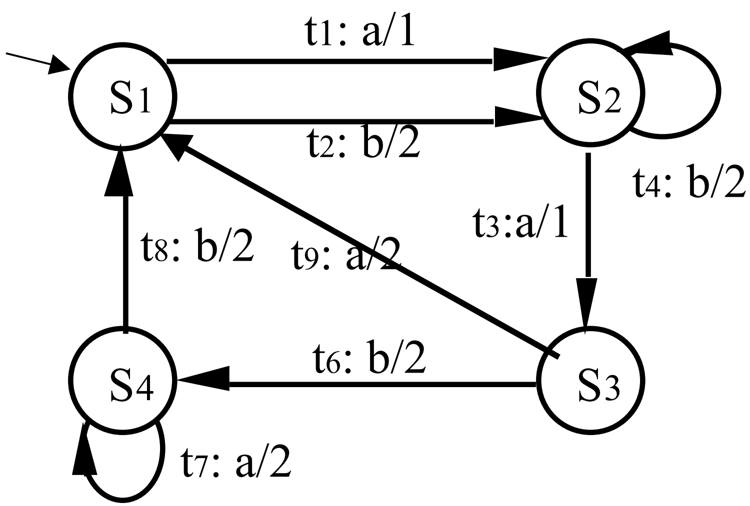
Transition Tour example

Test hypothesis: Initially connected machine



Transition tour

TT: t_1, t_4, t_3, t_9, t_2, t_3, t_6, t_7, t_8

TT (input/expected output): a/1.b/2.a/1.a/2.b/2.a/1.b/2.a/2.b/2

All state identification Methods

Distinguishing Sequence, UIO, W

Test hypothesis

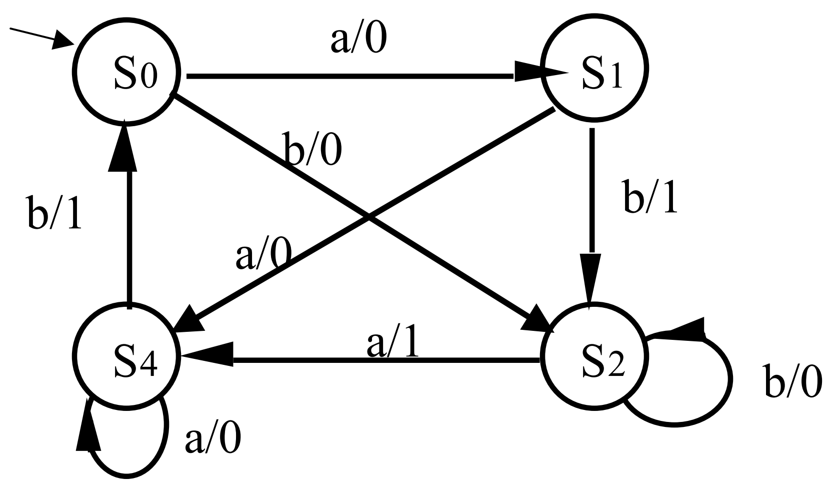
- H1) Strongly connected machine
- H2) Contain no equivalent states
- H3) deterministic
- H4) Completely specified machine
- H5) the failure which increases the number of states doesn't occur

The method is applied in two phases from the initial state

- phase 1) α-sequence to check that each state defined by the specification also exist in the implementation.
- phase 2) β-sequence to check all the individual transitions in the specification for correct output and transfer in the implementation.

DS Method

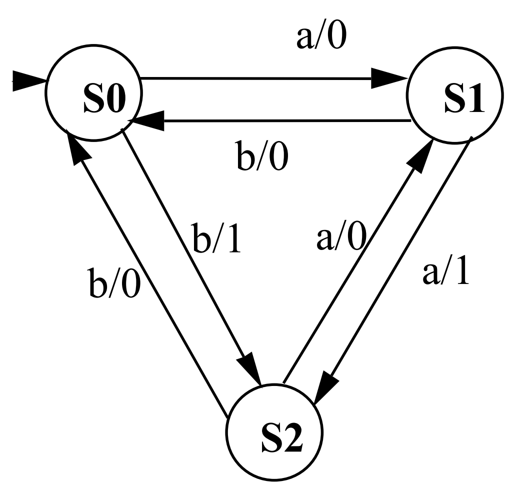
Assume that Reset transition r/- exist
Q1) Verify if “a.a” is a DS for S and explain why?
Q2) Find a DS different from “a.a” with length 2 for S



Wp method

Assume the reset is exist and bring the machine from any state to the initial state.

a) Find characterization set W and generate the set of test cases for the specification S using the Wp method.
b) Does S have a DS sequence? If not explain why?



Wp method

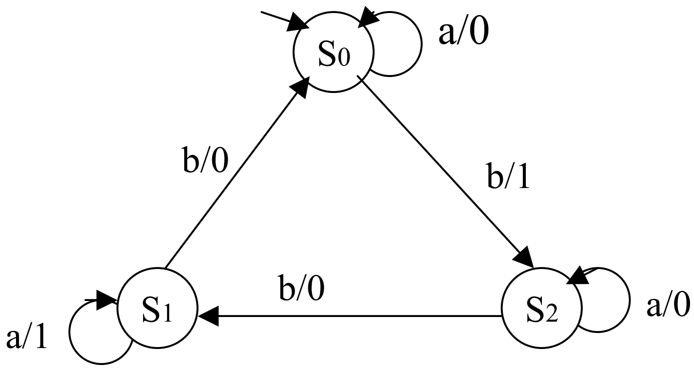
W={a, b}

S0 : a/0, b/1
S1 : a/1, b/0
S2 : a/0, b/0

Q = { ,a,b}
P = { ,a, b, a.b, a.a, b.a, b.b}

Phase 1 , Q.W = {r.a, r.b, r.a.a, r.a.b, r.b.a, r.b.b}
Phase 2, (P-Q).W = {r.a.b.a, r.a.b.b, r.a.a.a, r.a.a.b, r.b.a.a, r.b.b.a, r.b.b.b}

Examples Suite



Specification S

Transition tour:

Input	a.b.a.b.a.b
Output	0.1.0.0.1.0

Derive a DS of length up to 2 for S

a.b is a DS for S

State	S_0	S_1	S_2	S_0	S_1	S_2	S_0	S_1	S_2	
Input	a	a	a	b	b	b	a.b	a.b	a.b	
Output	0	1	0	1	0	0	0.1	1.0	0.0	

Comment: “a” as input at each state will loop on the state, sequence of “a.a.” cannot be a DS, the output will be 0.0.. or 1.1…

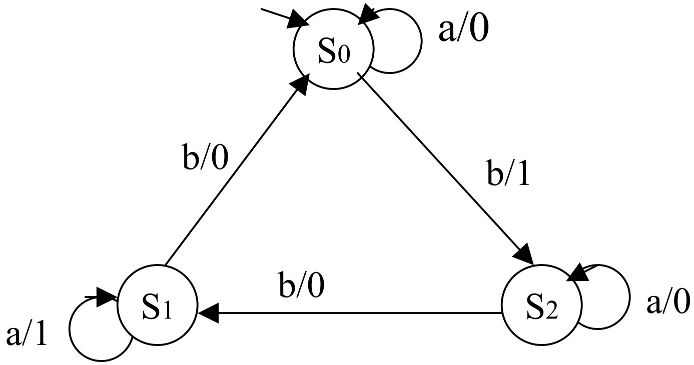
Q set: permits to reach each state from the initial state

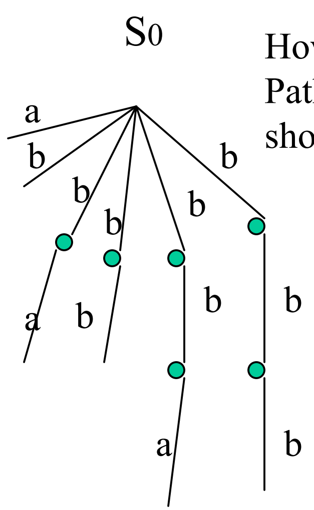
Q = { , b,b.b}

The first “b” to reach the state S_2

“b.b” to reach the state S_1.

P set is transition cover, permits to execute each transition at least one starting from the initial state





How to derive P set: find all Path starting from the size1 and up and each transition should be traversed at least once

P = {, a, b, b.a, b.b, b.b.a, b.b.b}

more than one p set may exist, this depends on the alternative paths that the automata may have.

The goal of the Phase 1 is identification of
the states in the implementation

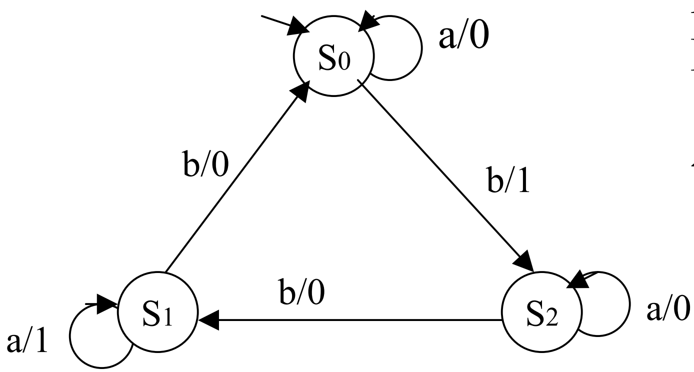
DS = a.b, **Q** = { , b,b.b}, P = {, a, b, b.a, b.b, b.b.a, b.b.b}

Phase 1

Q.DS = {r.**a.**b, r.b.**a**.b, r.b.b.**a.b**} Expected output of phase 1is:
{-.0.1, -.1.0.0, -.1.0.1.0}

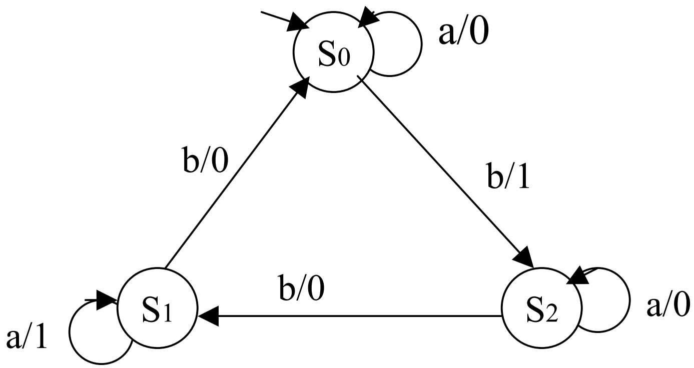
Phase 2 (DS in bold)

P.DS= {r.**a.b**, r.a.**a.b**,r.b.**a.b,** r.b.a.**a.b**, r.b.b.**a.b,** r.b.b.a.**a.b,** r.b.b.b.**a.b**}
{-,0.1, -.0.0.1, -.1.0.0.0, -.1.0.1.0, -.1.0.1.1.0, -.1.0.0.0.1}

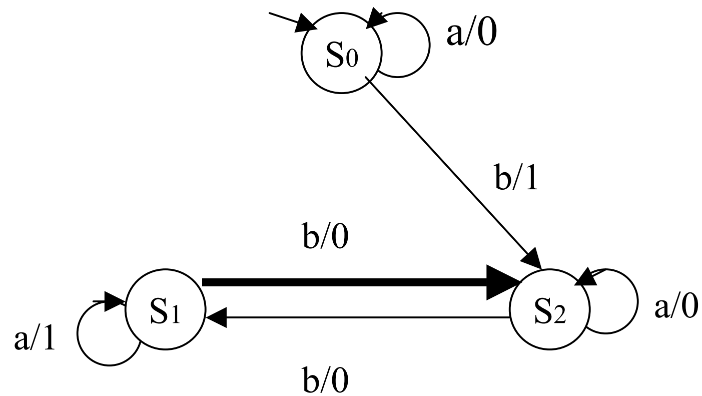


Note that, the test suites for phase 1 and 2 should be Derived from the specification and applied to the implementation to check it for output and transfer faults.

Specification S



Implementation I



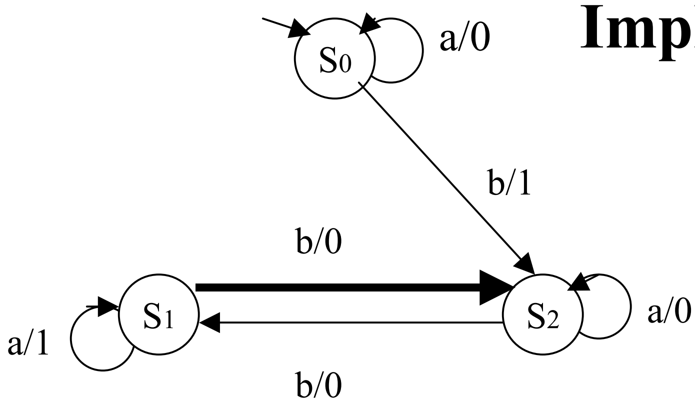
Apply the transition tour to the implementation I and comment

Transition tour applied to S

Input	a.b.a.b.a.b
Output of S	0.1.0.0.1.0
Output of I	0.1.0.0.1.0

The implementation I has a transfer fault,
the fault is not detected by
Transition tour.
Transition tour detects all output faults but
Doesn’t guarantee the detection of transfer faults

Implementation I

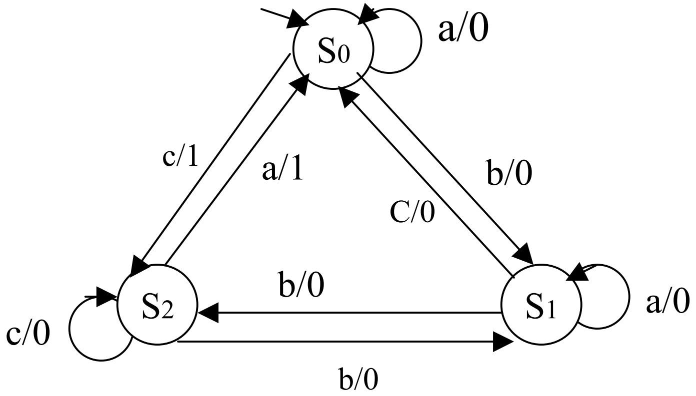


The goal of the Phase 1 is identification of
 the states in the implementation
DS = a.b, **Q** = { , b,b.b}, P = {, a, b, b.a, b.b, b.b.a, b.b.b}

Phase 1
Q.DS = {r.a.b, r.b.a.b, r.b.b.a.b} Expected output of phase 1is:
{-.0.1, -.1.0.0, -.1.0.1.0}
{-.0.1, -.1.0.0, -.1.0.1.0}) observed outputs from I

Phase 2 (DS in bold)
P.DS= {r.**a.b**, r.a.**a.b**, r.b.a.**a.b**, r.b.b.**a.b,** r.b.b.a.**a.b,** r.b.b.b.**a.b**}
{-.0.1, -.0.0.1, -.1.0.0.0, -.1.0.1.0, -.1.0.1.1.0, -.1.0.0.0.1} expected output
{-.0.1, -.0.0.1, -.1.0.0.0, -.1.0.1.0, -.1.0.1.1.0, -.1.0.0.0.**0**}observed output from I, transfer
↑ fault detected

Specification S



Derive a UIO sequence for S

State	S0 S1 S2	S0 S1 S2	S0 S1 S2	S0 S1 S2
Input	a a a	b b b	c c c	a.c a.c a.c
Output	0 0 1	0 0 0	1 0 0	0.1 0.0 1.1

UIO state S_0 = c/1
UIO state S_2 = a/1
UIO state S_1 = a/0.c/0

Transition tour for S

a.b.a.b.c.a.c.b.c
0.0.0.0.0.1.1.0.0