An Quick Introduction to Controller Synthesis

Moves - April 20, 2007

JF Raskin
U.L.B.

Context

- Make a model of the environment

  Environment

- Make clear the control objective:

  Bad

- Make a model of your control strategy:

  ControllerMod

- Verify :

  Does Environment || ControllerMod avoid Bad ?

Context

- Make a model of the environment

Environment

- Make clear the control objective:

Bad

Make the algorithmic synthesis of Controller

- ~~Verify~~ :

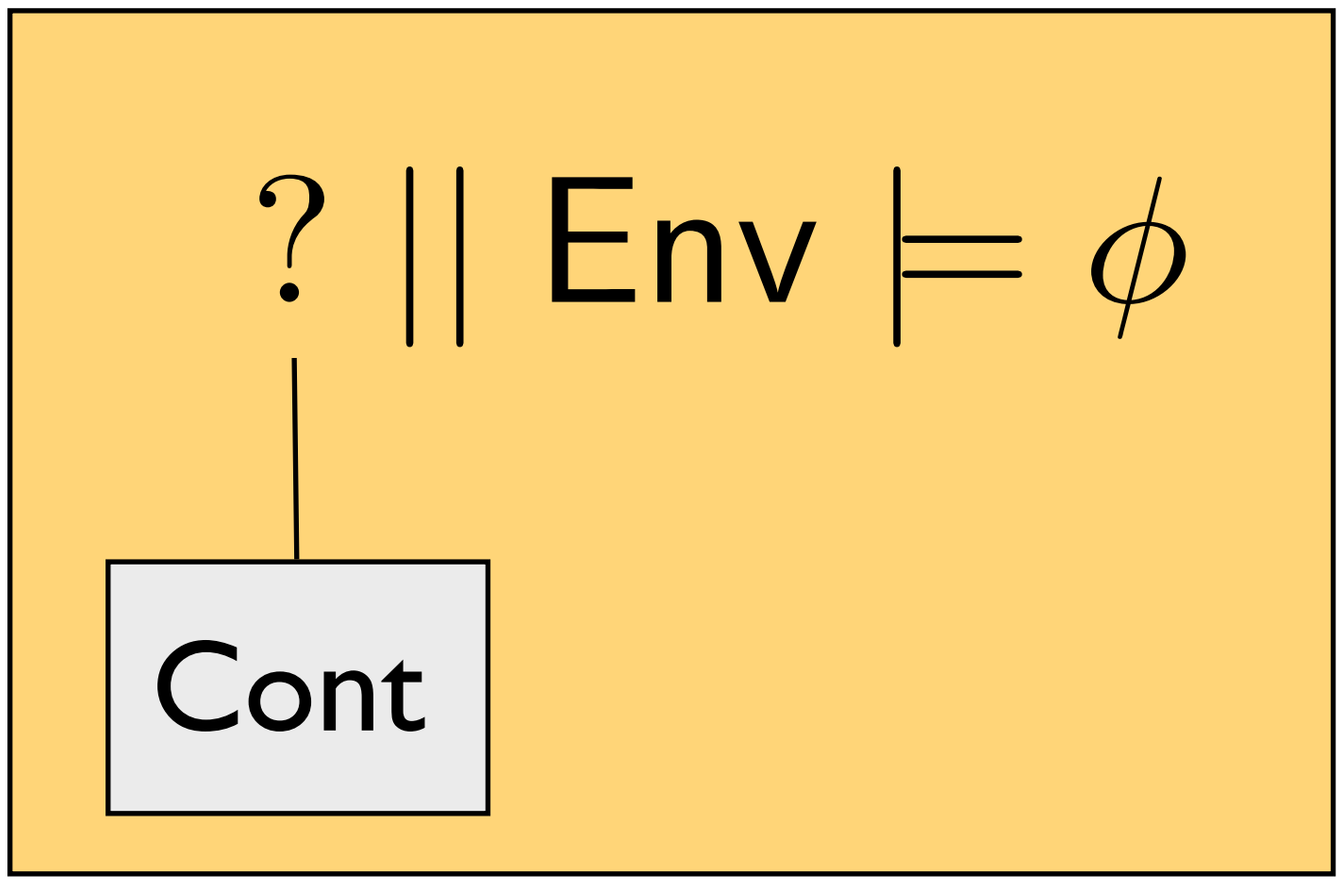
~~Does Environment ‖ ControllerMod avoid Bad ?~~

The synthesis problem

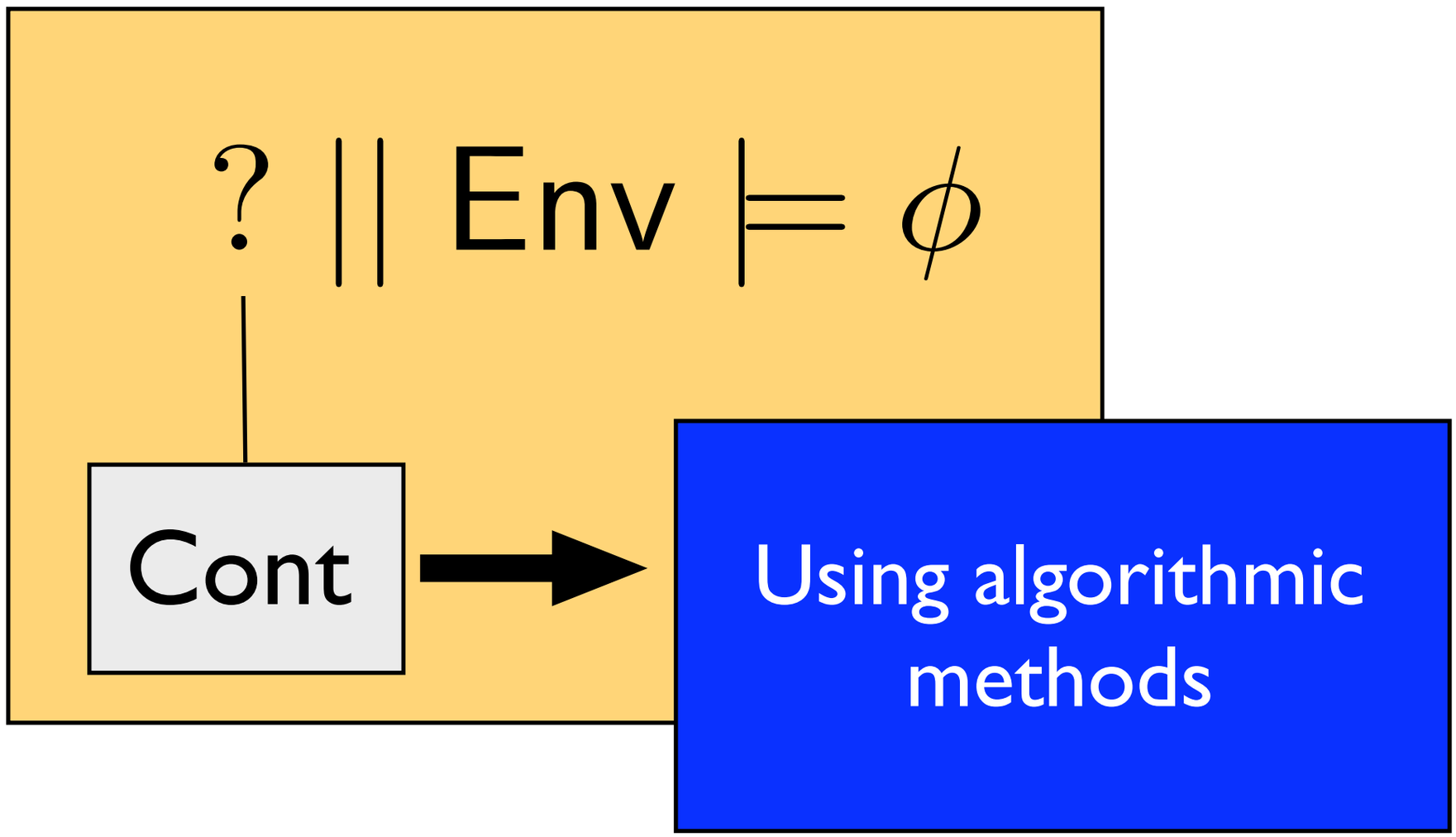
The synthesis problem

$$? \;||\; \text{Env} \models \phi$$

The synthesis problem



The synthesis problem



The synthesis problem

Specialize process A into C such that

$$A \geq C \text{ and } C \,||\, B \models \phi$$

So, C must refine A and
control B to **enforce** ϕ

Basic technics: finite state case

Are transition systems adequate for synthesis ?

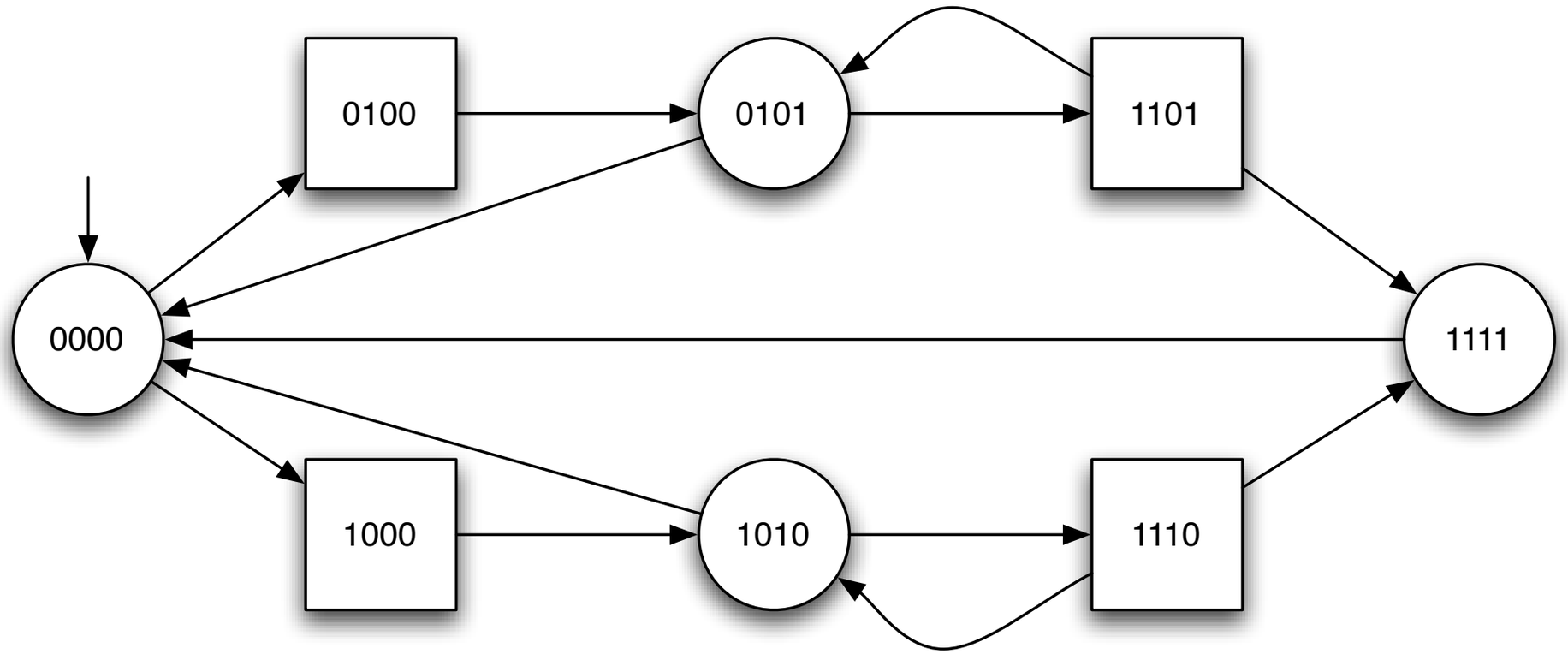
- For the verification problem, the semantics of processes is usually given by **transition systems**
- When we consider the transition system for A || B, we loose the information about the **components**

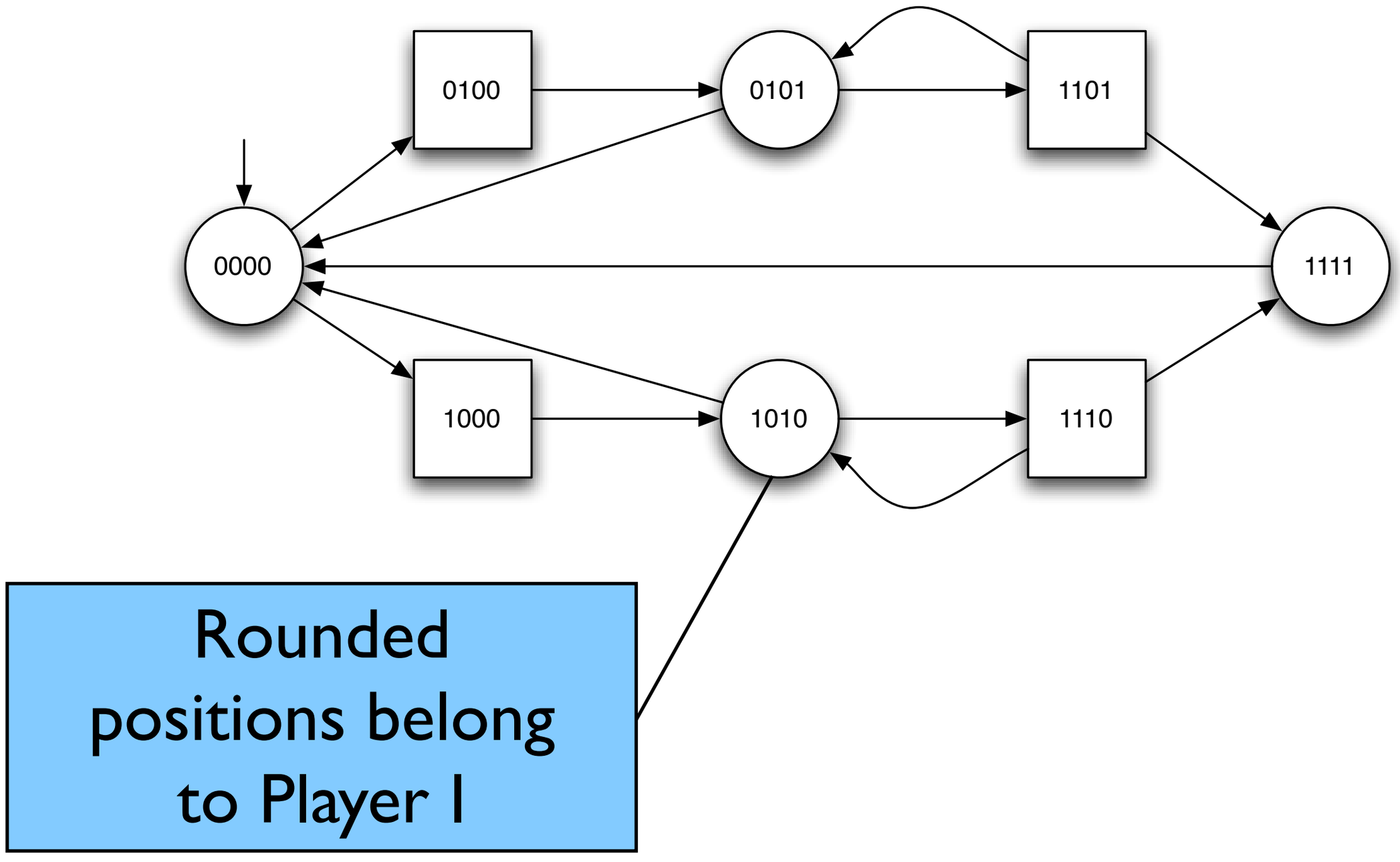
Are transition systems adequate for synthesis ?

- For the verification problem, the semantics of processes is usually given by **transition systems**
- When we consider the transition system for A || B, we loose the information about the **components**

So, we need richer models where **identities** of processes are explicit:
two-player game structures

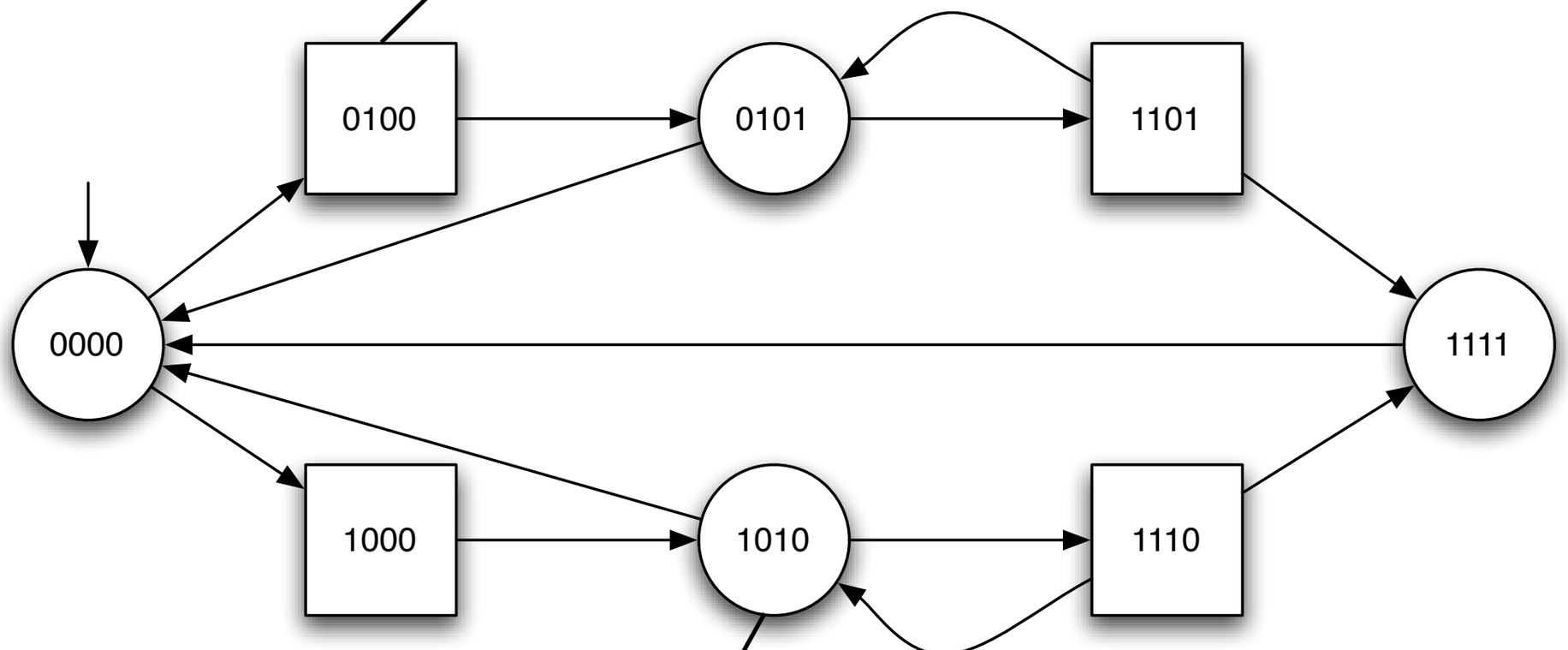
Two-player game structures


0100
0101
1101
0000
1111
1000
1010
1110



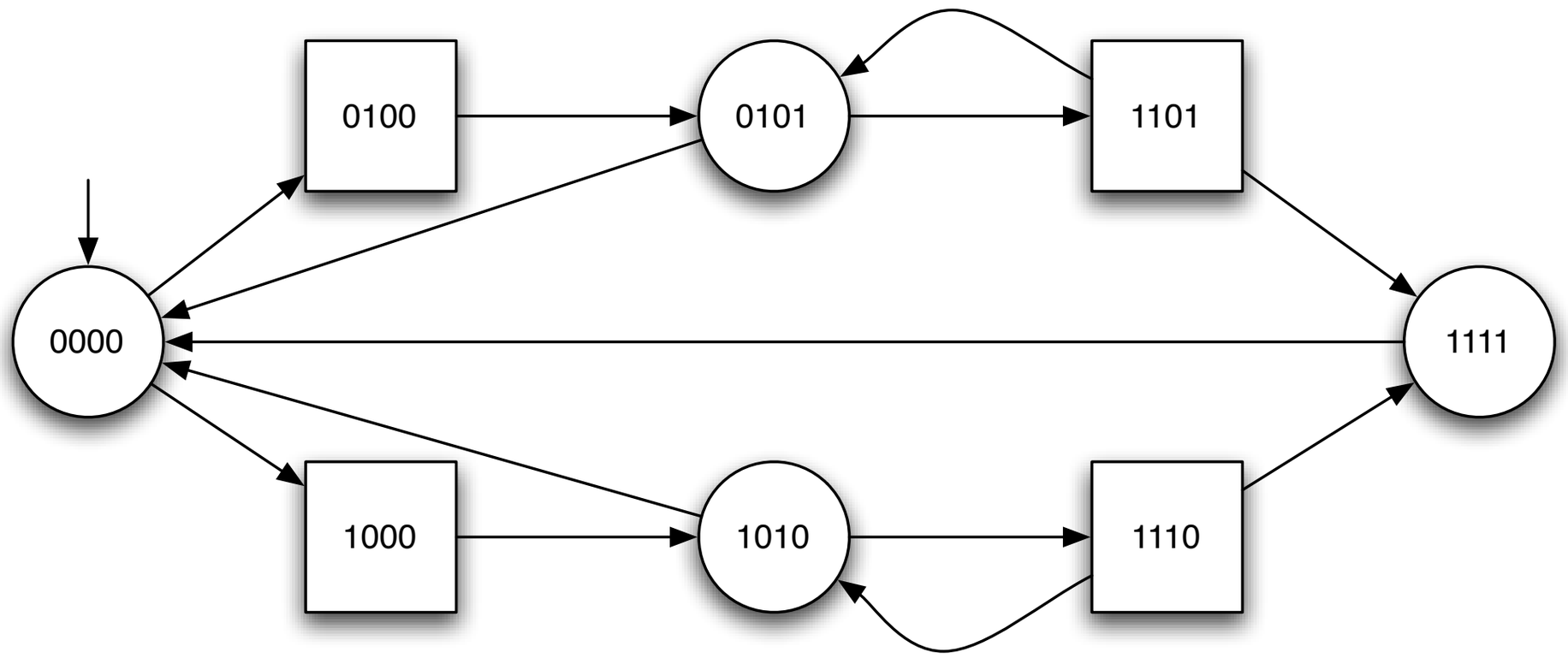
Rounded positions belong to Player I

Square positions belong to Player 2

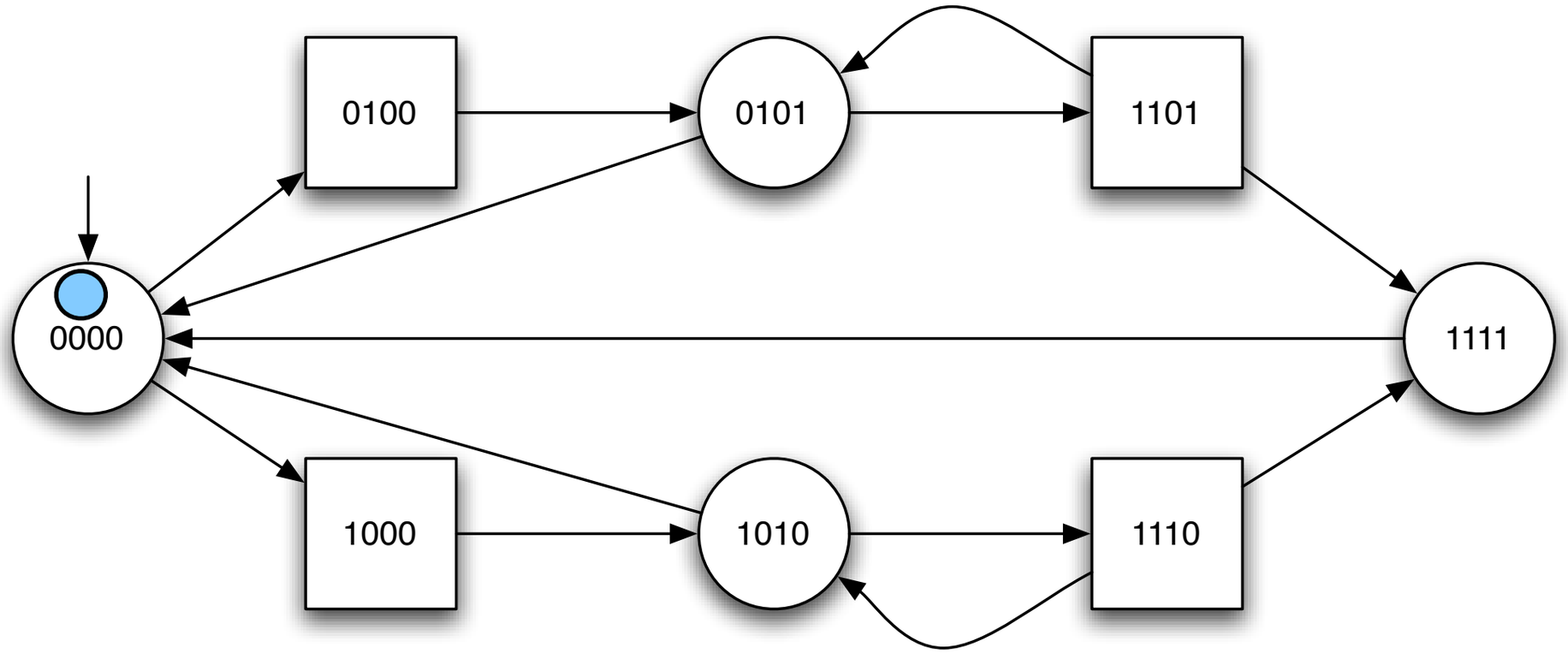


Rounded positions belong to Player 1

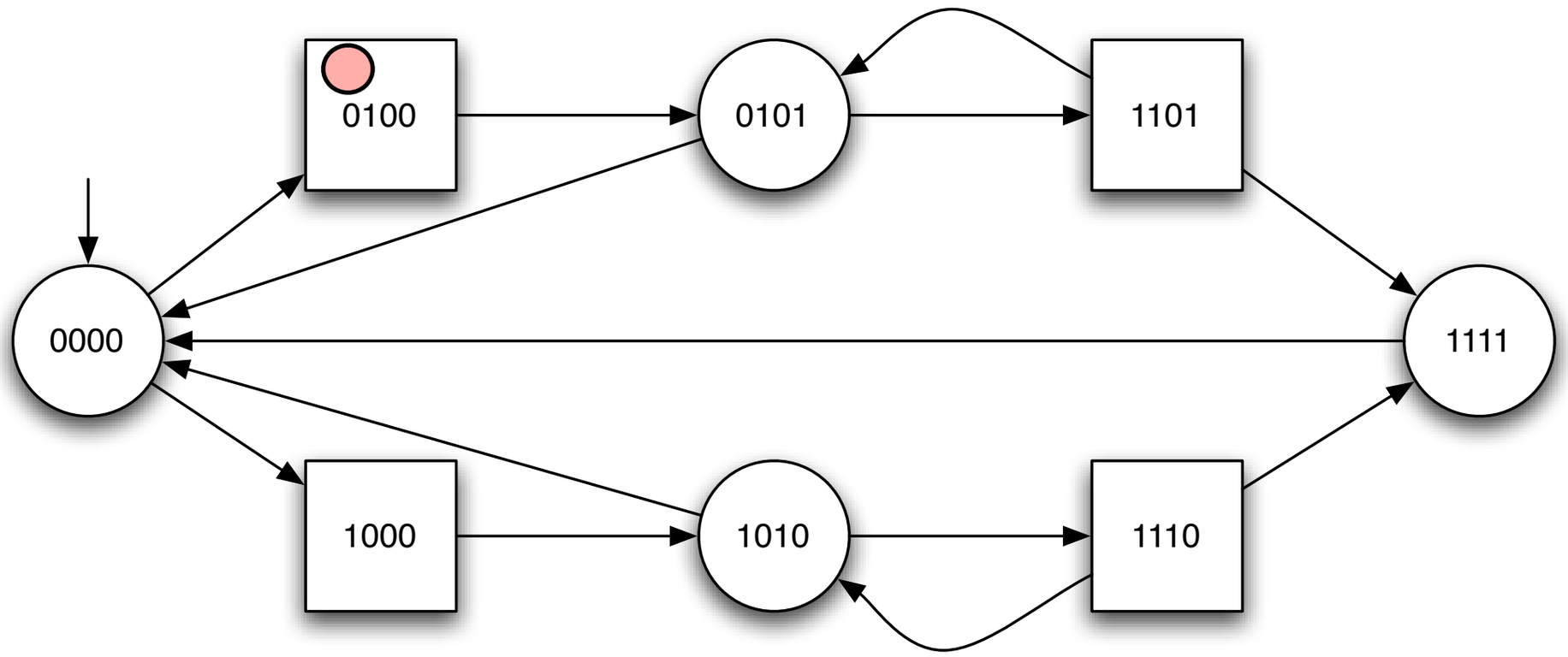
Rounded positions belong to Player 1
Square positions belong to Player 2



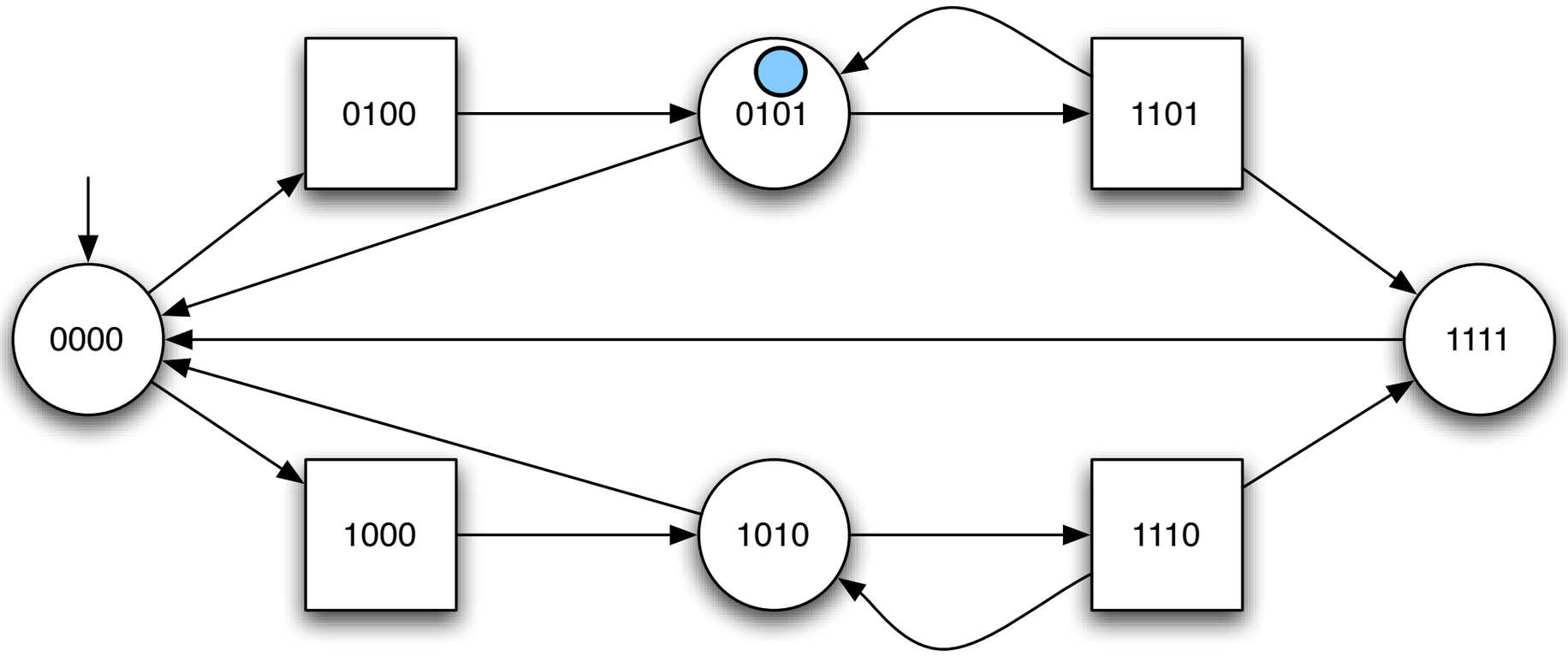
A game is played as follows: in each **round**, the game is in a **position**, if the game is in a rounded position, Player I resolves the **choice** for the next state, if the game is in a square position, Play 2 resolves the choice. The game is played for an **infinite number of rounds**.



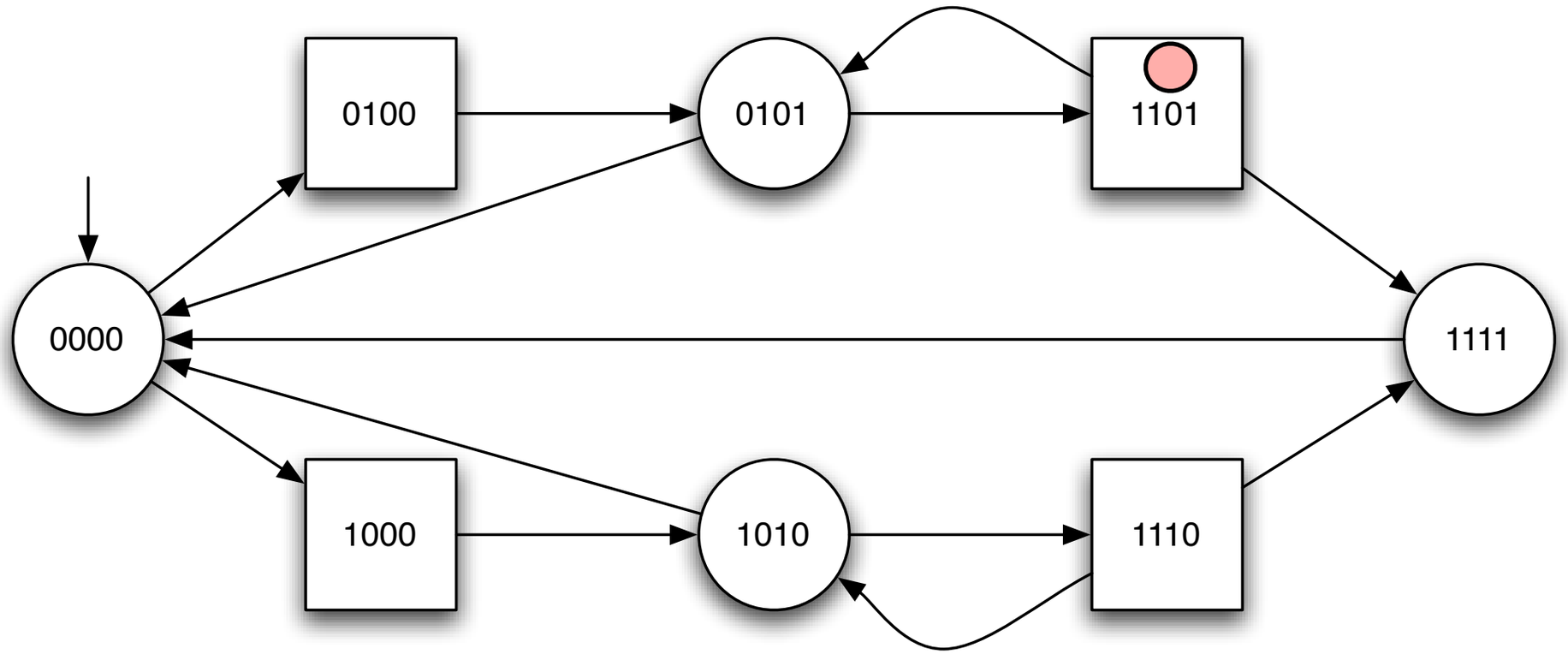
Play : 0000



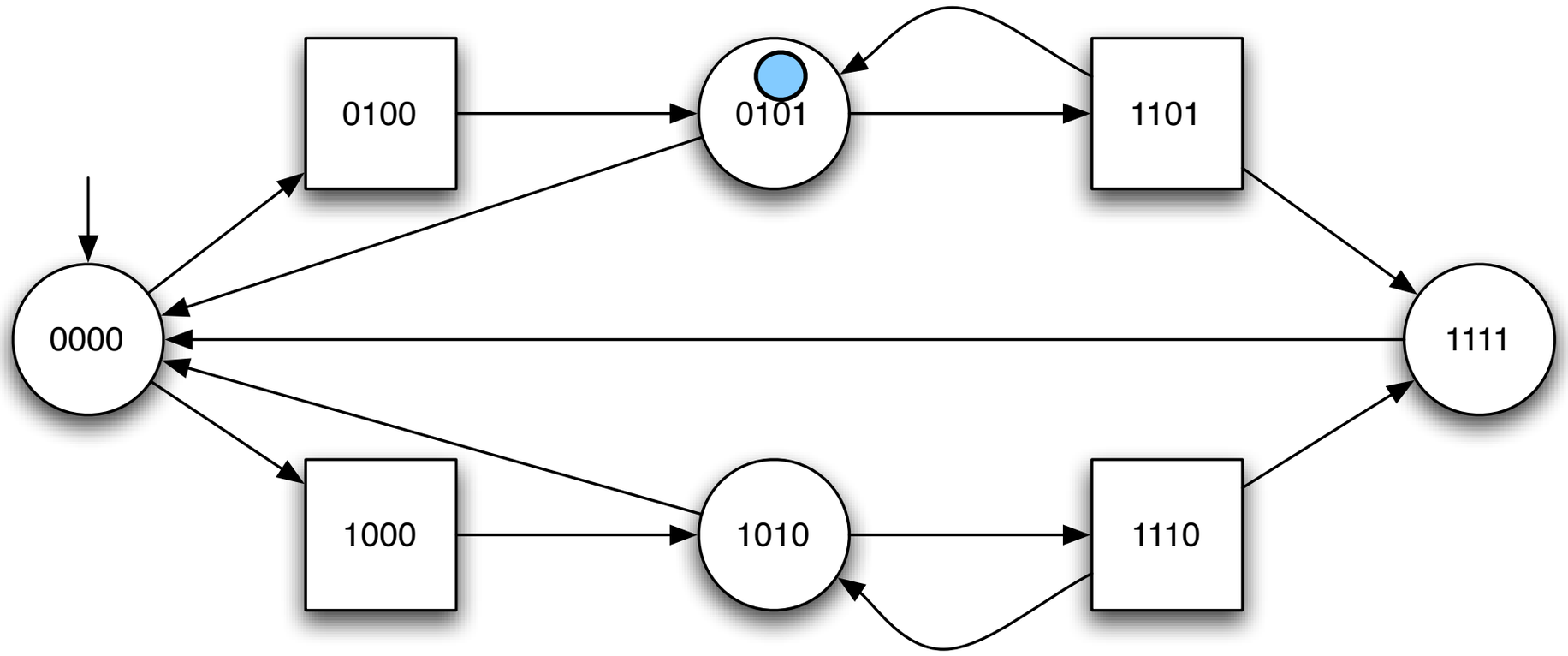
Play : 0000 0100



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

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

Plays, Prefixes of Plays

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

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

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

Plays, Prefixes of Plays

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

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

Notations

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

$w(i)$ denotes position *i*

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

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

Plays, Prefixes of Plays

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

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

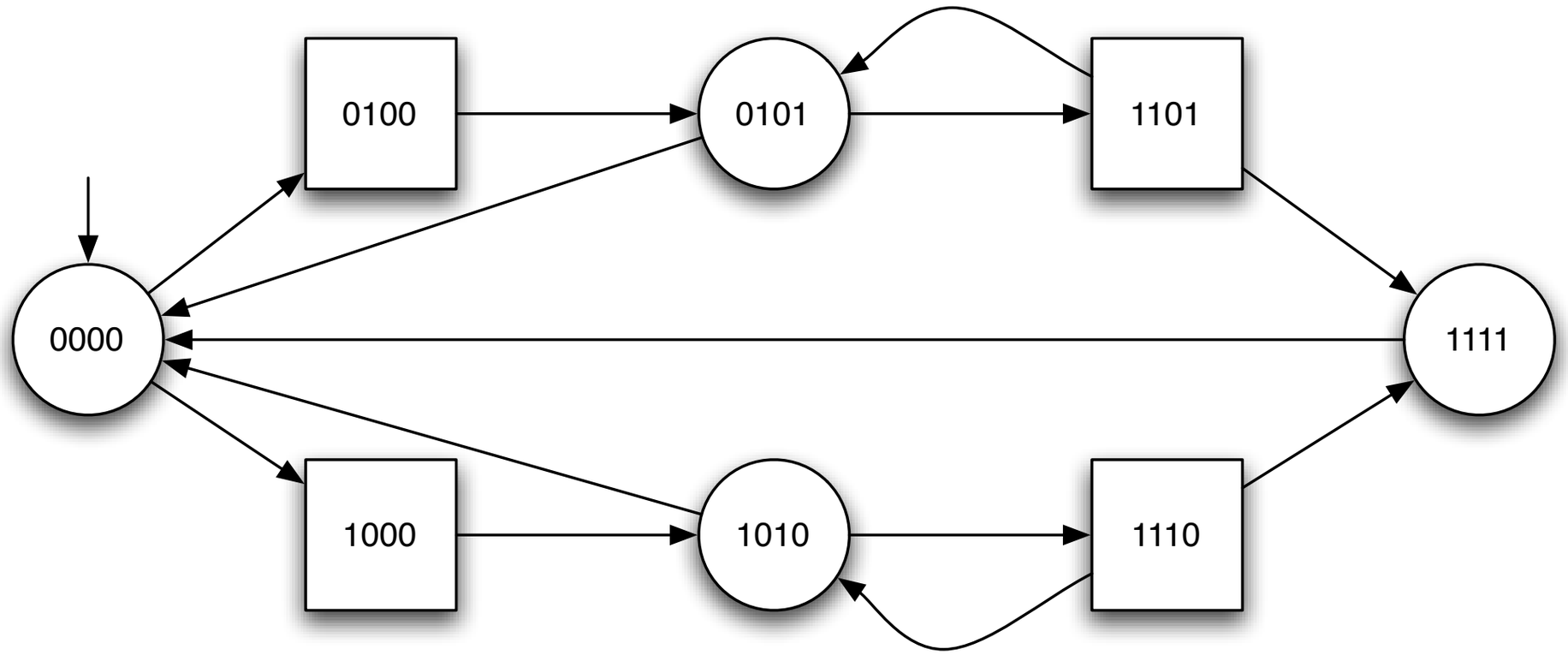
1) $w(0) = \iota$
2) $\forall i \geq 0 : \delta(w(i), w(i+1))$

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

$$\mathsf{PrefPlays}(G) = \{q_0 q_1 \ldots q_n \mid \exists w \in \mathsf{Plays}(G) \wedge \forall 1 \leq i \leq n : w(i) = q_i\}$$

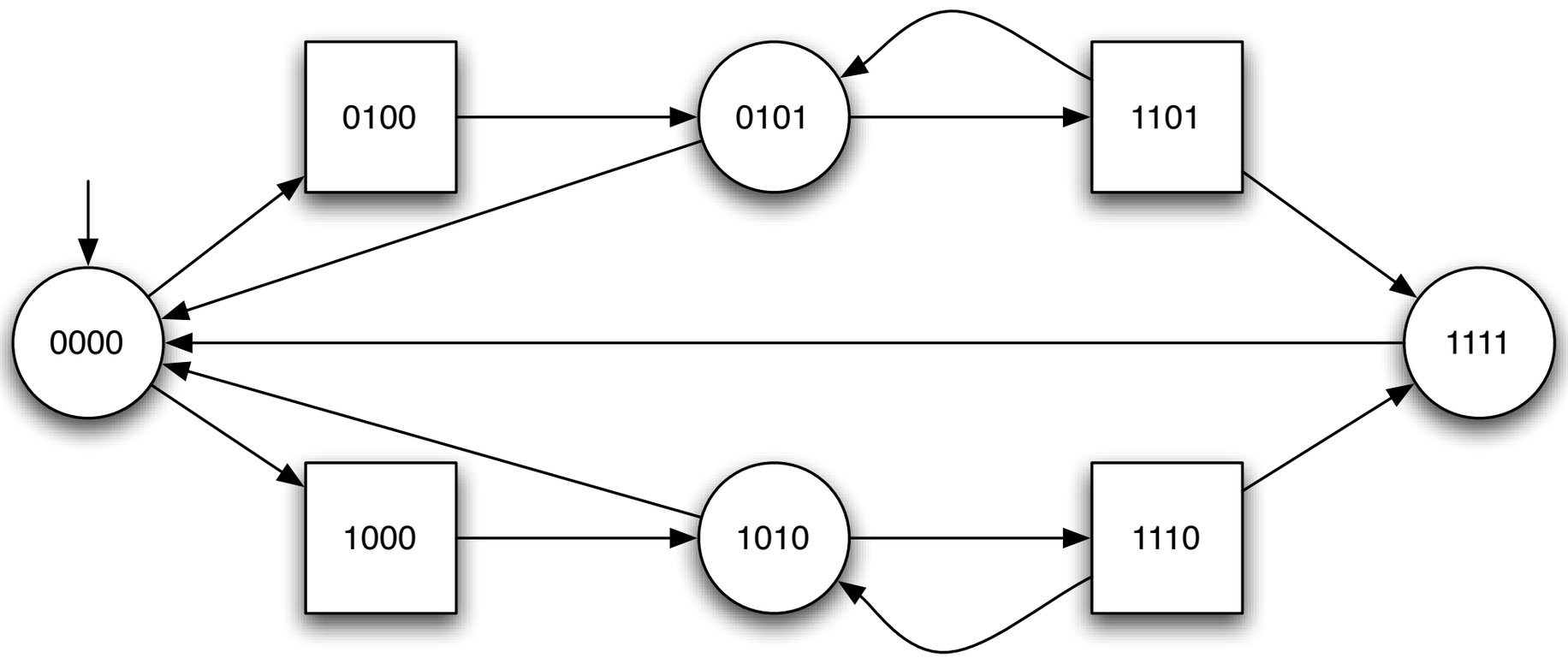
$$\mathsf{PrefPlays}_k(G) = \{w \in \mathsf{PrefPlays}(G) \wedge last(w) \in Q_k\}$$

Who is winning ?



Play : 0000 0100 0101 1101 ...

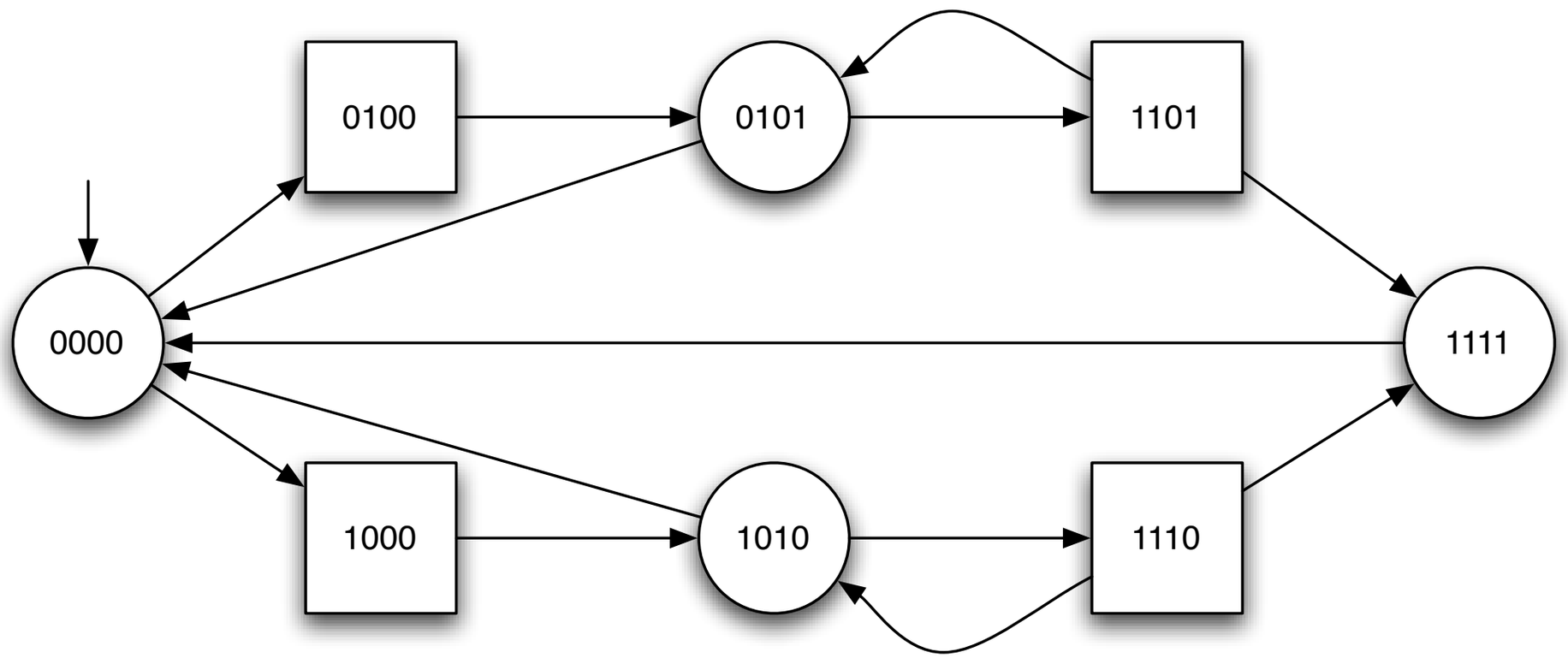
Who is winning ?



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:

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

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

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

Winning strategy

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

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

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

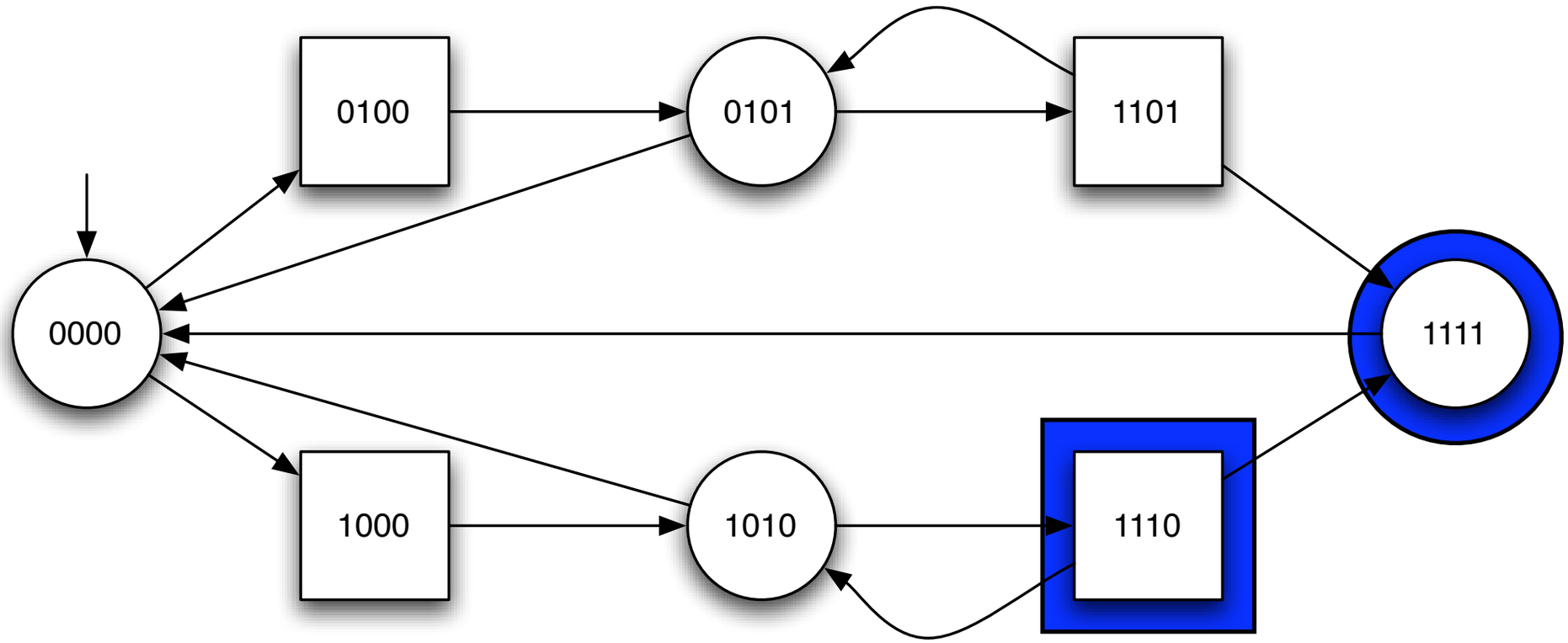
=

Controllers that enforce winning plays

Winning conditions

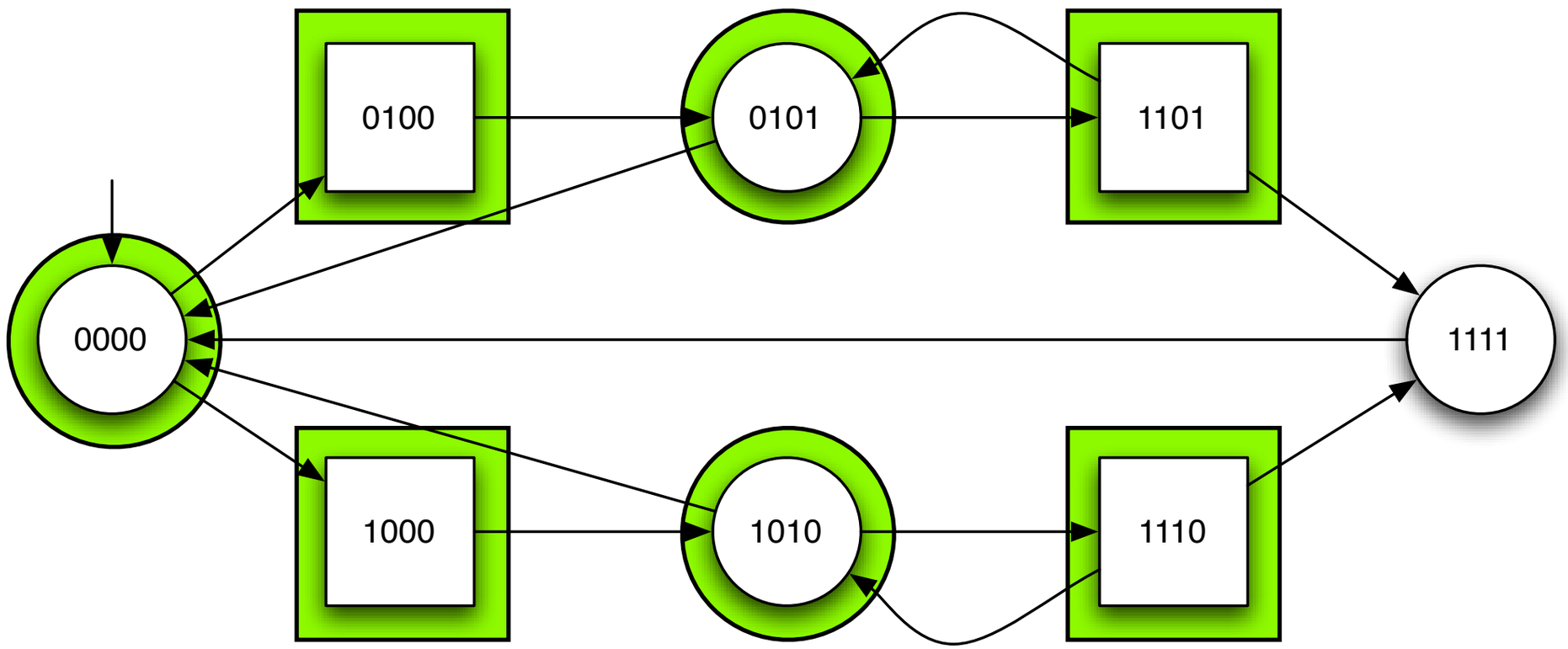
- **Not all** winning conditions are reasonable
- One often assumes that the set of winning plays is a **regular set**
- We show here how to solve **reachability** and **safety** games

A Reachability Game



Does Player I, who owns the rounded positions, have a strategy (against any choices of Player II) to reach the set $\{1101, 1111\}$?

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

Player *k* Controllable Predecessors

X is a set of positions

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

Set of Player I positions where he 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

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

Symmetrically

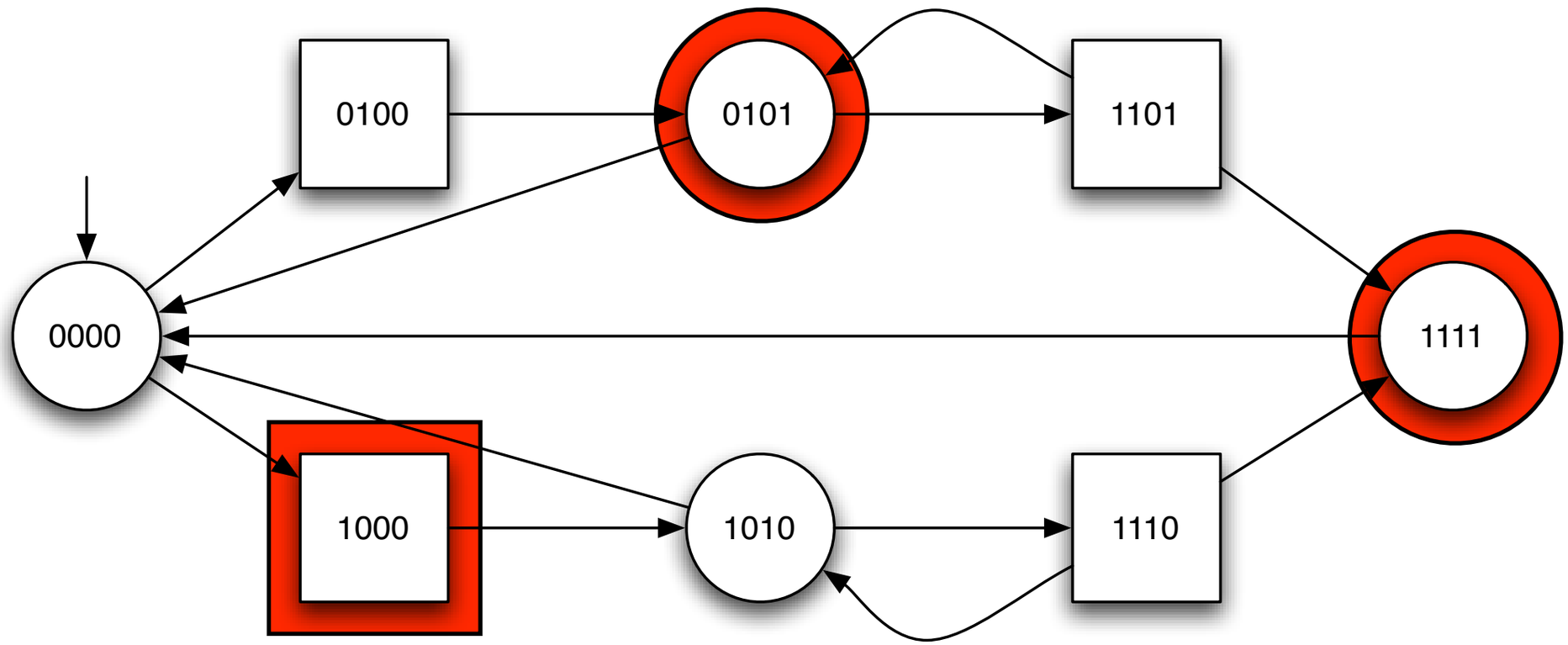
$$\mathsf{2CPre}_G(X) = \{q \in Q_2 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_1 \mid \forall q' : \delta(q, q') : q' \in X\}$$

Player k Controllable Predecessors

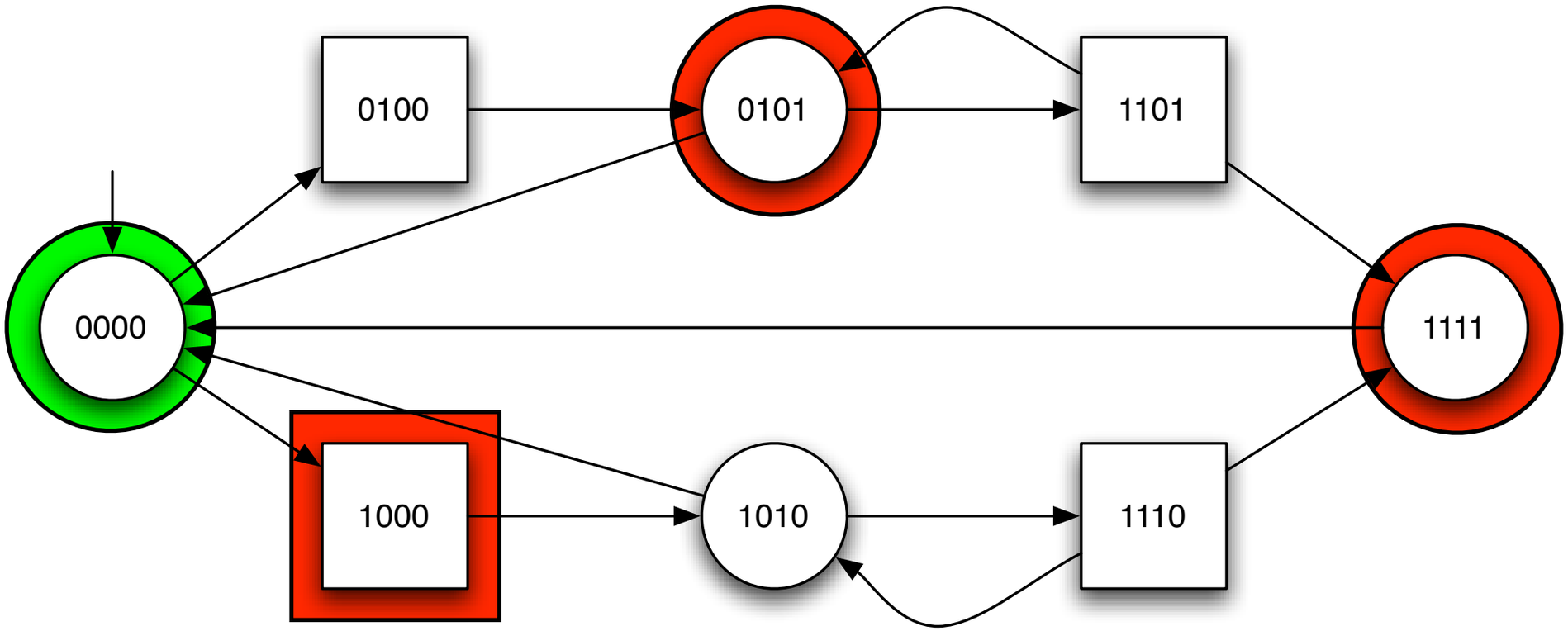
$$\mathsf{1CPre}_G(X) = \{q \in Q_1 \mid \exists q' : \delta(q, q') \wedge q' \in X\} \cup \{q \in Q_2 \mid \forall q' : \delta(q, q') : q' \in X\}$$

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

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



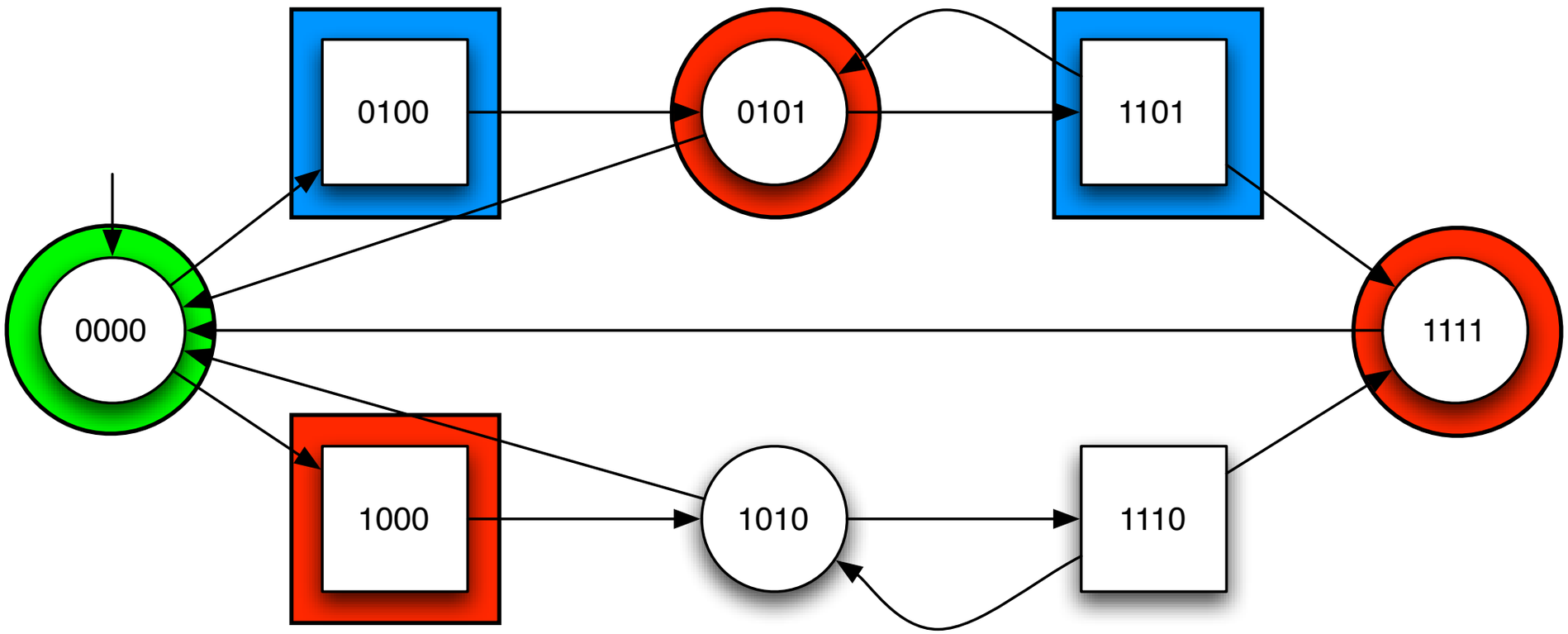
$$X = \{1000, 0101, 1111\}$$



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

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

Rounded positions,
there exists a red successor



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

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

Rounded positions,
there exists a red successor

Squared positions,
all successors are red

Fixpoints to Solve Games

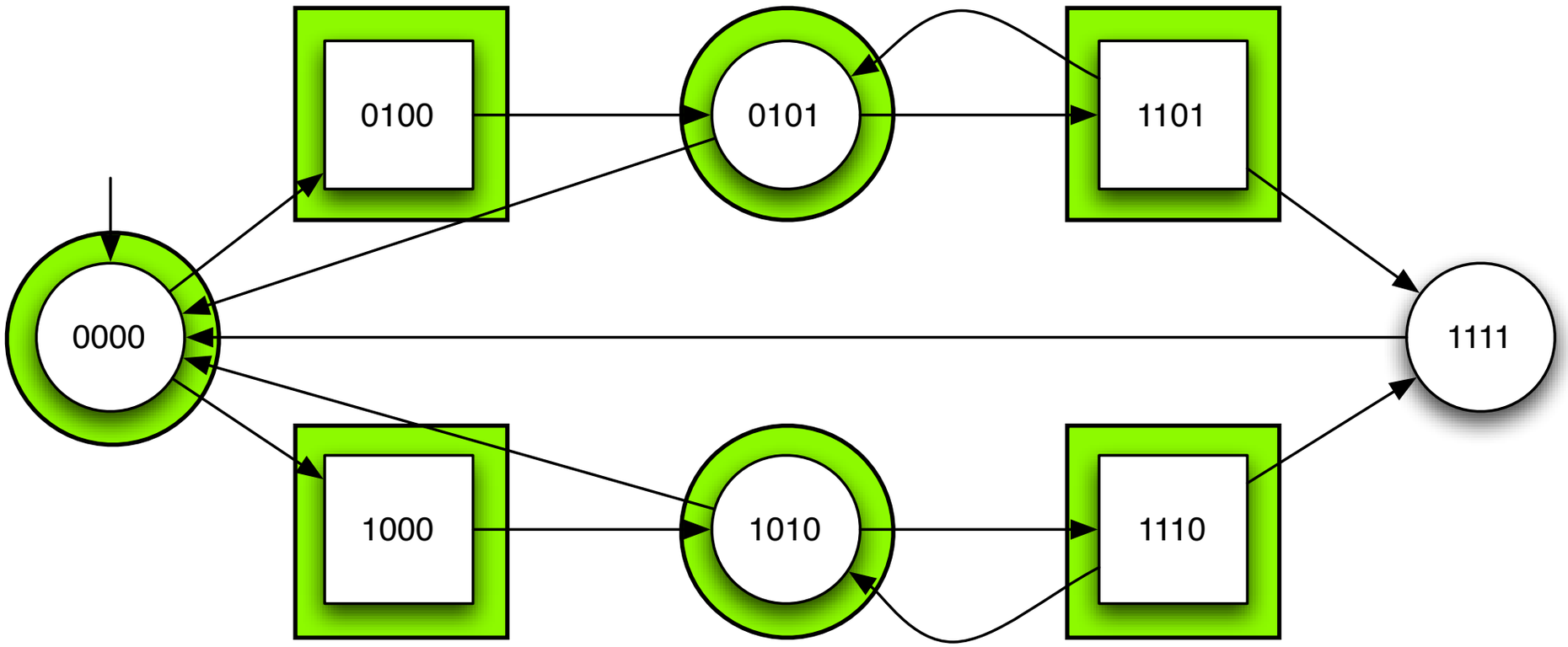
Reachability game for set Q

$$\mu X \cdot Q \cup \mathsf{1CPre}(X)$$

Safety game for set Q

$$\nu X \cdot Q \cap \mathsf{1CPre}(X)$$

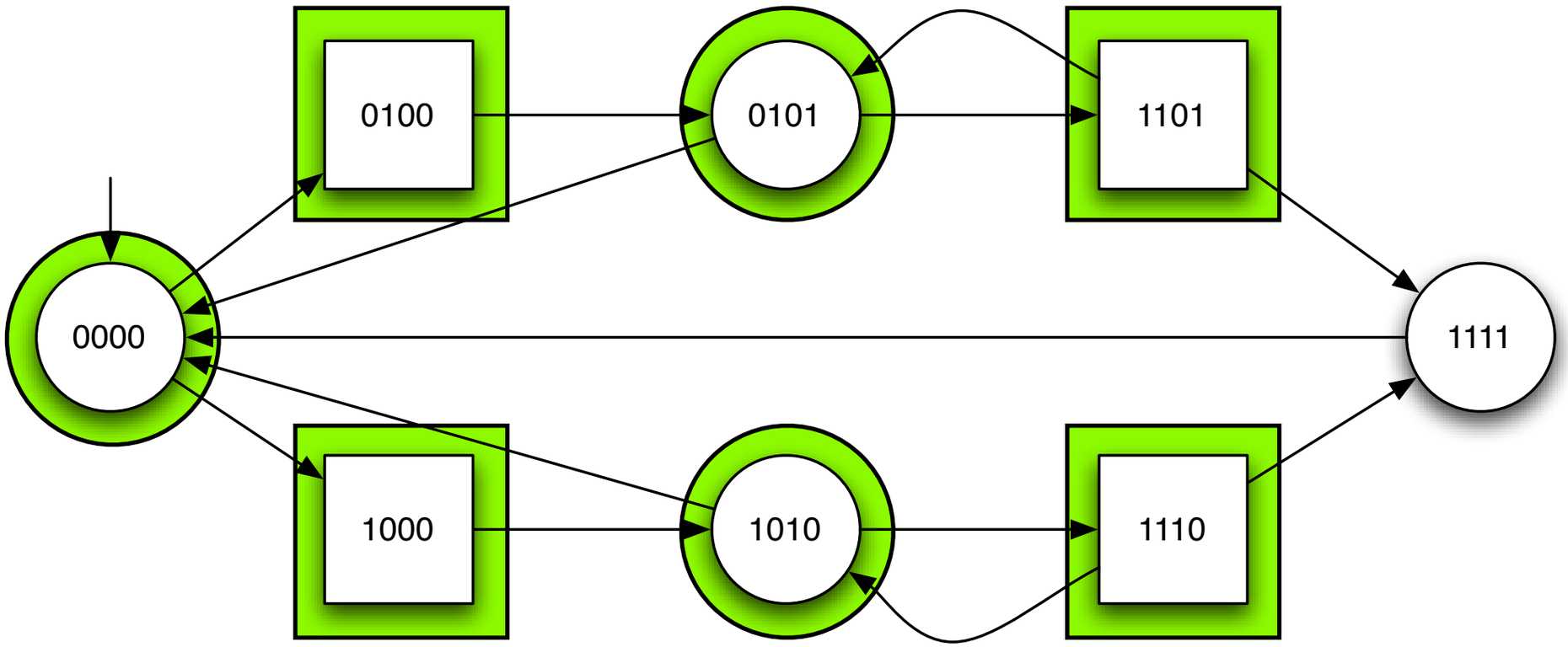
Fixpoint for a safety game



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

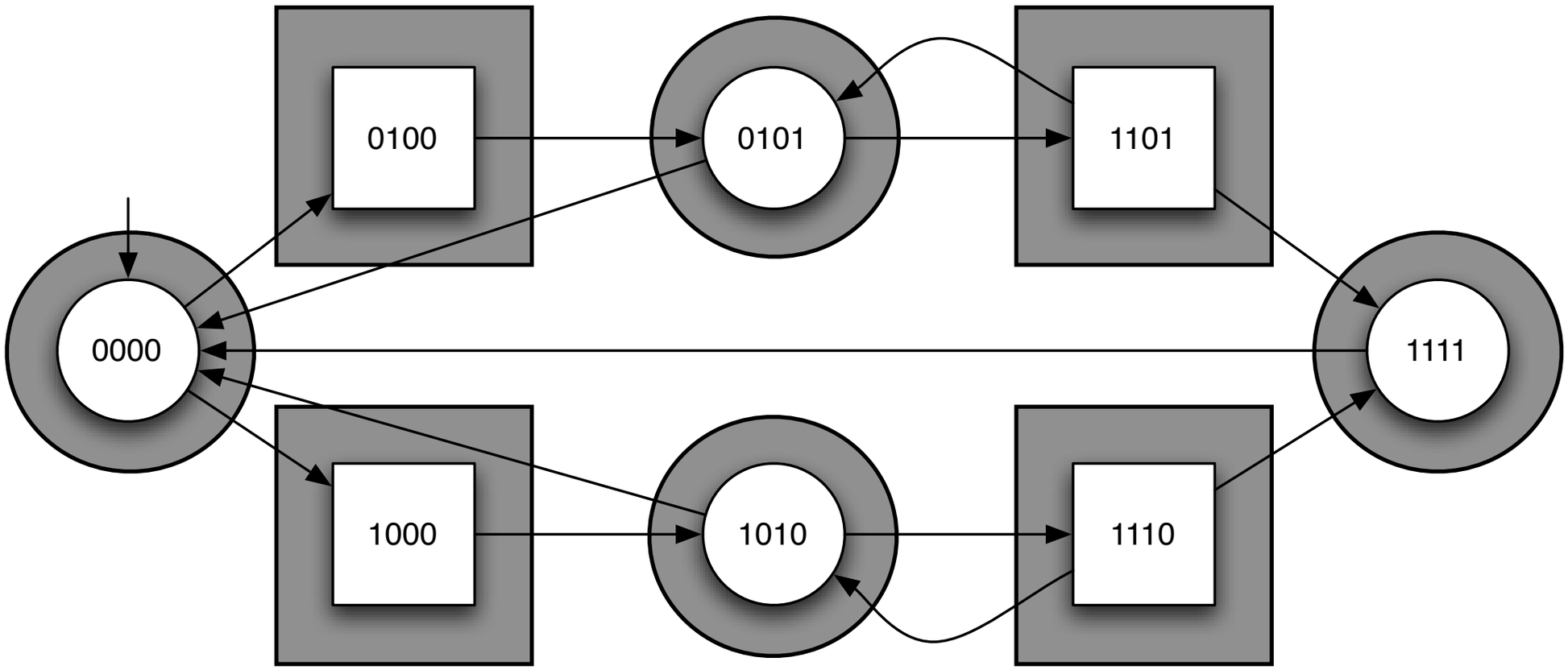


We must compute

$$\nu X \cdot (Q \setminus \{1111\}) \cap \mathsf{1CPre}(X)$$

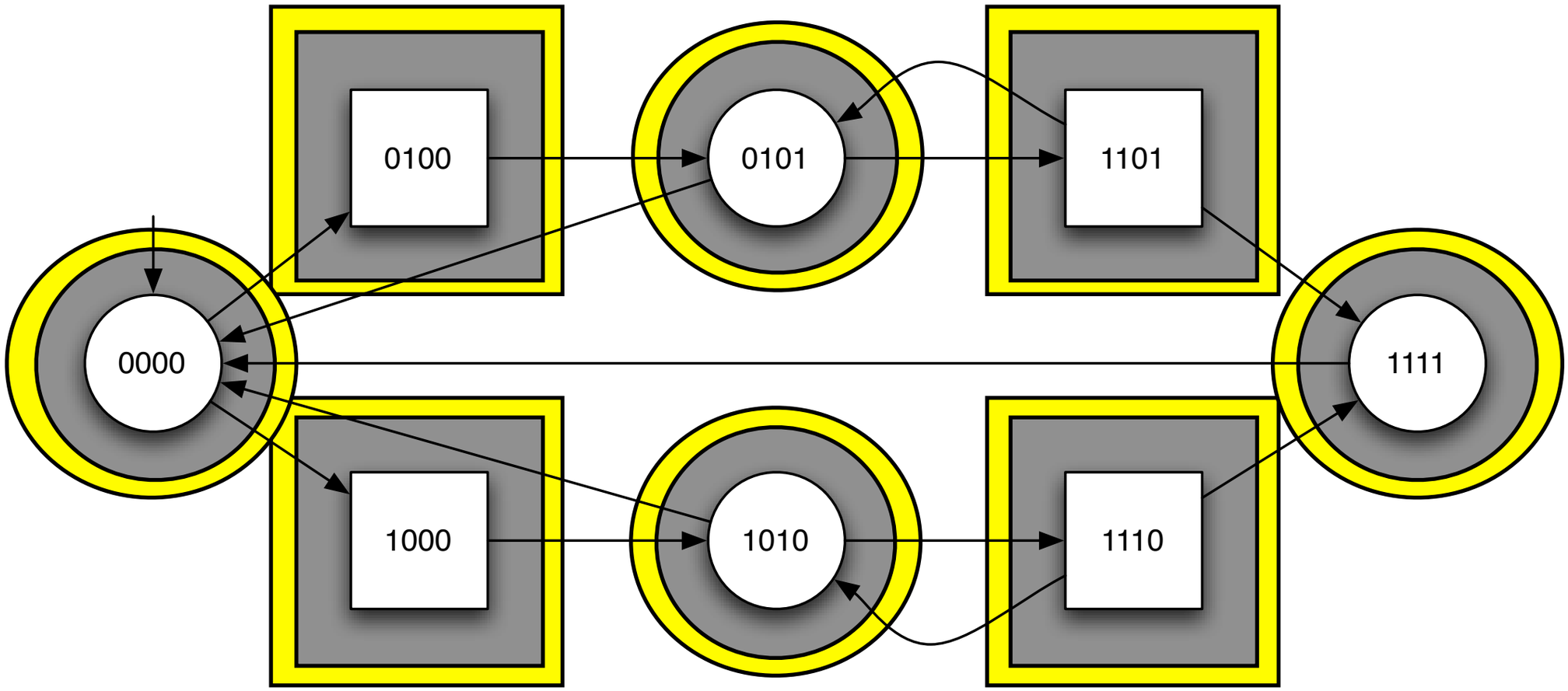
To do that, we use the Tarski fixpoint theorem.

Fixpoint for a safety game



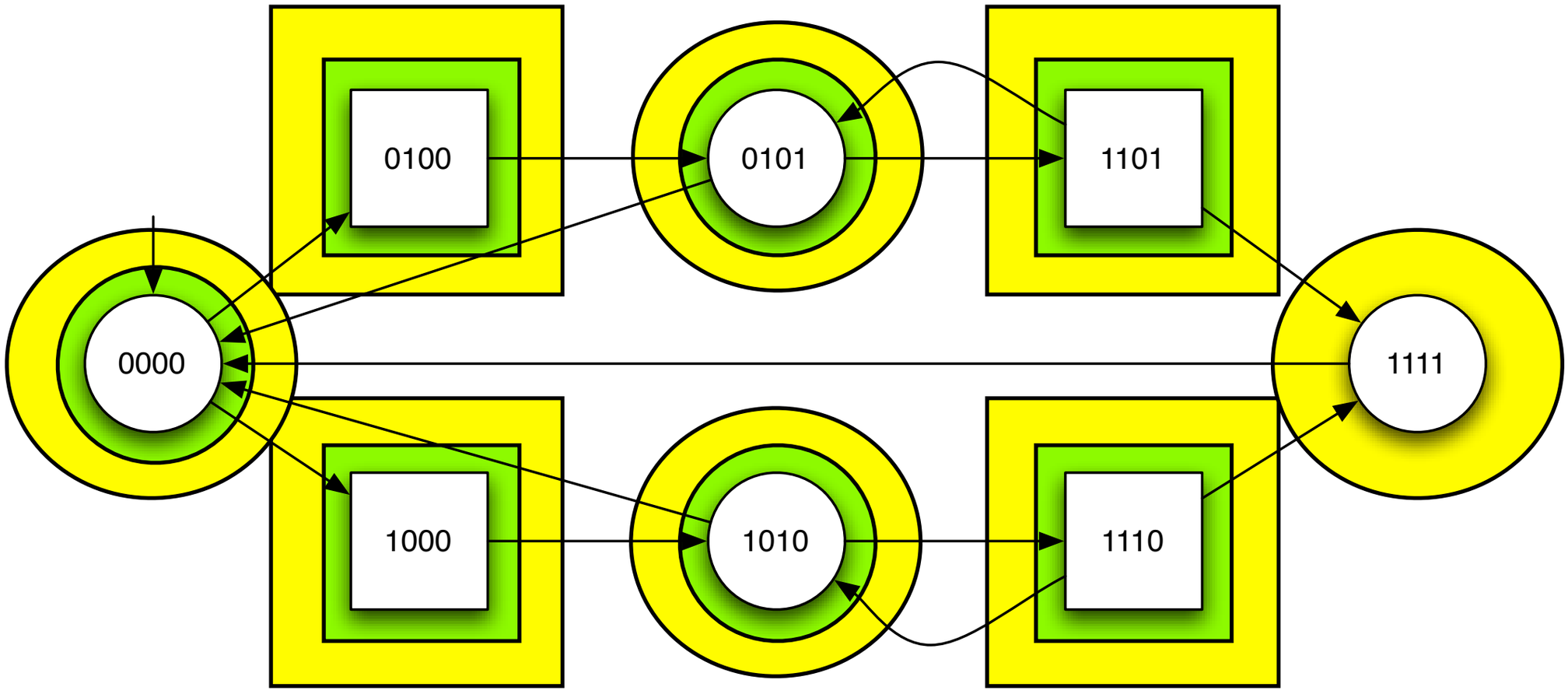
$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

Fixpoint for a safety game



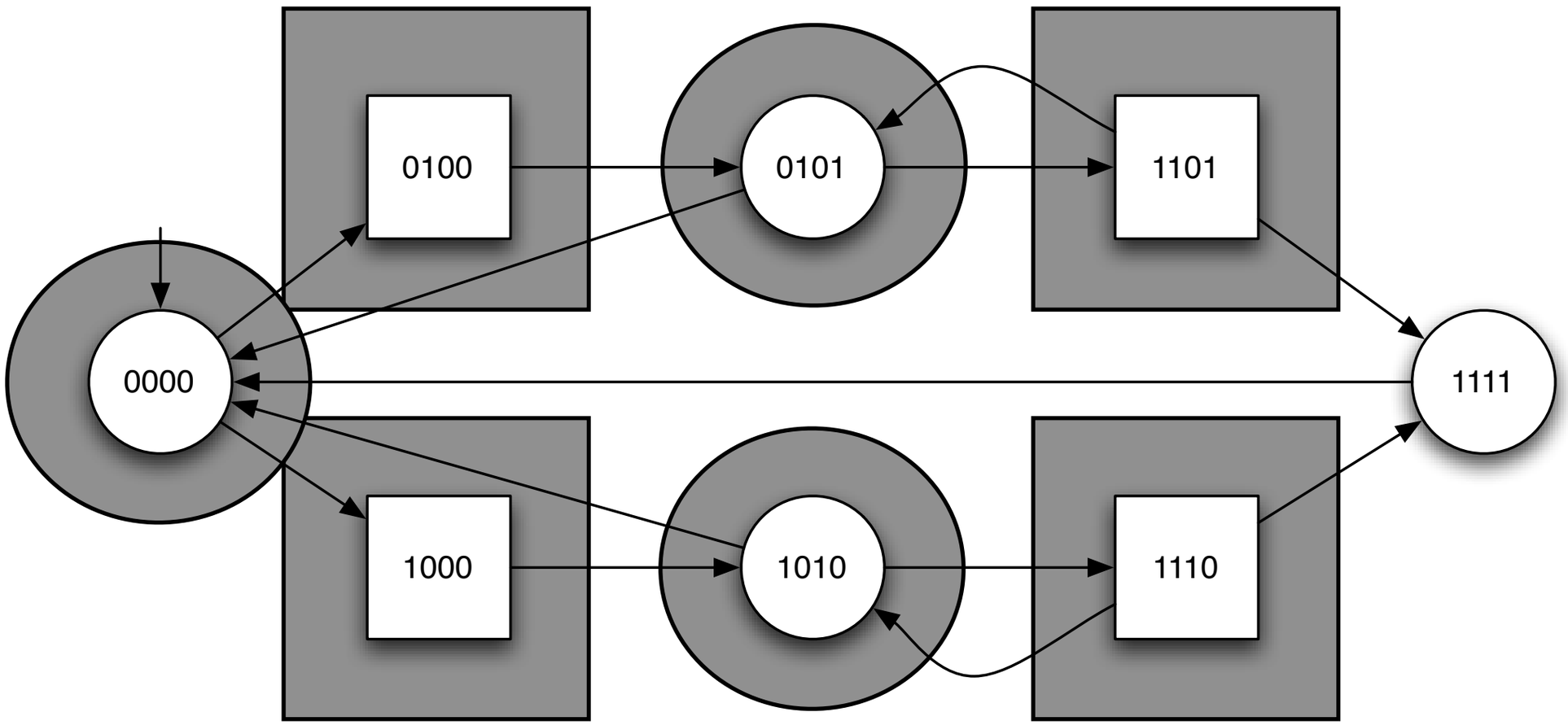
$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

Fixpoint for a safety game



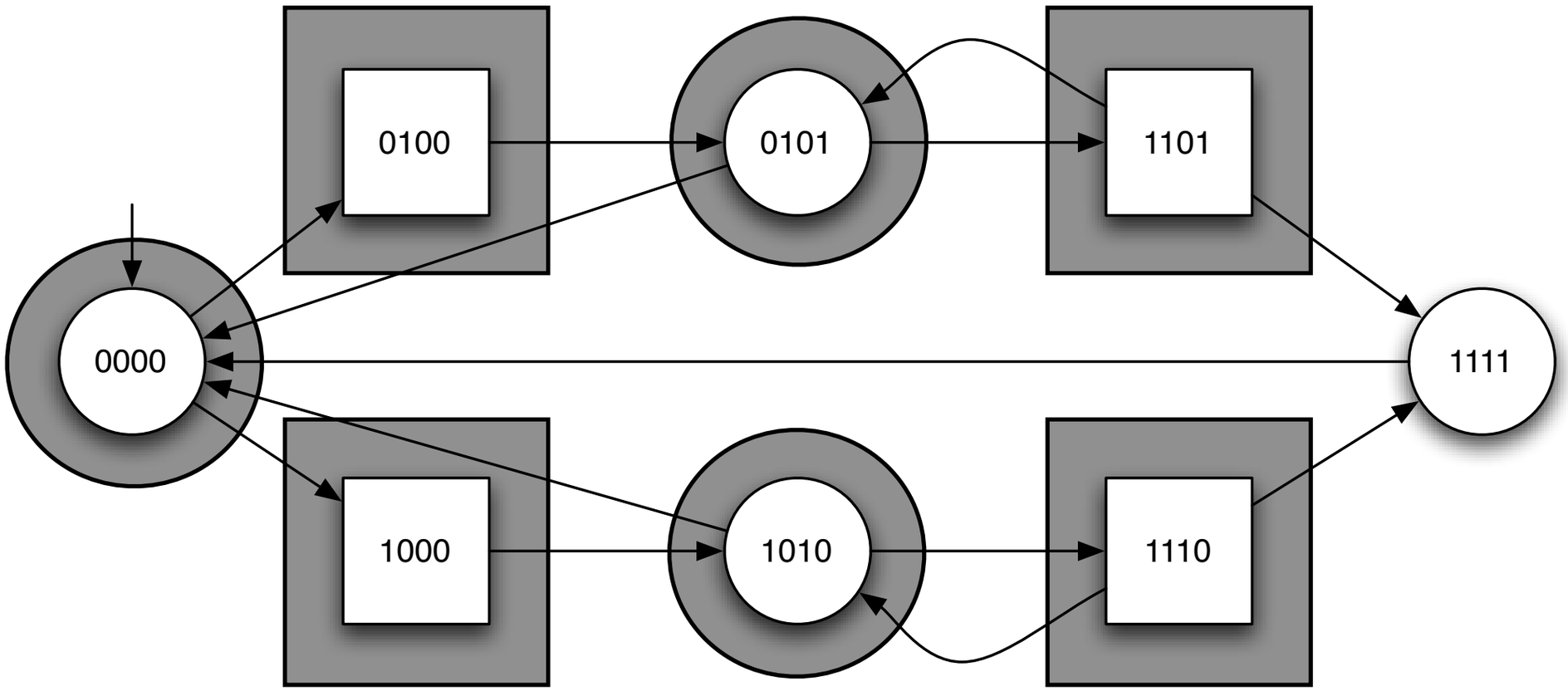
$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

Fixpoint for a safety game



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

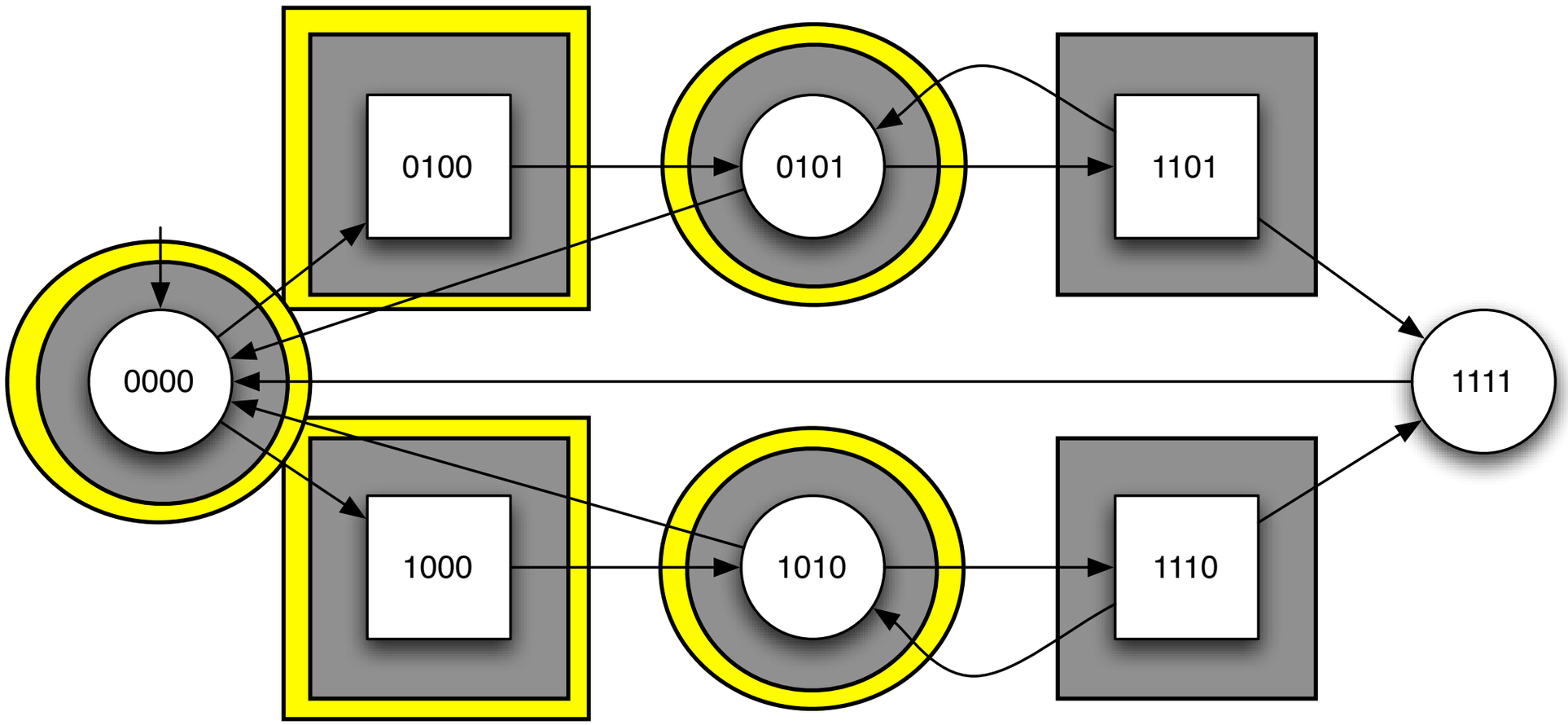
Fixpoint for a safety game



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

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

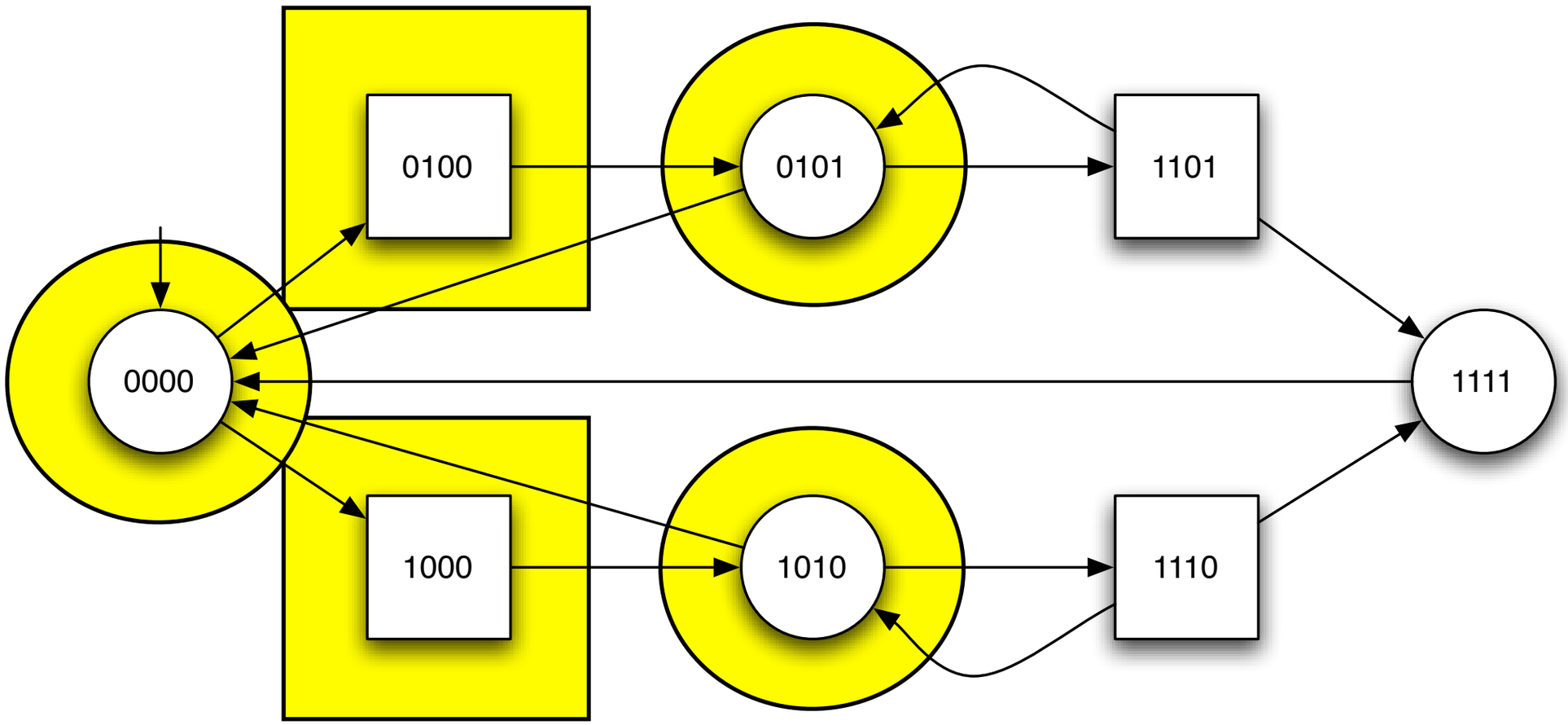
Fixpoint for a safety game



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

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

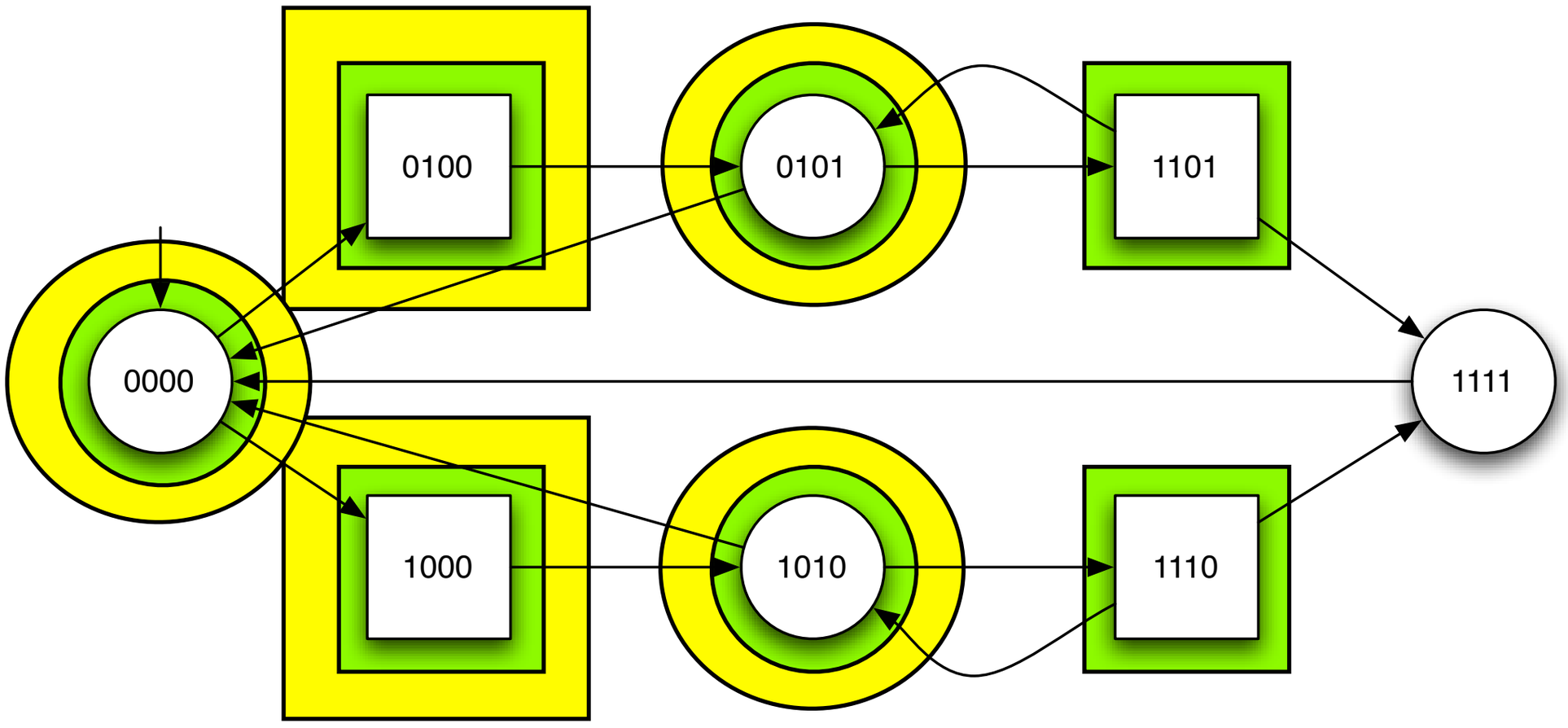
Fixpoint for a safety game



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

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

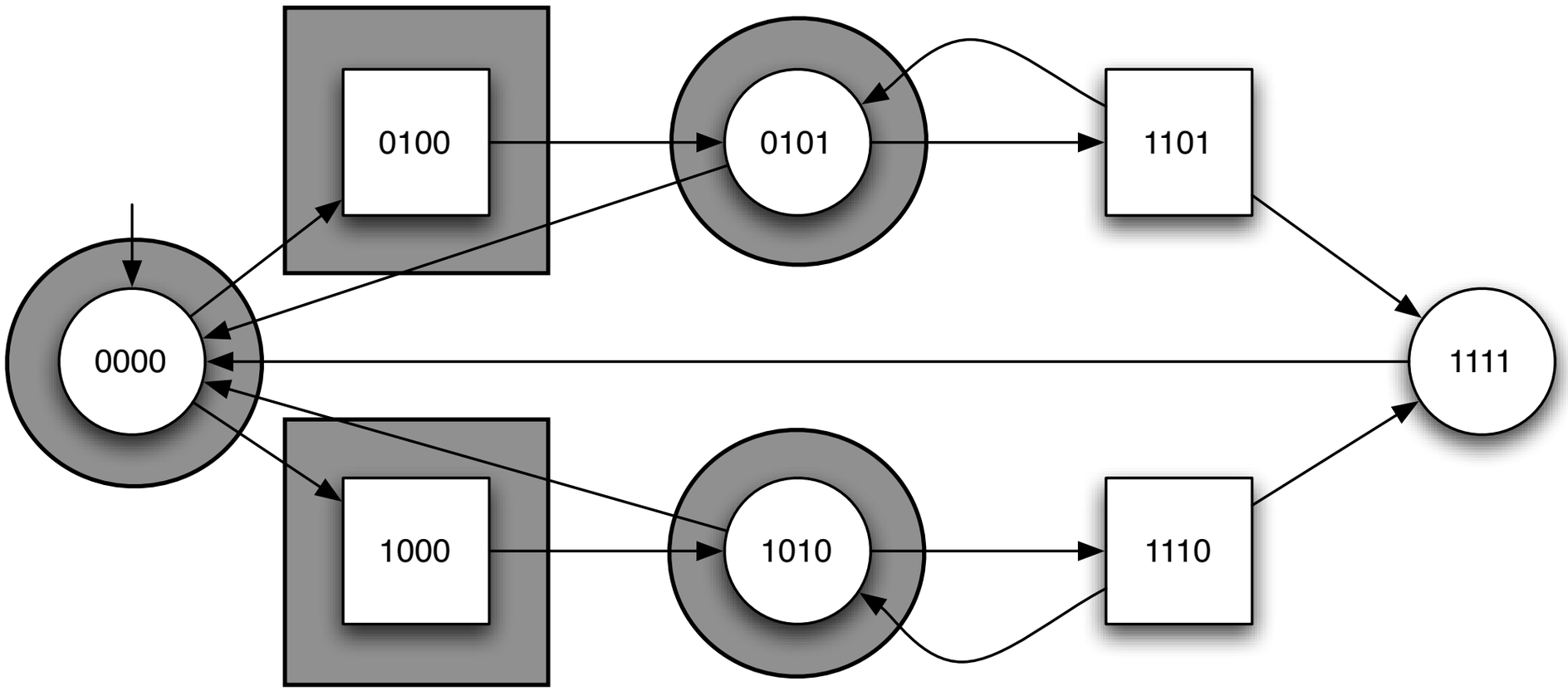
Fixpoint for a safety game



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

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

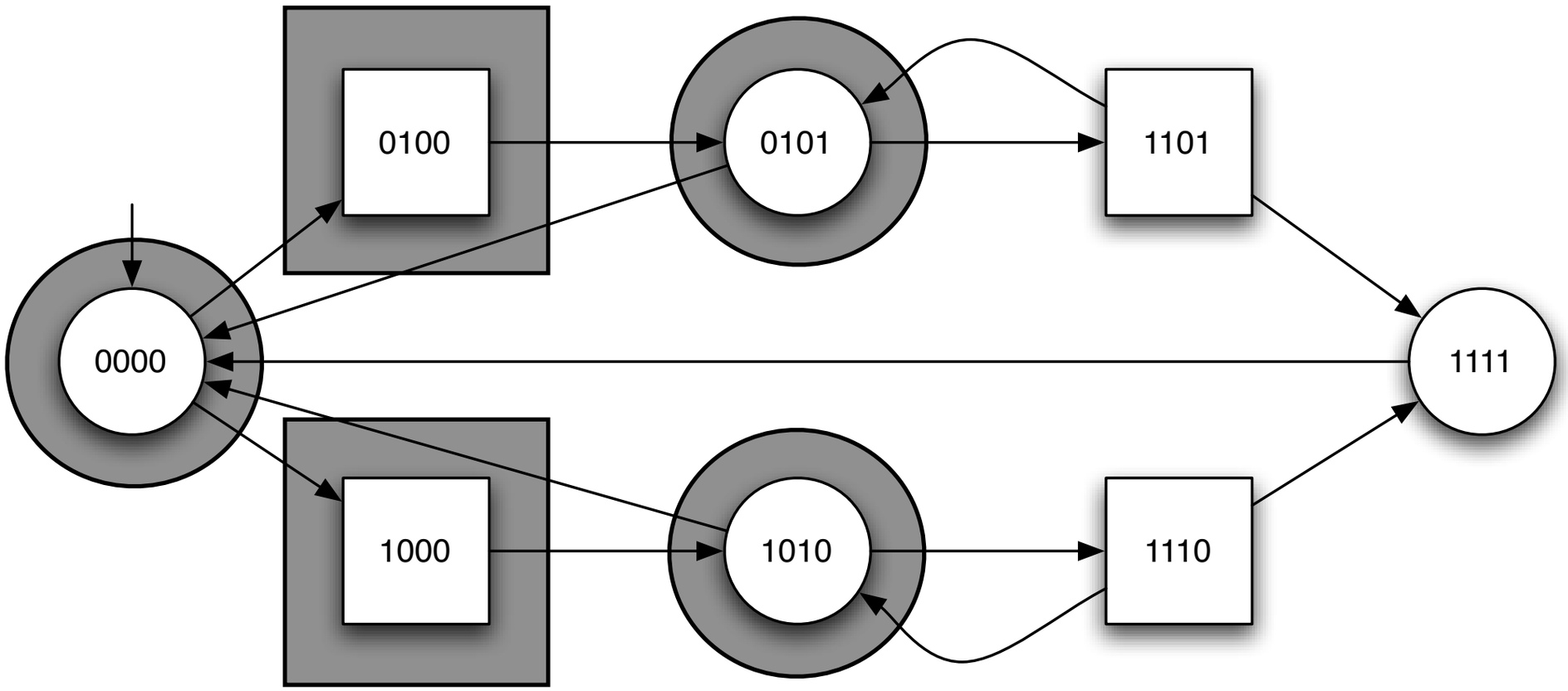
Fixpoint for a safety game



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

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

Fixpoint for a safety game

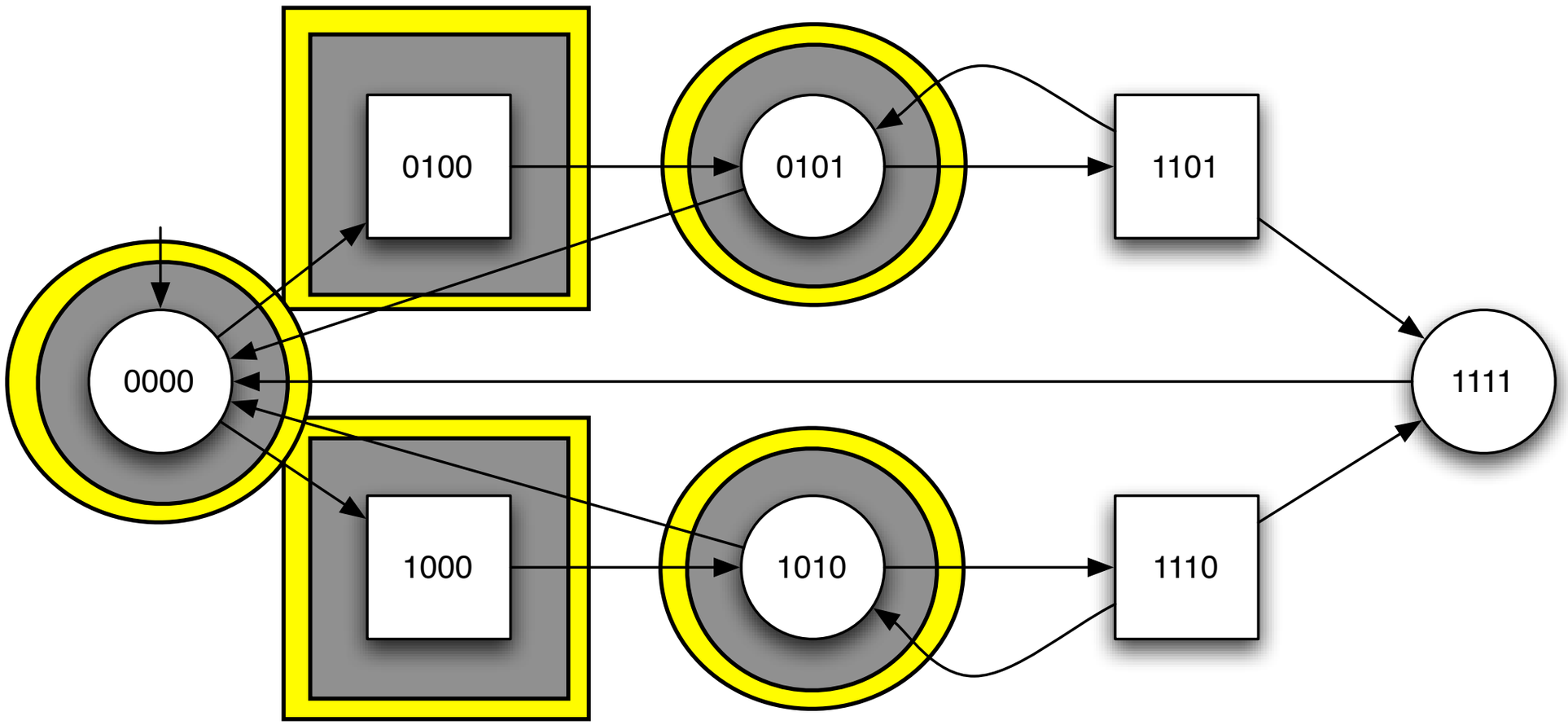


$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

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

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

Fixpoint for a safety game

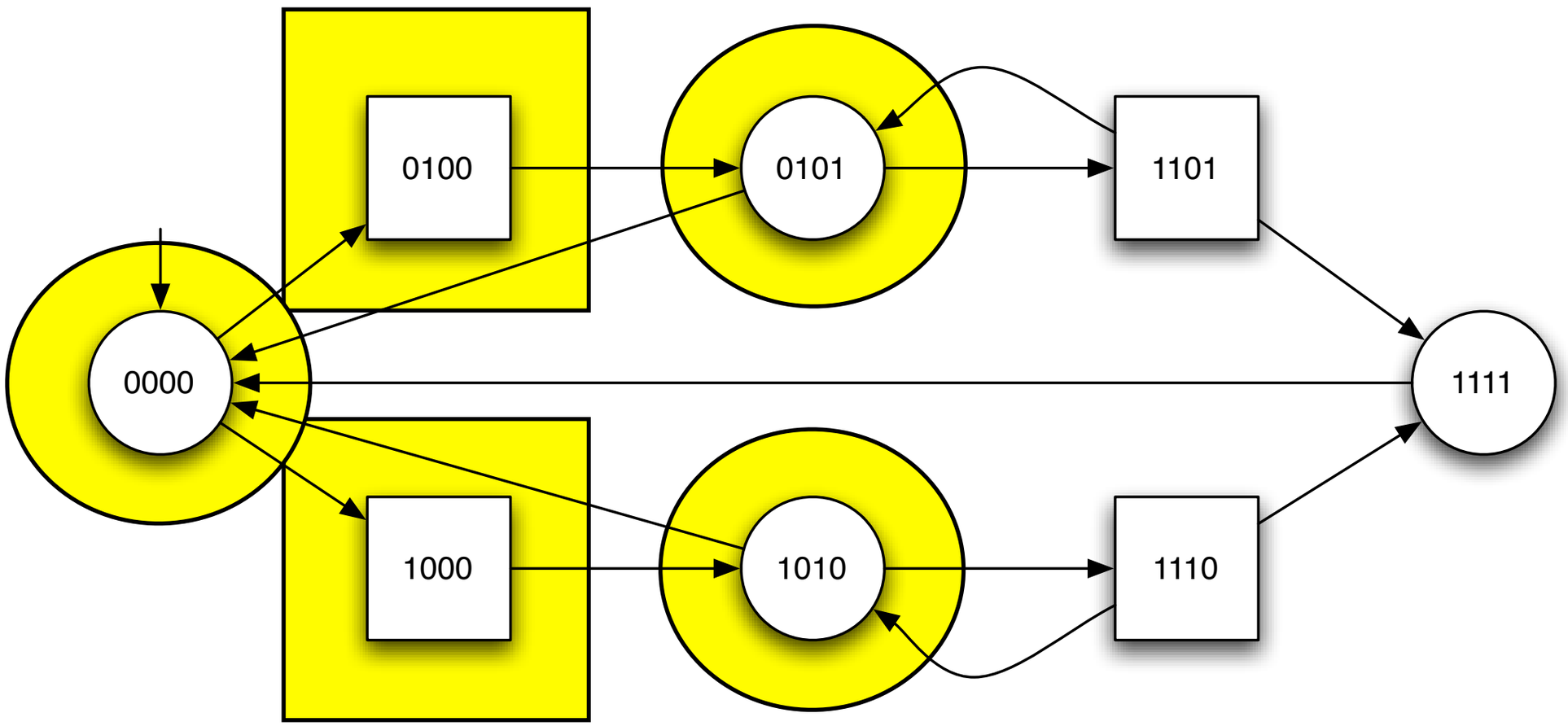


$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

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

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

Fixpoint for a safety game

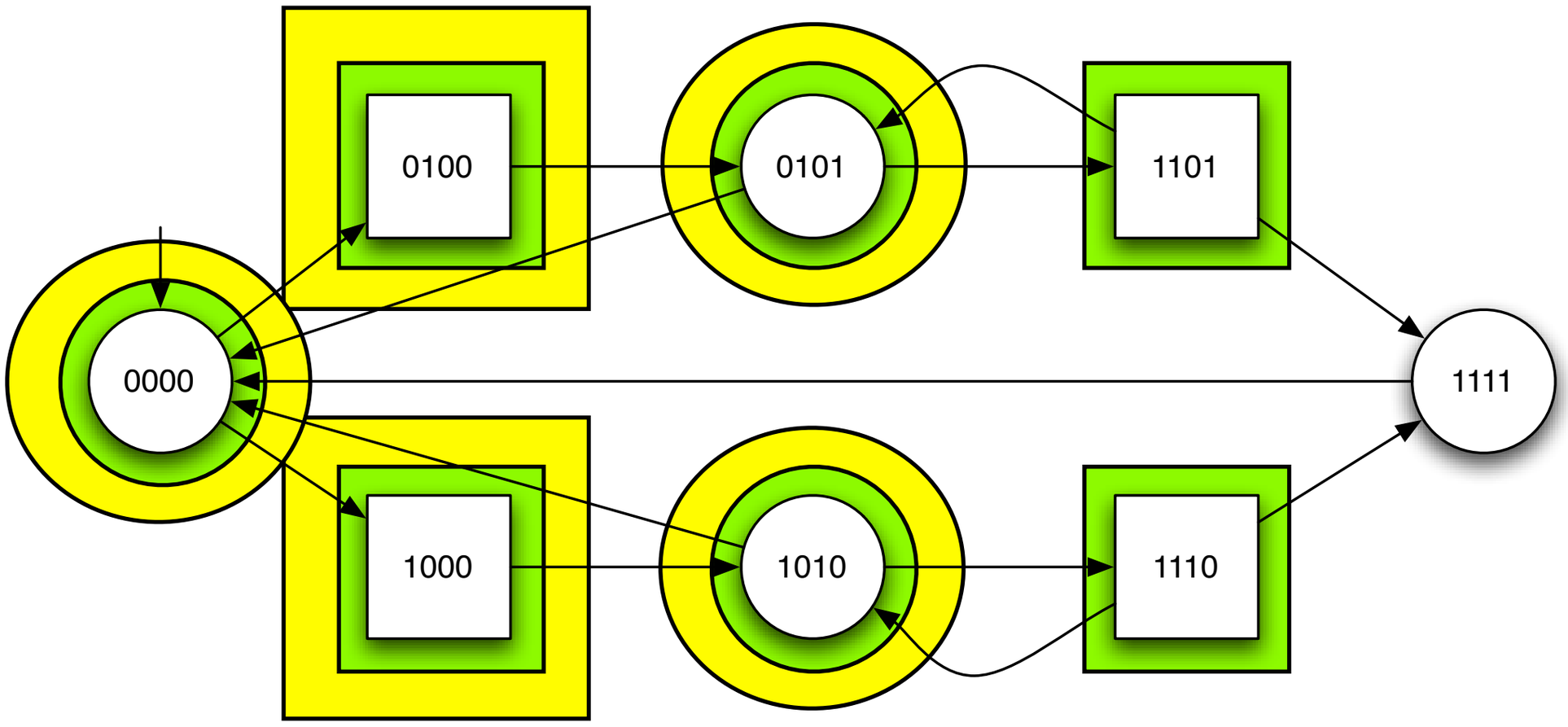


$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

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

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

Fixpoint for a safety game

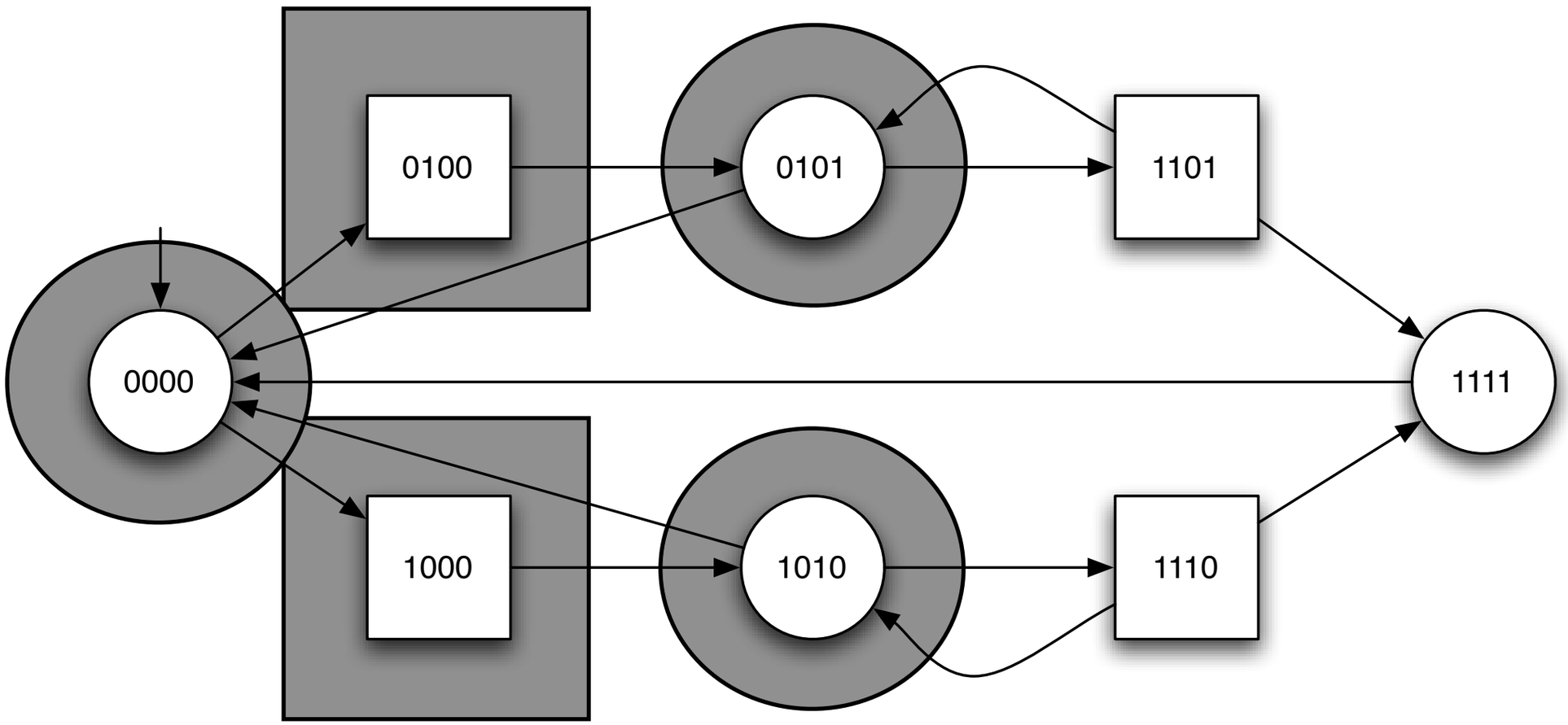


$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

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

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

Fixpoint for a safety game

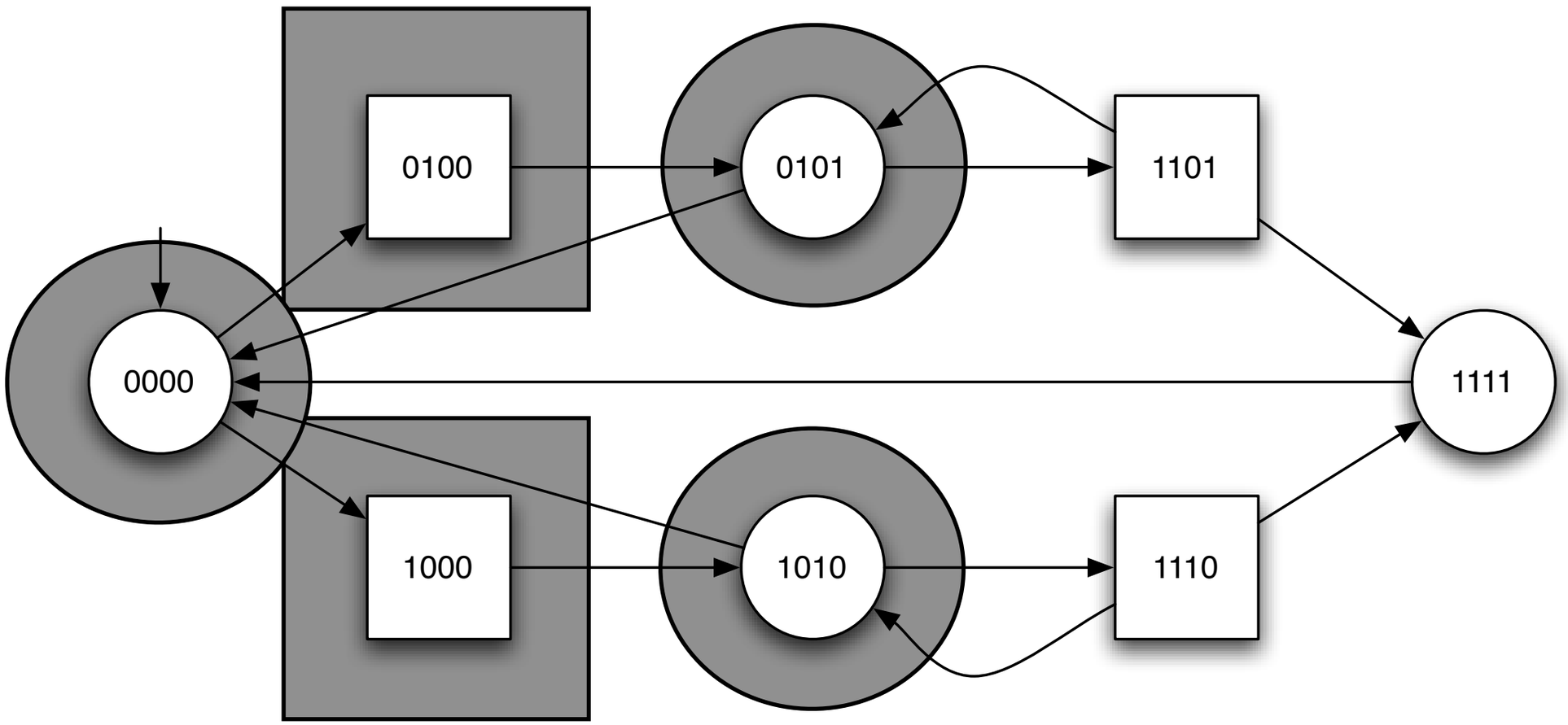


$$X_0 = (Q \setminus \{1111\}) \cap \mathsf{1CPre}(Q)$$

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

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

Fixpoint for a safety game



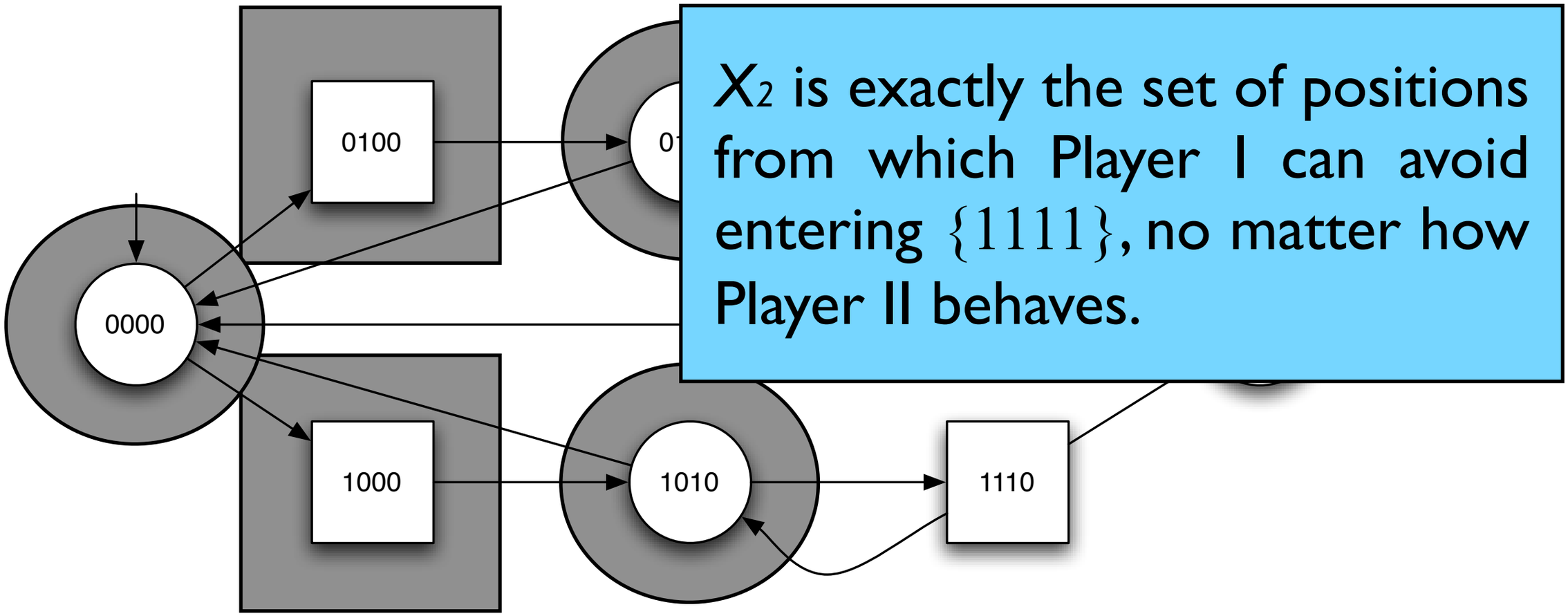
This is the greatest fixpoint

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

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

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

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.

This is the greatest fixpoint

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

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

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

Theorem

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

$$\iota \in \mu X \cdot Q \cup \mathsf{1CPre}(X)$$

Theorem

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

$$\iota \in \nu X \cdot Q \cap \mathsf{1CPre}(X)$$

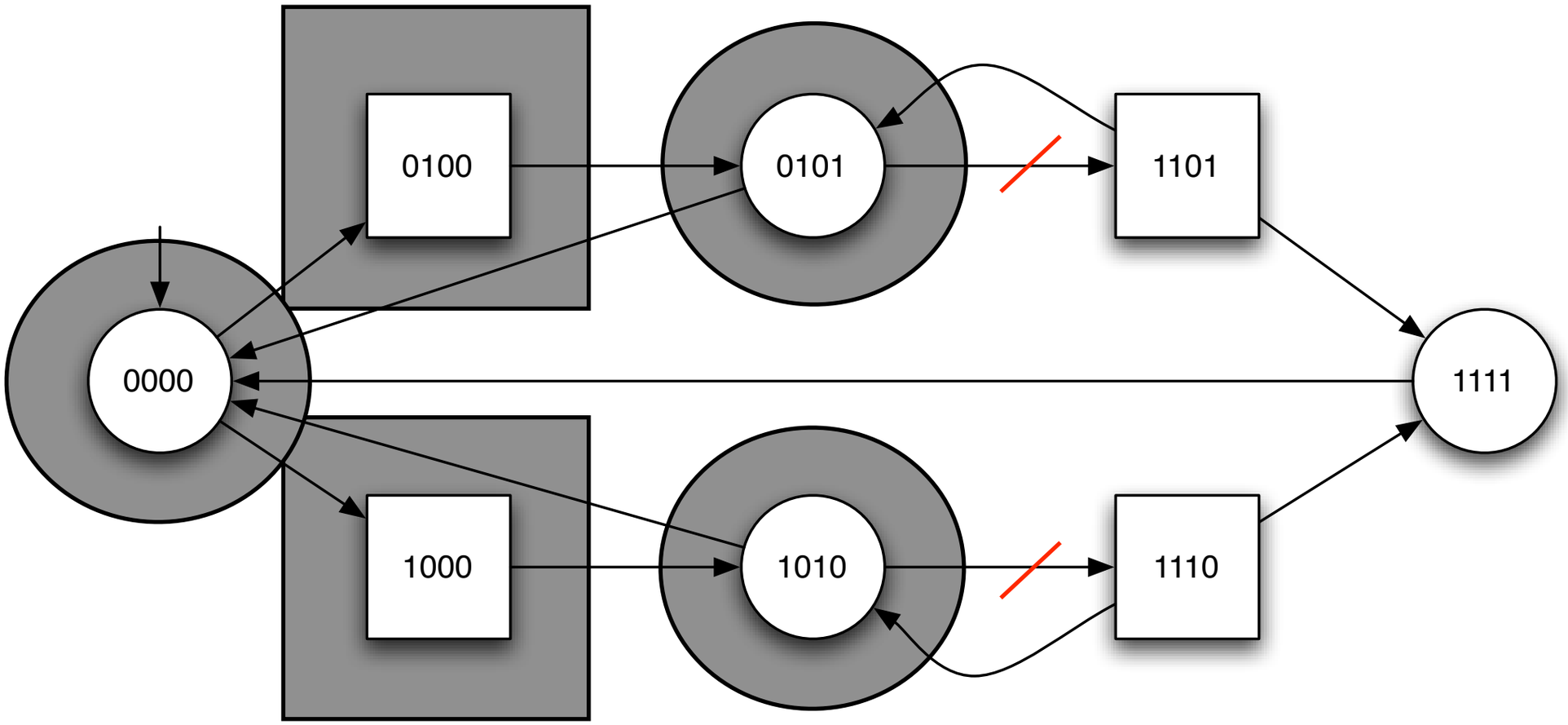
Some more results

Any finite state game with **regular objective** can be solved.

Some more results

Any finite state game with **regular objective** can be solved.

Strategies for safety and reachability games are **positional** (no need for memory).


0100
0101
1101
0000
1111
1000
1010
1110

Some more results

Any finite state game with **regular objective** can be solved.

Strategies for safety and reachability games are **positional** (no need for memory).

For more complicated games, like LTL games, finite **memory** is needed.

Some more results

Any finite state game with **regular objective** can be solved.

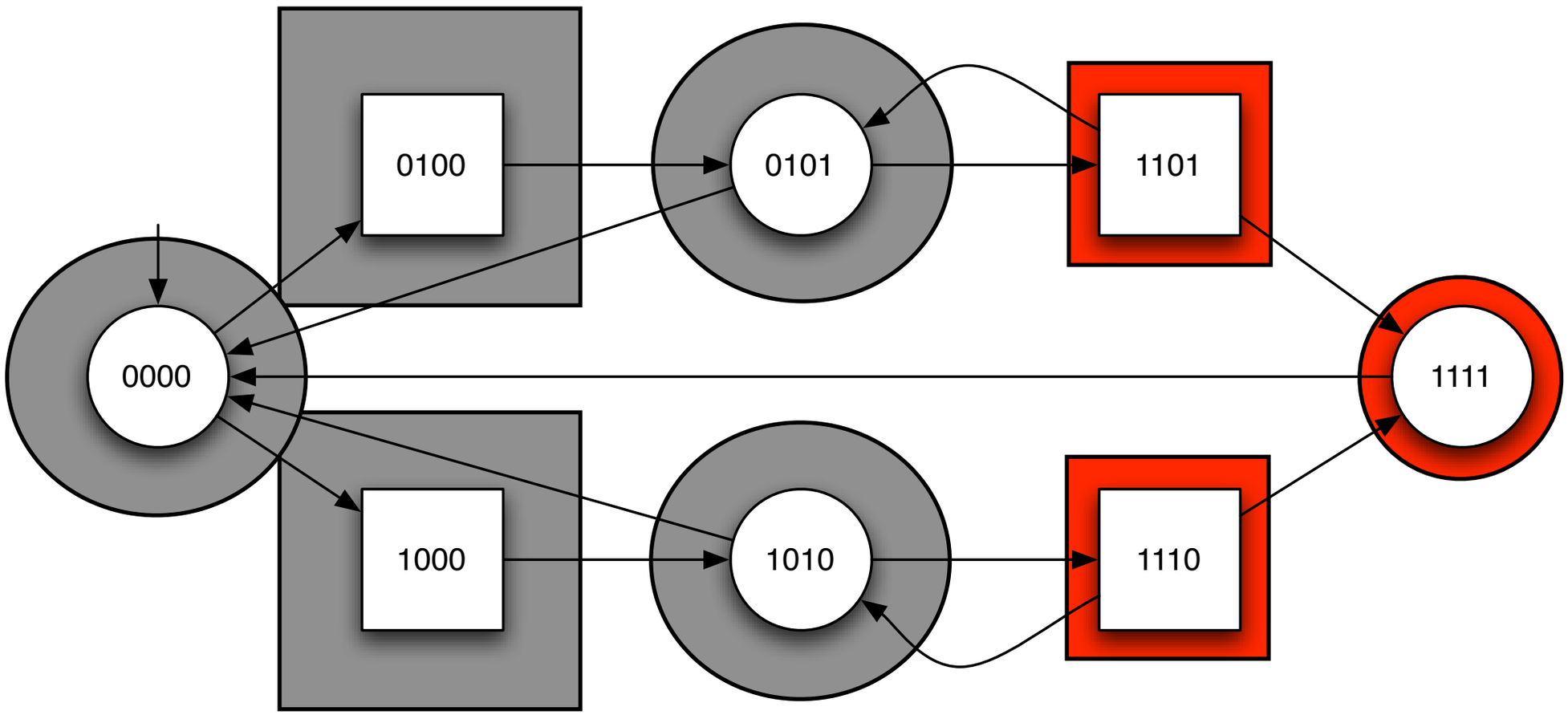
Strategies for safety and reachability games are **positional** (no need for memory).

For more complicated games, like LTL games, finite **memory** is needed.

Determinacy theorem: In positional games (where a position is owned by a player), games are determinate in the following sense :

For any regular set of plays *W*,

Player I has a strategy to win (G, W)
iff
Player II does not have a strategy to win $(G, \overline{W})$



From the red states, and only from those states, Player II has a strategy to reach the state 1111

Conclusion

- Synthesis: games, players, and strategies are powerful metaphors for controller synthesis;
- To go beyond, we need more/better theoretical foundations: games of imperfect information, randomized strategies, optimal strategies, infinite state games, etc.

General references on games and synthesis

[AHK02] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49:672–713, 2002.

[GH82] Y. Gurevich and L. Harrington. Trees, automata, and games. STOC 1982: In *Proceedings of the 14th International Symposium on Theory of Computing*, pages 60–65, ACM Press, 1984.

[PR90] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *Proceedings of the 31st International Symposium on Foundations of Computer Science*, pages 746–757. IEEE Computer Society Press, 1990.

[Rei84] J.H. Reif. The complexity of two-player games of incomplete information. *Journal on Computer and System Sciences*, 29:274–301, 1984.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In *Proceedings of the 12th International Symposium on Theoretical Aspects of Computer Science*, volume 900 of Lecture Notes in Computer Science, pages 1–13. Springer-Verlag, 1995.

[RW89] P.J.G. Ramadge and W.M. Wonham. The control of discrete-event systems. *IEEE Transactions on Control Theory*, 77:81–98, 1989.

References on timed and hybrid games

[AMPS98] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symp. System Structure and Control*, pages 469–474. Elsevier, 1998.

[BDMP02] P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. Research Report LSV-02-5, LSV, ENS de Cachan, France, 2002.

[CHR02] F. Cassez, T.A. Henzinger, and J.-F. Raskin. A comparison of control problems for timed and hybrid systems. In *Proc. 5th Int. Works. Hybrid Systems: Computation and Control (HSCC'02)*, volume 2289 of *LNCS*, pages 134–148. Springer, 2002.

[HHM99] T.A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In *Concurrency Theory*, Lect. Notes in Comp. Sci. 1664, pages 320–335. Springer, 1999.

[HK99] T.A. Henzinger and P.W. Kopke. Discrete-time control for rectangular hybrid automata. *Theor. Comp. Sci.*, 221:369–392, 1999.

References on implementability issues and robustness

[DDR04] M. De Wulf, L. Doyen, and J.-F. Raskin. Almost ASAP semantics: From timed models to timed implementations. In *HSCC 04: Hybrid Systems—Computation and Control*, Lecture Notes in Computer Science 2993, pages 296–310. Springer-Verlag, 2004.

Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robustness and Implementability of Timed Automata. In FORMATS'04, Lecture Notes in Computer Science, 3253, pp. 118-133, Springer Verlag, 2004.

Martin De Wulf, Laurent Doyen, Jean-François Raskin. Systematic Implementations of Timed Models. In Formal Methods Europe'05, LNCS 3582, pp. 139-156, Springer Verlag, 2005.

K. Altisen and S. Tripakis. Implementation of timed automata: an issue of semantics or modeling?. In FORMATS'05 (to appear). A previous version of this paper is available as VERIMAG Technical Report TR-2005-12.

[Pur98] Anuj Puri. Dynamical properties of timed automata. In *Proceedings of Formal Techniques in Real-Time and Fault-Tolerant Systems, 5th International Symposium, FTRTFT'98, Lyngby, Denmark, September 14-18, 1998*, volume 1486 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 1998.

[GHJ97] V. Gupta, T.A. Henzinger, and R. Jagadeesan. Robust timed automata, *HART 97: Hybrid and Real-time Systems*. LNCS 1201, Springer-Verlag, 331–345, 1997.