GASICS

Games for **A**nalysis and **S**ynthesis of **I**nteractive **C**omputational **S**ystems

Highlights of Contributions

JF Raskin - Université Libre de Bruxelles (ULB)

Danish Agency for Science Technology and Innovation
Ministry of Science Technology and Innovation

Deutsche Forschungsgemeinschaft DFG

fnrs LA LIBERTÉ DE CHERCHER

European Science Foundation
SETTING SCIENCE AGENDAS FOR EUROPE

Berlin, September 16, 2011

Plan of the talk

- Consortium
- Verification and synthesis
- Game graphs - basic framework
- **Beyond** the basic framework
- **Highlights of contributions:**
    - Energy games
    - LTL games
    - Nash equilibria in game graphs
    - Game graphs with imperfect information

The GASICS consortium

Core partners

Associated partners

U Warwick

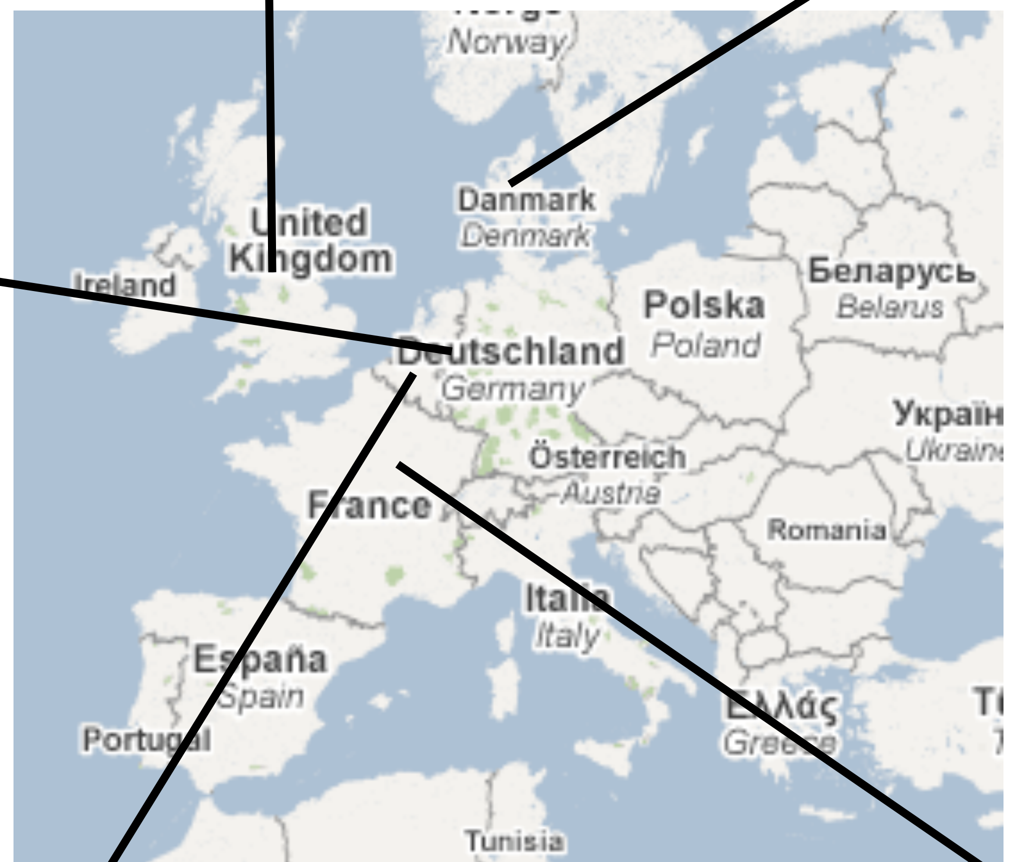
Prof. Marcin Jurdzinski

U Aalborg (FNU)

Prof. Kim Larsen

RWTH Aachen (DFG)

Prof. Wolfgang Thomas.



Université Libre de Bruxelles (FNRS)

Prof. Jean-François Raskin

U Paris 7
ENS Cachan

Dr. Olivier Serre

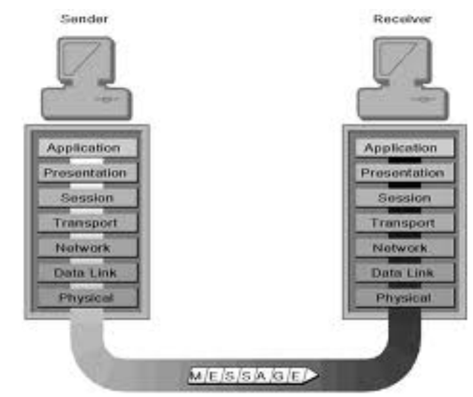
Dr. Nicolas Markey

Verification and Synthesis of Reactive Systems

Reactive Systems

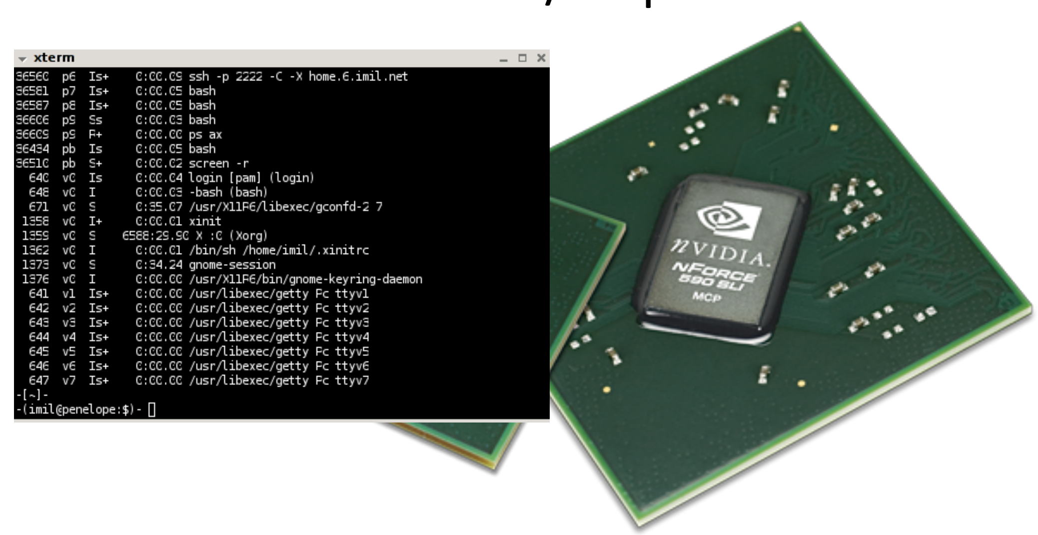
Embedded Control

Communication Protocols



Security Protocols

Parts of OS/Chipset



Reactive systems maintain a continuous interaction with their environment

CAV of Reactive Systems

>>> Safety critical applications: need for **formal methods** <<<

Model of the system and of its environment	$\models$?	Spec. of the system
Automata and extensions	Model-checking Algorithms **(C.E.S. Turing Award)**	Temporal Logics **(A. Pnueli Turing Award)**
Env || Sys	$\models$?	ψ

CAV of Reactive Systems

>>> Safety critical applications: need for **formal methods** <<<

Model of the system and of its environment

$\models$?

...stem

Automat...

...ecking Algorit... **(C.E.C...**

...g Award)

Env...

$\models$?

ψ

Verification is done **a posteriori**, after important design decisions

Model-checking is a “high level” **debugging** tool !

Beyond Verification: Synthesis

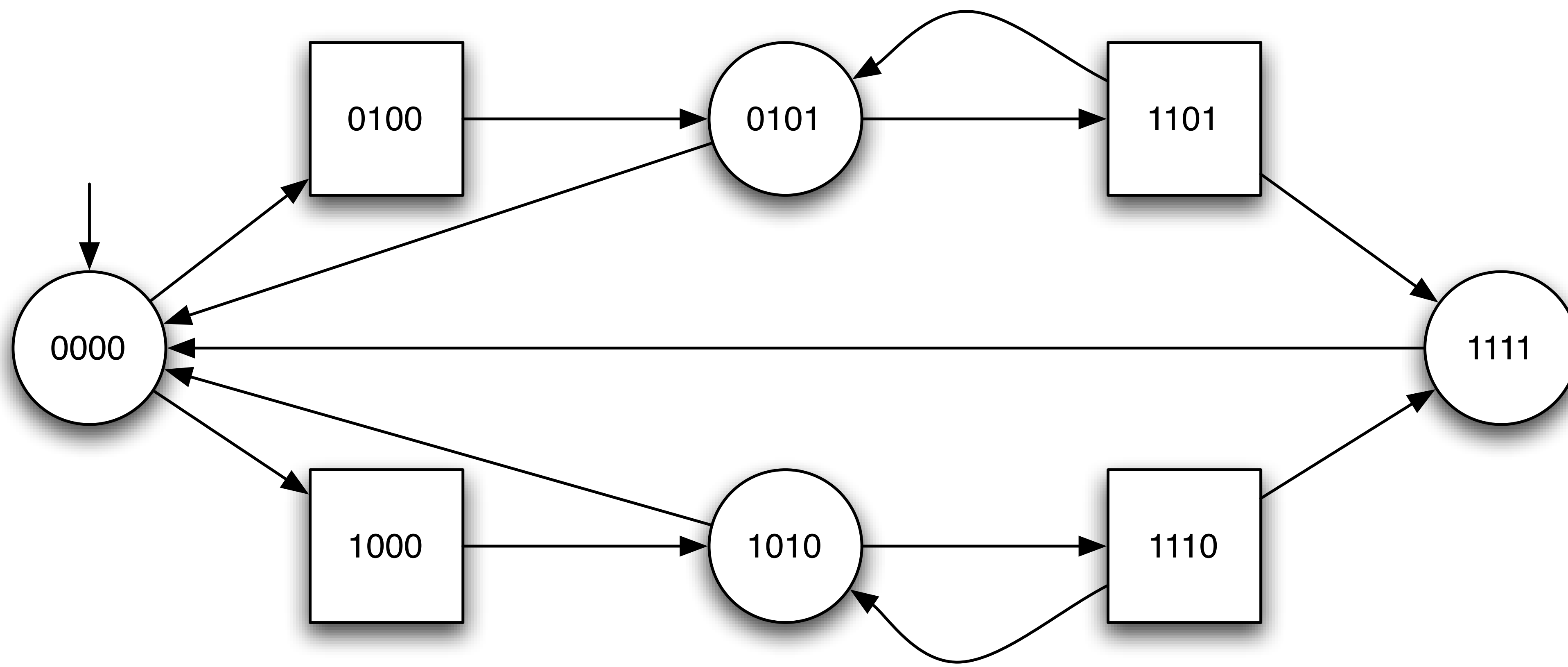
Basic Framework:
2-Player Zero-Sum Games

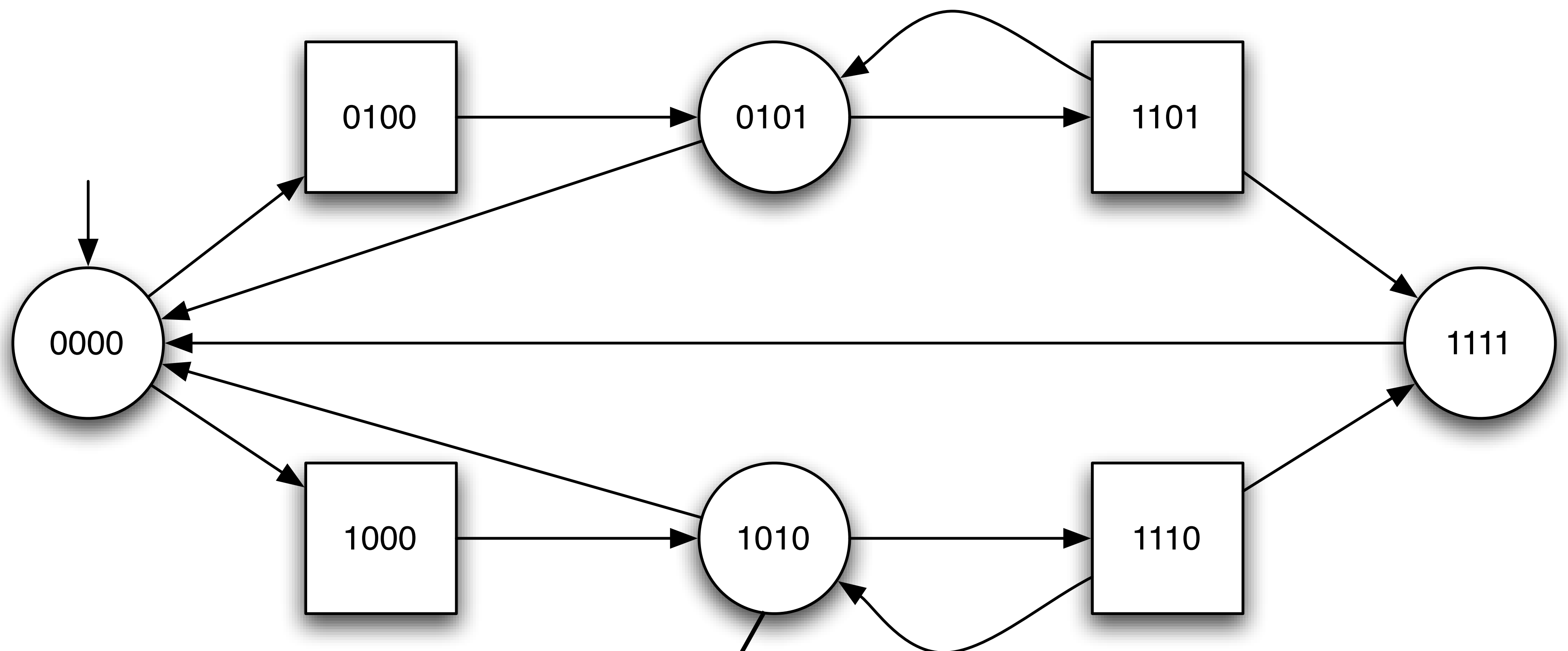
$$\boxed{\text{Env}} \;||\; \boxed{\textbf{?}} \quad \models \quad \boxed{\psi}$$

- Sys is constructed by an algorithm
- Sys is **correct** by construction
- Underlying theory: **2-player zero-sum games**
- Env is **adversarial** (worst-case assumption)

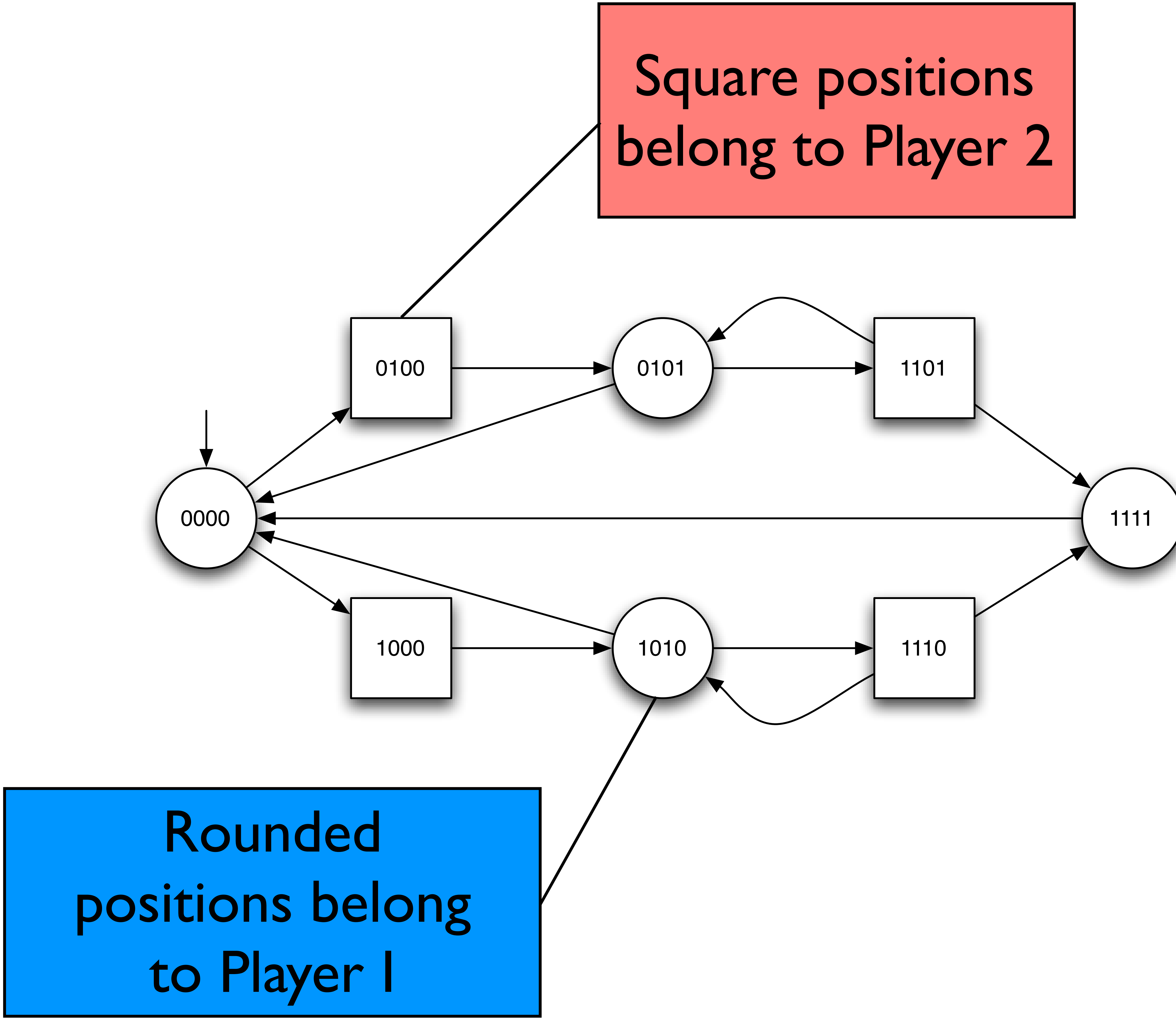
Correct Sys = Winning strategy

Game Graphs for Synthesis
The Basic Framework


0100
0101
1101
0000
1111
1000
1010
1110

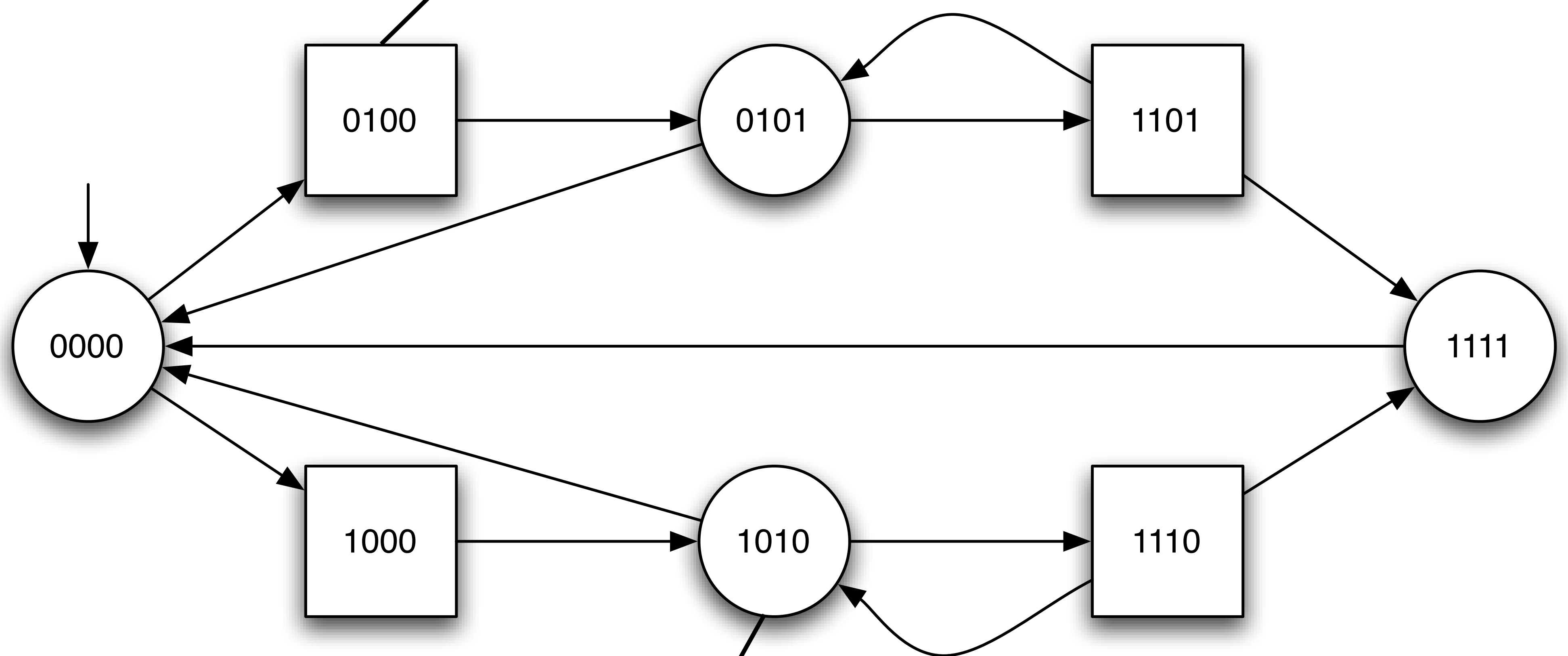


Rounded positions belong to Player I


Square positions belong to Player 2
0100
0101
1101
0000
1111
1000
1010
1110
Rounded positions belong to Player 1

The controller=

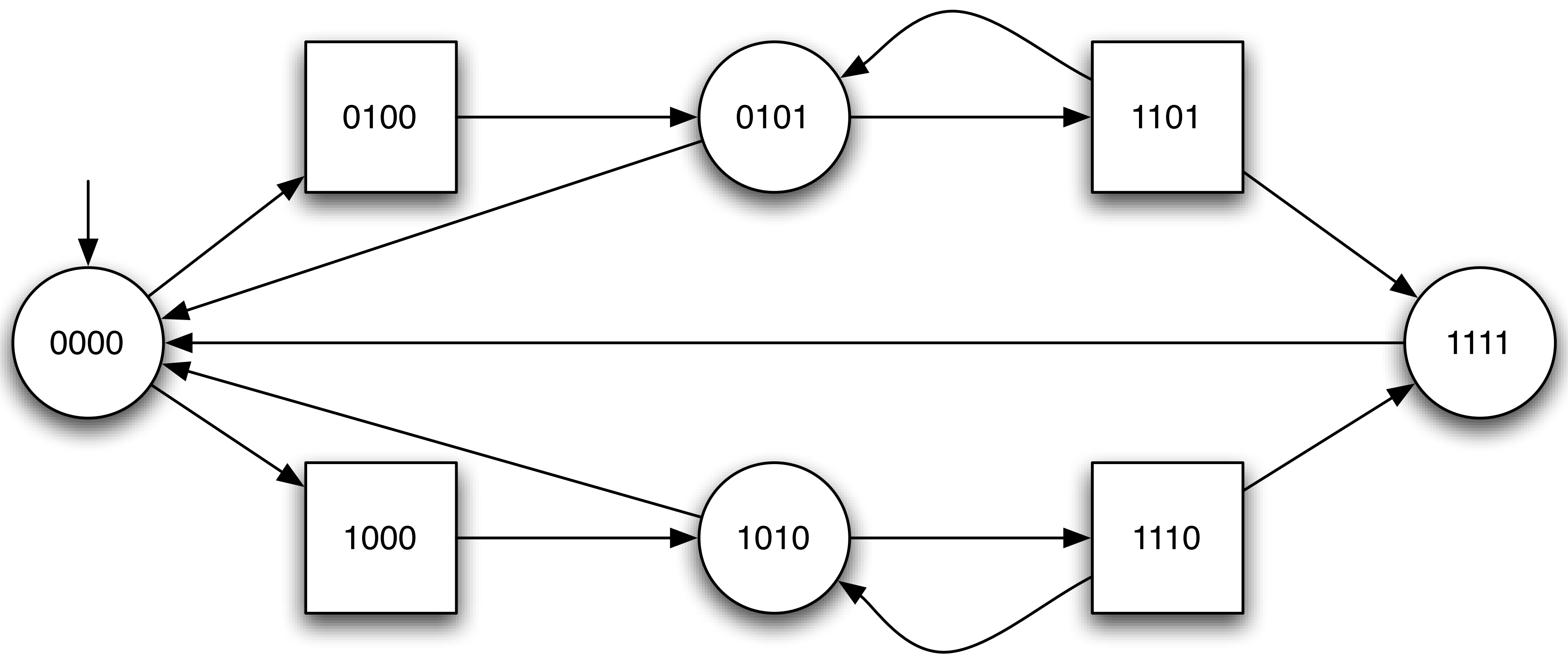
Square positions belong to Player 2



Rounded positions belong to Player 1

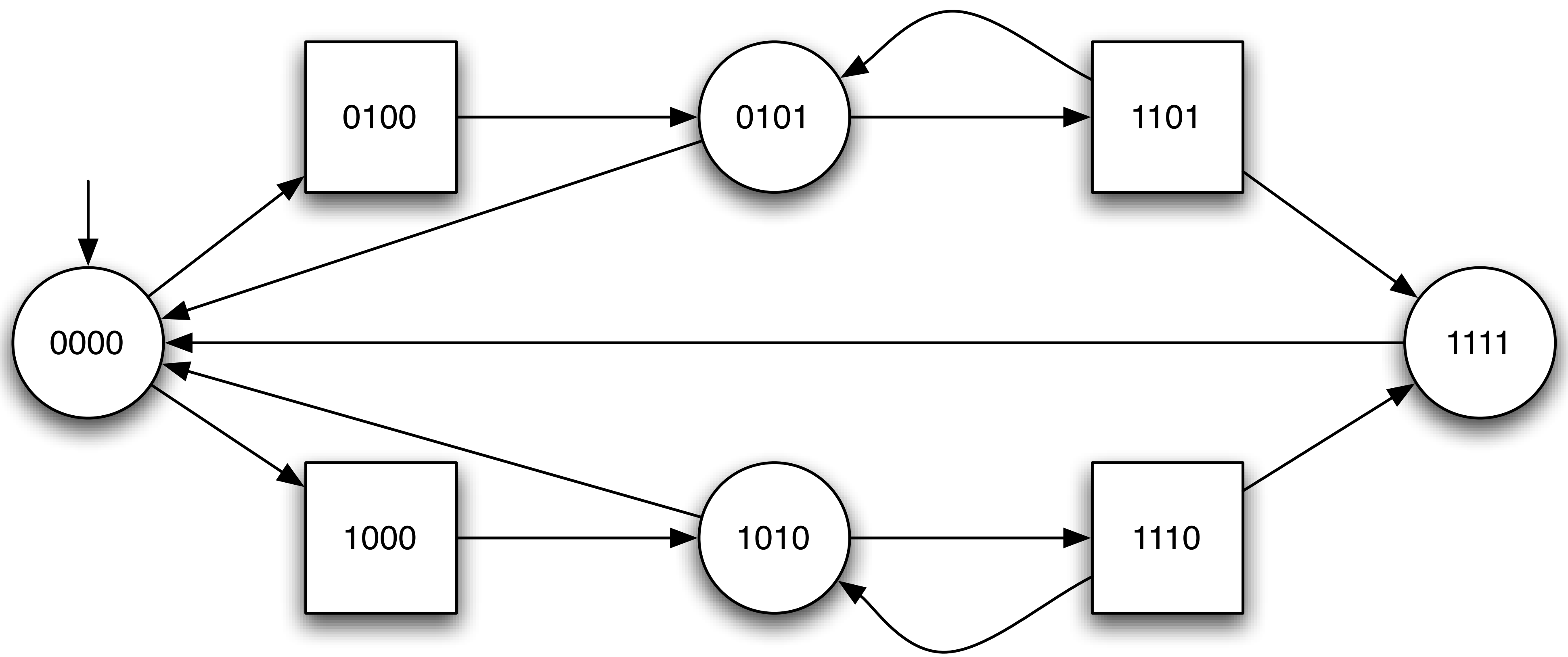
=The environment

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



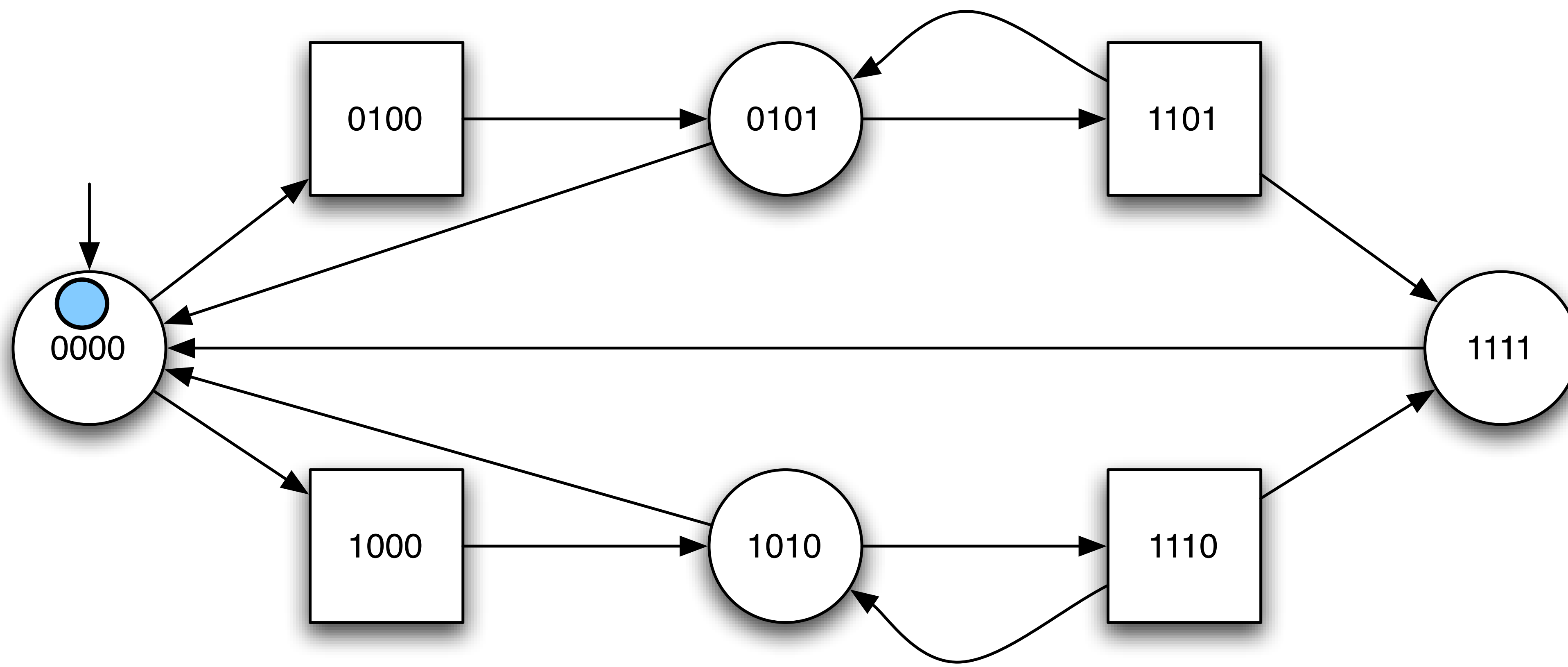
A game is played as follows: in each **round**, the game is in a **position**, if the game is in a rounded position, Player 1 resolves the **choice** for the next state, if the game is in a square position, Play 2 resolves the choice. The game is played for an **infinite number of rounds**.

Player 1 = Environment
Player 2 = Controller

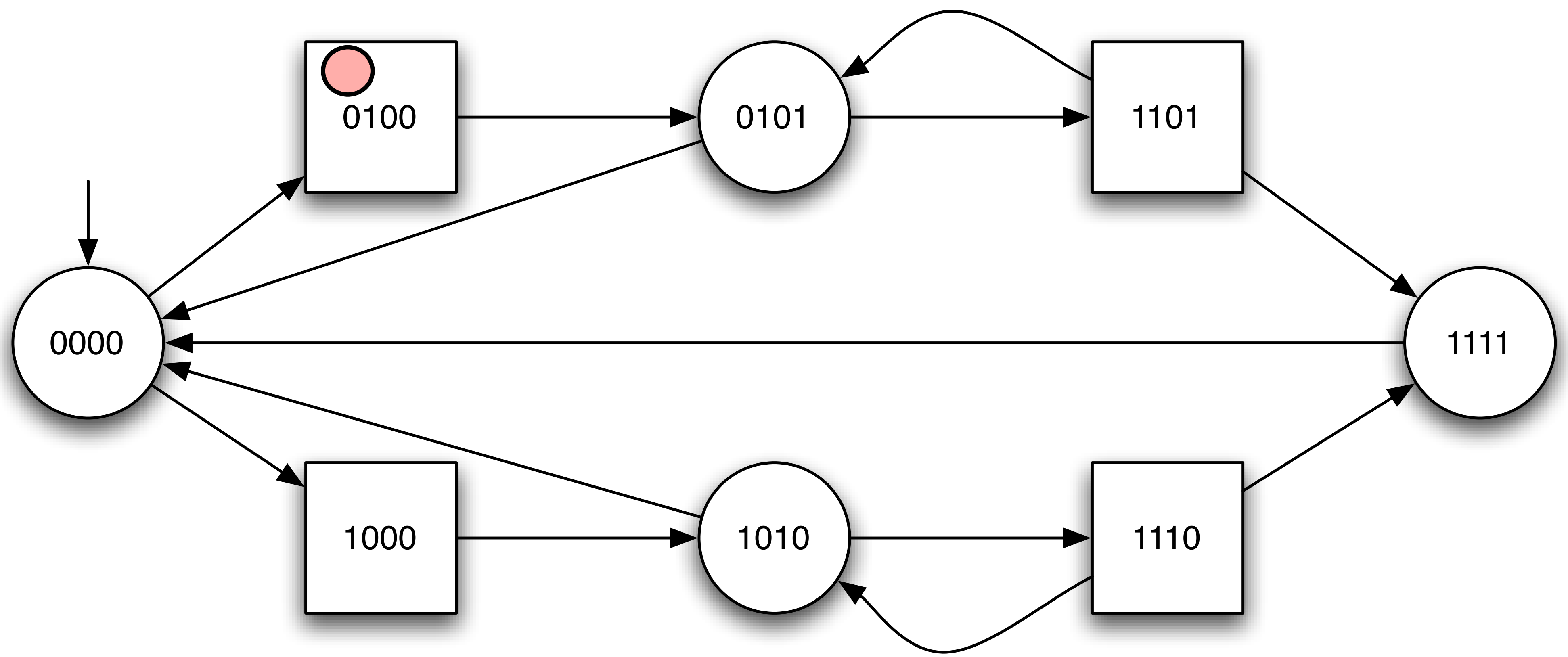


☛ The **choices** of the controller are to be interpreted as **decisions** that are to be taken to **control** the environment.

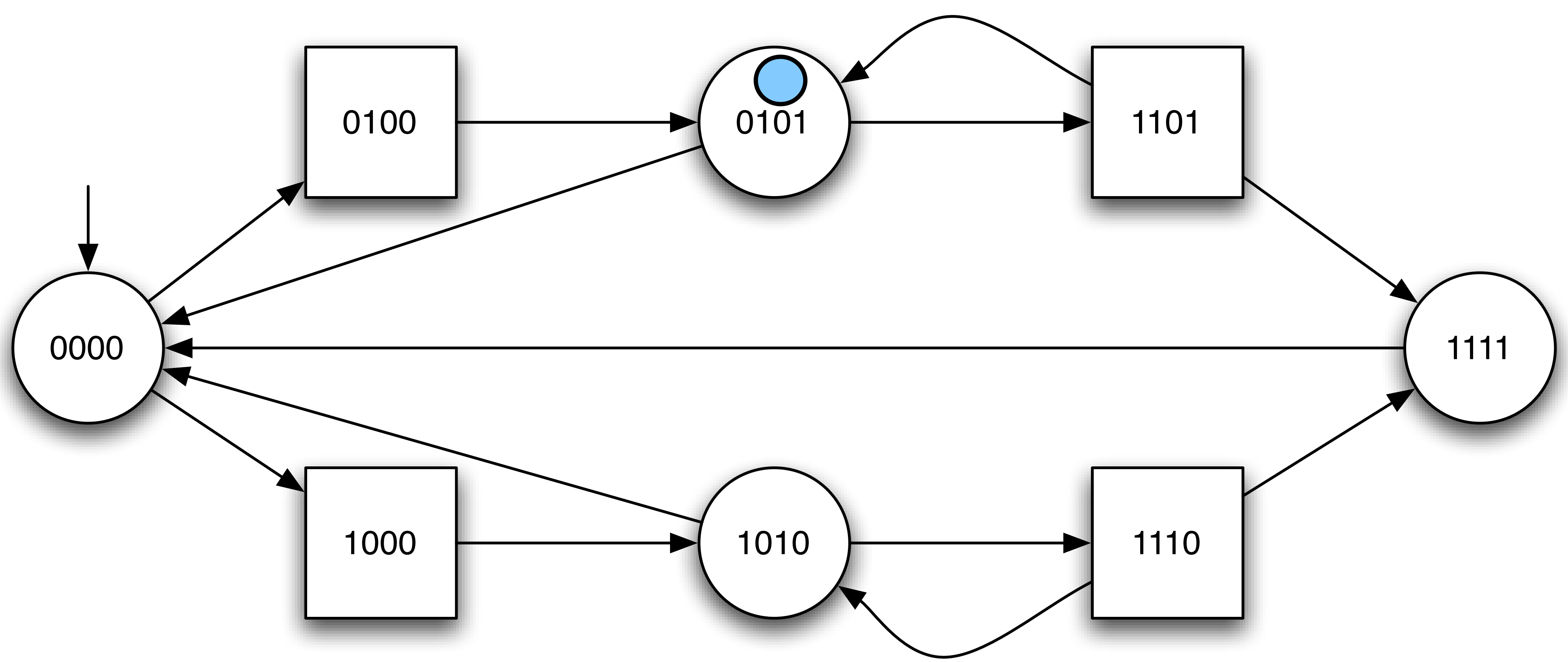
☛ The **choices** of the environment are beyond the control of the designer of the system and they must be interpreted as **adversarial**.



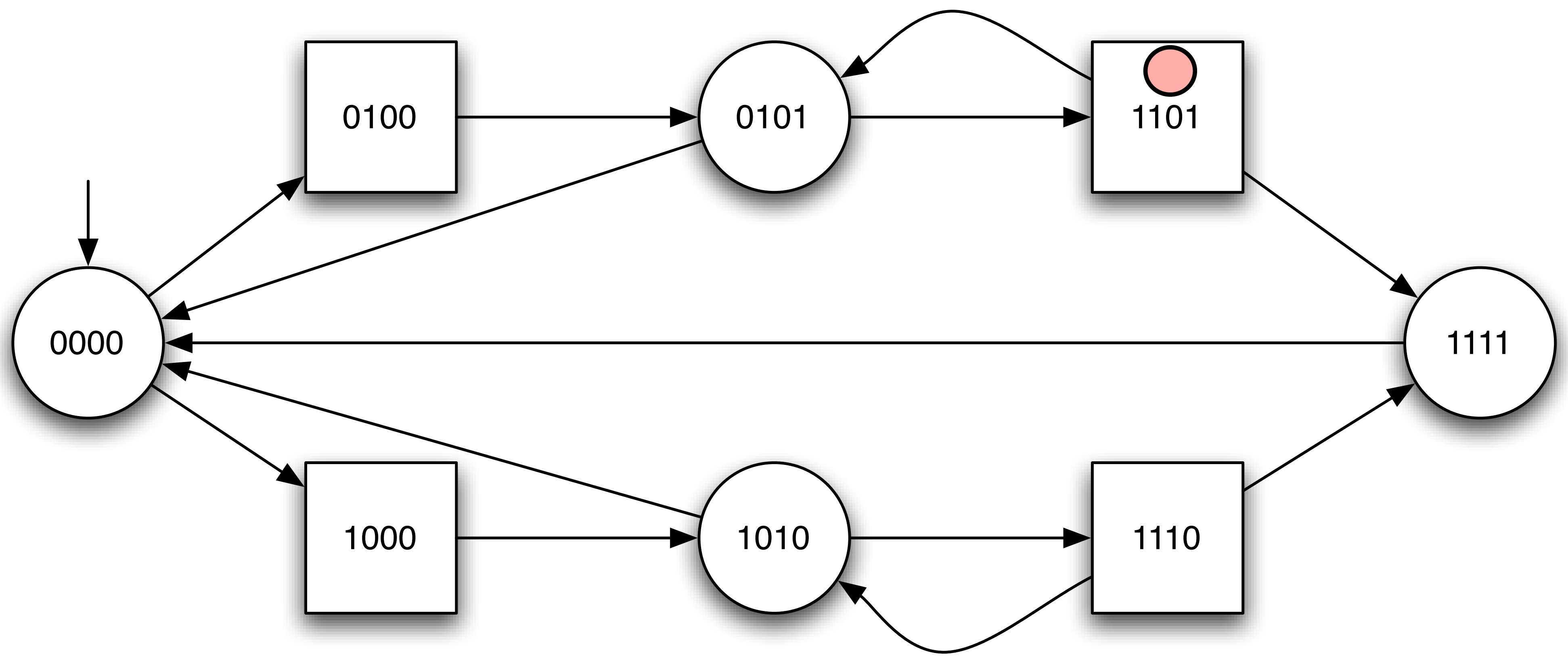
Play : 0000



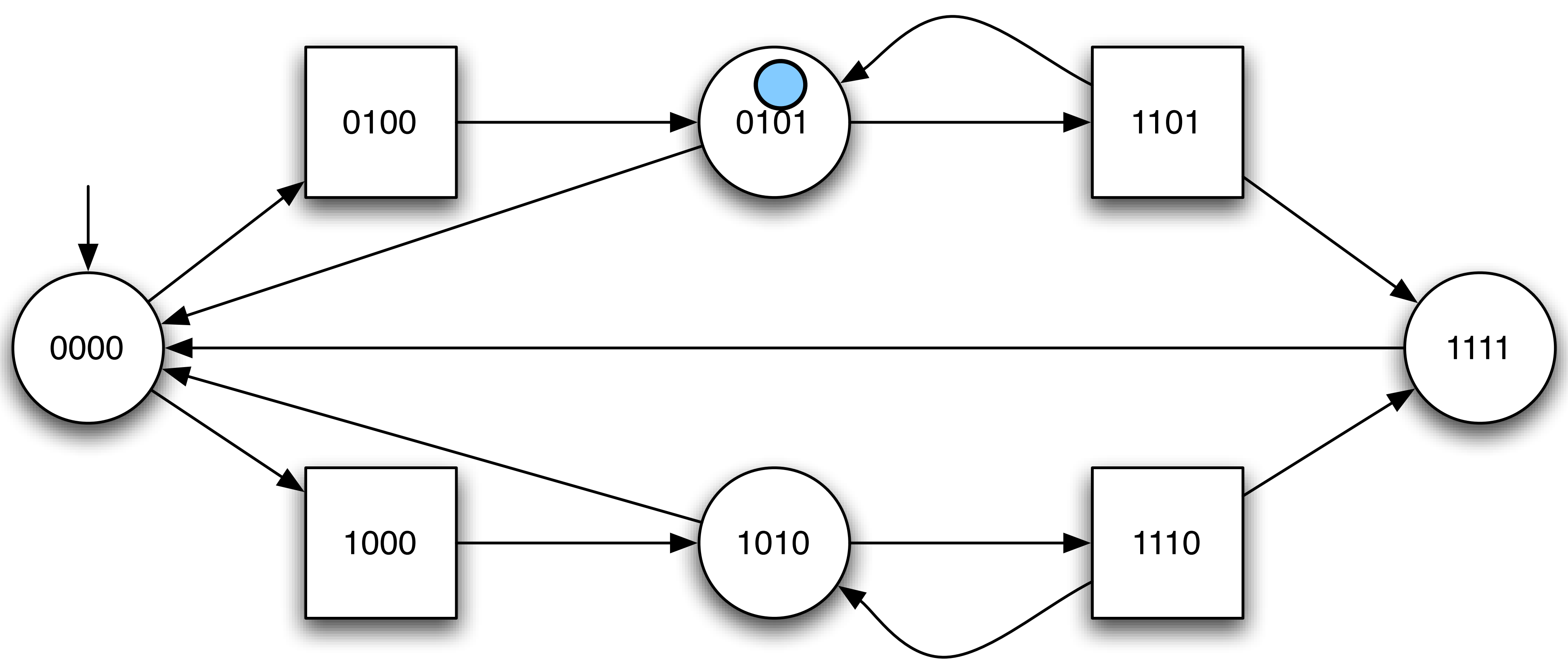
Play : 0000 0100



Play : 0000 0100 0101

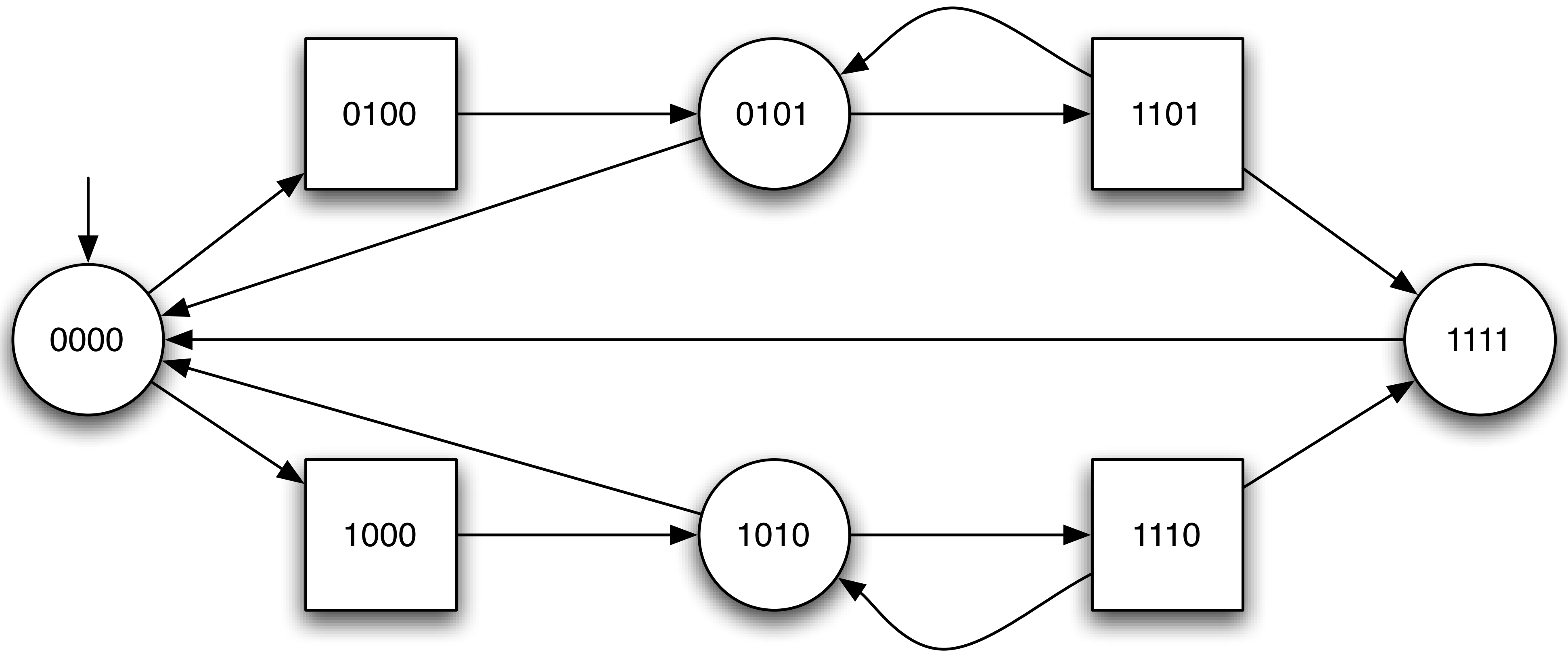


Play : 0000 0100 0101 1101



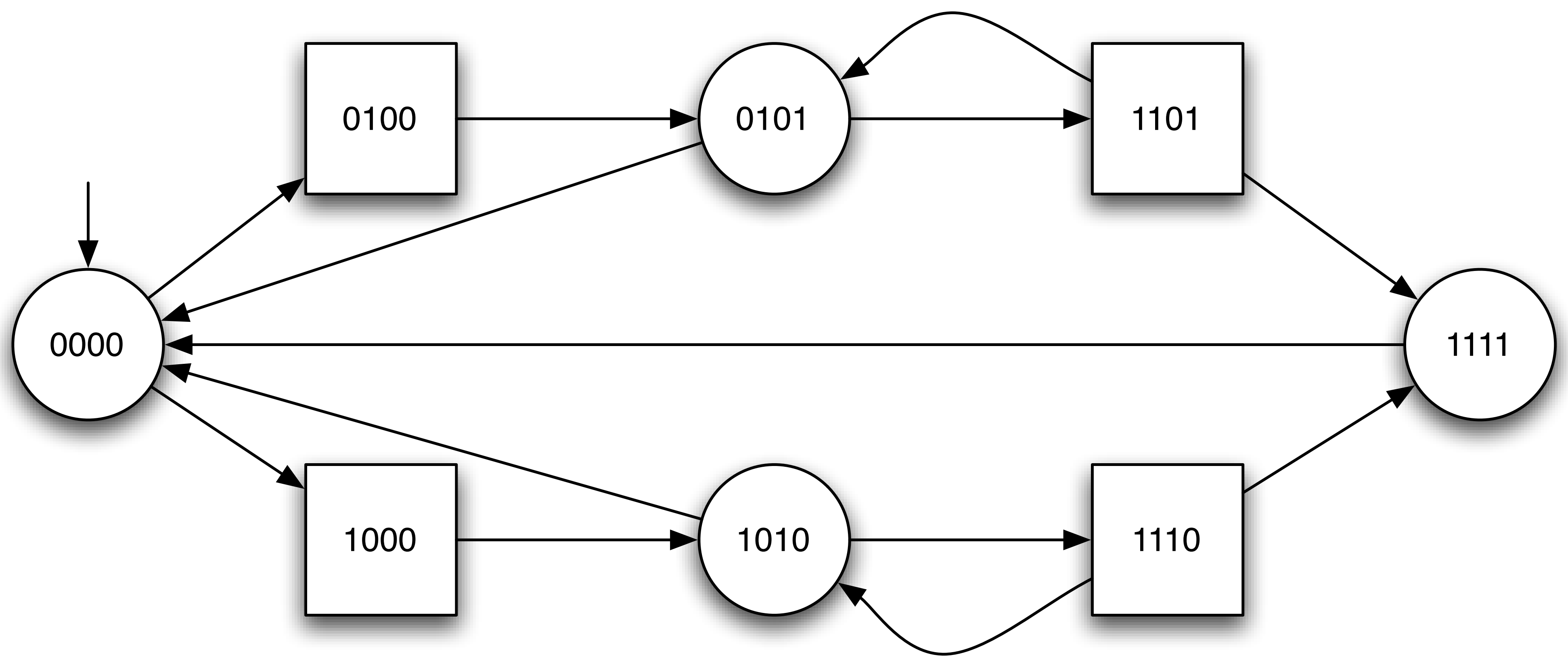
Play : 0000 0100 0101 1101 ...

Who is winning ?



Play : 0000 0100 0101 1101 ...

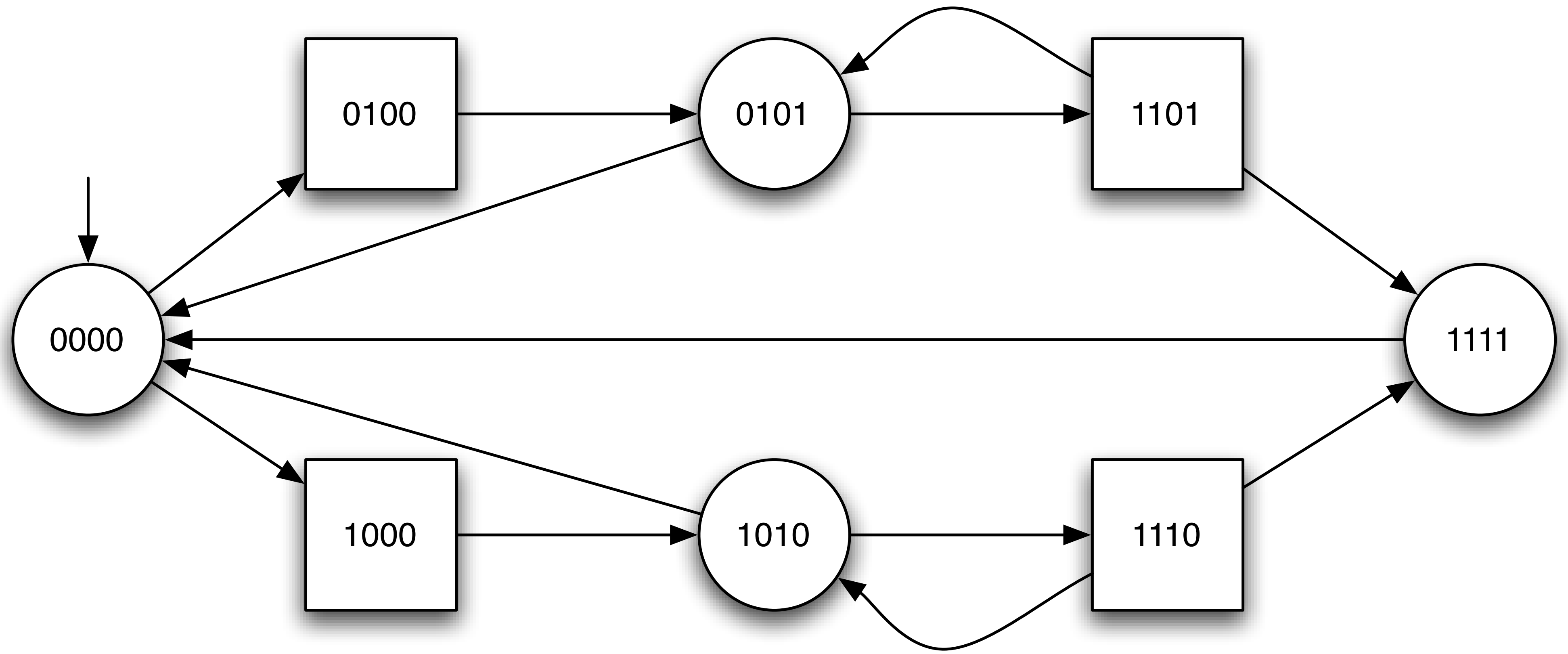
Who is winning ?



Play : 0000 0100 0101 1101 ...

=Trace, behavior of the system under design

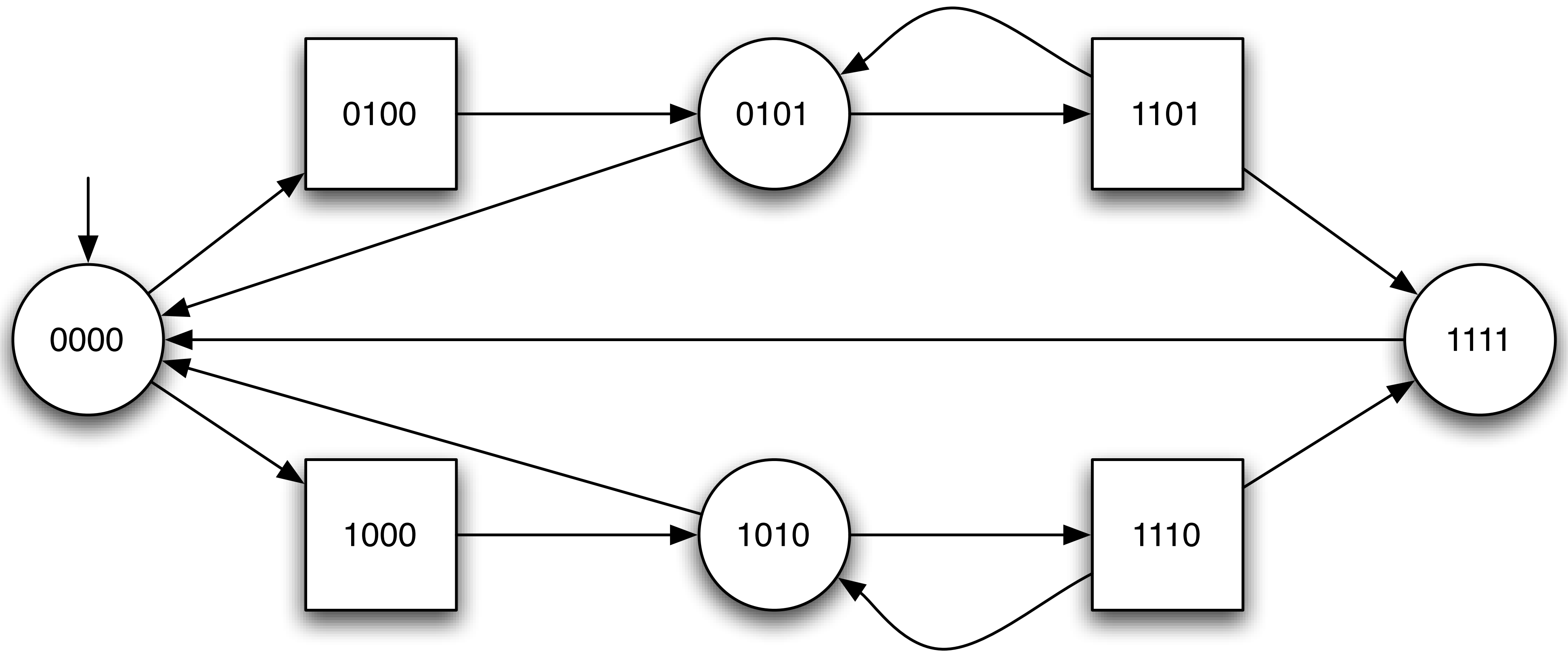
Who is winning ?



Play : 0000 0100 0101 1101 ...

Is this a good or a bad play for Player 2 ?

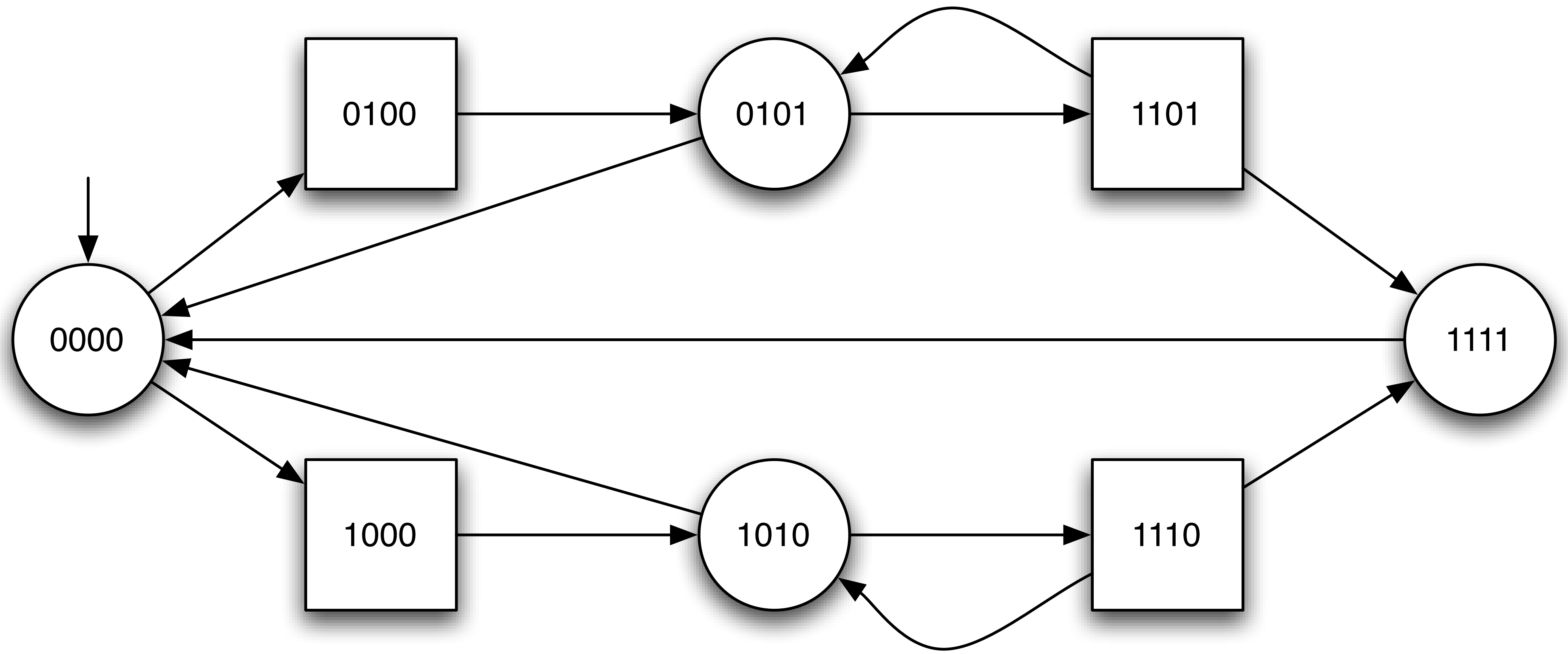
Who is winning ?



A **winning condition** (for Player 2)
is a set of plays

$$W \subseteq (Q_1 \cup Q_2)^\omega$$

Who is winning ?



A **winning condition** (for Player 2)
is a set of plays

A property !

Game

=

Two-player game structure

+

Winning condition for Player 2

Game

=

Two-player game structure

+

Winning condition for Player 2

The specification !

Strategies

Players are playing **according to strategies.**

$$\lambda_I(\underbrace{0011\ 1001\ 1101\ \overbrace{0011}^{\text{Player I's position}}}_{\text{prefix of play}})=\underbrace{1110}_{\text{Choice for the next position}}$$

$$\boxed{\lambda_I : S^{*}\bullet S_I \rightarrow S}$$

Strategies

Players are playing **according to strategies.**

Player I's
position

$\lambda_1(0011\ 1001\ 1101\ 0011)=1110$

prefix of play

Choice for
the n

Symmetrically for Player 2

S

Winning strategies

=

Controllers that enforce winning plays

=

Correct programs

Beyond the Basic Framework

Extensions

2 Players zero-sum games played on graphs with Boolean objectives

Extensions

2 Players zero-sum games played on graphs with Boolean objectives

Richer structures

-Real-time and clocks
-Infinite state spaces
-Observations (imperfect information)
-Quantitative information (weight/probabilities)

Extensions

2 Players zero-sum games played on graphs with Boolean objectives

Richer structures

-Real-time and clocks
-Infinite state spaces
-Observations (imperfect information)
-Quantitative information (weight/probabilities)

Richer objectives
Richer spec. languages

-Quantitative objectives
-LTL objectives
-ATL objectives

Extensions

2 Players zero-sum games played on graphs with Boolean objectives

Richer structures

-Real-time and clocks
-Infinite state spaces
-Observations (imperfect information)
-Quantitative information (weight/probabilities)

Richer objectives
Richer spec. languages

-Quantitative objectives
-LTL objectives
-ATL objectives

Richer solution concepts

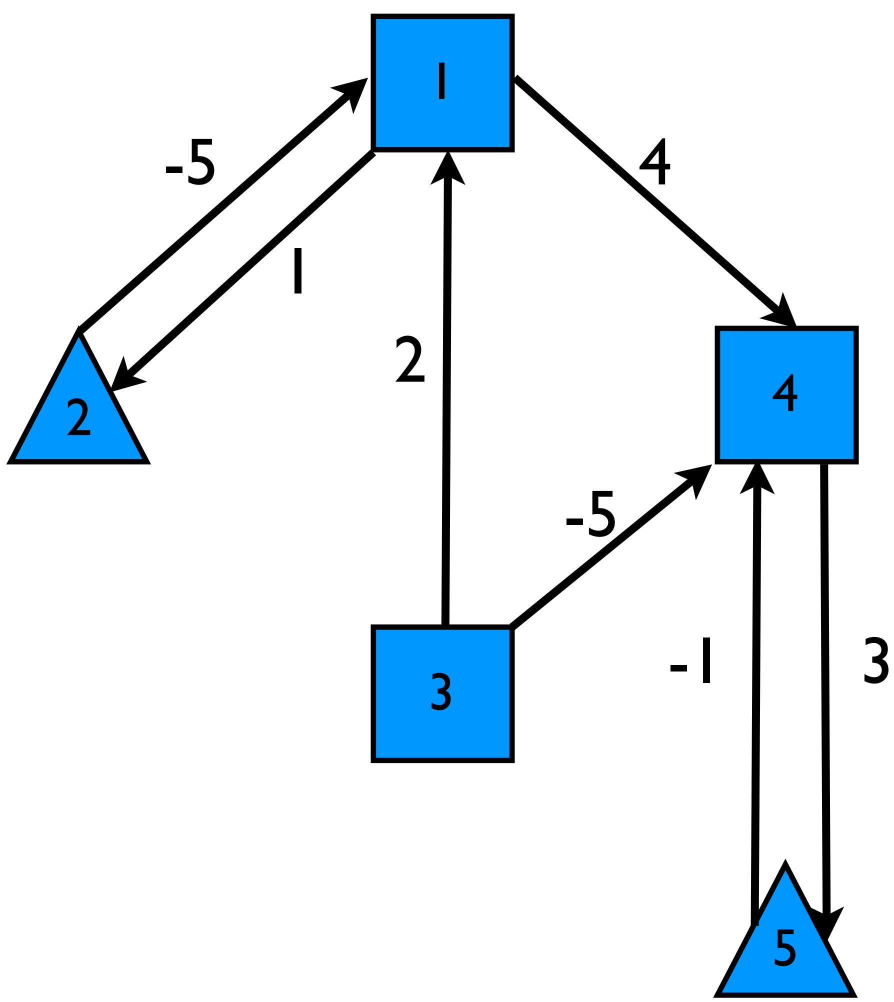
-Nash equilibria
-Regret minimization
-Observation based strategies
-Stutter-invariant strategies

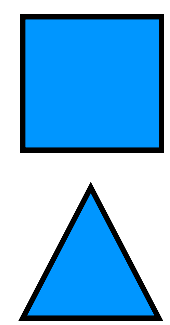
Highlights of Contributions

Highlights of Contributions

Quantitative Aspects of Games

Energy Games [UAA-LSV-ULB]





: positions of **maximizer**

: positions of **minimizer**

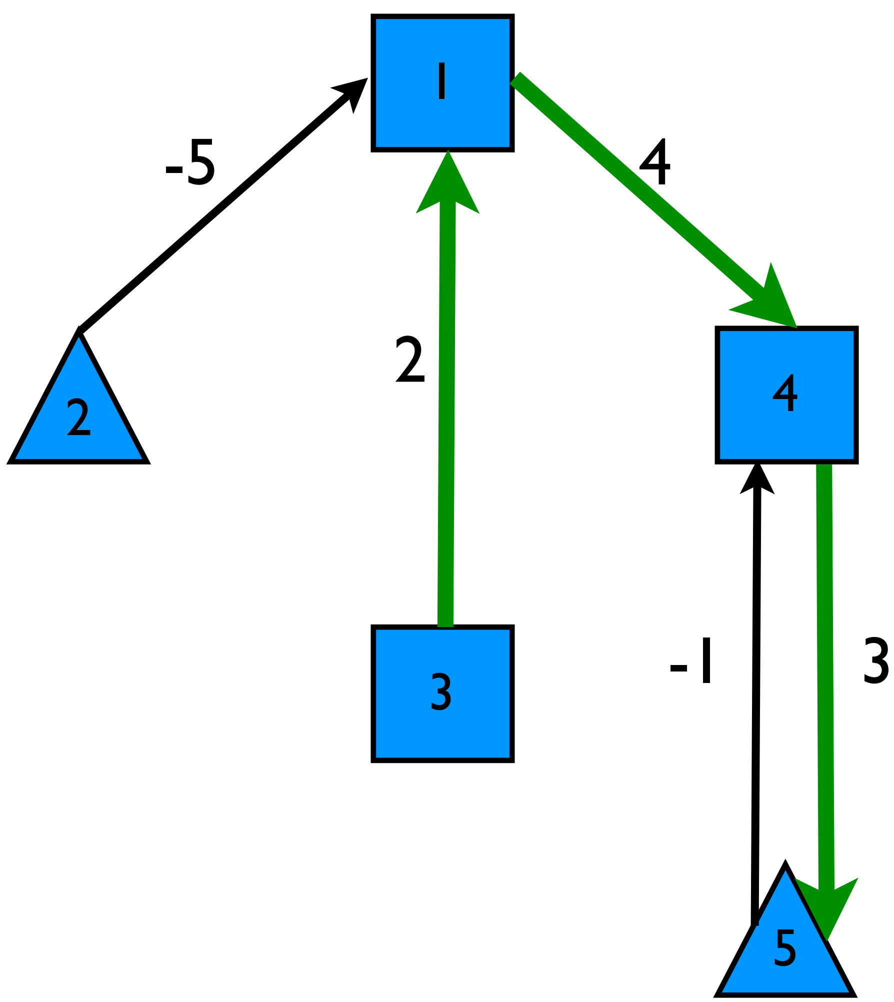
Edges are labelled with energy consumptions or energy gains

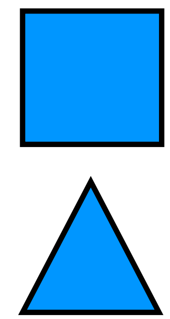
Initial energy level : 7

Play : (1,2) (2,1) (1,4) (4,5) (5,4) (4,5) (5,4) ...

EL : 7 8 3 7 10 9 10 9 ...

Energy Games [UAA-LSV-ULB]





: positions of **maximizer**

: positions of **minimizer**

Edges are labelled with energy consumptions or energy gains

Initial energy level : 7

Play : (1,2) (2,1) (1,4) (4,5) (5,4) (4,5) (5,4) ...

EL : 7 8 3 7 10 9 10 9 ...

Energy Games [UAA-LSV-ULB]

- Inter-reducibility (log-space) of EG and mean-payoff games (MP)
- **Simple fixed-point algorithm** for solving them
  - Improvement on known **pseudo**-polynomial time algorithms (open problem: existence of polynomial time solution ?)
  - From **O**($EV^3W \log E/V$) to **O**(EVW) for strategy synthesis
- Other important progresses on algorithms and theoretical understanding of Discounted-MP-Parity games **[Warwick]:**
  - Relation with the **linear complementarity problem**
  - Better understanding of **strategy improvement algorithms**

Energy Games [UAA-LSV-ULB]

- **Applications**: useful for modeling embedded systems
  - AAU+LSV studied real-time extensions, see more

    in **"UppAal Tiga"** by Prof. Kim Larsen
    Session 4 - Saturday 18:05
  - Further theoretical questions triggered by a case-study from another European project **Quasimodo** [AAU-ULB]
- Extensions:
  - Lower/Upper bounds [UAA-LSV]
  - Multi-dimensions [UAA-ULB].

More on Sunday !

Talk by Prof. Wolfgang Thomas (RWTH)

“Logic and infinite games: results and perspectives”

Sunday 9:40

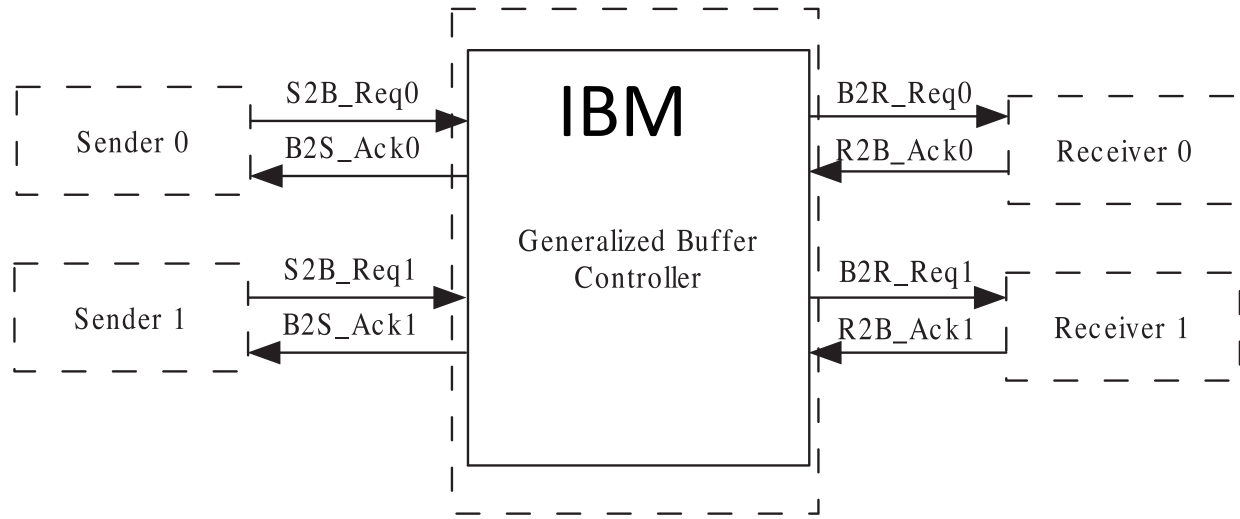
Highlights of Contributions

LTL Games

LTL Games - New algorithms [CFV-ULB]

```
## interface ##

.inputs s2b_req0 s2b_req1 s2b_ack0 s2b_ack1
.outputs b2s_ack0 b2s_ack1 b2s_req0 b2s_req1

################################################
[spec_unit sb_0]
################################################
assume s2b_req0=0;
assume G((s2b_req0=1 * b2s_ack0=0) -> X(s2b_req0=1));
assume G(b2s_ack0=1 -> X(s2b_req0=0));

b2s_ack0=0;
G((s2b_req0=0 * X(s2b_req0=1)) -> X(b2s_ack0=0 * X(F(b2s_ack0=1))));
G((b2s_ack0=0 * X(s2b_req0=0)) -> X(b2s_ack0=0));
G(b2s_ack0=0 + b2s_ack1=0);

################################################
[spec_unit sb_1]
################################################
assume s2b_req1=0;
assume G((s2b
assume

0) -> X(b2r_req0=1));
(b2r_req0=0)) ->
q0=0 U (b2r_req0=0 * b2r_req1=1)));
b2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

################################################
[spec_unit br_1]
################################################
assume r2b_ack1=0;
assume G(b2r_req1=0 -> X(r2b_ack1=0));
assume G(b2r_req1=1 -> X(F(r2b_ack1=1)));

b2r_req1=0;
G(r2b_ack1=1 -> X(b2r_req1=0));
G((b2r_req1=1 * r2b_ack1=0) -> X(b2r_req1=1));
G((b2r_req1=1 * X(b2r_req1=0)) ->
   X(b2r_req1=0 U (b2r_req1=0 * b2r_req0=1)));
G((b2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

group_order = (sb_0 br_1) (sb_1 br_0);
```

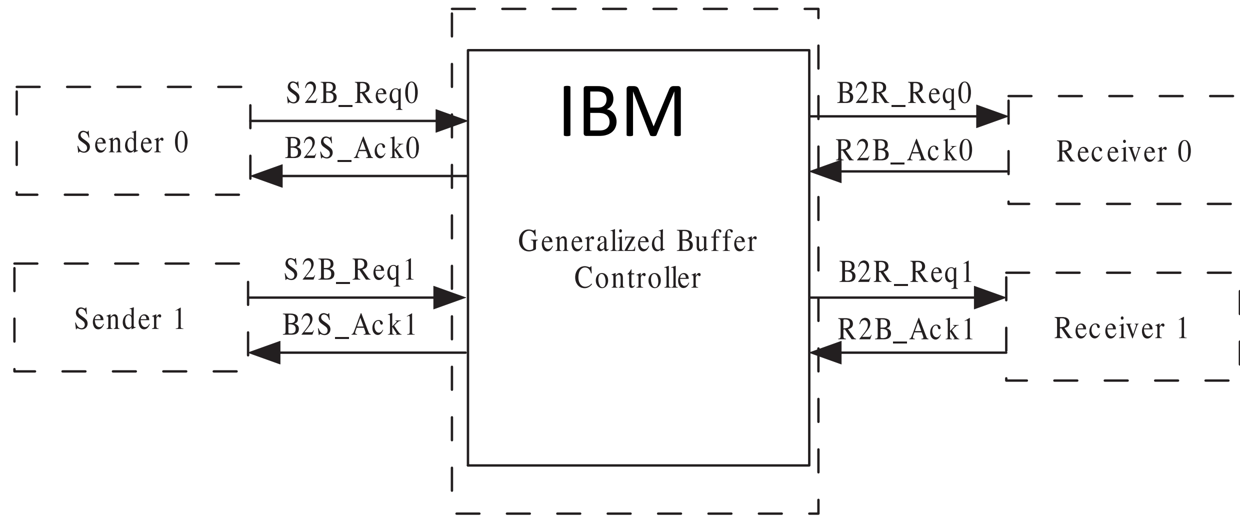
Specification:
LTL Formula $\land \psi_i$



Synthesis of an Implementation

Theory: Pnueli-Rosner89
2ExpTimeC problem

LTL Games - New algorithms [CFV-ULB]




```
## interface ##

.inputs s2b_req0 s2b_req1 s2b_ack0 s2b_ack1
.outputs b2s_ack0 b2s_ack1 b2s_req0 b2s_req1

##################################################
[spec_unit sb_0]
##################################################
assume s2b_req0=0;
assume G((s2b_req0=1 * b2s_ack0=0) -> X(s2b_req0=1));
assume G(b2s_ack0=1 -> X(s2b_req0=0));

b2s_ack0=0;
G((s2b_req0=0 * X(s2b_req0=1)) -> X(b2s_ack0=0 * X(F(b2s_ack0=1))));
G((b2s_ack0=0 * X(s2b_req0=0)) -> X(b2s_ack0=0));
G(b2s_ack0=0 + b2s_ack1=0);

##################################################
[spec_unit sb_1]
##################################################
assume s2b_req1=0;
assume G((s2b
assume
```

Specification:
LTL Formula $\wedge \psi_i$

```
0) -> X(b2r_req0=1));
(b2r_req0=0)) ->
q0=0 U (b2r_req0=0 * b2r_req1=1)));
2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

##################################################
[spec_unit br_1]
##################################################
assume r2b_ack1=0;
assume G(b2r_req1=0 -> X(r2b_ack1=0));
assume G(b2r_req1=1 -> X(F(r2b_ack1=1)));
```

Synthesis of an Implementation

Theoretical bound
on size of solutions:
2^(2^50)=2^1125899906842624

Theory: Pnueli-Rosner89
2ExpTimeC problem

LTL Games - New algorithms [CFV-ULB]

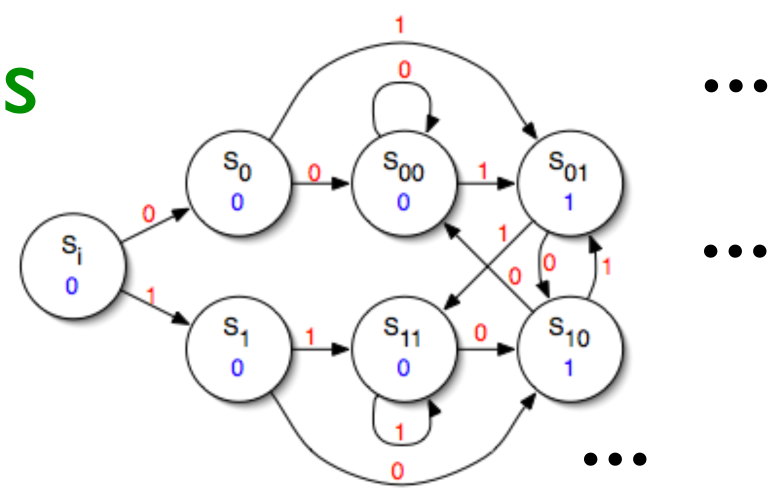
```
## interface ##

.inputs s2b_req0 s2b_req1 s2b_ack0 s2b_ack1
.outputs b2s_ack0 b2s_ack1 b2s_req0 b2s_req1

################################################
[spec_unit sb_0]
################################################
assume s2b_req0=0;
assume G((s2b_req0=1 * b2s_ack0=0) -> X(s2b_req0=1));
assume G(b2s_ack0=1 -> X(s2b_req0=0));

b2s_ack0=0;
G((s2b_req0=0 * X(s2b_req0=1)) -> X(b2s_ack0=0 * X(F(b2s_ack0=1))));
G((b2s_ack0=0 * X(s2b_req0=0)) -> X(b2s_ack0=0));
G(b2s_ack0=0 + b2s_ack1=0);

[spec_unit sb_1]
assume s2b_req1=0;
...
0) -> X(b2r_req0=1));
b2r_req0=0)) ->
eq0=0 U (b2r_req0=0 * b2r_req1=1)));
b2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

################################################
[spec_unit br_1]
################################################
assume r2b_ack1=0;
assume G(b2r_req1=0 -> X(r2b_ack1=0));
assume G(b2r_req1=1 -> X(F(r2b_ack1=1)));

b2r_req1=0;
G(r2b_ack1=1 -> X(b2r_req1=0));
G((b2r_req1=1 * r2b_ack1=0) -> X(b2r_req1=1));
G((b2r_req1=1 * X(b2r_req1=0)) ->
   X(b2r_req1=0 U (b2r_req1=0 * b2r_req0=1)));
G((b2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

group_order = (sb_0 br_1) (sb_1 br_0);
```

Specification:
LTL Formula $\wedge \psi_i$

Recent progresses:
Safraless methods (Vardi et al.)
Symbolic data-structure: antichains

Acacia Tool (ULB) 2009-...

Good news:
46 states
0.5 seconds



LTL Games - New algorithms [CFV-ULB]

```
## interface ##

.inputs s2b_req0 s2b_req1 s2b_ack0 s2b_ack1
.outputs b2s_ack0 b2s_ack1 b2s_req0 b2s_req1

################################################
[spec_unit sb_0]
################################################
assume s2b_req0=0;
assume G((s2b_req0=1 * b2s_ack0=0) -> X(s2b_req0=1));
assume G(b2s_ack0=1 -> X(s2b_req0=0));

b2s_ack0=0;
G((s2b_req0=0 * X(s2b_req0=1)) -> X(b2s_ack0=0 * X(F(b2s_ack0=1))));
G((b2s_ack0=0 * X(s2b_req0=0)) -> X(b2s_ack0=0));
G(b2s_ack0=0 + b2s_ack1=0);

################################
[spec_unit sb_1]
##########################
assume s2b_req1=0;
assume G((s2b
assume
```

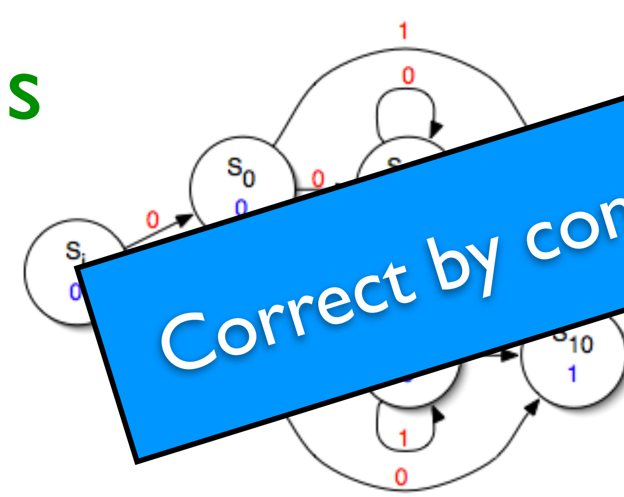
Specification:
LTL Formula $\wedge \psi_i$

```
0) -> X(b2r_req0=1));
(b2r_req0=0)) ->
eq0=0 U (b2r_req0=0 * b2r_req1=1)));
b2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

################################################
[spec_unit br_1]
################################################
assume r2b_ack1=0;
assume G(b2r_req1=0 -> X(r2b_ack1=0));
assume G(b2r_req1=1 -> X(F(r2b_ack1=1)));

b2r_req1=0;
G(r2b_ack1=1 -> X(b2r_req1=0));
G((b2r_req1=1 * r2b_ack1=0) -> X(b2r_req1=1));
G((b2r_req1=1 * X(b2r_req1=0)) ->
   X(b2r_req1=0 U (b2r_req1=0 * b2r_req0=1)));
G((b2r_req0=0) + (b2r_req1=0) );
G((s2b_req0=1 + s2b_req1=1) -> X(F(b2r_req0=1 + b2r_req1=1)));

group_order = (sb_0 br_1) (sb_1 br_0);
```

Recent progresses:
Safraless methods (Vardi et al.)
Symbolic data-structure: antichains

Acacia Tool (ULB) 2009-...

Good news:
46 states
0.5 seconds



Correct by construction !

LTL + Parameters [RWTH]

- LTL with parameters:

  $$\psi \equiv \Box\,(\,\text{req} \Longrightarrow \Diamond_{\leq \mathbf{x}}\,\text{grant}\,)$$

- Question:

  Given a game structure G, for which **values** of the parameter x in objective ψ can Player I win the game ?

- Complexity:

  Surprisingly **same** complexity as plain LTL (2ExpTime)

Alternating Time-Temporal Logic

- Logic to speak about strategic behavior of agents
- Introduced by Alur and Henzinger in the end of 90s
- Extensions studied in GASICS:

“ATL and extensions”

Nicolas Markey (LSV)

Session 4 - Saturday - 17:15

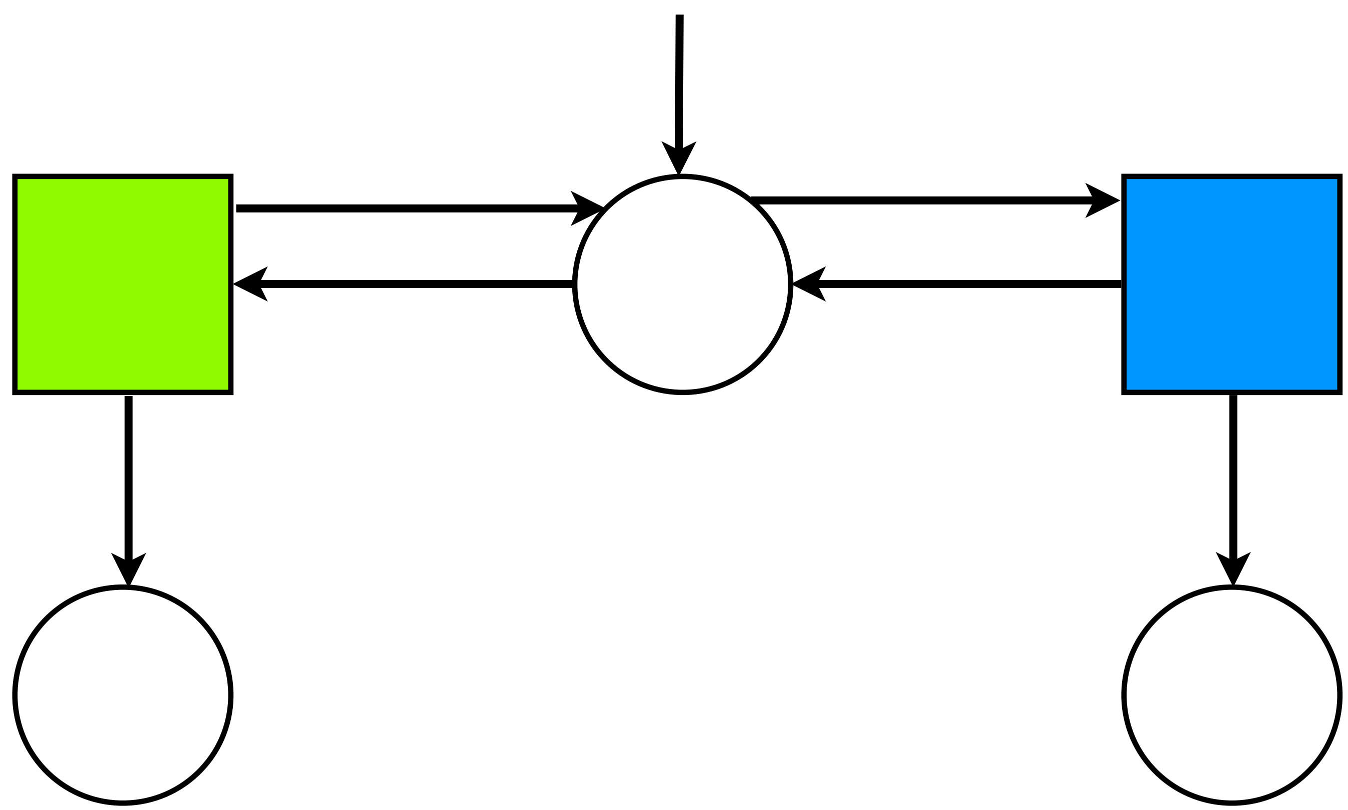
Highlights of Contributions

Nash Equilibria in Game Graphs

Non-Zero Sum Games - NE

□ Player 1 : □ ◇ **green**

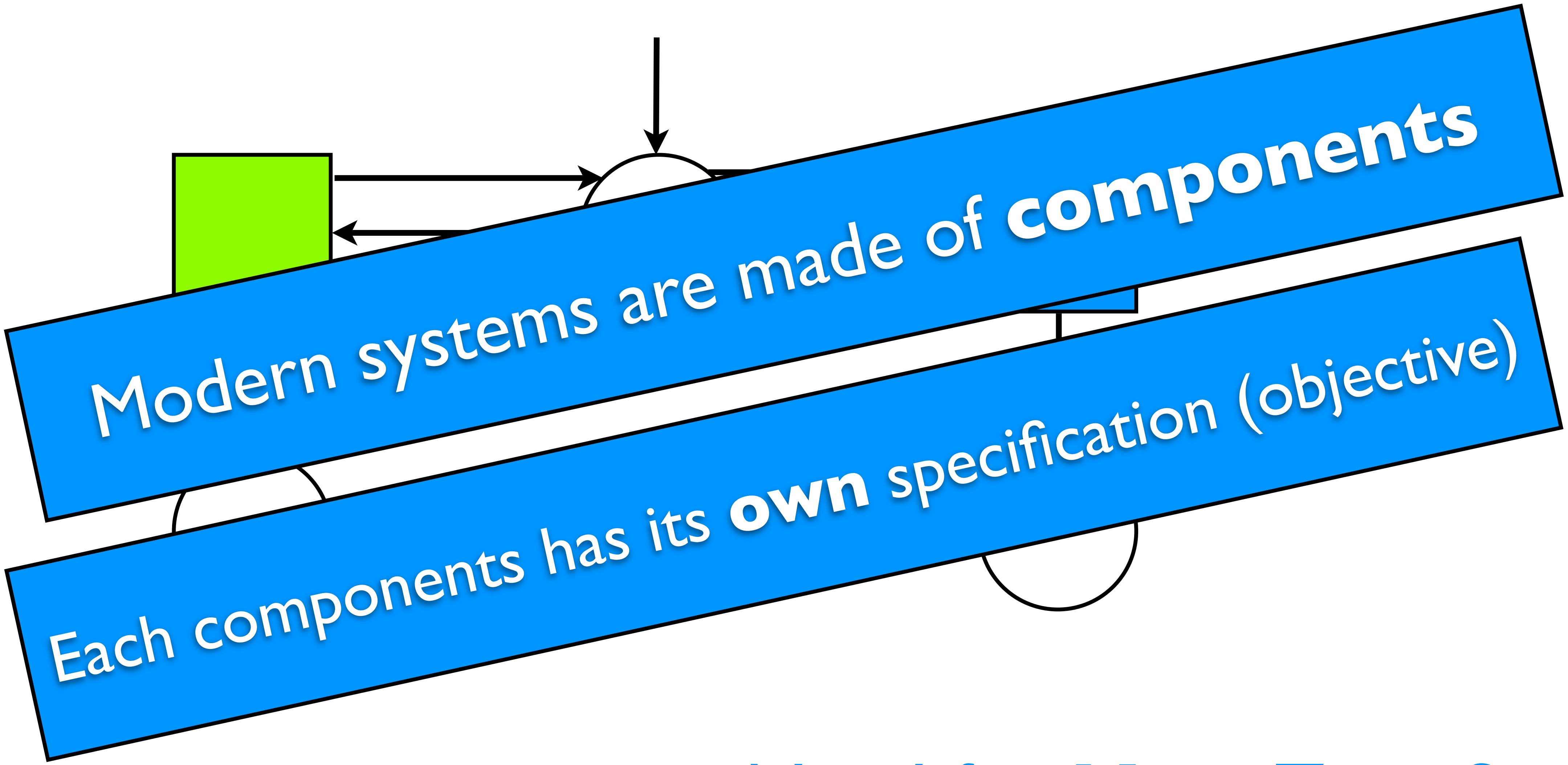
○ Player 2 : □ ◇ **blue**



Non-Zero Sum Games - NE

□ Player 1 : □◇ **green**

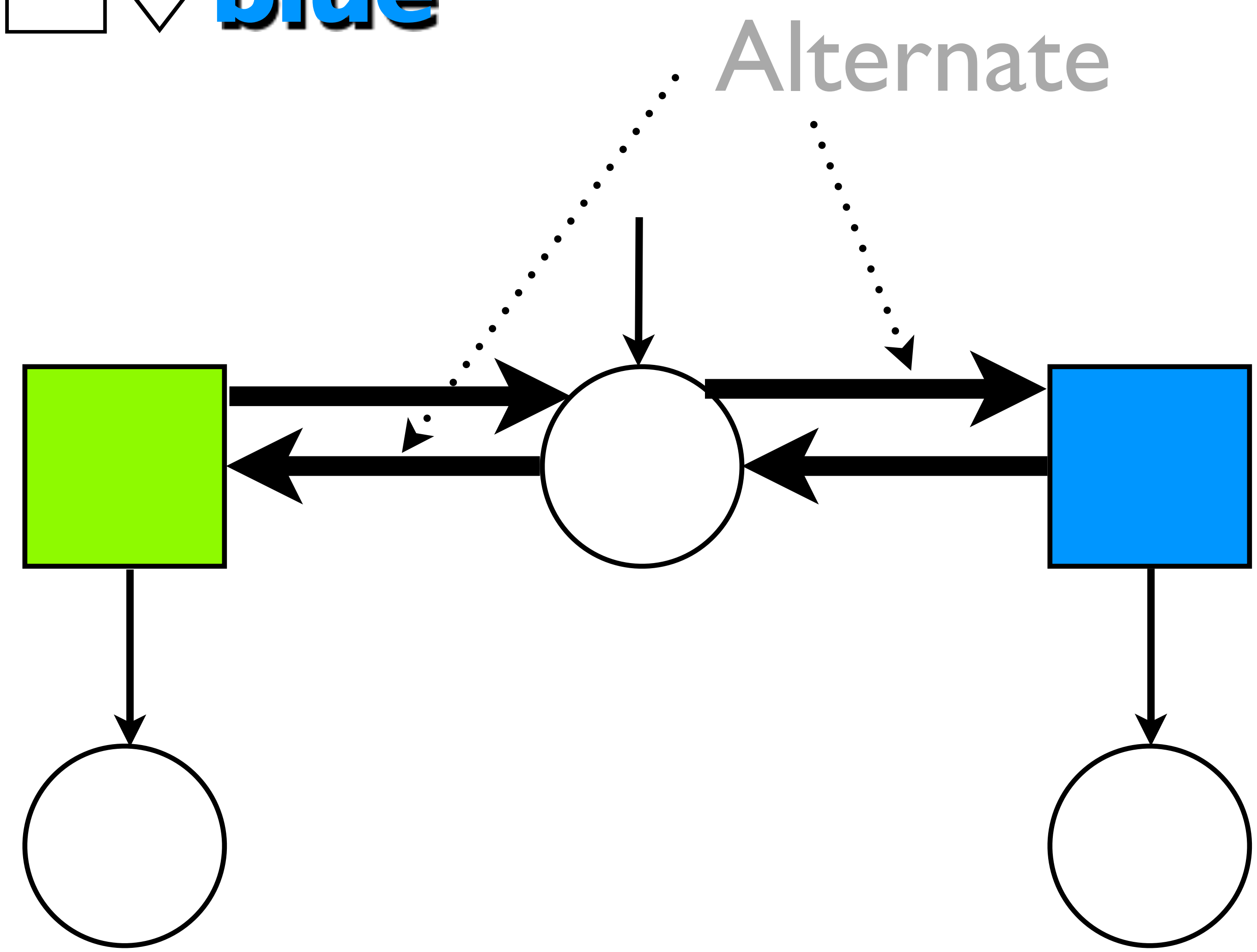
○ Player 2 : □◇ **blue**



>>> Need for **Non**-Zero Sum

Non-Zero Sum Games - NE

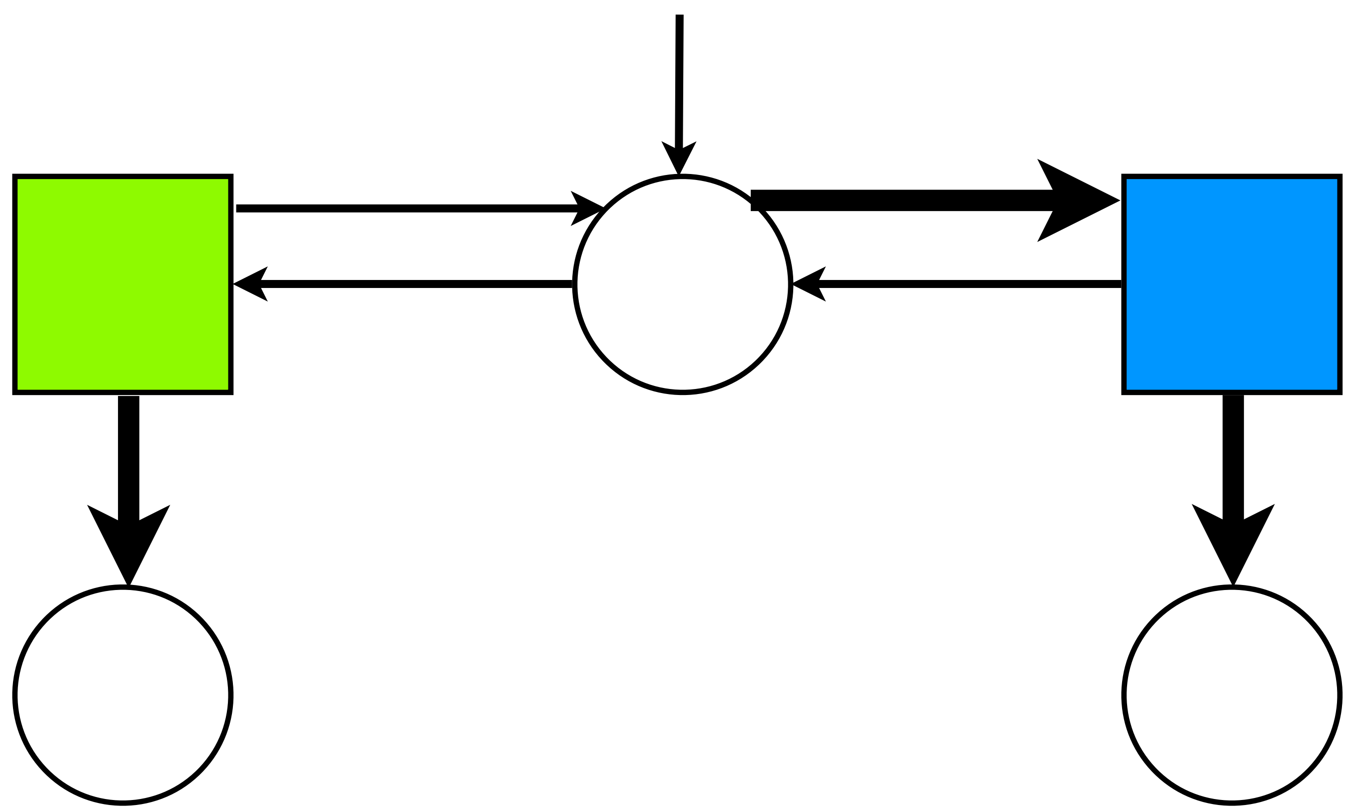
- Player 1 : ☐ ◇ **green**
- Player 2 : ☐ ◇ **blue**



NE as (1,1)

Non-Zero Sum Games - NE

- □ Player 1 : □◇ **green**
- ○ Player 2 : □◇ **blue**



A less interesting NE: (0,0)

No improvement by unilateral change

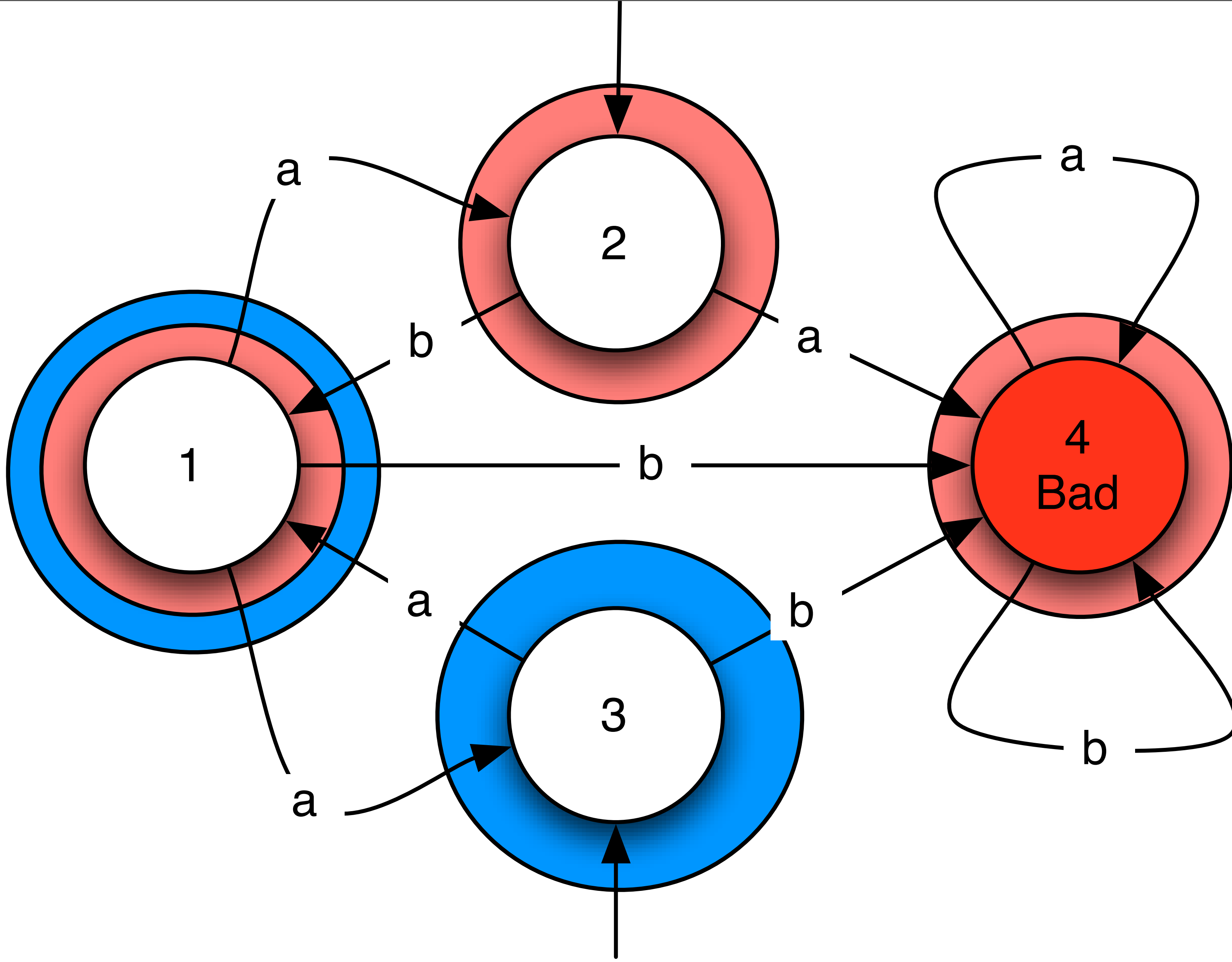
Non-Zero Sum Games - NE

- Do Nash Equilibria always **exist** in finite game graphs ?
  Yes for ω-regular objectives [GraëdelUmmels-**LINT**]
  - Is it the case for **quantitative** reachability objectives ?
    Yes, more details in

    "Nash equilibrium in quantitative games played on graphs"
    by Ms. Julie de Pril [CFV-UMons] - Session 4 - Saturday 17:40.
    - **Finite memory** is sufficient
- NE for timed games (∞-state systems) by [LSV-RWTH]
- Alternatives to NE: Regret minimization [CFV-ULB]

Highlights of Contributions

Games with Imperfect Information



$$q_0 = \top$$

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

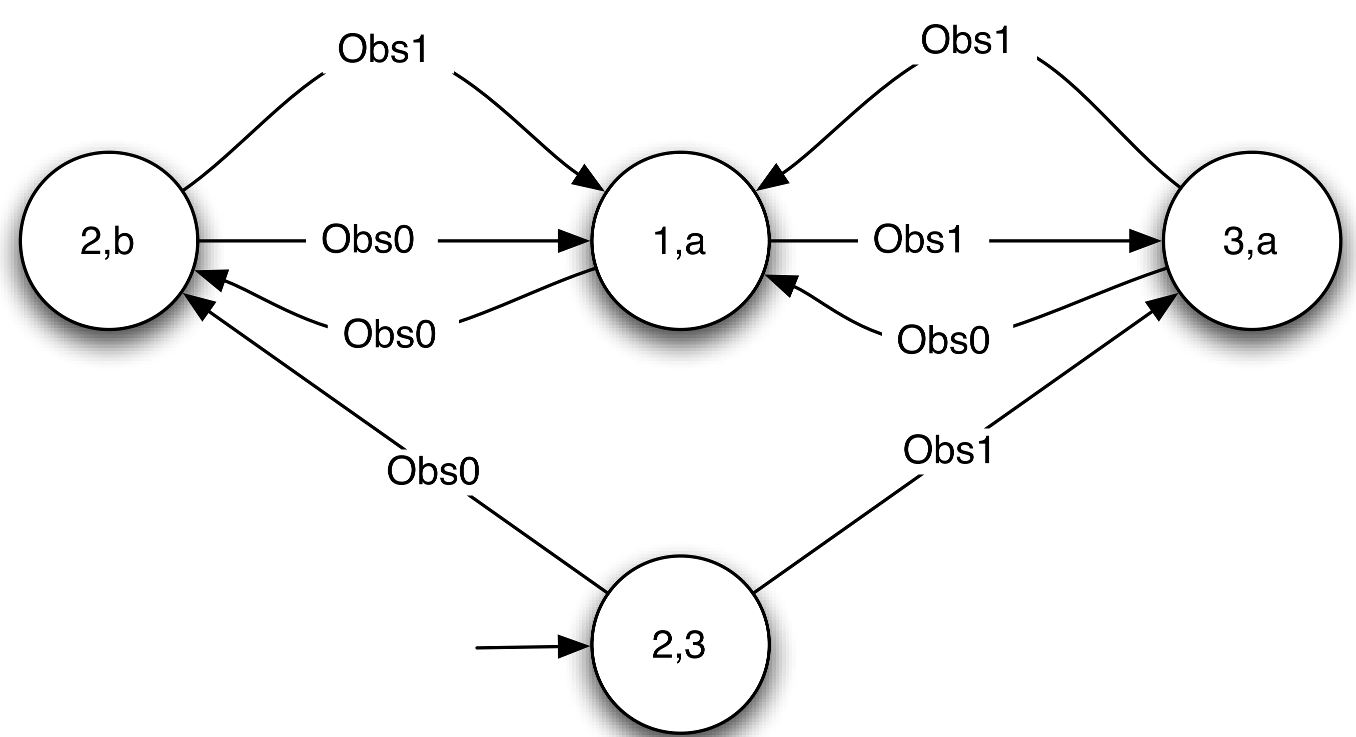
$$q_2 = \{\{2\}_b, \{1,3\}_a\}$$

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

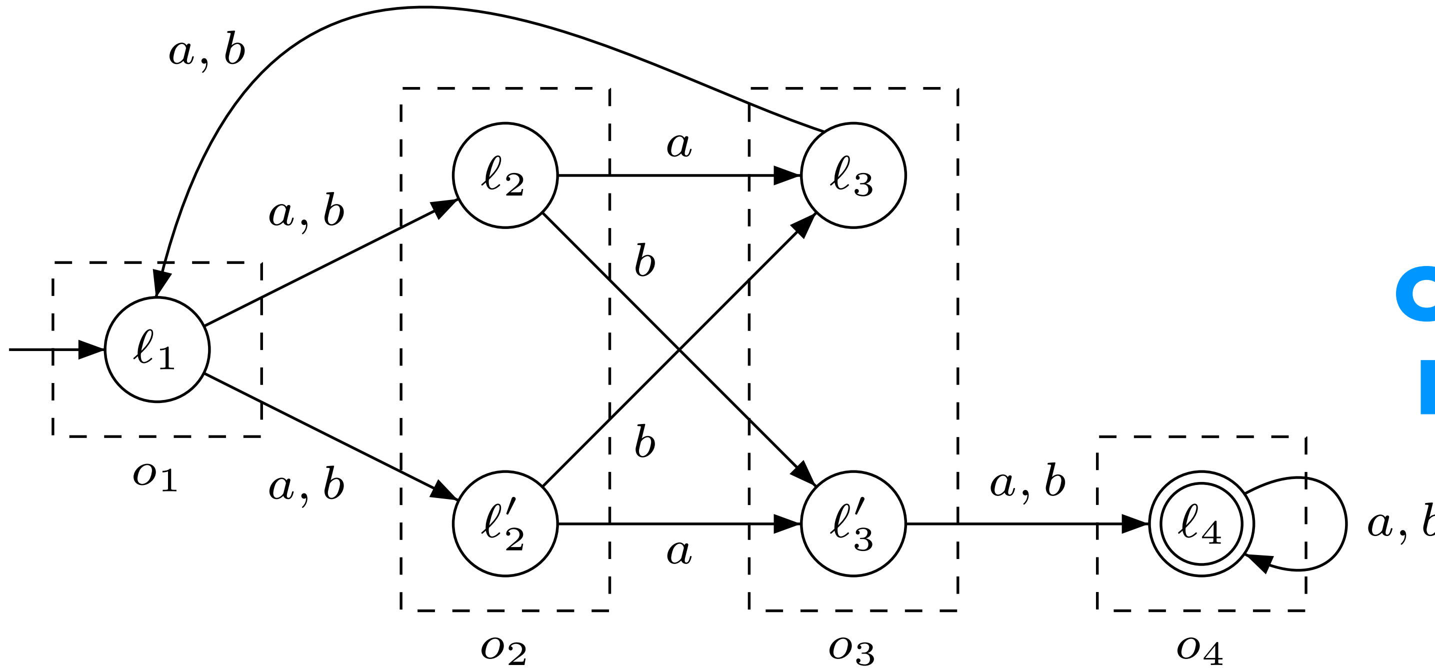
$$q_4 = \{\{1\}_a, \{2\}_b, \{3\}_a\}$$

Fixed point

Need for memory:



Randomized Strategies



Objective: Reach O_4

Consider Player 1 playing this simple following **randomized** strategy: when receiving observation o_2, play uniformly at random a and b.

Clearly, each time that it enters o_2, the probability to reach l_3' in the next round is 1/2. In **the long run**, the probability to reach l_3', and thus l_4, is 1.

We say that Player 1 **almost-surely** wins the reachability game (probability 1).

Randomized strategies are **more powerful** than deterministic strategies for reachability.

Games with Imperfect Information

- **Memory** and **randomization** are necessary for winning games with imperfect information (even for reachability objectives)
- Semantics: Player 2 perfectly informed or not [LIAFA]

- Symbolic fixed-point algorithms [CFV-ULB-LSV]
- Decidability/Undecidability frontier [LIAFA-CFV ULB-LSV]
- Relation with tree automata [LIAFA]

Conclusion

- Progresses:
  - From basic model to **richer** models
  - on **theory**, on **algorithms**, and towards **applications**
  - More on http://www.ulb.ac.be/di/gasics/ (**91** published papers)

Future Works

- We need to better understand:
  - which **solution concepts** are needed for synthesis of complex reactive systems
  - how to import techniques from verification:
    - abstraction/approximation
    - compositional reasoning
    - ...

Thanks for your attention!