Workbook

On automaton

Made by:

joven, sean

likigan, Samuel

mateo, Christian

citcs-2h

What are functions?

> systems are always interacting with each other and input comes and goes all together

> one is a simple button and as you press it, something happens.

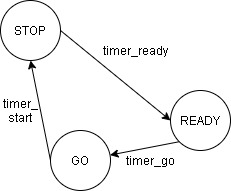
What are actions?

> events happening in an instance of time

> e.g. party, meeting, weddings, etc.

> we also call it atomic events, these have zero duration

An example would be a stoplight

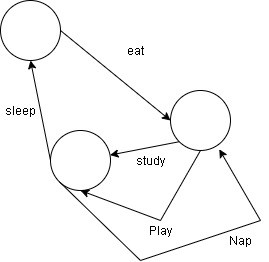
So in here it shows a behavior of how a stoplight would/should work. We have the initial state or the start state which is stop. Then transition using a timer we would arrive at ready state. The timer for transition then it gets to state go.

\* states are stop, ready and go

\* transitions are timer\_ready, timer\_go and timer\_ start

Trace: timer\_ready, timer\_go, timer\_stop

> this shows a trace/system trace.

Exercise: 10pts Identify the diff. traces.Answers: eat,study,sleep

eat,play,sleep

eat,study,nap,study,sleep

eat,play,nap,study,sleep

eat,study,nap,play,sleep

eat,play,nap,play,sleep

extra: Is this part of the trace

eat,study,nap,study,nap,play,nap,play,study,nap,study,sleep

What is a non-deterministic behavior?

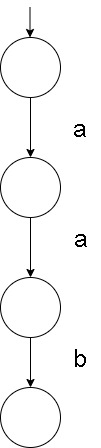
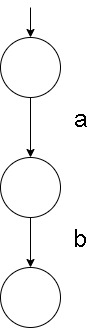
> if and only if it has only two out-going transitions

> explaining: if it has 2 of the same transition, example would be the stop light, you were suppose to stop at the read light but you still go since there are no cars coming along. Another would be an alarm clock it would still sound even though you made it stop.

What is a deterministic behavior?

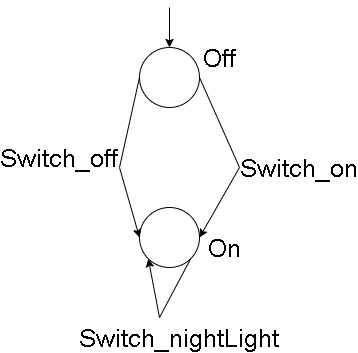
> only has 1 out-going transition

> since NDB has 2 of the same deterministic only has 1 and only 1 transition, example would be the light bulb, its off then switch its on, and Vice Versa nothing more less.

Non-deterministic Deterministic

Simple Exercise: 15pts: Does this show non-deterministic behavior? Explain

It is since it’s still on. Has 2 of the same transitions



Identify the diff. traces:

Switch\_on,Switch\_nightLight,Switch\_off

Switch\_on,Switch\_off

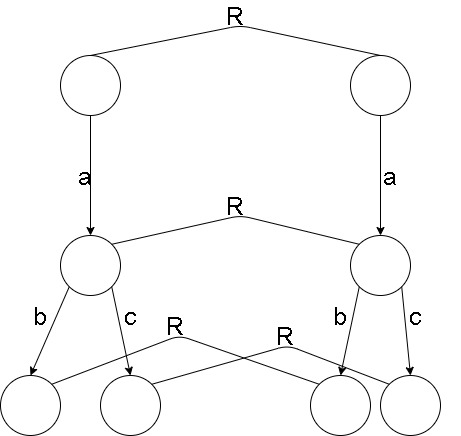
Does switch\_nightlight keep the light on infinitely? Reason?

Yes it would still stay at the state on given the image and when tracing it would be switch\_nightlight…

Bisimulation Equivalence \* Bisimilar

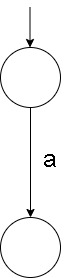
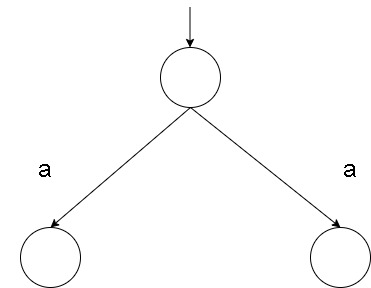
> if A has similar transitions with B then they are equal

> All transitions A have similar transitions with B.

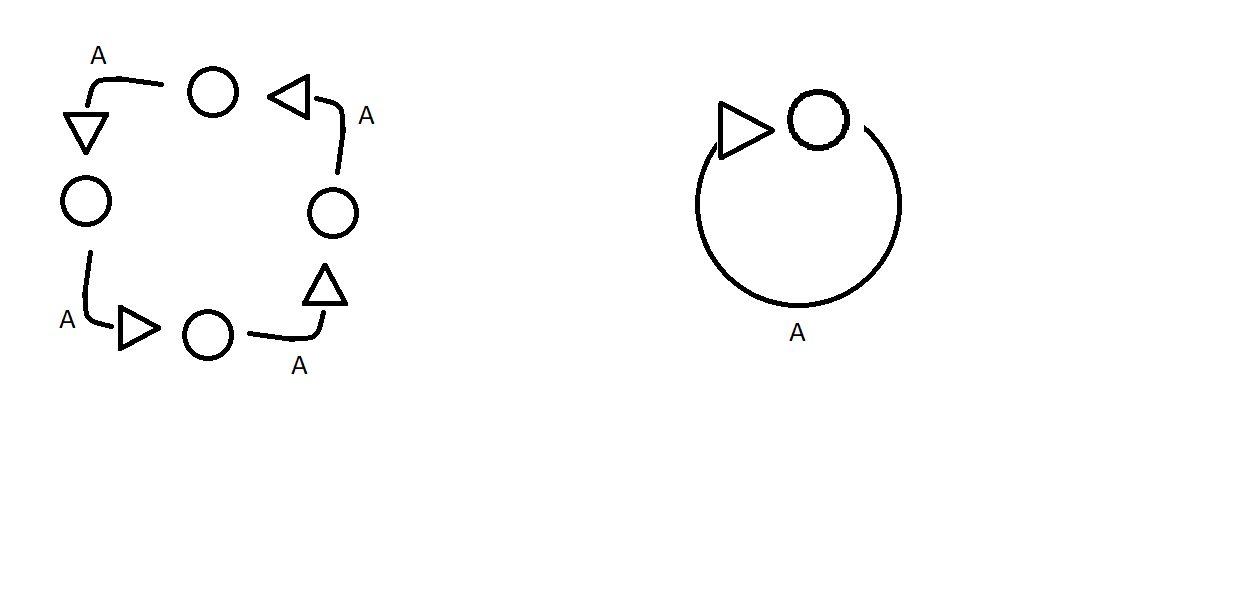


Theorem: Every transition system has a unique minimal transition system that is bisimulation equivalent to it.

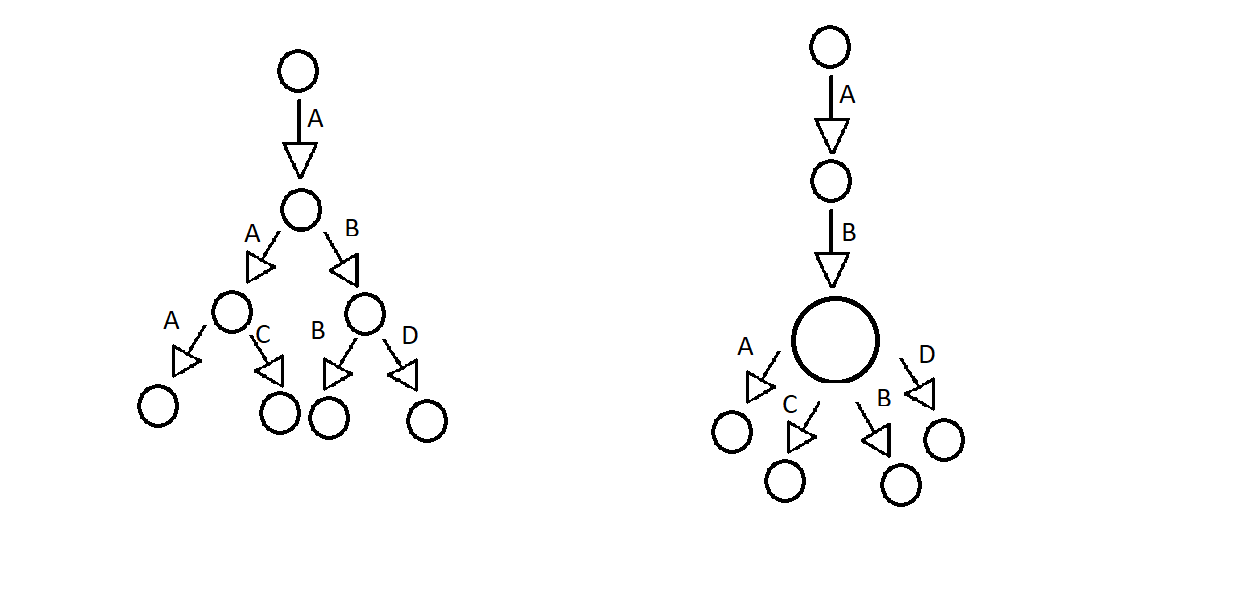
\* Reason why this is not bisimilar is that there are 2 transitions of a and the other only has 1



\*This just minimalizes it and since they show similar behavior. This means a is repeating itself.

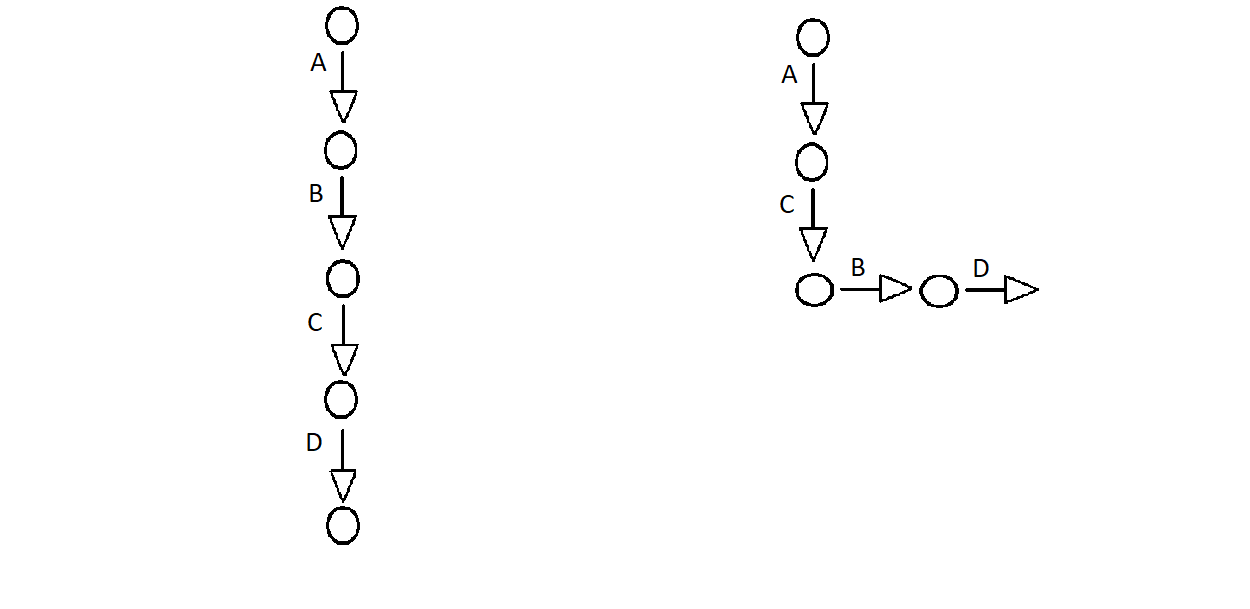
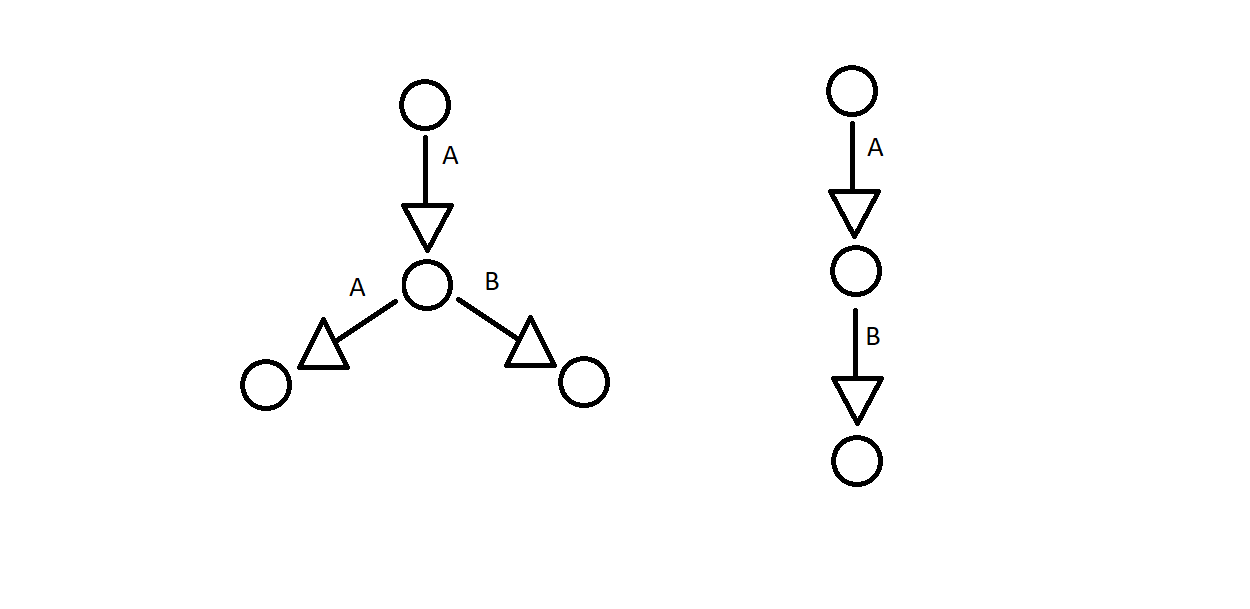


This is an example of the theorem

Exercise: 20 pts Does this show bisimulation equivalence?

\*No because at transition b cannot do two different paths at the other one.

\*No because at transition b cannot do two different paths at the other one.



\* Yes, same transitions as with the other.

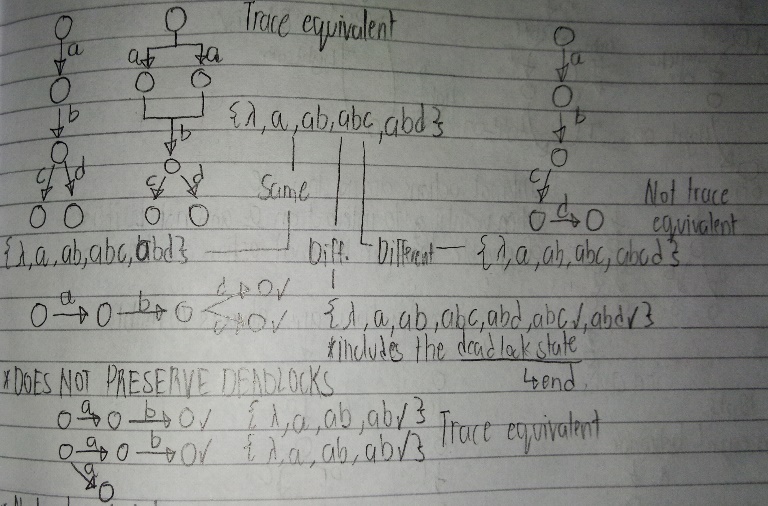
Trace Equivalence

> sequence of actions that can be executed

> {this} = {this} {A,B} = {A,B} {A} is not equal {B}

\*Nasty to calculate

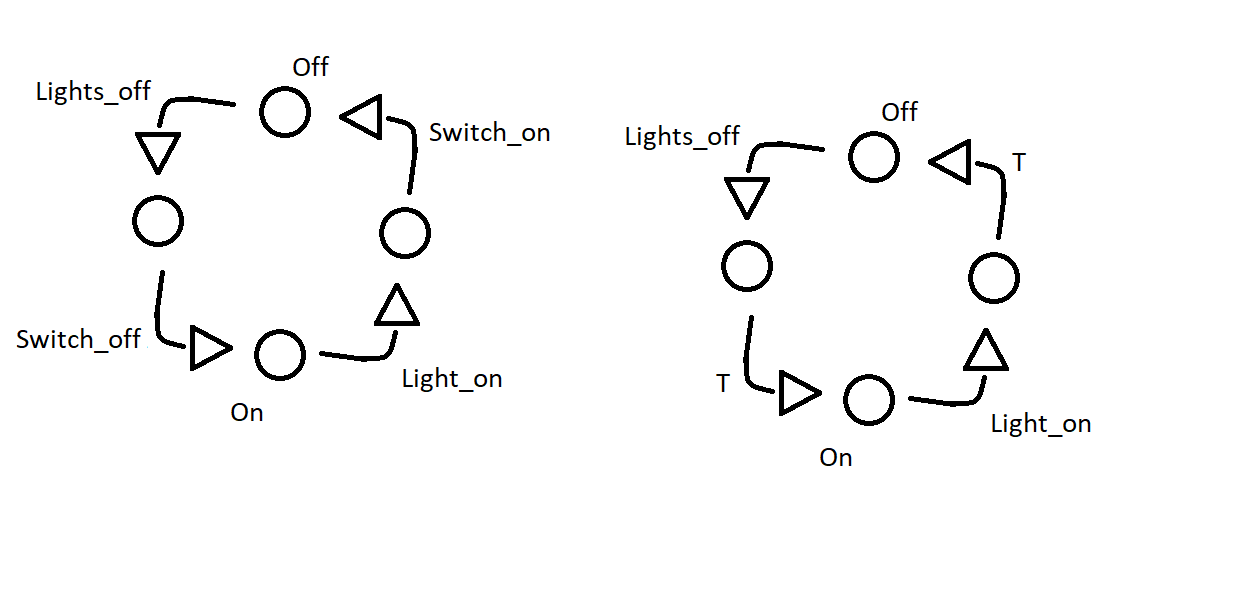
> an infinite transition system requires O(m log n) time n states m transitions

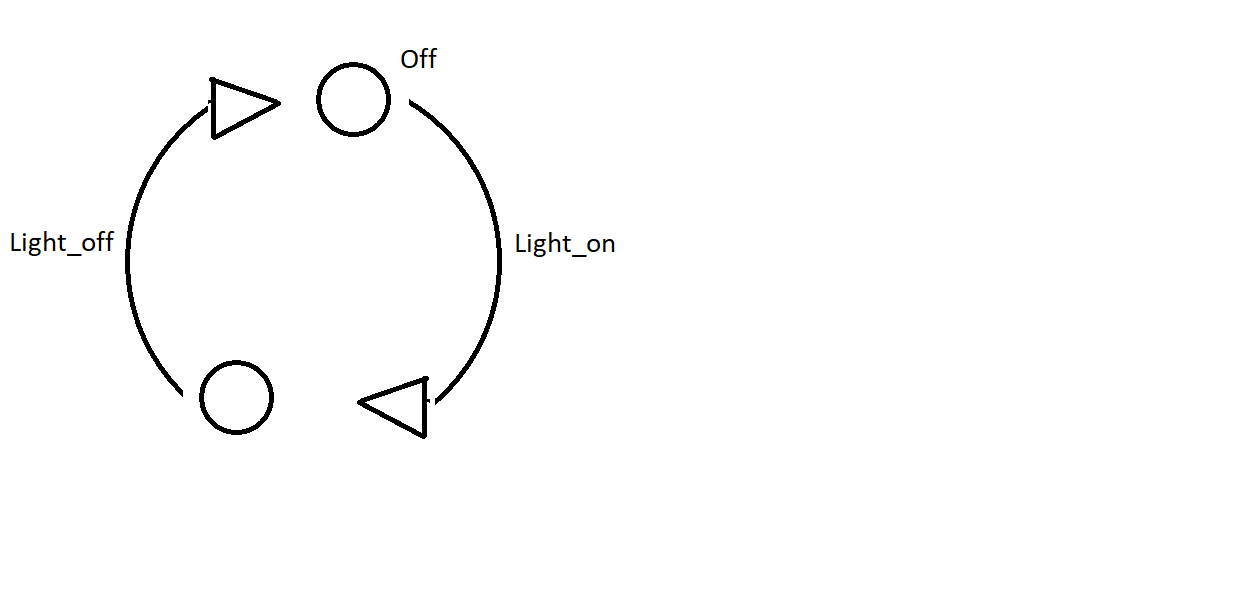
> For Deterministic Transition System bisimulation and trance equivalence coincide

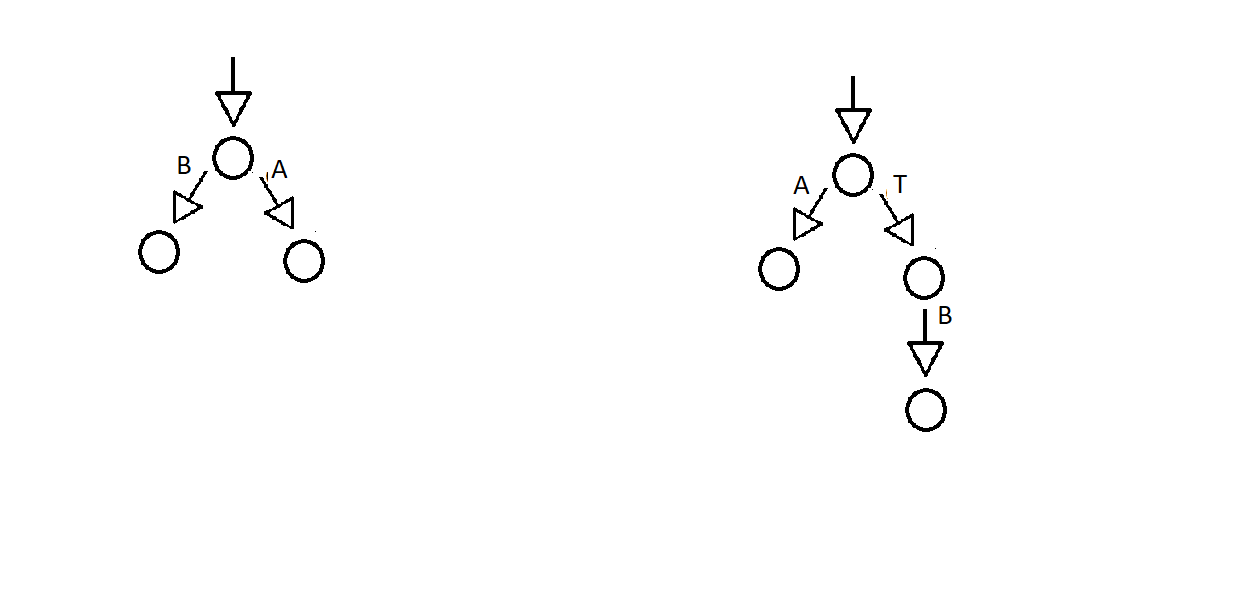
Internal Action or tau action

> ordinary action which we cannot observe directly

Example: robbing a bank, secretly escaping, atoms splitting, cells splitting

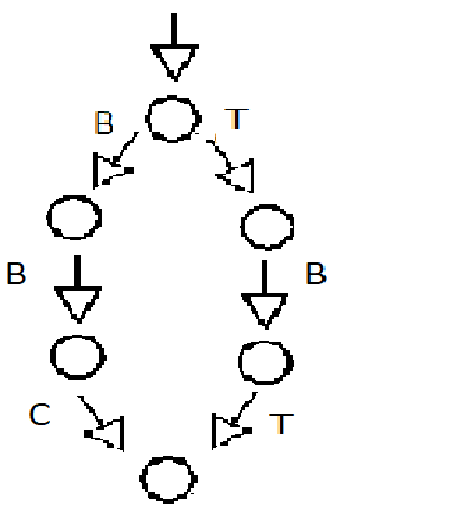
Internal actions are not invisible



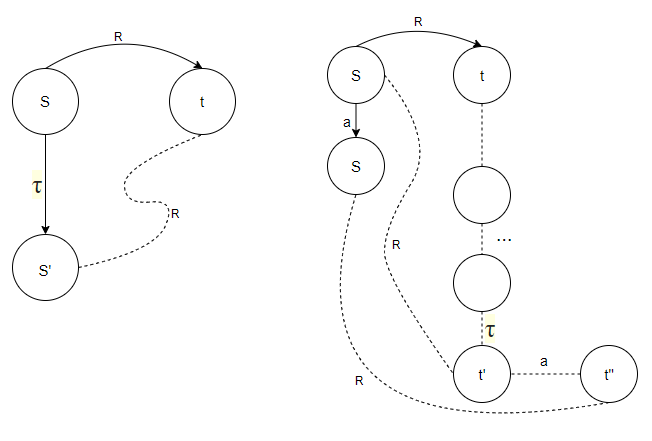
When wanting to interact on a and not b it is observable that an internal action can happen.

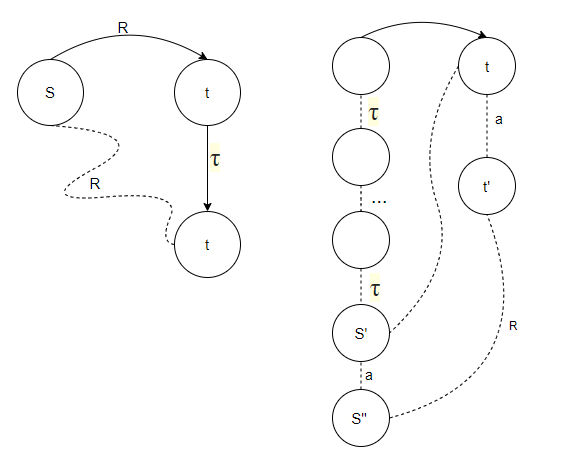
Exercise: 10pts

Show an equal behavior

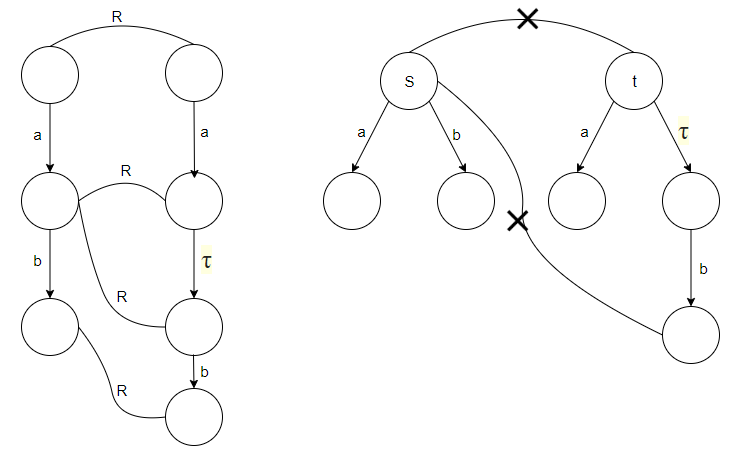


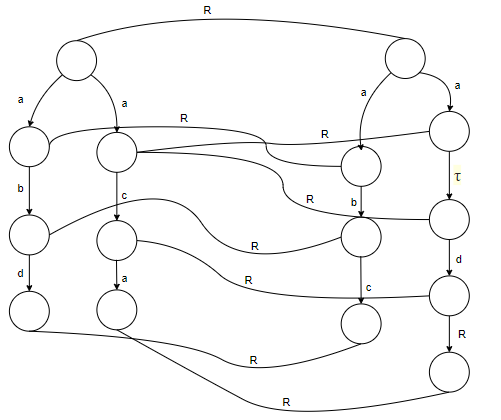
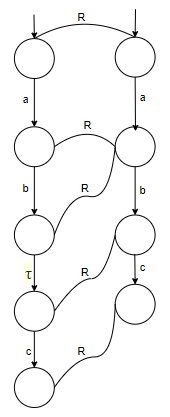
Branching Bisimulation

> combination of bisimulation and internal actions

>same condition as Bisimulation equivalence the difference is the addition of hidden action

> not branching because a cannot do the action at the left(figure at the left) while the other one is a branching bisimulation(figure at the right).





Question(30pts)? Are these branching bismilar? Explain.

Combining behavior

> rooted branching bisimulation relation R is a branching bisimulation that also satisfies

\*Strong bisimulation implies Rooted Branching Bisimulation implies branching bisimulation

Divergence Branching Bisimulation

>an infinite loop of tau or internal actions, this state is now called a “divergence”

> In branching bisimulation it removes the internal actions, in another term it merges them and then it will be removed.

> a way to avoid this is through the condition that  
if one state has an infinite loop of internal action then the other also has an infinite loop as well.

Weak Trace Equivalence

> the same as trace equivalence but we omit the internal action.

{{},a,ab,ac}

{{},a,ab,ac}

Properties

> Weak trace reduction allows to remove all internal actions from behavior

> There is no unique minimal transition system modulo weak trace equivalence

> it is hard to calculate, use it on small transitions

Weak Bisimulation

> has the same properties as branching bisimulation the only difference here is that the end points are the only ones related

> another is that it has the same symmetric case and that rooted weak bisimulation just needs at least 1 related tau in order it to be a rooted weak bisimulation. Another neat thing is that rooted is more efficient than weak bisimulation.

Complete Trace Equivalence

> a complete trace has a deadlock or a terminating state

> though sometimes blocking it or preventing for a transition to happen may lead to an incomplete trace.

Failure Equivalence: completed trace congruence  
Refusal Set: Set of actions that cannot be done in a state

Failure Pair: trace and the refusal set where the trace ends

When to use which behavioral equivalence

> Strong bisimulation. Always safe

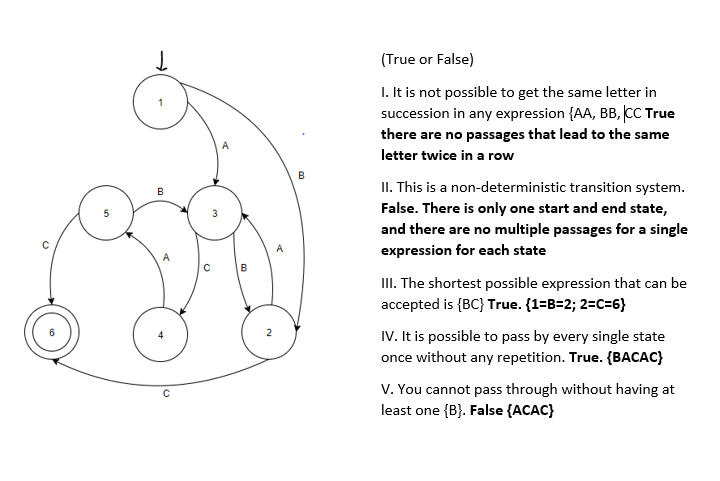
> Trace equivalence. Sequences are important and branching behavior is not.

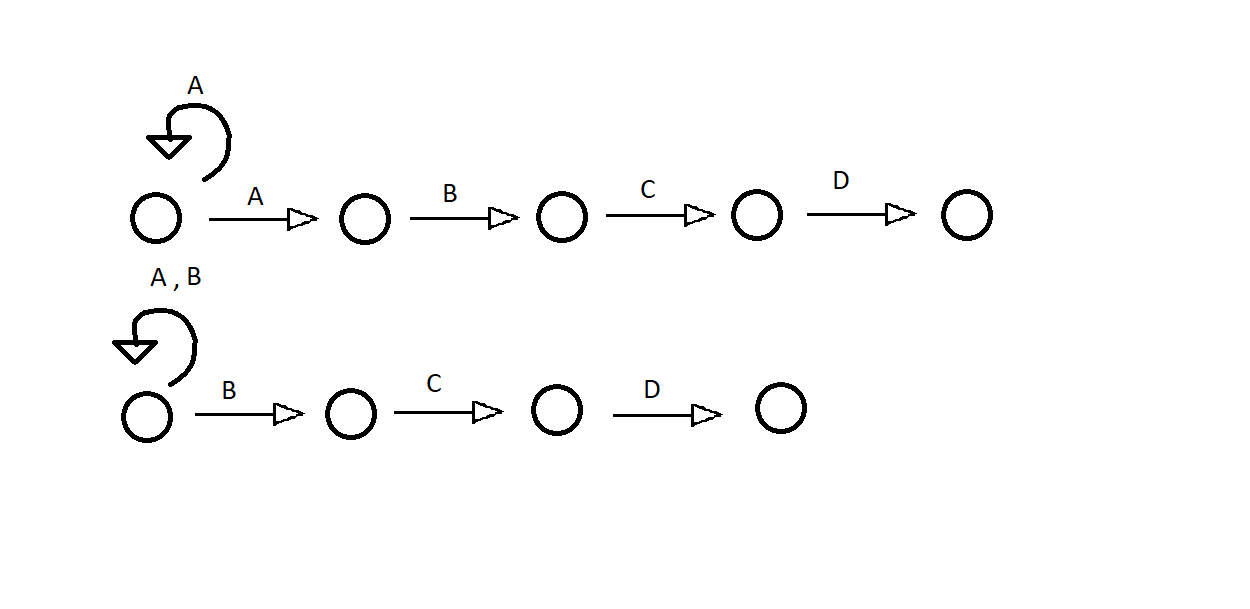
> Failure equivalence. If behavior is not strongly bisimilar, this is a 2nd best choice, in a setting with ordinary process operators.

>Divergence preserving branching bisimulation. Used to remove internal steps. Always a safe choice.

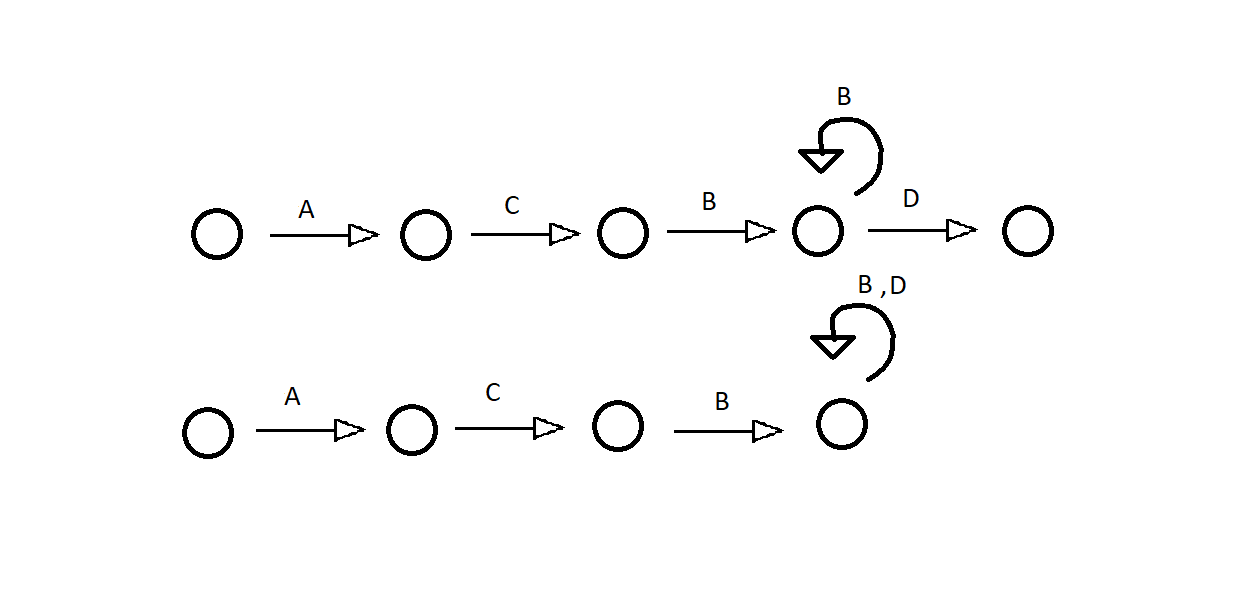
>Branching bisimulation. Safe choice when eventualities are not important as it removes infinite tau loops.

Weak Bisimulation. For practical purposes generally equivalent to branching bisumulation. Remove’s all tau. Does not preserve branching behavior.

Other Exercises:

Give the trace of these figures and determine if they are equal.

{A,AA,AAA…,AB,ABC,ABCD}

{A,B,AB,BB,ABCD,BBCD}

{A,AC,ACB,ACB,ACBD,ACB,ACBB,ACBBB,…,}

{A,AC,ACB,ACBB,ACBD,…}