

Controller Synthesis and Implementability Issues

J.-F. Raskin
Université Libre de Bruxelles

MOVEP'06
Bordeaux, June 23, 2006

Content

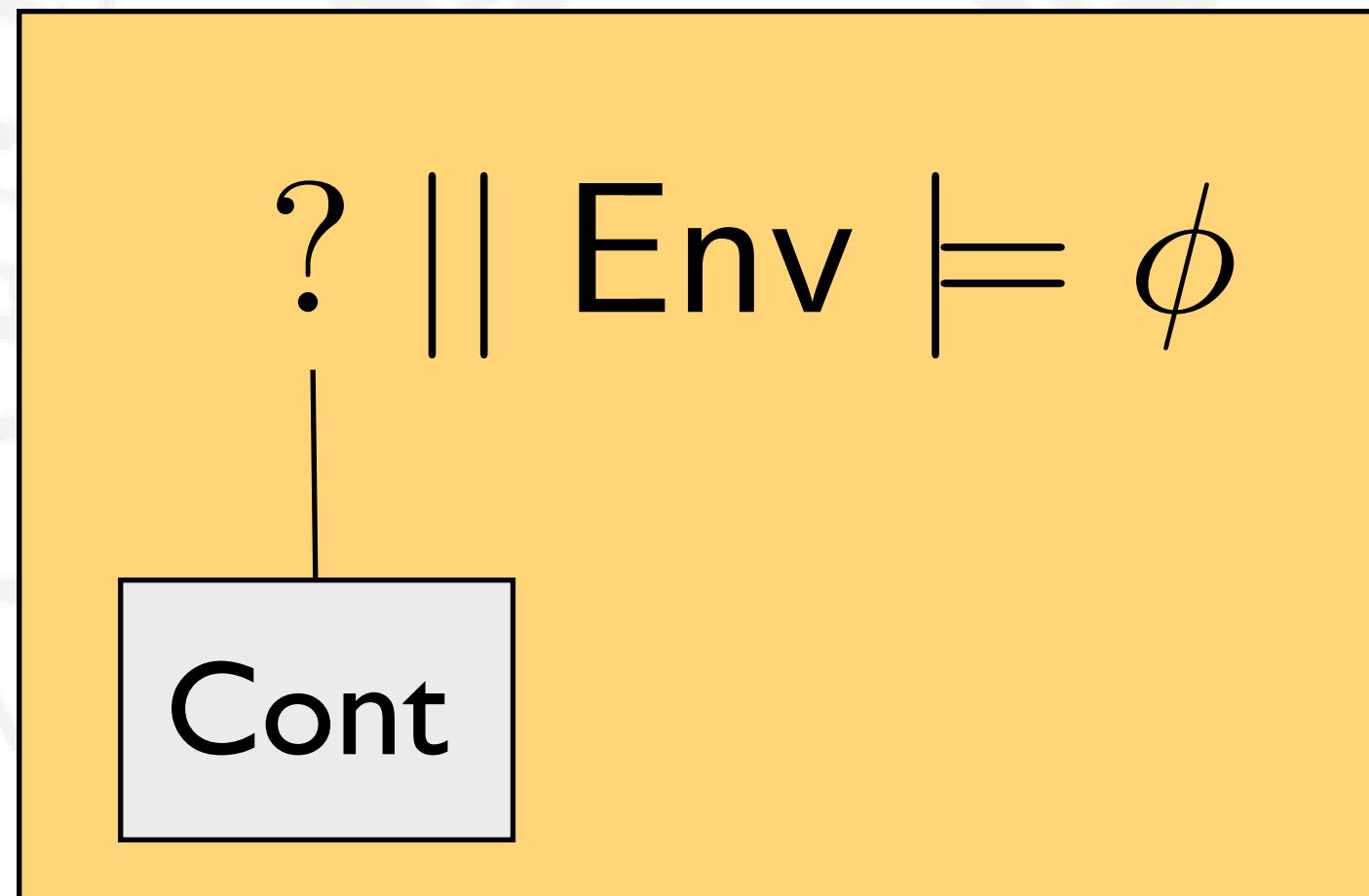
- Controller synthesis problem
- Two-player game structures
- Safety games (of perfect information)
- Imperfect information: motivations
- The lattice of antichains
- CPre over the lattice of antichains
- Application to discrete time control of RHA
- Application to the universality problem of NFA
- Conclusion & perspectives

The synthesis problem

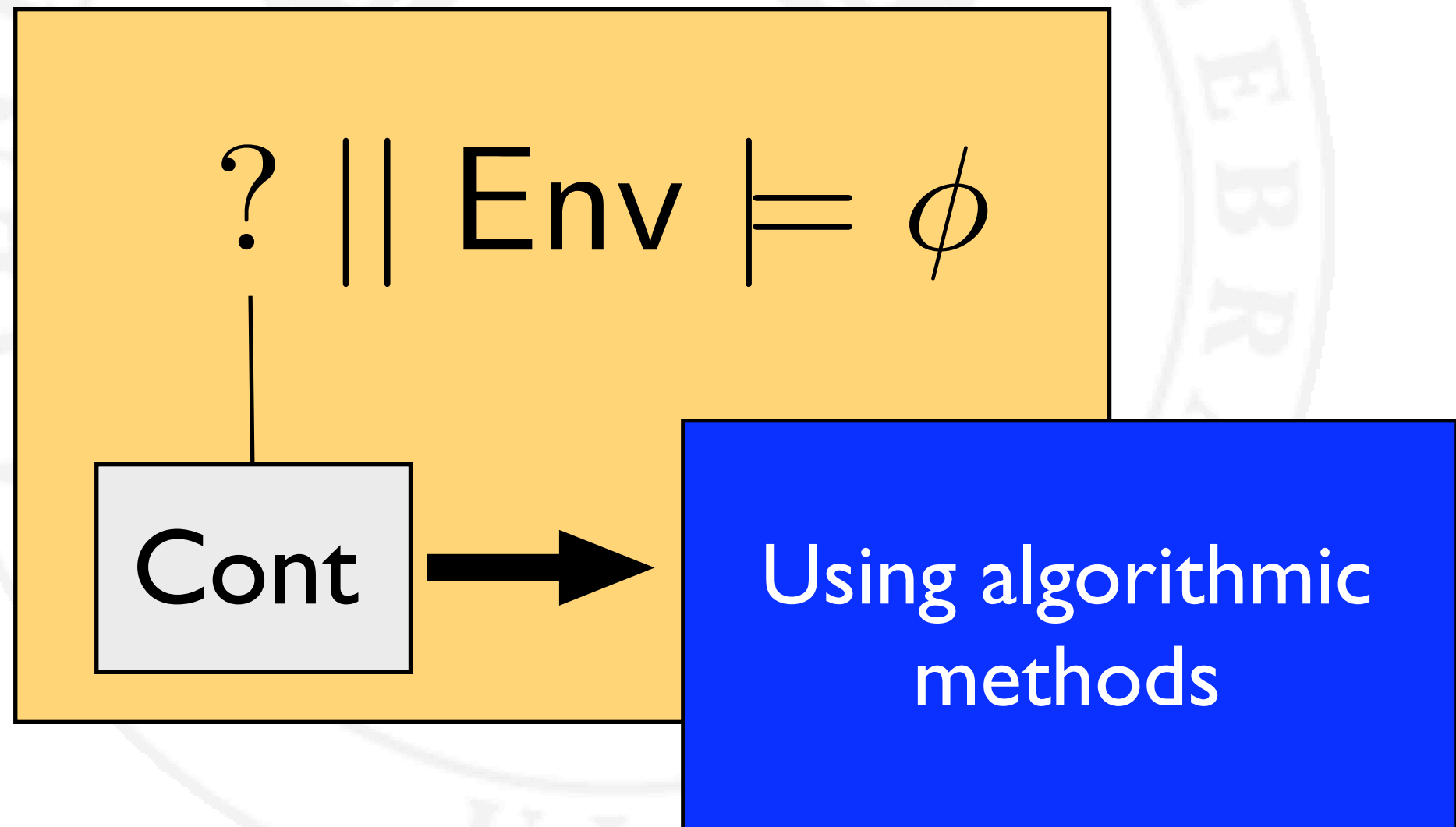
The synthesis problem

$$? \parallel Env \models \phi$$

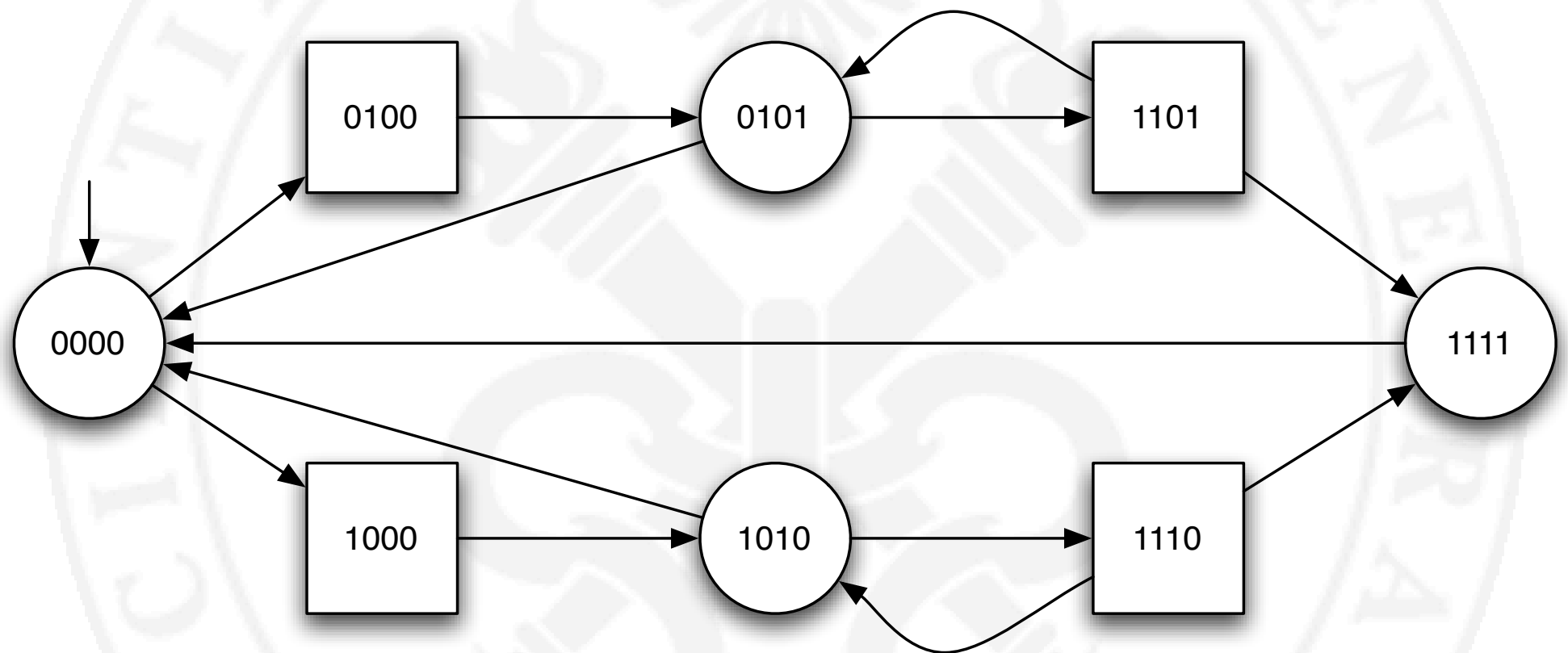
The synthesis problem

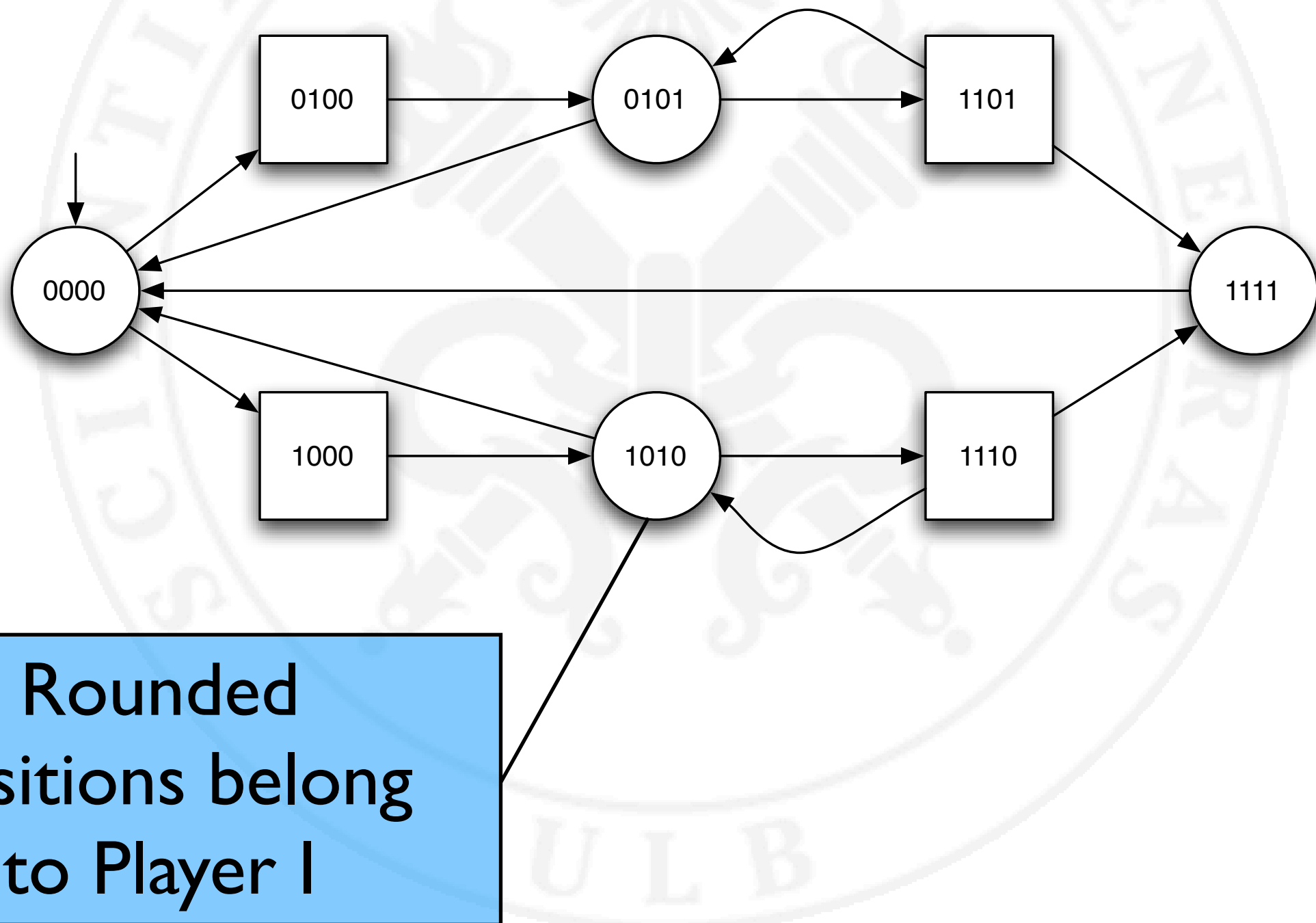


The synthesis problem

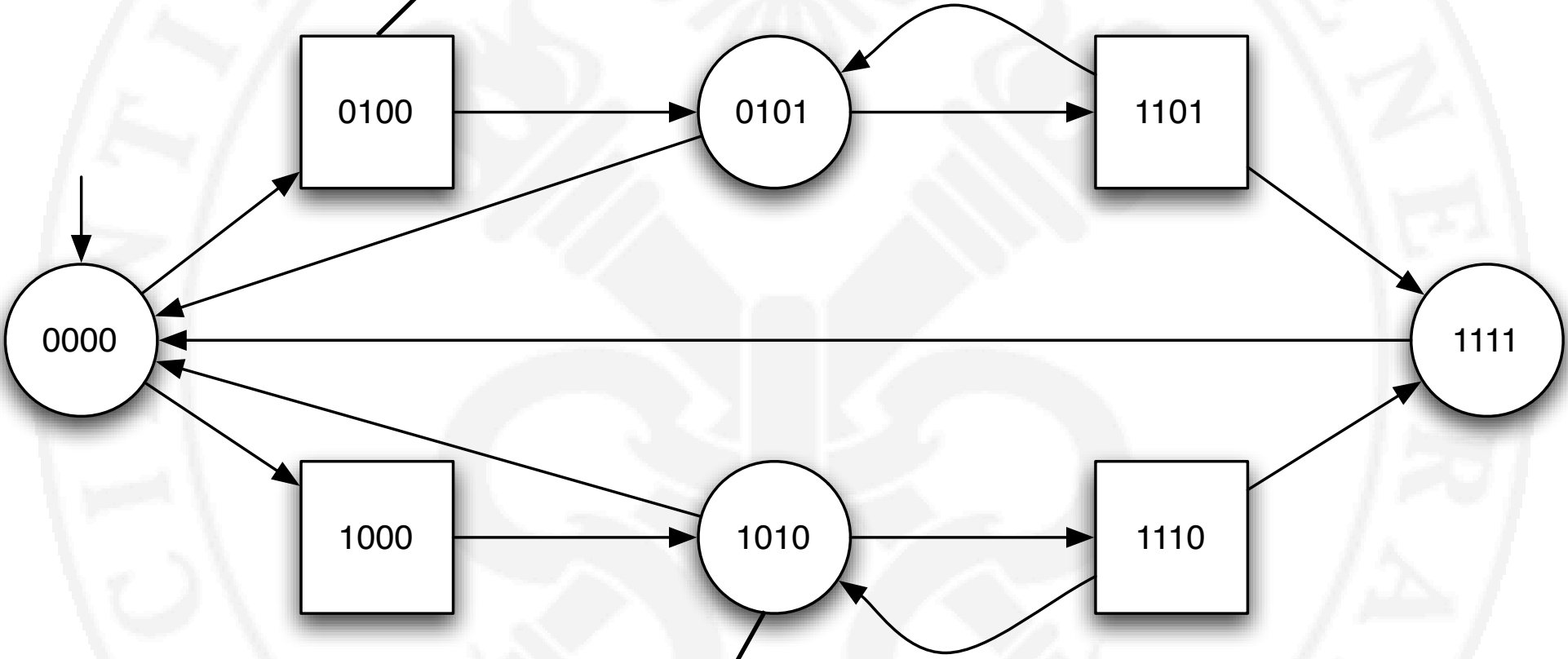


Two-player game structures



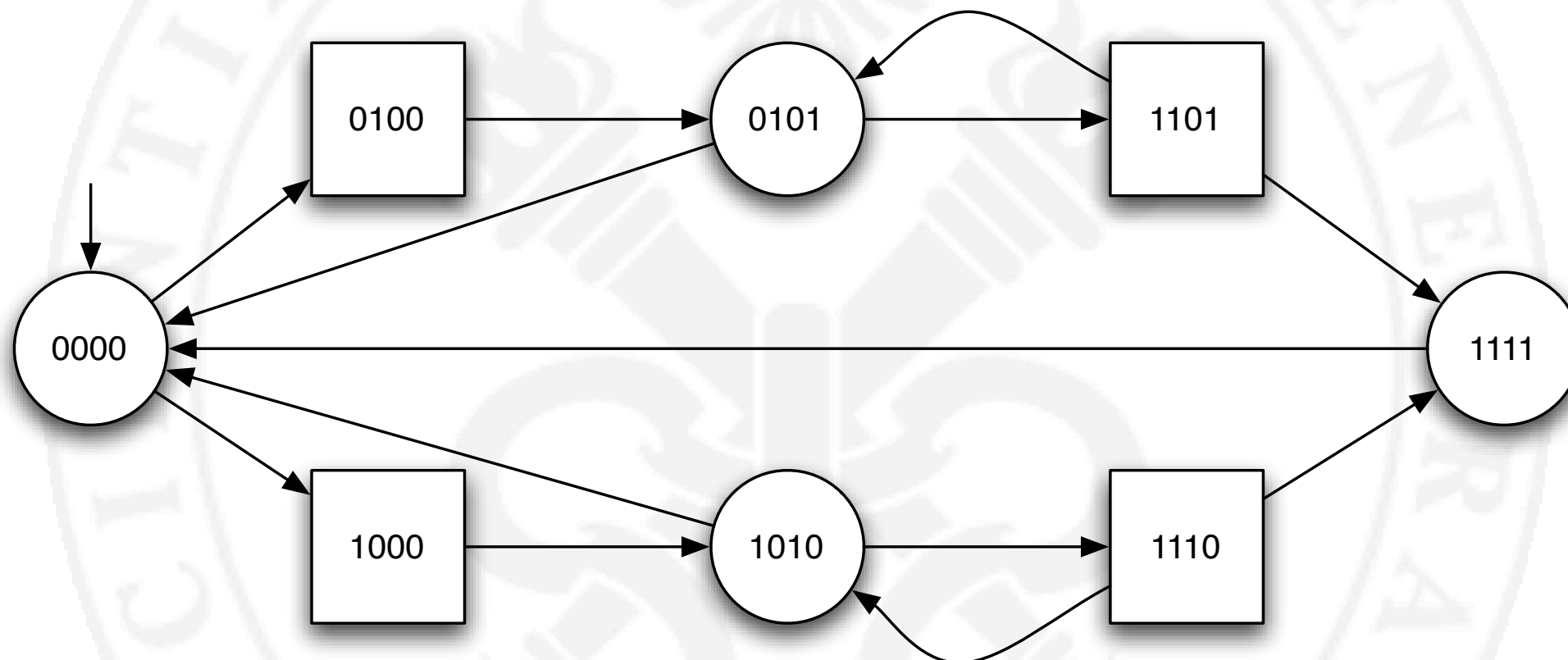


Square positions belong to Player 2

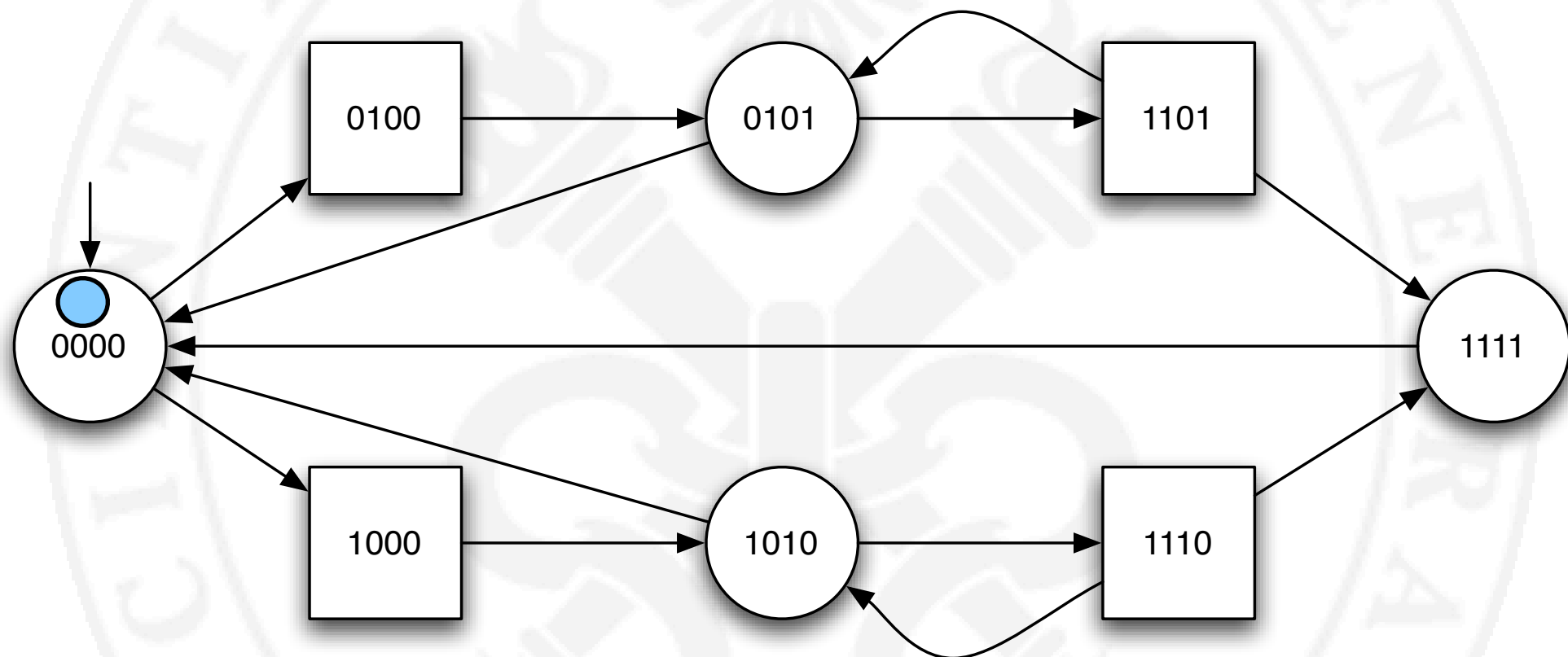


Rounded positions belong to Player 1

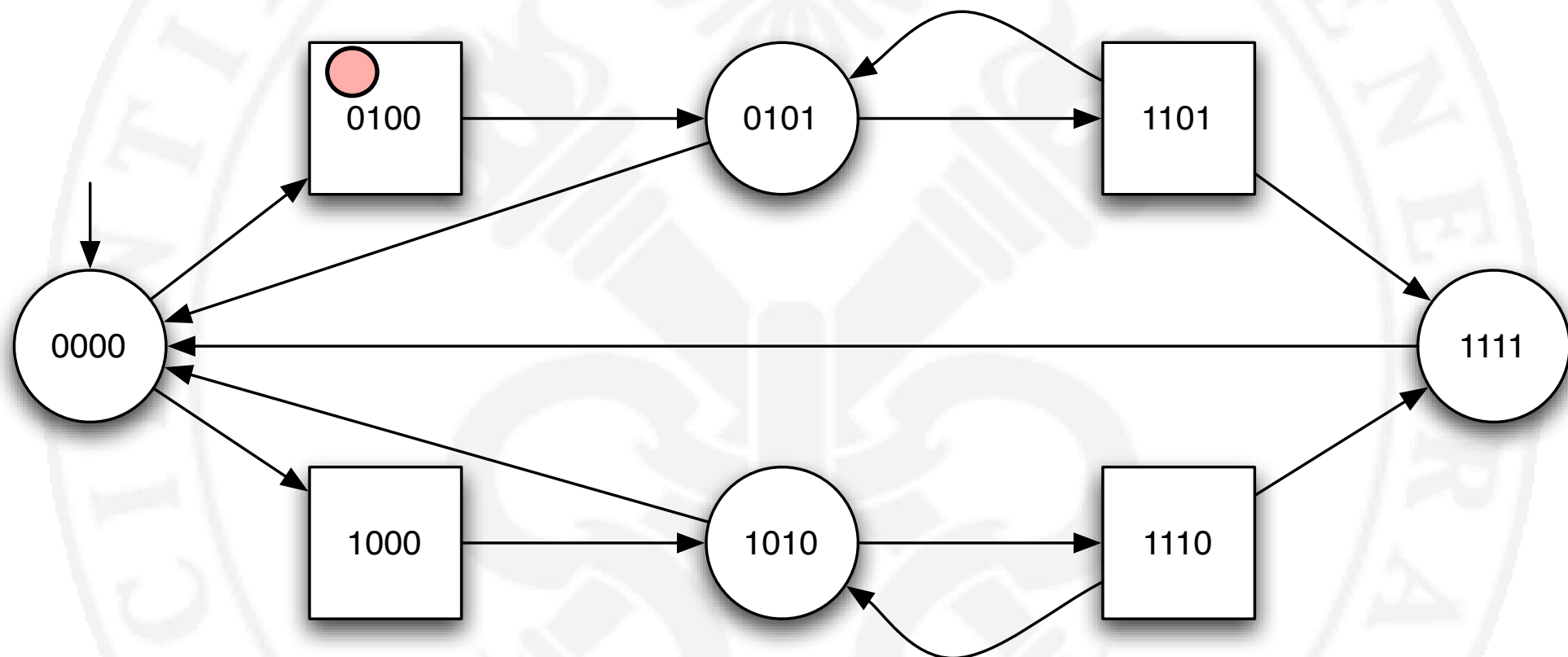
Rounded positions belong to Player 1
 Square positions belong to Player 2



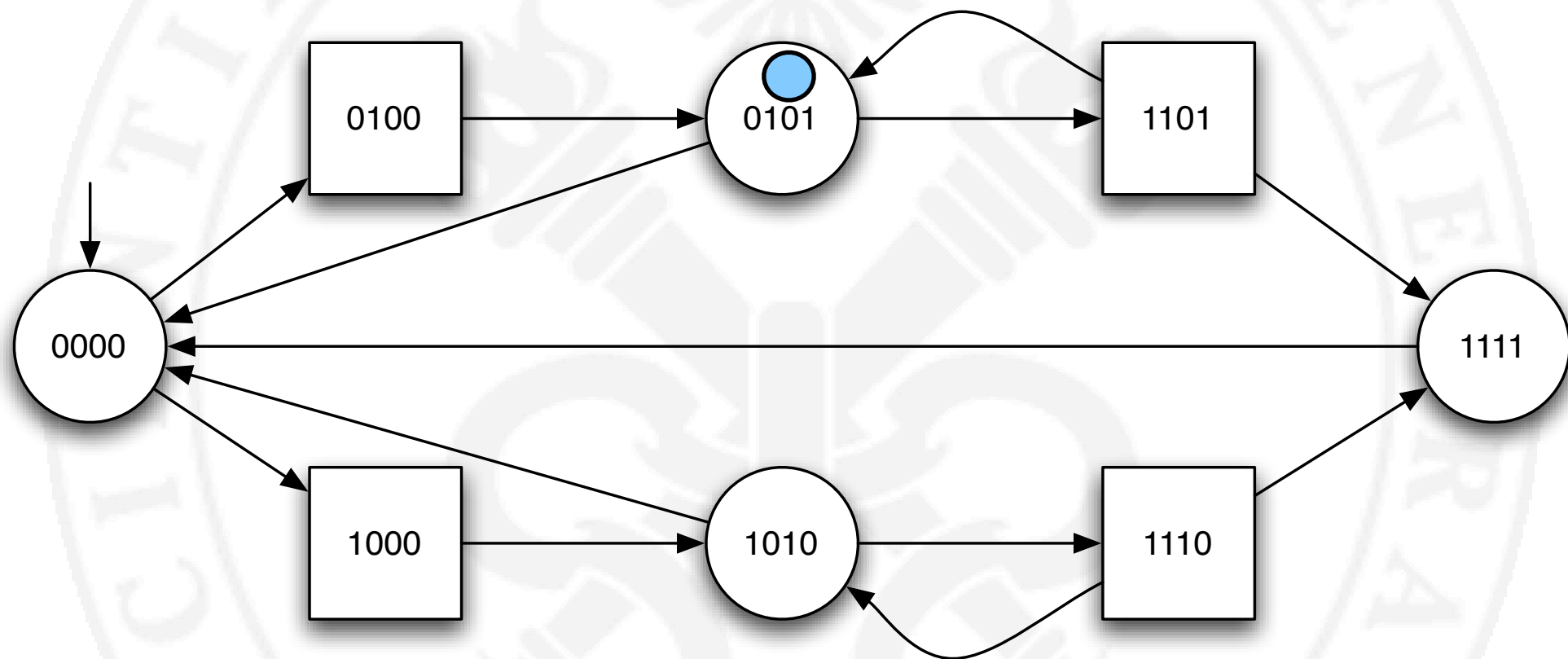
A game is played as follows: in each **round**, the game is in a **position**, if the game is in a rounded position, Player 1 resolves the **choice** for the next state, if the game is in a square position, Player 2 resolves the choice. The game is played for an **infinite number of rounds**.



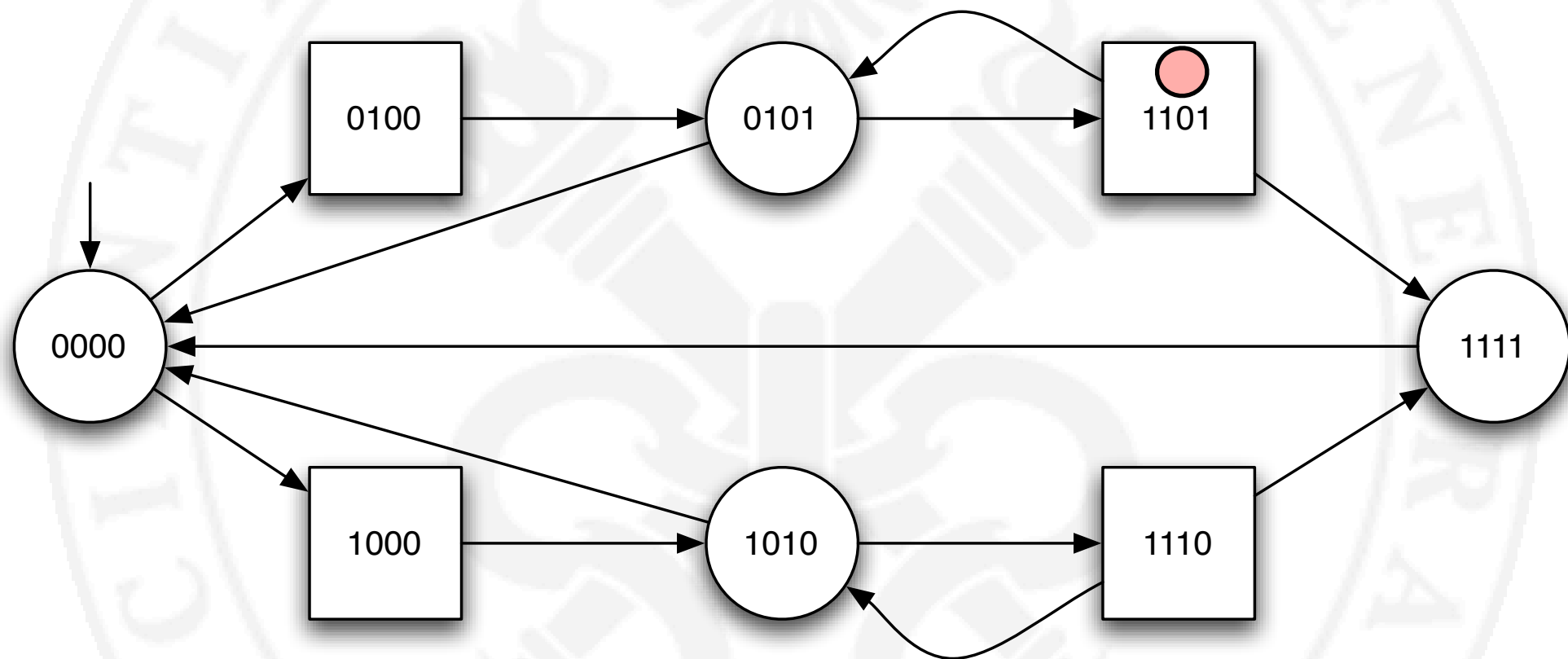
Play : 0000



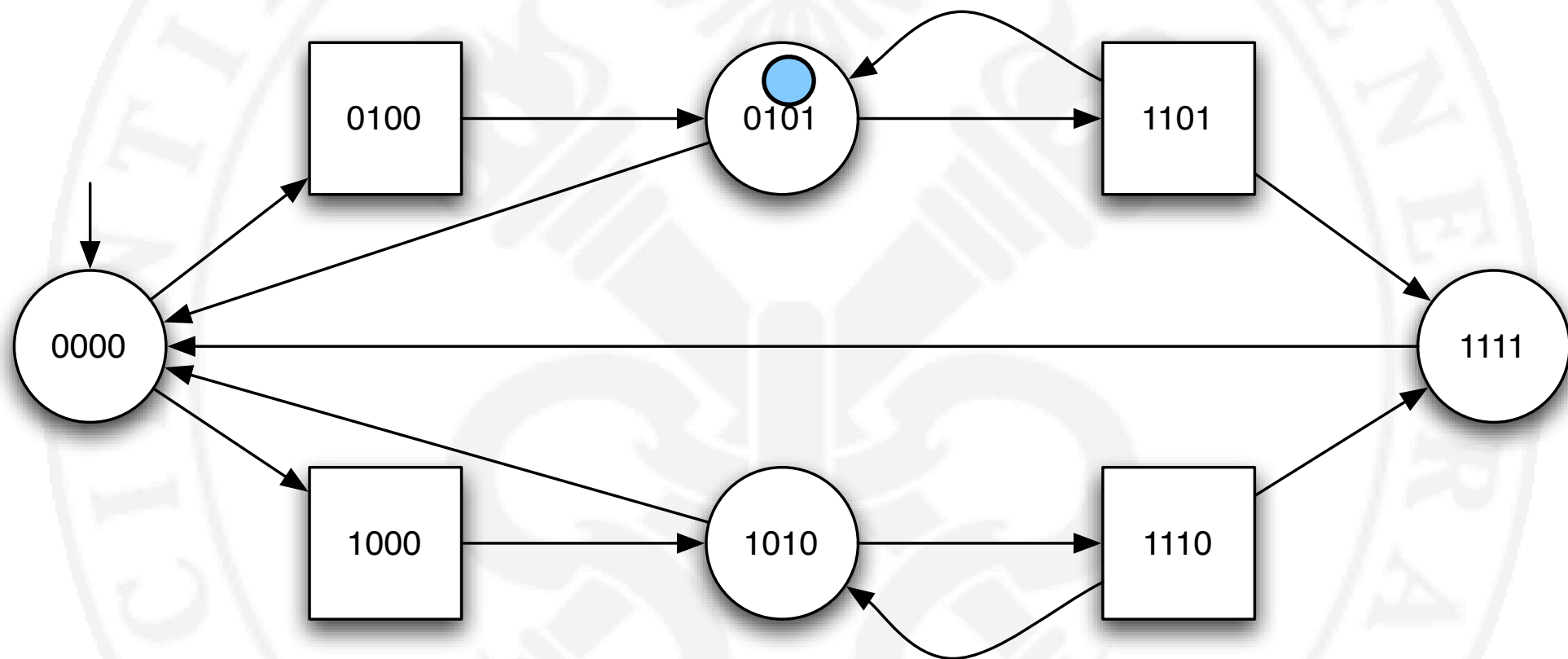
Play : 0000 0100



Play : 0000 0100 0101



Play : 0000 0100 0101 1101



Play : 0000 0100 0101 1101 ...

Two-player Game Structure

A **two-player game structure** is a tuple

$G = \langle Q_1, Q_2, \iota, \delta \rangle$ where:

Q_1 and Q_2 are two (finite and) disjoint sets
of **positions**

$\iota \in Q_1 \cup Q_2$ is the **initial** position of the game

$\delta \subseteq (Q_1 \cup Q_2) \times (Q_1 \cup Q_2)$ is the **transition
relation** of the game

We assume that $\forall q \in Q_1 \cup Q_2 : \exists q' \in Q_1 \cup Q_2 : \delta(q, q')$

Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$,

$w = q_0 q_1 \dots q_n \dots$ is a **play** in G if

Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$,

$w = q_0 q_1 \dots q_n \dots$ is a **play** in G if

$$\forall i \geq 0 : q_i \in Q_1 \cup Q_2$$

Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$,

$w = q_0 q_1 \dots q_n \dots$ is a **play** in G if

Notations

Let $w = q_0 q_1 \dots q_n \dots$:

$w(i)$ denotes position i

$w(0, i)$ denotes the prefix
up to position i

$last(w(0, i)) = w(i)$

Plays, Prefixes of Plays

Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$,

$w = q_0 q_1 \dots q_n \dots$ is a **play** in G if

$$1) \quad w(0) = \iota$$

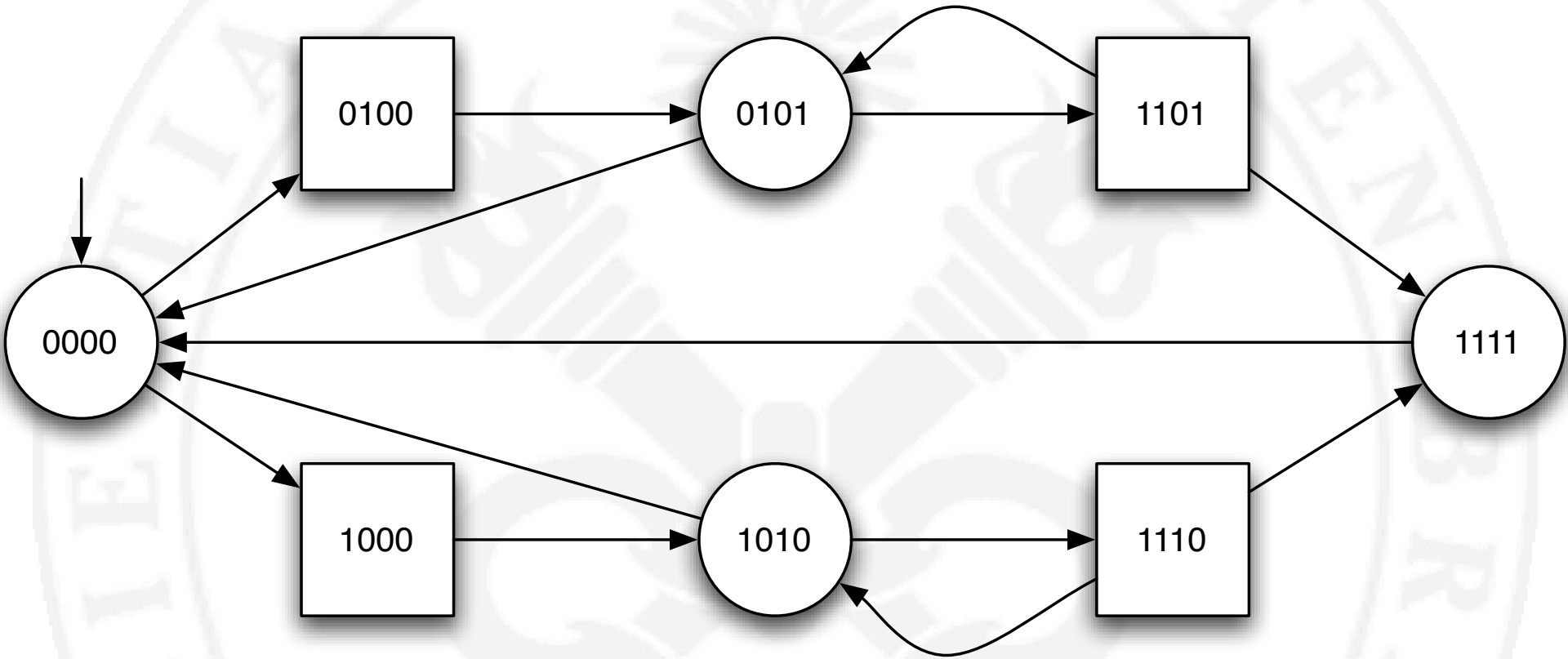
$$2) \quad \forall i \geq 0 : \delta(w(i), w(i+1))$$

We denote the set of plays in G by : $\text{Plays}(G)$
and

$$\text{PrefPlays}(G) = \{q_0 q_1 \dots q_n \mid \exists w \in \text{Plays}(G) \wedge \forall 0 \leq i \leq n : w(i) = q_i\}$$

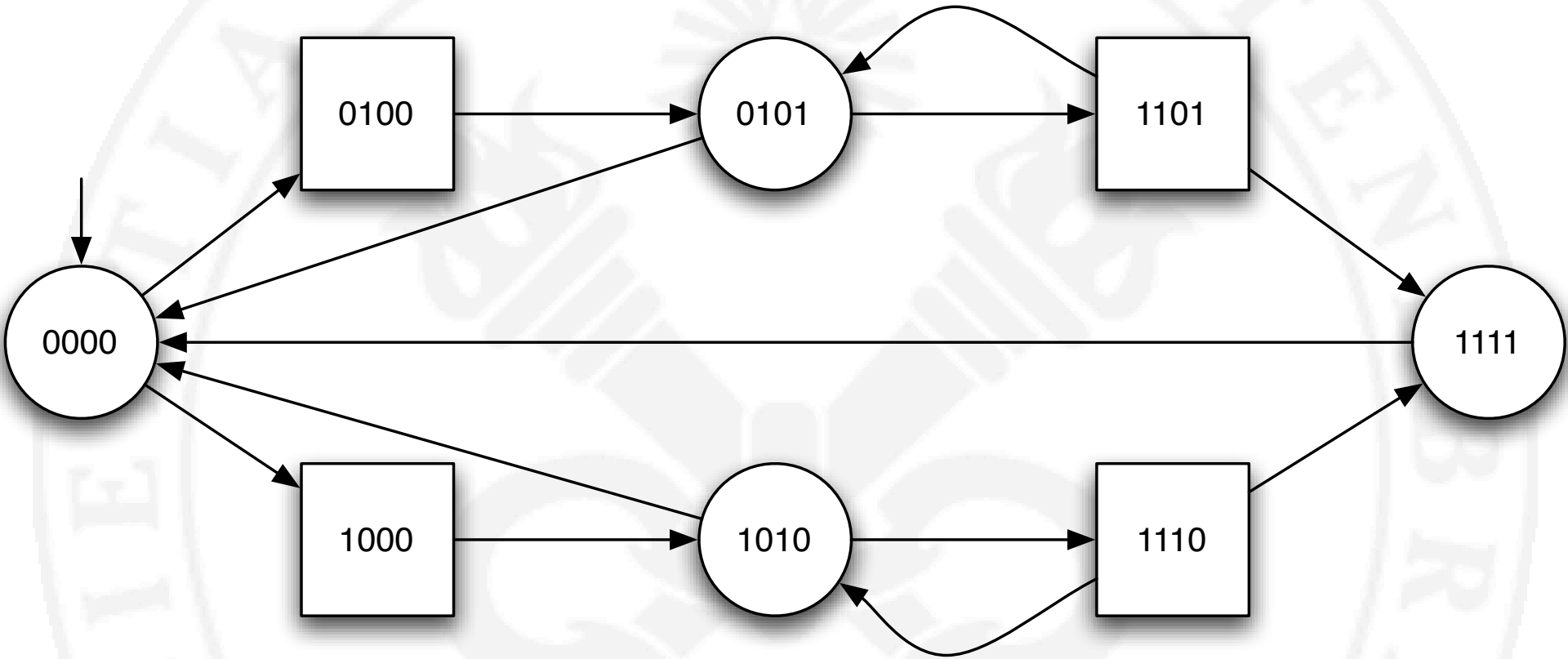
$$\text{PrefPlays}_k(G) = \{w \in \text{PrefPlays}(G) \wedge \text{last}(w) \in Q_k\}$$

Who is winning ?



Play : 0000 0100 0101 1101 ...

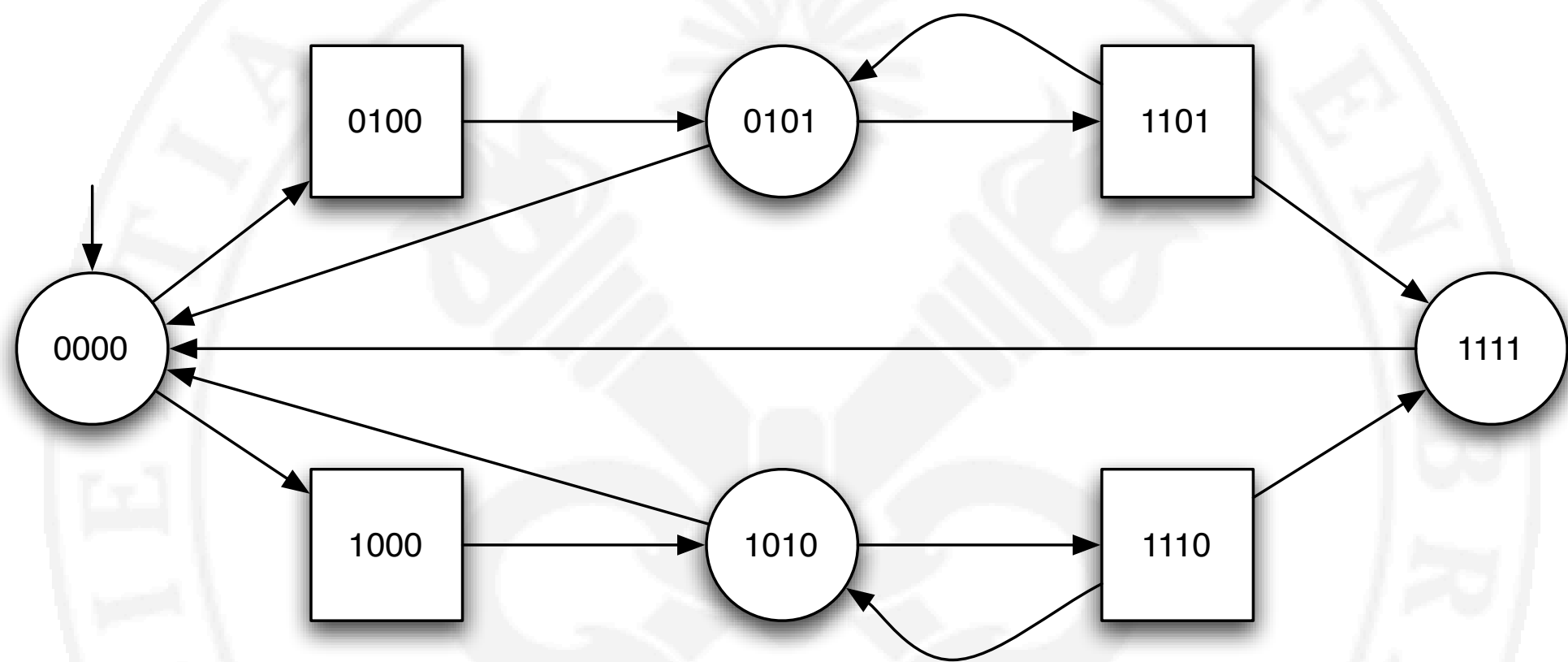
Who is winning ?



Play : 0000 0100 0101 1101 ...

Is this a **good** or a **bad** play for **Player k** ?

Who is winning ?



A winning condition (for Player k)
is a set of plays
$$W \subseteq (Q_1 \cup Q_2)^\omega$$

Game
=
Two-player game structure
+
Winning condition for Player k

Strategies

Players are playing **according to strategies**.

A **Player k strategy** in G is a function:

$$\lambda : \text{PrefPlays}_k(G) \rightarrow Q_1 \cup Q_2$$

with the restriction that:

$$\forall w \in \text{PrefPlays}_k(G) : \delta(\text{last}(w), \lambda(w))$$

Outcome of a strategy

w is a possible **outcome** of the Player k strategy λ if

$$\forall i \geq 0 : w(i) \in Q_k : w(i+1) = \lambda(w(0, i))$$

w is a play where Player k plays according to strategy λ

Outcome of a strategy

w is a possible **outcome** of the Player k strategy λ if

$$\forall i \geq 0 : w(i) \in Q_k : w(i+1) = \lambda(w(0, i))$$

The set of plays that have this property is denoted

$$\text{Outcome}_k(G, \lambda)$$

Winning strategy

- Given a pair (G, W)
- We say that Player k wins the game (G, W) if and only if:

$$\exists \lambda : \text{Outcome}_k(G, \lambda) \subseteq W$$

Winning strategy

- Given a pair (G, W)
- We say that Player k wins the game (G, W) if and only if:

$$\exists \lambda : \text{Outcome}_k(G, \lambda) \subseteq W$$

That is, no matter how the other player resolves his choices, when player k **plays according to** λ , the resulting play belongs to W . Player k can **force** the play to be in W .

Winning strategy

- Given a pair (G, W)
- We say that Player k wins the game (G, W) if and only if:

$$\exists \lambda : \text{Outcome}_k(G, \lambda) \subseteq W$$

We say λ that is a **winning strategy** for player k in the game (G, W)

Winning strategies

=

**Controllers that enforce
winning plays**

Safety Games

Safety Game

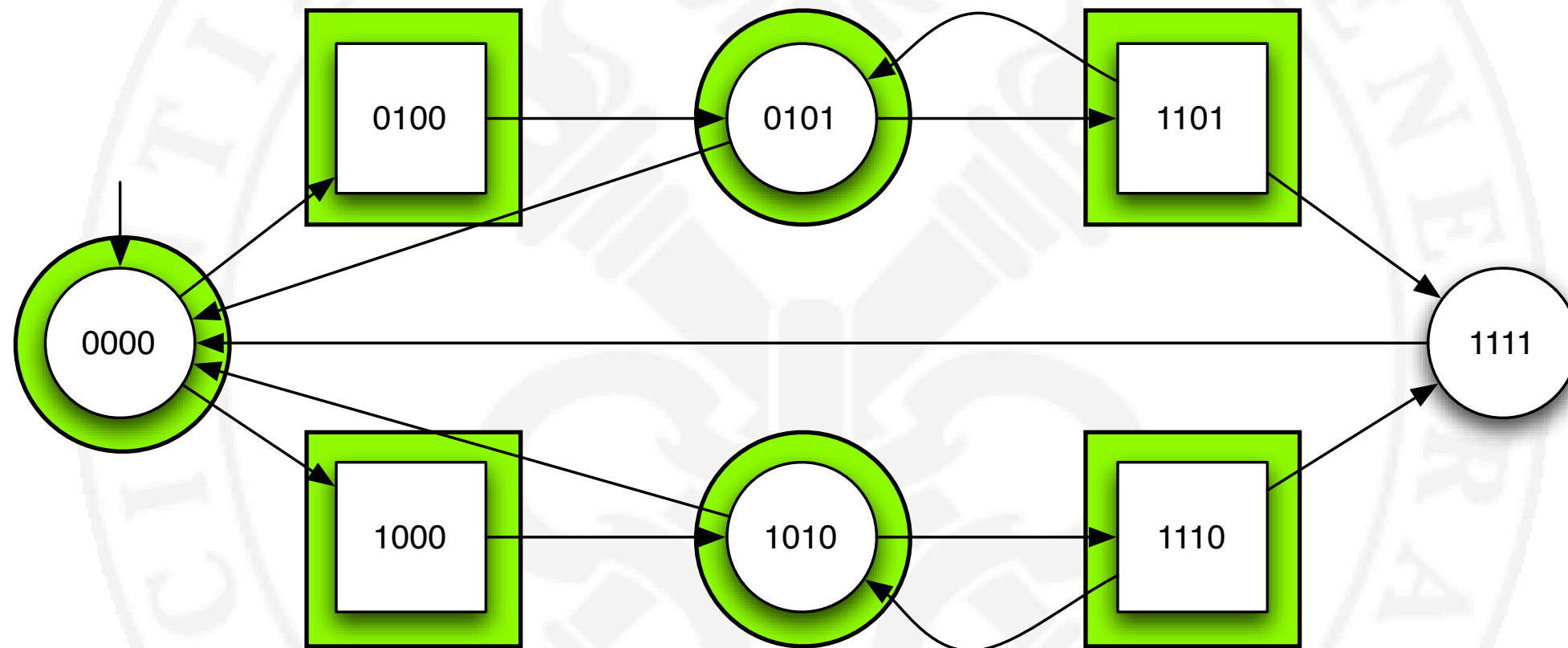
(G, W) is a **safety game** if

$$\exists Q \subseteq Q_1 \cup Q_2 : W = \{w \in \text{Plays}(G) \mid \forall i \geq 0 : w(i) \in Q\}$$

That is W is the set of plays that stay within a given set of positions Q .

$$\text{Safe}(G, Q)$$

A Safety Game



Does Player I, who owns the rounded positions, have a strategy (against any choices of Player II) to stay within the set of states

$$Q \setminus \{1111\} ?$$

Symbolic algorithms to solve games

Complete lattices

A **complete lattice** is a partially ordered set (L, \leq) where every subset of L has a **least upper bound** (often called join or supremum) and a **greatest lower bound** (often called meet or infimum).

Given $M \subseteq L$, **lub**(M) is a value of L such that :

- (i) for all $m \in M : m \leq \mathbf{lub}(M)$ and
- (ii) for all $m' \in L$,
if for all $m \in M : m \leq m'$ then $\mathbf{lub}(M) \leq m'$

Given $M \subseteq L$, **glb**(M) is a value of L such that :

- (i) for all $m \in M : \mathbf{glb}(M) \leq m$ and
- (ii) for all $m' \in L$,
if for all $m \in M : m' \leq m$ then $m' \leq \mathbf{glb}(M)$

Example of complete lattice

2^S , the set of subsets of a set S , ordered by set inclusion \subseteq forms a complete lattice.

Its *least upper bound* is given by union :

$$\text{lub}\{S_1, S_2, \dots, S_n\} = \cup\{S_1, S_2, \dots, S_n\}$$

Its *greatest lower bound* is given by intersection :

$$\text{glb}\{S_1, S_2, \dots, S_n\} = \cap\{S_1, S_2, \dots, S_n\}$$

The *least element* of the lattice is \emptyset and the *largest element* is S .
The powerset complete lattice is noted

$$\langle 2^S, \subseteq, \cup, \cap, S, \emptyset \rangle$$

Monotone functions and fixed points

Let $\langle L, \sqsubseteq, \sqcup, \sqcap, \top, \perp \rangle$ be a complete lattice, let $f : L \rightarrow L$.

We say that f is **monotone** iff

$$\forall l_1, l_2 \in L : l_1 \sqsubseteq l_2 \Rightarrow f(l_1) \sqsubseteq f(l_2)$$

f is **Scott-continuous** iff $\sqcup\{f(l) \mid l \in X\} = f(\sqcup X)$
for any chain X .

We say that l is a fixed point of f iff $l = f(l)$

Any monotone function f over a complete lattice L has:

a **least fixed point**: $\text{lfp } f = \sqcap\{l \mid l = f(l)\}$

a **greatest fixed point**: $\text{gfp } f = \sqcup\{l \mid l = f(l)\}$

Monotone functions and fixed points

Let $\langle L, \sqsubseteq, \sqcup, \sqcap, \top, \perp \rangle$ be a complete lattice, let $f : L \rightarrow L$.
We say that f is **monotone** iff

$$\forall l_1, l_2 \in L : l_1 \sqsubseteq l_2 \implies f(l_1) \sqsubseteq f(l_2)$$

f is **Scott-continuous** iff
for any chain X .

We say that l is a fixed point of f if $l = f(l)$.

Any monotone function f has

a **least fixed point**: $\text{lfp } f = \sqcap \{l \mid l = f(l)\}$

a **greatest fixed point**: $\text{gfp } f = \sqcup \{l \mid l = f(l)\}$

Monotony is equivalent
to Scott-continuity on
any **finite** complete
lattice.

Player k Controllable Predecessors

X is a set of positions

$$1CPre_G(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') : q' \in X\}$$

Set of Player I positions where she has a choice of successor that lies in X

Set of Player II positions where all her choices for successors lie in X

Player k Controllable Predecessors

$$1\text{CPre}_G(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') : q' \in X\}$$

Symmetrically

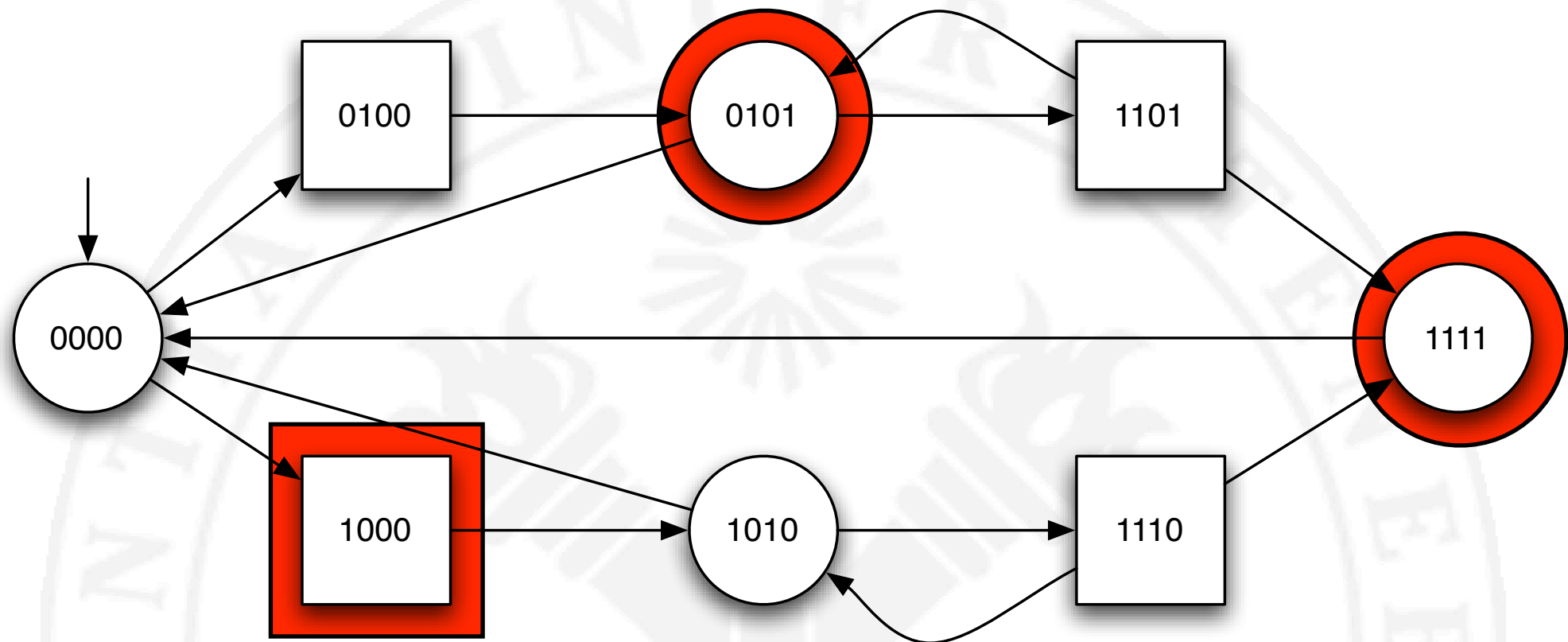
$$2\text{CPre}_G(X) = \{q \in Q_2 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_1 \mid \forall q' : \delta(q, q') : q' \in X\}$$

Player k Controllable Predecessors

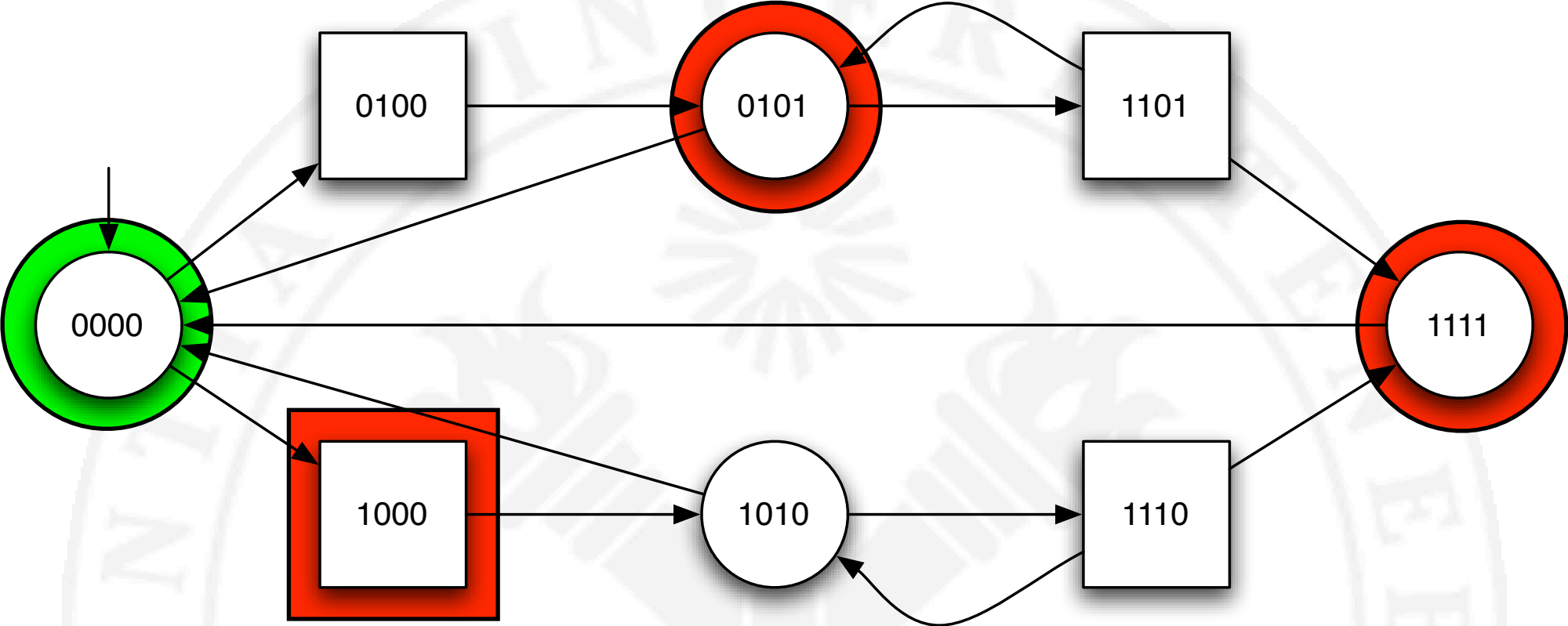
$$1\text{CPre}_G(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') : q' \in X\}$$

Monotonic functions over $\langle 2^{Q_1 \cup Q_2}, \subseteq \rangle$

$$2\text{CPre}_G(X) = \{q \in Q_2 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_1 \mid \forall q' : \delta(q, q') : q' \in X\}$$



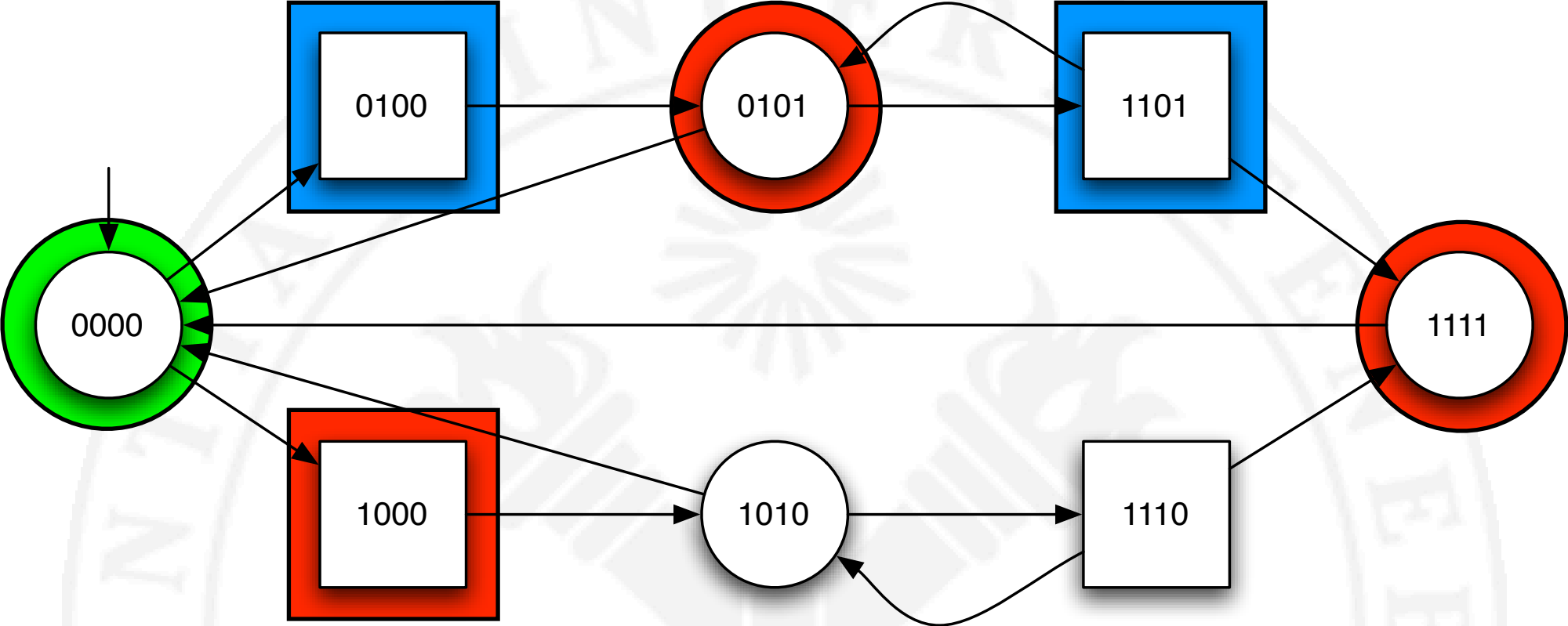
$$X = \{1000, 0101, 1111\}$$



$$X = \{1000, 0101, 1111\}$$

$$1CPre(X) = \{0000\} \cup \{0100, 1101\}$$

Rounded positions,
there exists a red successor



$$X = \{1000, 0101, 1111\}$$

$$1CPre(X) = \{0000\} \cup \{0100, 1101\}$$

Rounded positions,
there exists a red successor

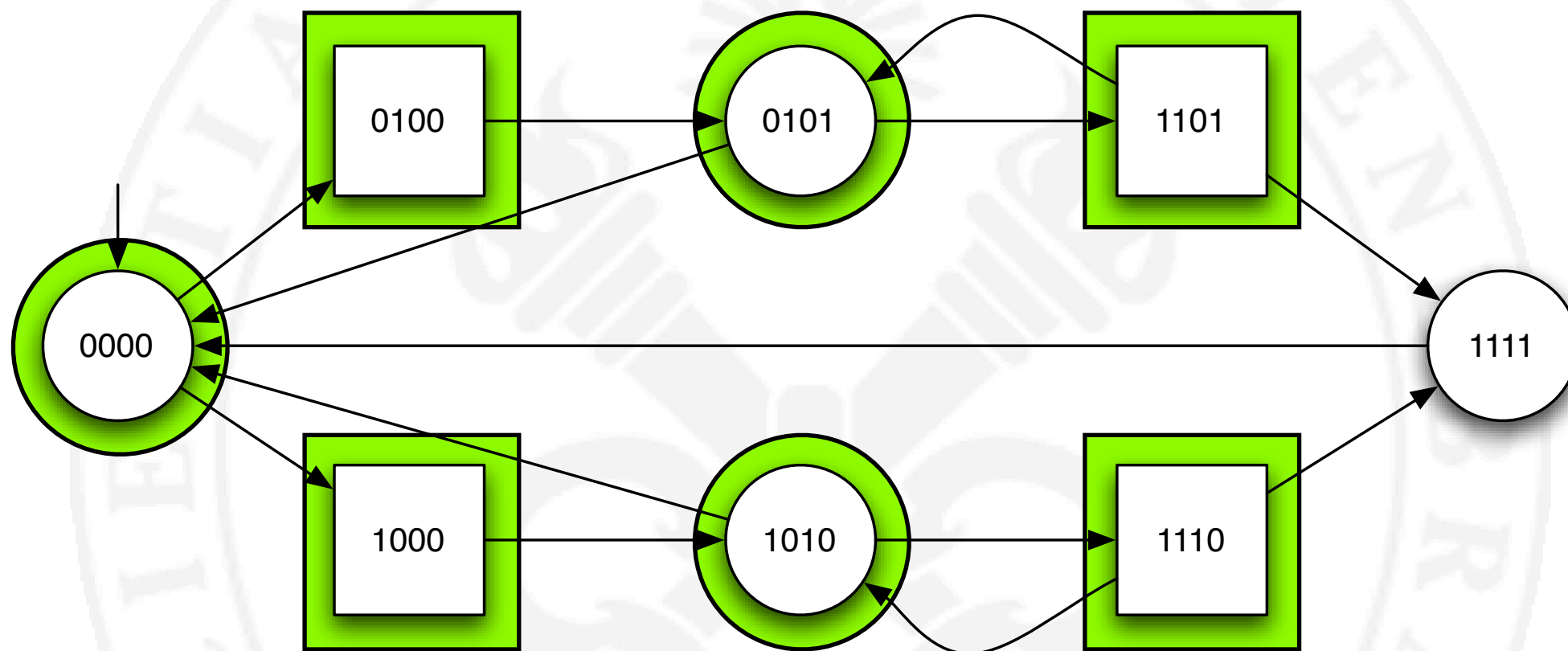
Squared positions,
all successors are red

Fixed points to solve games

Let Q be a set of safe states, the states in which Player I can force the game to within Q is given by the following fixed point expression :

$$\cup \{R \mid R = Q \cap \text{CPre}_1(R)\}$$

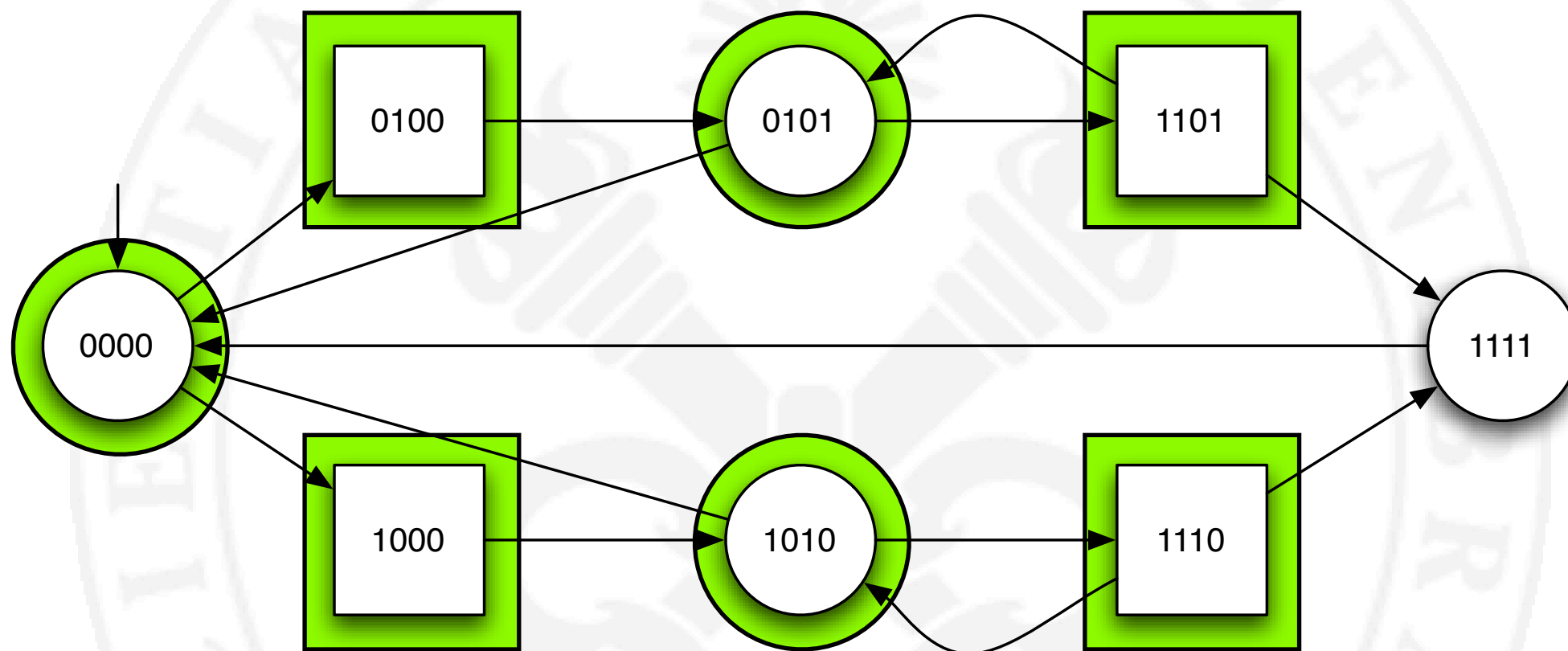
Fixpoint for a safety game



Does Player I, who owns the rounded positions, have a strategy to stay within the set of states

$$Q \setminus \{1111\}?$$

Fixpoint for a safety game



We must compute

$$\cup \{R \mid R = (Q_1 \cup Q_2) \setminus \{1111\} \cap \text{CPre}_1(R)\}$$

To do that, we use the Tarski fixpoint theorem.

Tarski-Kleene Theorem

Let $\langle L, \sqsubseteq, \sqcup, \sqcap, \top, \perp \rangle$ be a complete lattice, the f be a Scott-continuous function on L , then

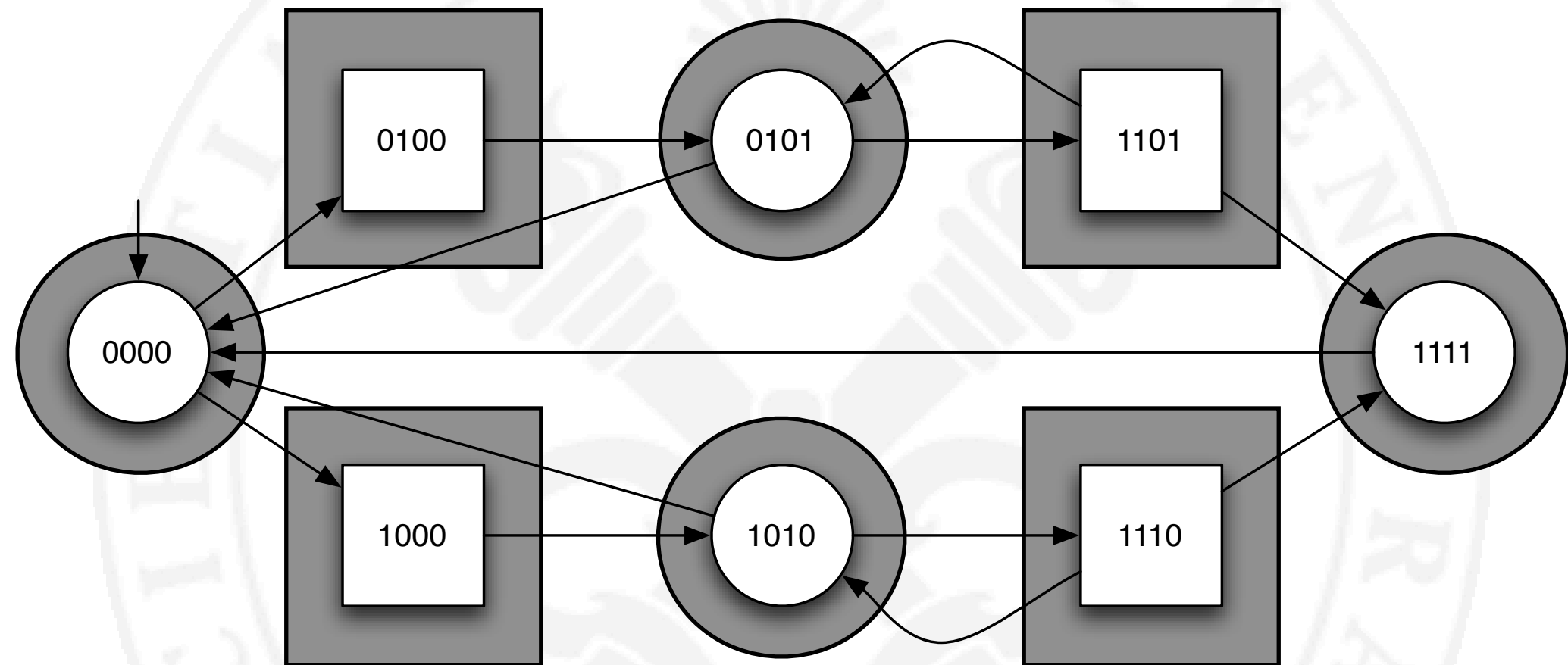
lfp f is the limit of the sequence :

$$f(\perp), f(f(\perp)), \dots, f(\dots f(\perp)\dots), \dots$$

gfp f is the limit of the sequence :

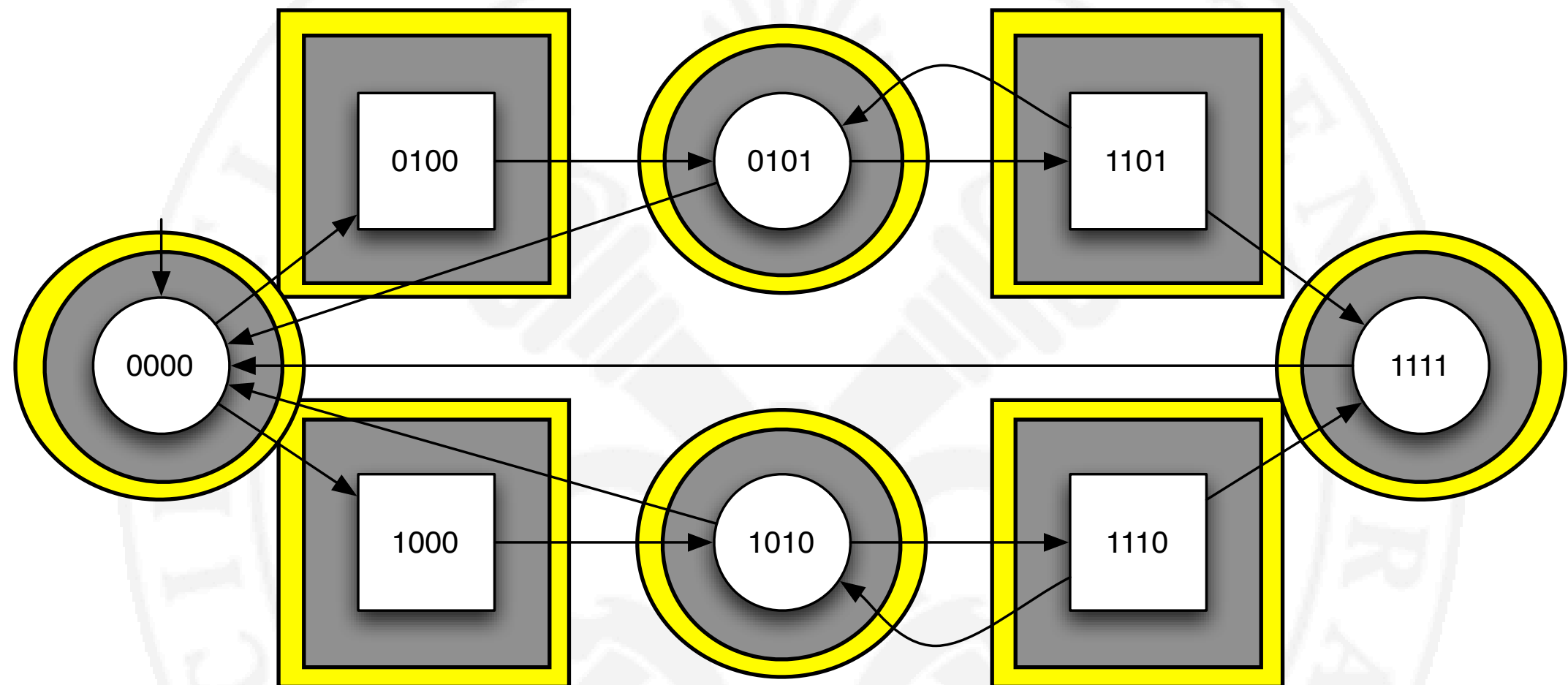
$$f(\top), f(f(\top)), \dots, f(\dots f(\top)\dots), \dots$$

Fixpoint for a safety game



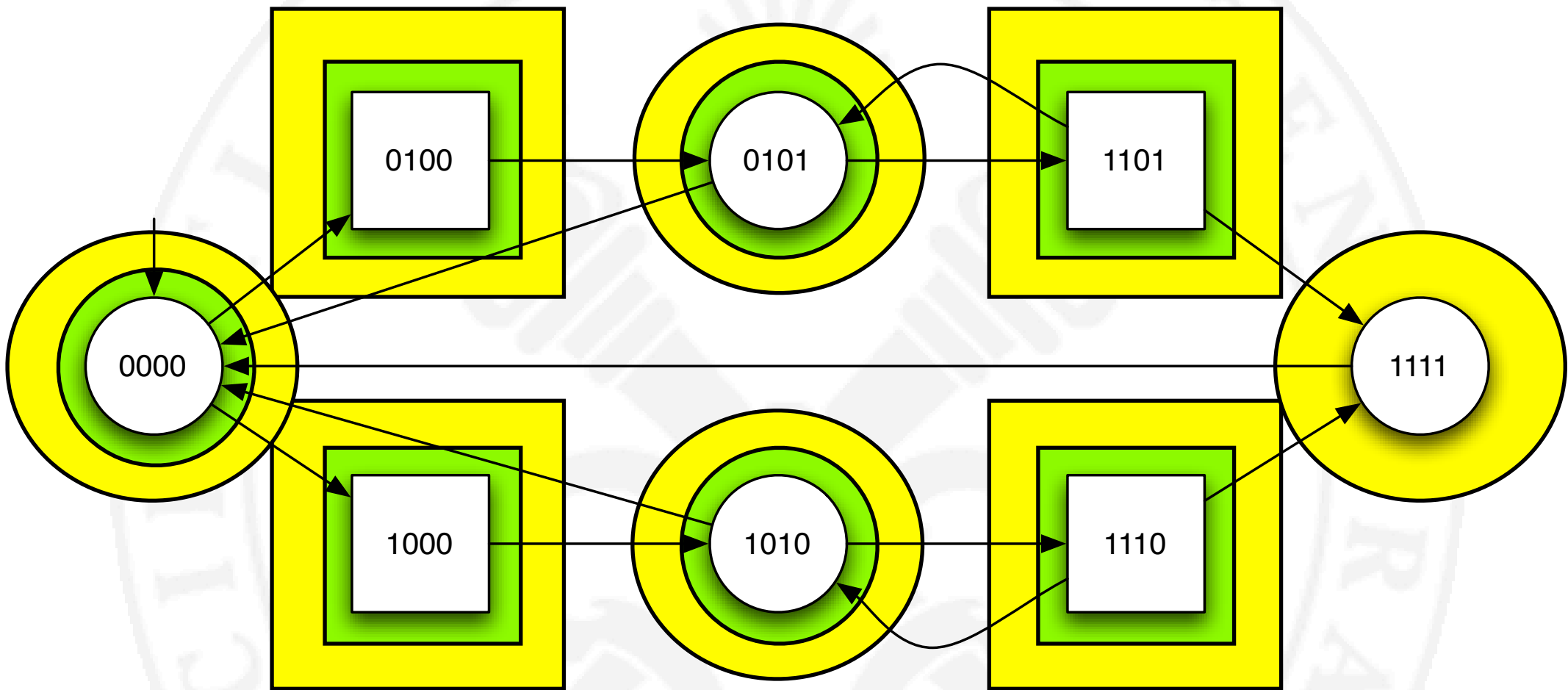
$$X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q)$$

Fixpoint for a safety game



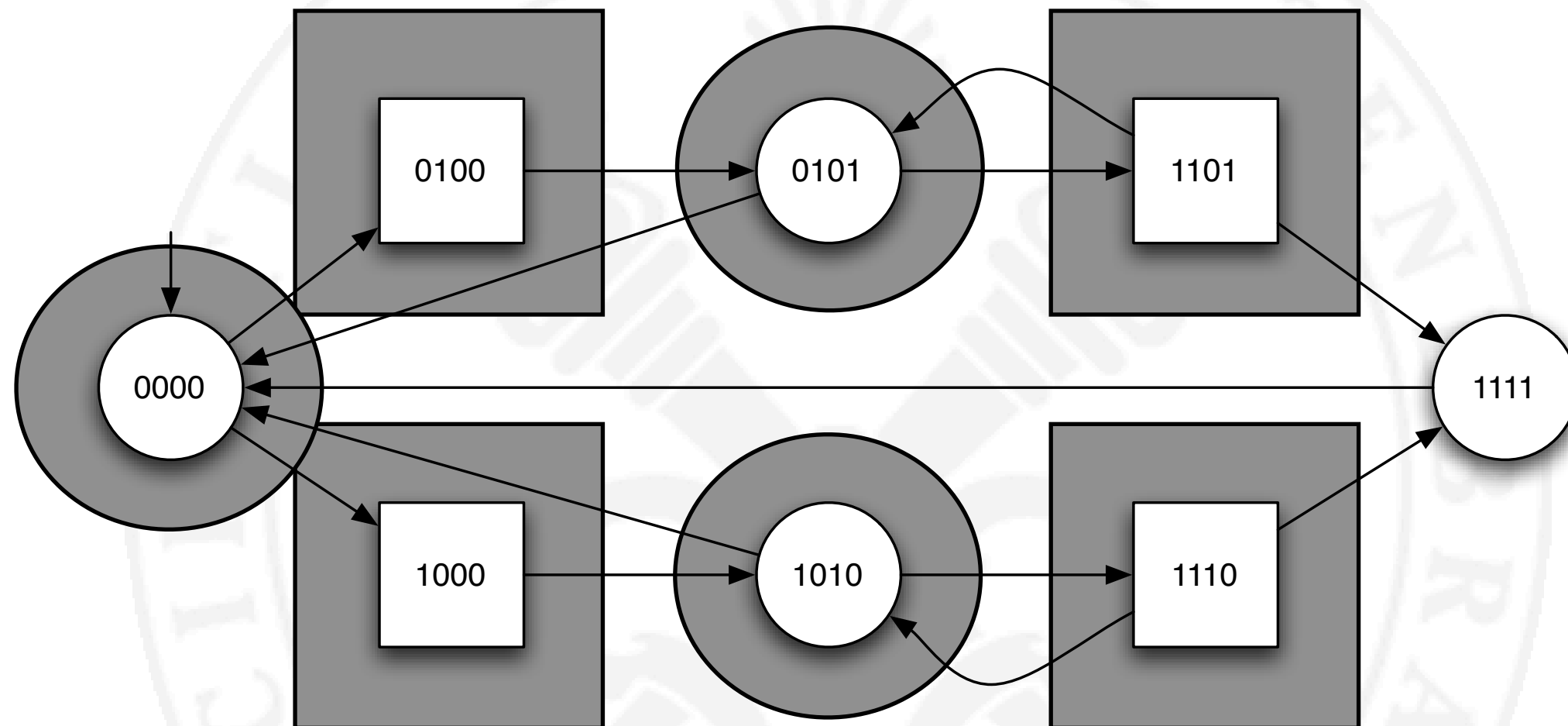
$$X_0 = (Q \setminus \{1111\}) \cap \mathbf{1CPre}(Q)$$

Fixpoint for a safety game



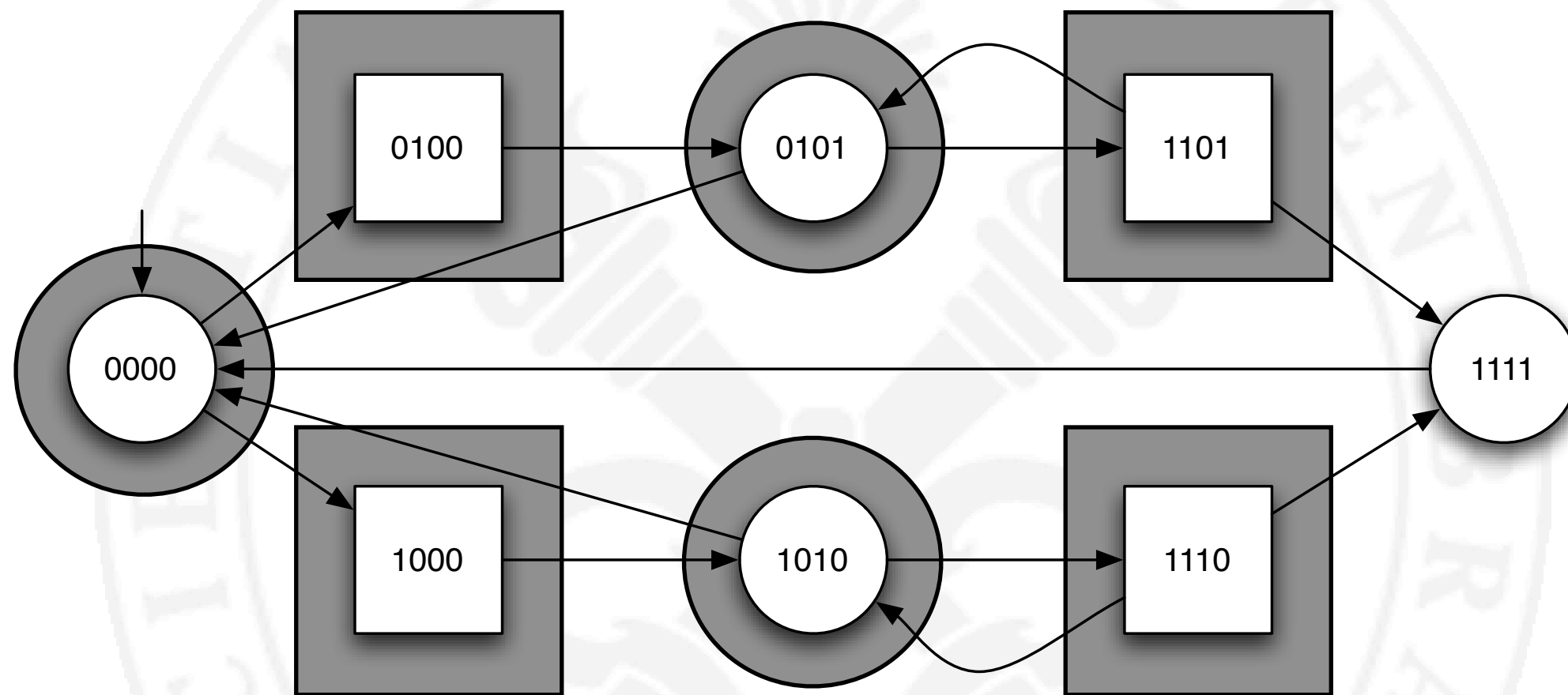
$$X_0 = (Q \setminus \{1111\}) \cap \text{1CPre}(Q)$$

Fixpoint for a safety game



$$X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q)$$

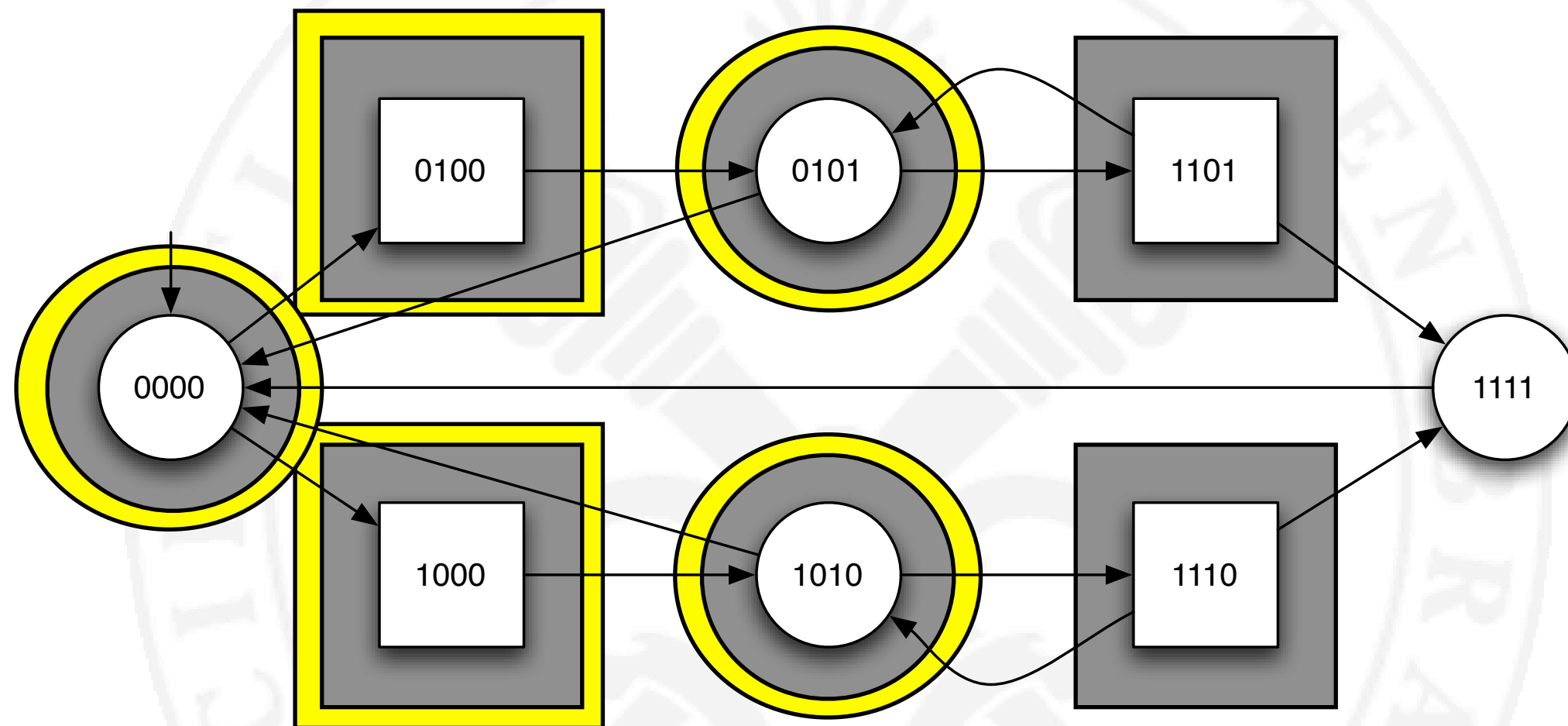
Fixpoint for a safety game



$$X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0)$$

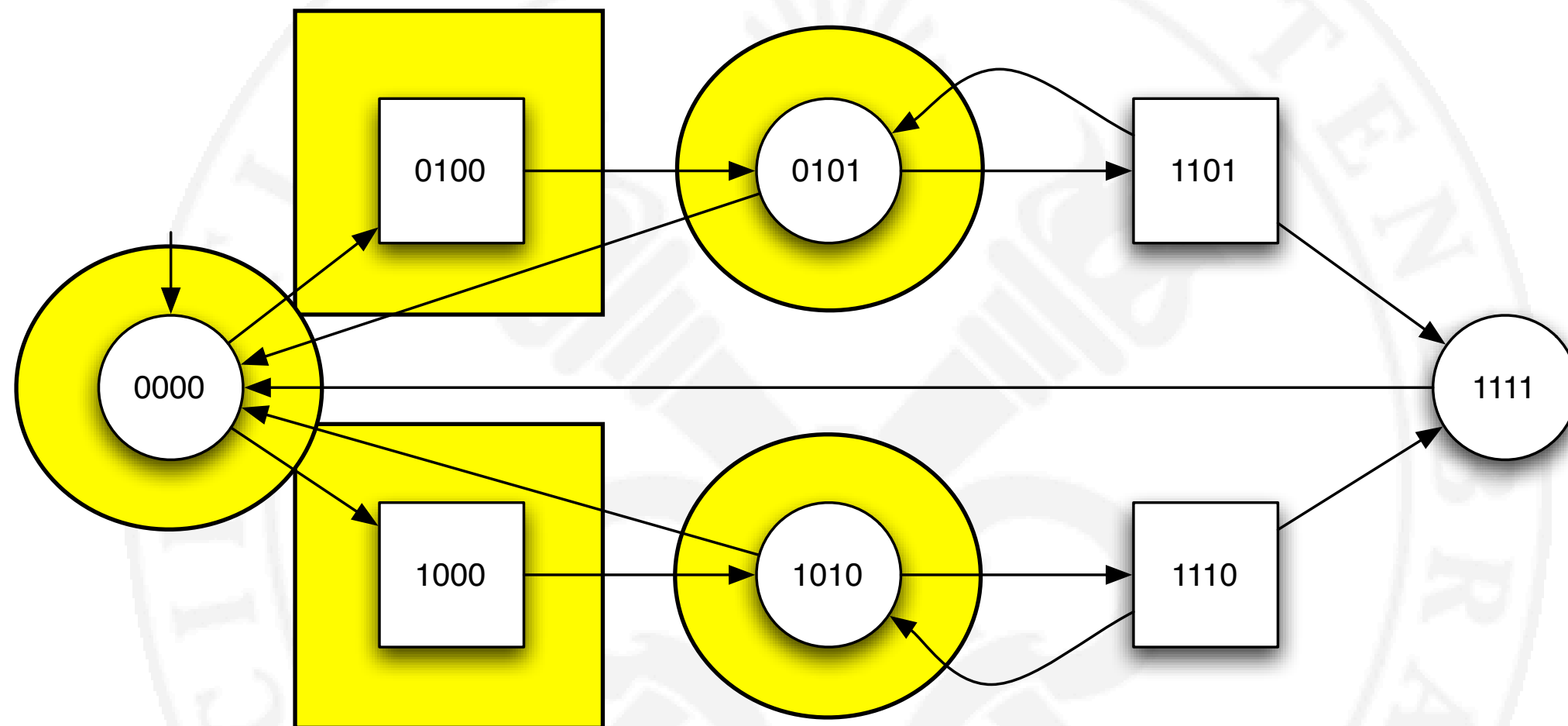
Fixpoint for a safety game



$$X_0 = (Q \setminus \{1111\}) \cap 1CPre(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

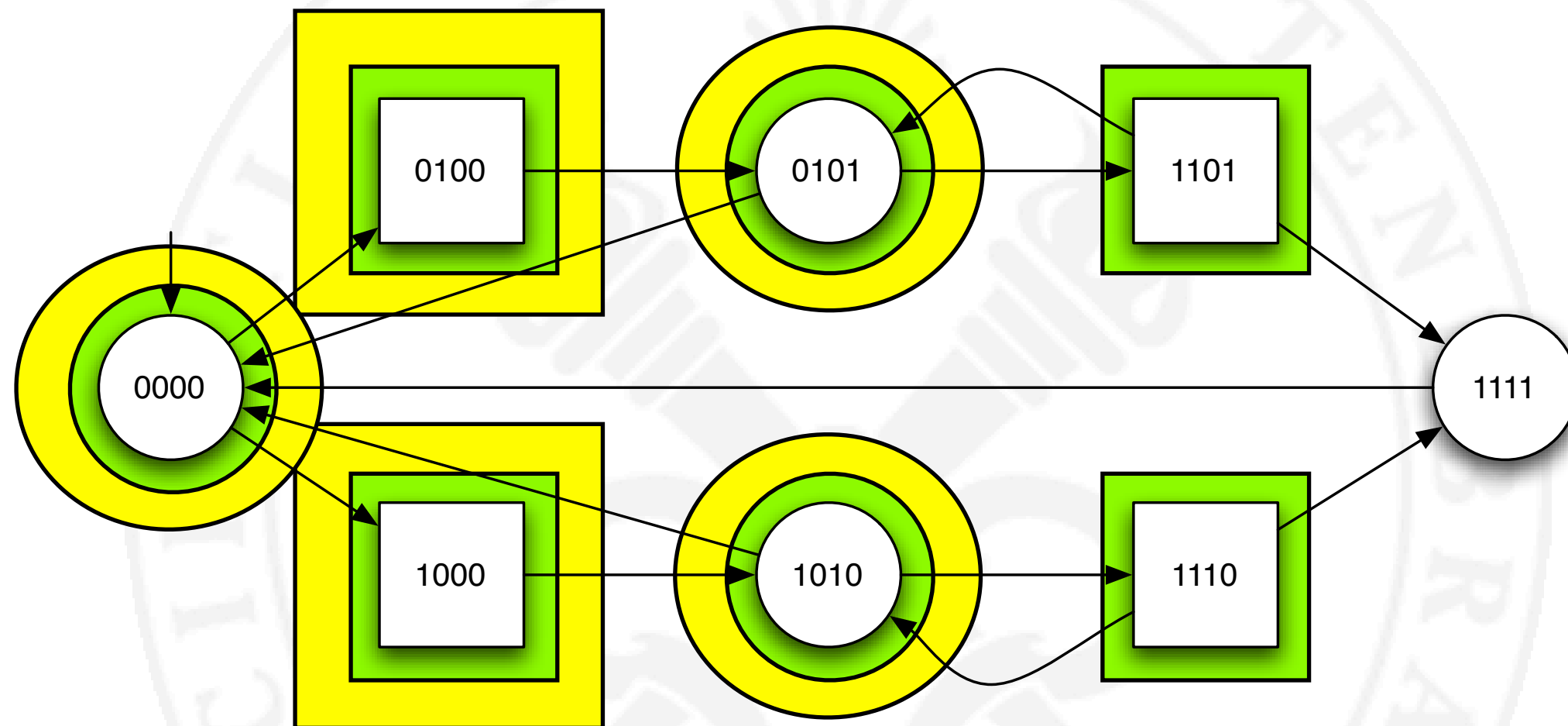
Fixpoint for a safety game



$$X_0 = (Q \setminus \{1111\}) \cap 1CPre(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

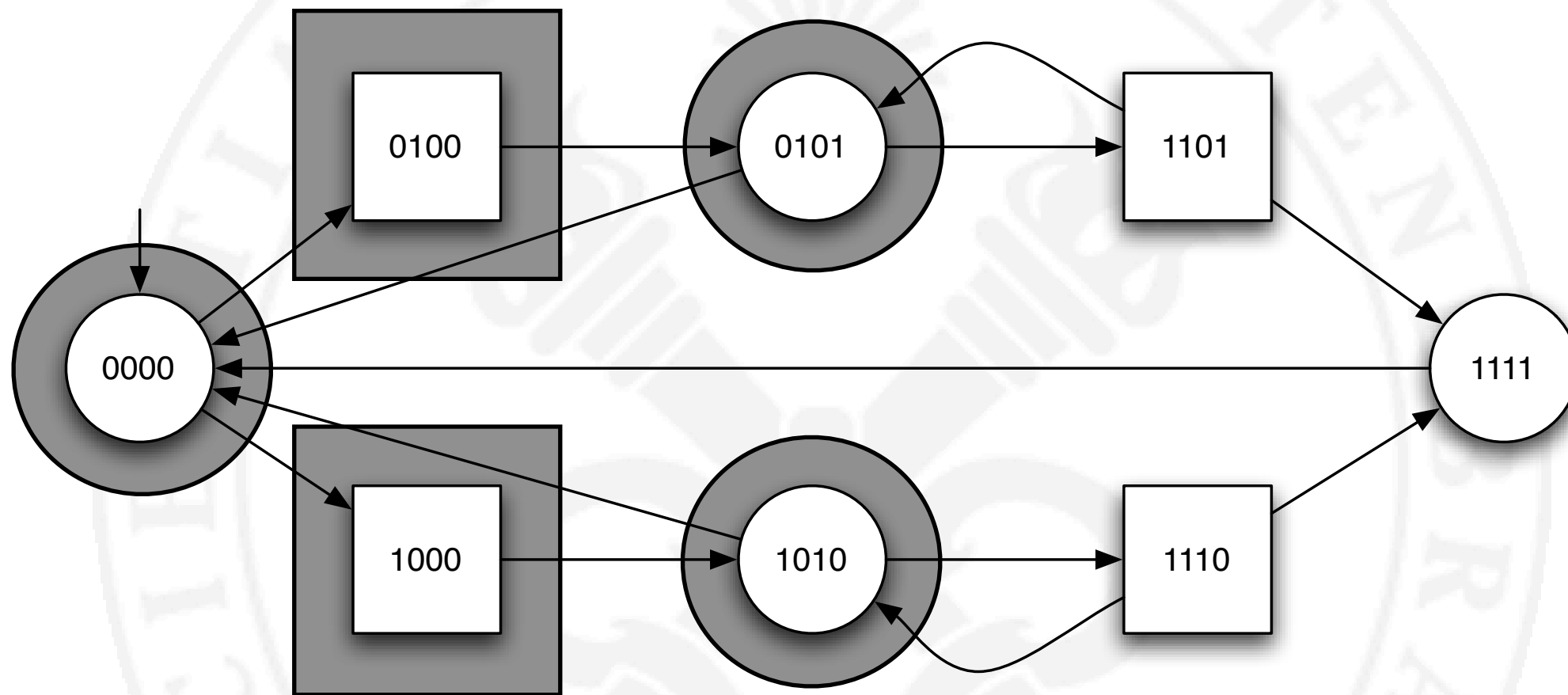
Fixpoint for a safety game



$$X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0)$$

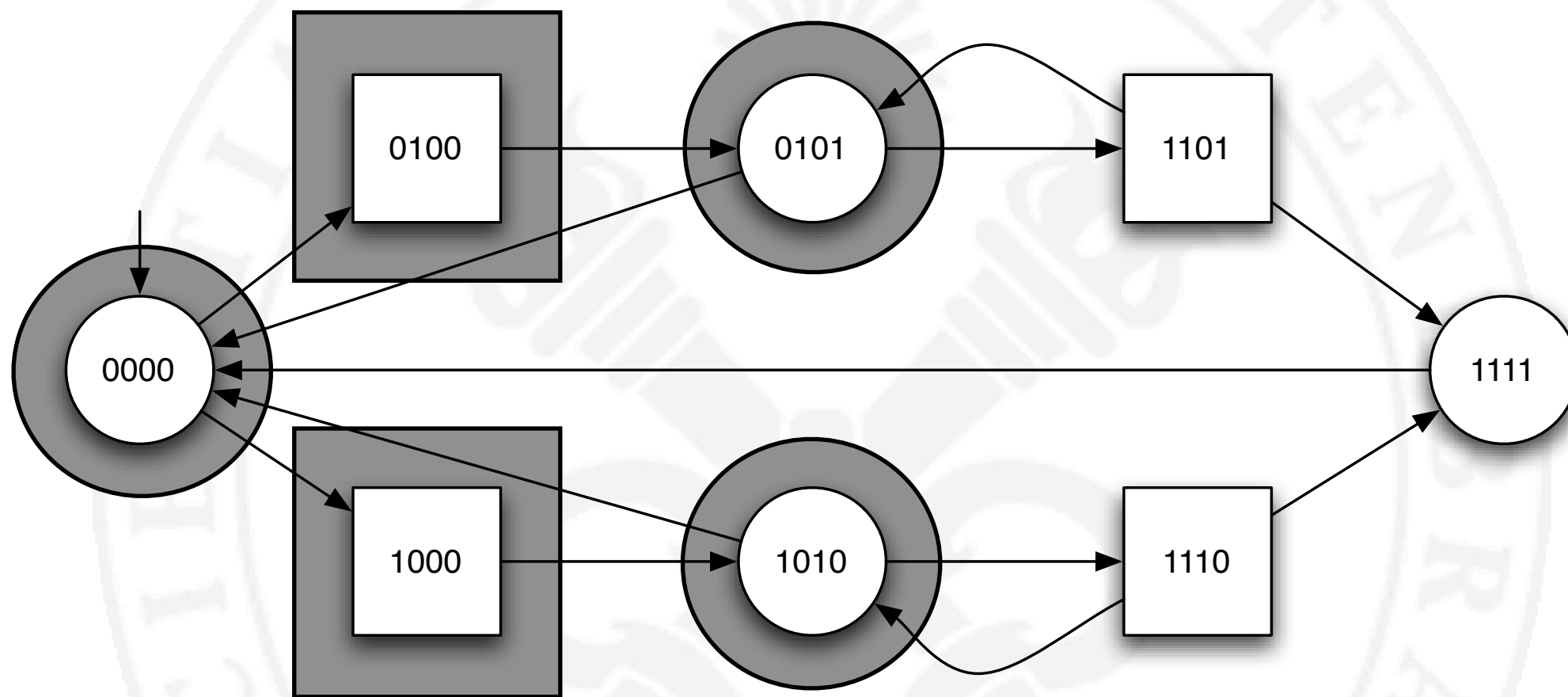
Fixpoint for a safety game



$$X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0)$$

Fixpoint for a safety game

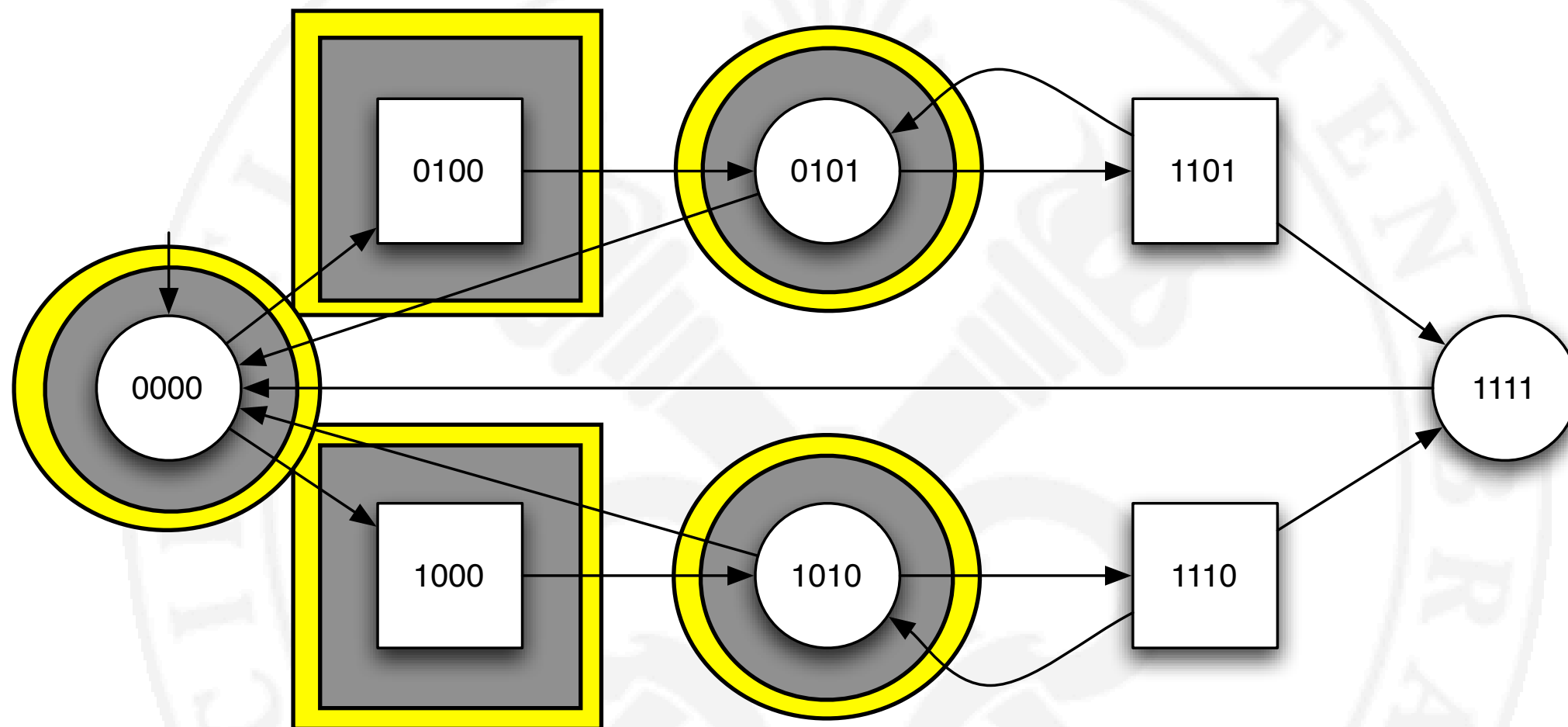


$$X_0 = (Q \setminus \{1111\}) \cap 1\text{CPre}(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap 1\text{CPre}(X_1)$$

Fixpoint for a safety game

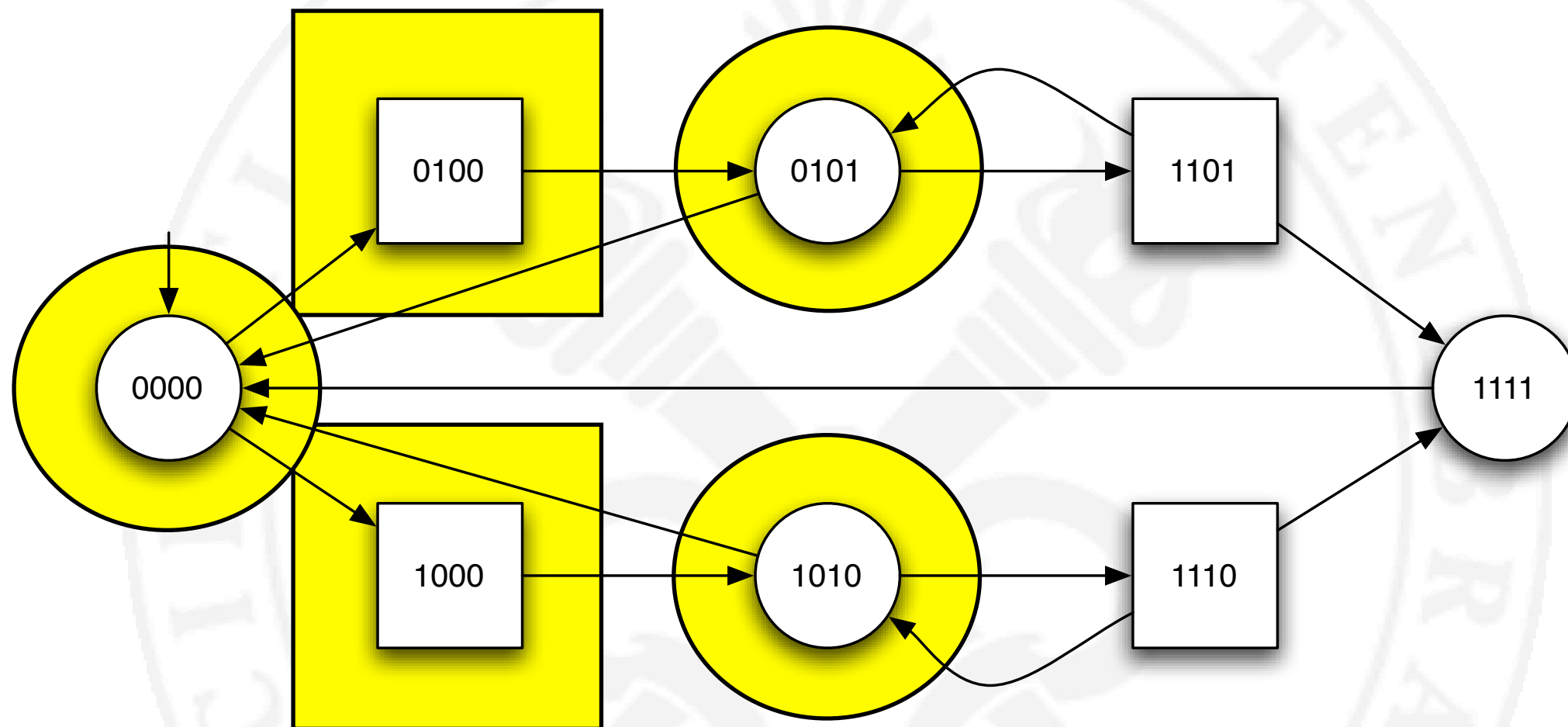


$$X_0 = (Q \setminus \{1111\}) \cap \text{1CPre}(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap \text{1CPre}(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap \text{1CPre}(X_1)$$

Fixpoint for a safety game

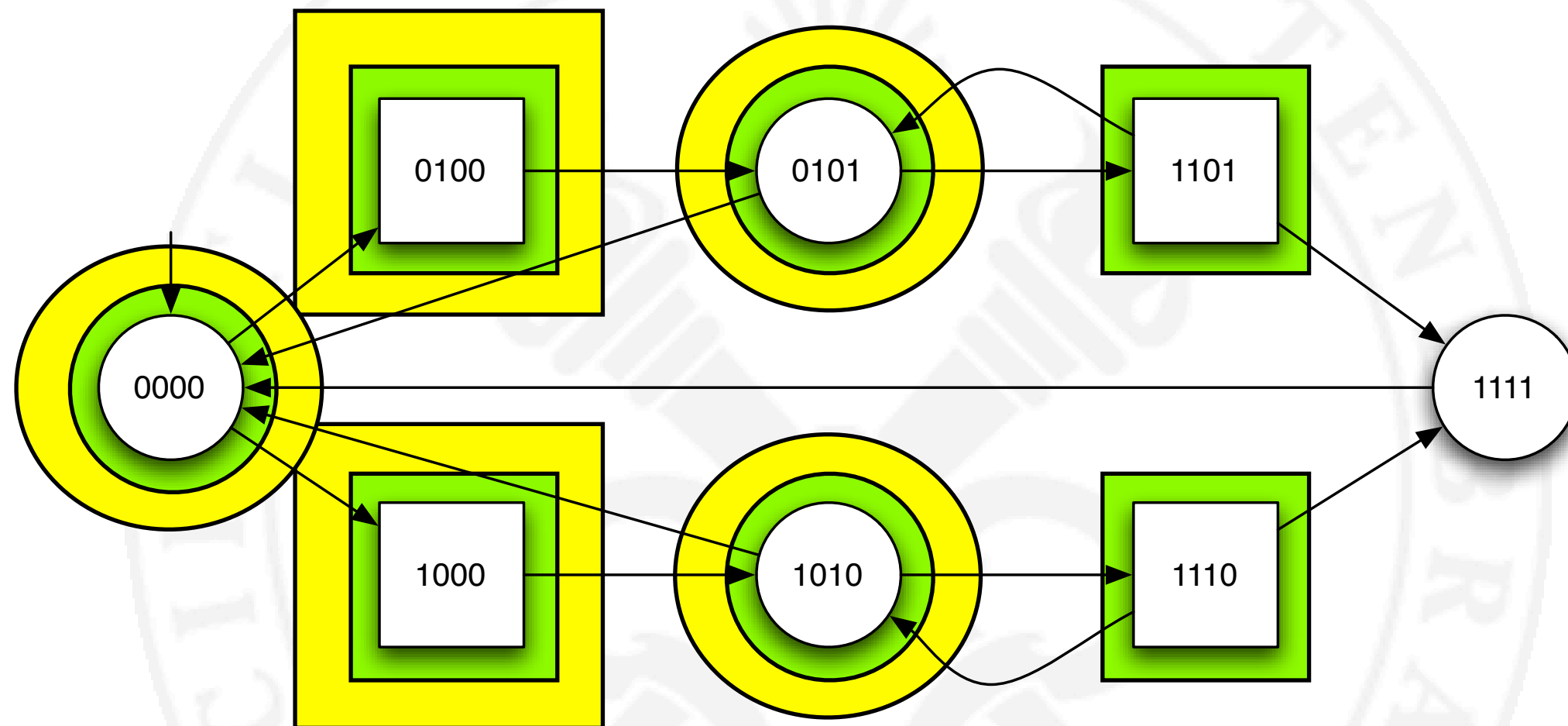


$$X_0 = (Q \setminus \{1111\}) \cap \text{1CPre}(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap \text{1CPre}(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap \text{1CPre}(X_1)$$

Fixpoint for a safety game

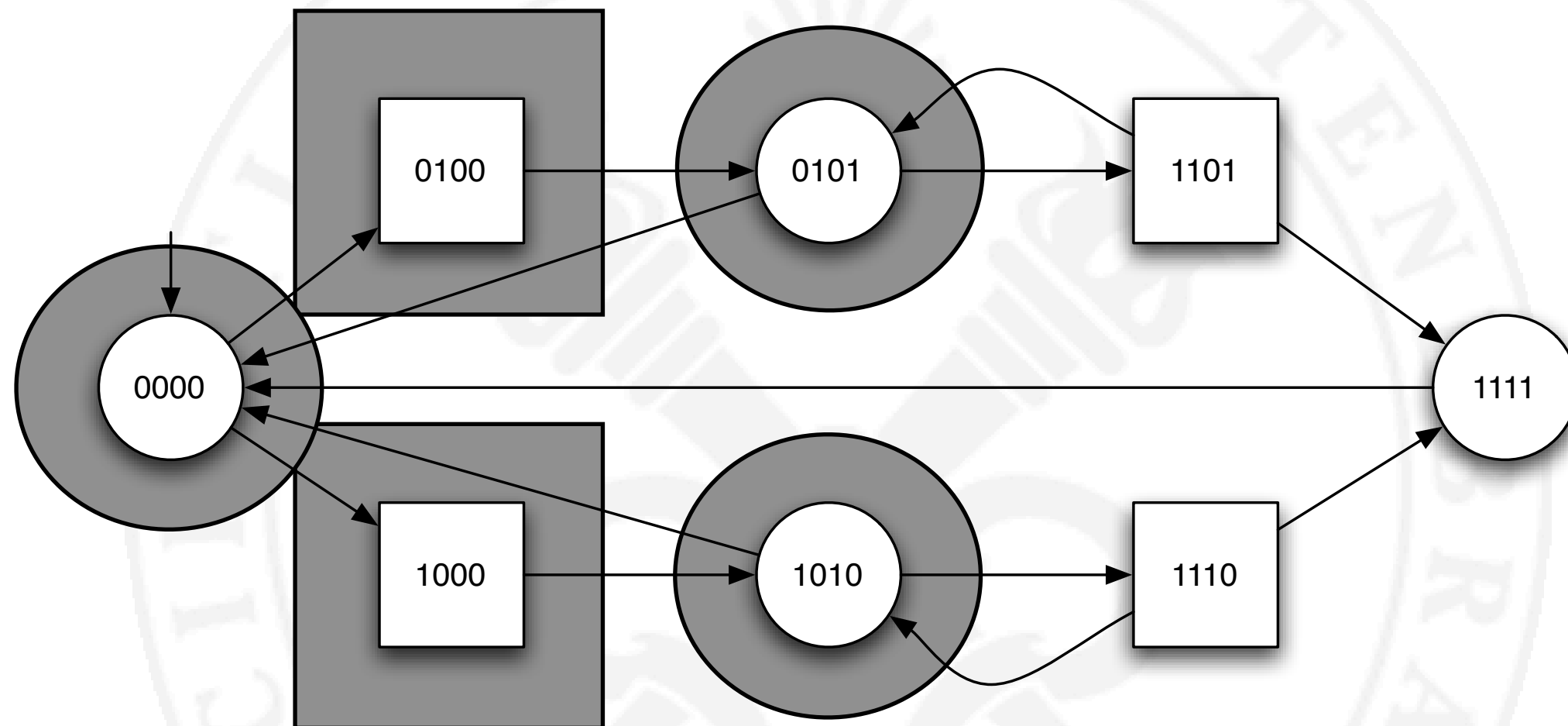


$$X_0 = (Q \setminus \{1111\}) \cap 1CPre(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap 1CPre(X_1)$$

Fixpoint for a safety game

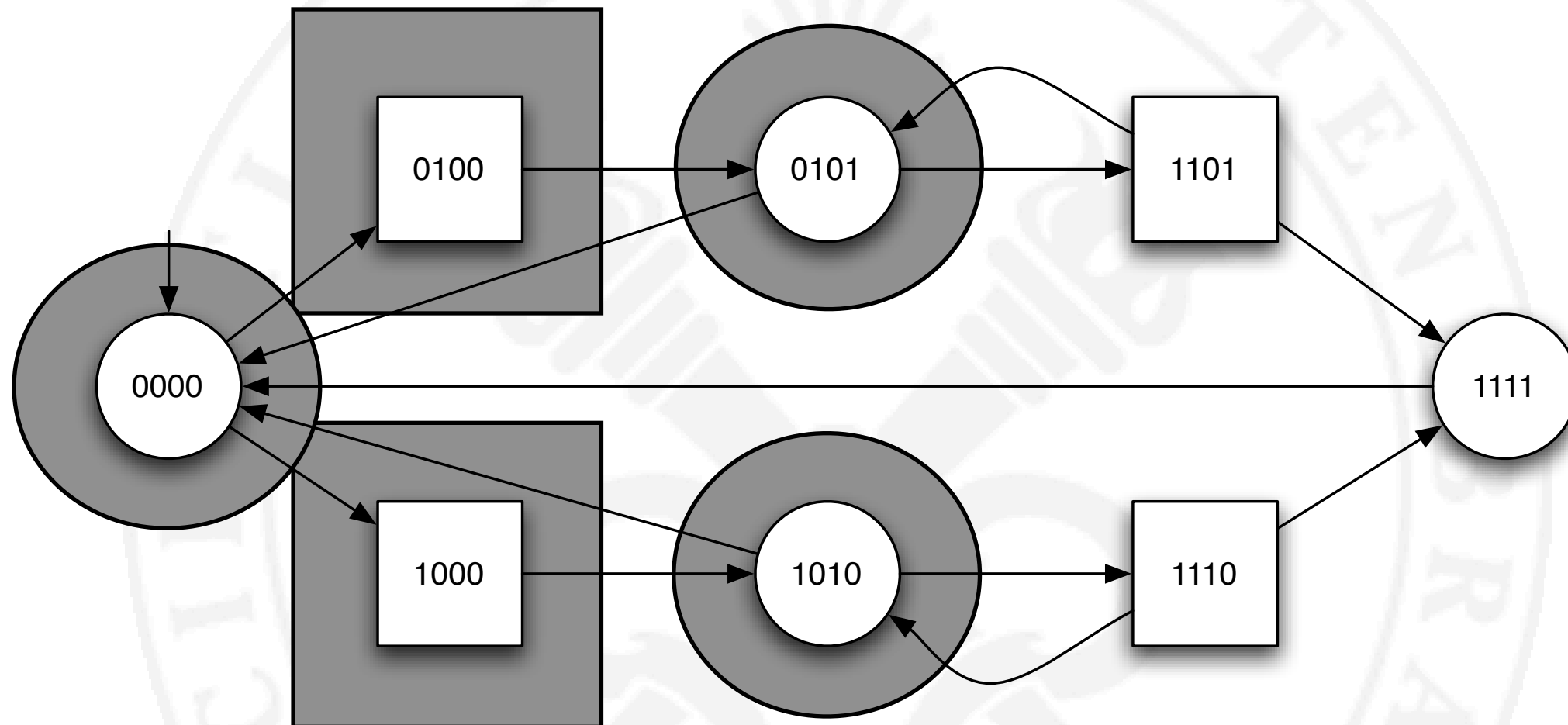


$$X_0 = (Q \setminus \{1111\}) \cap 1CPre(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap 1CPre(X_1)$$

Fixpoint for a safety game



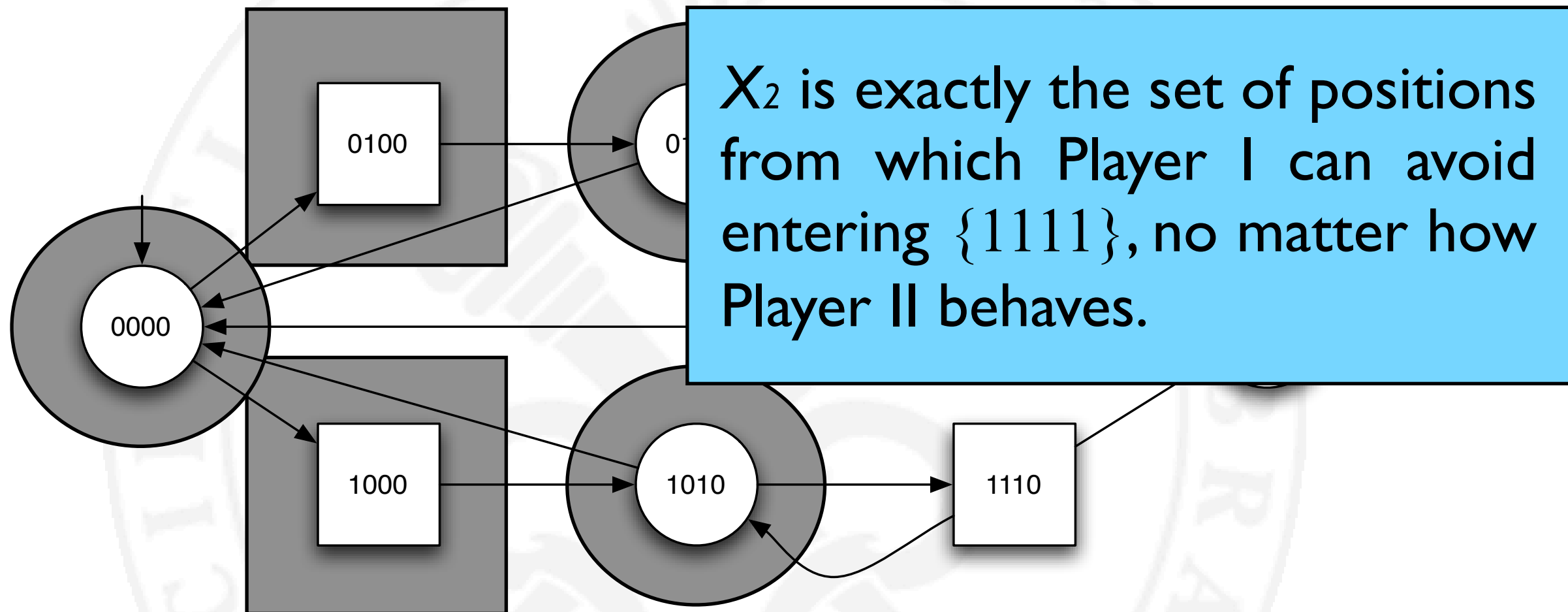
This is the
greatest
fixed point

$$X_0 = (Q \setminus \{1111\}) \cap 1CPre(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap 1CPre(X_1) = X_1$$

Fixpoint for a safety game



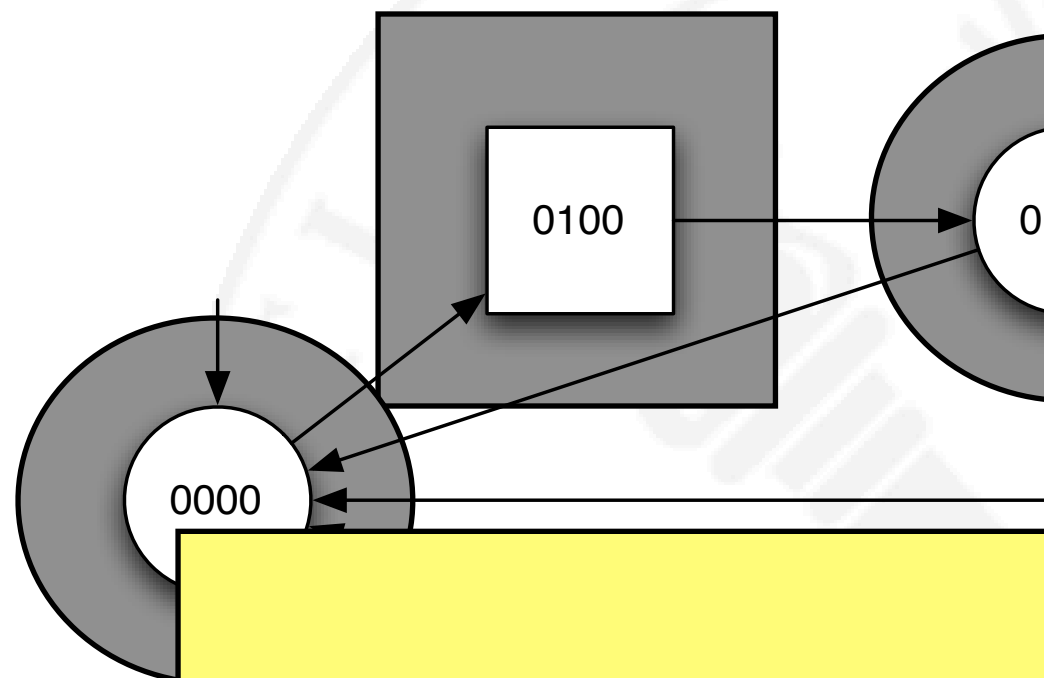
This is the
greatest
fixed point

$$X_0 = (Q \setminus \{1111\}) \cap 1CPre(Q)$$

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap 1CPre(X_1) = X_1$$

Fixpoint for a safety game



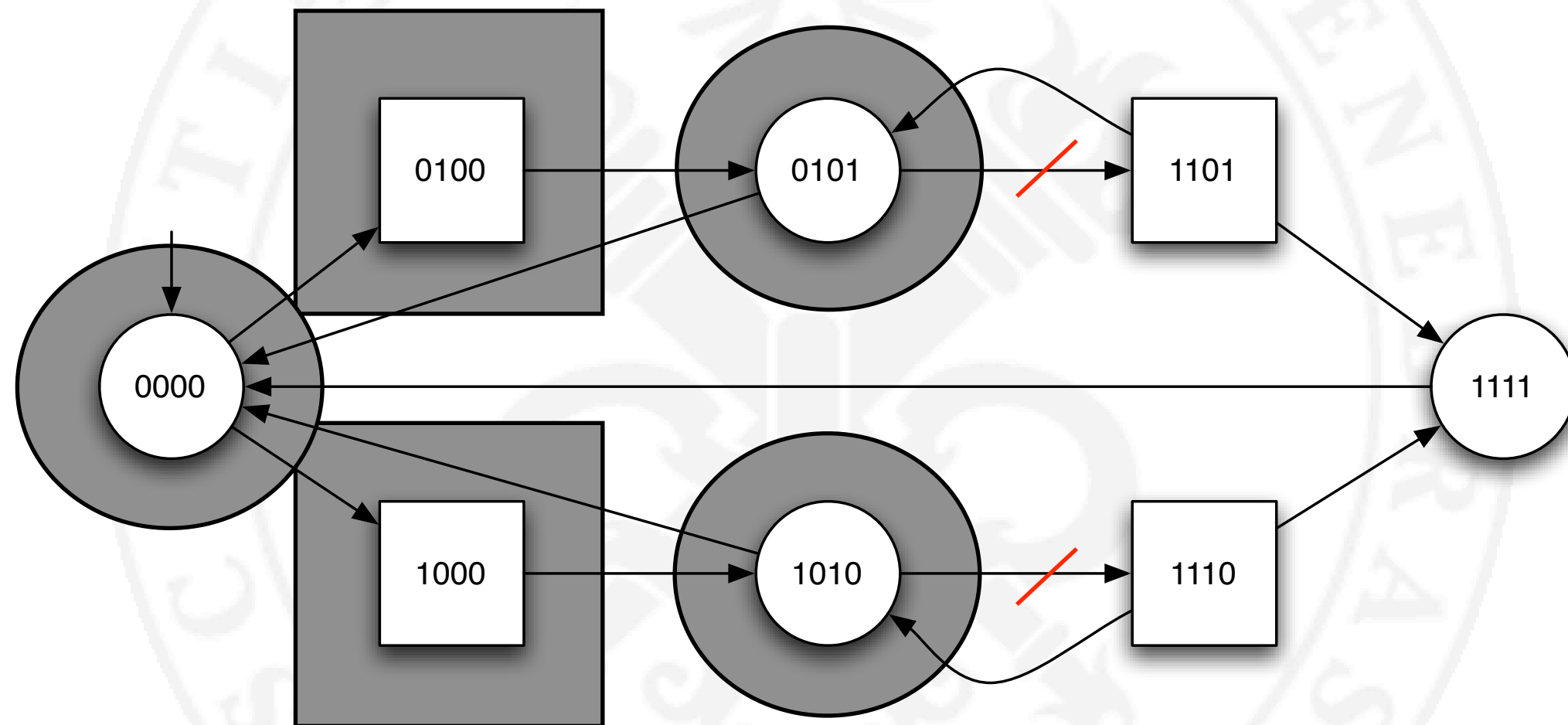
X_2 is exactly the set of positions from which Player I can avoid entering $\{1111\}$, no matter how Player II behaves.

Player I has a positional (memoryless) strategy to win the game

This is the greatest fixed point

$$X_1 = (Q \setminus \{1111\}) \cap 1CPre(X_0)$$

$$X_2 = (Q \setminus \{1111\}) \cap 1CPre(X_1) = X_1$$

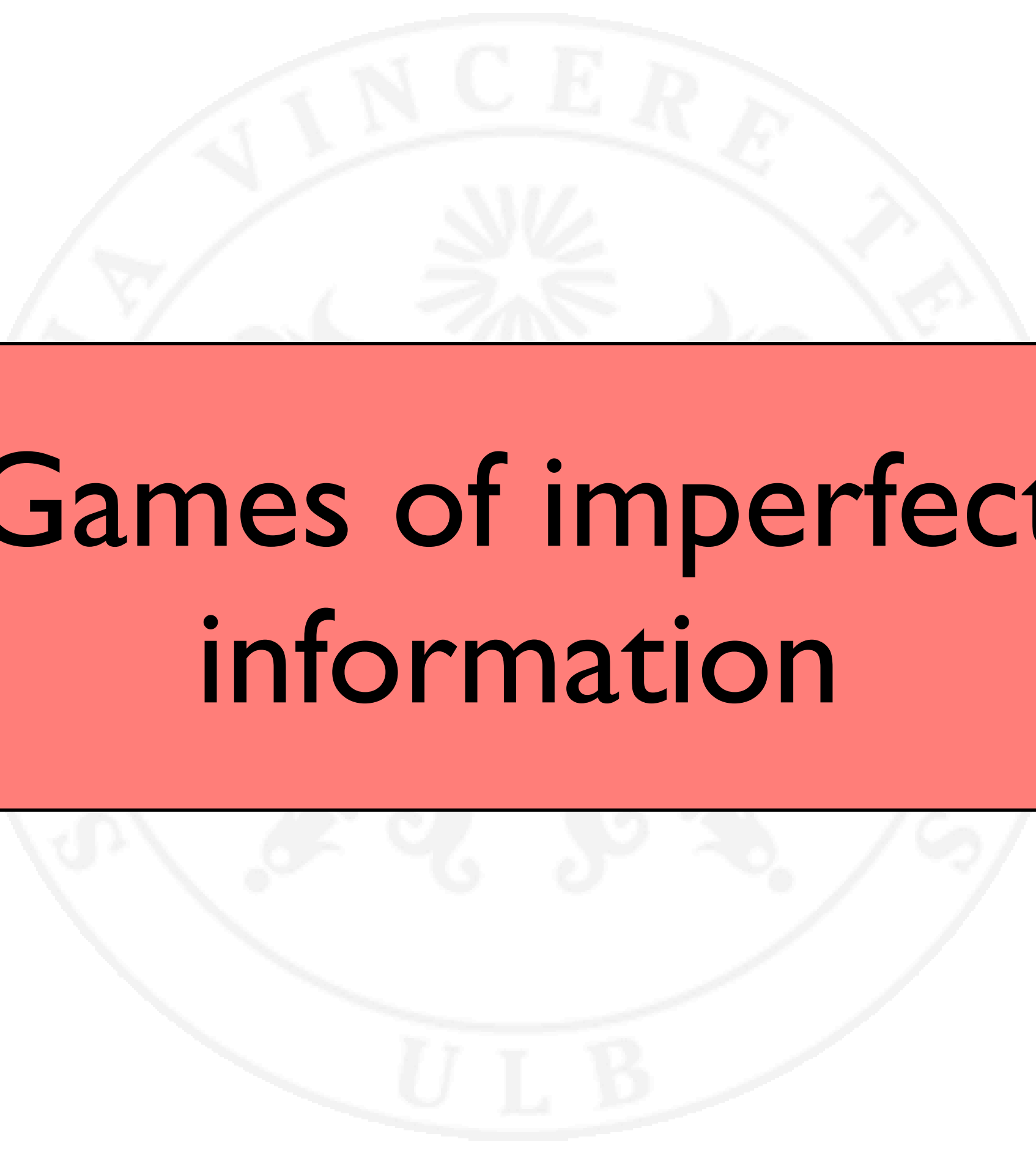


Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$ be a TGS, let $\text{Reach}(G, Q)$ be a **reachability** game defined on G , Player I has a winning strategy for this game iff

$$\iota \in \cap \{R \mid R = Q \cup \text{CPre}_1(R)\}$$

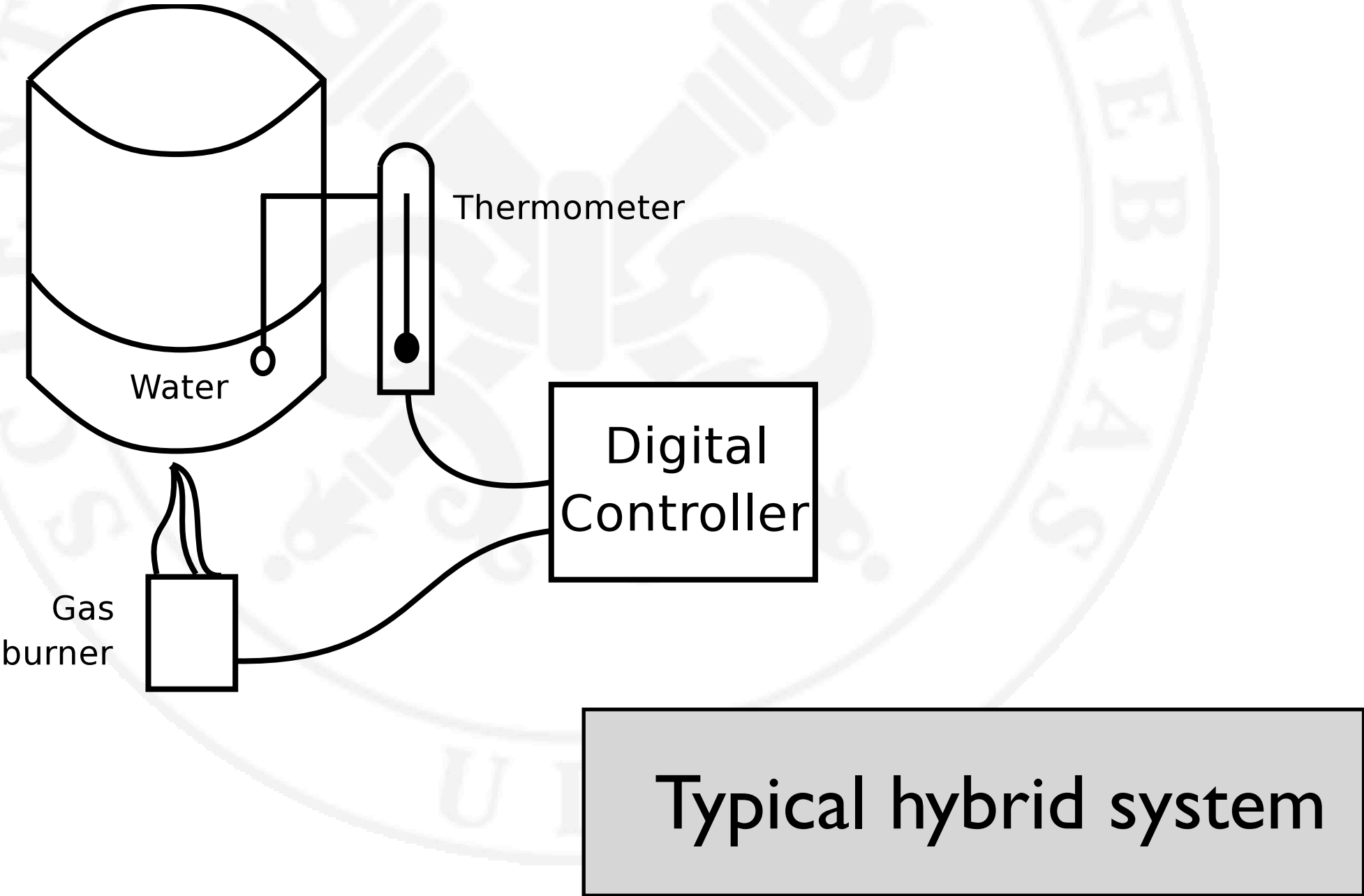
Let $G = \langle Q_1, Q_2, \iota, \delta \rangle$ be a TGS, let $\text{Safe}(G, Q)$ be a **safety** game defined on G , Player I has a winning strategy for this game iff

$$\iota \in \cup \{R \mid R = Q \cap \text{CPre}_1(R)\}$$

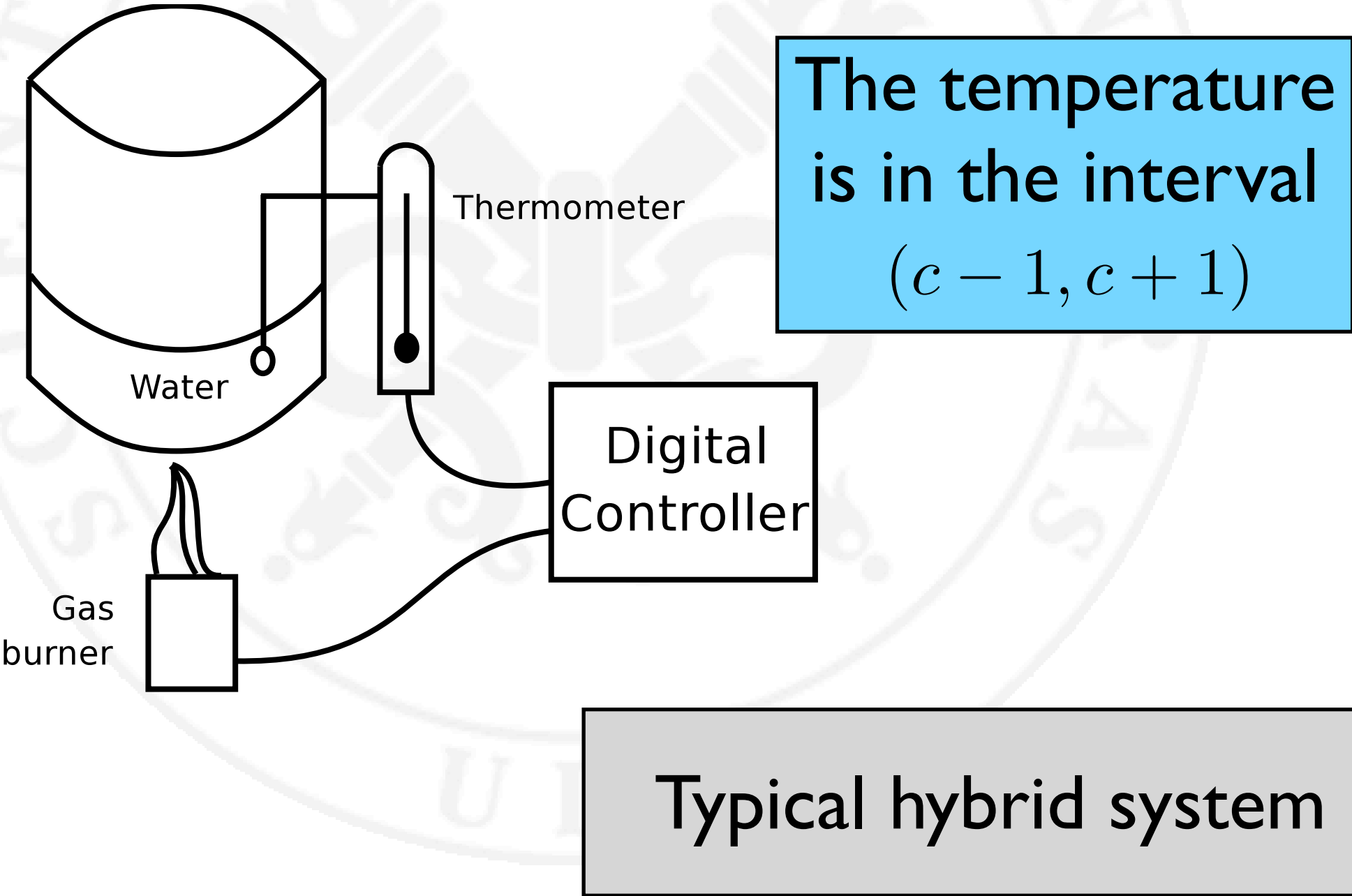
A large, faint, circular watermark of the ULB logo is centered in the background. It features a sunburst in the center, surrounded by the Latin phrase 'E PLURIBUS UNUM' and the letters 'ULB' at the bottom.

Games of imperfect information

Perfect information hypothesis?

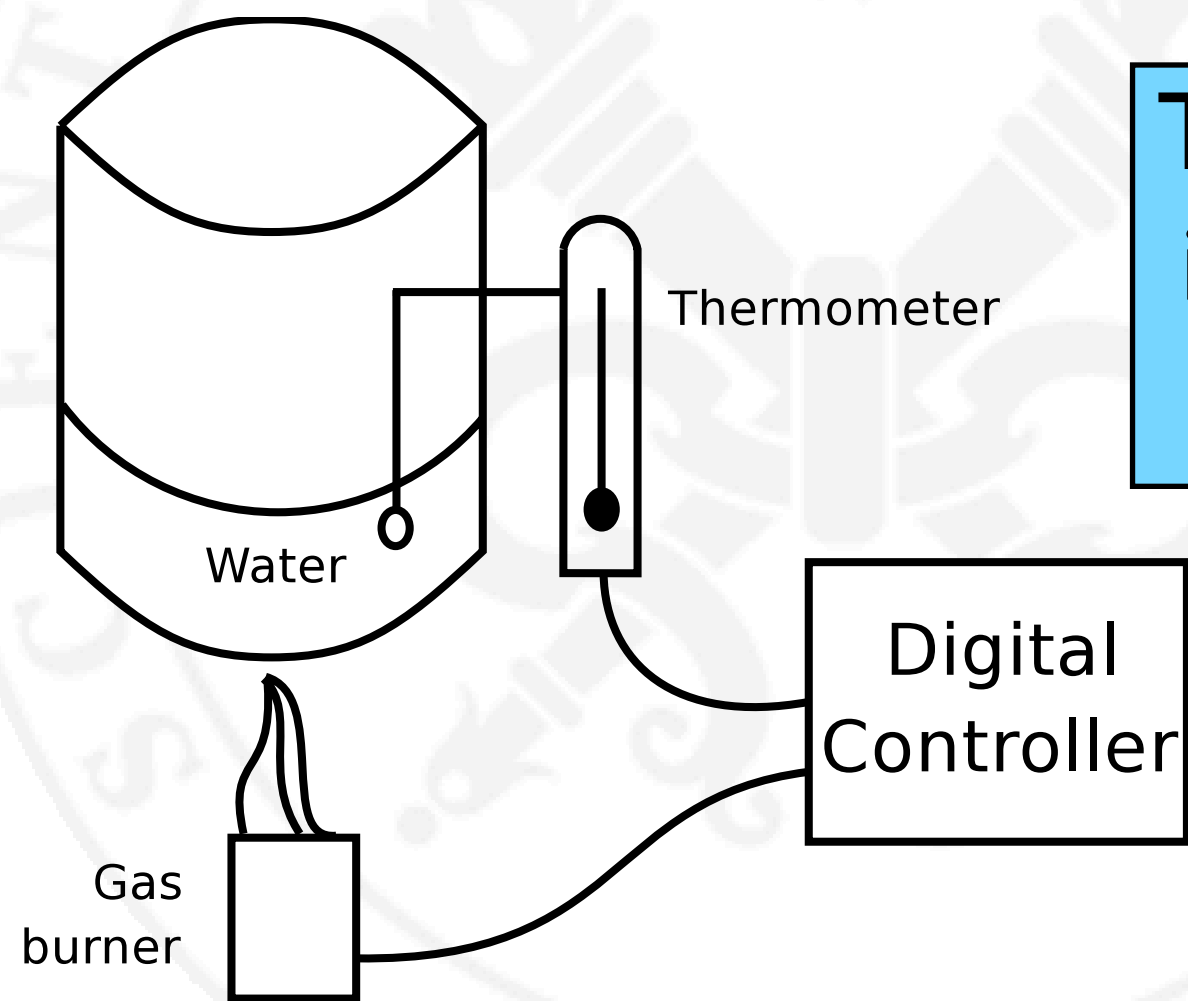


Perfect information hypothesis?



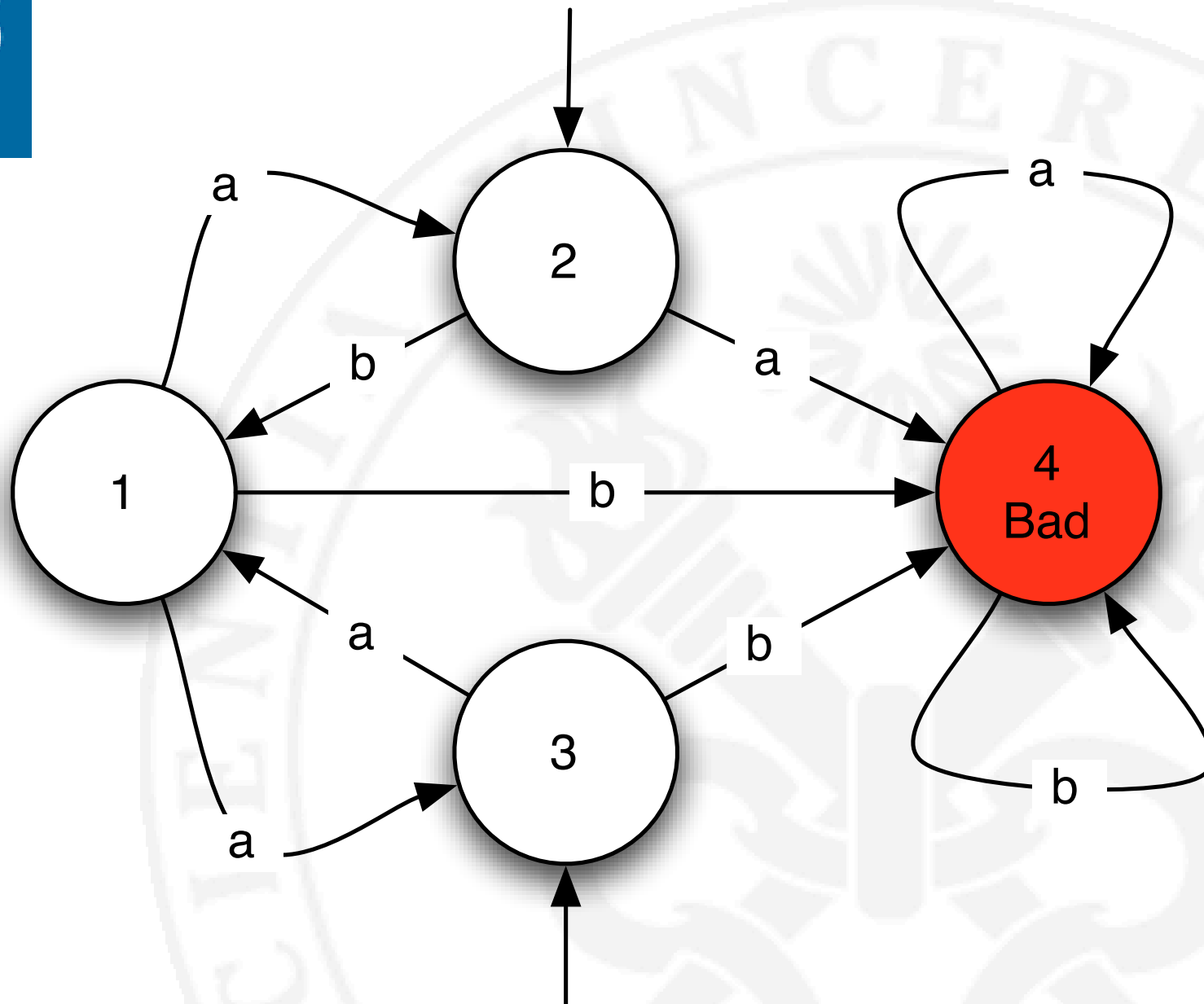
Perfect information hypothesis?

Finite precision = imperfect information

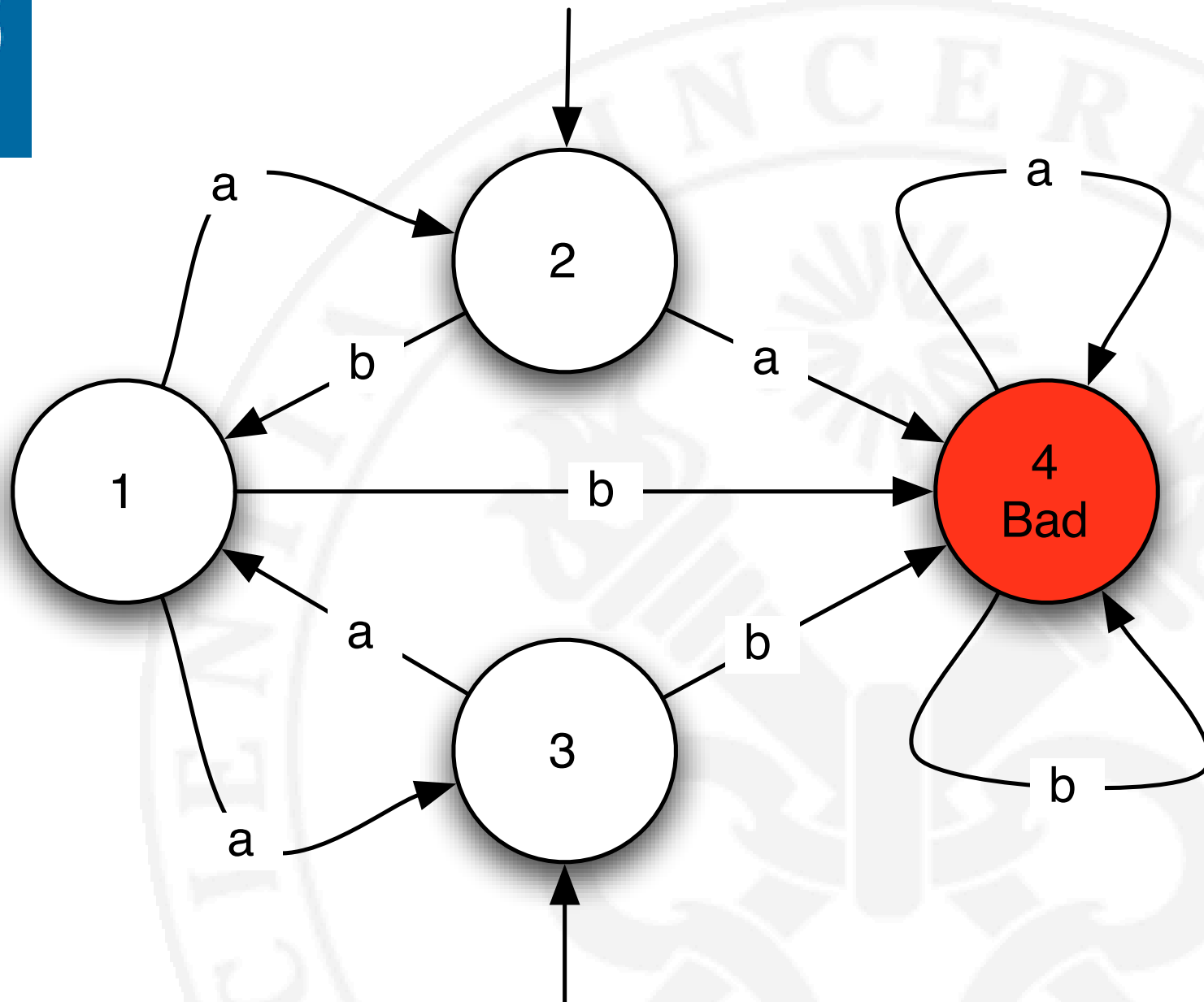


The temperature
is in the interval
 $(c - 1, c + 1)$

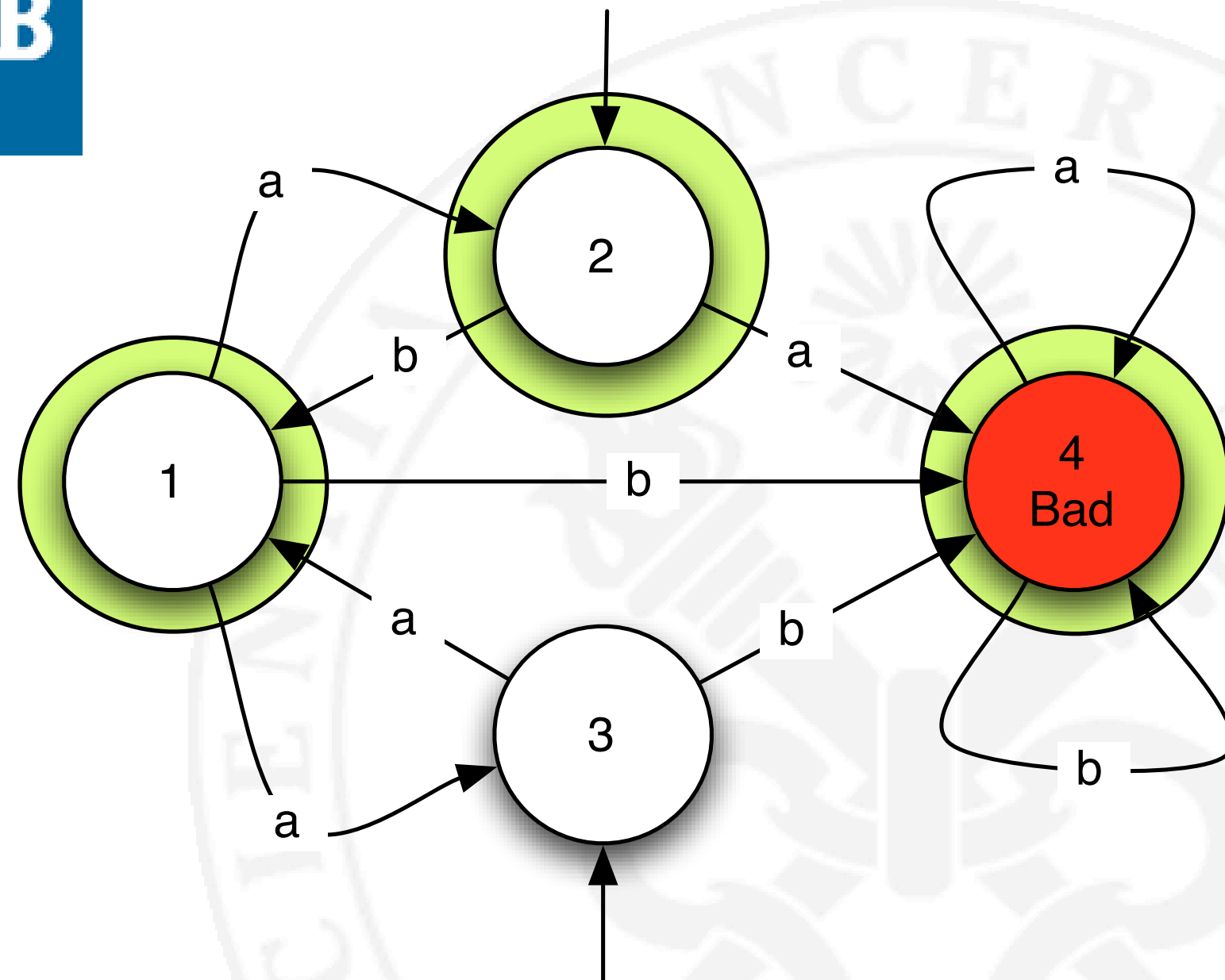
Typical hybrid system



Player 0 chooses a letter
Player 1 resolves nondeterminism

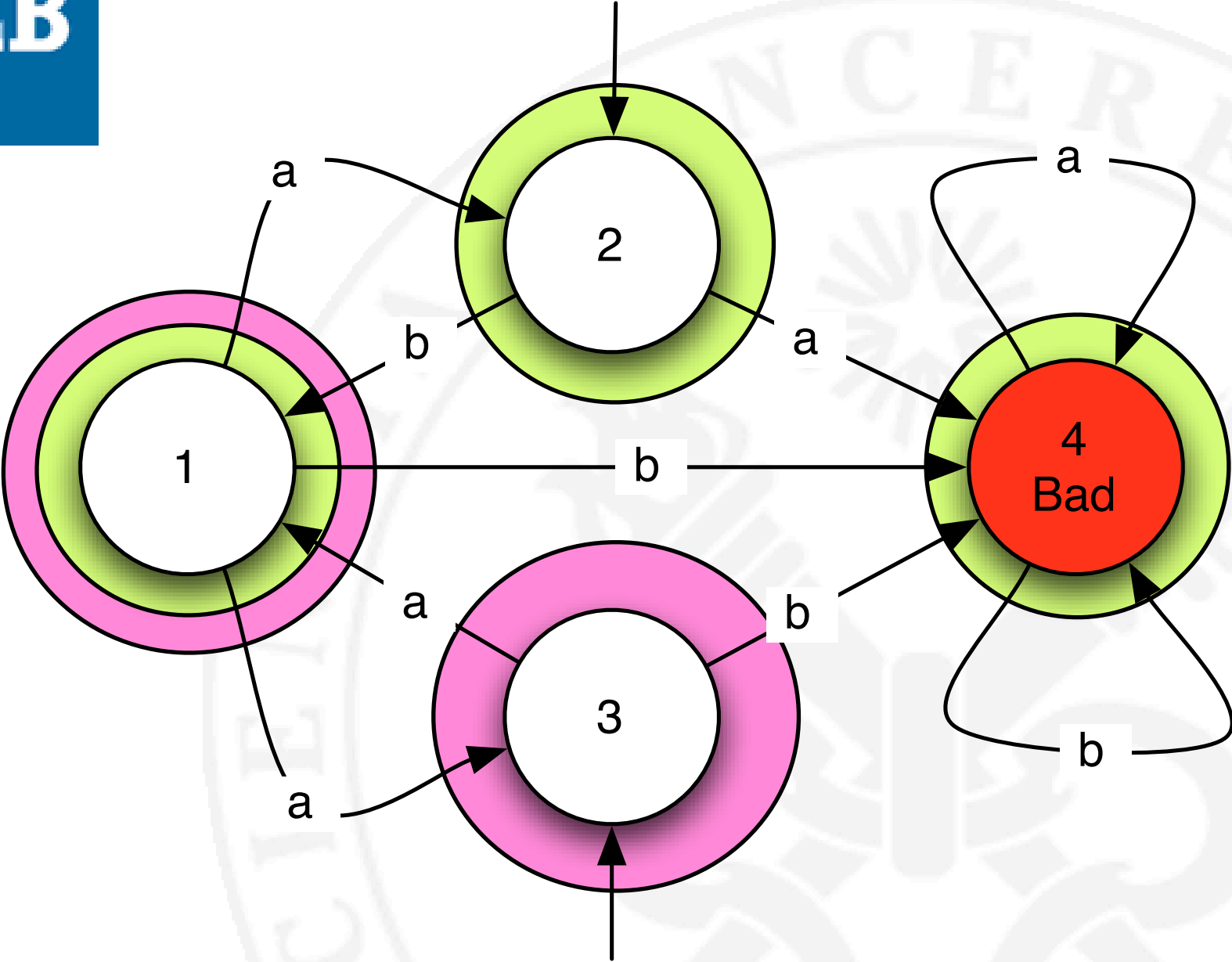


Imperfect information



Obs 0

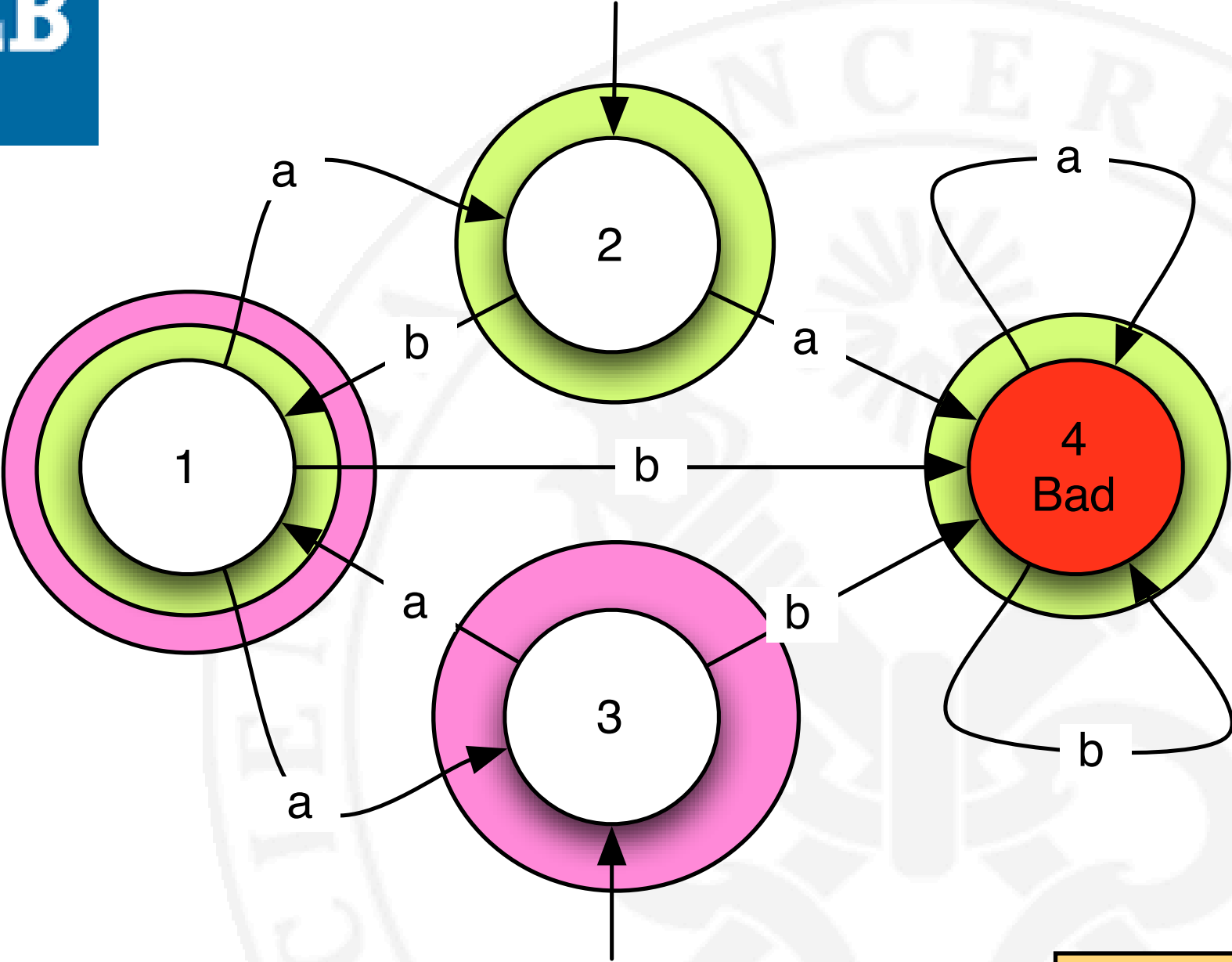
Imperfect information



Obs 0

Obs 1

Imperfect information

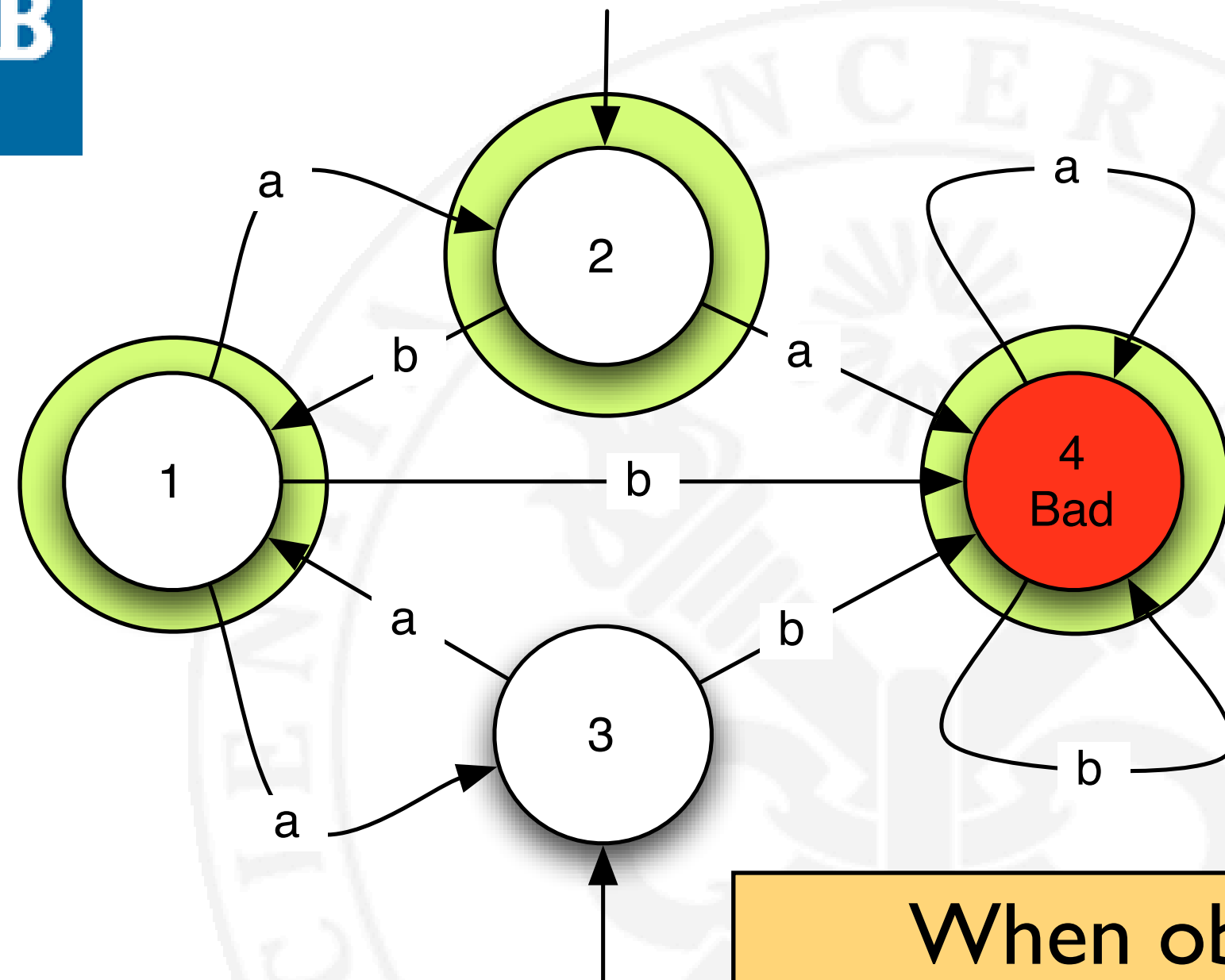


Obs 0

Obs 1

Slight generalization of incomplete information

Imperfect information



Obs 0

When observing Obs 0,
there is no unique good choice:
memory is necessary

Imperfect information

Games / Strategies

- A game of *imperfect information*:
game structure + observation structure
- *Observation structure* : (\mathbf{Obs}, γ) where \mathbf{Obs} is a finite set of observations and γ maps every observation to a set of states (we require that every state has at least one observation).
- A *observation based strategy* is a function that maps every sequence $o_1 \sigma_1 o_2 \dots o_n$ to a letter in Σ .

Our objective is to find an algorithm to construct **observation based strategies** that avoid **Bad**.

Games / Strategies

Notation: a game structure of imperfect information is a tuple $(S, S_0, \Sigma, \rightarrow, \text{Obs}, \gamma)$.

tion structure

where **Obs** is a finite

set of observations and γ maps every observation to a set of states (we require that every state has at least one observation).

-A *observation based strategy* is a function that maps every sequence $o_1 \sigma_1 o_2 \dots o_n$ to a letter in Σ .

Our objective is to find an algorithm to construct **observation based strategies** that avoid **Bad**.

Games / Strategies

Notation: a game structure of imperfect information is a tuple

$(S, \Sigma, \gamma, \text{Obs})$

tion structure

set of
set of
one of

Those games generalize games of *perfect information* where **Obs=S** and γ is the identity function

a finite
ion to a
least

-A *observation based strategy* is a function that maps every sequence $o_1 \sigma_1 o_2 \dots o_n$ to a letter in Σ .

Our objective is to find an algorithm to construct **observation based strategies** that avoid **Bad**.

Games / Strategies

Notation: a game structure of imperfect information is a tuple

$(S, \Sigma, \gamma, \text{Obs})$

tion structure

Those games generalize games

a finite
ion to a

set of
set of
one of
when
iden

Those games generalize games
of *incomplete information*:

-A *observation*
every sequ

in that case **Obs partitions**
the state space S . [Rei84]

Our objective is to find an algorithm to construct
observation based strategies that avoid *Bad*.

Classical Approaches

- To solve games of perfect information :
 - (elegant) fixed point algorithms using a **controllable predecessor** operator
- To solve games of imperfect information
 - [Reif84] builds a game of perfect information using a knowledge-based **subset construction and** then solve this games using classical techniques

Classical Approaches

- To solve games of imperfect information
 - (elementary) After a finite prefix of a game, Player I has a partial knowledge of the current state of the game : **a set of states**
 - To solve games of imperfect information
 - [Reif84] builds a game of perfect information using a knowledge-based **subset construction** and then solve this games using classical techniques

Classical Approaches

- To solve After a finite prefix of a game, Player I has a partial knowledge of the current state of the game : **a set of states**
- (elementary)

We propose here a new solution that avoid the **preliminary** explicit subset construction.

subset construction and then solve this games using classical techniques

A fixed point algorithm

We define a *controllable predecessor* operator for a **set of sets of states** q

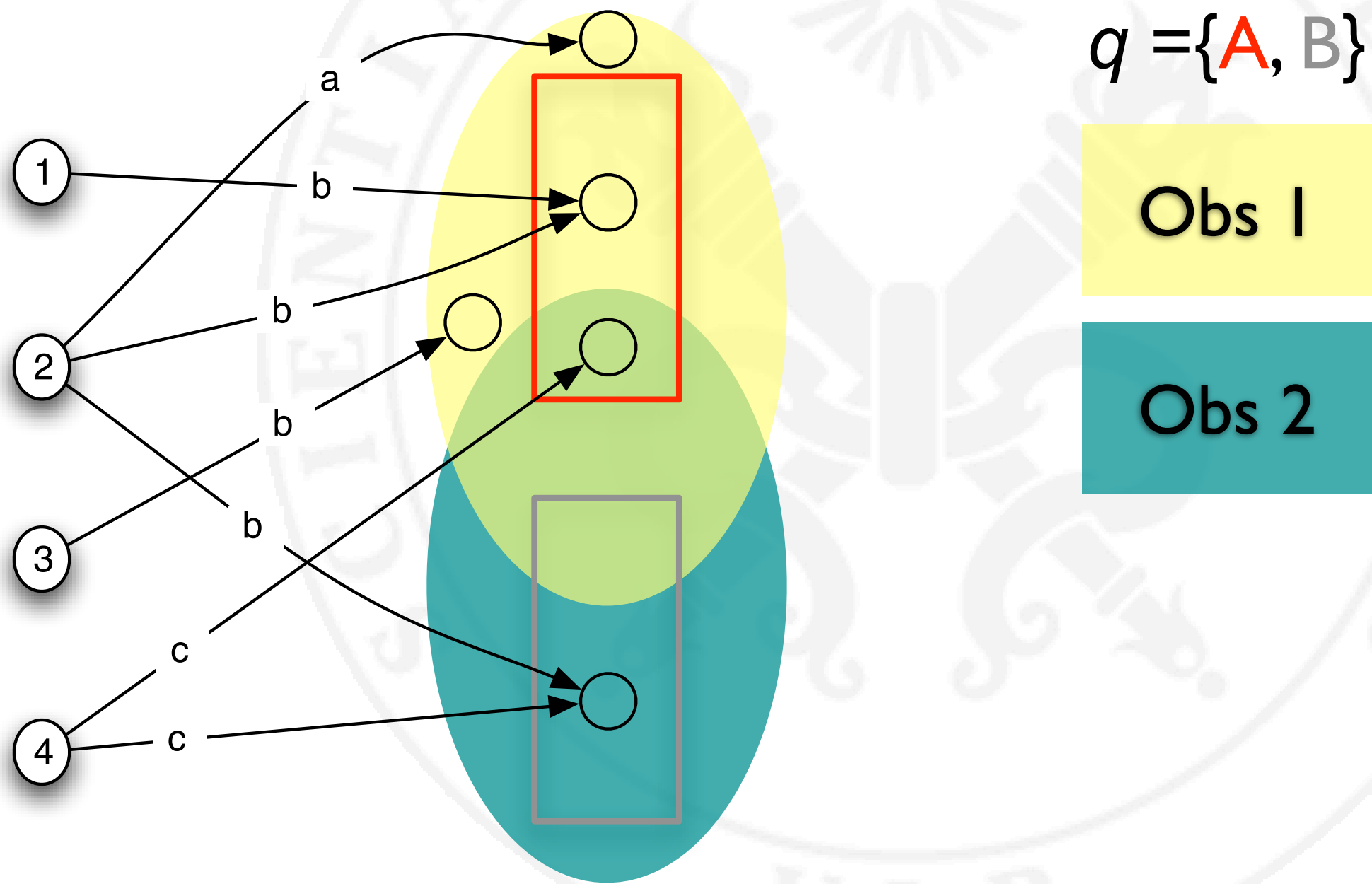
$$\text{CPre}(q) = \{s \subseteq \overline{\text{Bad}} \mid \exists \sigma \in \Sigma \cdot \forall \text{obs} \in \text{Obs} \cdot \exists s' \in q : \text{Post}_\sigma(s) \cap \gamma(\text{obs}) \subseteq s'\}$$

(i) s does not intersect with **Bad**,

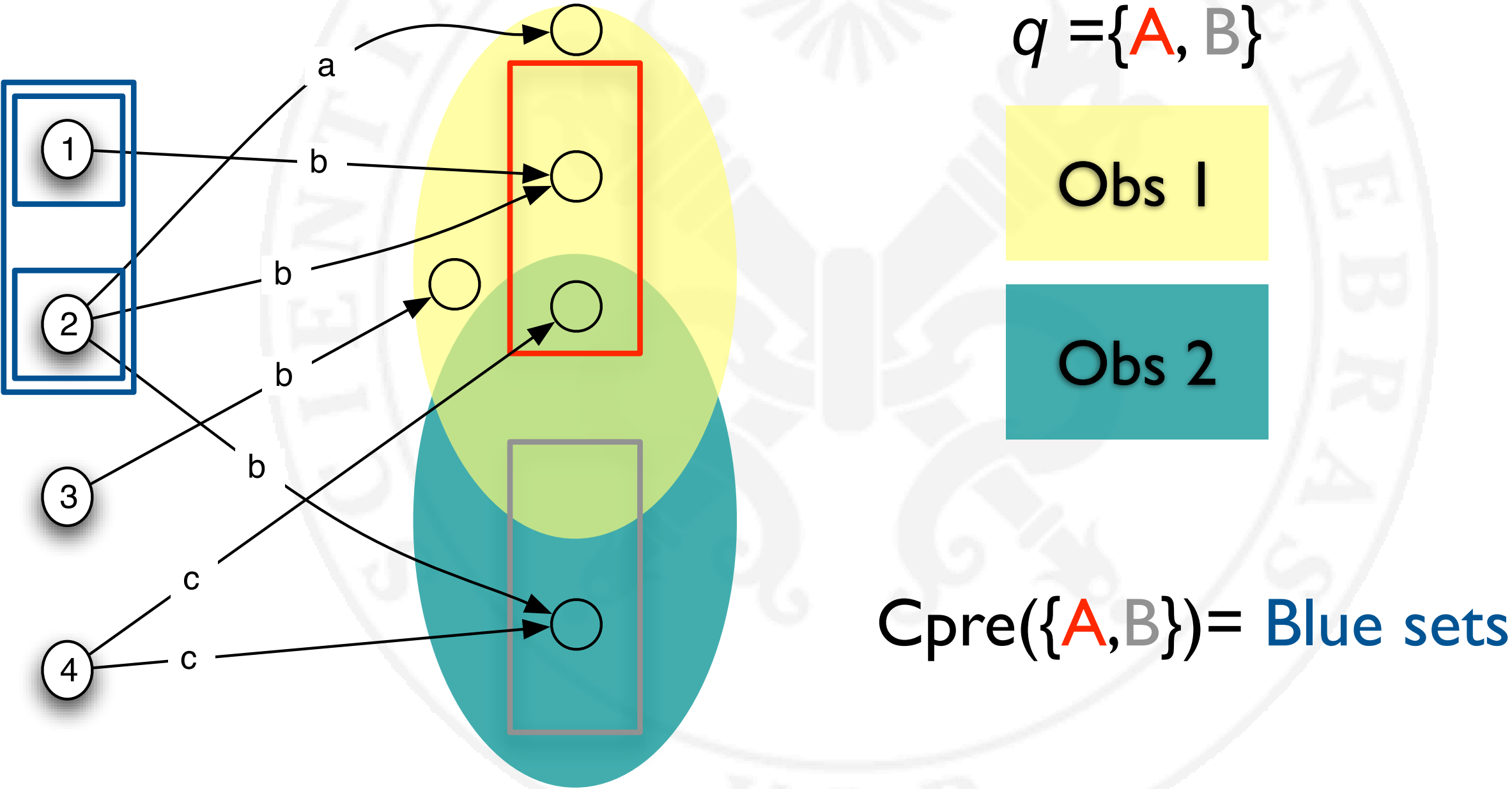
(ii) there exists σ s.t. the set of possible successors of s by σ is covered by q

- (a)** no matter how the adversary resolves non-determinism,
- (b)** no matter the compatible observation **Obs**

Example



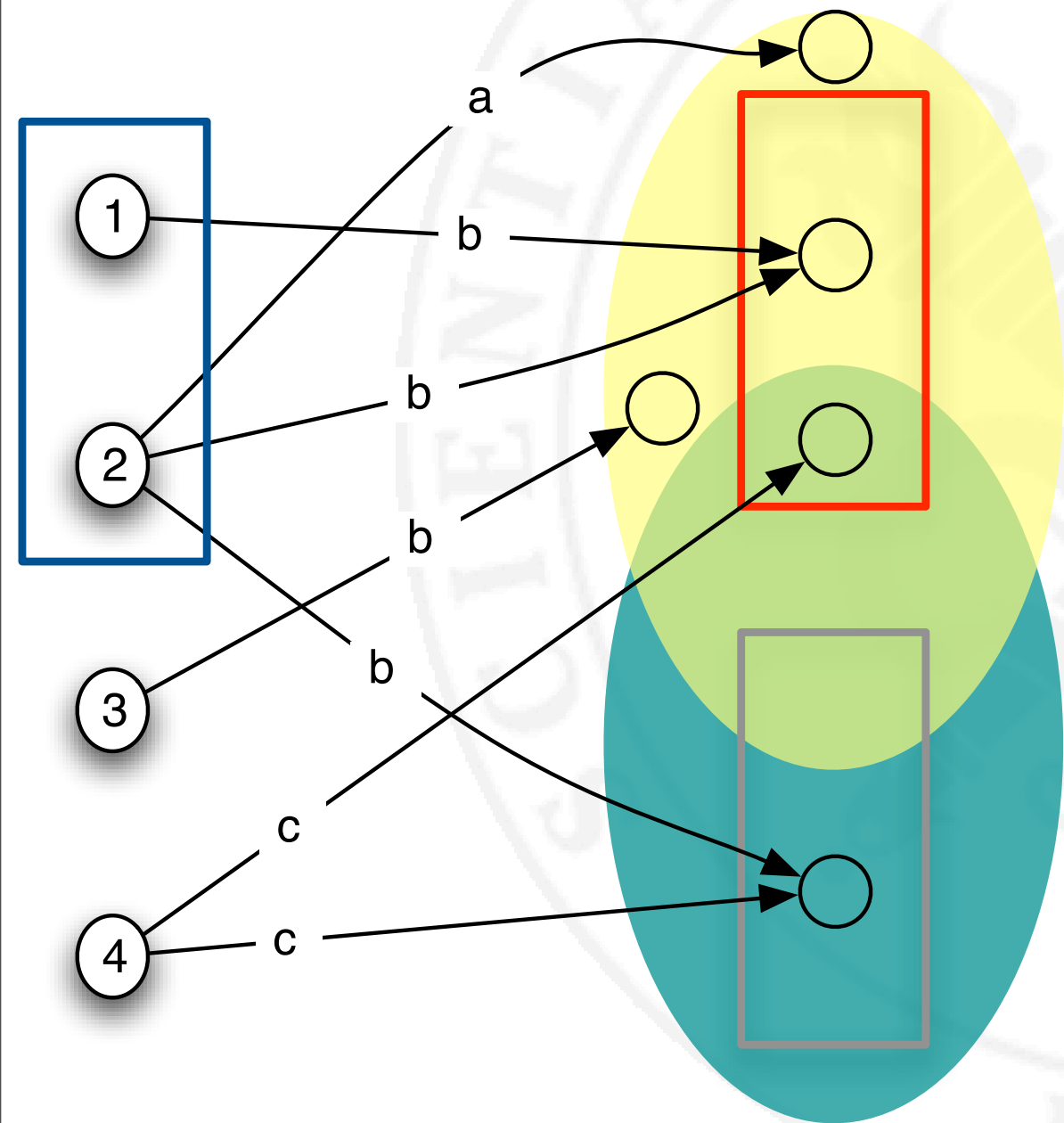
Example



Maximal sets

If there is a strategy for set A,
there is a strategy for any B included in A

It is enough to keep only
the **maximal sets**



$$\text{CPre}(q) = \left[\{s \subseteq \overline{\text{Bad}} \mid \exists \sigma \in \Sigma \cdot \forall \text{obs} \in \text{Obs} \cdot \exists s' \in q : \text{Post}_\sigma(s) \cap \gamma(\text{obs}) \subseteq s'\} \right]$$

Antichains

Definition 4 [Antichain of sets of states] An *antichain* on the partially ordered set $\langle 2^S, \subseteq \rangle$ is a set $q \subseteq 2^S$ such that for any $A, B \in q$ we have $A \not\subseteq B$.

Let us call L the set of antichains on S .

Definition 5 [\sqsubseteq] Let $q, q' \in 2^{2^S}$ and define $q \sqsubseteq q'$ if and only if

$$\forall A \in q : \exists A' \in q' : A \subseteq A'$$

$$\text{lub} : q_1 \sqcup q_2 = \lceil \{s \mid s \in q_1 \vee s \in q_2\} \rceil$$

$$\text{glb} : q_1 \sqcap q_2 = \lceil \{s_1 \cap s_2 \mid s_1 \in q_1 \wedge s_2 \in q_2\} \rceil$$

The minimal element is \emptyset , the maximal element $\{S\}$.

$\langle L, \sqsubseteq \rangle$ is a complete lattice.

CPre over antichains

$$\text{CPre}(q) = [\{s \subseteq \overline{\text{Bad}} \mid \exists \sigma \in \Sigma \cdot \forall \text{obs} \in \text{Obs} \cdot \exists s' \in q : \text{Post}_\sigma(s) \cap \gamma(\text{obs}) \subseteq s'\}]$$

- CPre is a **monotone** function over the lattice of antichains
- CPre has a *least* and a *greatest* fixed point

Advantage : we only keep the needed information to find a strategy

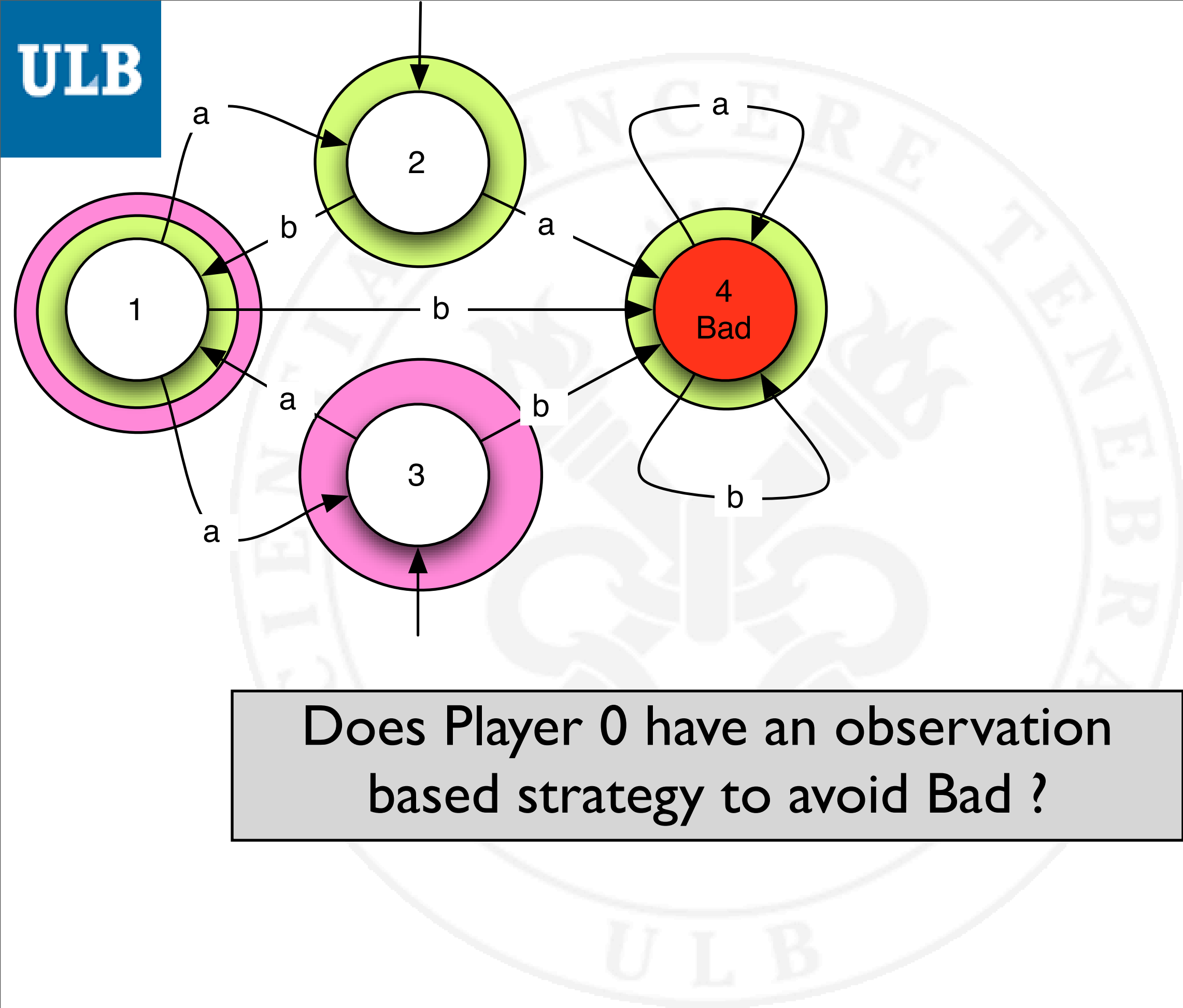
Main theorem

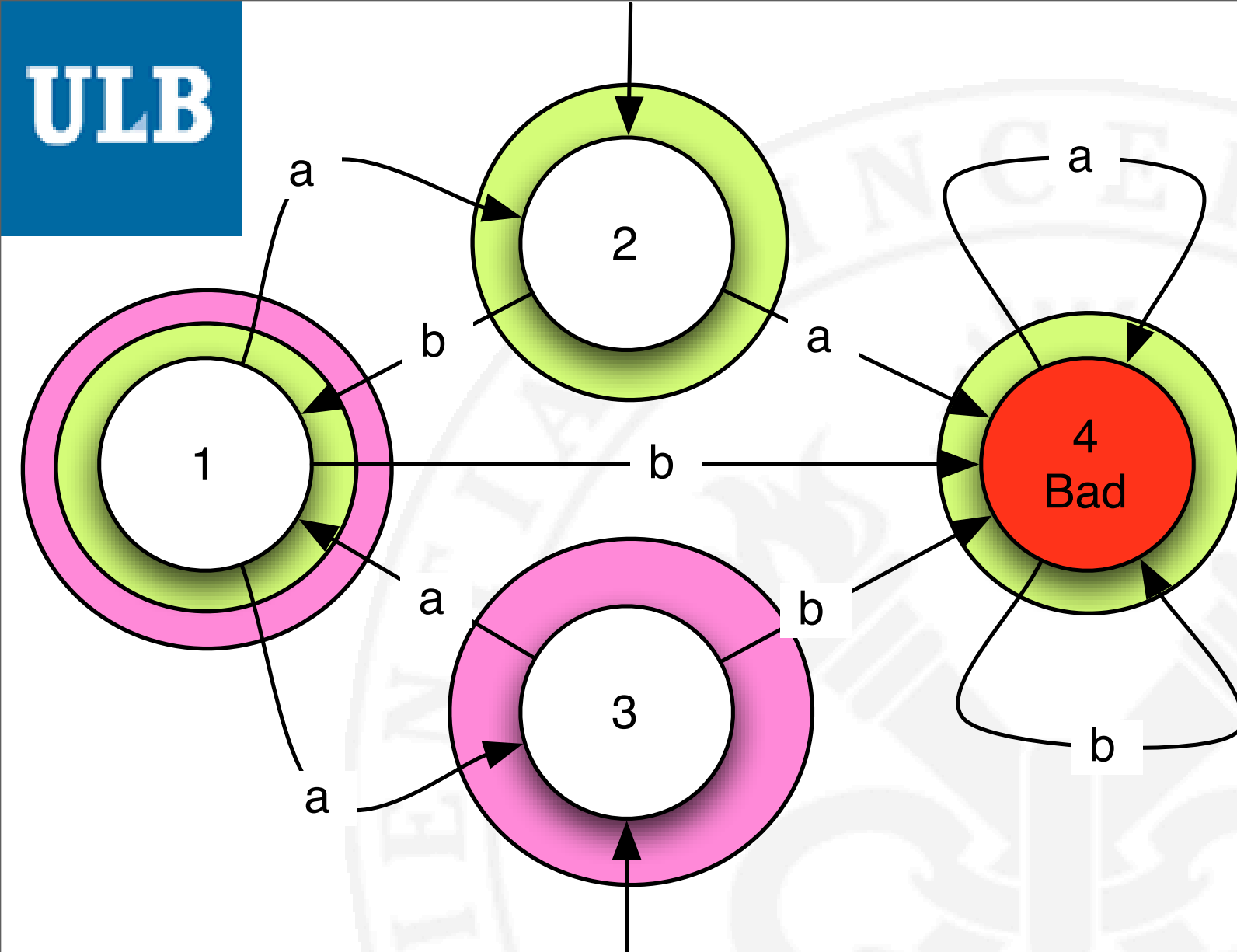
Let $G = \langle S, S_0, \Sigma, \rightarrow, \text{Obs}, \gamma \rangle$

be a two-player game of imperfect information. Player I has a winning observation based strategy to avoid Bad, **iff**

$$\{S_0 \cap \gamma(\text{obs}) \mid \text{obs} \in \text{Obs}\} \sqsubseteq \bigsqcup \{q \mid q = \text{CPre}(q)\}.$$

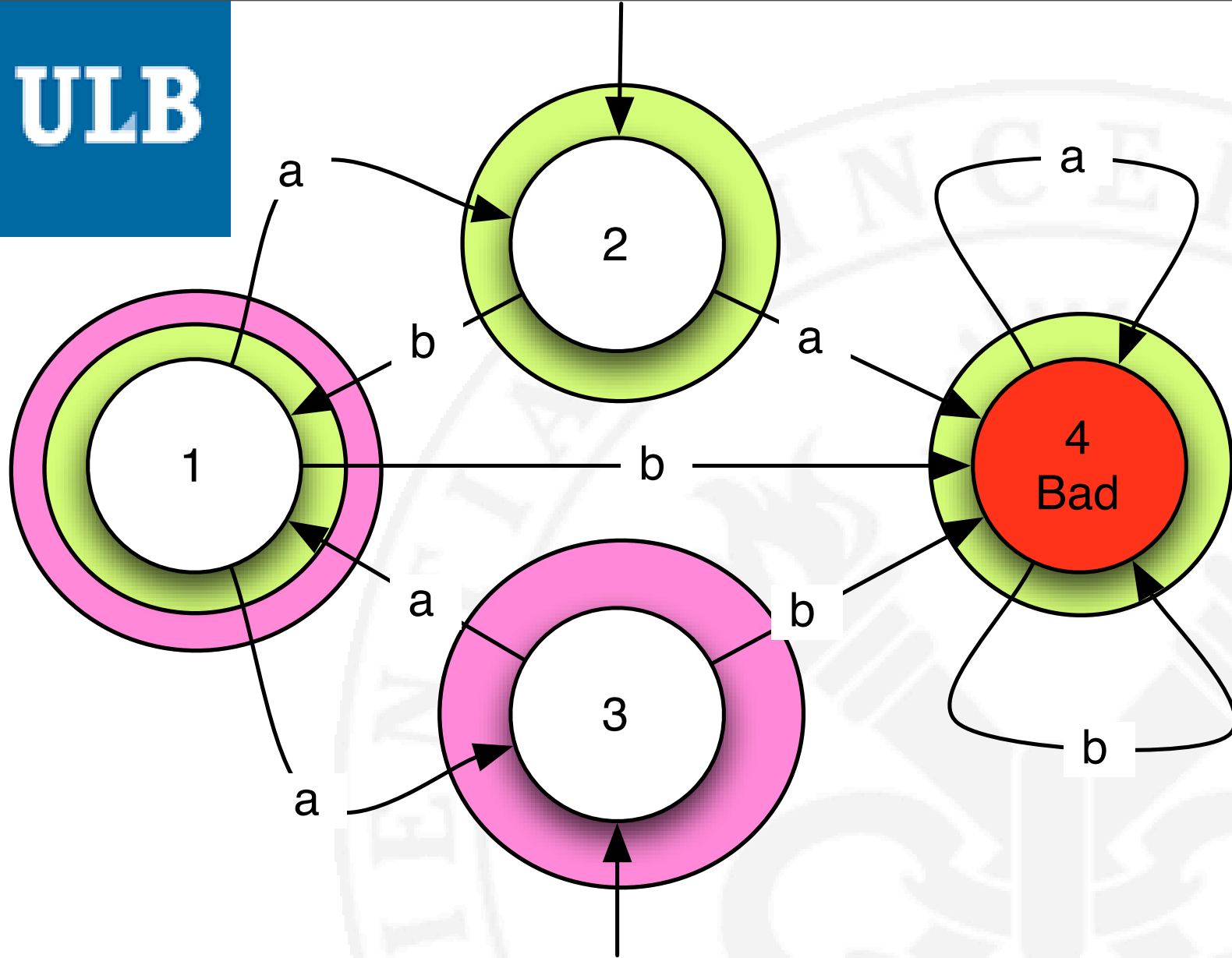
We can extract a strategy from the fixed point





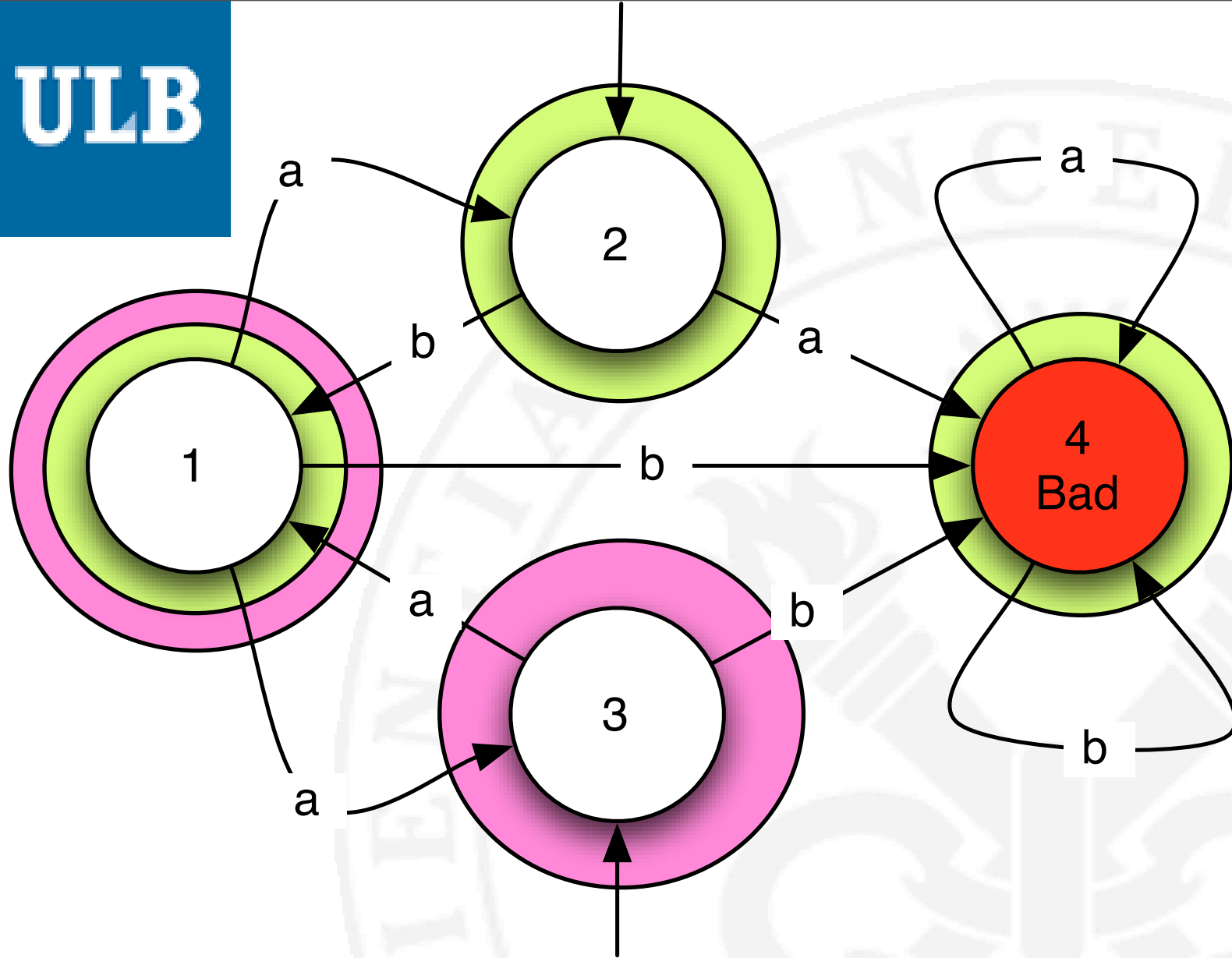
Does Player 0 have an observation based strategy to avoid Bad ?

Let us compute the *gfp* of CPre over L.



$$q_0 = \top$$

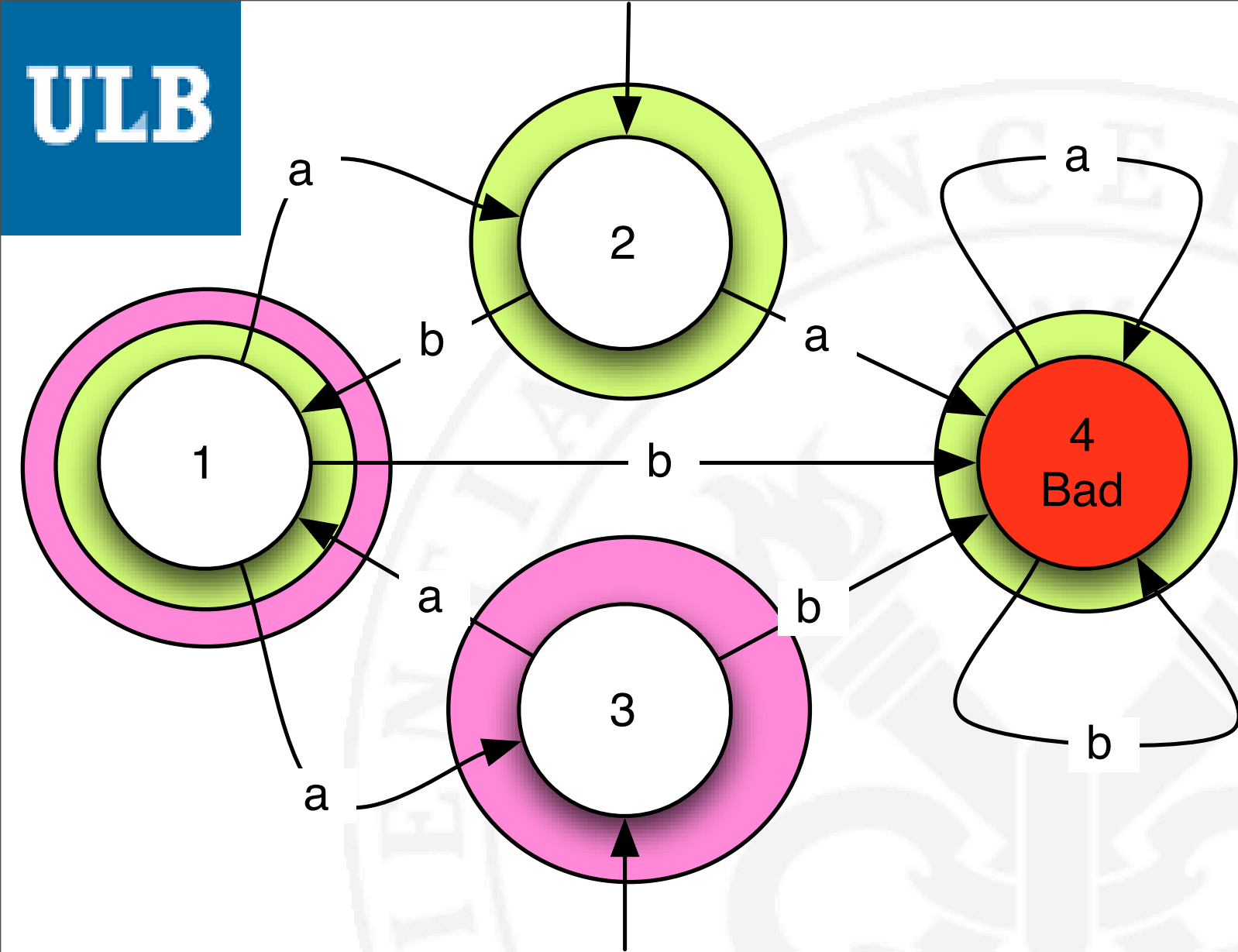
$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$



$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

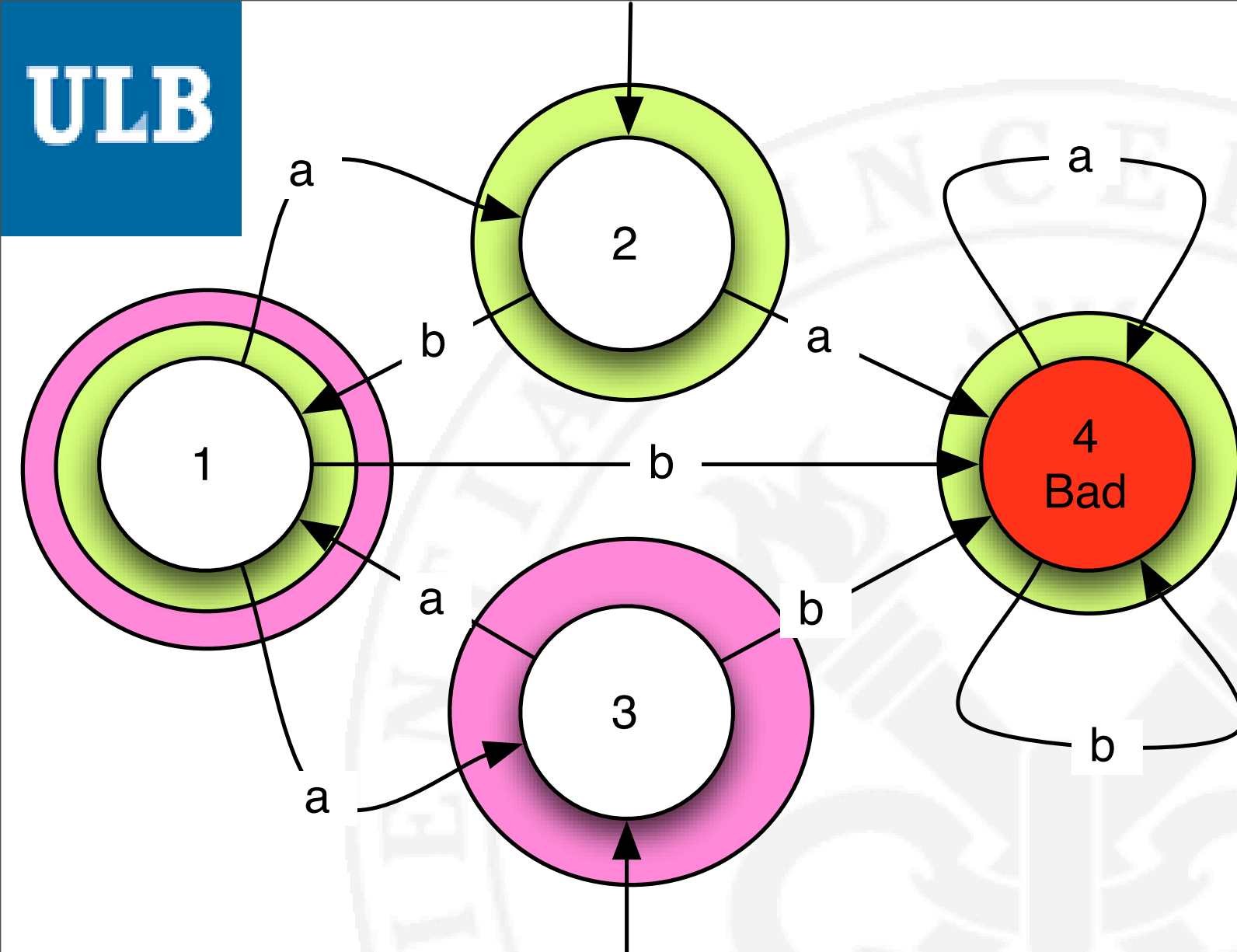
$$q_2 = \text{CPre}(\{\{1, 2, 3\}\})$$



$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$\begin{aligned} q_2 &= \text{CPre}(\{\{1, 2, 3\}\}) \\ &= \{\{2\}_b, \{1, 3\}_a\} \end{aligned}$$



$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$\begin{aligned} q_2 &= \text{CPre}(\{\{1, 2, 3\}\}) \\ &= \{\{2\}_b, \{1, 3\}_a\} \end{aligned}$$

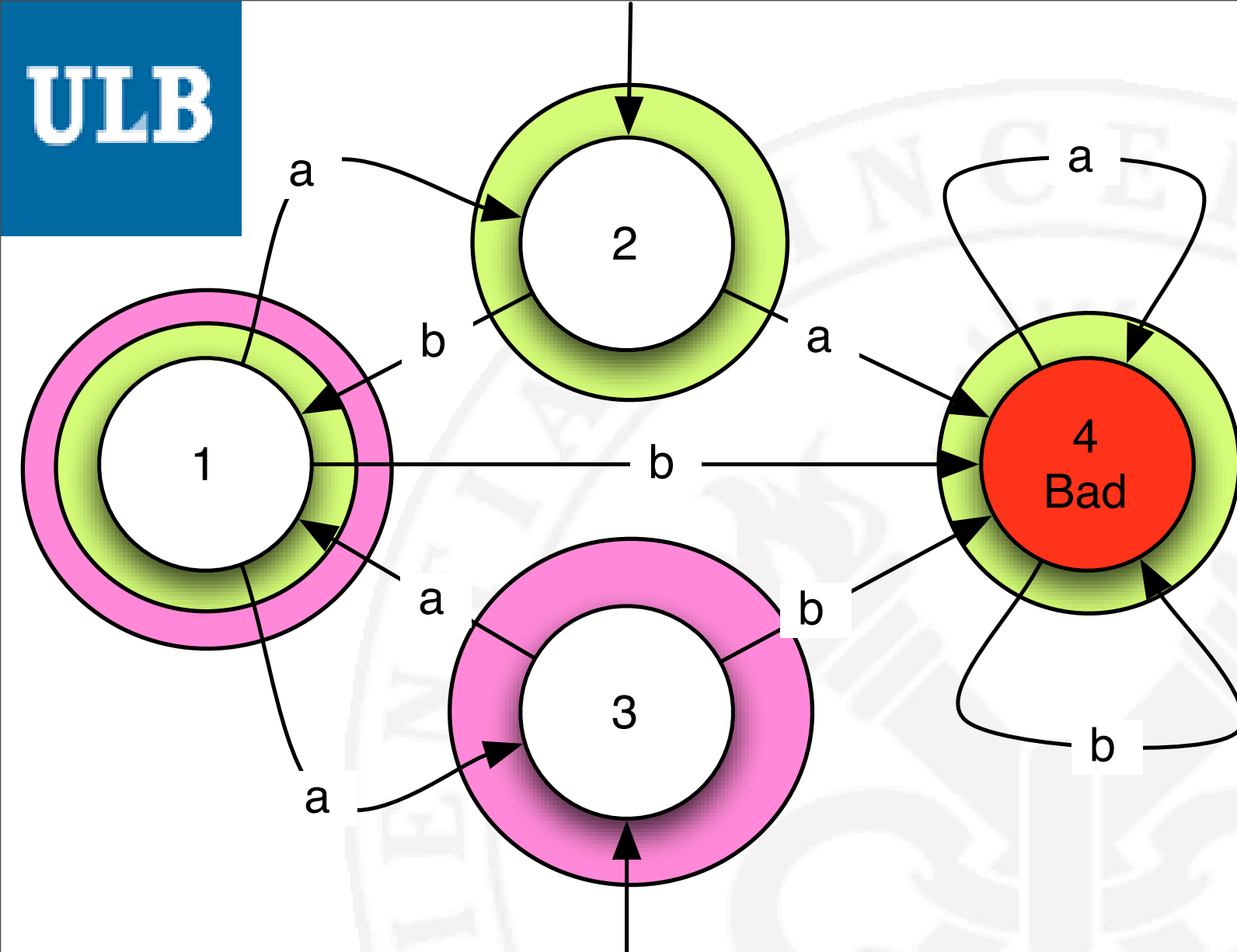
Indeed,

$$\text{Post}_a(\{1, 3\}) \cap \{1, 2, 4\} \subseteq \{1, 2, 3\}$$

$$\text{Post}_a(\{1, 3\}) \cap \{1, 3\} \subseteq \{1, 2, 3\}$$

$$\text{Post}_b(\{2\}) \cap \{1, 3\} \subseteq \{1, 2, 3\}$$

$$\text{Post}_b(\{2\}) \cap \{1, 2, 4\} \subseteq \{1, 2, 3\}$$

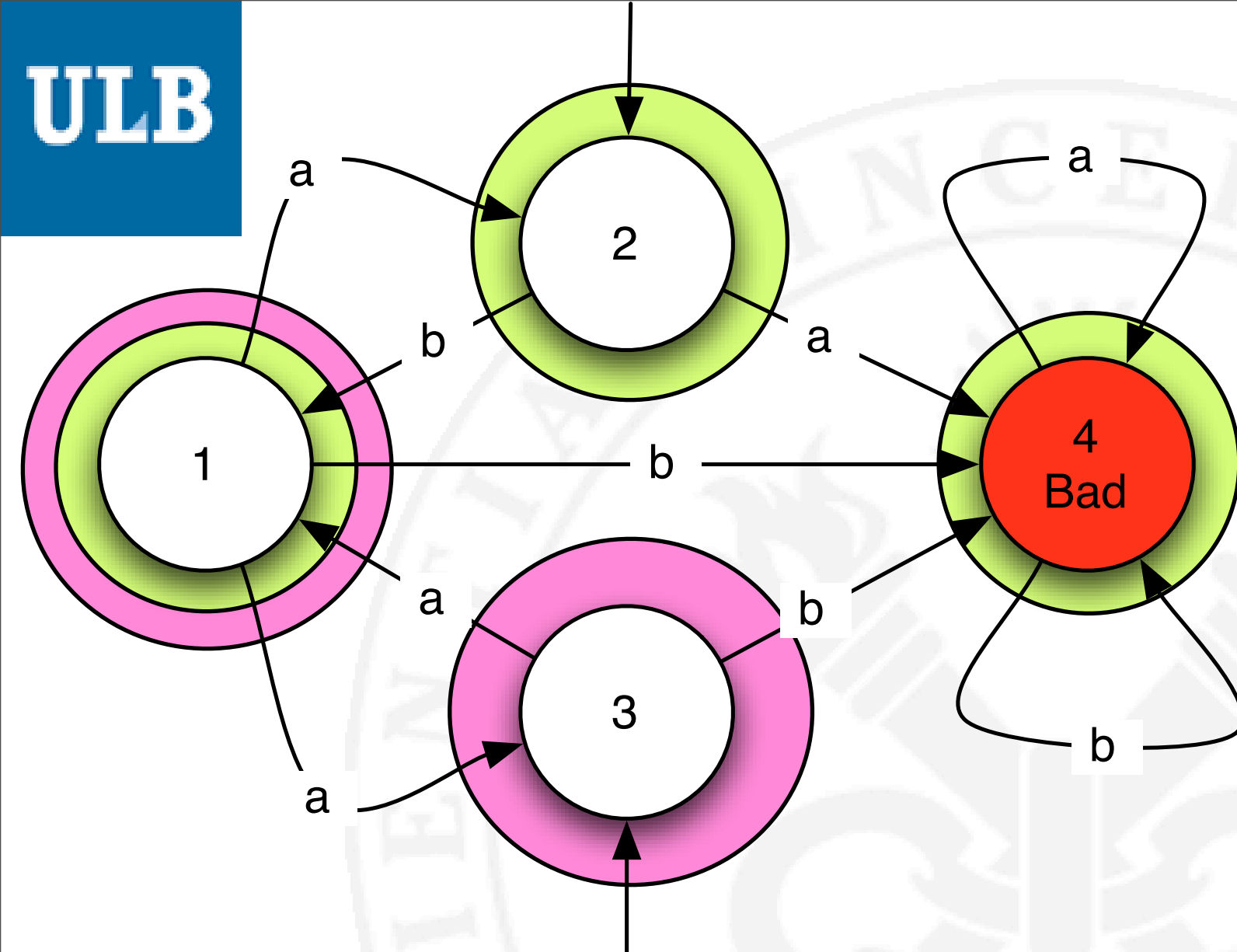


$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

$$q_3 = \text{CPre}(\{\{2\}, \{1, 3\}\})$$

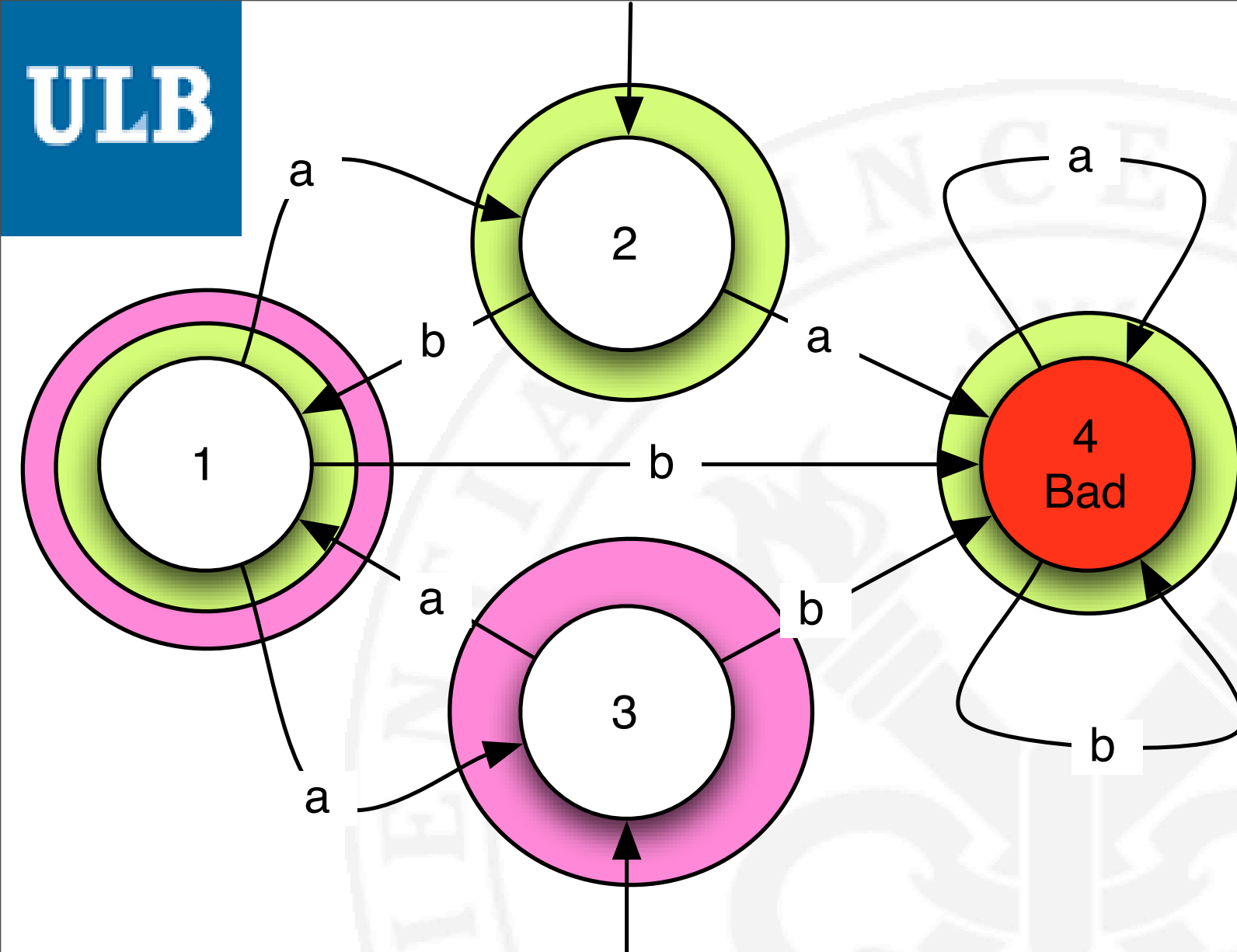


$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

$$\begin{aligned} q_3 &= \text{CPre}(\{\{2\}, \{1, 3\}\}) \\ &= \{\{1\}_a, \{2\}_b, \{3\}_a\} \end{aligned}$$



$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

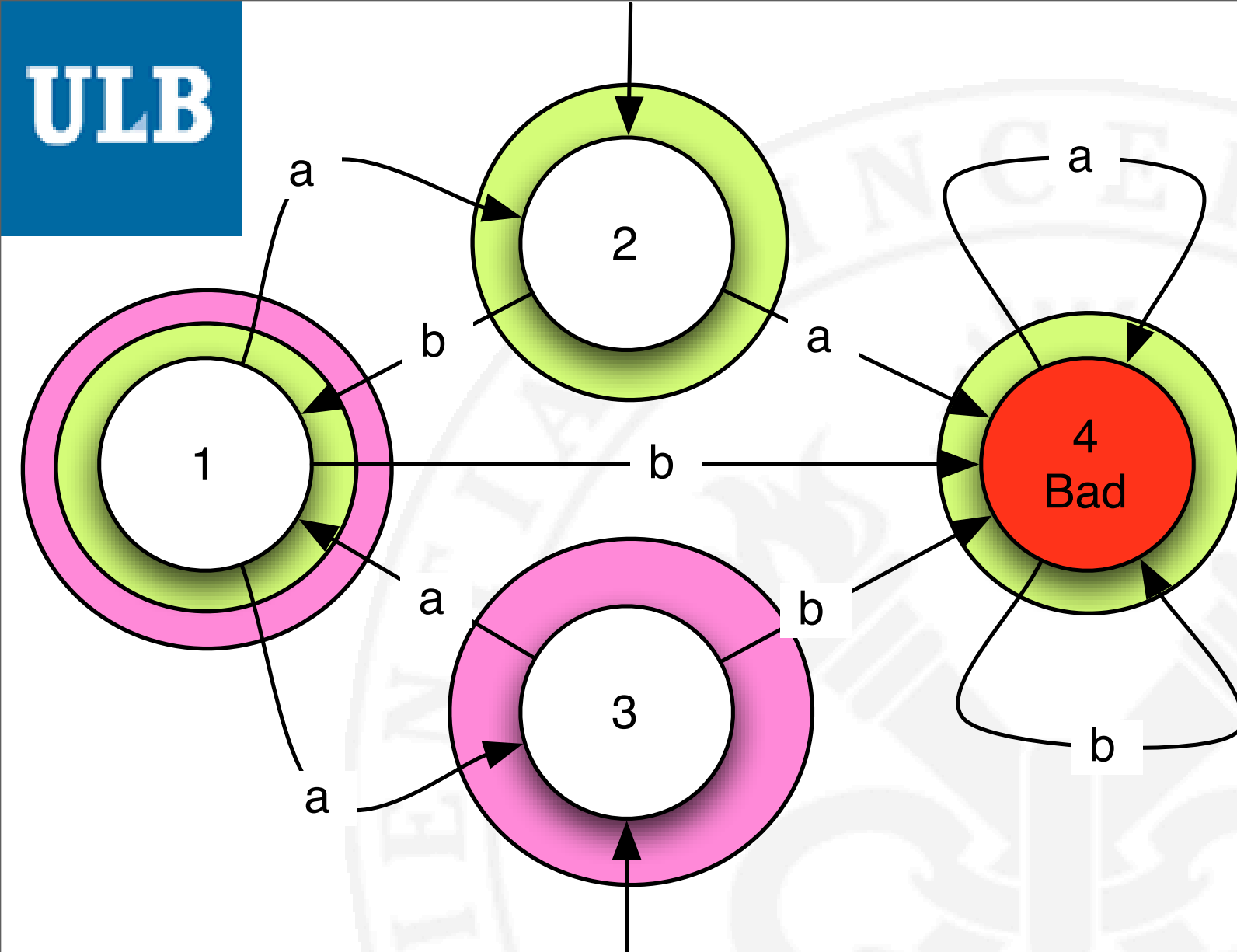
$$\begin{aligned} q_3 &= \text{CPre}(\{\{2\}, \{1, 3\}\}) \\ &= \{\{1\}_a, \{2\}_b, \{3\}_a\} \end{aligned}$$

Indeed,

$$\text{Post}_a(\{1\}) \cap \{1, 2, 4\} \subseteq \{2\}$$

$$\text{Post}_a(\{1\}) \cap \{1, 3\} \subseteq \{3\}$$

Adding any state would break this property



$$q_0 = \top$$

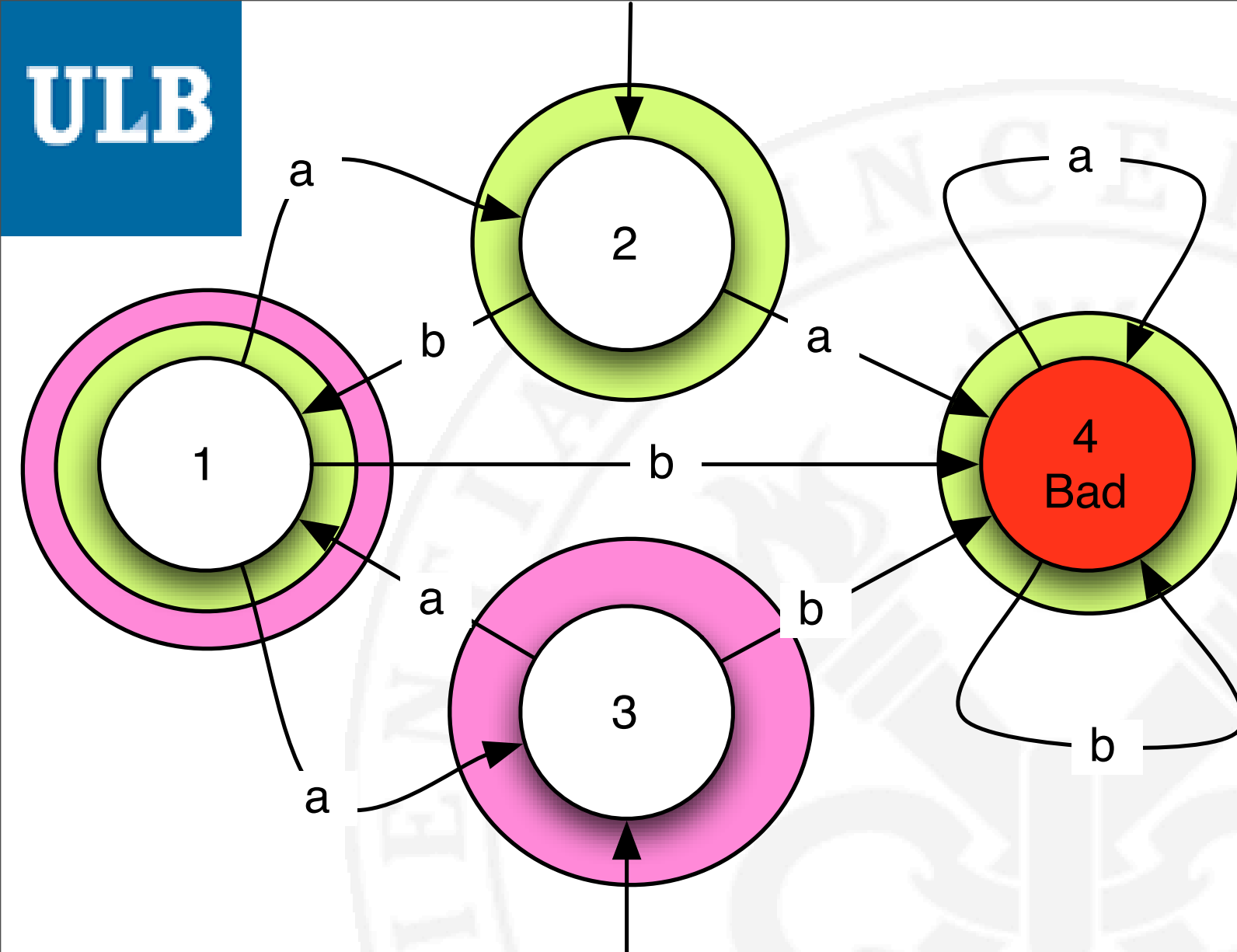
$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

$$q_3 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

$$q_4 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

Fixed point



$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

$$q_3 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

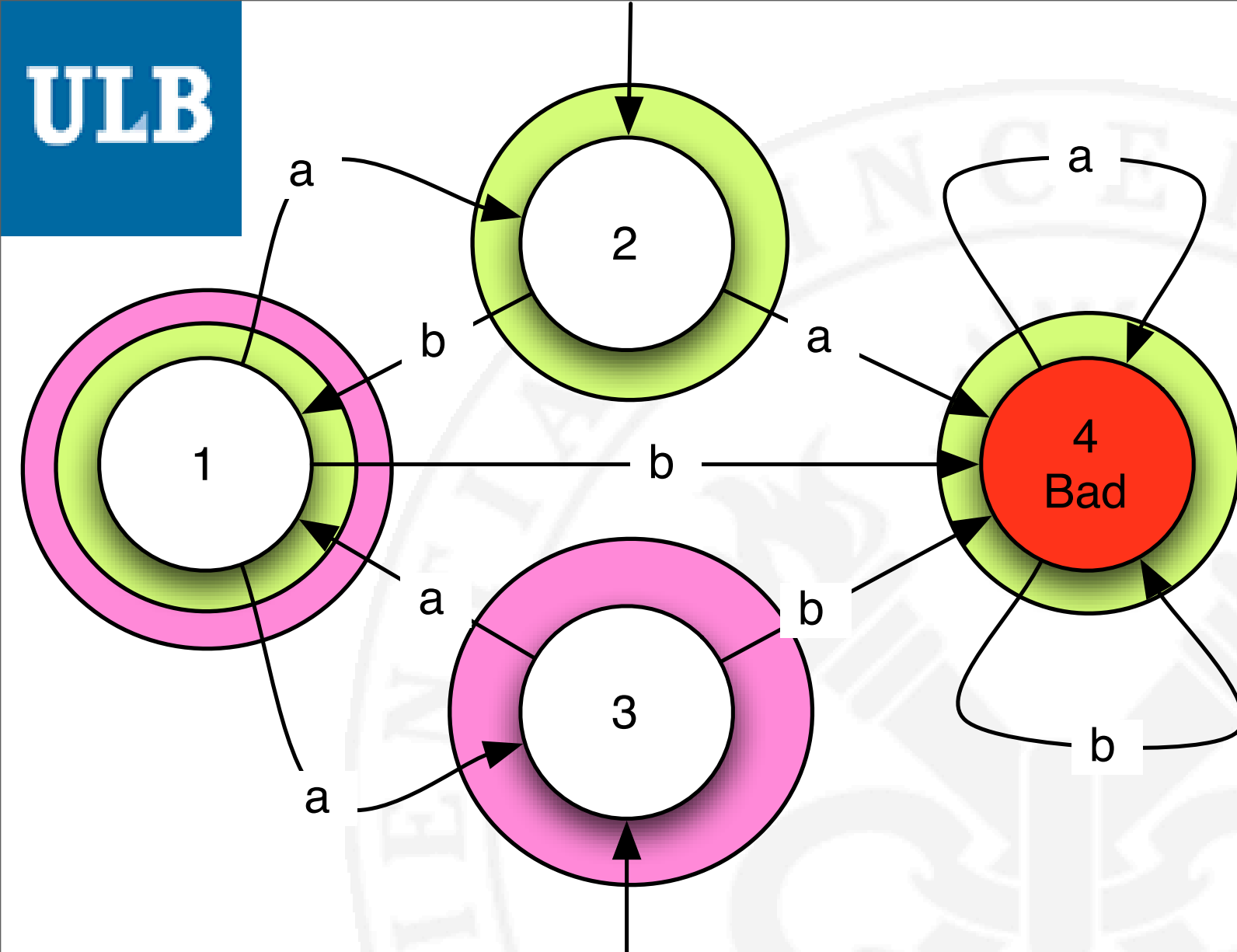
$$q_4 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

Fixed point

We have

$$\{\{2, 3\} \cap \text{Obs}_0, \{2, 3\} \cap \text{Obs}_1\} \sqsubseteq \sqcup \{q \mid q = \text{CPre}(q)\}$$

and so, Player 0 has an observation based winning strategy to avoid Bad



$$q_0 = \top$$

$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

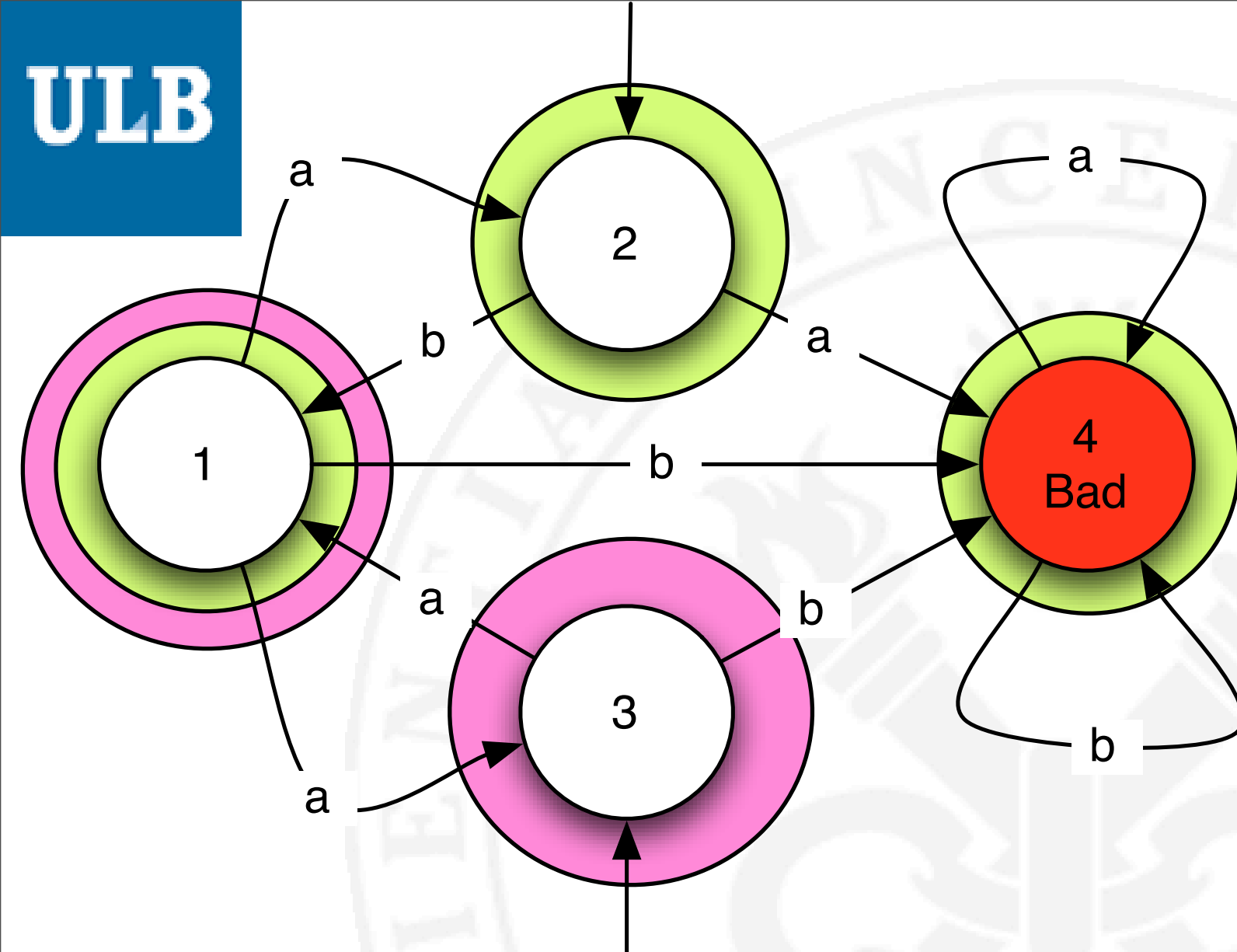
$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

$$q_3 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

$$q_4 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

Fixed point

We can extract a strategy from the fixed point



$$q_0 = \top$$

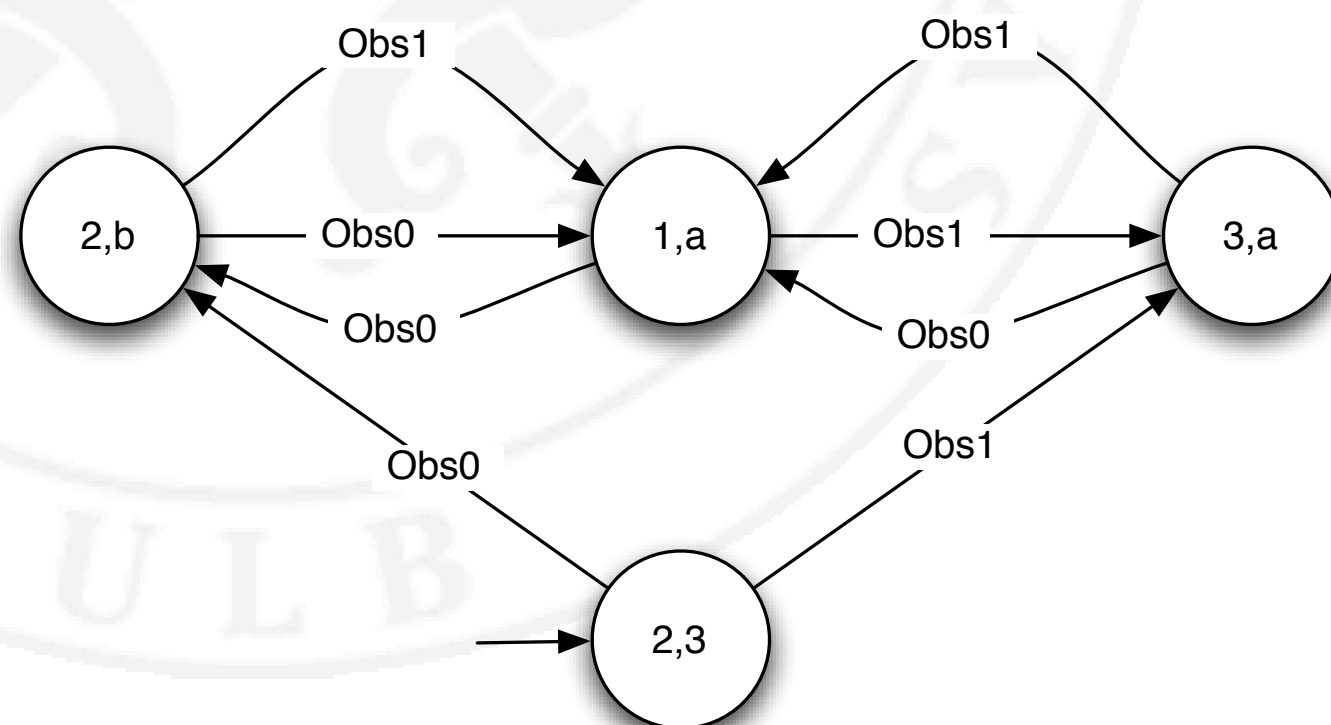
$$q_1 = \{\{1, 2, 3\}_{a,b}\}$$

$$q_2 = \{\{2\}_b, \{1, 3\}_a\}$$

$$q_3 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

$$q_4 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

Fixed point



Complexity for finite state games

- The imperfect information control problem is *EXPTIME-complete*
- There exist finite state games of incomplete information for which the algorithm of [Rei84] requires an exponential time where our algorithm needs only polynomial time

Complexity for finite state games

- The imperfect information control problem is *EXPTIME-complete*
- There exist finite state games of incomplete information for which the algorithm of [Rei84] requires a
our algorithm needs

We compute exactly what is needed to control the system for a given objective

Infinite state games

We drop the assumption that S is finite

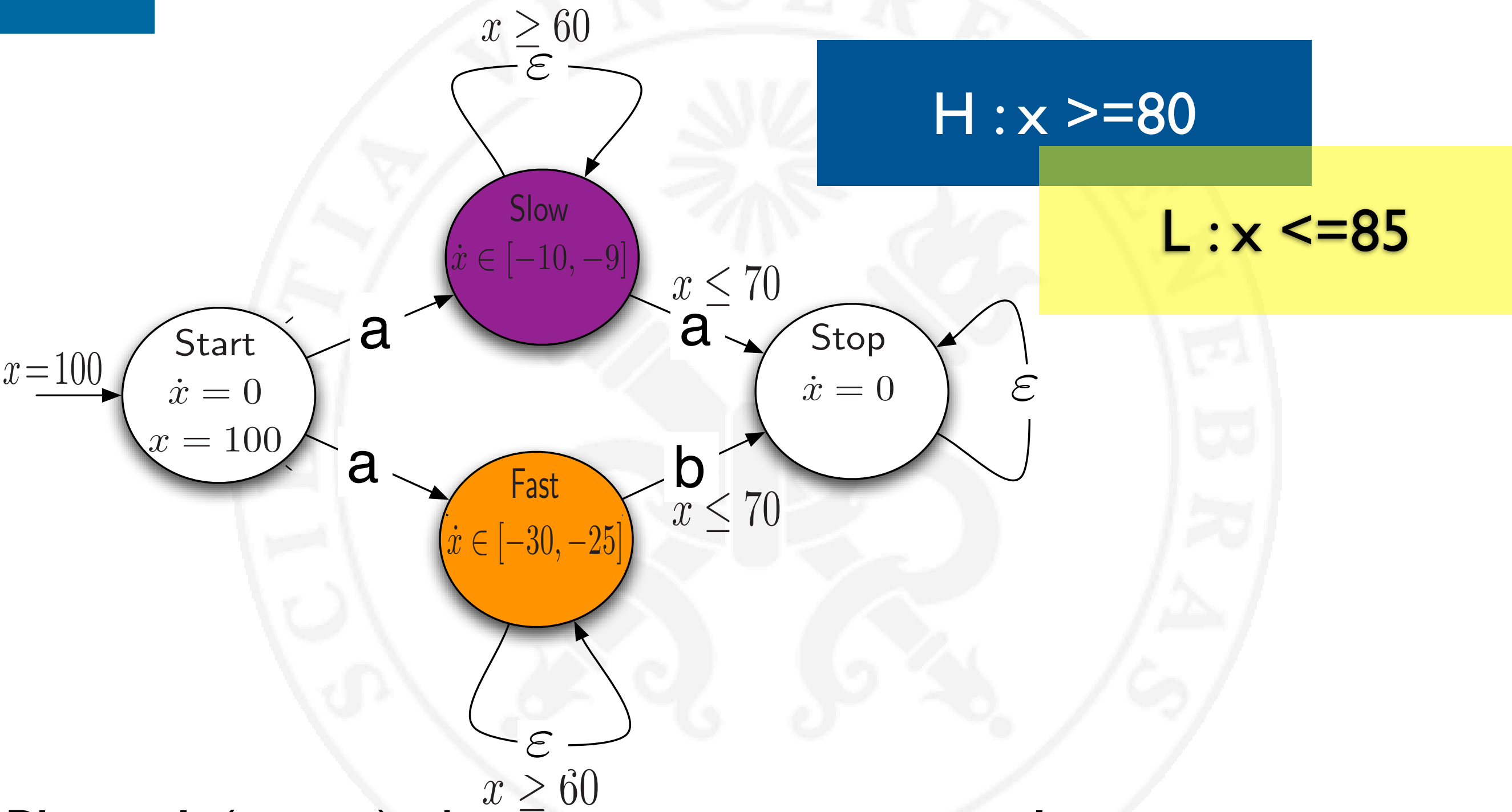
Our fixed point algorithm will terminate **if**

There exists a **finite quotient** of the state space

Post, Enabled, γ are **definable using this quotient**

Application : Discrete Time Control of RHA

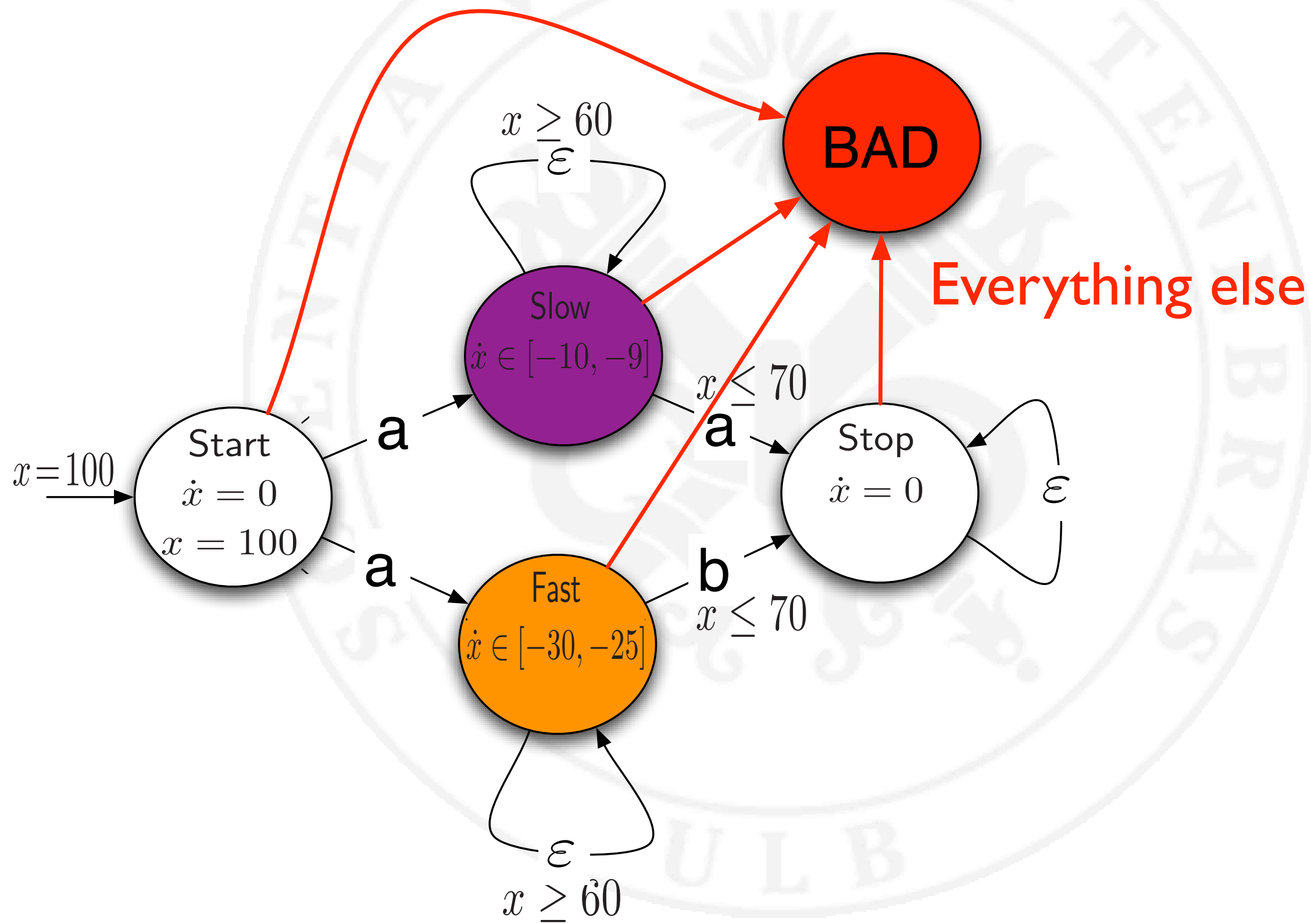
Discrete time control of RHA



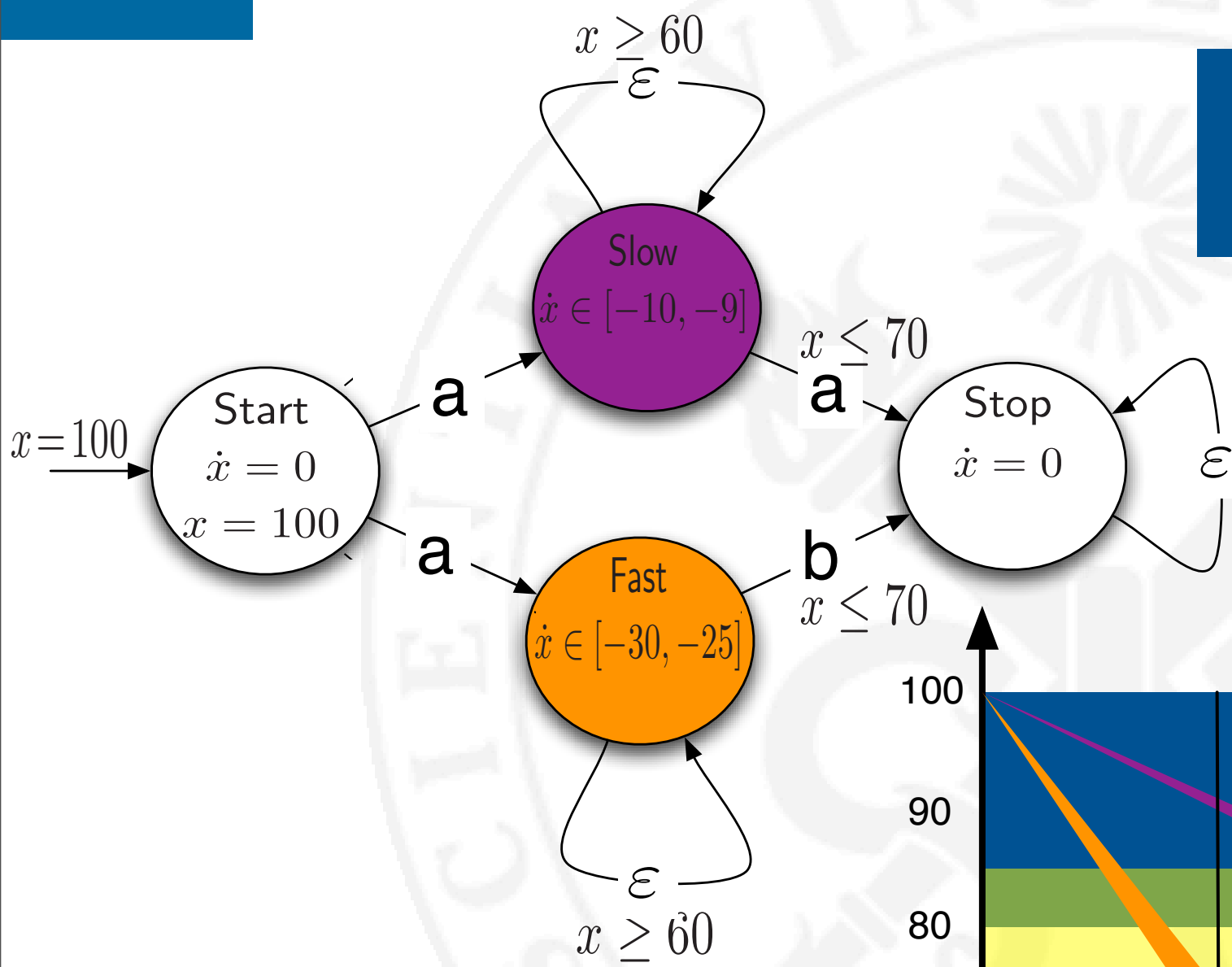
Player 1 (contr.) chooses an action every 1 time unit

Player 2 (env.) resolves nondeterminism
(in discrete and continuous steps).

Discrete time control of RHA

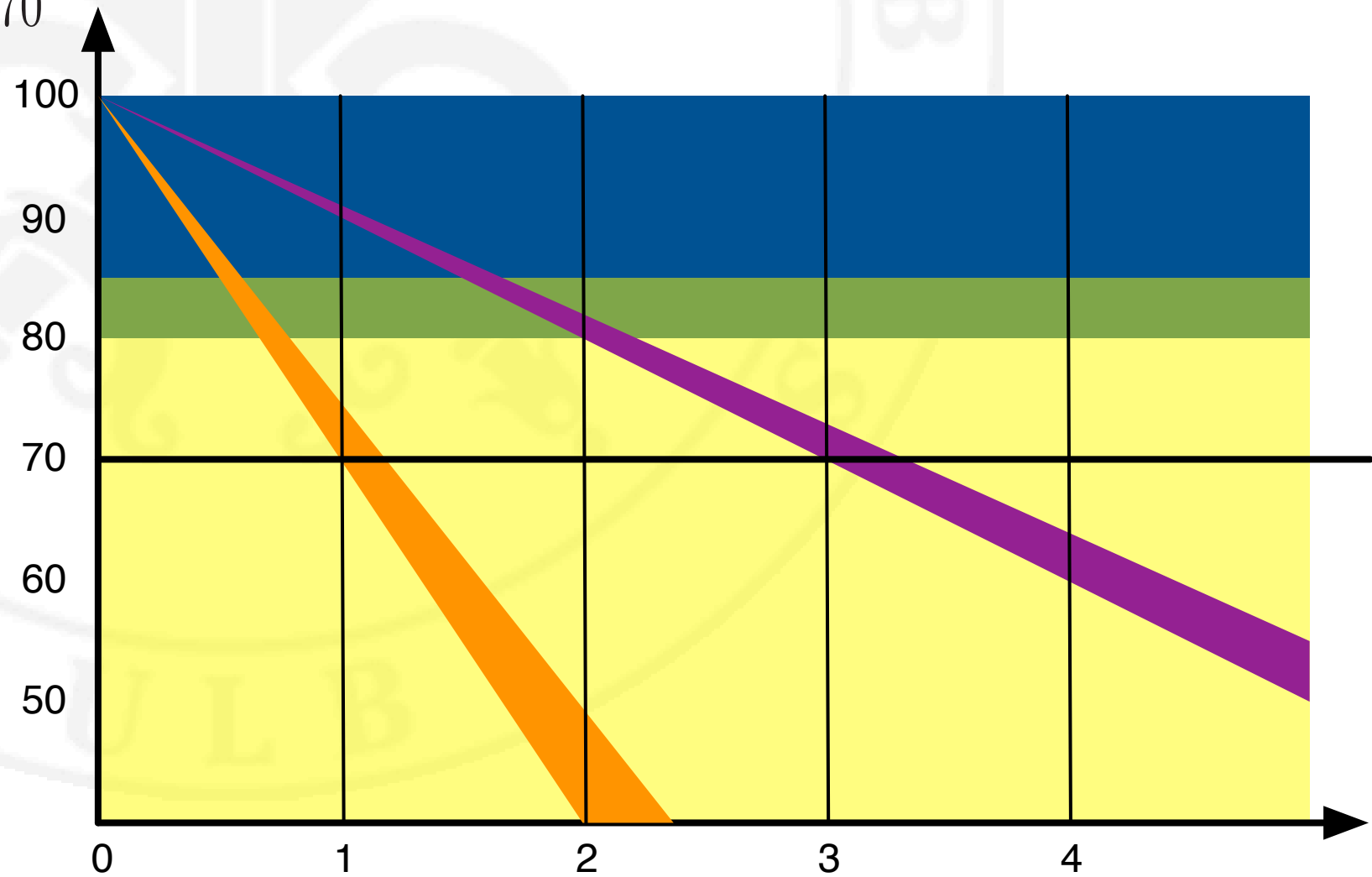


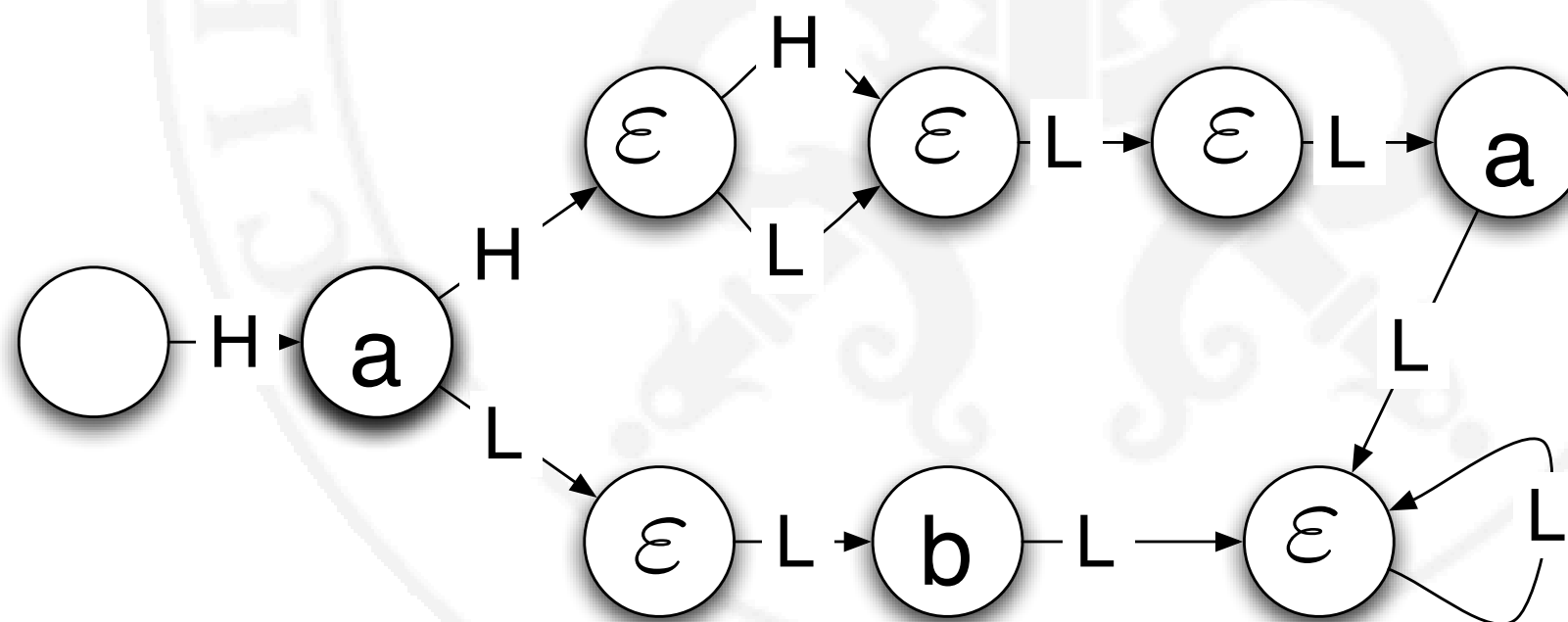
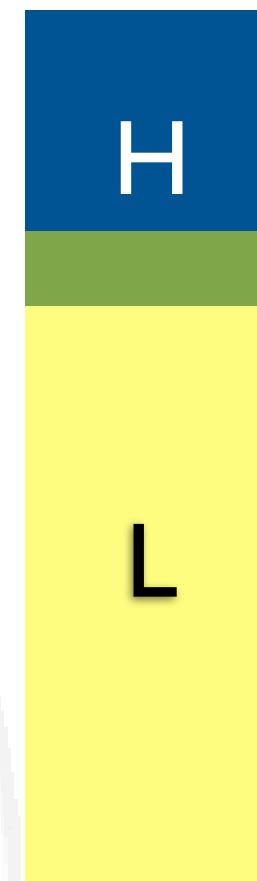
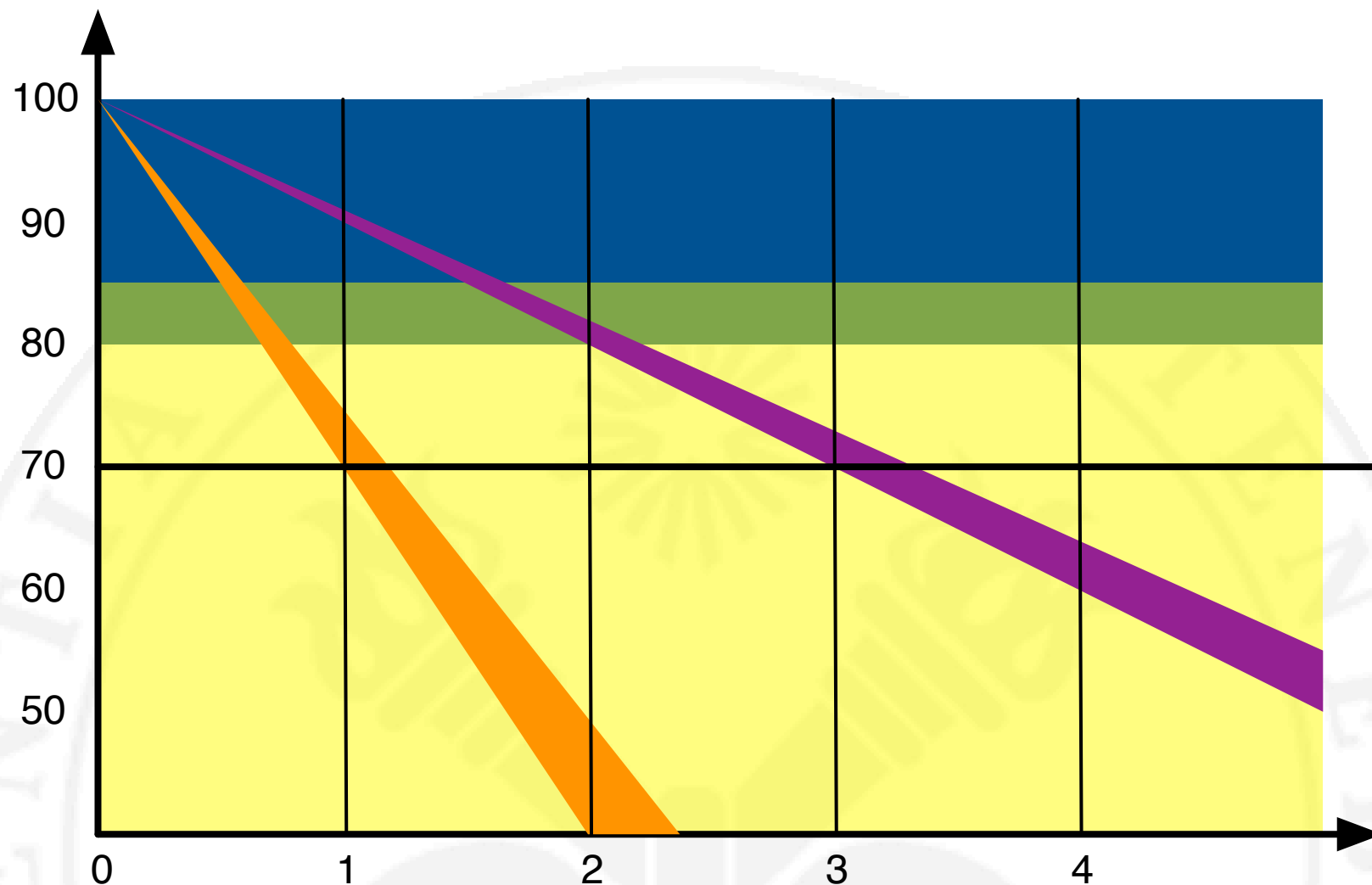
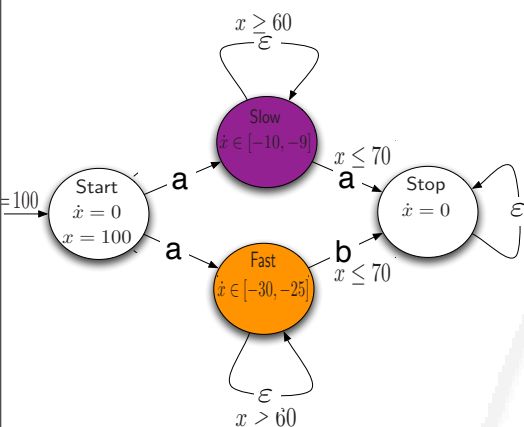
Discrete time control of RHA



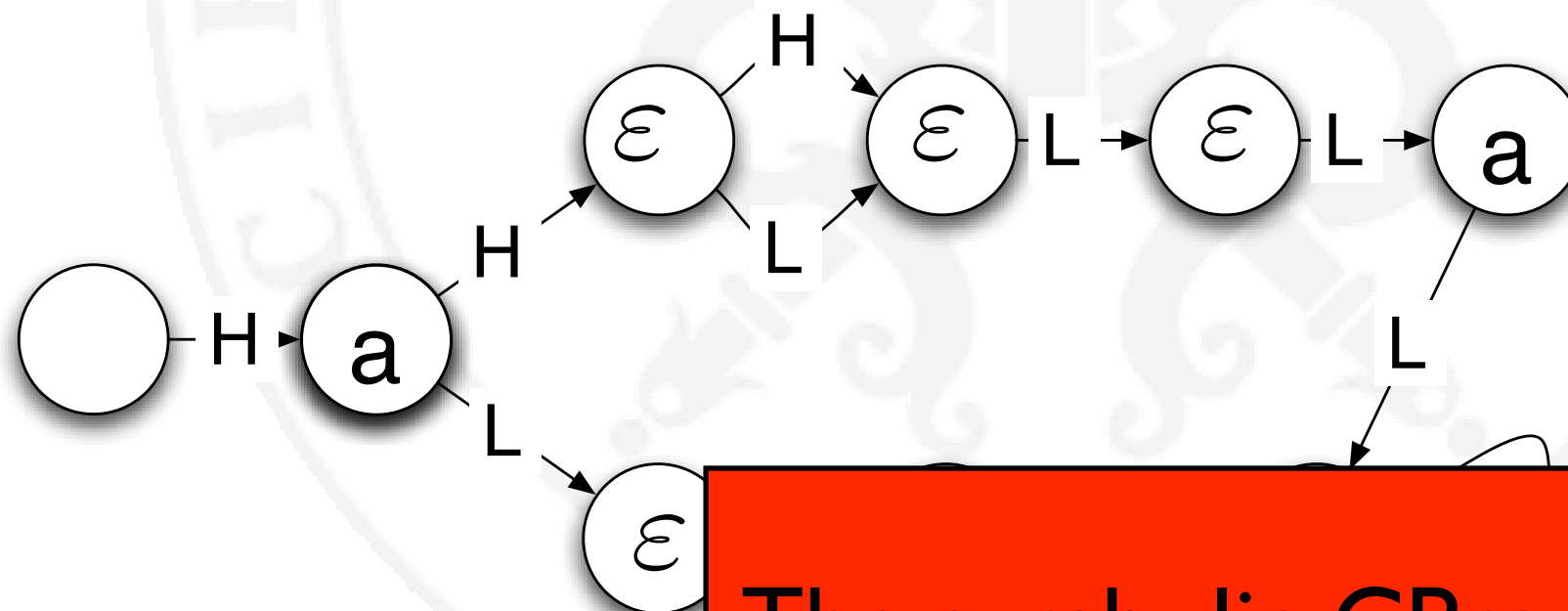
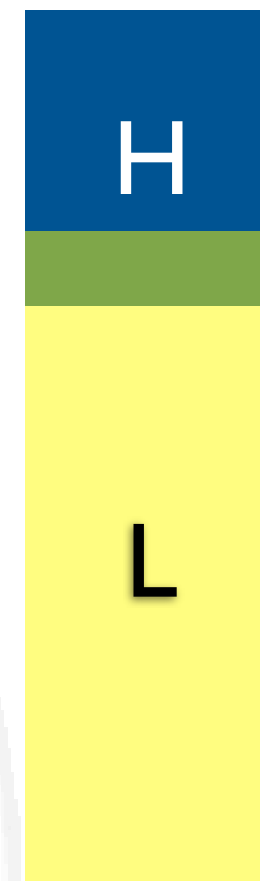
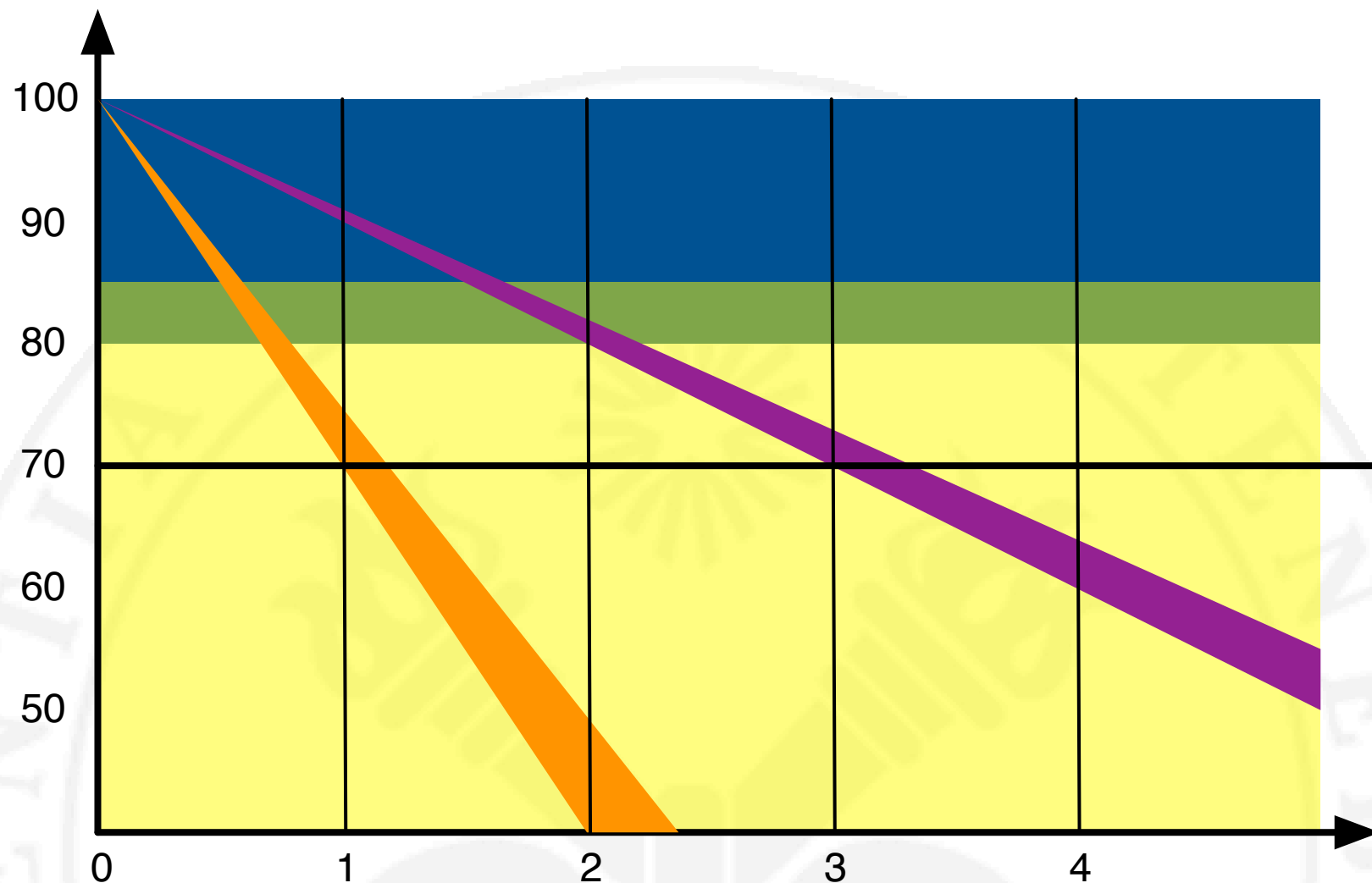
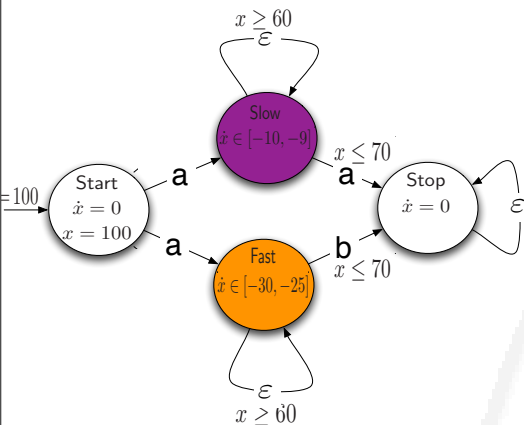
H : $x \geq 80$

L : $x \leq 85$





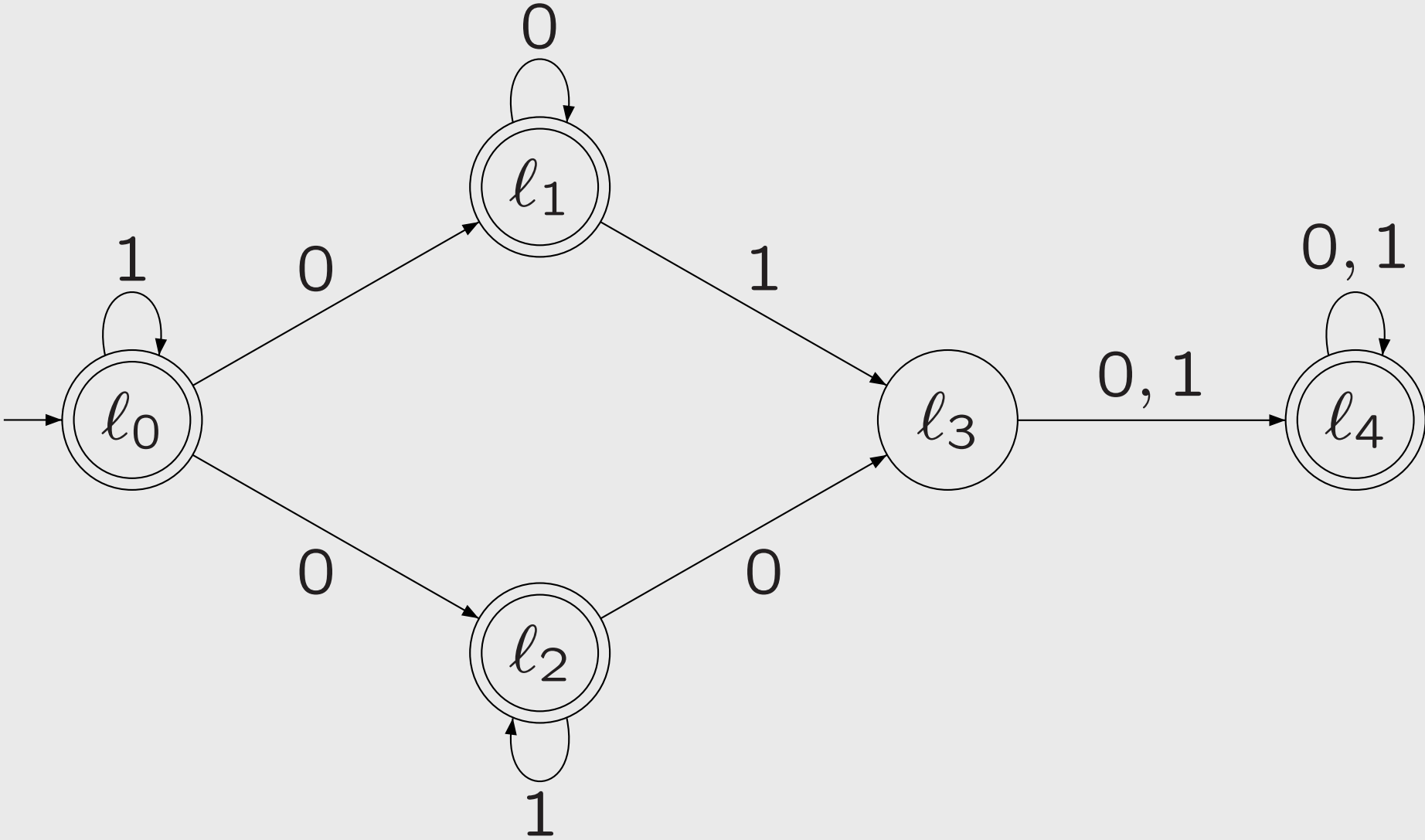
The Strategy



The symbolic CPre can be encoded in the script language of HyTech

Another application:
avoiding determinization
when testing
universability of NFA

Universality of NFA



Universality of NFA

Consider a game played by a **protagonist** and an **antagonist**

The **protagonist** wants to establish that \mathcal{A} is not universal.

The **protagonist** has to provide a finite word w such that no matter how the **antagonist** reads it using \mathcal{A} , the automaton ends up in a rejecting location.

\Rightarrow This is a **one-shot** game.

Universality of NFA

Consider a game played by a **protagonist** and an **antagonist**

The **protagonist** wants to establish that A is not universal.

The **protagonist** has to provide a finite word w such that no matter how the **antagonist** reads it using A , the automaton ends up in a rejecting location.

\Rightarrow This is a **one-shot** game.

The game is turn-based: the **protagonist** provides the word w one letter at a time, and the **antagonist** updates the state of A . The **protagonist cannot observe** the state chosen by the **antagonist**.

\Rightarrow This is a **blind** game (or game of null information).

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Consider the following **controllable predecessor operator** over sets of sets of locations. For $q \subseteq 2^{\text{Loc}}$, let:

$$\text{CPre}(q) = \{s \mid \exists s' \in q \cdot \exists \sigma \in \Sigma \cdot \forall \ell \in s \cdot \forall \ell' \in \text{Loc} : \delta_A(\ell, \sigma, \ell') \rightarrow \ell' \in s'\}$$

So $s \in \text{CPre}(q)$ if there is a set $s' \in q$ that is reached from any location in s , reading input letter σ , that is $\text{Post}_\sigma(s) \subseteq s'$.

\Rightarrow CPre encodes the **blindness** of the game.

Let $\mathcal{A} = \langle \text{Loc}, \ell_I, \Sigma, \delta_A, F \rangle$.

Theorem:

$$\{\ell_I\} \in \mu x. (\text{CPre}(x) \cup \{T\})$$

iff

Protagonist has a strategy to win G_T

iff

\mathcal{A} is not universal

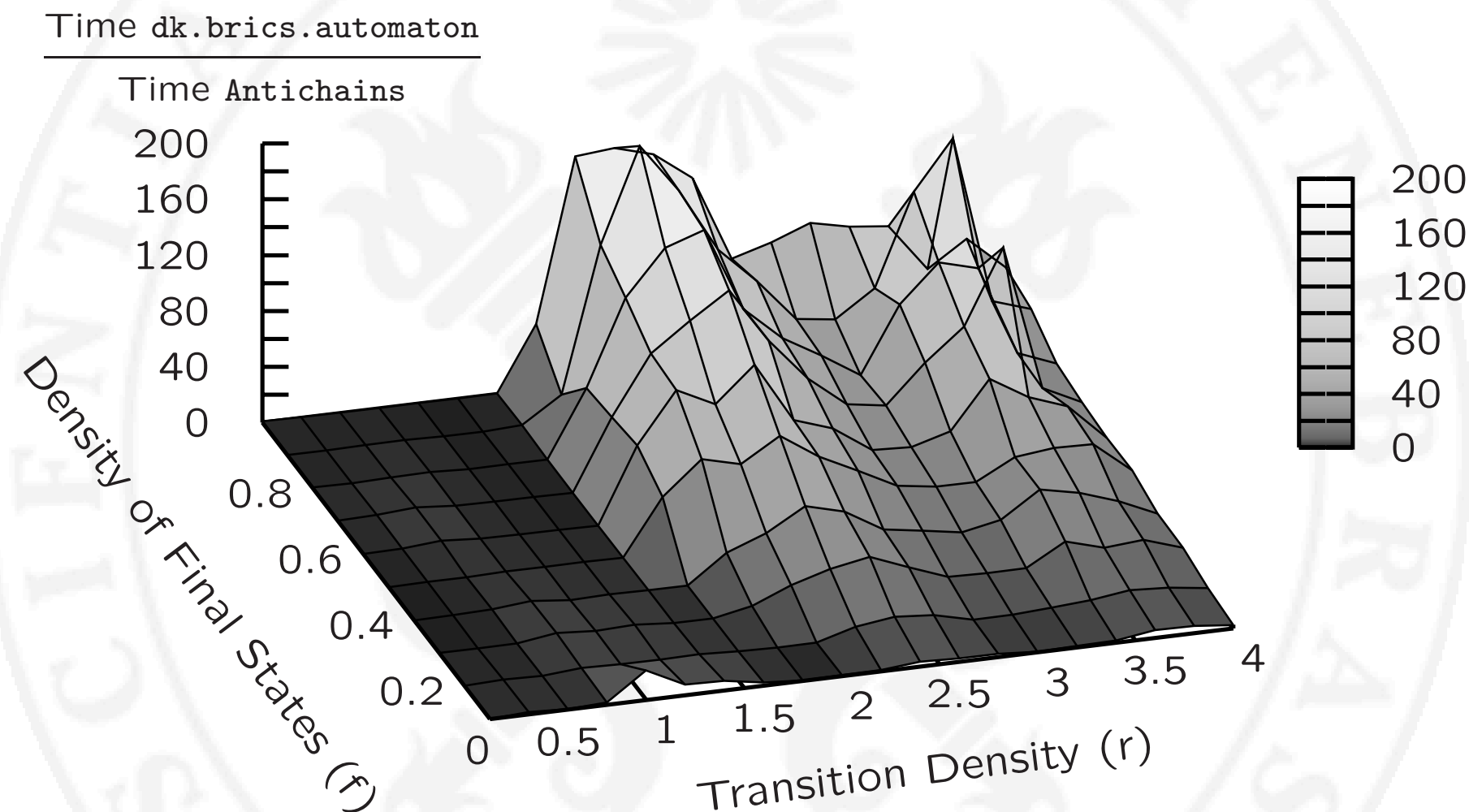
Claim: For $s_1 \subseteq s_2$, if $\text{Post}_\sigma(s_2) \subseteq s'$ then $\text{Post}_\sigma(s_1) \subseteq s'$
and if $s_2 \in \text{CPre}(\cdot)$, then $s_1 \in \text{CPre}(\cdot)$

Idea: Keep in $\text{CPre}(x)$ only the **maximal** elements.

Universality - Experimental results (1)

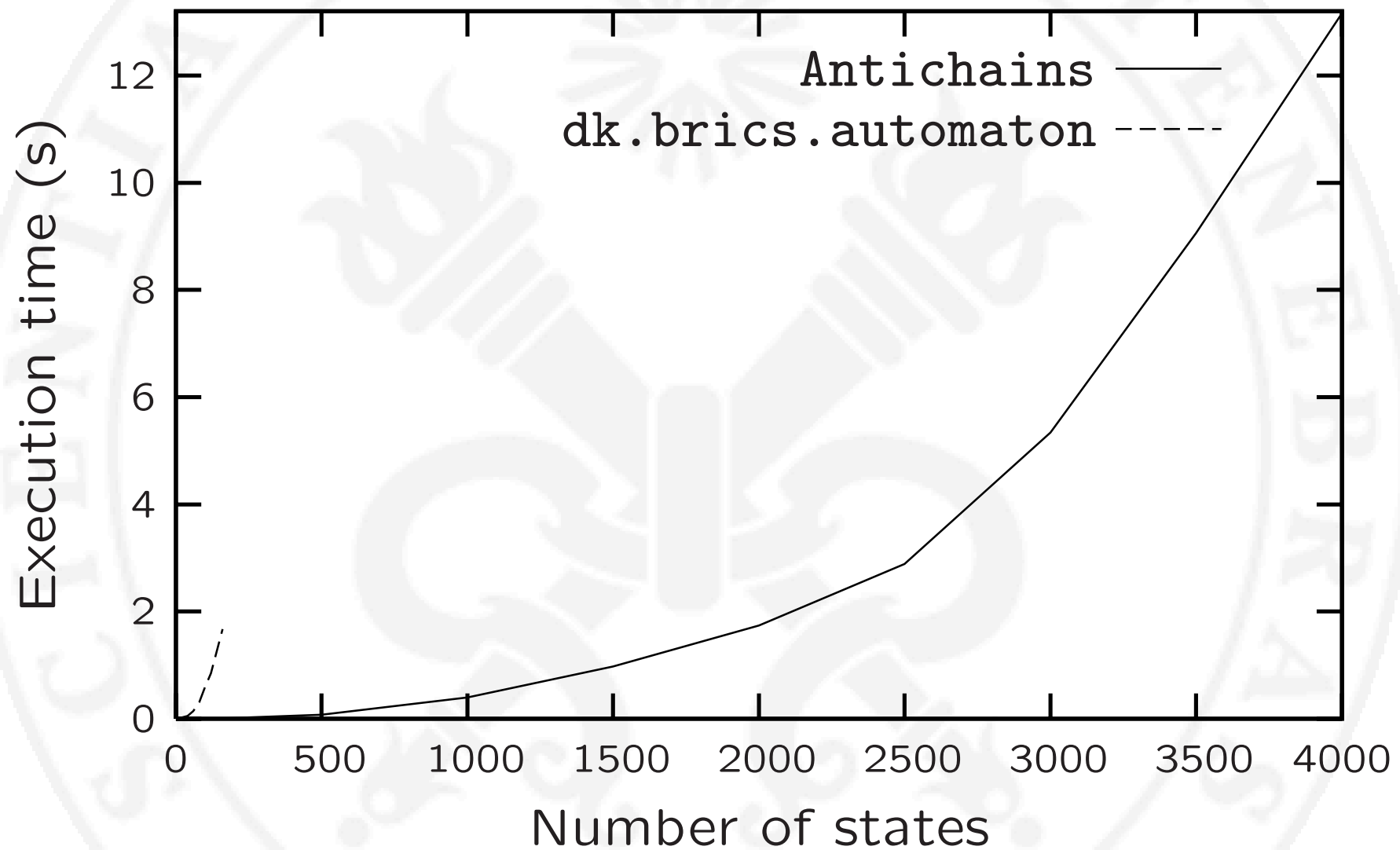
- We compare our algorithm [Antichains](#) with the best⁽¹⁾ known algorithm [dk.brics.automaton](#) by Anders Møller.
(1) According to "*D. Tabakov, M. Y. Vardi. Experimental Evaluation of Classical Automata Constructions. LPAR 2005*".
- We use a randomized model to generate the instances (automata of 175 locations). Two parameters:
 - Transition density: $r \geq 0$
 - Density of accepting states: $0 \leq f \leq 1$

Universality - Experimental results (2)



Each sample point: 100 automata with $|Loc| = 175$, $\Sigma = \{0, 1\}$.

Universality - Experimental results (3)



- Transition density: $r = 2$.
- Density of accepting states: $f = 1$.

Works also for

- *language inclusion* between NFA
- *emptiness* of AFA
- See proceedings of next CAV !

(joint work with Martin De Wulf, Laurent Doyen and Tom Henzinger)

Conclusion/Perspectives

- We propose a **lattice theory** to solve games of imperfect information, those games are needed to make the synthesis of **robust controllers** (= finite precision).
- Our technique computes only the information that is needed to find a winning strategy, i.e. we **avoid** the explicit subset construction.
- Applicable to **discrete time control** of RHA and useful to solve efficiently **classical problems** for NFA and AFA.
- Perspectives : continuous time control, finite automata on infinite words, efficient implementation issues, etc.

- [ACH⁺95] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.
- [AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
- [AHK02] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.
- [CC77] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252, 1977.
- [DDR06] M. De Wulf, L. Doyen, and J.-F. Raskin. A lattice theory for solving games of imperfect information (extended version). Technical Report 58, U.L.B. – Federated Center in Verification, 2006. <http://www.ulb.ac.be/di/ssd/cfv/publications.html>.
- [HHWT95] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. A user guide to HYTECH. In *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 1019, pages 41–71. Springer-Verlag, 1995.
- [HK99] T.A. Henzinger and P.W. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221:369–392, 1999.
- [MPS95] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In *STACS’95*, volume 900 of *Lecture Notes in Computer Science*, pages 229–242. Springer, 1995.
- [Rei84] John H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984.