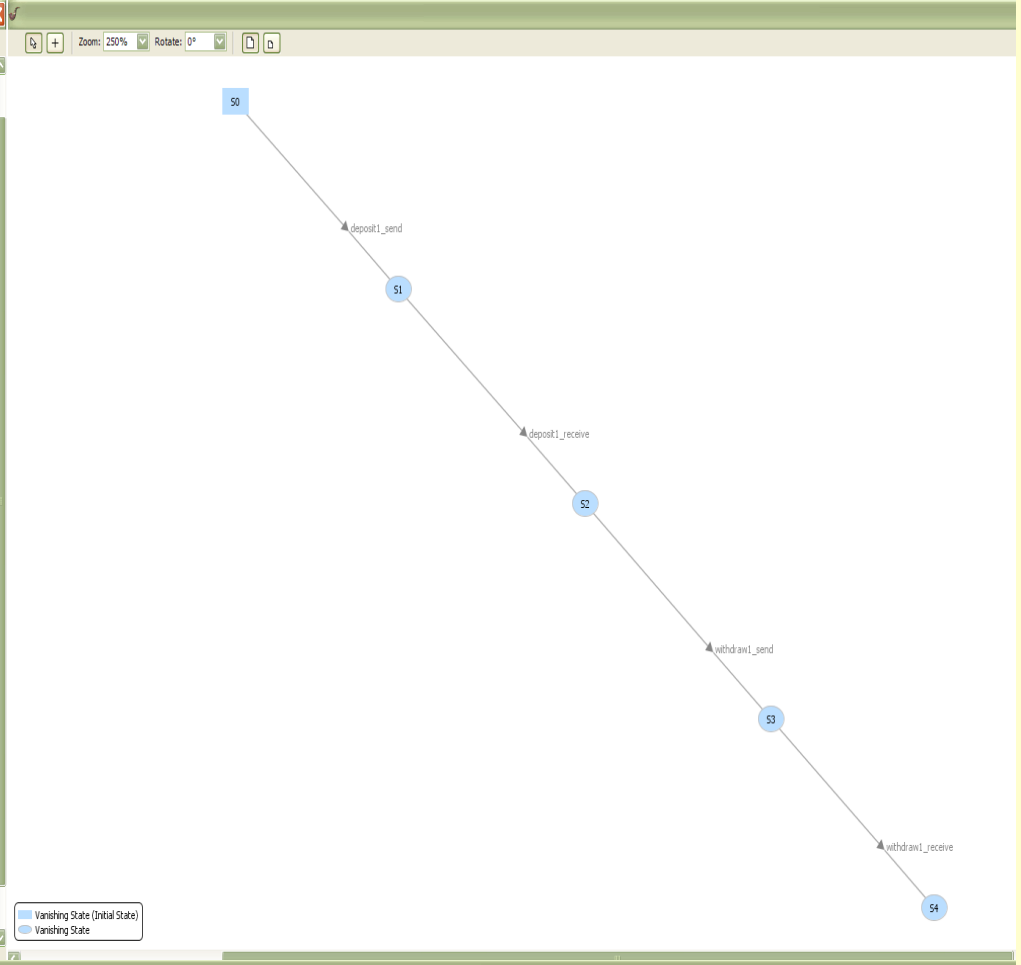
Consistent Case Final petri-net

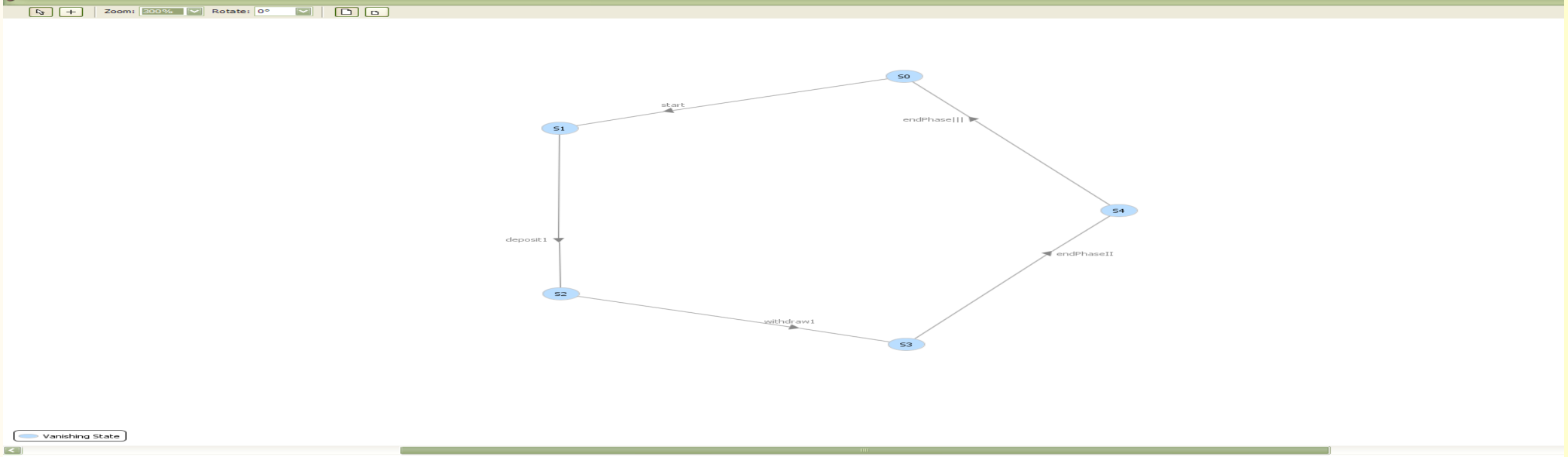


Analysis


Zoom: 100%
Rotate: 0°
S0
start
S1
deposit1_send
S2
deposit1
S3
withdraw1_send
S4
withdraw1
S5
endPhaseII
S6
withdrawPresent
S8
depositPresent
S7
withdrawPresent
S9
withdrawNotPresent
depositNotPresent
S17
depositNotPresent
S16
endPhaseII
S15
withdraw1Dual
S14
dual(1)
S13
deposit1Dual
S12
dual
S11
start
S10
endPhase|||
withdrawNotPresent
depositNotPresent
S18
Vanishing State
Marking corresponds to {balance, final, init_Active, mbx1_Deposit1, mbx2_withdraw2, pre, post, phaseI, PhaseII, PhaseIII, post_ActiveDeposit1, post_ActiveWithdraw1, post(1), pre_ActiveDeposit1, pre_ActiveWithdr
Hover mouse over nodes to view state marking

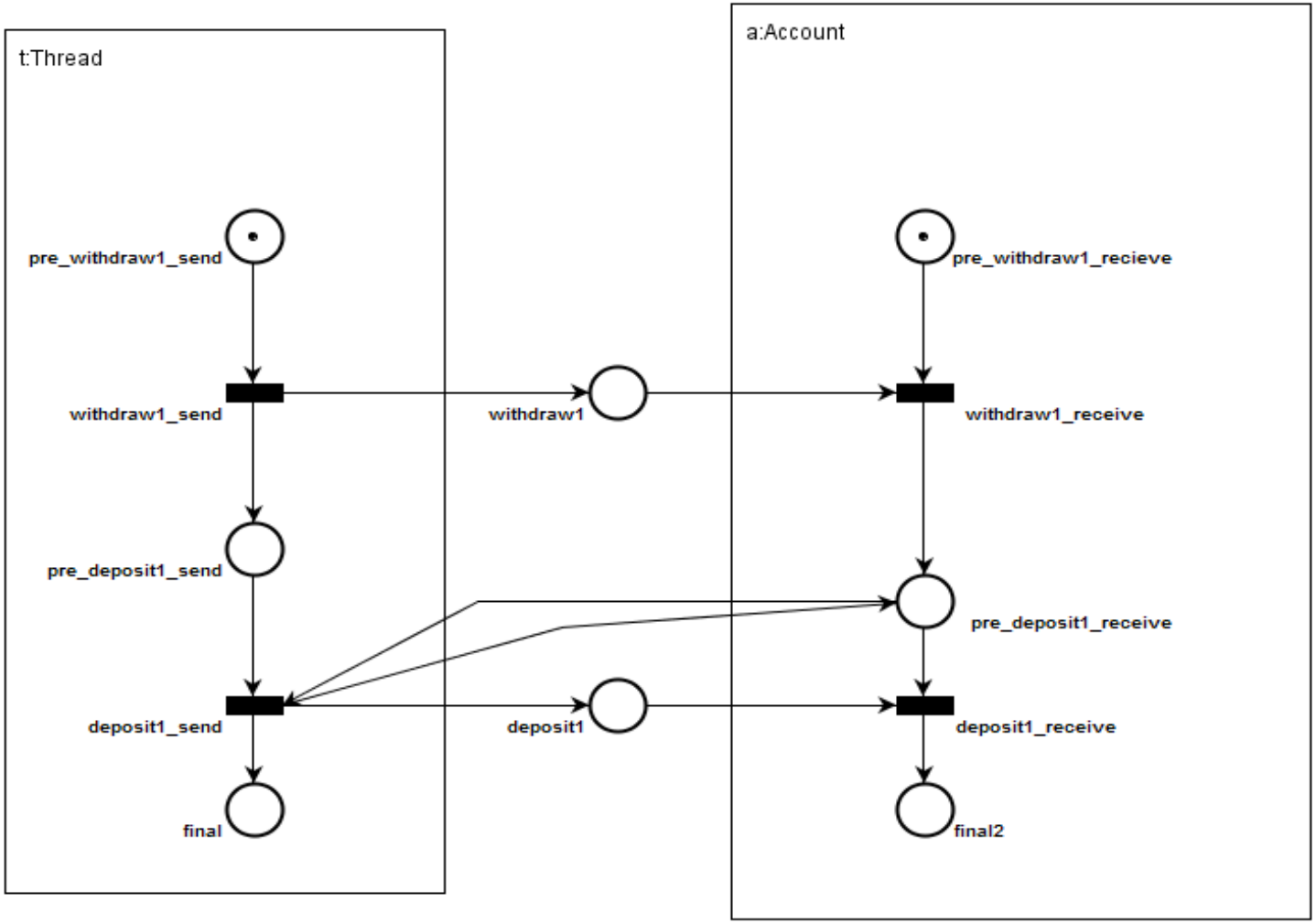




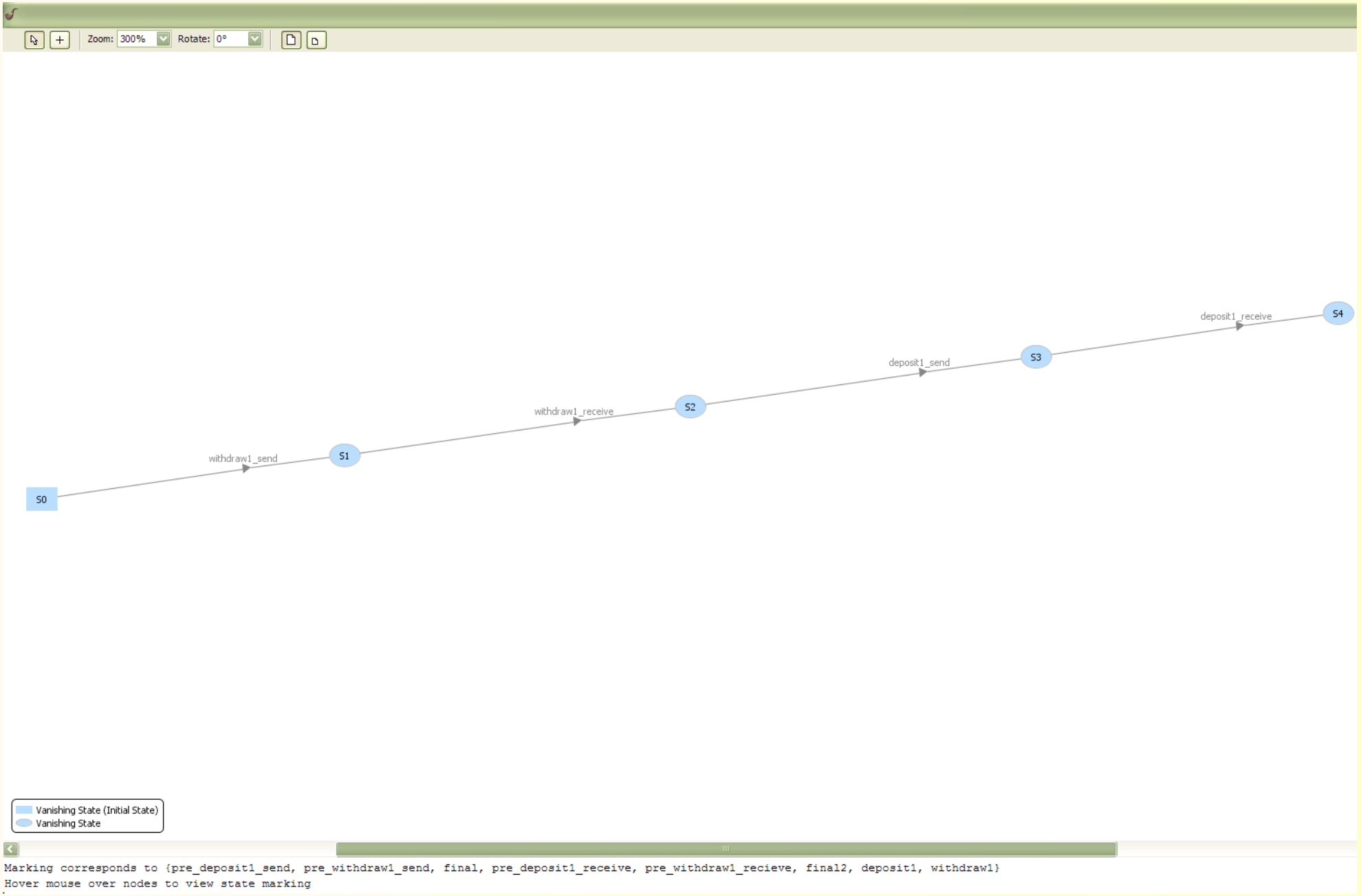


Marking corresponds to (pre_ActiveDeposit1, post_ActiveDeposit1, init_Active, mbx1_Deposit1, pre_ActiveWithdraw1, post_ActiveWithdraw1, mbx2_withdraw2, balance, phaseI, PhaseII, PhaseIII)
Hover mouse over nodes to view state marking

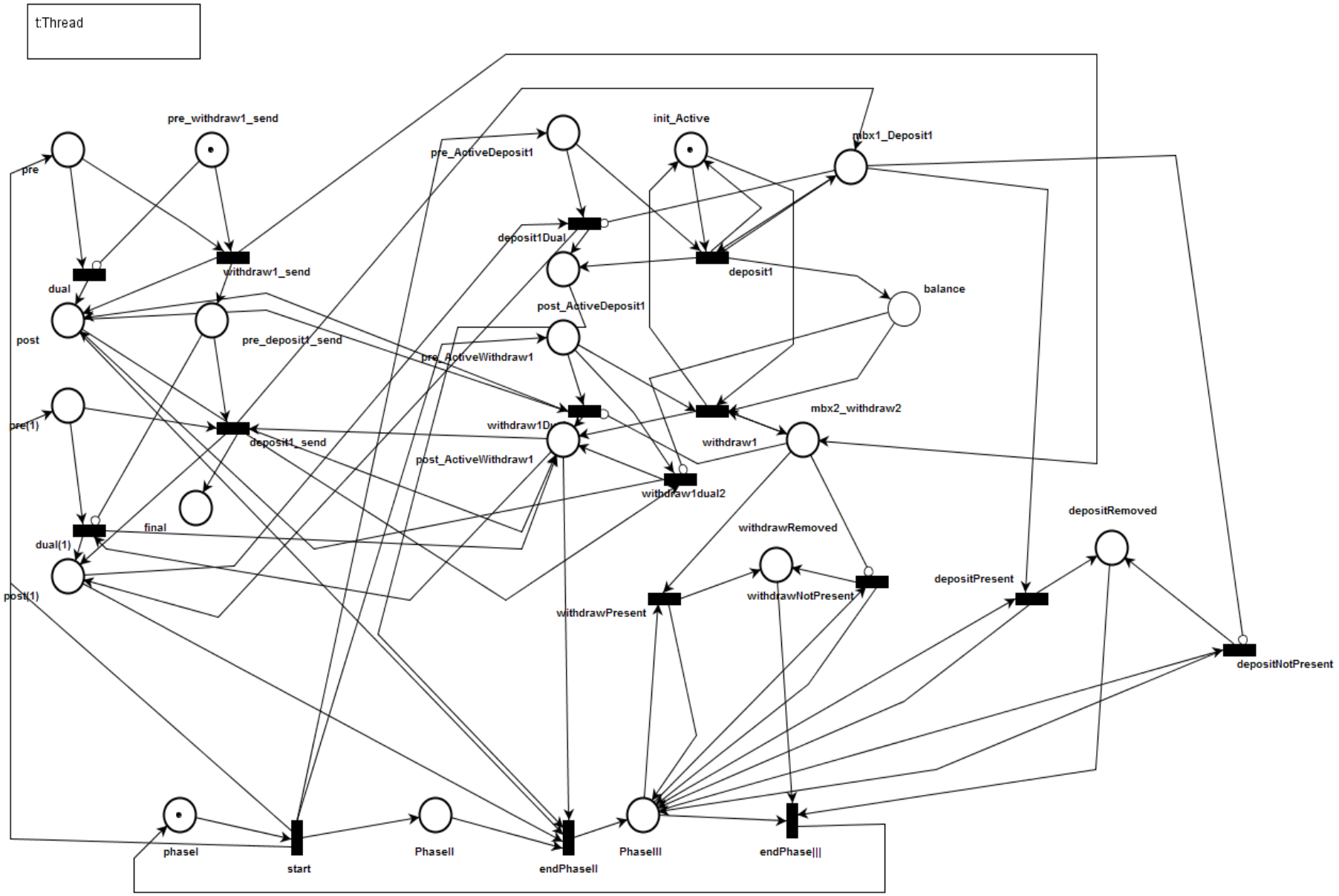
Inconsistent Case



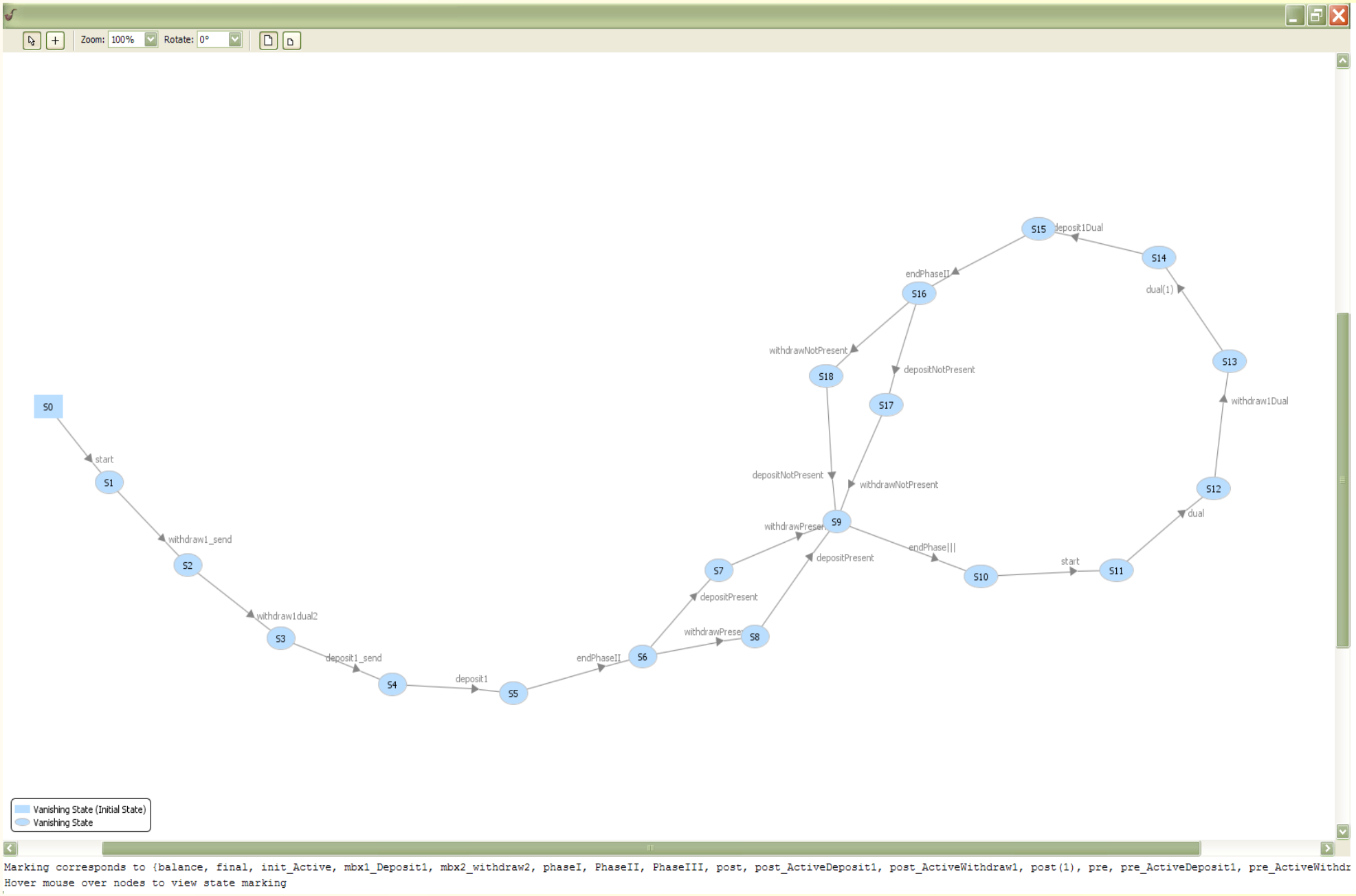
Reachability

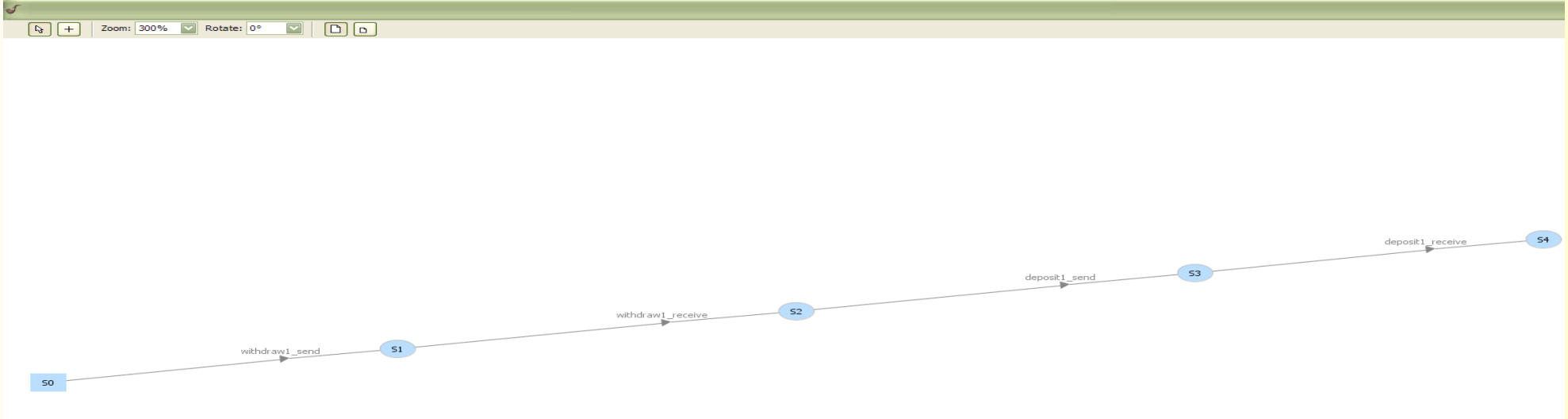


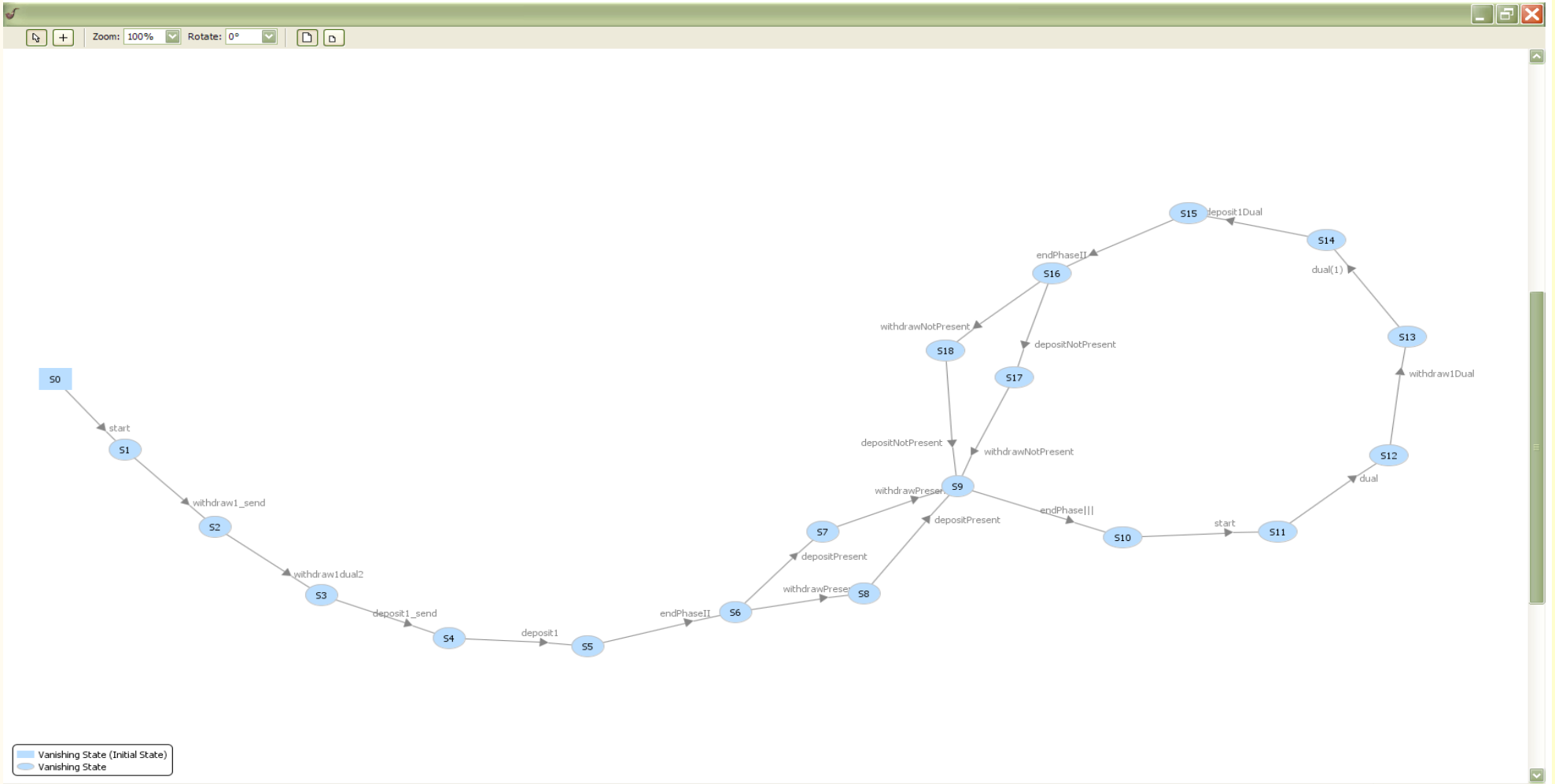
Inconsistent Case Final Petri-Net



Reachability Analysis







Marking corresponds to {balance, final, init_Active, mbx1_Deposit1, mbx2_withdraw2, phaseI, PhaseII, PhaseIII, post, post_ActiveDeposit1, post_ActiveWithdraw1, post(1), pre, pre_ActiveDeposit1, pre_ActiveWithdr
Hover mouse over nodes to view state marking

Conclusion
+
Future Work

- Only a subset of state charts and sequence diagrams
  - History, broadcast, guards
  - Alt, opt, ref, loops
- Need for automation
- Better analysis
  - Different tools maybe?

References

- Simona Bernardi , Susanna Donatelli , José Merseguer. From UML sequence diagrams and statecharts to analysable petri net models. Proceedings of the 3rd international workshop on Software and performance, July 24-26, 2002, Rome, Italy.
- Juan de Lara and Hans Vangheluwe. Computer aided multi-paradigm modelling to process petri-nets and statecharts. In International Conference on Graph Transformations (ICGT), volume 2505 of Lecture Notes in Computer Science, pages 239-253. Springer-Verlag, October 2002. Barcelona, Spain.
- Bas Graaf and Arie Van Deursan. Model-Driven Consistency Checking of Behavioural Specifications. Proceedings of the Fourth International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, Pages 115-126. March 2007. Braga, Portugal.
- Professor Hans Vangheluwe

QUESTIONS??