ETH zürich

dynamo.ethz.ch

Temporal Logic and Model Checking

Jiahui Xu
DYNAMO group

Computer Engineering and
Networks Laboratory

Schedule

We have four exercise sessions:

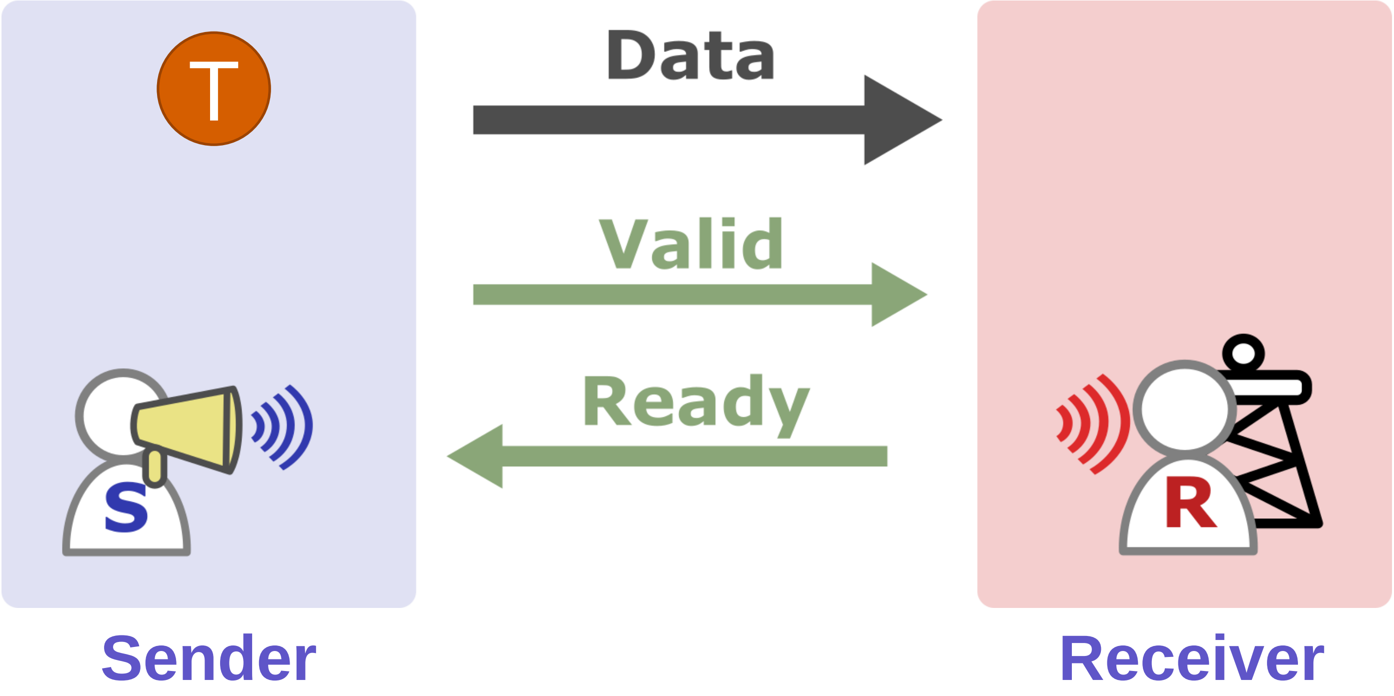
- 30.11.2023: set operations, characteristic functions, BDDs
- 07.12.2023: reachability analysis and temporal logic
- 14.12.2023: Petri nets
- 21.12.2023: time Petri nets

Specification Using Temporal Logic

Elastic systems: **computation modules** interconnected by **channels**.
Channels: propagate data, equipped with bidirectional **handshake signals**.

Specification Using Temporal Logic

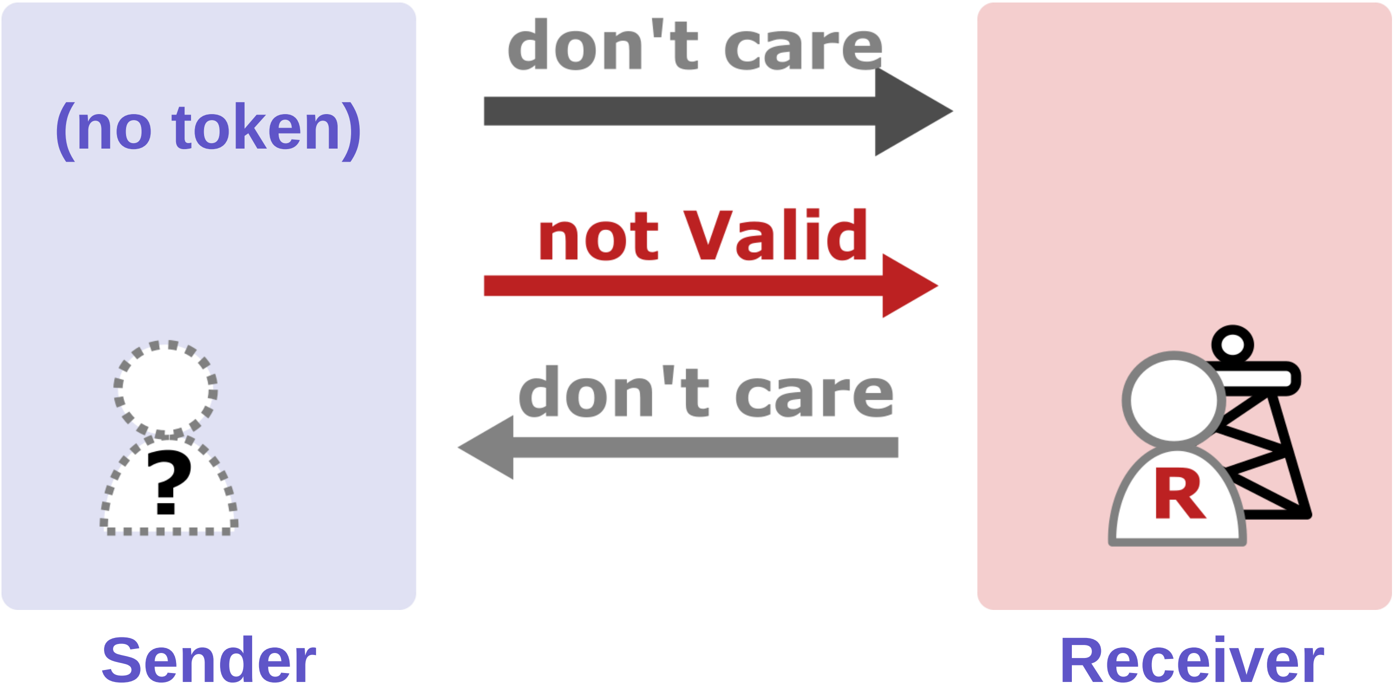
Elastic systems: **computation modules** interconnected by **channels**.
Channels: propagate data, equipped with bidirectional **handshake signals**.



This state is called **transfer**

Specification Using Temporal Logic

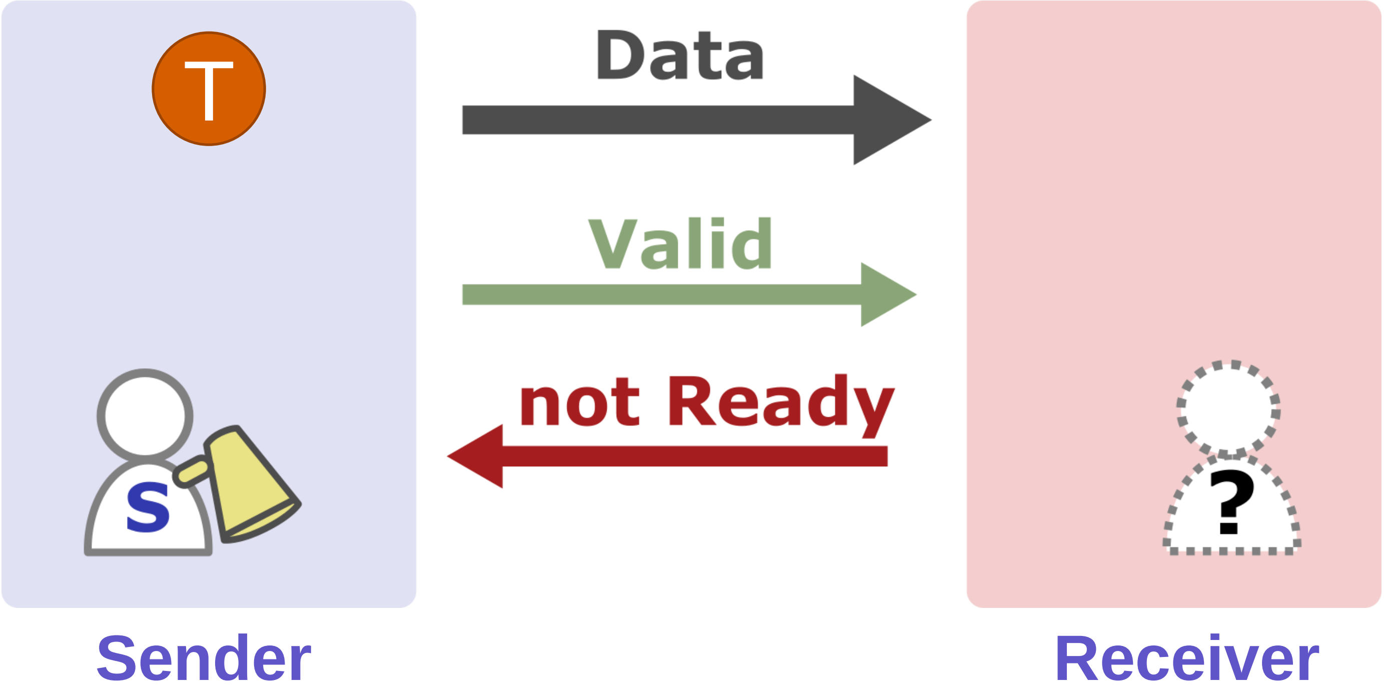
Elastic systems: **computation modules** interconnected by **channels**.
Channels: propagate data, equipped with bidirectional **handshake signals**.



This state is called **idle**

Specification Using Temporal Logic

Elastic systems: **computation modules** interconnected by **channels**.
Channels: propagate data, equipped with bidirectional **handshake signals**.



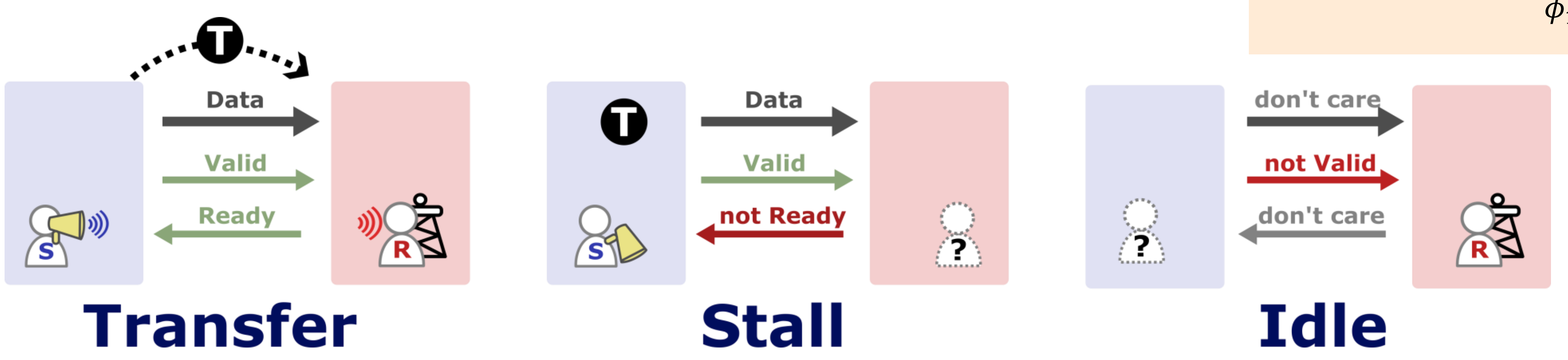
This state is called **stall**

Specification Using Temporal Logic

Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ NeXt ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Elastic systems: **computation modules** interconnected by
Channels: propagate data, equipped with bidirectional **han**



Your turn! Please describe the following properties using CTL formulas:

(a) Liveness: each request (sender asserts a valid) in the channel should eventually be acknowledged (receiver asserts ready).
(b) Fairness: the receiver ready signal should assert infinitely often.
(c) Persistency: when the sender asserts its valid signal high, then it should be remained high until its respective ready is also high.

For each of the problems, can you come up with more than 1 solution?

Specification Using Temporal Logic

Elastic systems: **computation modules** interconnected by
Channels: propagate data, equipped with bidirectional **han**

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Your turn! Please describe the following properties using CTL formulas:

(a) Liveness: each request (sender asserts a valid) in the channel should eventually be acknowledged (receiver asserts ready).

AG (valid → **AF** ready).

(b) Fairness: the receiver ready signal should assert infinitely often.

AG AF (ready).

(c) Persistency: when the sender asserts its valid signal high, then it should be remained high until its respective ready is also high.

AG ((valid ∧ ¬ready) → **AX** valid).

Specification Using Temporal Logic

Elastic systems: **computation modules** interconnected by **channels**.
Channels: propagate data, equipped with bidirectional **handshake signals**.

SELF: Specification and design of synchronous elastic circuits

Jordi Cortadella
Universitat Politècnica de Catalunya
Barcelona, Spain

Mike Kishinevsky
Strategic CAD Lab, Intel Corp.
Hillsboro, OR, USA

Bill Grundmann
Strategic CAD Lab, Intel Corp.
Hillsboro, OR, USA

Dynamically Scheduled High-level Synthesis

Lana Josipović, Radhika Ghosal, and Paolo Ienne
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
CH–1015 Lausanne, Switzerland

If you are interested in the ongoing research on this topic…

Recap: CTL Formula vs the Set of States That Satisfy the CTL Formula

Some important concepts to clarify, here is an example (a is a property that a state can take):

- **AG** a is a **CTL formula**
- ⟦**AG** a⟧ is the **set of states** that satisfy this formula
- We say a state machine TS satisfies the formula **AG** a if the set of initial states of TS is a subset of ⟦**AG** a⟧.

Model Checking CTL Specifications (I)

Do they make a difference?

Determine **the set of states** where the formula holds:

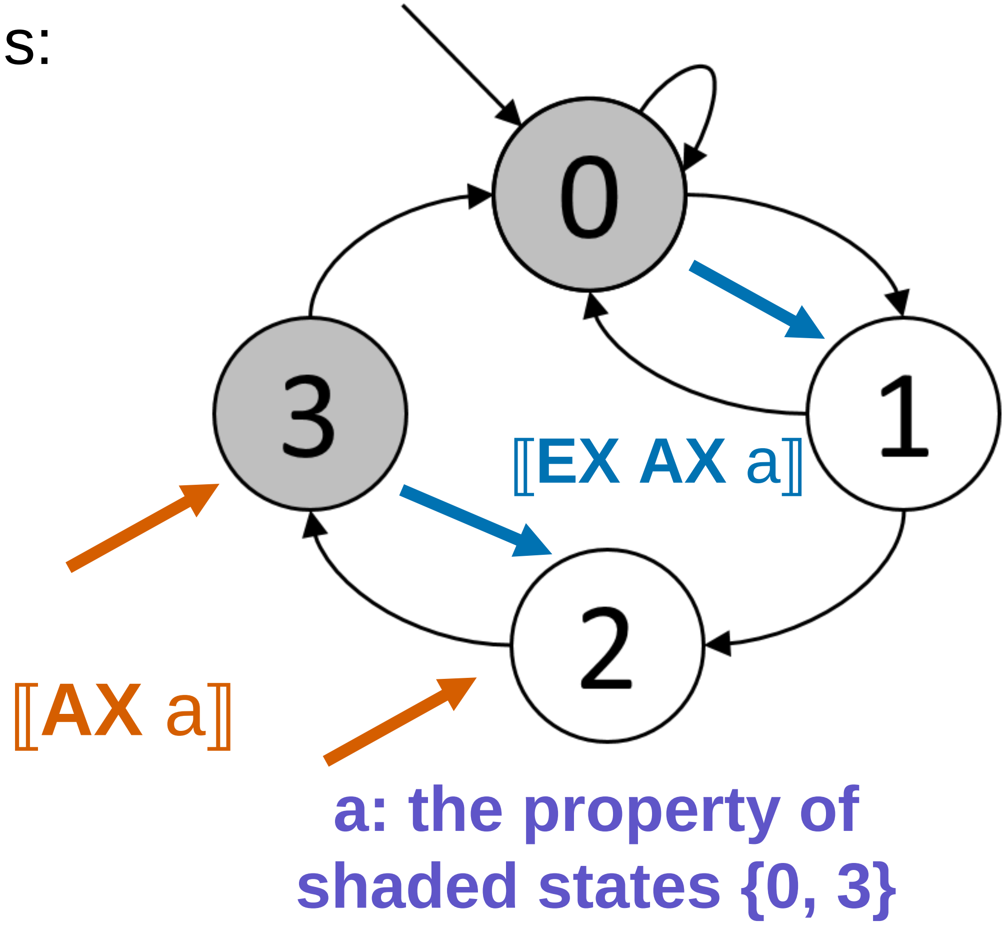
$$Q_c := [\![\mathbf{EX}\,\mathbf{AX}\,a]\!]$$

Step 1: find ⟦**AX** a⟧, the set of states where **AX** a is true

⟦**AX** a⟧ = {2, 3}, we name b as the CTL property AX a.

Step 2: find ⟦**EX** b⟧, the set of states where **EX** b is true

⟦**EX AX** a⟧ = ⟦**EX** b⟧ = {1, 2}.



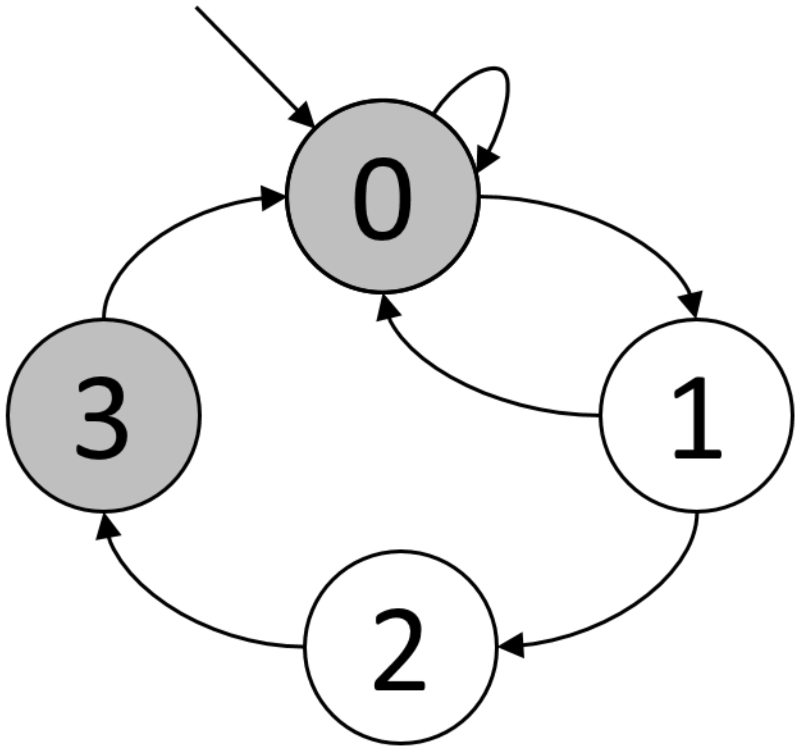
Model Checking CTL Specifications (I)

Your turn! Determine the set of states where the formula holds:

$$Q_a := [\![\mathbf{EF}\, a]\!]$$

$$Q_b := [\![\mathbf{EG}\, a]\!]$$

$$Q_d := [\![\mathbf{EF}\,(a \wedge \mathbf{EX}\, \neg a)]\!]$$



a: the property of shaded states {0, 3}

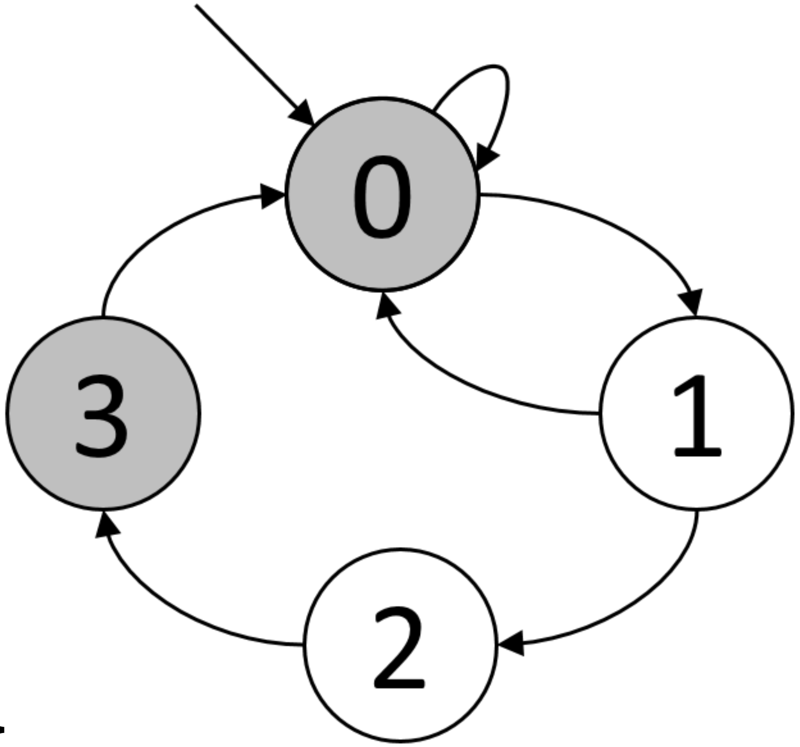
Model Checking CTL Specifications (I)

Your turn! Determine the set of states where the formula holds:

$Q_a := [\![\mathbf{EF}\, a]\!]$ {0, 1, 2, 3}

$Q_b := [\![\mathbf{EG}\, a]\!]$ {0, 3}

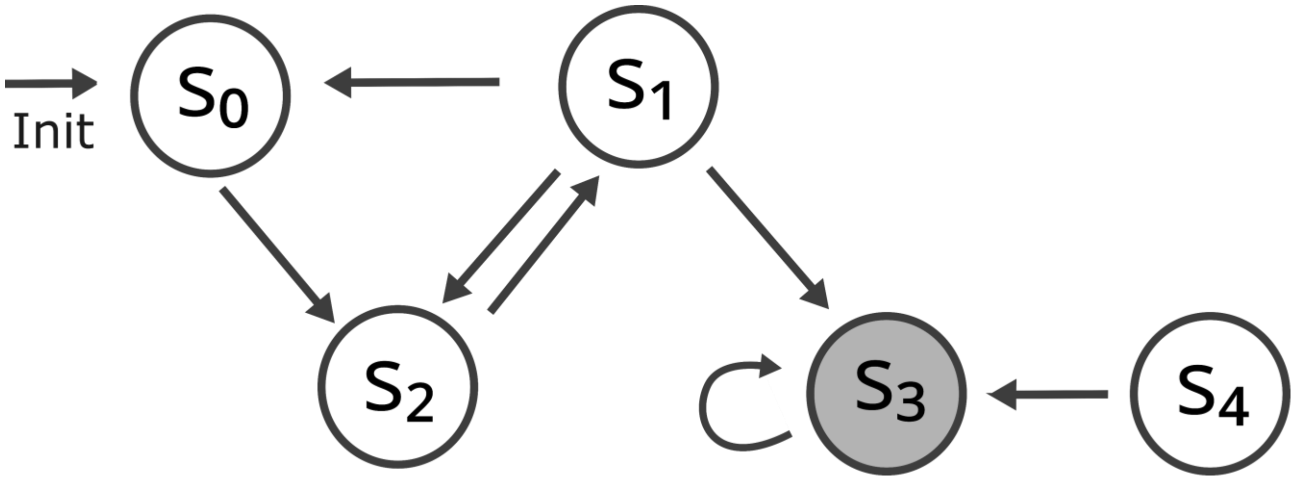
$Q_d := [\![\mathbf{EF}\,(a \wedge \mathbf{EX}\,\neg a)]\!]$ {0, 1, 2, 3}



a: the property of shaded states {0, 3}

Model Checking CTL Specifications (II): Adapted from Exam HS22, Q7

- Consider a is an property that state s3 holds.
- Find ⟦**AF EG** a⟧.
- **Please also show how you find the fixed-point.**



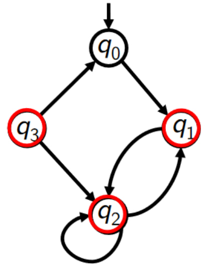
Hint: first, we need to find ⟦EG a⟧:

Step 0:
initial set of states Q0 := {s3}.

Step 1:
What is the predecessor set Pre(Q0)?
What is the set of states after the first iteration: Q1?

Computing CTL formula: EG ϕ

- Example for EG ϕ: Compute EG q_2



$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} =$

$Q_0 = [\![q_2]\!] = \{q_2\}$

$Q_1 = \{q_2\} \cap \mathrm{Pre}(\{q_2\}, \delta) = \{q_2\}$

$[\![\mathrm{EG} q_2]\!] = Q_2 = \{q_2\}$

How to compute AF?

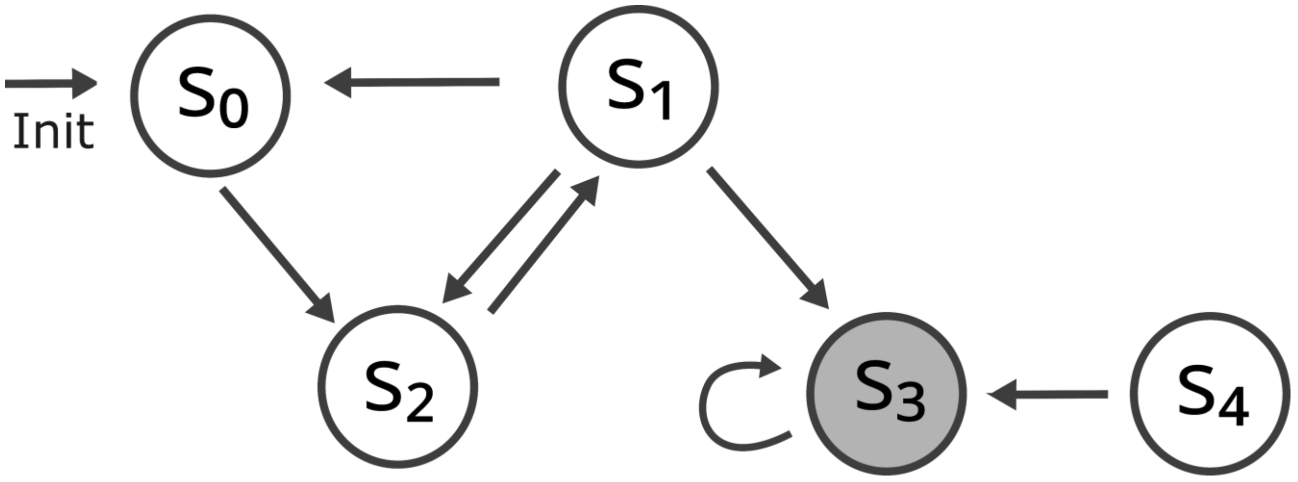
Compute other CTL expressions as:

$\mathrm{AF}\phi \equiv \neg\mathrm{EG}(\neg\phi) \quad \mathrm{AG}\phi \equiv \neg\mathrm{EF}(\neg\phi) \quad \mathrm{AX}\phi \equiv \neg\mathrm{EX}(\neg\phi)$

Your turn! Please complete the rest!

Model Checking CTL Specifications (II): Adapted from Exam HS22, Q7

- Consider a is an property that state s3 holds.
- Find ⟦**AF EG** a⟧.
- **Please also show how you find the fixed-point.**



Step 0:
initial set of states Q0 := {s3}.

Step 1:
Predecessor set Pre(Q0) := {s1, s3, s4}
First iteration: Q1 := Pre(Q0) ∩ Q0 = {s3}

Q0 == Q1: we found a fixed-point!

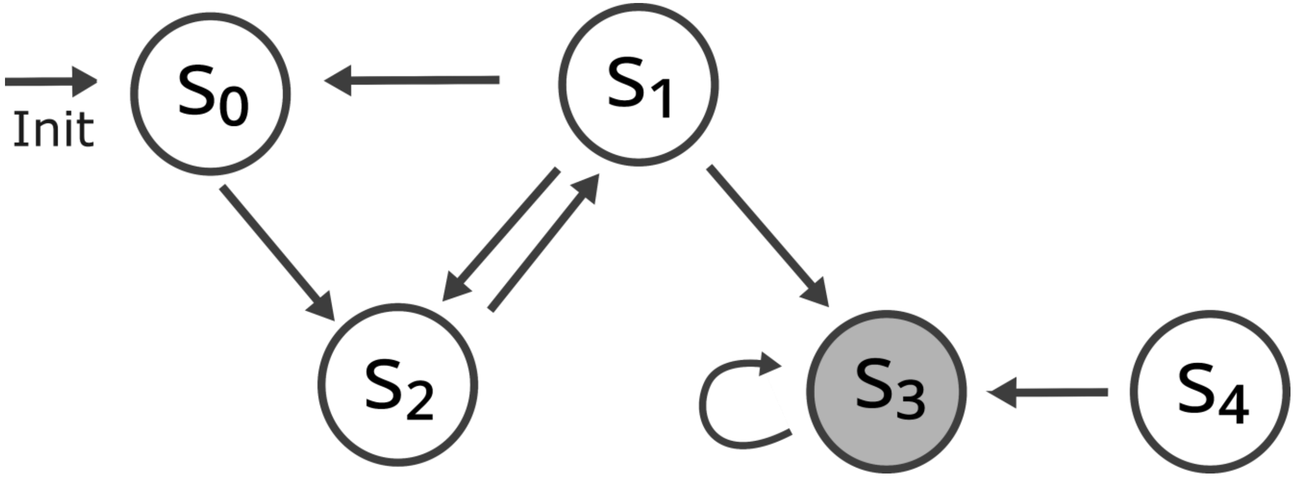
Step 2:
We say ⟦**EG** a⟧ = {s3}; we label all states that satisfy ⟦**EG** a⟧ with b.
We need to find ⟦**AF** b⟧.

Step 3:
⟦**AF** b⟧ is ⟦¬**EG** ¬b⟧.

How to compute AF?

Compute other CTL expressions as:
$\mathrm{AF}\phi \equiv \neg\mathrm{EG}(\neg\phi) \quad \mathrm{AG}\phi \equiv \neg\mathrm{EF}(\neg\phi) \quad \mathrm{AX}\phi \equiv \neg\mathrm{EX}(\neg\phi)$

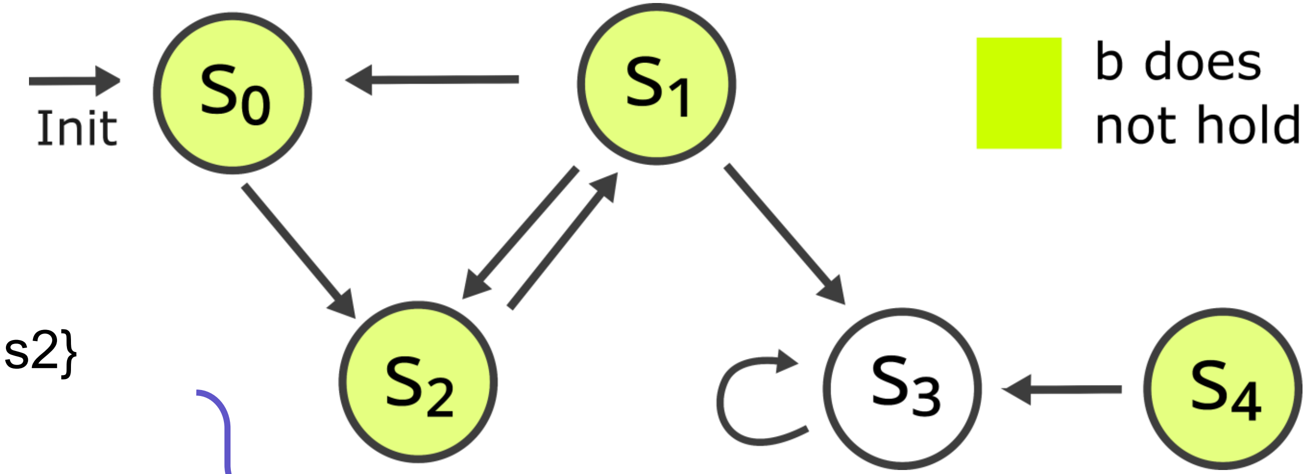
- Consider a is an property that state s3 holds.
- Find ⟦**AF EG** a⟧.
- **Please also show how you find the fixed-point.**



⟦b⟧ = ⟦**EG** a⟧ = {s3}

Step 3:
⟦**AF** b⟧ is {s0, s1, s2, s3, s4} \ ⟦**EG** ¬b⟧.

Step 4:
initial set of states Q0 := {s0, s1, s2, s4}.



Step 5:
Predecessor set Pre(Q0) := {s0, s1, s2}
First iteration: Q1 := Pre(Q0) ∩ Q0 = {s0, s1, s2}

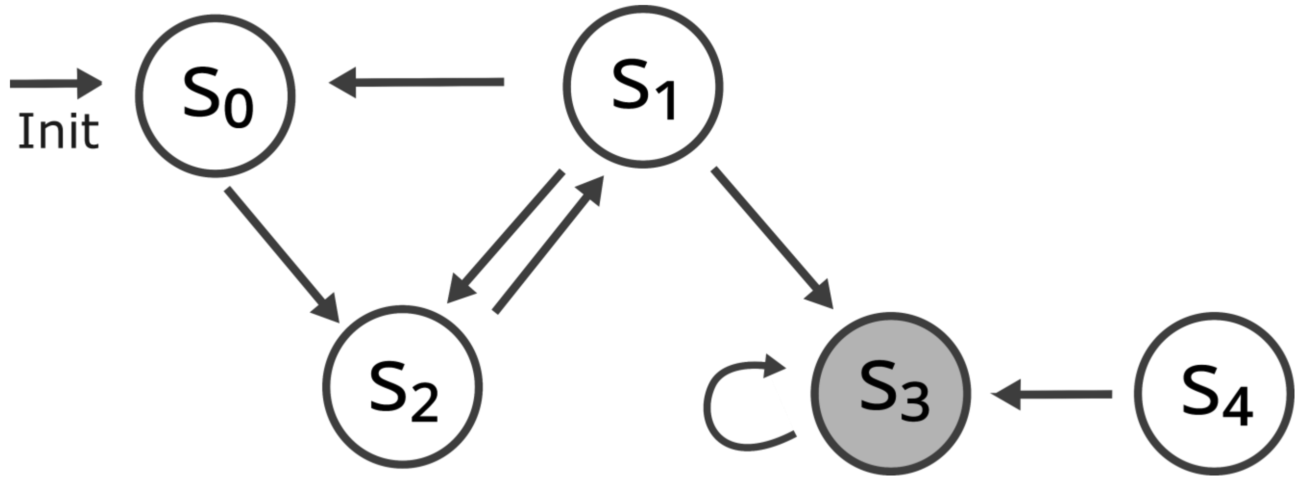
Step 6:
Predecessor set Pre(Q1) := {s0, s1, s2}
Second iteration: Q2 := Pre(Q1) ∩ Q1 = {s0, s1, s2}

Q1 == Q2: we found a fixed-point!

⟦EG ¬b⟧ = {s0, s1, s2}

Model Checking CTL Specifications (II): Adapted from Exam HS22, Q7

- Consider a is an property that state s3 holds.
- Find ⟦**AF EG** a⟧.
- **Please also show how you find the fixed-point.**



⟦b⟧ = ⟦**EG** a⟧ = {s3}

Step 7:
⟦**EG** ¬b⟧ = Q2 = {s0, s1, s2}

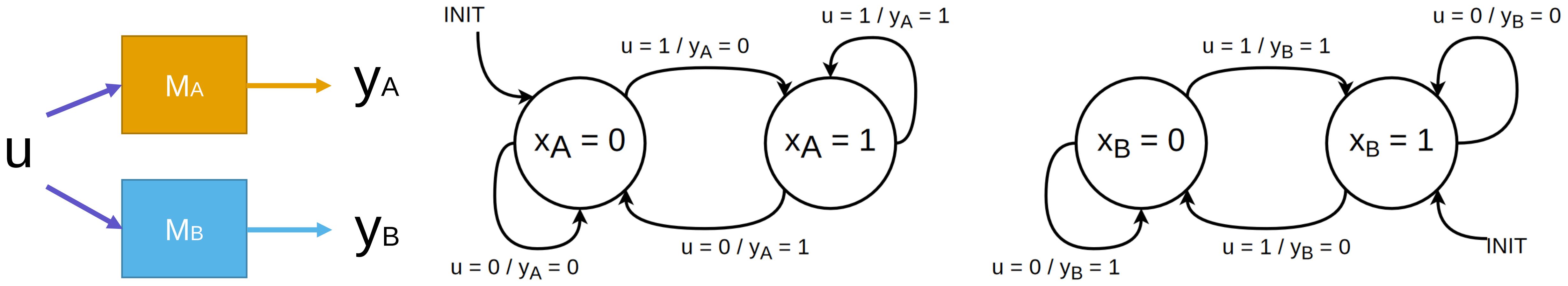
Step 8:
⟦**AF EG** a⟧ = ⟦ ¬ **EG** ¬b⟧ = ⟦*true*⟧ \ ⟦**EG** ¬b⟧ = Q2 = {s3, s4}

- In the exercise sheet there is also a discussion on how to formulate this as an algorithm (i.e., a model checking algorithm).

Now you can implement a model checker that checks arbitrary CTL formula!

Comparison Between Two State Machines

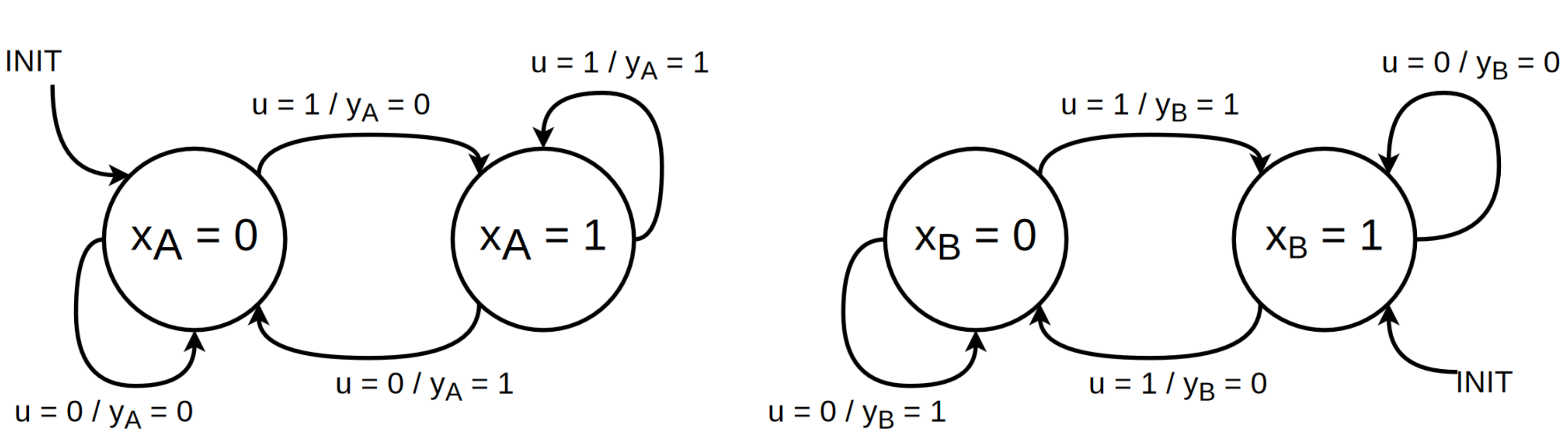
Last time we saw how to compare two **combinational circuits**; today we will see how to check the equivalence of two **state machines** (sequential).



(A) (B)

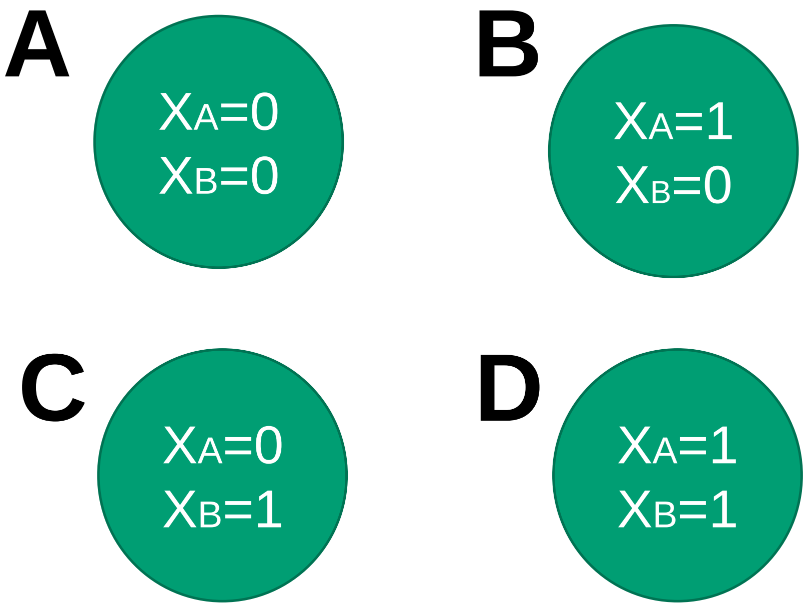
Problem: for arbitrary values of input u, do the two state machines always produce the same value of y?

Comparison Between Two State Machines



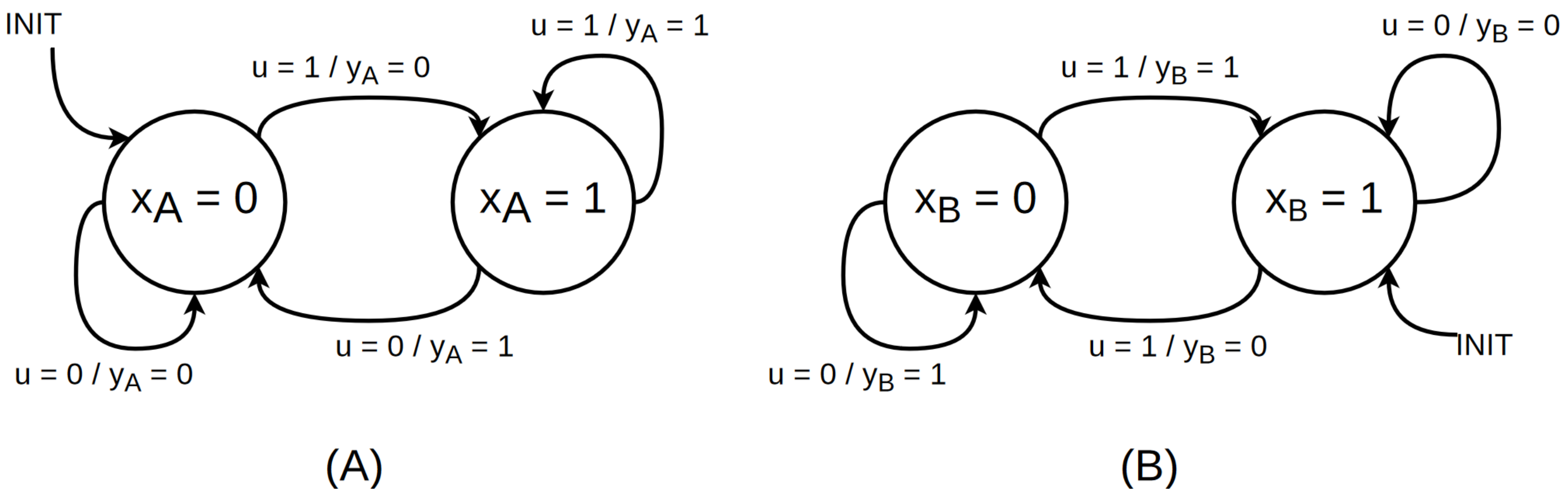
(A) (B)

Idea: compute the joint machine, by enumerating all combinations of states

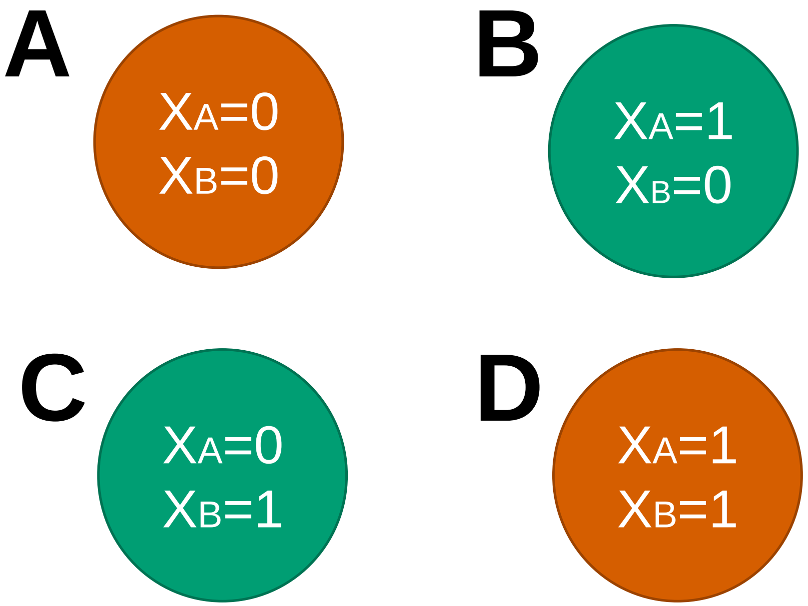


The state space of the joint state machine

Comparison Between Two State Machines

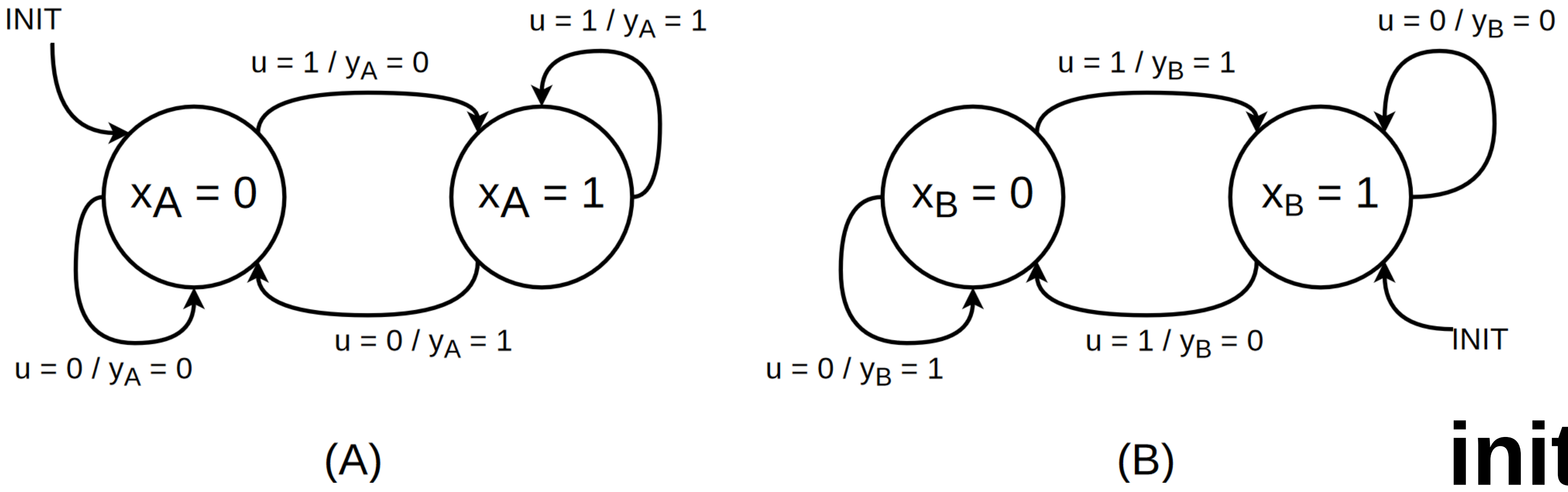


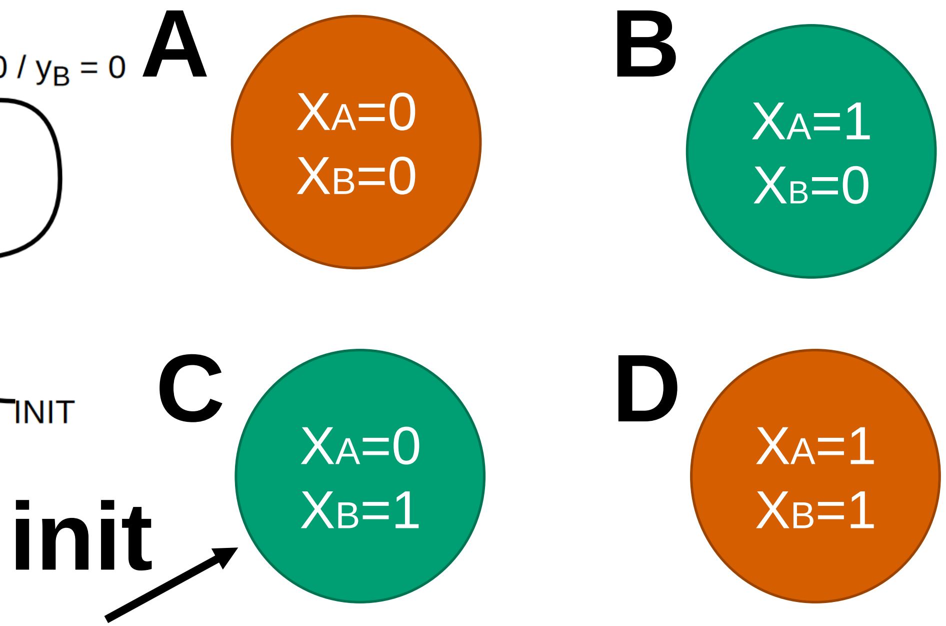
If state A or D are reachable (outputs are different), then two machines are not equivalent



The state space of the joint state machine

Comparison Between Two State Machines

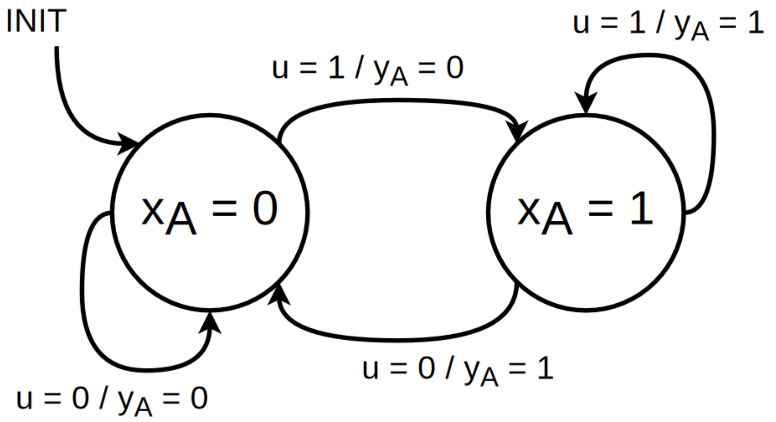




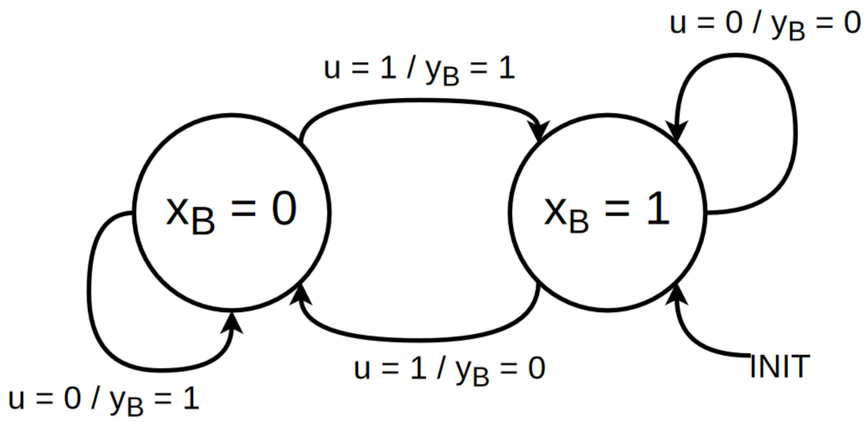
Initial state is C (X_B=0, X_B=1).

The state space of the joint state machine

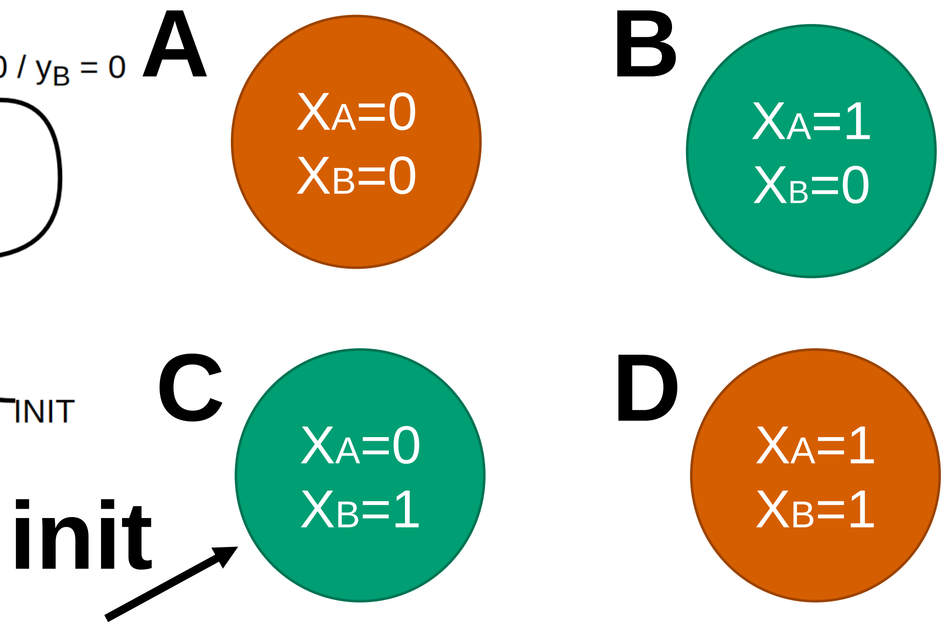
Comparison Between Two State Machines



(A)



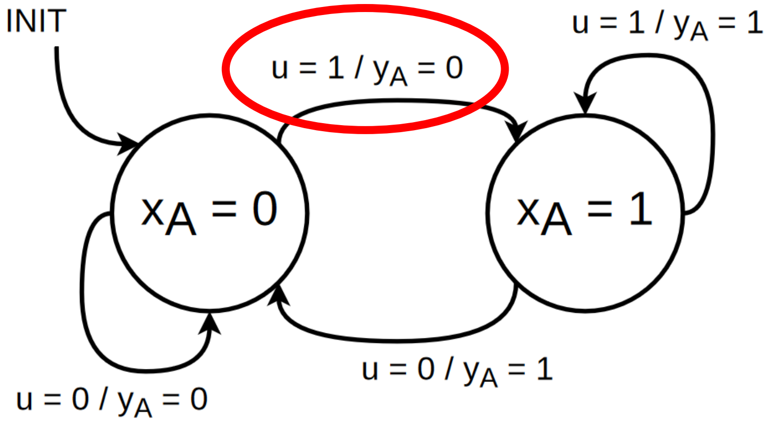
(B)



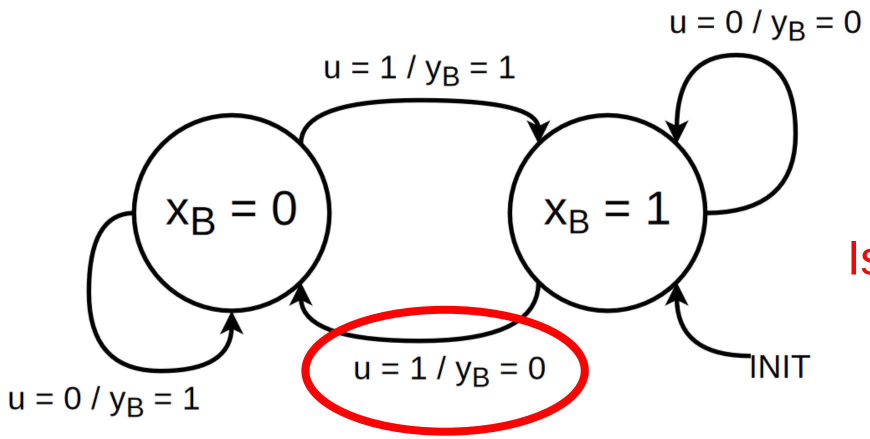
The state space of the joint state machine

- Suppose the translation relations of M_A, M_B, and the joint machine are RA, RB, and RJ.
- A transition for this product machine is denoted as (X_A, X_B, X_A', X_B').
- (X_A, X_B, X_A', X_B') is in the RJ if there exists a value of u such that (X_A, X_A') is in RA and (X_B, X_B') is in RB.

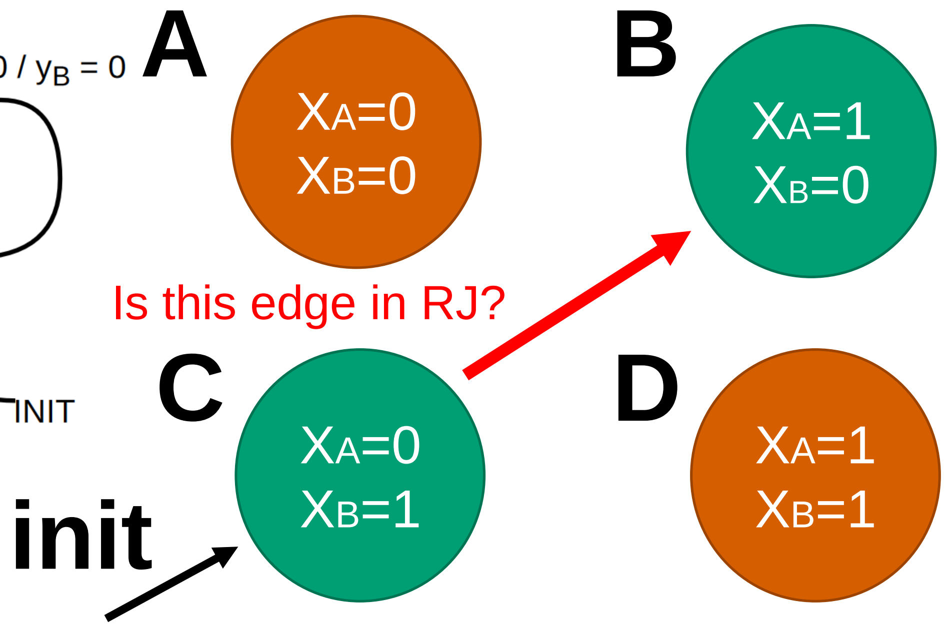
Comparison Between Two State Machines



(A)



(B)



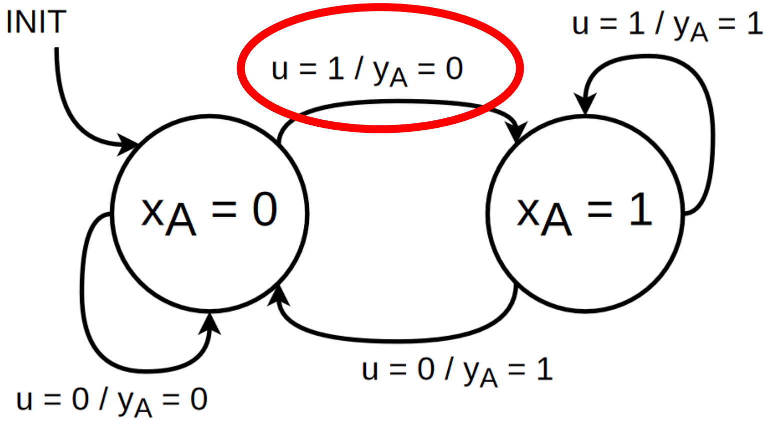
The state space of the joint state machine

- Suppose the translation relations of M_A, M_B, and the joint machine are RA, RB, and RJ.
- A transition for this product machine is denoted as (X_A, X_B, X_A', X_B').
- (X_A, X_B, X_A', X_B') is in the RJ if there exists a value of u such that (X_A, X_A') is in RA and (X_B, X_B') is in RB.

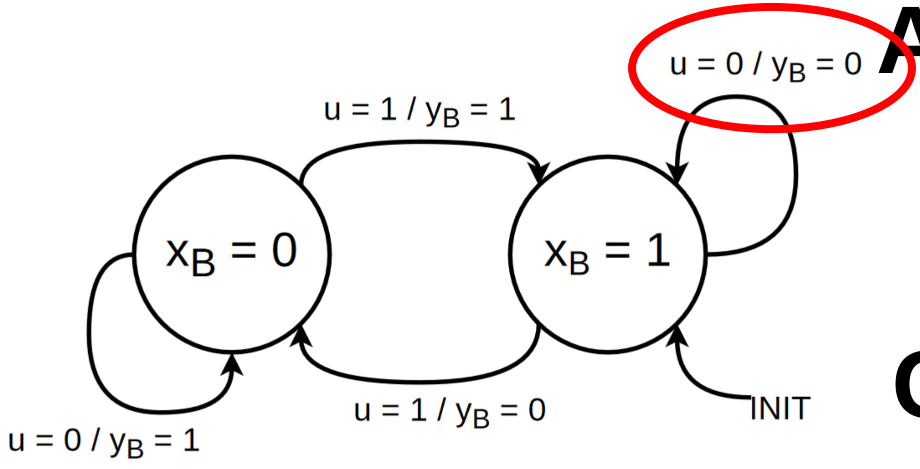
The edge is in RJ (i.e., such u exists):

- when u=1, (XA=0, XA'=1) is in RA; when u=1, (XB=1, XB'=0) is in RB

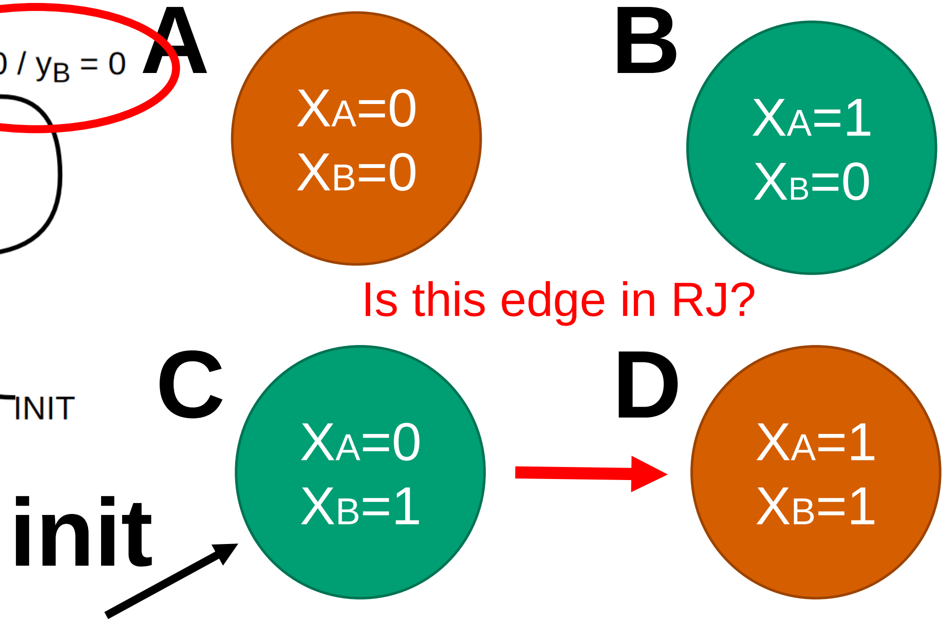
Comparison Between Two State Machines



(A)



(B)



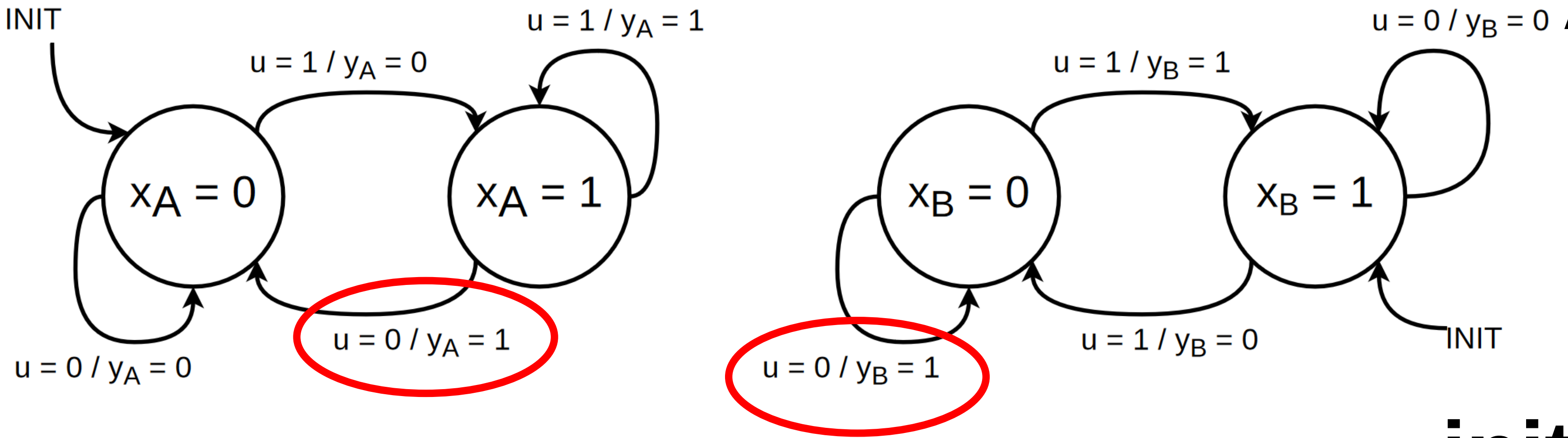
The state space of the joint state machine

- Suppose the translation relations of M_A, M_B, and the joint machine are RA, RB, and RJ.
- A transition for this product machine is denoted as (X_A, X_B, X_A’, X_B’).
- (X_A, X_B, X_A’, X_B’) is in the RJ if there exists a value of u such that (X_A, X_A’) is in RA and (X_B, X_B’) is in RB.

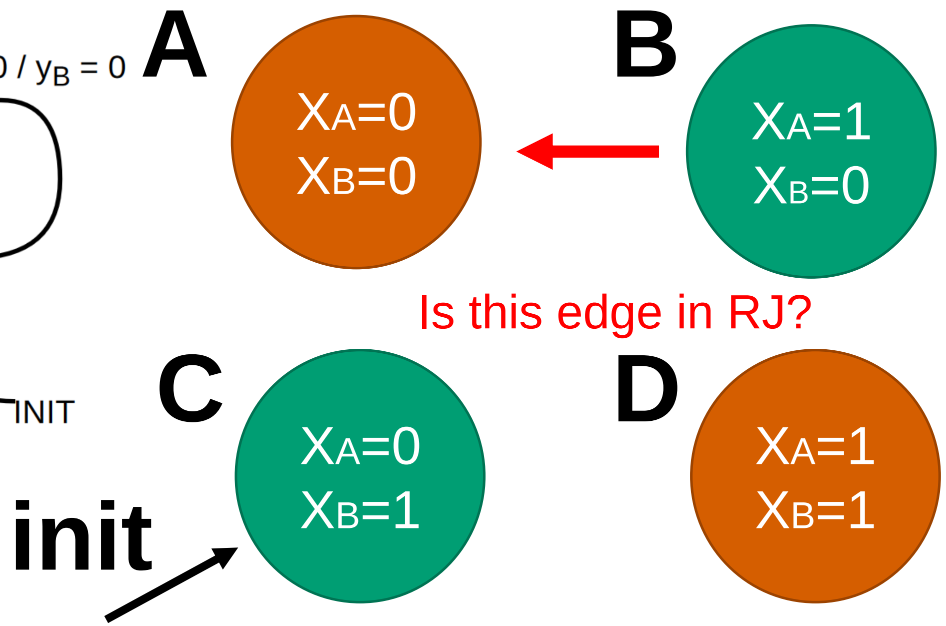
The edge is not in RJ (i.e., such u doesn’t exist):

- when u=1, (XA=0, XA’=1) is in RA; when u=0, (XB=1, XB’=1) is in RB
- This is unsatisfiable!

Comparison Between Two State Machines



(A) (B)



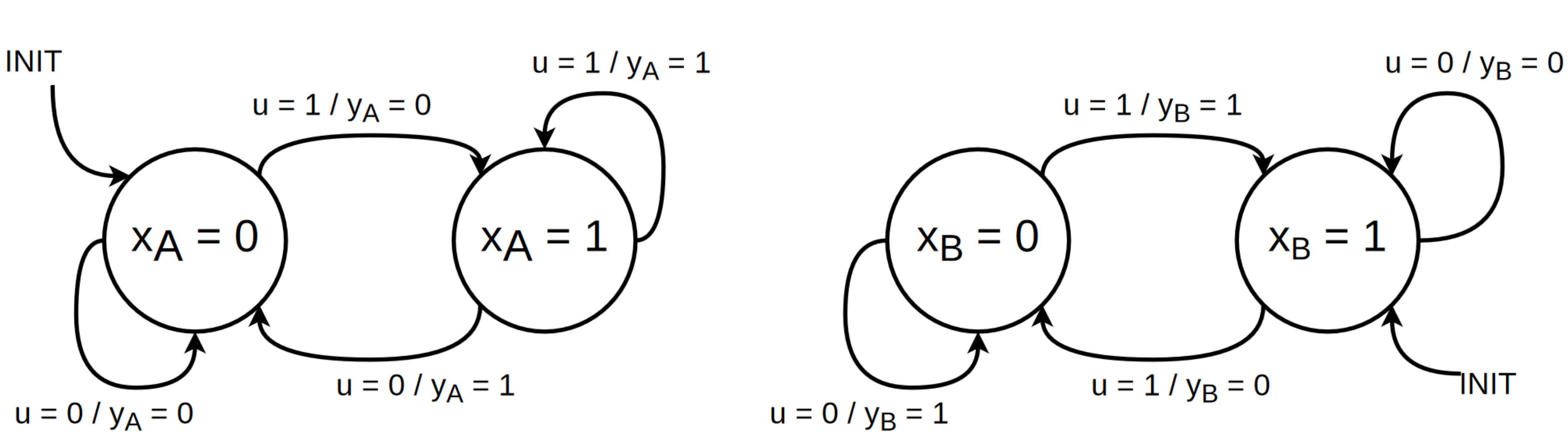
The state space of the joint state machine

- Suppose the translation relations of M_A, M_B, and the joint machine are RA, RB, and RJ.
- A transition for this product machine is denoted as (X_A, X_B, X_A', X_B').
- (X_A, X_B, X_A', X_B') is in the RJ if there exists a value of u such that (X_A, X_A') is in RA and (X_B, X_B') is in RB.

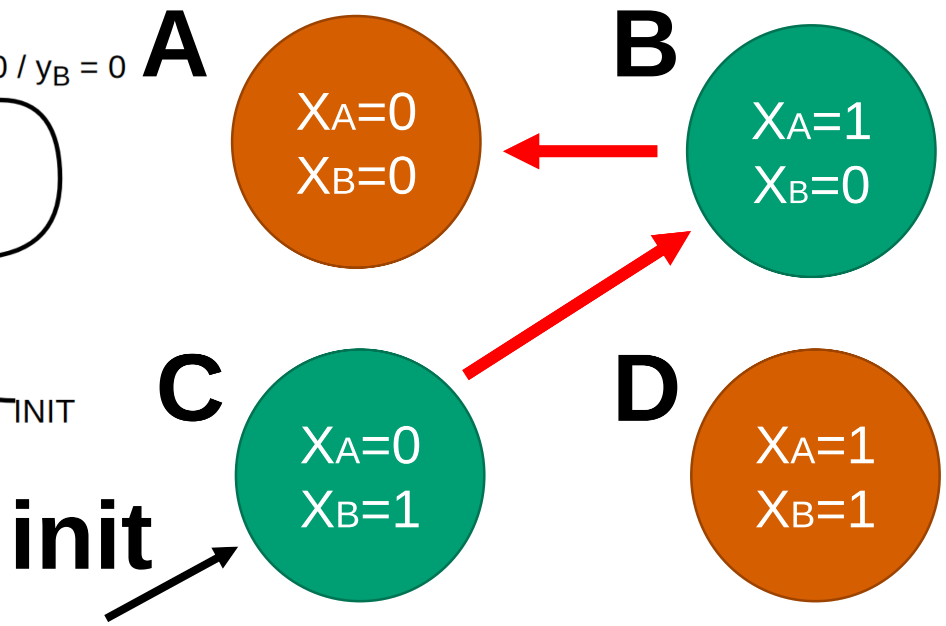
The edge is in RJ:

- when u=0, (XA=1, XA'=0) is in RA; when u=0, (XB=0, XB'=0) is in RB

Comparison Between Two State Machines



(A) (B)



The state space of the joint state machine

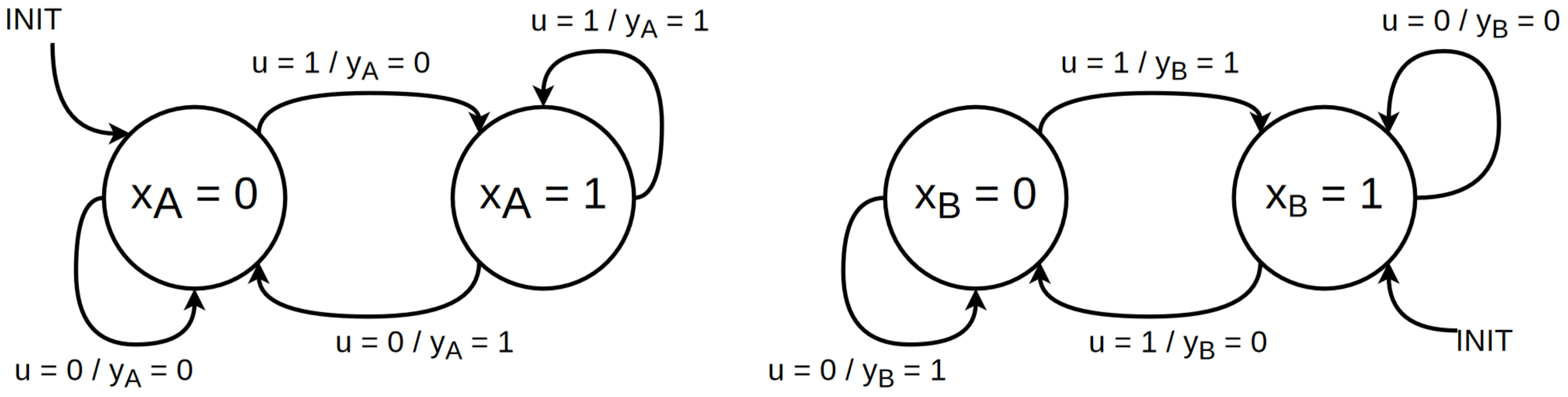
The state machines are not equivalent! We have found a trace that leads us to state A

Comparison Between Two State Machines

$Q_0 = \{q_0\}$

$Q_{i+1} = Q_i \cup Suc(Q_i, \delta)$ until $Q_{i+1} = Q_i$

$\psi_{Q_{i+1}}(q') = \underbrace{\psi_{Q_i}(q')}_{q' \text{ is already in } Q_i} + \underbrace{(\exists q : \psi_{Q_i}(q) \cdot \psi_\delta(q, q'))}_{\text{There is a state } q \text{ in } Q_i \text{ with transition } q \to q'}$



(A) (B)

Your turn! Determine the followings:

(a) Determine the characteristic function $\psi_A(x_A, x'_A, u)$ and $\psi_B(x_B, x'_B, u)$ of the transition relation for the two state machines A and B.

(b) Determine the characteristic function $\psi_f(x_A, x'_A, x_B, x'_B)$ of the transition relation for the joint state machines. Note: $\psi_f(x_A, x'_A, x_B, x'_B) \coloneqq \big(\exists u\colon \psi_A(x_A, x'_A, u) \cdot \psi_B(x_B, x'_B, u)\big)$.

(c) Determine the characteristic function $\psi_X(x_A, x_B)$ of the set of reachable states of the product state machines.