Games with Imperfect Information: Theory and Algorithms

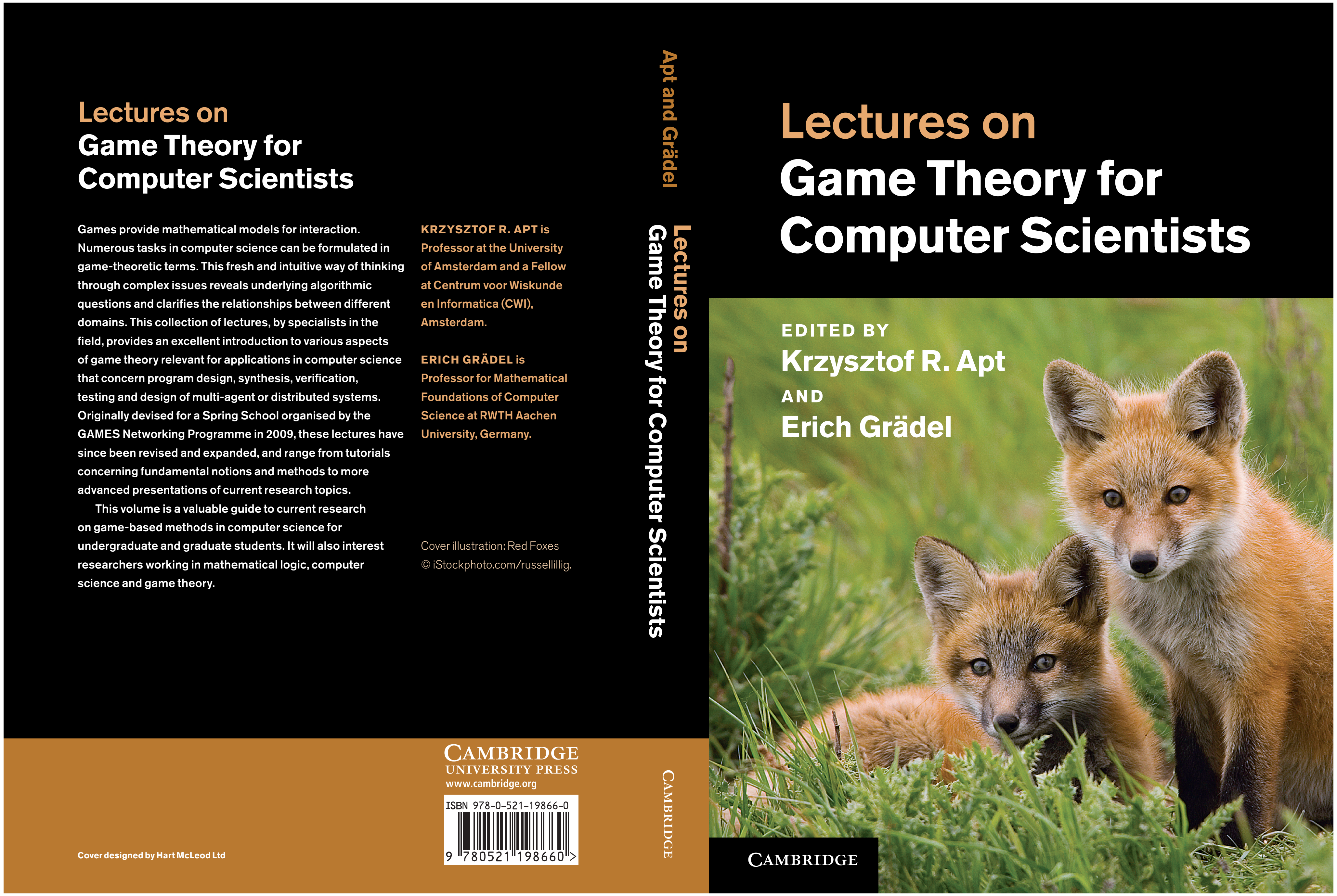
Jean-François Raskin
Université Libre de Bruxelles

(based on joint works with Krishnendu Charterjee, Martin De Wulf, **Laurent Doyen**, Emmanuel Filiot, Thomas Henzinger, Nicolas Maquet, Jin Nayong)

GYPSy
Rennes, November 15, 2010

Available in early 2011

Lecture notes



Lectures on Game Theory for Computer Scientists

Games provide mathematical models for interaction. Numerous tasks in computer science can be formulated in game-theoretic terms. This fresh and intuitive way of thinking through complex issues reveals underlying algorithmic questions and clarifies the relationships between different domains. This collection of lectures, by specialists in the field, provides an excellent introduction to various aspects of game theory relevant for applications in computer science that concern program design, synthesis, verification, testing and design of multi-agent or distributed systems. Originally devised for a Spring School organised by the GAMES Networking Programme in 2009, these lectures have since been revised and expanded, and range from tutorials concerning fundamental notions and methods to more advanced presentations of current research topics.

This volume is a valuable guide to current research on game-based methods in computer science for undergraduate and graduate students. It will also interest researchers working in mathematical logic, computer science and game theory.

KRZYSZTOF R. APT is Professor at the University of Amsterdam and a Fellow at Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

ERICH GRÄDEL is Professor for Mathematical Foundations of Computer Science at RWTH Aachen University, Germany.

Cover illustration: Red Foxes
© iStockphoto.com/russellillig.

Cover designed by Hart McLeod Ltd

CAMBRIDGE
UNIVERSITY PRESS
www.cambridge.org

ISBN 978-0-521-19866-0
9 780521 198660

Apt and Grädel

Lectures on Game Theory for Computer Scientists

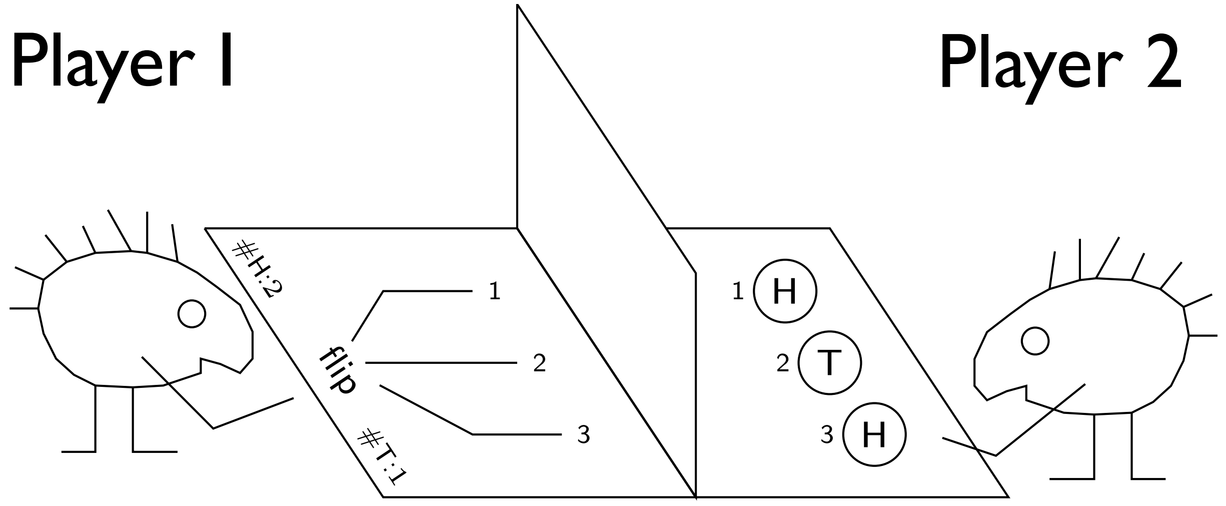
CAMBRIDGE

Lectures on Game Theory for Computer Scientists

EDITED BY
Krzysztof R. Apt
AND
Erich Grädel

CAMBRIDGE

3-coin game

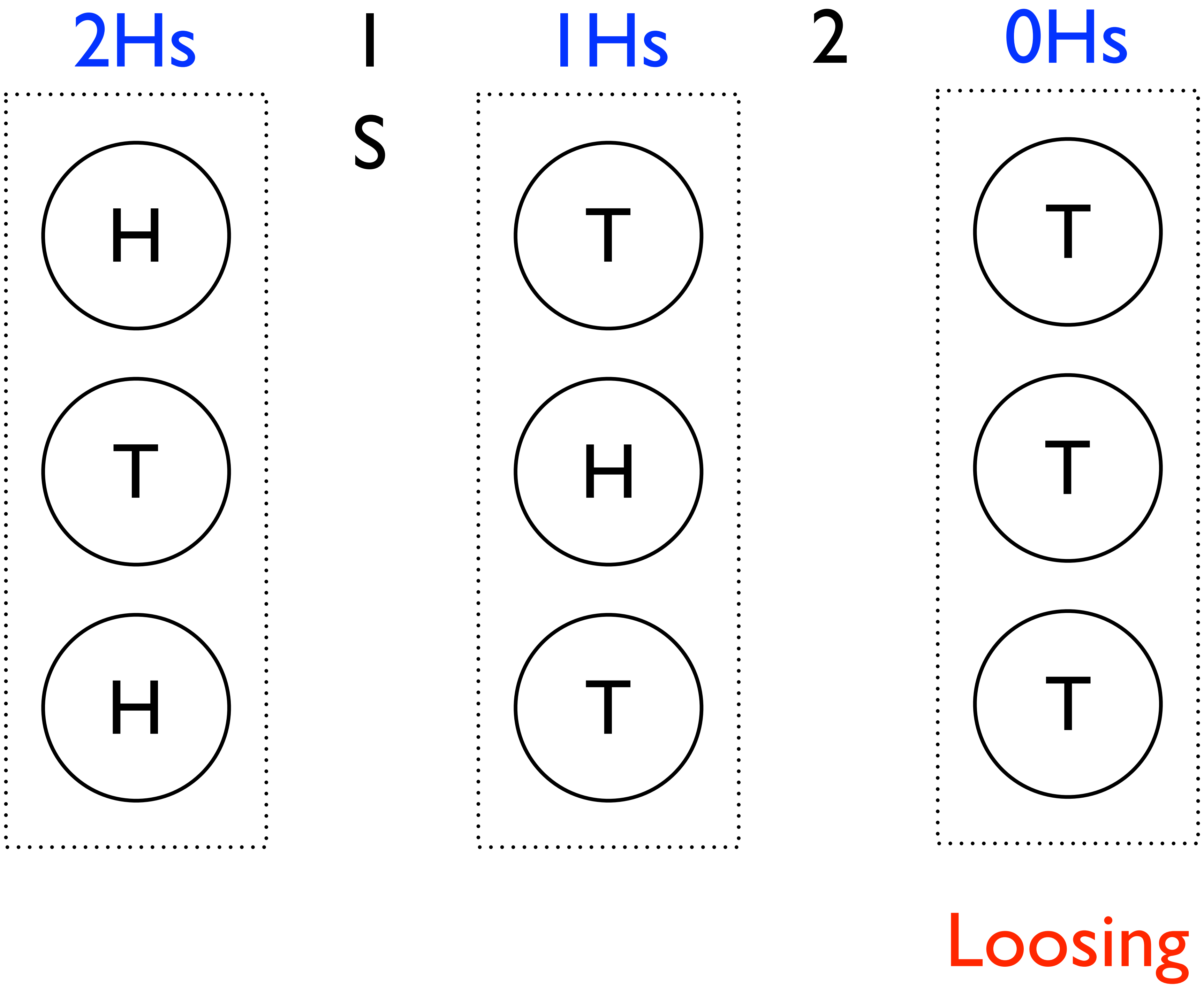


- Player 1 does not see the coins but knows how many coins are on H (imperfect information). Player 2 does see them (perfect information).

- Initially, two coins are on H. Then rounds are played as follows: Player 1 chooses a coin C in {*1,2,3*}. Player 2 flips C, then he decides to exchange or not the position of the other two coins. He announces the number of H to Player 1.

- Player 1 wins when all coins are on H, Player 2 wins when all coins are on T or if the game never reaches 3 coins on H.

3-coin game



2Hs 1 S 1Hs 2 0Hs

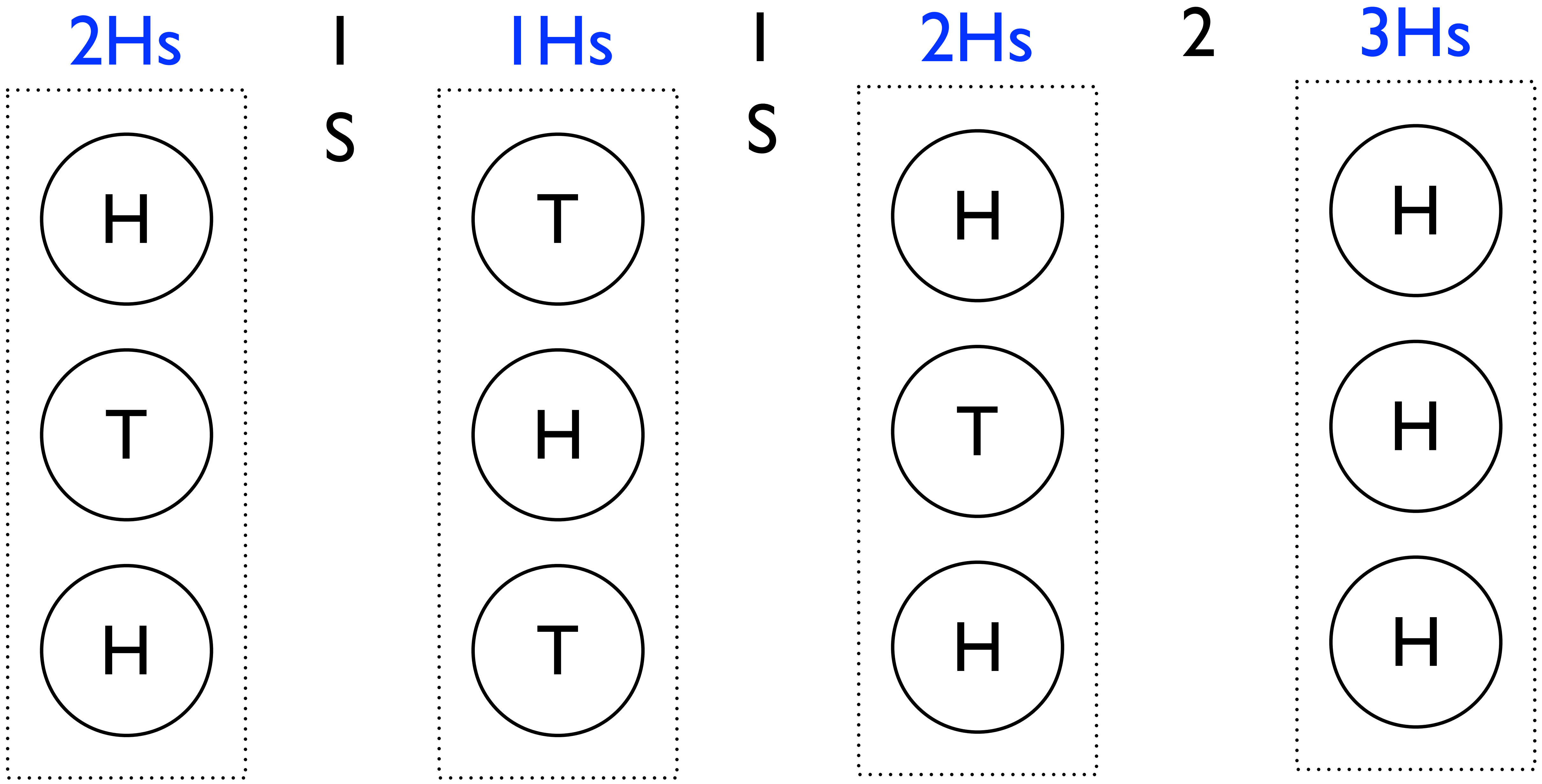
H T H

T H T

T T T

Loosing

3-coin game



2Hs

H

T

H

1

S

1Hs

T

H

T

1

S

2Hs

H

T

H

2

3Hs

H

H

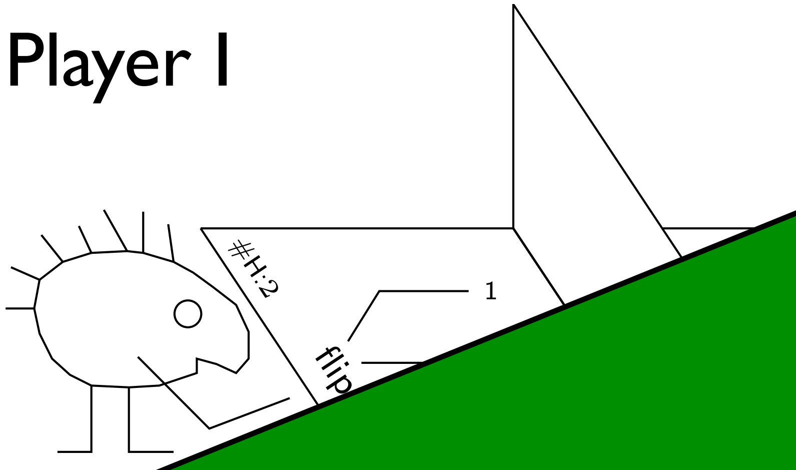
H

Winning

3-coin game

Player I

Player 2



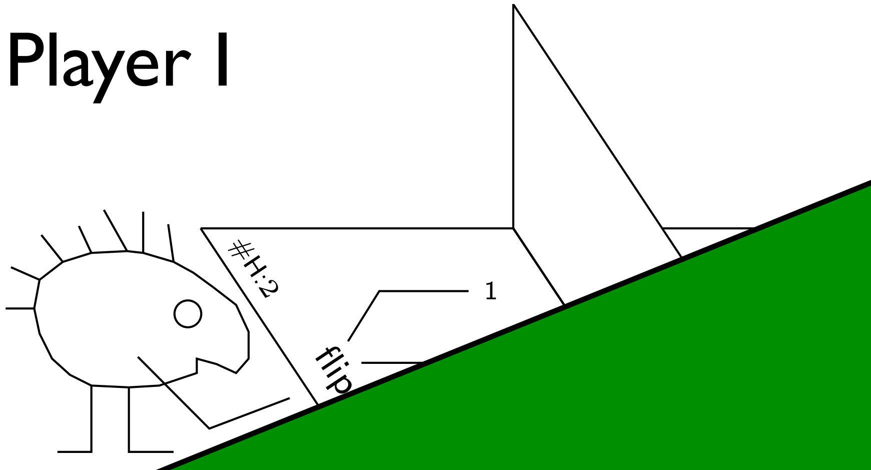
Does Player I have a strategy to win this game ?

- Player I H (imperfect informat
- Initially, two layer 1 chooses a coin C in {1, ange or not the position of the other of H to Player 1.
- Player 1 wins wh H, Player 2 wins when all coins are on T or if the game never r oins on H.

3-coin game

Player 1

Player 2



Does Player 1 win in the variant where Player 2 is not allowed to exchange the coins that are not flipped ?

- Player 1 H (imperfect informat

- Initially, two layer 1 chooses a coin C in {1, ange or not the position of the other of H to Player 1.

- Player 1 wins w H, Player 2 wins when all coins are on T or if the game never r oins on H.

Content of this course

- **Game structures with imperfect information** to model games such as the 3-coin game.

- Two variants:
deterministic strategies vs **randomized strategies**.

- Algorithms to **decide** who is winning and **synthesize winning strategies** when they exist.

Plan

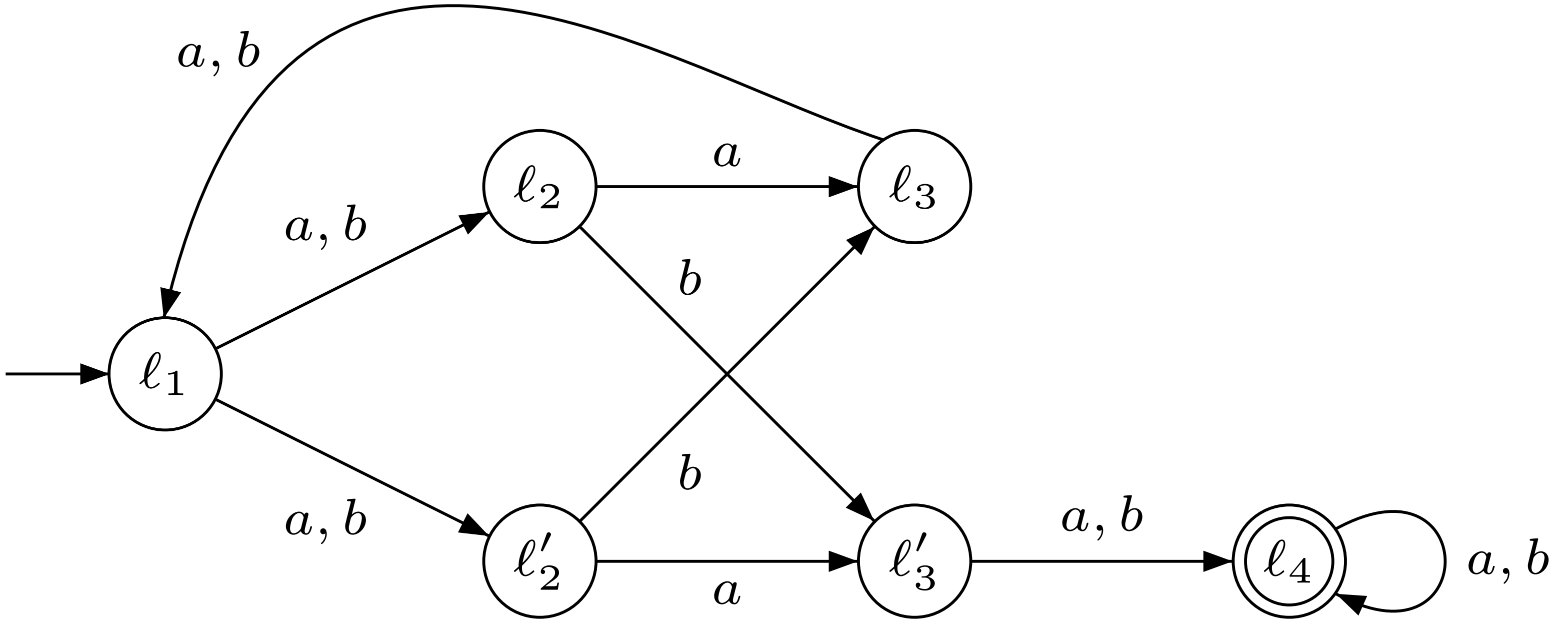
- Preliminaries:
  Game structures with **perfect** information
- Game structures with **imperfect** information
- **Deterministic** strategies (with memory)
- **Efficient** algorithms (antichains)
  + applications in automata theory
- **Randomized** strategies (with memory)

Preliminaries
Games of perfect information

Game structure of perfect information

- A two-player **game structure** of **perfect information** $(L, l_{init}, \Sigma, \Delta)$ is composed of:

  (*i*) *L* is a finite set of **locations**,
  (*ii*) l_{init} is the **initial location**,
  (*iii*) Σ is a finite alphabet of **actions**, and
  (*iv*) $\Delta \subseteq L \times \Sigma \times L$ is a set of **transitions** s.t. $\forall l \in L \cdot \exists \sigma \in \Sigma \cdot \exists (l, \sigma, l') \in \Delta$.



Rounds

- Games of perfect information are played by the two players for an **infinite** number of **rounds**.
  - *Round 0.* The game starts in the initial location l_i.
  - *Round i.* If l_l is the current location,

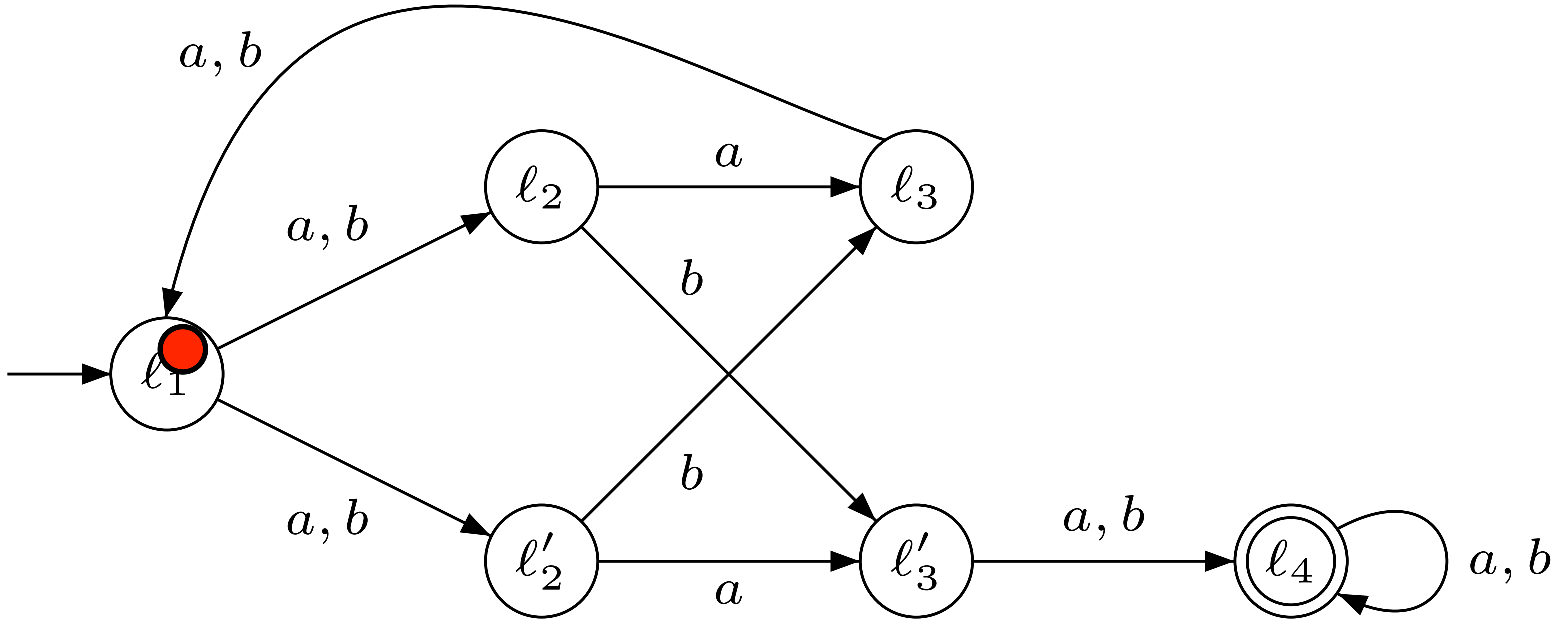
    ① Player 1 **choose an action $\sigma \in \Sigma$**, and

    ② Player 2 **resolves nondeterminism**

    by choosing a location in $\{ l_2 \mid (l_1, \sigma, l_2) \in \Delta \}$

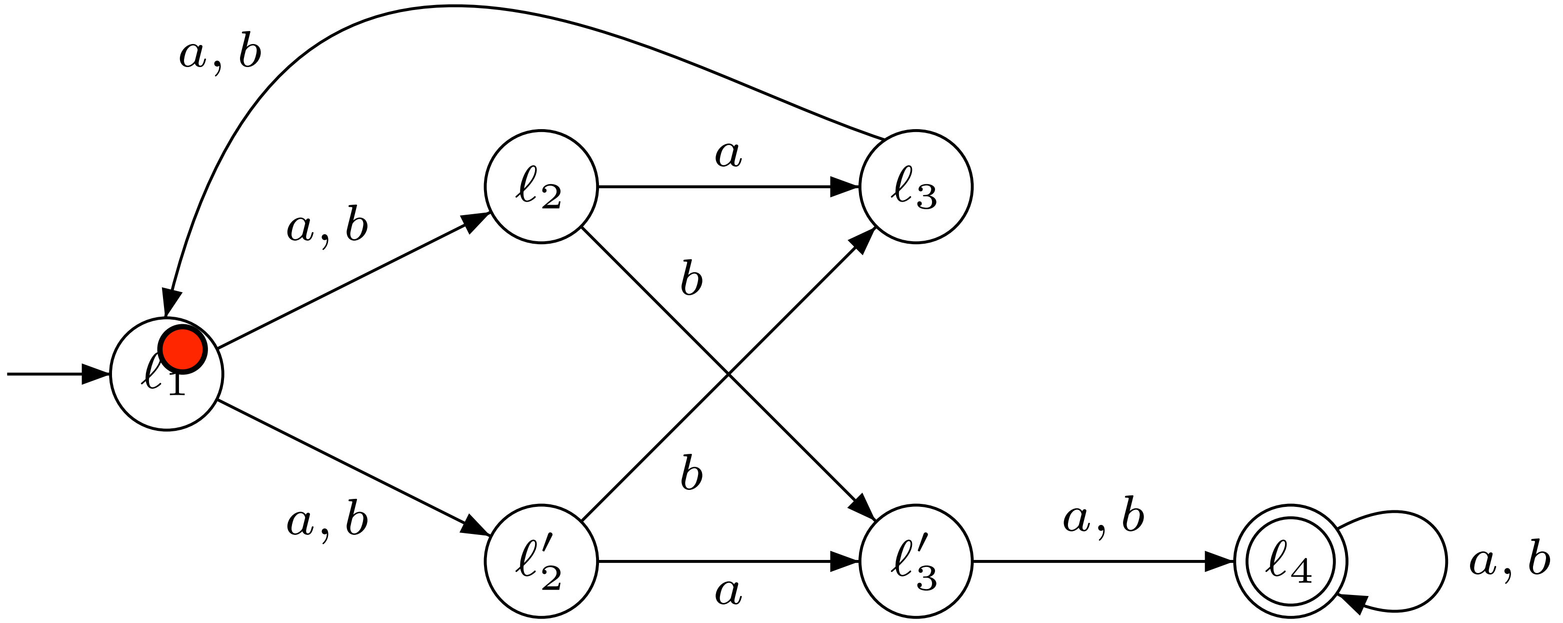
  - *Round i+1* is started.

Rounds



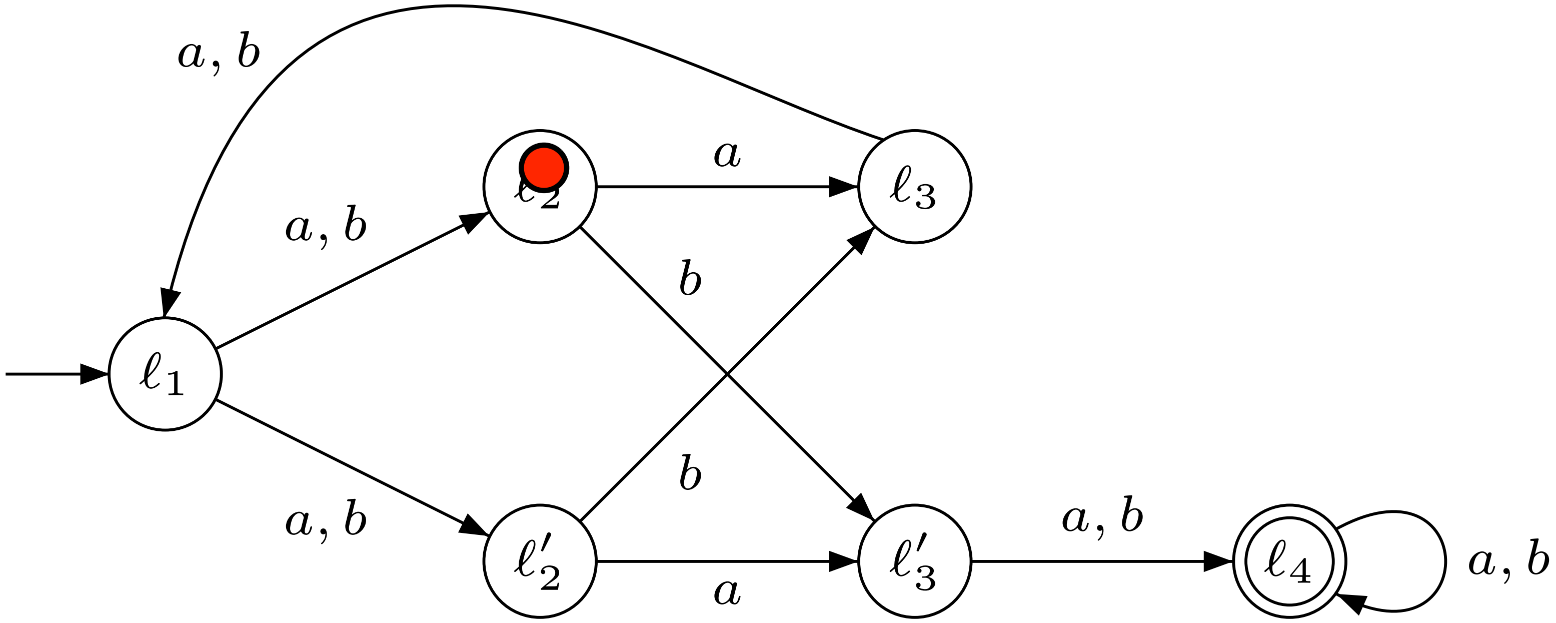
I_I

Rounds



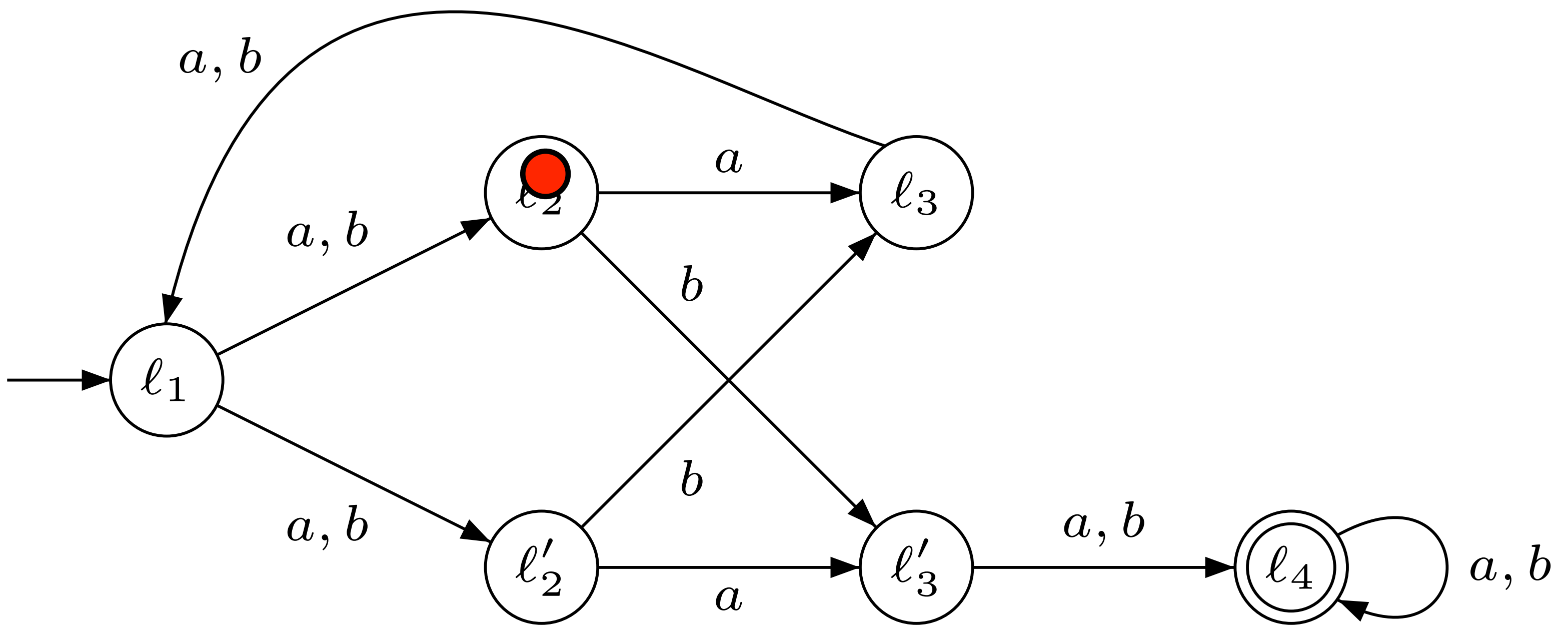
l_I a

Rounds



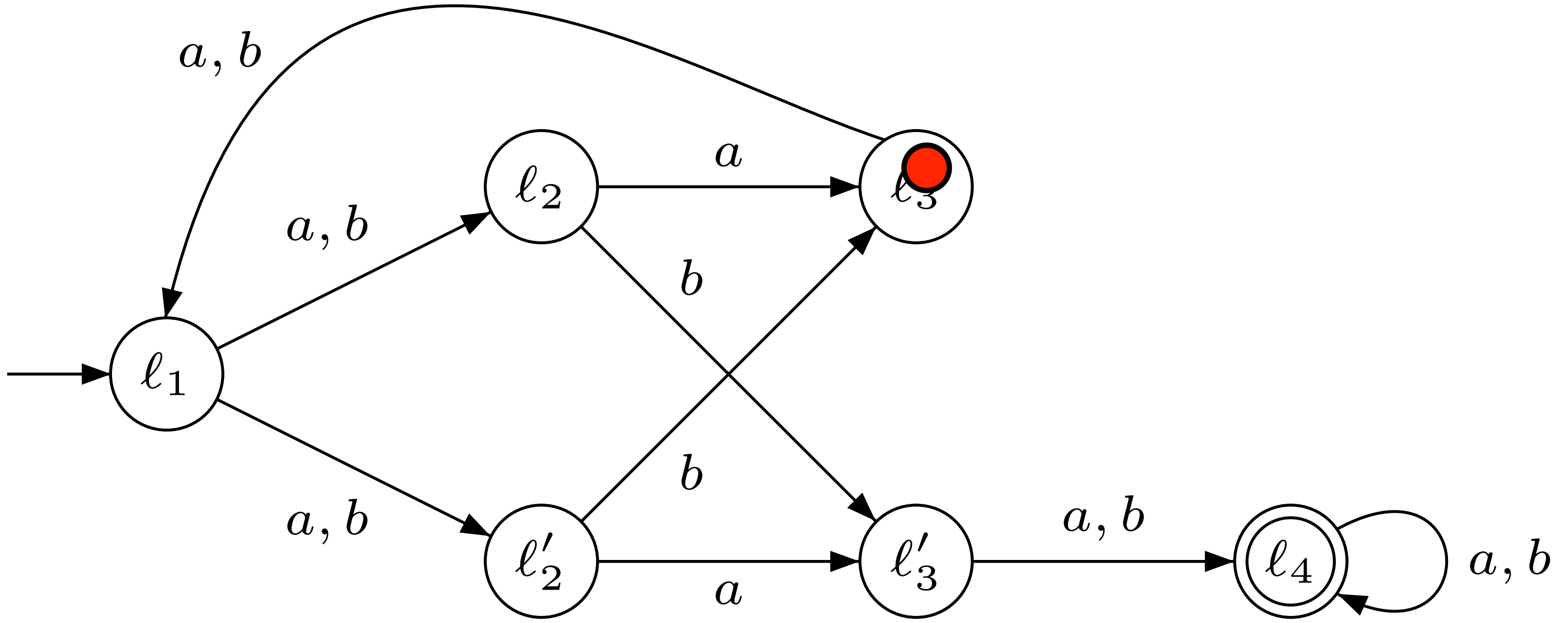
$l_1\ a\ l_2$

Rounds



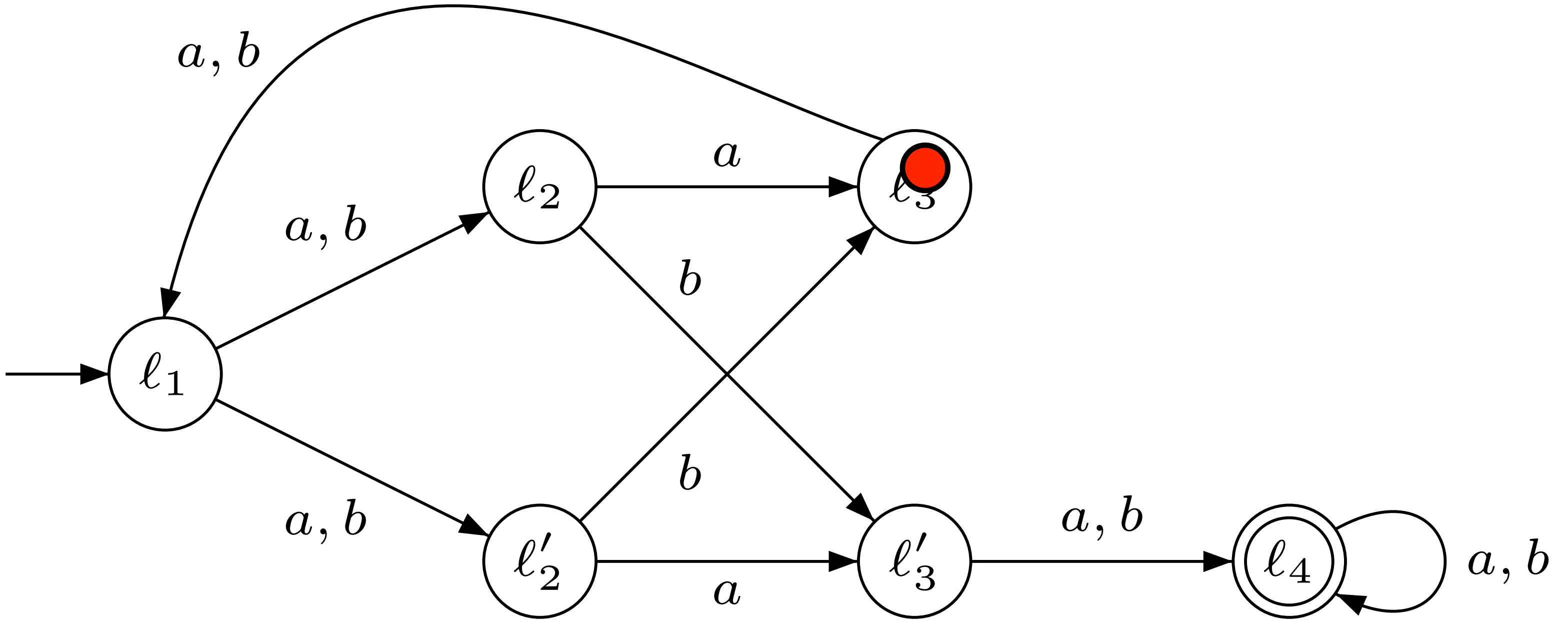
l_1 a l_2 a

Rounds



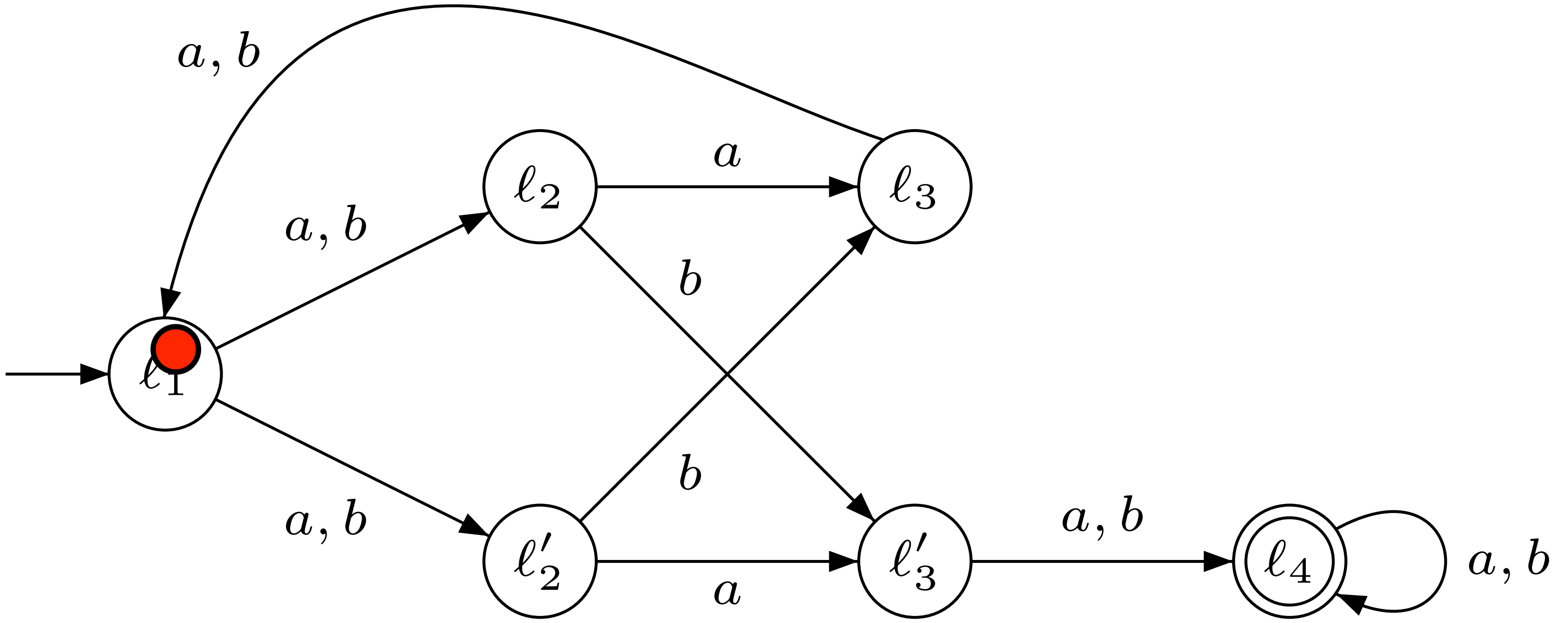
l_1 a l_2 a l_3

Rounds



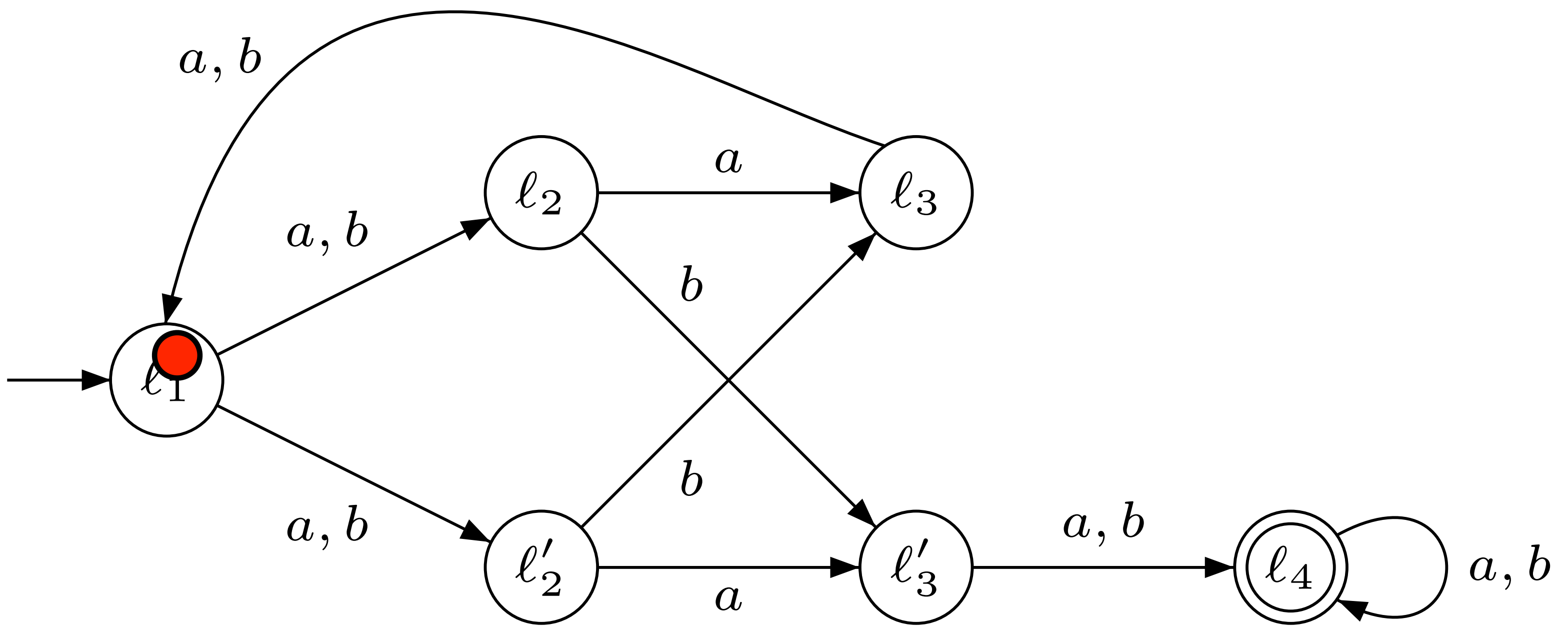
l_1 a l_2 a l_3 b

Rounds



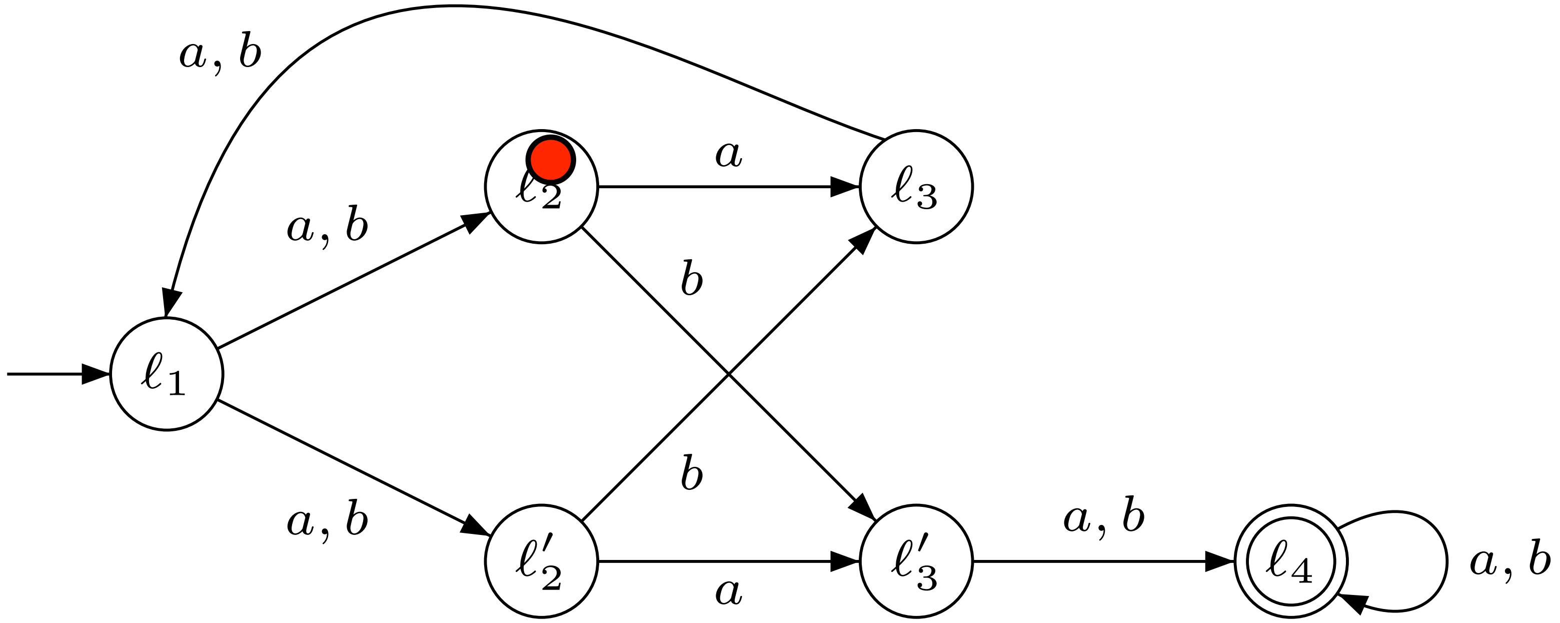
l_1 a l_2 a l_3 b l_1

Rounds



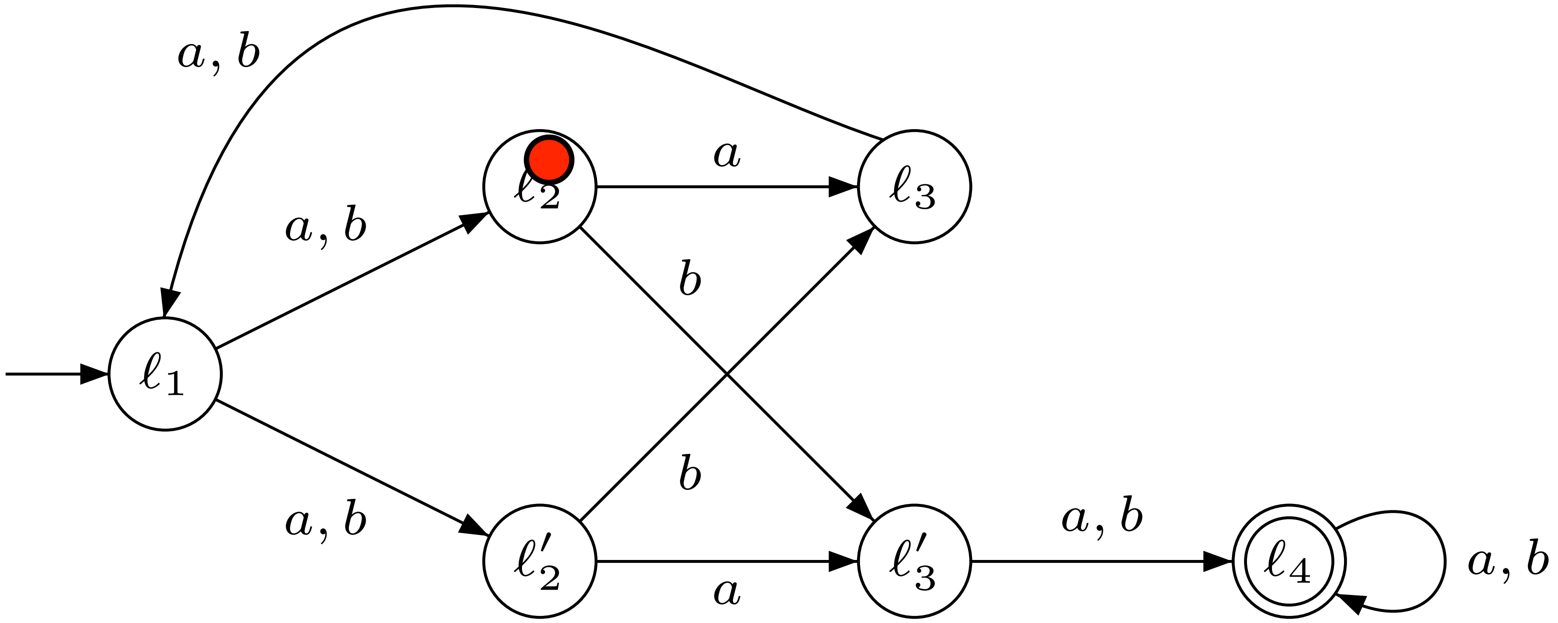
$l_1\ a\ l_2\ a\ l_3\ b\ l_1\ b$

Rounds



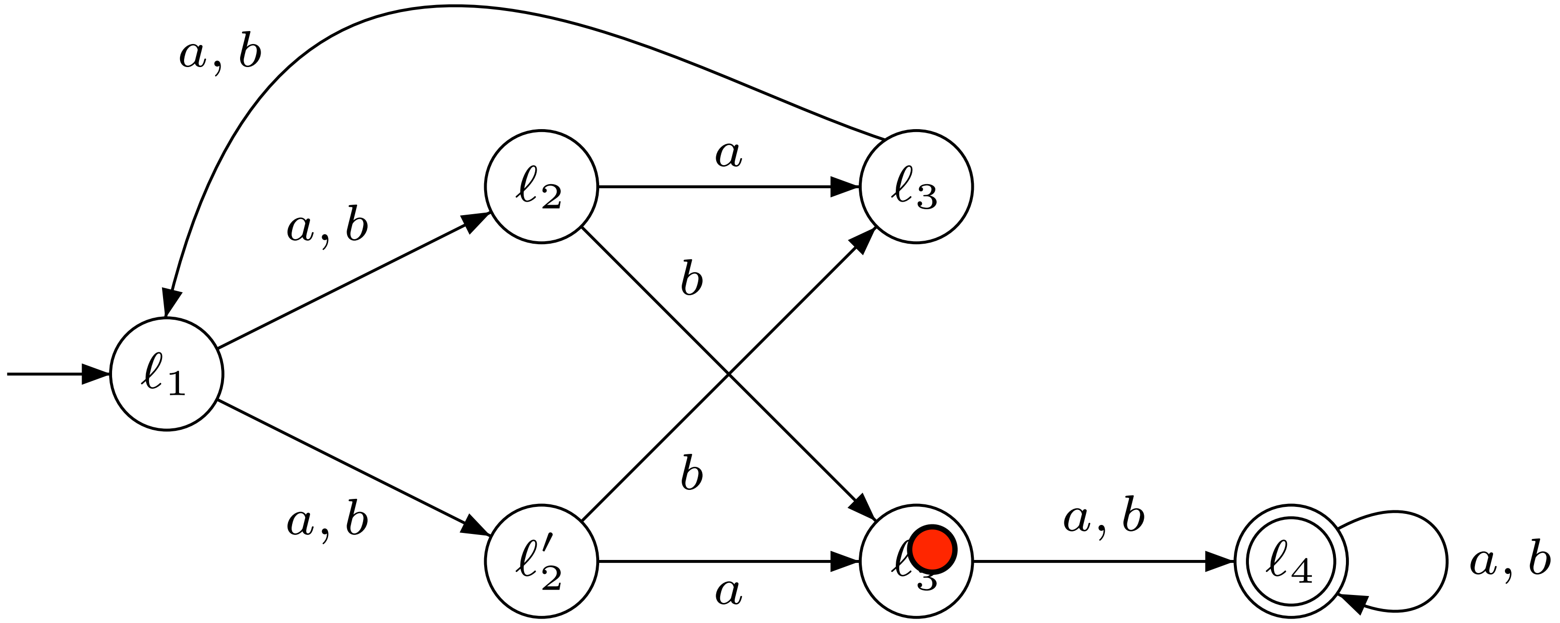
$l_1\ a\ l_2\ a\ l_3\ b\ l_1\ b\ l_2$

Rounds



l_1 a l_2 a l_3 b l_1 b l_2 b

Rounds



l_1 a l_2 a l_3 b l_1 b l_2 b l_3' ...

Play, Inf, History

- A **play** is an infinite sequence of locations $\pi = l_0 l_1 \ldots l_n \ldots$ such that
  - $l_0 = l_{init}$, and
  - $\forall i \geq 0 \cdot \exists \sigma \in \Sigma \cdot (l_i, \sigma, l_{i+1}) \in \Delta$.
- We denote by **Inf**(π) the set of locations that appear **infinitely many times** along π.
- A **history** of a play $\pi = l_0 l_1 \ldots l_n \ldots$ is a finite prefix of the play.
  - $\pi(j) = l_0 l_1 \ldots l_j$. is the prefix that ends in position $j \geq 0$.
  - Its **length**, denoted $|\pi(j)| = j+1$.
  - We use **Last**($\pi(j)$) to denote l_j.

Deterministic strategies
Memoryless strategies

- A **deterministic strategy for Player 1** is a function **$\alpha : L^+ \rightarrow \Sigma$** that maps histories to actions.

  A_G denotes the set of Player 1's strategies in game G.

- A **deterministic strategy for Player 2** is a function **$\beta : L^+ \times \Sigma \rightarrow L$** s.t.

  $$\forall \rho \in L^+ \cdot \forall \sigma \in \Sigma \cdot (\mathbf{Last}(\rho), \sigma, \beta(\rho,\sigma)) \in \Delta.$$

  B_G denotes the set of Player 2's strategies in game G.

- A strategy $\alpha \in A_G$ is **memoryless** if

  $$\forall \rho, \rho' \in L^+ \cdot \mathbf{Last}(\rho) = \mathbf{Last}(\rho') \Rightarrow \alpha(\rho) = \alpha(\rho').$$

  i.e. memoryless strategy depends only on the last location of the history.

Outcome of deterministic strategies

- The **outcome** of a deterministic strategy α for Player 1 and of a deterministic strategy β for Player 2 is the play

  $\pi = l_0 l_1 \ldots l_n \ldots$ such that: 1) $l_0 = l_{init}$
  2) $\forall i \geq 0$, $\sigma_i = \alpha(\pi(i))$ and $l_{i+1} = \beta(\pi(i), \sigma_i)$.

  This play is denoted by **outcome**(G, α, β)

- A play π is **consistent** with a Player 1's strategy α if

  $\pi =$ **outcome**(G, α, β) for some Player 2's strategy β.

- We note **Outcome$_1$**(G, α) the set of plays consistent with α.

Objectives (winning conditions)

- Given a game structure $G=(L,l_i,\Sigma,\Delta)$, an **objective** is a set of sequences of locations, i.e. a subset of L^ω. By $\not\rho$ we denote $L^\omega\setminus\rho$.
- A **reachability objective** is defined by a set of target locations $T\subseteq L$.
  $$\textbf{Reach}(T)=\{\ l_0\ l_1\ l_2\ \ldots \mid \exists j\geq 0 \bullet l_j \in T\ \}.$$
- A **safety objective** is defined by a set of safe locations $S\subseteq L$.
  $$\textbf{Safe}(S)=\{l_0\ l_1\ l_2\ \ldots \mid \forall j\geq 0 \bullet l_j \in S\ \}.$$
- A **Büchi objective** is defined by a set of target locations $T\subseteq L$.
  $$\textbf{Büchi}(T)=\{\ \pi \mid \textbf{Inf}(\pi)\cap T\neq\varnothing\ \}.$$
- A **coBüchi objective** is defined by a set of safe locations $S\subseteq L$.
  $$\textbf{coBüchi}(S)=\{\ \pi \mid \textbf{Inf}(\pi)\subseteq S\ \}.$$
- A **parity objective** is defined by a function $pr : L \rightarrow \{\ 0,1,\ldots,d\ \}$.
  $$\textbf{Parity}(pr)=\{\ \pi \mid \underline{\text{min}}\{\ pr(l) \mid l \in \textbf{Inf}(\pi)\ \} \text{ is } \underline{\text{even}}\ \}.$$

Objectives (winning conditions)

- Given a game structure $G=(L,l_i,\Sigma,\Delta)$, an **objective** is a set of sequences of locations, i.e. a subset of L^ω. By $\overline{\rho}$ we denote $L^\omega \setminus \rho$.

- A **reachability objective** is defined by a set of target locations $T \subseteq L$.

  Reach$(T)=\{\ l_0\ l_1\ l_2\ \ldots \mid \exists j \geq 0 \bullet l_j \in T\ \}$.

- A **safety objective** is defined by a set of safe locations $S \subseteq L$.

  Safe$(S)=\{l_0\ l_1\ l_2\ \ldots \mid \forall j \geq 0 \bullet l_j \in S\ \}$.

- A **Büchi objective** is defined by a set of ta

  Büchi$(T)=\{\ \pi \mid$ I

- A **coBüchi obj**

- A **parity object** $p : L \rightarrow \{\ 0,1,\ldots,d\ \}$.

  Parity $(\cdot) \mid l \in$ **Inf**$(\pi)\ \}$ is <u>even</u> $\}$.

Parity objectives nicely **generalize** all the other classes of objectives !

Parity is expressive enough to cover all **omega-regular objectives** !

Surely-winning - Determinacy

- Let G be a game structure and $\rho \subseteq L^{\omega}$ be an objective.

- The deterministic strategy α is **surely-winning** in G for ρ iff $\mathbf{Outcome_1}(G,\alpha) \subseteq \rho$. (similarly for Player 2).

- We say that (G,ρ) is **determined** iff either
  Player 1 has a surely-winning strategy α for the objective ρ, **or**
  Player 2 has a surely-winning strategy β for the objective $\overline{\rho}$.

- **Theorem (Determinacy).** For all game structures of perfect information G, for all parity objectives ρ, the game (G,ρ) is determined.

- **Theorem (Memoryless).** For all game structures of perfect information G, for all parity objectives ρ:

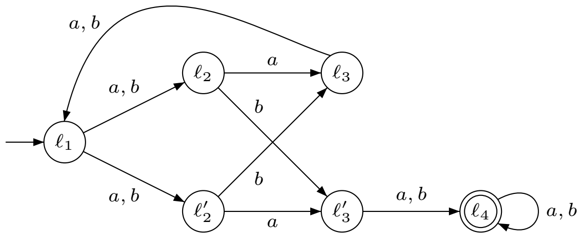
  Player 1 (Player 2) has a surely-winning strategy in (G,ρ)

  iff

  Player 1 (Player 2) has a **memoryless** surely-winning strategy in (G,ρ).

Summary

Game structure



Rounds, plays, history

l_1 *a* l_2 *a* l_3 *b* l_1 *b* l_2 *b* l_3' ...

Strategies

Player 1proposes letters: $\alpha : L^+ \to \Sigma$
Player 2 resolves nondeterm.: $\beta : L^+ \times \Sigma \to L$

Objectives

$\rho \subseteq L^\omega$
Safety, Reachability, (co)Büchi, Parity

Player 1 **wins** (G,ρ) iff $\exists\alpha\bullet\forall\beta\bullet\text{outcome}(\alpha,\beta)\in\rho$
iff $\neg(\exists\beta\bullet\forall\alpha\bullet\text{outcome}(\alpha,\beta)\in\cancel{\rho})$

Algorithms - Cpre

- The **controller predecessor operator**

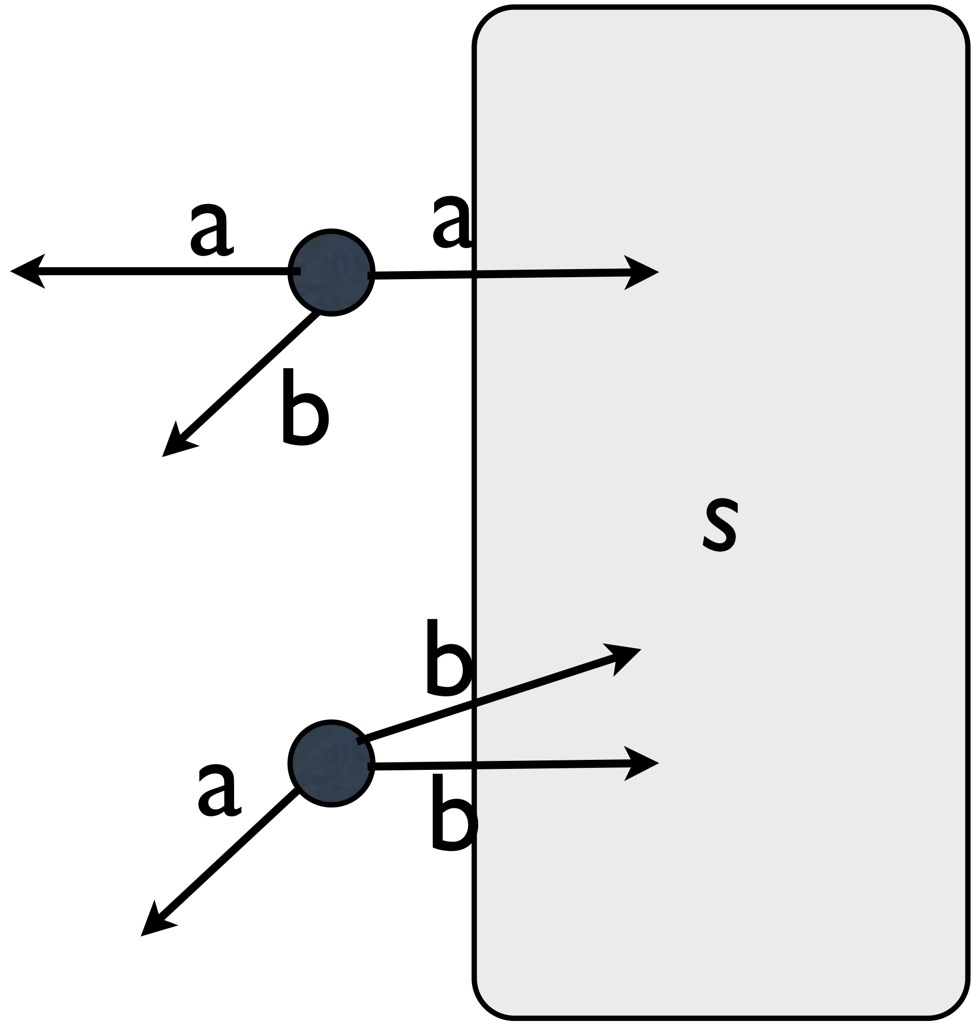
$$\mathbf{Cpre} : 2^L \rightarrow 2^L$$

given a set of locations $s \subseteq L$, returns the set of locations $l \in L$, from which Player I can force the game to be in S in the next round.

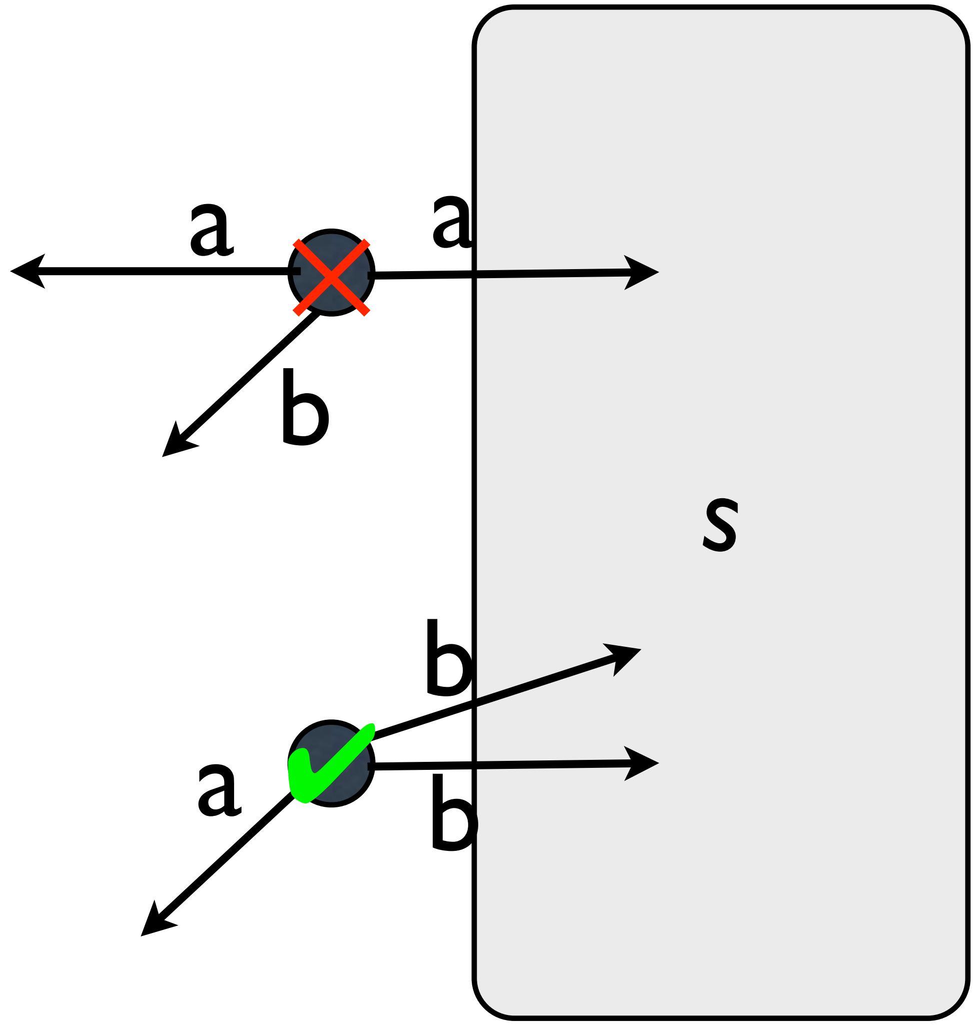
$$\mathbf{Cpre}(S) = \{ l \mid \exists \sigma \in \Sigma \cdot \forall l' \in L \cdot (l,\sigma,l') \in \Delta \Longrightarrow l' \in S \}$$
$$= \{ l \mid \exists \sigma \in \Sigma \cdot \mathbf{post}_{G,\sigma}(l) \subseteq S \}$$

where $\mathbf{post}_{G,\sigma}(l)$ is the set of successors of l by σ in G.

Algorithms - Cpre



Algorithms - Cpre



Algorithms - Safety

- Let G be a game structure of perfect information, $S \subseteq L$.
- To solve the game for the safety objective **Safe**(S), we must compute the set of locations $W \subseteq L$ from which player 1 can maintain the game within S for any number of rounds.
- Clearly $W \subseteq S$, and

  if W^i is the set of locations from which Player 1 can keep the game within S for i steps,

  then $W^{i+1} \subseteq W^i$, and W^{i+1} is exactly the set of locations within S from which Player 1 can force the game to be in W^i in the next round, i.e. $W^{i+1} = S \cap \mathbf{Cpre}(W^i)$.

Algorithms - Safety

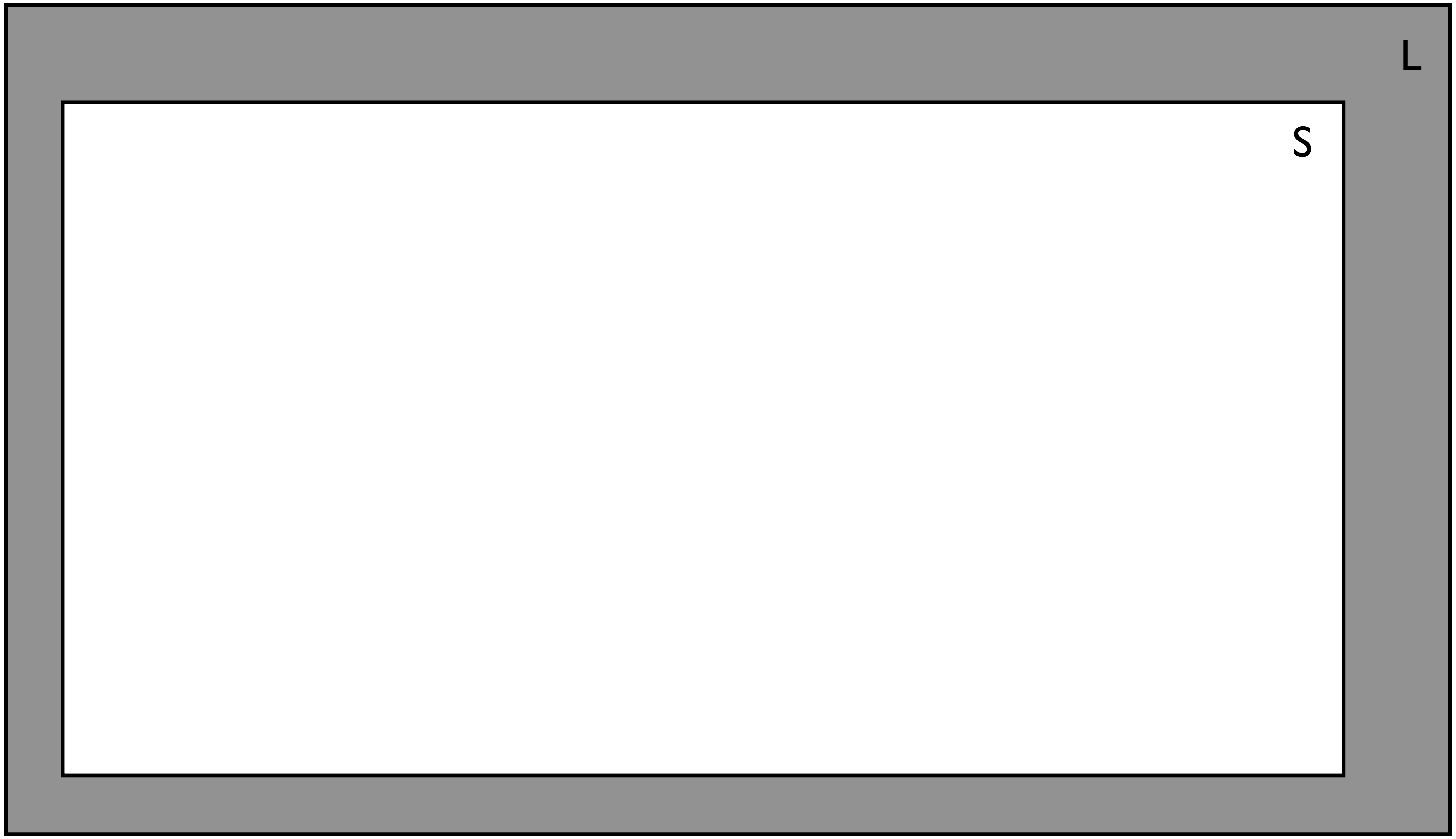
- So the set of surely-winning locations for Player 1 are obtained as the limit of the following sequence:

  $W^0 = S$;
  $W^{i+1} = S \cap \mathbf{Cpre}(W^i)$, for all $i \geq 0$.

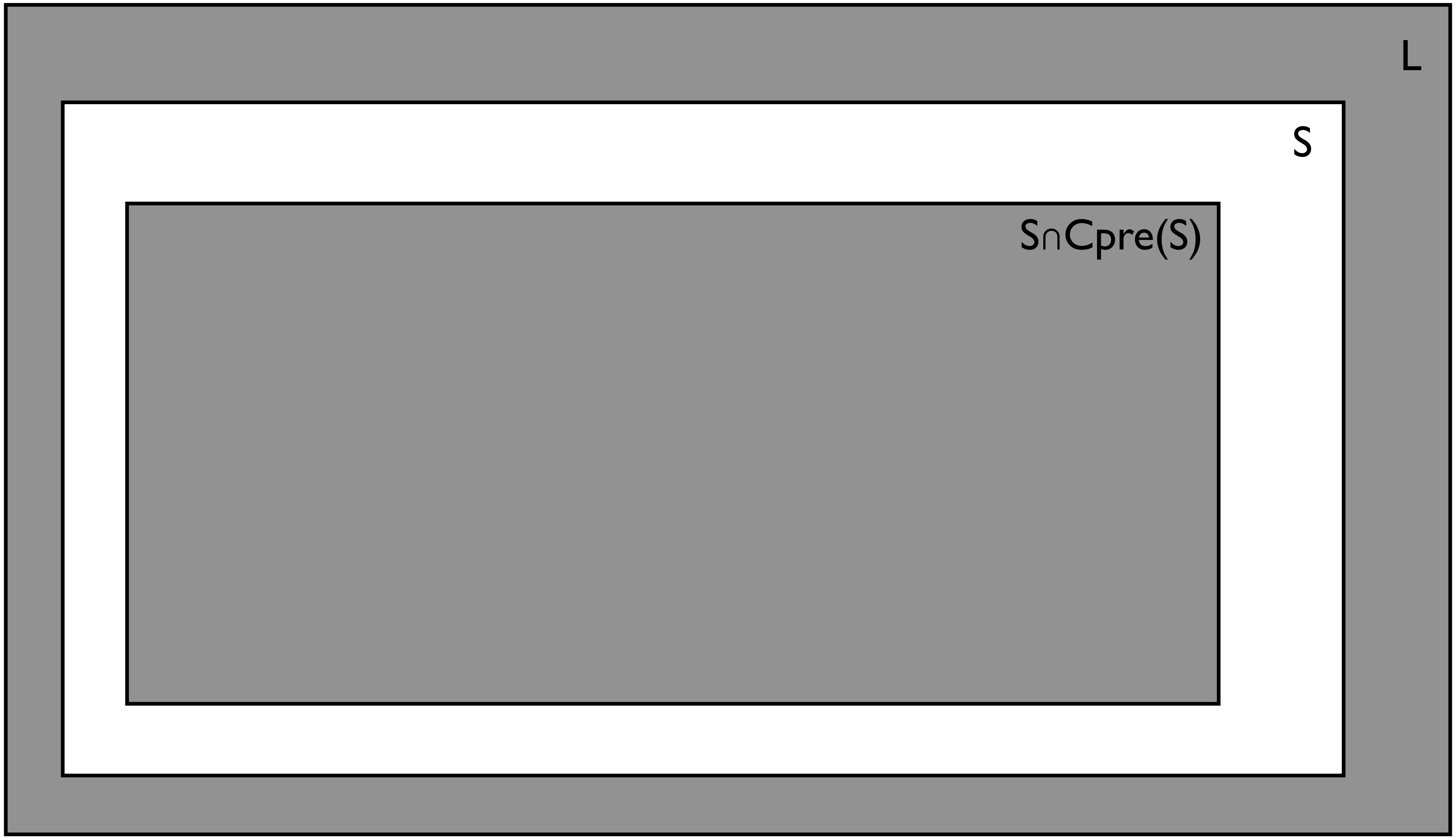
  This sequence stabilizes after at most $|S|$ steps. The limit is the greatest solution of the equation $W = S \cap \mathbf{Cpre}(W)$.

Algorithms - Safety

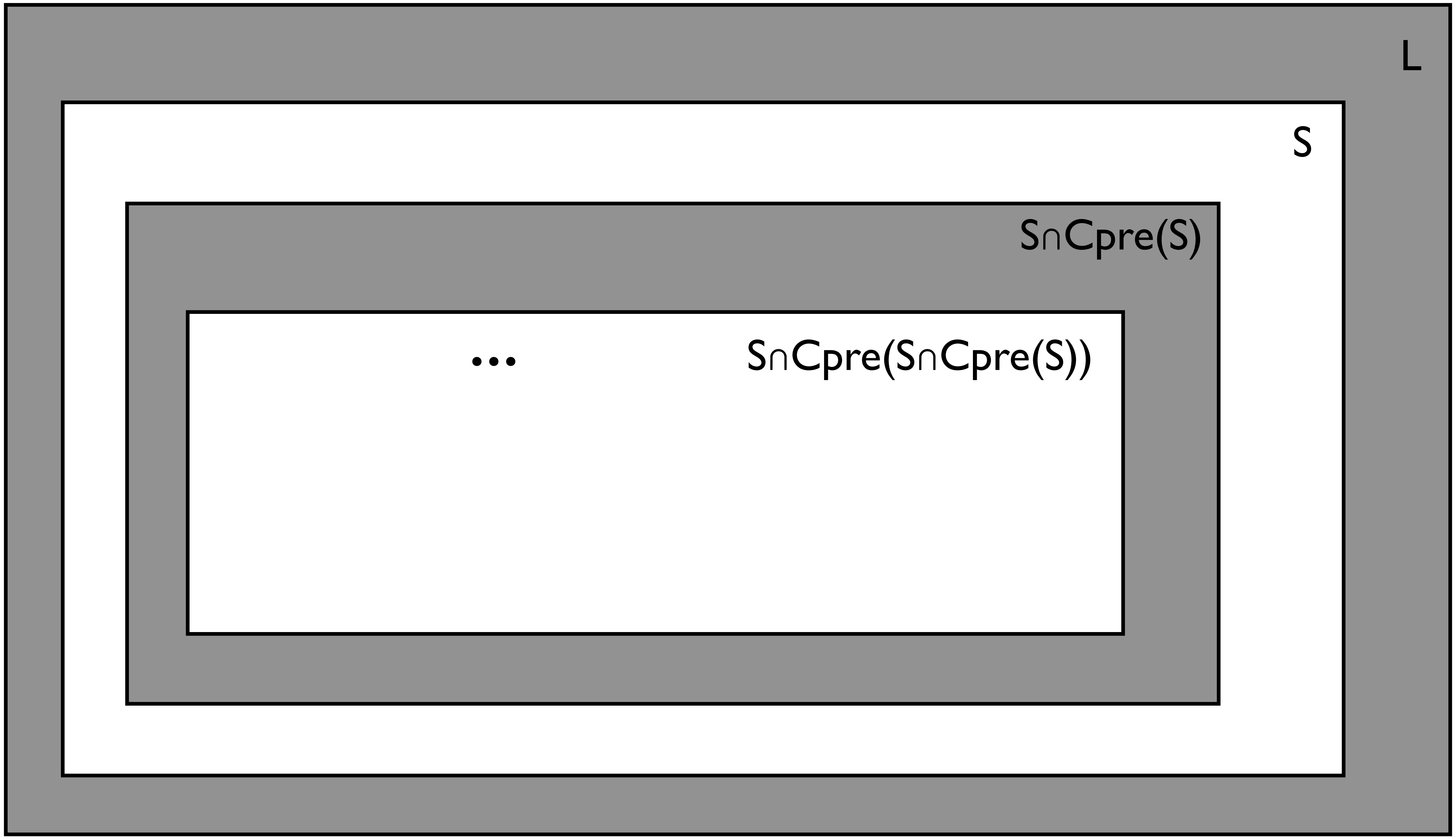
Algorithms - Safety



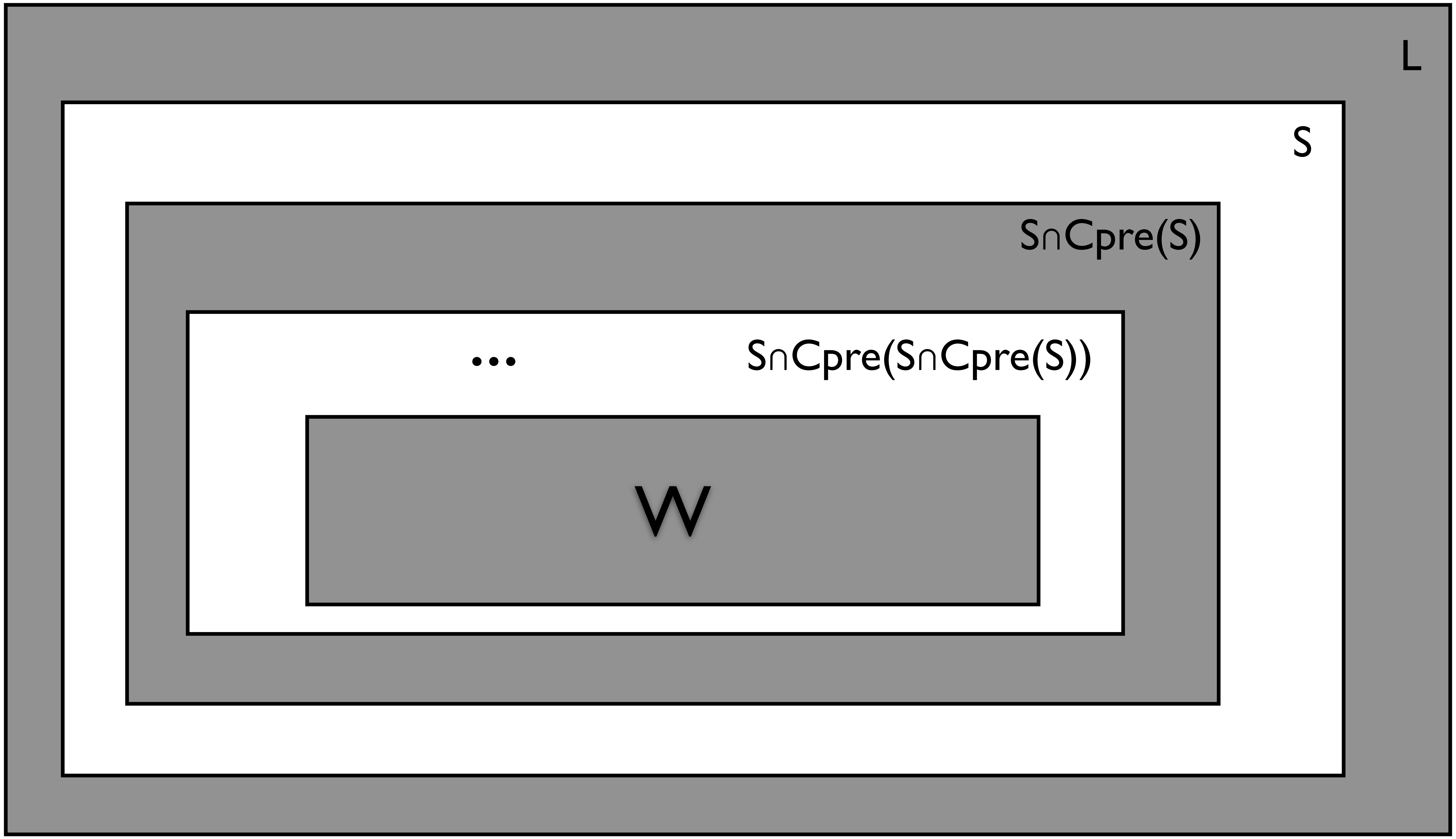
Algorithms - Safety



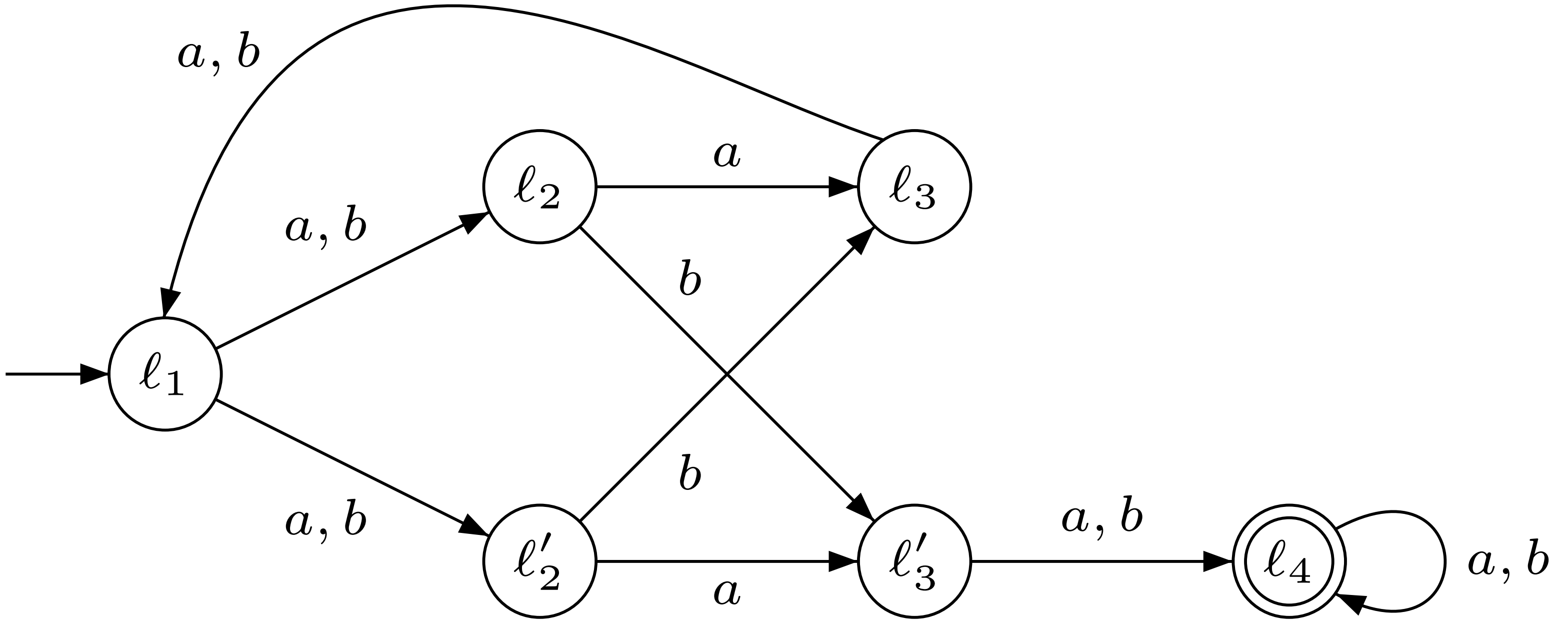
Algorithms - Safety



Algorithms - Safety

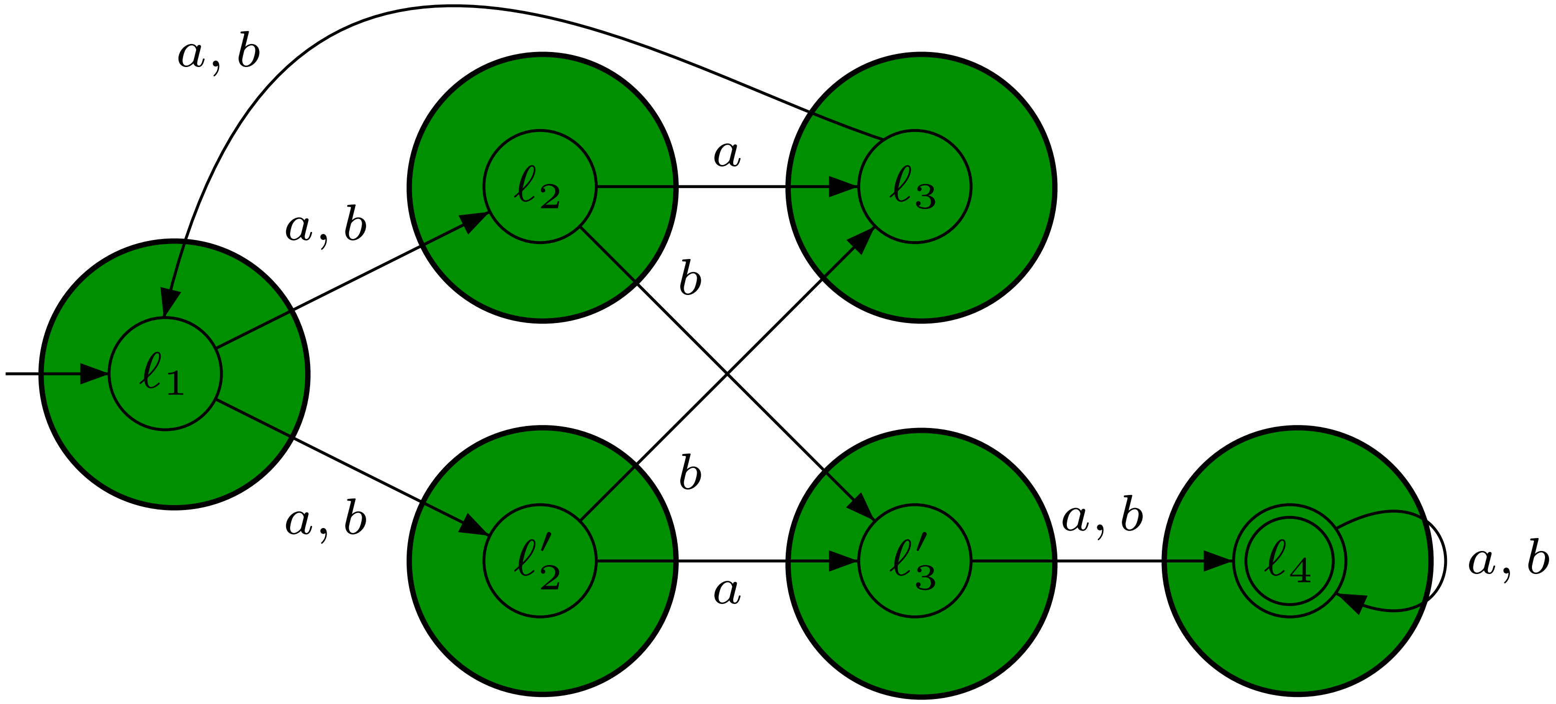


Algorithms - Safety



Let us compute the surely winning locations for the objective **Safe**(L\{*l*$_4$}).

Algorithms - Safety

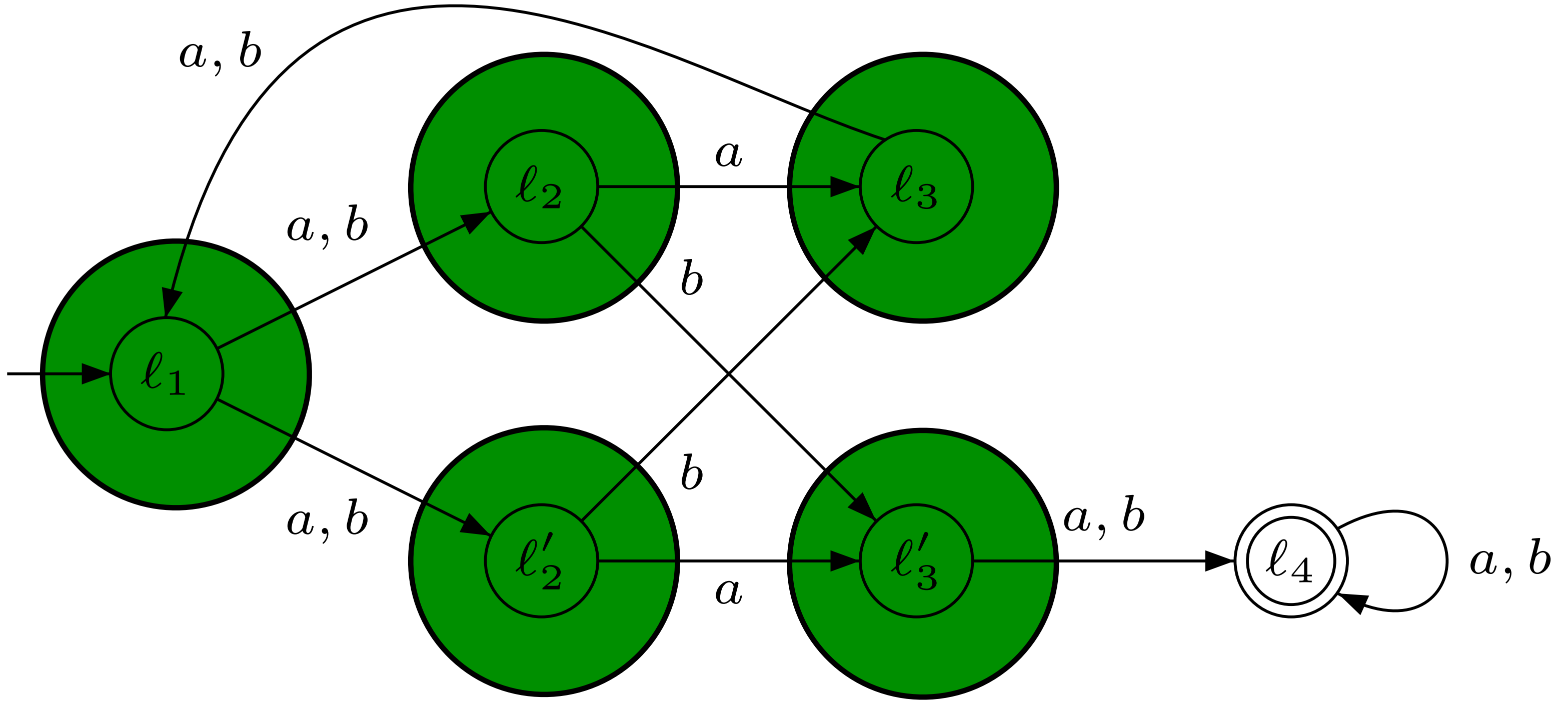


$W^0 = L \setminus \{l_4\}$

$W^1 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_4\}) = L \setminus \{l_3', l_4\}$

$W^2 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_3', l_4\}) = L \setminus \{l_3', l_4\}$

Algorithms - Safety

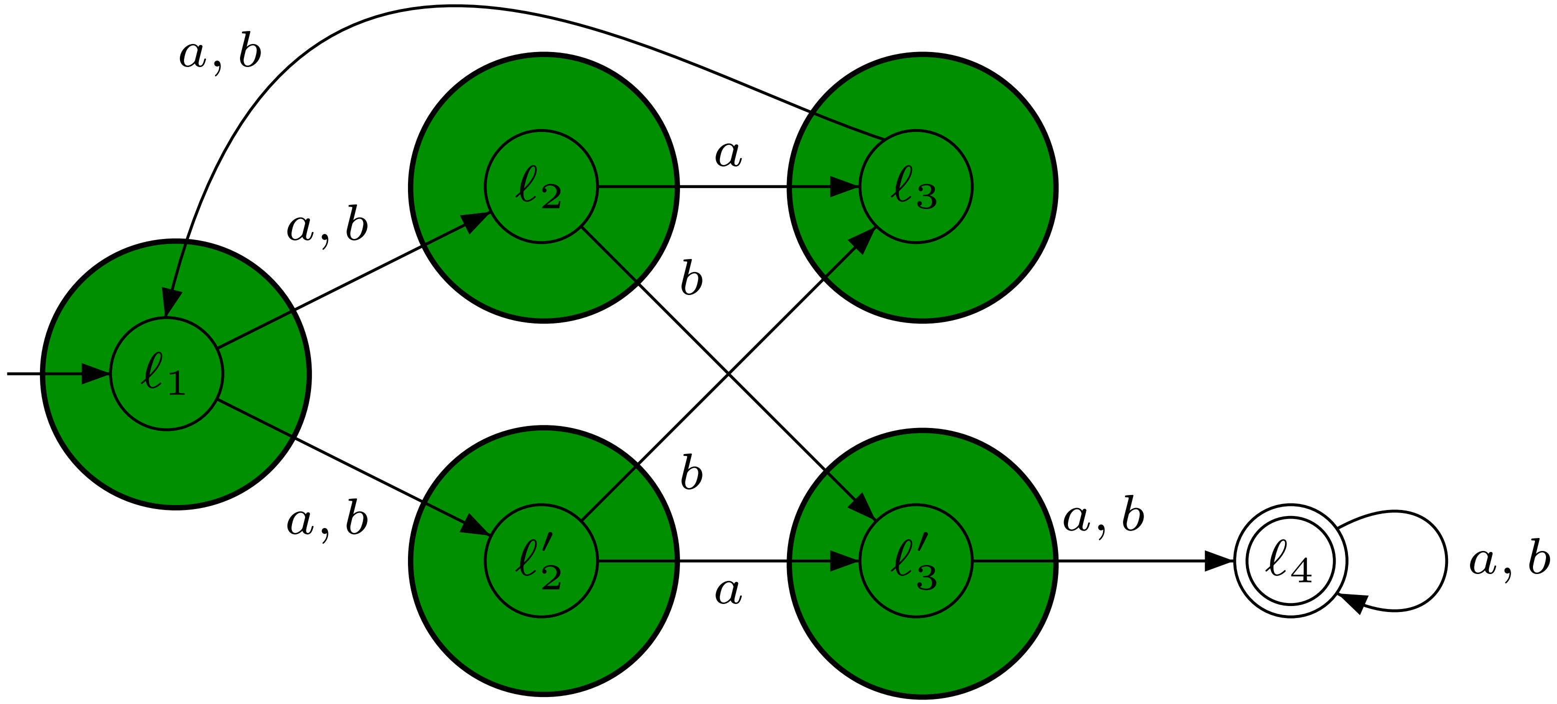


$W^0 = L \setminus \{l_4\}$

$W^1 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_4\}) = L \setminus \{l_3', l_4\}$

$W^2 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_3', l_4\}) = L \setminus \{l_3', l_4\}$

Algorithms - Safety

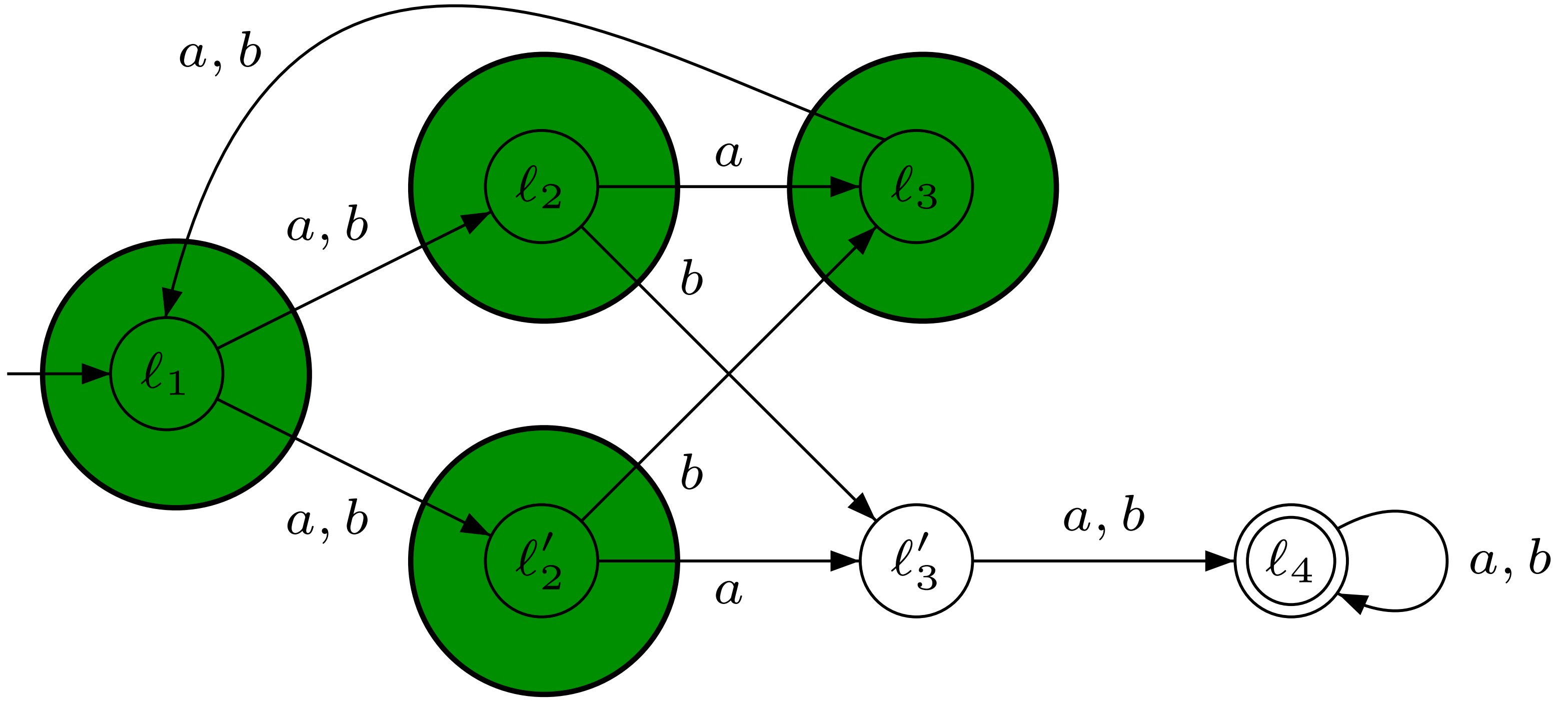


$W^0 = L \setminus \{l_4\}$

$W^1 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_4\}) = L \setminus \{l_3', l_4\}$

$W^2 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_3', l_4\}) = L \setminus \{l_3', l_4\}$

Algorithms - Safety

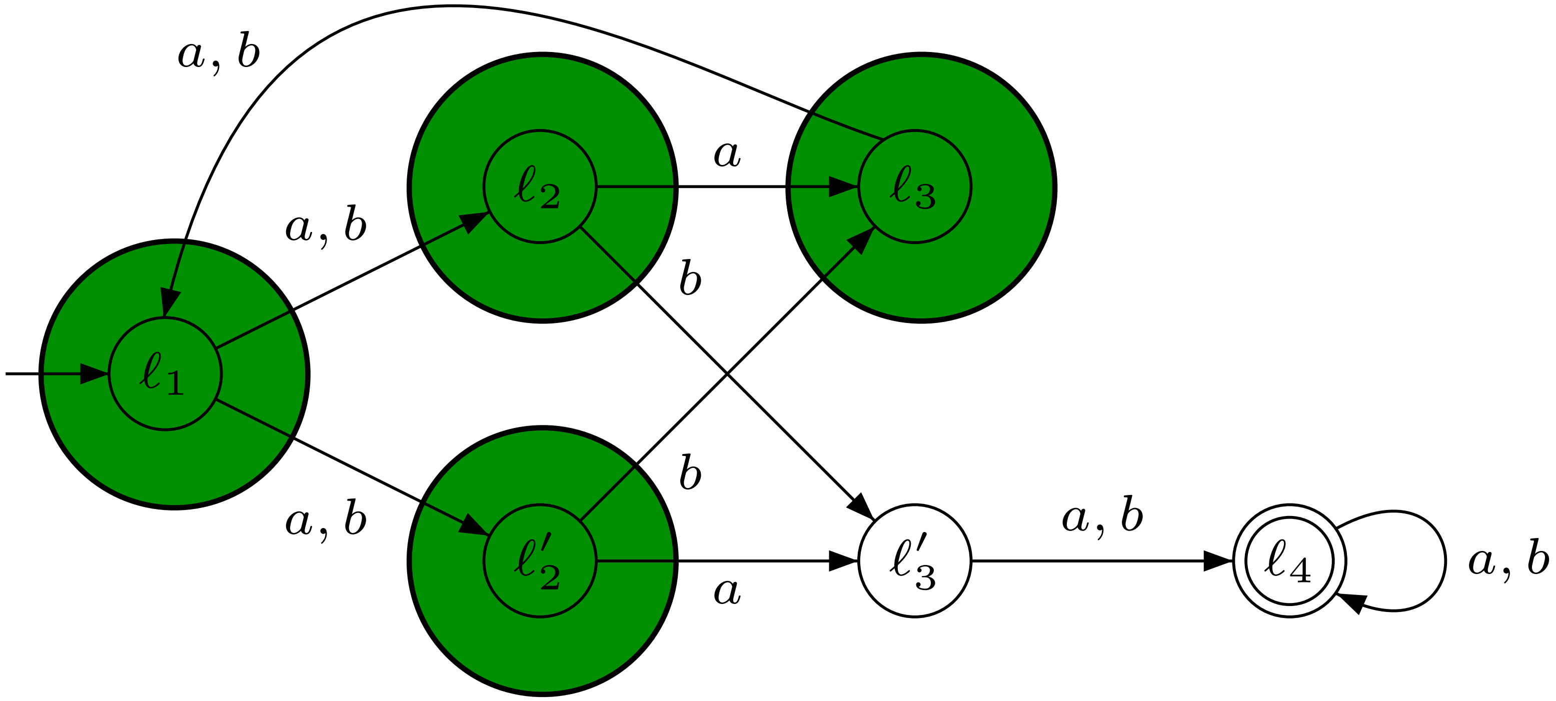


$W^0 = L \backslash \{l_4\}$

$W^1 = L \backslash \{l_4\} \cap \mathbf{Cpre}(L \backslash \{l_4\}) = L \backslash \{l_3', l_4\}$

$W^2 = L \backslash \{l_4\} \cap \mathbf{Cpre}(L \backslash \{l_3', l_4\}) = L \backslash \{l_3', l_4\}$

Algorithms - Safety

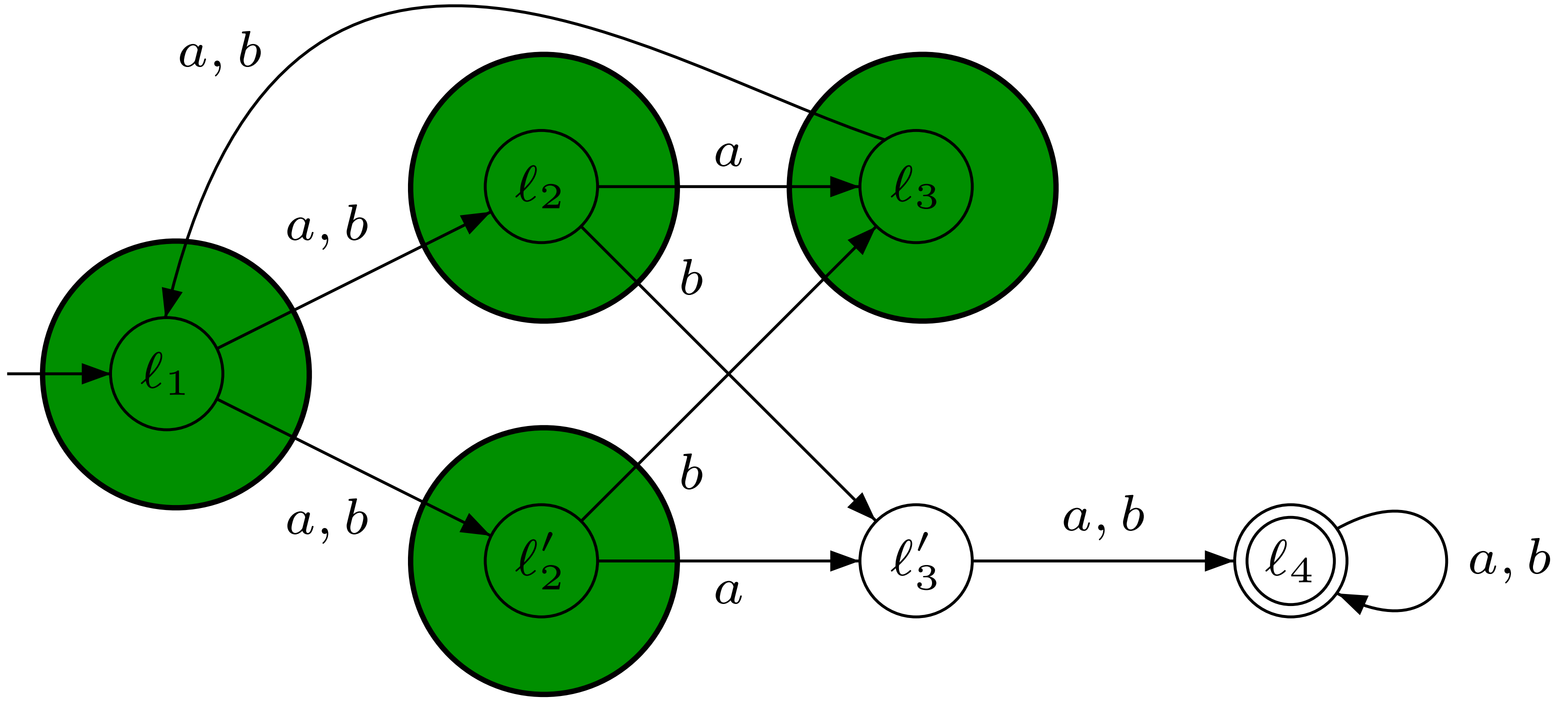


$W^0 = L \setminus \{l_4\}$

$W^1 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_4\}) = L \setminus \{l_3', l_4\}$

$W^2 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_3', l_4\}) = L \setminus \{l_3', l_4\}$

Algorithms - Safety



$W^0 = L \setminus \{l_4\}$

$W^1 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_4\}) = L \setminus \{l_3', l_4\}$

$W^2 = L \setminus \{l_4\} \cap \mathbf{Cpre}(L \setminus \{l_3', l_4\}) = L \setminus \{l_3', l_4\}$

Fixpoint

Algorithms - Reachability

- Let G be a game structure of perfect information, $T \subseteq L$.
- To solve the game for the reachability objective **Reach**(T), we must compute the set of locations $W \subseteq L$ from which player 1 can drive the game into T no matter how Player 2 resolves nondeterminism.
- Clearly $T \subseteq W$, and

  if W^i is the set of locations from which Player 1 can force the game to reach T in i steps or less,

  then W^{i+1} is the set of locations from which Player 1 can force W^i in the next round, i.e. $W^{i+1} = W^i \cup \mathbf{Cpre}(W^i)$.

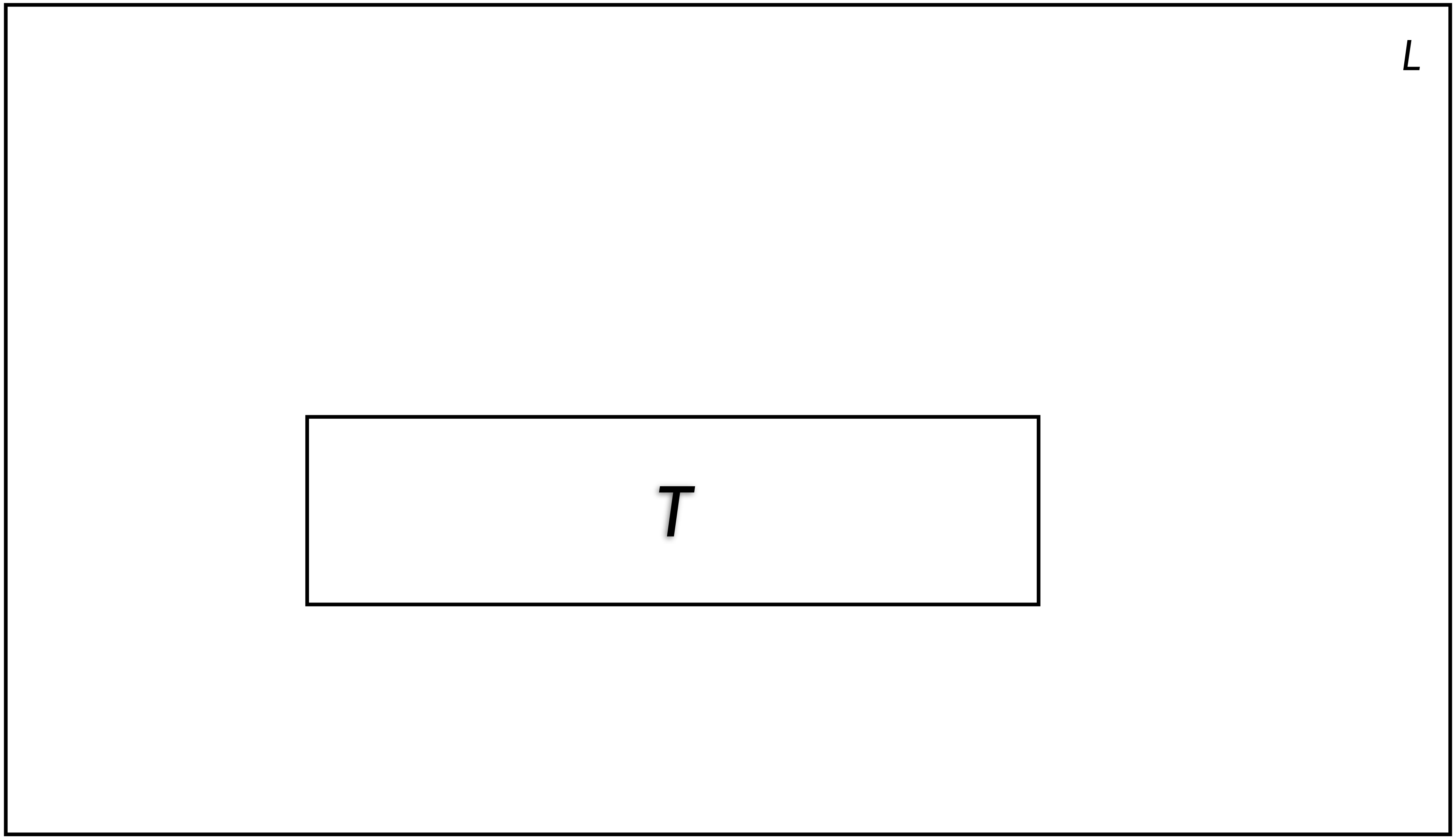
Algorithms - Reachability

- So the set of surely-winning locations for Player 1 are obtained as the limit of the following sequence:

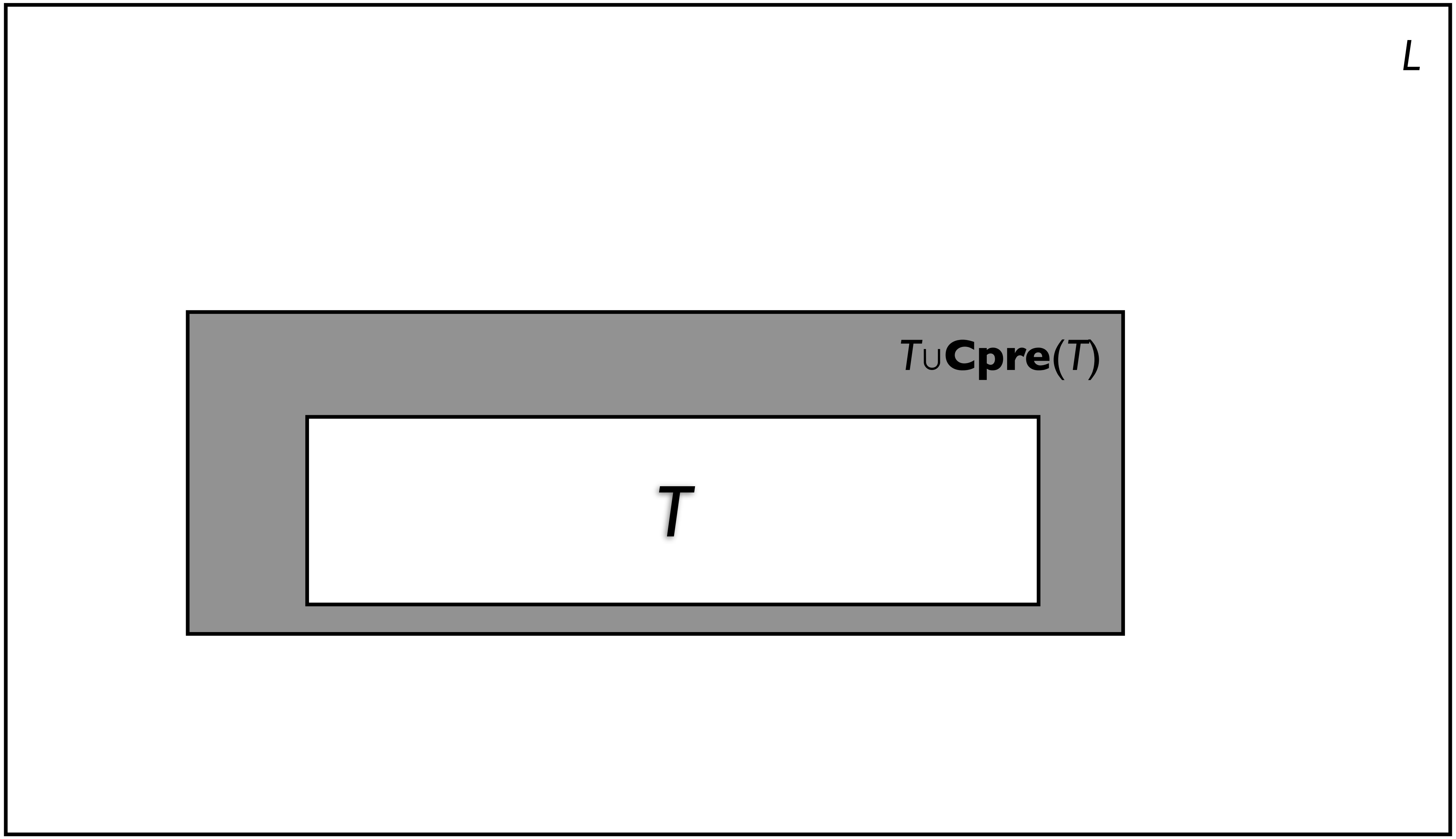
  $W^0 = T$;
  $W^{i+1} = T \cup \mathbf{Cpre}(W^i)$, for all $i \geq 0$.

  This sequence stabilizes after at most $|L|$ steps. The limit is the least solution of the equation $W = T \cup \mathbf{Cpre}(W)$.

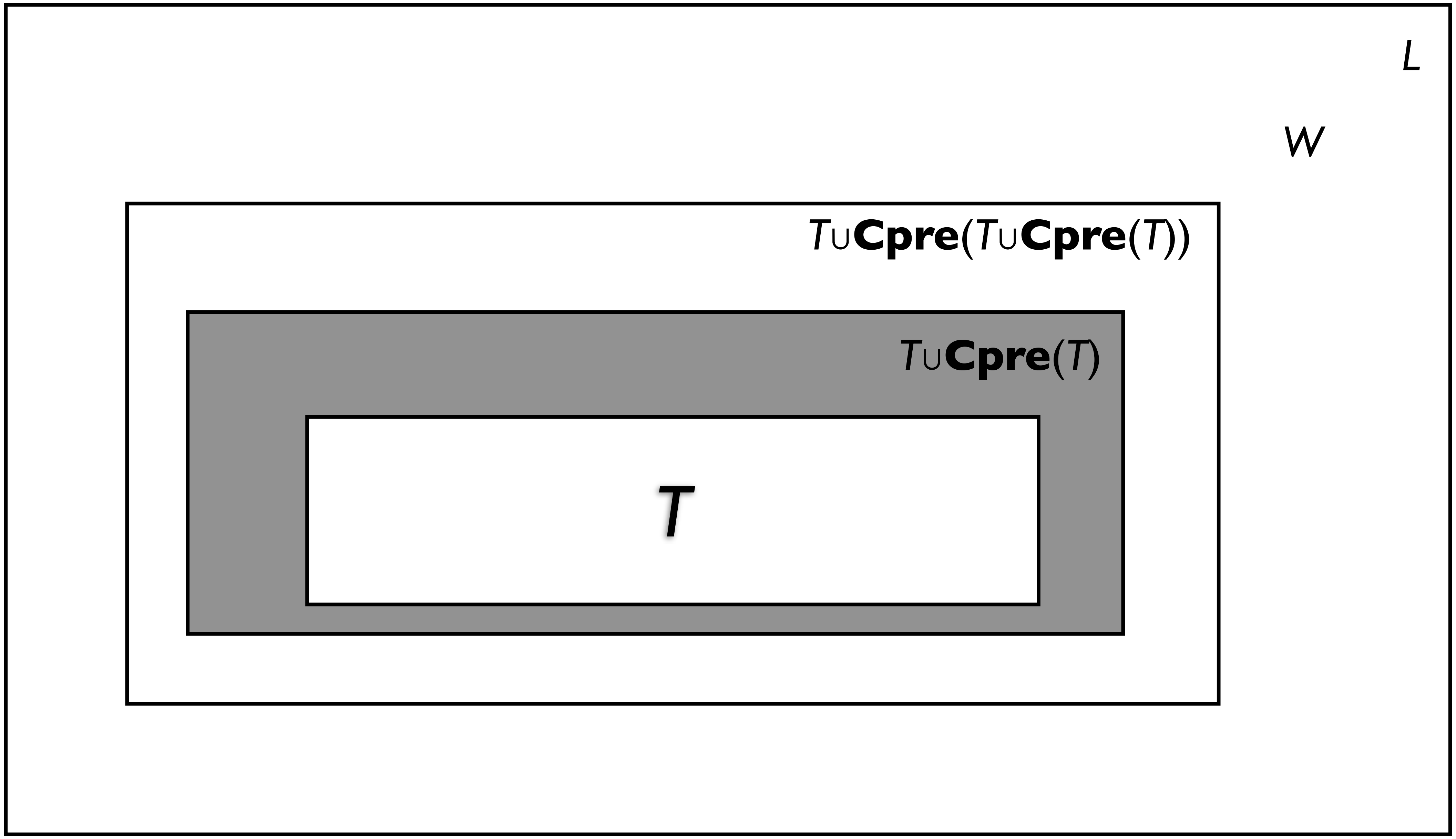
Algorithms - Reachability



Algorithms - Reachability



Algorithms - Reachability



L

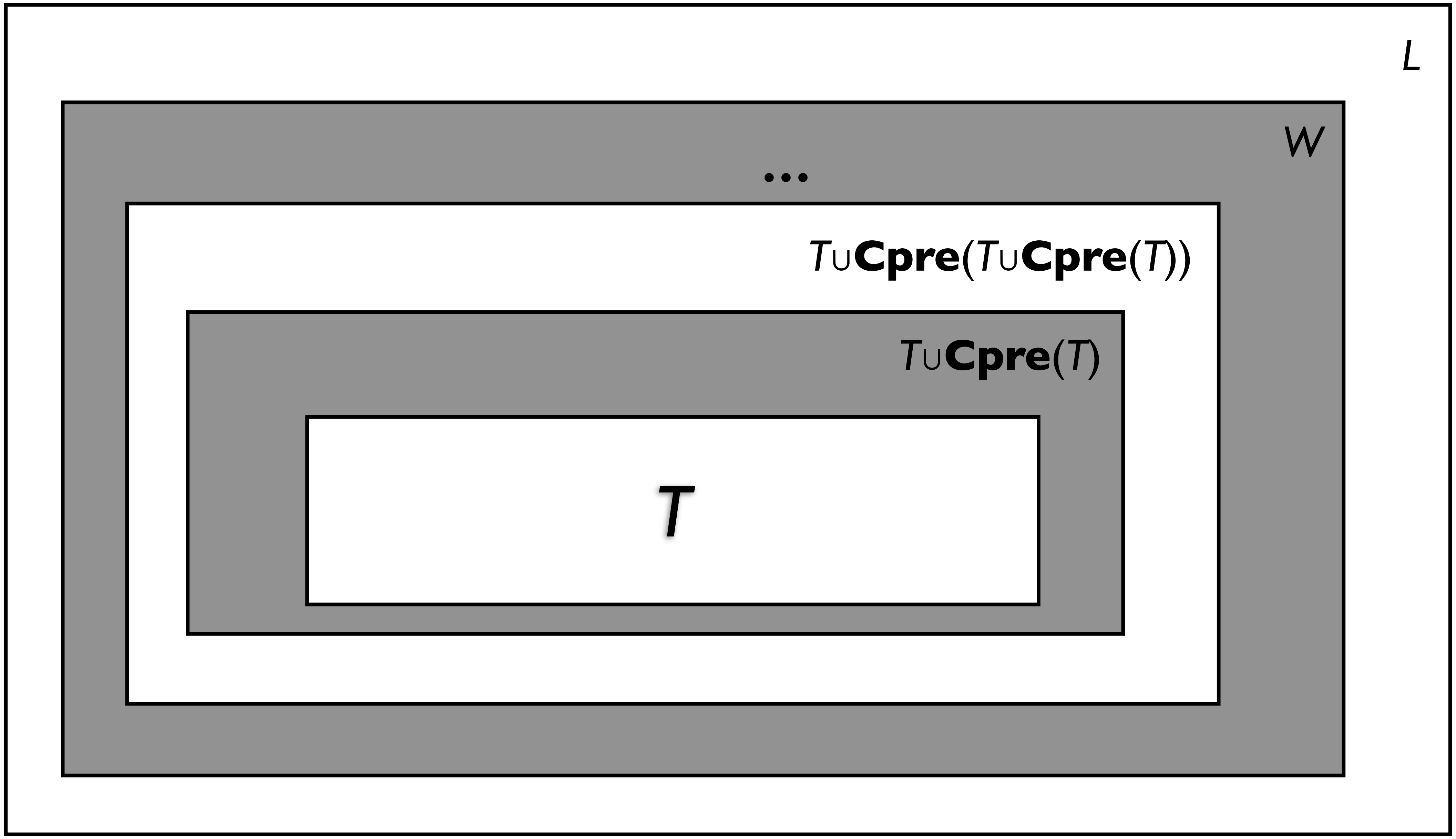
W

T∪**Cpre**(*T*∪**Cpre**(*T*))

T∪**Cpre**(*T*)

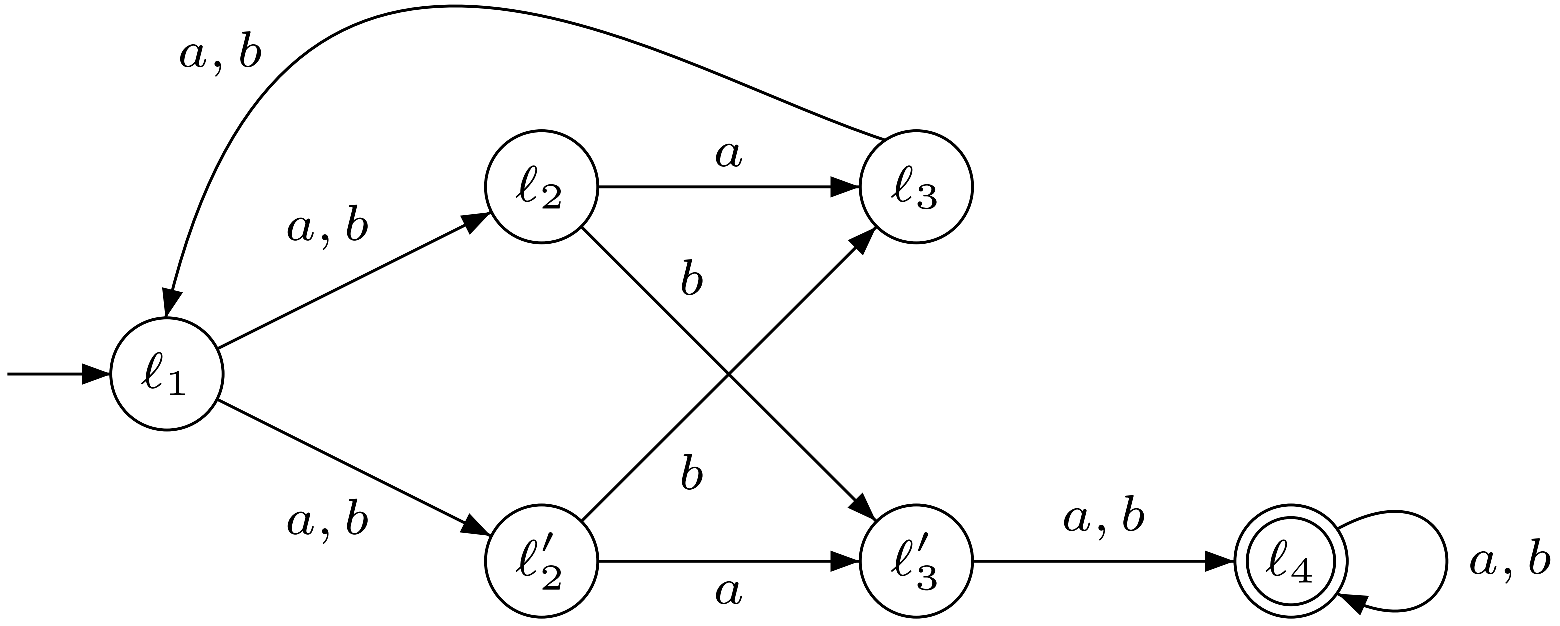
T

Algorithms - Reachability



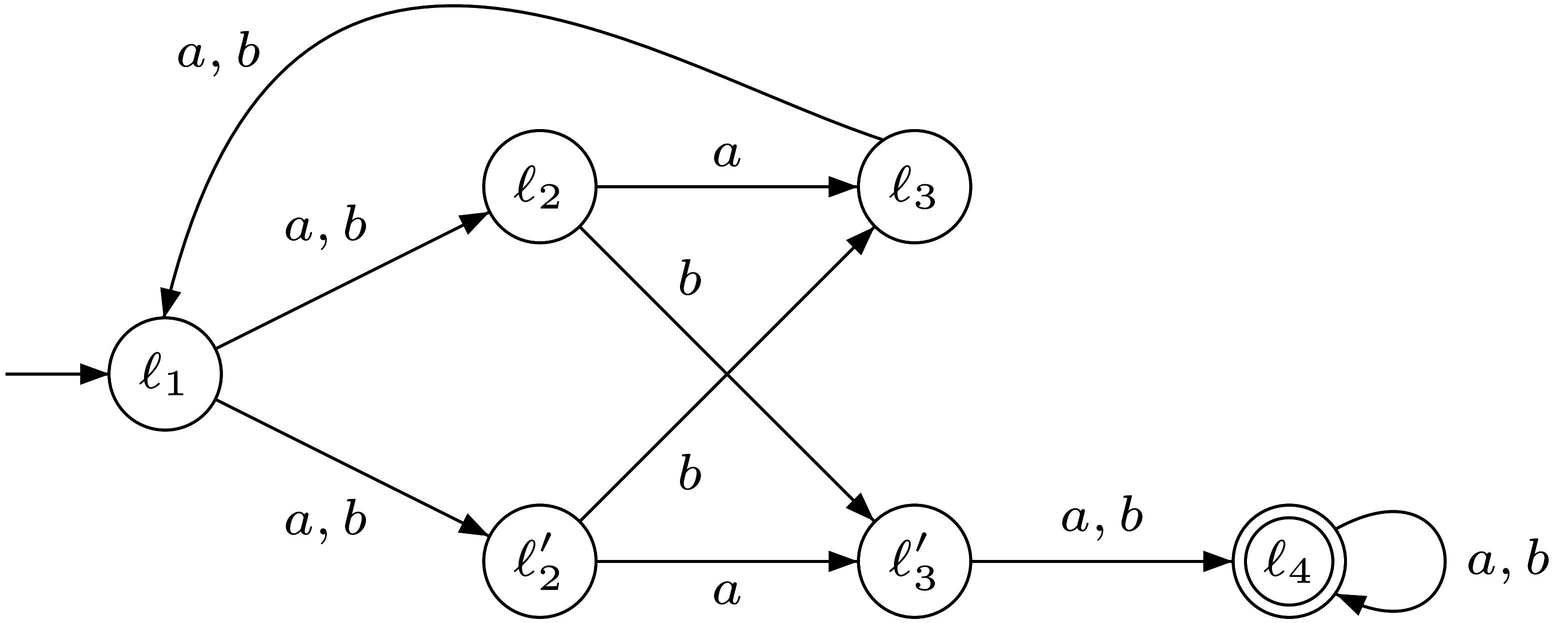
L
W
…
$T \cup \mathbf{Cpre}(T \cup \mathbf{Cpre}(T))$
$T \cup \mathbf{Cpre}(T)$
T

Algorithms - Reachability



Let us compute the surely winning locations for the objective **Reach**({*l*$_4$}).

Algorithms - Reachability



$W^0=\{l_4\}$

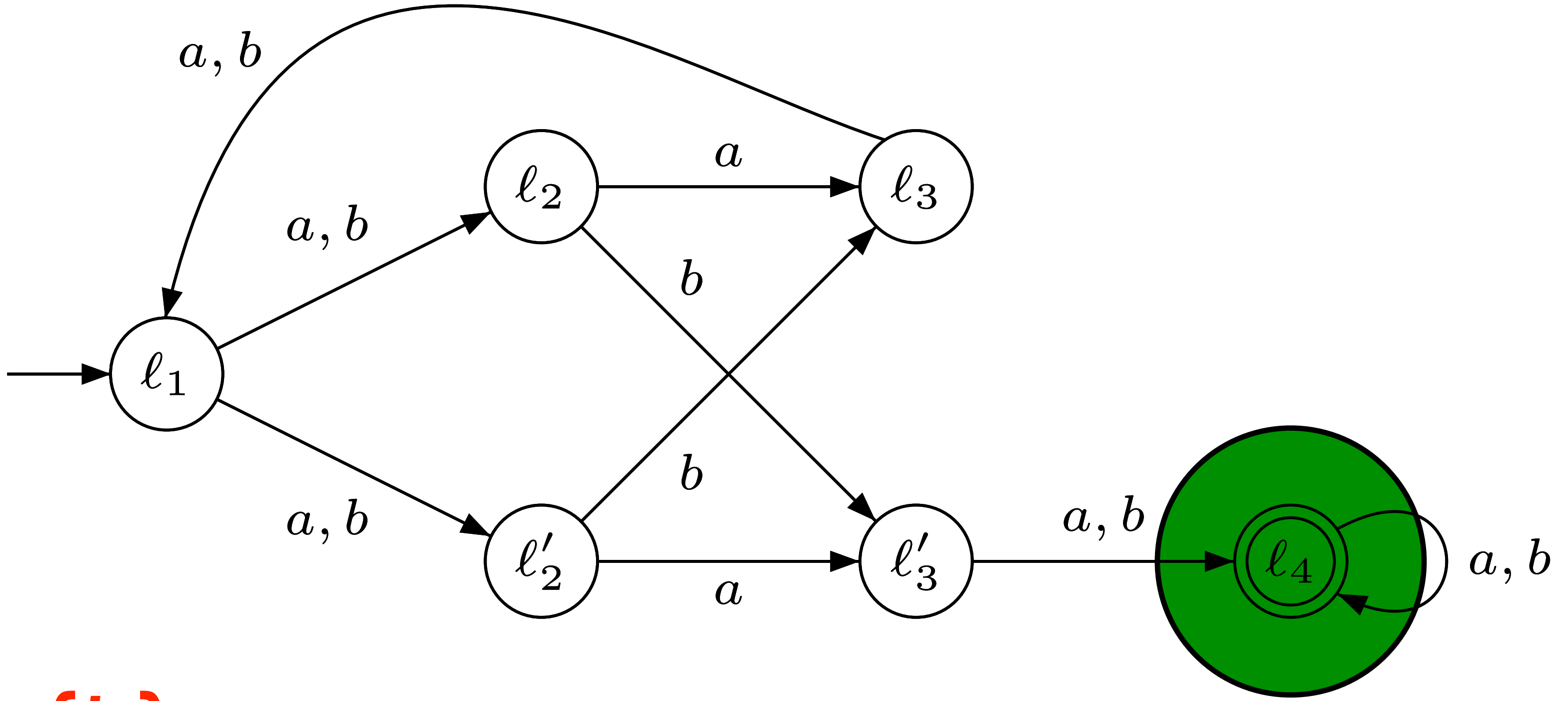
$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Reachability



$W^0=\{l_4\}$

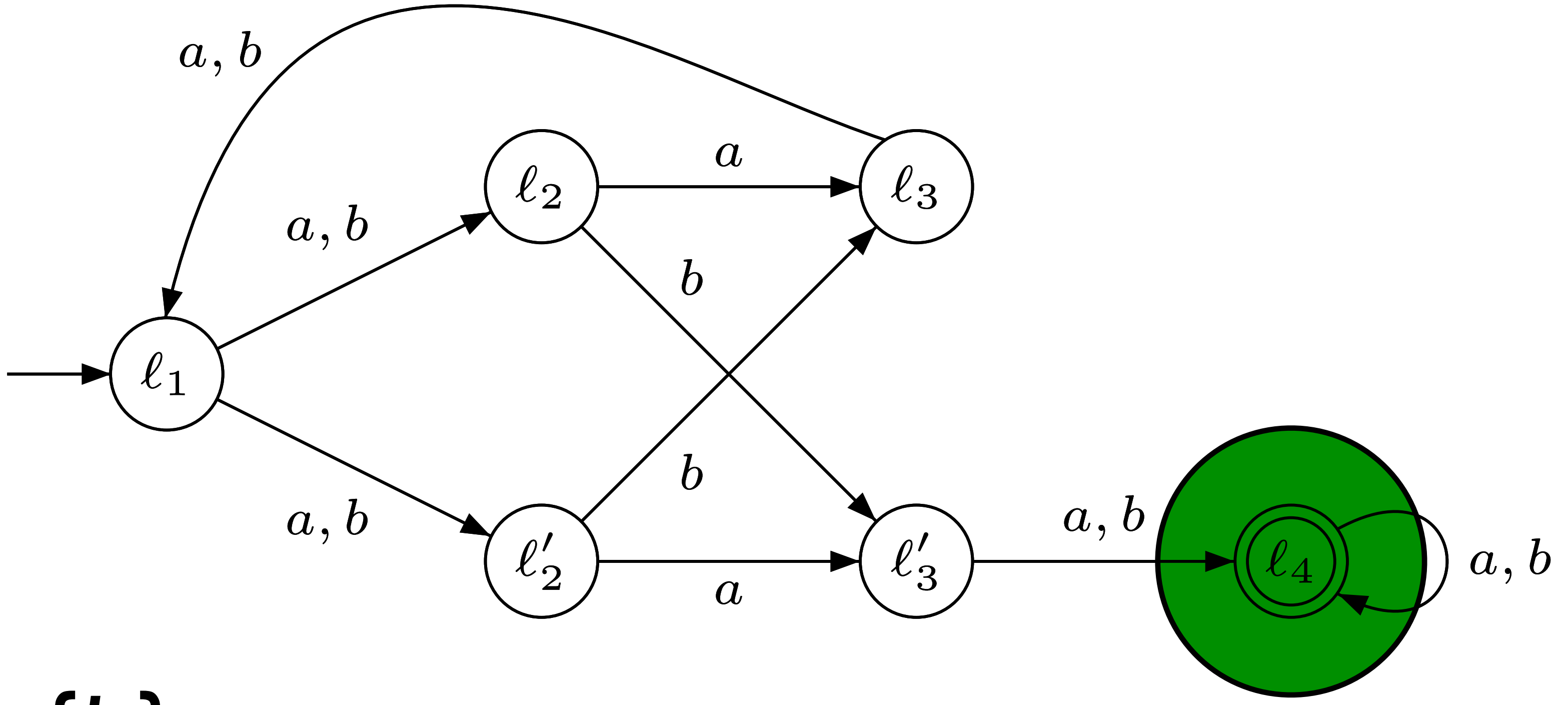
$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Reachability



$W^0=\{l_4\}$

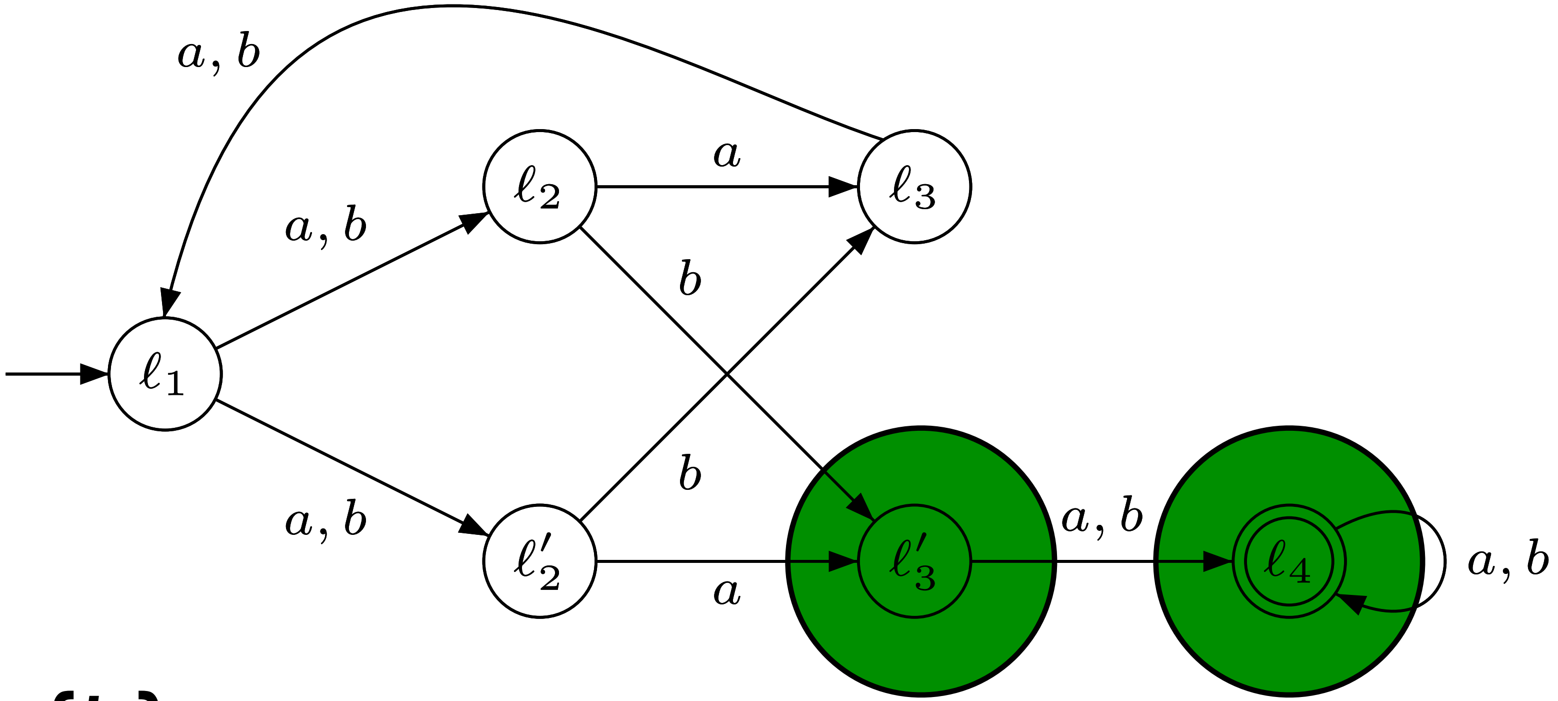
$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Reachability



$W^0 = \{l_4\}$

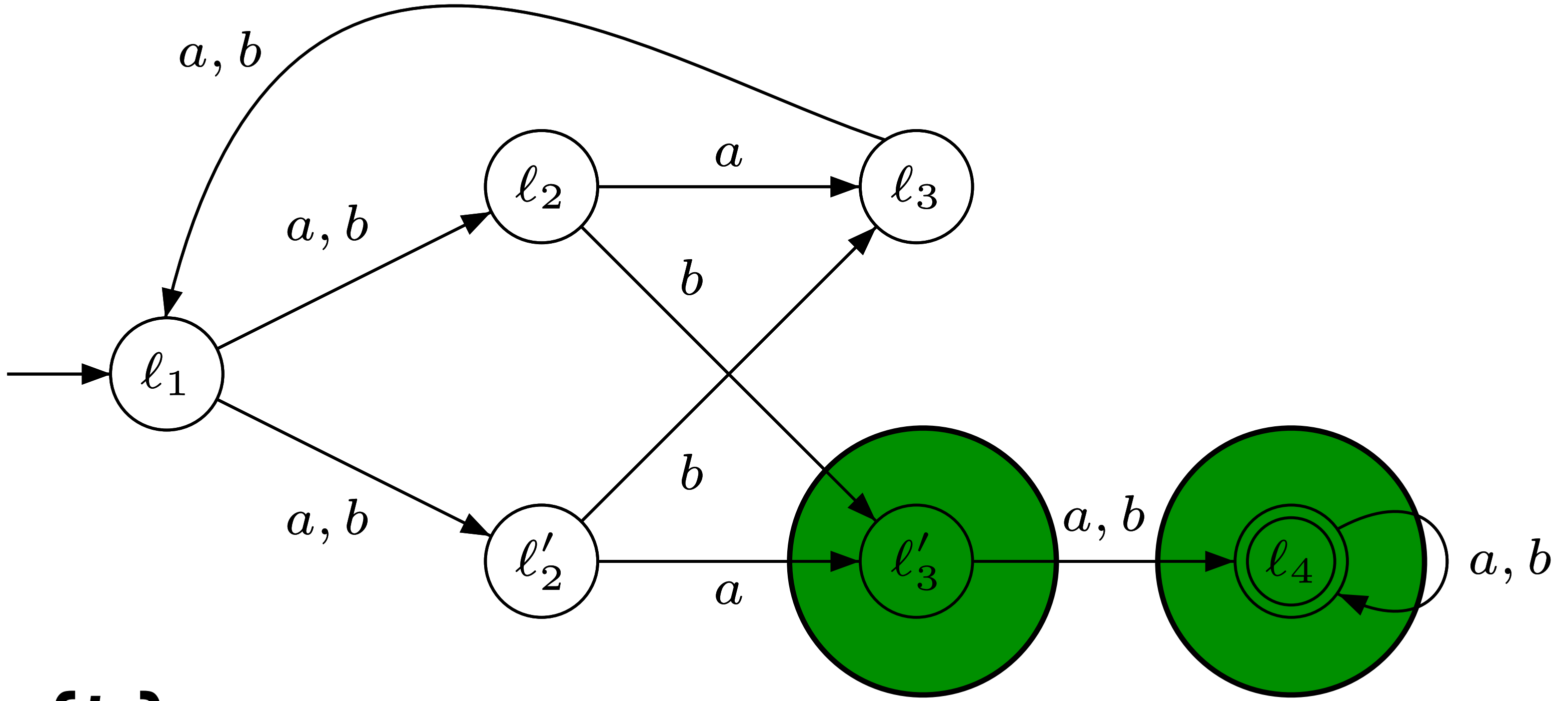
$W^1 = \{l_4\} \cup \mathbf{Cpre}(\{l_4\}) = \{l_3', l_4\}$

$W^2 = \{l_4\} \cup \mathbf{Cpre}(\{l_3', l_4\}) = \{l_2, l_2', l_3', l_4\}$

$W^3 = \{l_4\} \cup \mathbf{Cpre}(\{l_2, l_2', l_3', l_4\}) = \{l_1, l_2, l_2', l_3', l_4\}$

$W^4 = \{l_4\} \cup \mathbf{Cpre}(\{l_1, l_2, l_2', l_3', l_4\}) = L$

Algorithms - Reachability



$W^0 = \{l_4\}$

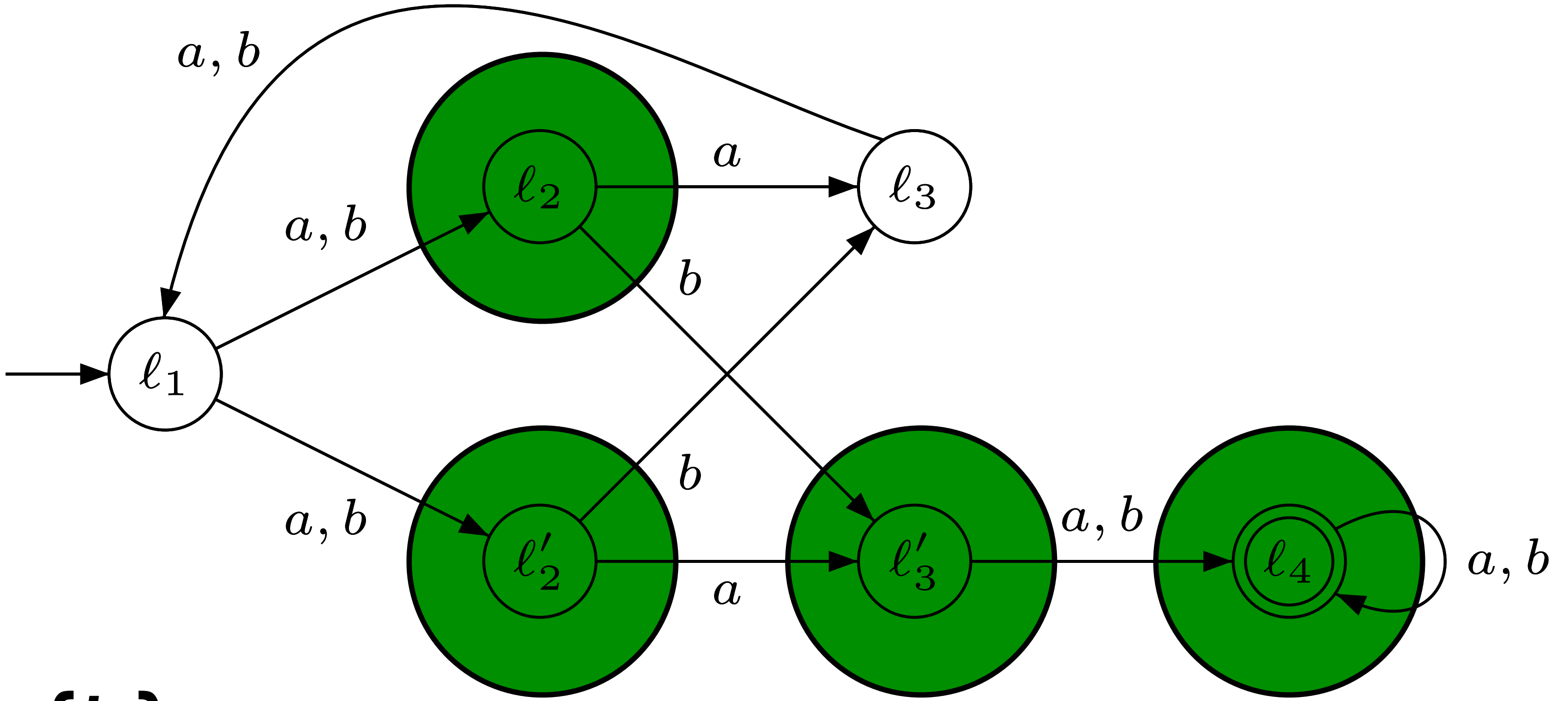
$W^1 = \{l_4\} \cup \mathbf{Cpre}(\{l_4\}) = \{l_3', l_4\}$

$W^2 = \{l_4\} \cup \mathbf{Cpre}(\{l_3', l_4\}) = \{l_2, l_2', l_3', l_4\}$

$W^3 = \{l_4\} \cup \mathbf{Cpre}(\{l_2, l_2', l_3', l_4\}) = \{l_1, l_2, l_2', l_3', l_4\}$

$W^4 = \{l_4\} \cup \mathbf{Cpre}(\{l_1, l_2, l_2', l_3', l_4\}) = L$

Algorithms - Reachability



$W^0=\{l_4\}$

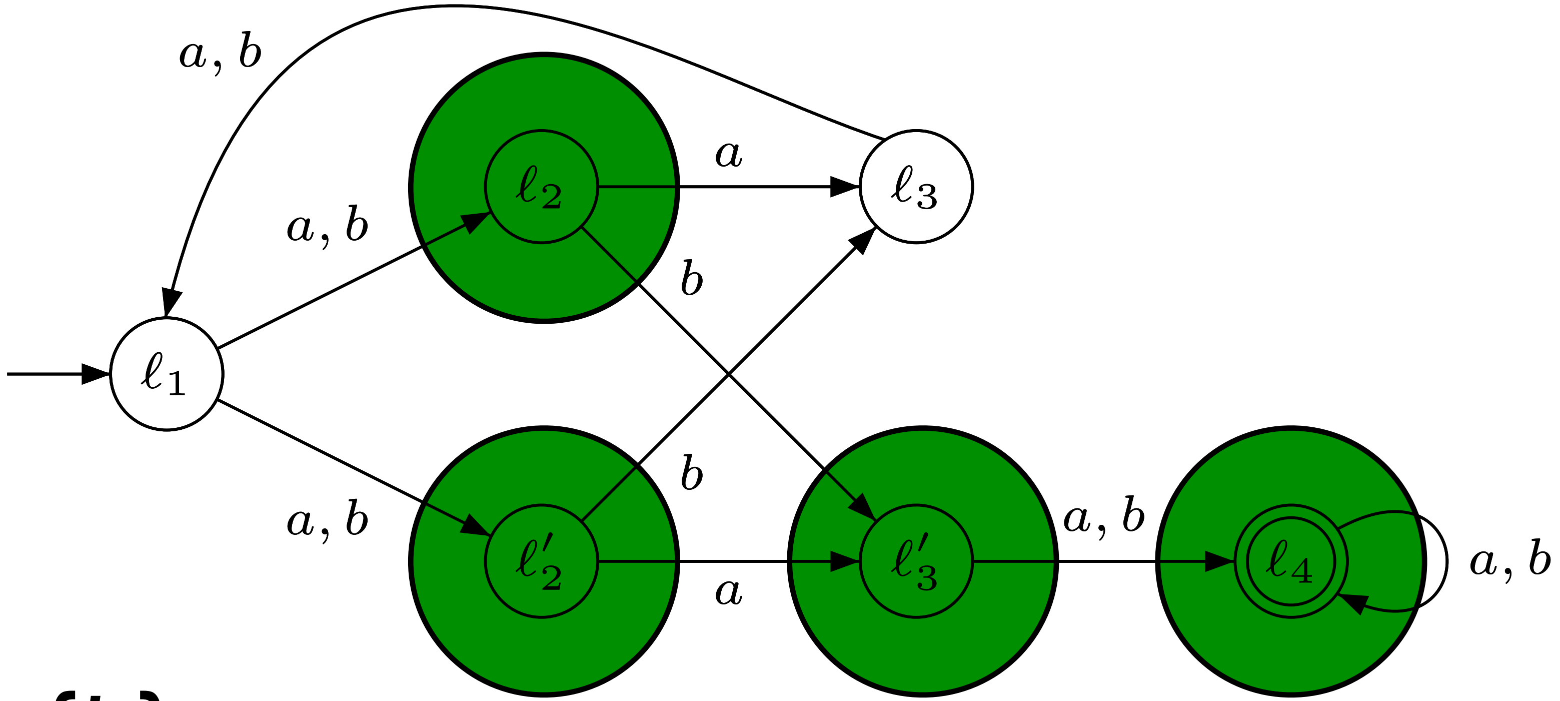
$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Reachability



$W^0=\{l_4\}$

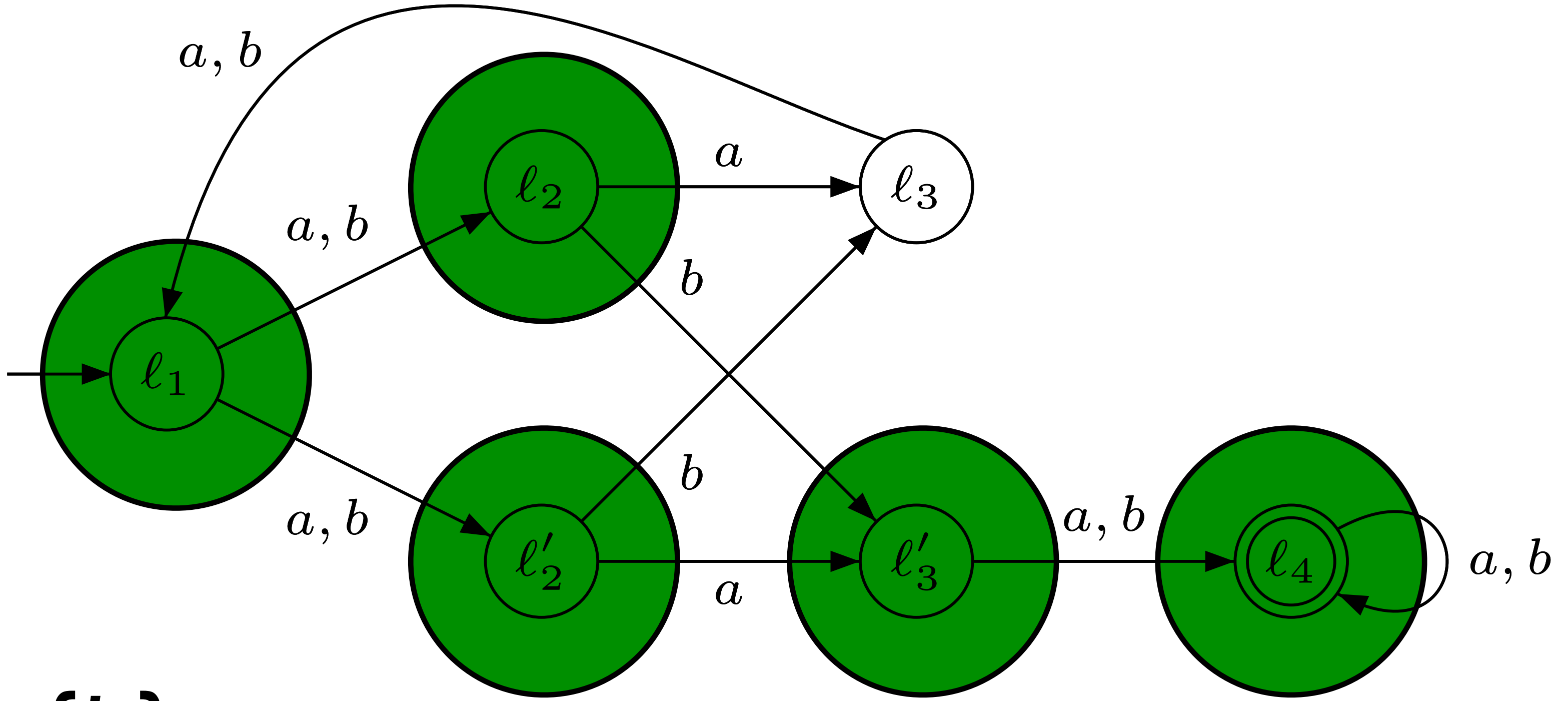
$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Reachability



$W^0 = \{l_4\}$

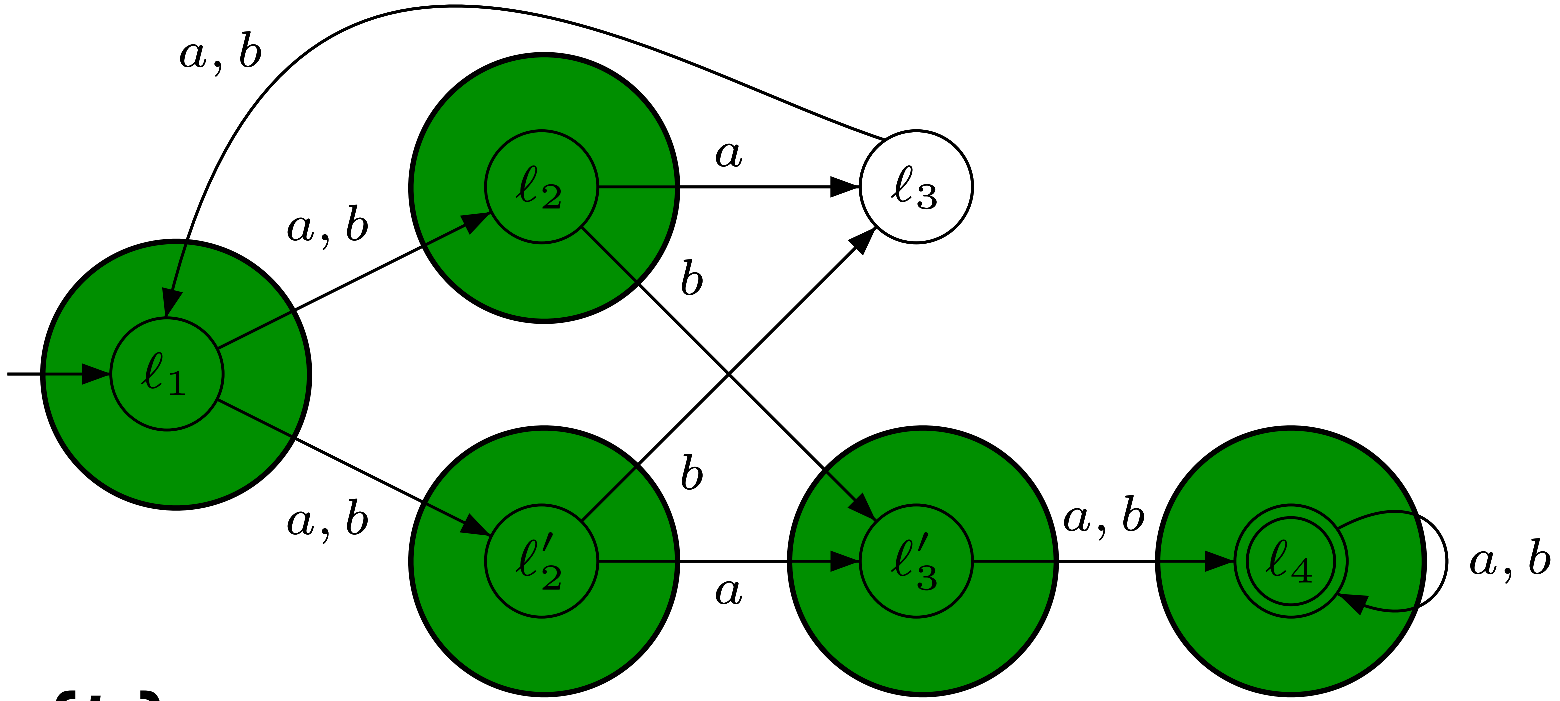
$W^1 = \{l_4\} \cup \mathbf{Cpre}(\{l_4\}) = \{l_3', l_4\}$

$W^2 = \{l_4\} \cup \mathbf{Cpre}(\{l_3', l_4\}) = \{l_2, l_2', l_3', l_4\}$

$W^3 = \{l_4\} \cup \mathbf{Cpre}(\{l_2, l_2', l_3', l_4\}) = \{l_1, l_2, l_2', l_3', l_4\}$

$W^4 = \{l_4\} \cup \mathbf{Cpre}(\{l_1, l_2, l_2', l_3', l_4\}) = L$

Algorithms - Reachability



$W^0=\{l_4\}$

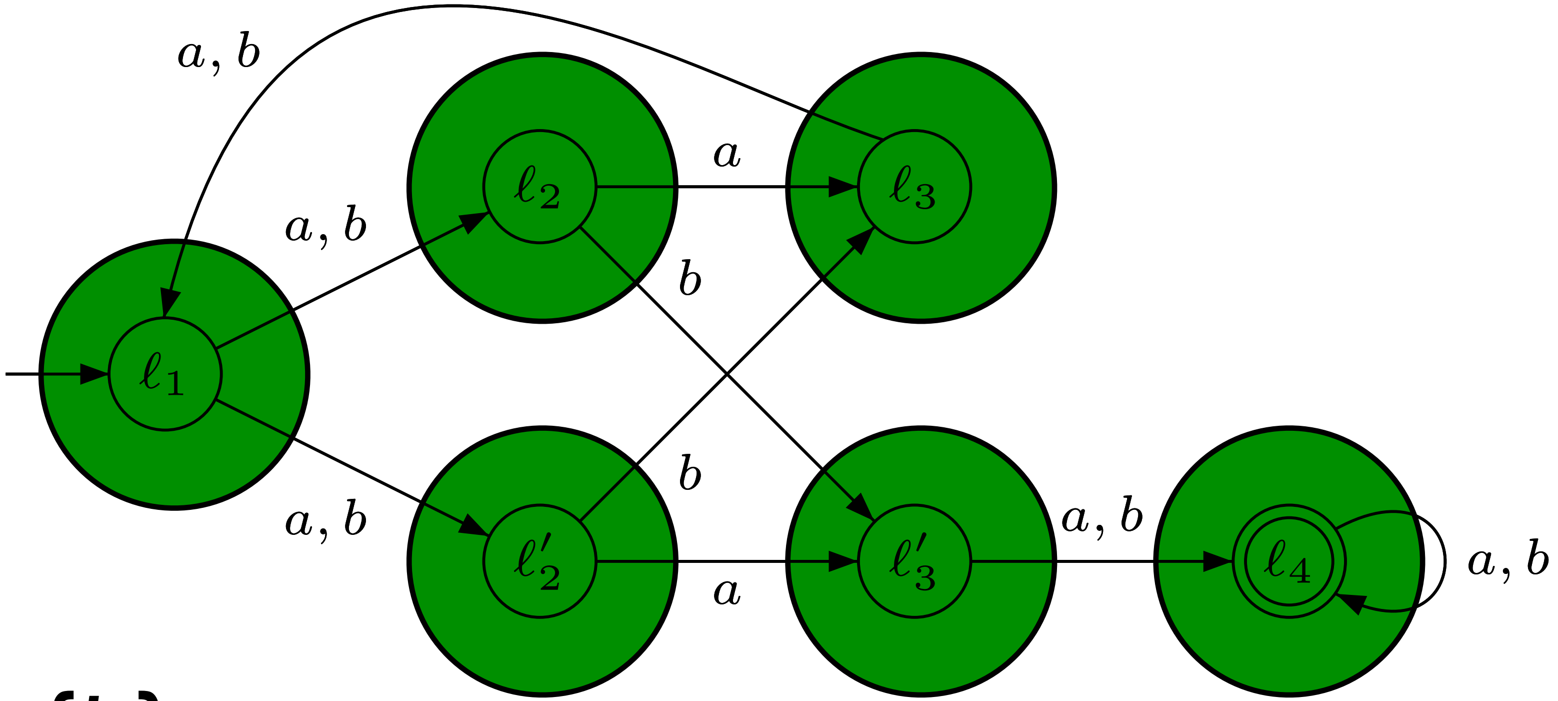
$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Reachability



$W^0=\{l_4\}$

$W^1=\{l_4\}\cup\mathbf{Cpre}(\{l_4\})=\{l_3',l_4\}$

$W^2=\{l_4\}\cup\mathbf{Cpre}(\{l_3',l_4\})= \{l_2,l_2',l_3',l_4\}$

$W^3=\{l_4\}\cup\mathbf{Cpre}(\{l_2,l_2',l_3',l_4\})=\{l_1,l_2,l_2',l_3',l_4\}$

$W^4=\{l_4\}\cup\mathbf{Cpre}(\{l_1,l_2,l_2',l_3',l_4\})=L$

Algorithms - Parity
Reduction to Safety

- We provide a simple reduction to **safety** games (similar to [BJW02] but simpler, see also [BD08] - FSTTCS08 paper).

- Let $G=(L,l_{init},\Sigma,\Delta)$ and $pr:L\rightarrow\{0,...,d\}$ defining the objective $\varphi=$**Parity**(pr). We extend G as follows:

  - We associate to each **odd priority** p a counter $c(p)$ which takes values in the set $\{0,...,n_p\}\cup\{\infty\}$, n_p being the number of locations with priority p in G.

  - Initially, all counters have value 0. The counter $c(p)$ is **incremented** when a location l with priority p is visited, and it is **reset** when a location l with an **even** priority $p'<p$ is visited.

- *Remark*. As Büchi and co-Büchi are special cases of parity, they can be handled by this reduction too.

Algorithms - Parity
Reduction to Safety

- Notation: $[n]$ denotes the set $\{0,1,2,\ldots,n\} \cup \{\infty\}$.

  For $v \in [n]$, $v \oplus 1 = \infty$ if $v \in \{n,\infty\}$, and $v \oplus 1 = v+1$ otherwise.

- Let us consider $G=(L,l_{init},\Sigma,\Delta)$ and $pr : L \rightarrow \{1,\ldots,d\}$ defining the parity objective $\varphi=$**Parity**(pr). We construct the game **PS**$(G)=(L',l_i',\Sigma,\Delta')$ where:

  - $L'=L\times[n_1]\times[n_3]\times\ldots\times[n_d]$

  - $l_{init}'=(l_{init},0,0,\ldots,0)$

  - $\Delta'=\{\ ((q,c),\sigma,(q',\mathbf{update}(c,p)))\ |\ (q,\sigma,q') \text{ and } p=pr(q')\ \}$

  where **update**$((c_1,c_3,\ldots,c_d),p)=$ $\begin{cases} (c_1,\ldots,c_{p-1},0,\ldots,0) & \text{if } p \text{ is even} \\ (c_1,\ldots,c_{p-1},c_p\oplus 1,c_{p+1},\ldots,c_d) & \text{if } p \text{ is odd} \end{cases}$

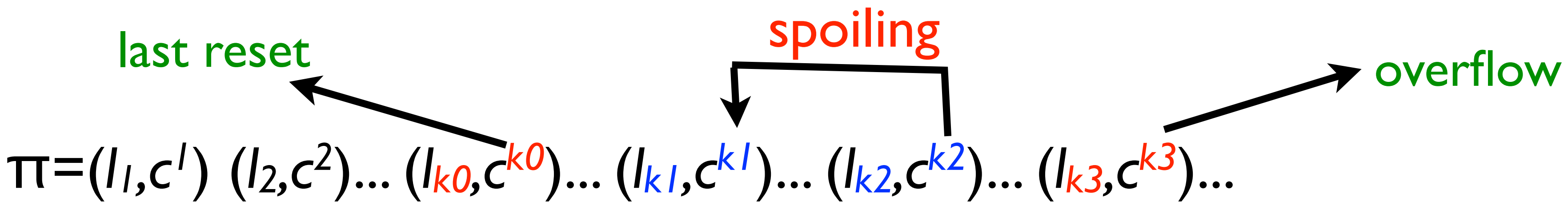
  and **PS**$(\varphi)=$**Safe**(T) where $T=L' \cap (\ L\times\{0,1,2,\ldots,n\}^{\lceil d/2 \rceil})$

  i.e. **no counter overflow**.

Algorithms - Parity Reduction to Safety

Theorem. Player I has a surely-winning strategy in the parity game (G,φ) iff Player I has a surely-winning strategy in the safety game $(\mathbf{PS}(G),\mathbf{PS}(\varphi))$.

Proof. Assume that Player I has a surely-winning strategy in (G,φ), then Player I has a memoryless surely-winning strategy α. Consider the Player I memoryless strategy α' in $(\mathbf{PS}(G),\mathbf{PS}(\varphi))$ which plays in $(l,c_1,\ldots,c_d)$ the action $\alpha(l)$. We claim that α' is surely-winning. Assume it is not the case, then as α is memoryless there exists $\pi\in\mathbf{Outcome_I}(\mathbf{PS}(G),\alpha')$ such that:



$$\pi=(l_1,c^1)\ (l_2,c^2)\ldots\ (l_{k_0},c^{k_0})\ldots\ (l_{k_1},c^{k_1})\ldots\ (l_{k_2},c^{k_2})\ldots\ (l_{k_3},c^{k_3})\ldots$$

and $c^{k_3}(p)=\infty$ for some odd priority p which was last reset in position k_0. Between position k_0 and position k_3, n_{p+1} locations with priority p has been visited without ever visiting any location with an even priority less than p. As n_p is the number of locations with priority p, there are two positions k_1 and k_2 are such that $l_{k_1}=l_{k_2}$ and its priority is p. Then clearly Player 2 has a spoiling strategy in (G,φ), a contradiction.

The other direction is established similarly after using the determinacy theorem.

Algorithms - Parity Reduction to Safety

- Notation: $[n]$ denotes the set $\{0,1,2,\ldots,n\}\cup\{\infty\}$.

  For $v\in[n]$, $v\oplus 1=\infty$ if $v\in\{n,\infty\}$, and $v\oplus 1=v+1$ otherwise.

- Let us consider $G=(L,l_i,\Sigma,\Delta)$ and $pr : L \rightarrow \{1$ rity objective $\varphi=$**Parity**(pr). We construct t :

  - $L'=L$
  - $l_i'=($
  - $\Delta'=\{$

  where $,\ldots,0)$ if p is even

  $(c_1,\ldots,c_{p-1},c_p\oplus 1,c_{p+1},\ldots,c_d)$ if p is odd

  and **PS**(φ)=**Safe**(T) where $T=L'\cap(\,L\times\{0,1,2,\ldots,n\}^{\lceil d/2\rceil})$

  i.e. no counter overflow.

Note that the size of the safety game is bounded by **O**$(\,n\,(n/d)^{\lceil d/2\rceil})$

Games with perfect information
Summary

- **Simple games**
  Player 1 chooses actions,
  Player 2 resolves nondeterminism
- **Rich objectives**
  Safety, reachability, Büchi, co-Büchi, and parity.
- **Simple algorithms**
  Simple fixed points for safety and reachability.
  Büchi, co-Büchi and parity can be easily and elegantly reduced to safety.

Games of imperfect information

Surely-winning

Imperfect information - Motivations

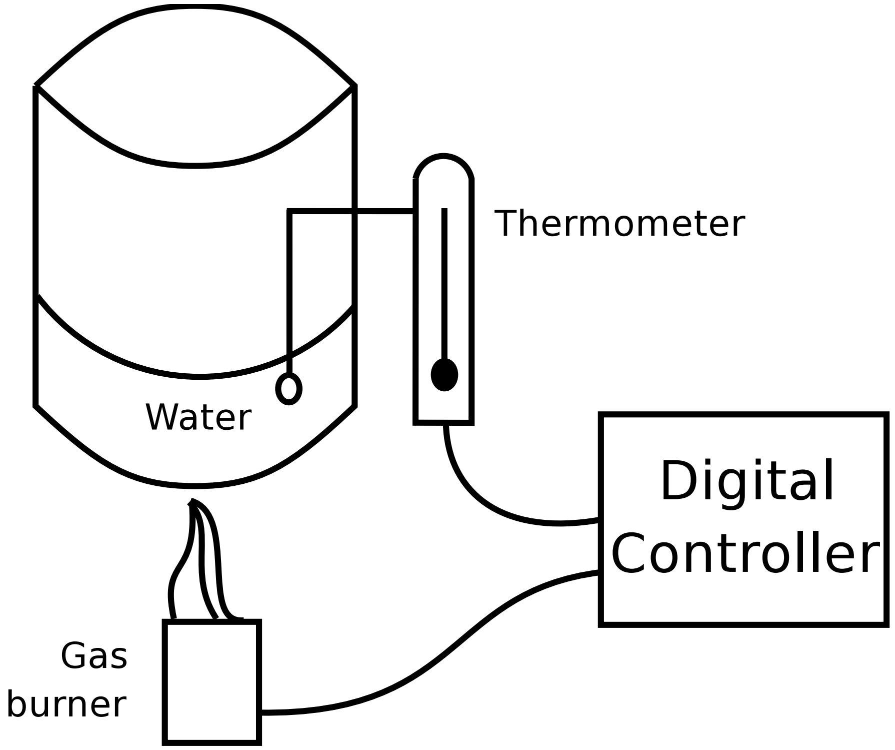
- Games structure with perfect information makes the **strong assumption** that the players can observe the state of the game and the previous moves before playing.

- This is often **unrealistic** in the design of reactive systems because components have an internal state that is not visible to other components (e.g. local variables).

- Also, sometimes we need to consider that components choose their moves simultaneously and independently of the others (concurrent games, not considered here, see works by de Alfaro, Kuferman, Henzinger, etc).

➡ **We need models with imperfect information.**

Imperfect information - Motivations

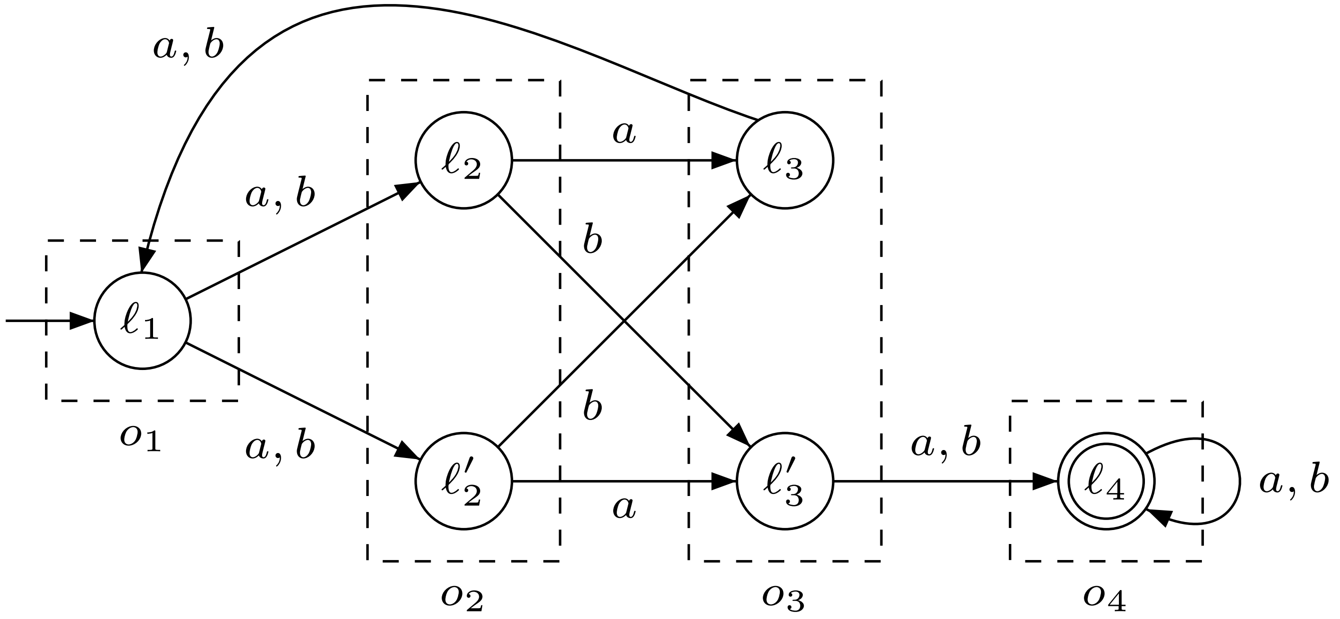


Typical hybrid system

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

Finite precision = imperfect information

Games Structure of Imperfect information

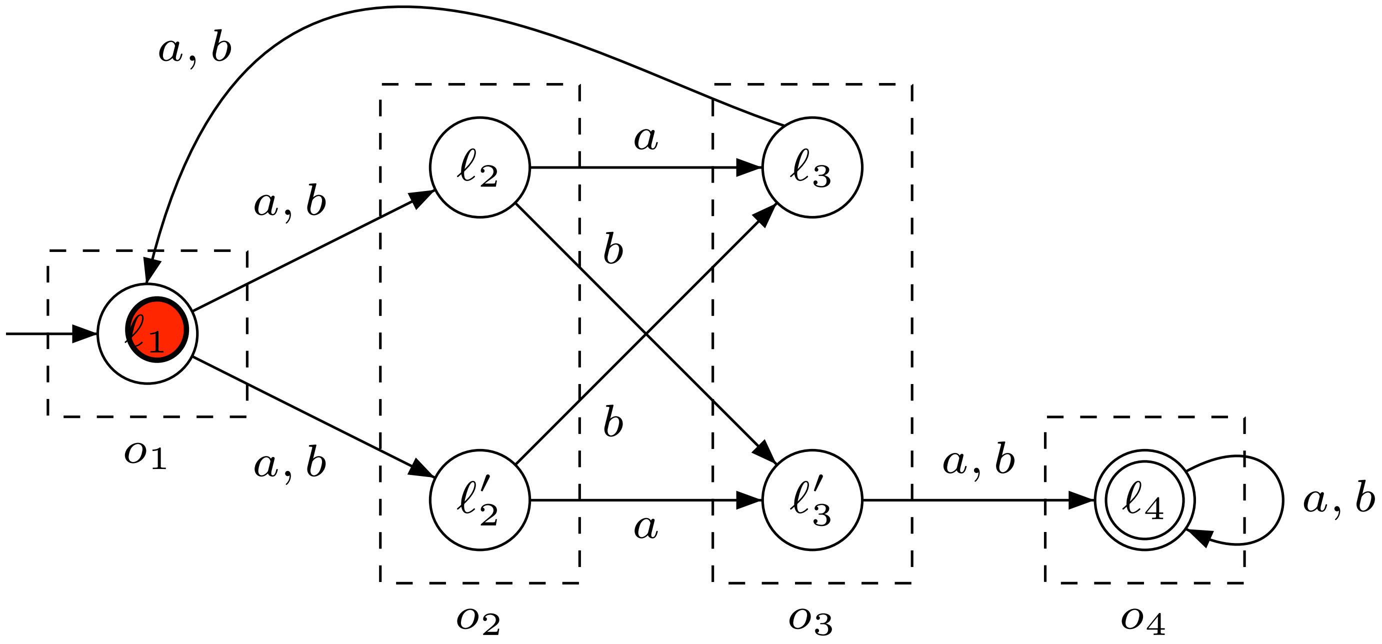


- A **game structure of imperfect information** is a tuple *G*=(*L*,*l*$_i$,Σ,Δ,*Obs*) where *G* is a game graph as before and *Obs*={*o*$_1$,*o*$_2$,...,*o*$_n$} is a partition of *L* called the **set of observations**.

Observation of a play

- Given $l \in L$, we note ***obs(l)***
  the **observation** $o \in Obs$ such that $l \in o$.
- The **observation of a play** $\pi = l_0 l_1 \ldots l_n \ldots$
  is the sequence $obs(\pi) = obs(l_0) obs(l_1) \ldots obs(l_n) \ldots$
- When playing, **only** the observation of the current location is revealed to Player 1.

Observation of a play

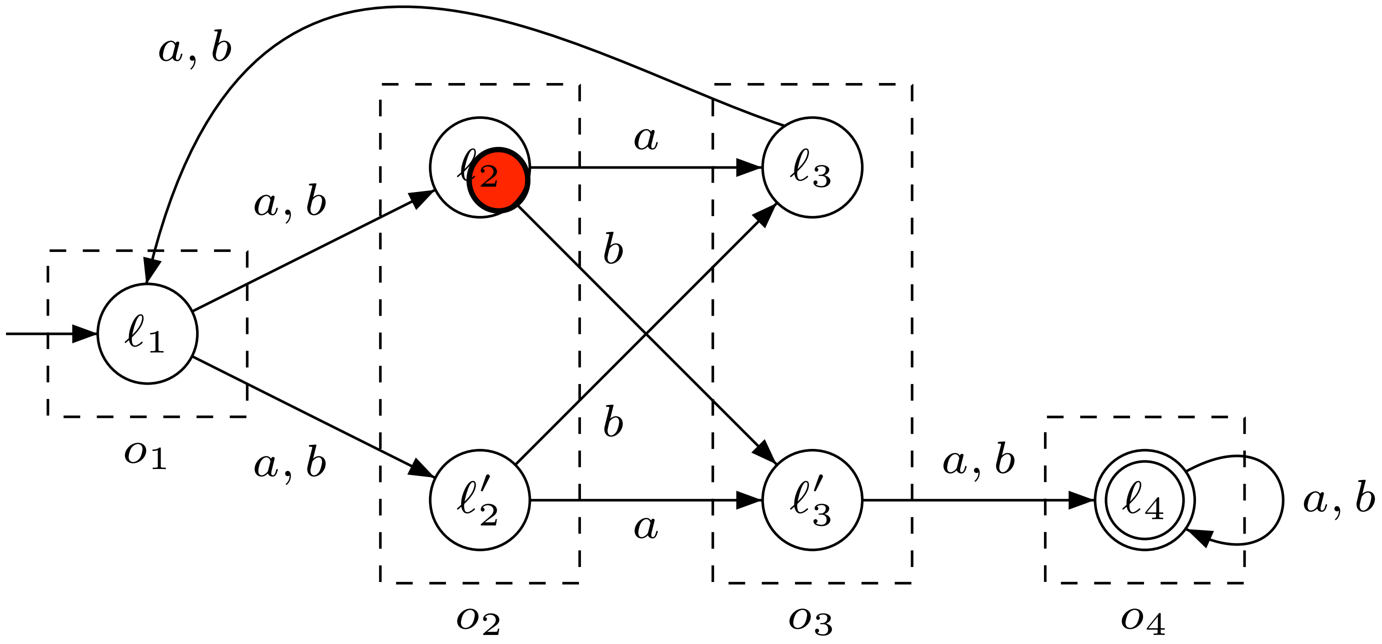


$\pi = l_1\ l_2\ l_3'\ (l_4)^\omega$

$obs(\pi) = obs(l_1)\ obs(l_2)\ obs(l_3')\ obs(l_4)^\omega$

$= o_1\ o_2\ o_3\ (o_4)^\omega$

Observation of a play

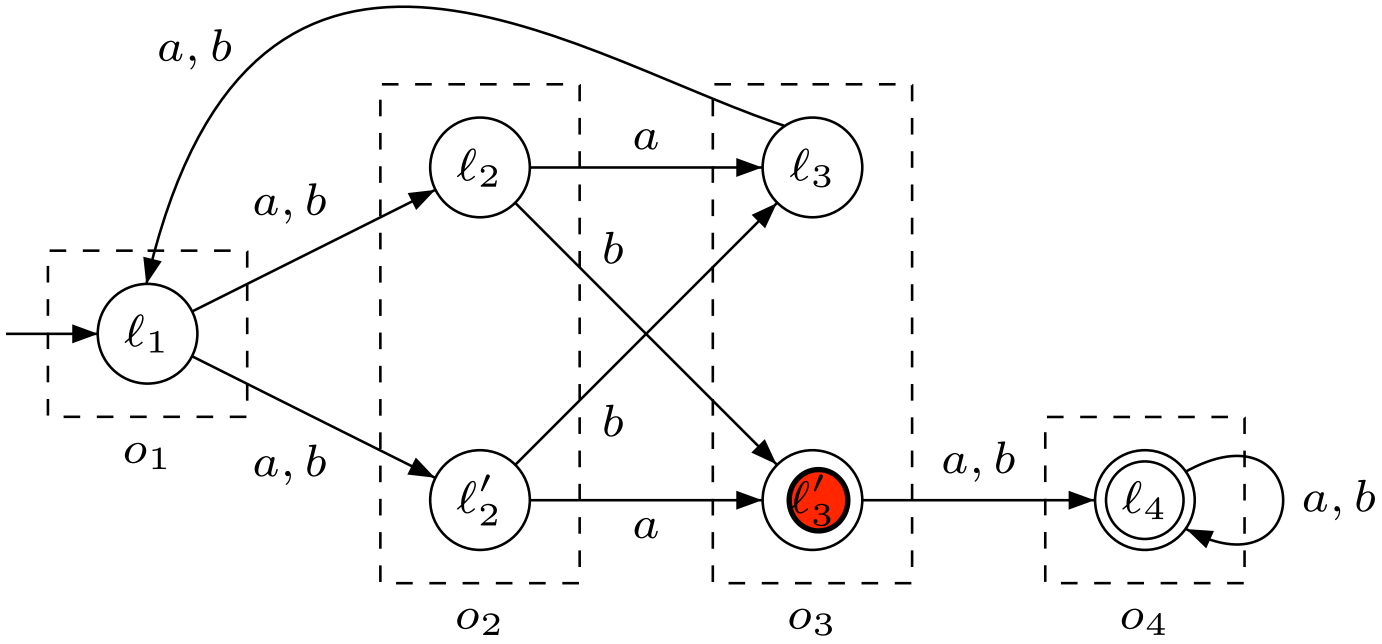


$\pi = l_1\ l_2\ l_3'\ (l_4)^{\omega}$

$obs(\pi) = obs(l_1)\ obs(l_2)\ obs(l_3')\ obs(l_4)^{\omega}$

$= o_1 \quad o_2 \quad o_3 \quad (o_4)^{\omega}$

Observation of a play

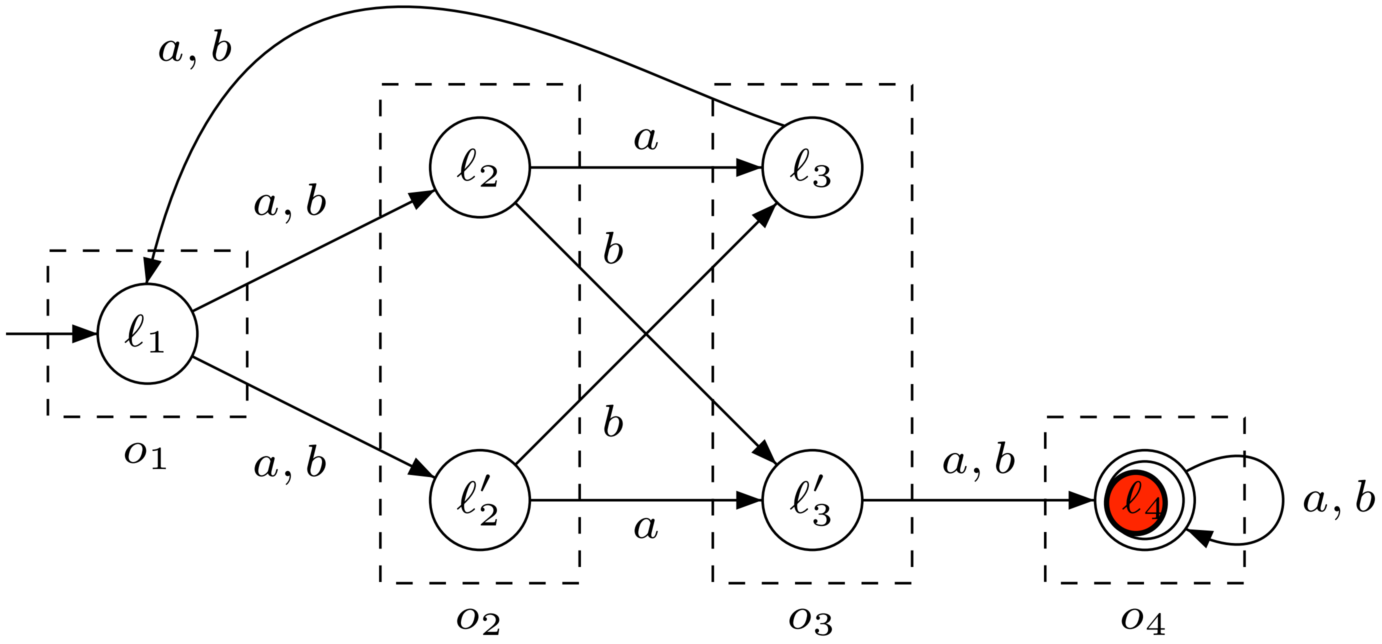


$\pi = l_1\ l_2\ l_3'\ (l_4)^\omega$

$obs(\pi) = obs(l_1)\ obs(l_2)\ obs(l_3')\ obs(l_4)^\omega$

$= o_1 \quad o_2 \quad o_3 \quad (o_4)^\omega$

Observation of a play



$\pi = l_1\ l_2\ l_3'\ (l_4)^\omega$

$obs(\pi) = obs(l_1)\ obs(l_2)\ obs(l_3')\ obs(l_4)^\omega$

$= o_1 \quad o_2 \quad o_3 \quad (o_4)^\omega$

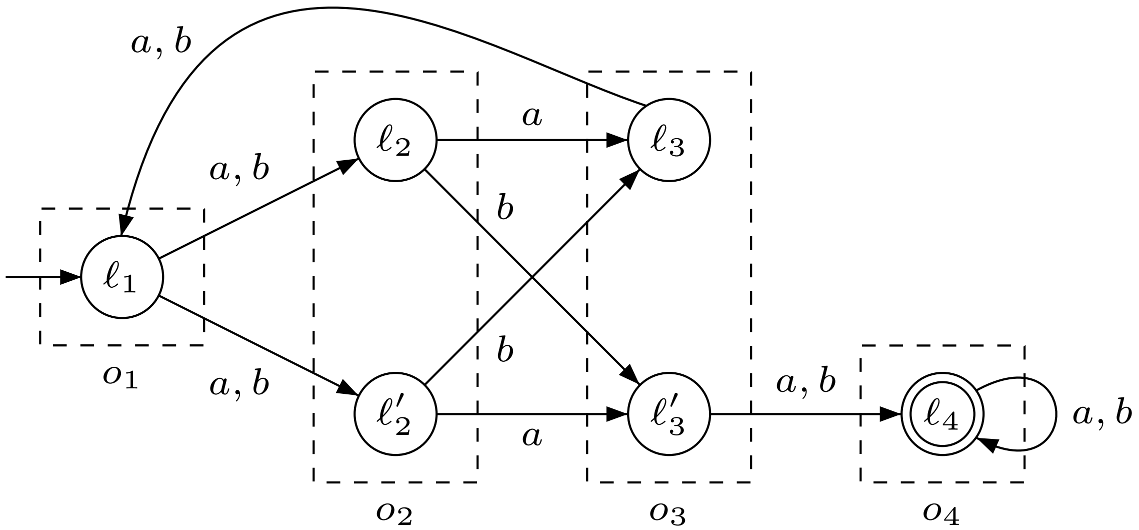
Observation-based strategies

- An **observation-based strategy** for Player 1:

$$\alpha : L^{+} \to \Sigma \text{ such that}$$

$$\forall \rho, \rho' \cdot obs(\rho) = obs(\rho') \implies \alpha(\rho) = \alpha(\rho')$$

- **Example.** Let $\rho = l_1 l_2$ and $\rho' = l_1 l_2'$.
  If α is observation-based, and if $\alpha(\rho) = \sigma$,
  then $\alpha(\rho') = \sigma$ because $obs(\rho) = o_1 o_2 = obs(\rho')$.



Observable objectives

- An objective in a game of imperfect information is a set of plays φ as before but we require that φ is **observable** for Player 1, that is:

$$\forall \pi \in \varphi \cdot \forall \pi' \cdot obs(\pi)=obs(\pi') \implies \pi' \in \varphi.$$

- Clearly, observable objectives can be defined as subsets of Obs^{ω}.
- In the sequel, we assume that:
  - reachability and safety objectives are defined by unions of target observations.
  - parity objectives, we assume that they are defined as functions $pr{:}Obs \rightarrow \{0,...,d\}$.

    This ensures that those objectives are observable.

Surely-winning observation based strategies

- A (deterministic) observation-based strategy

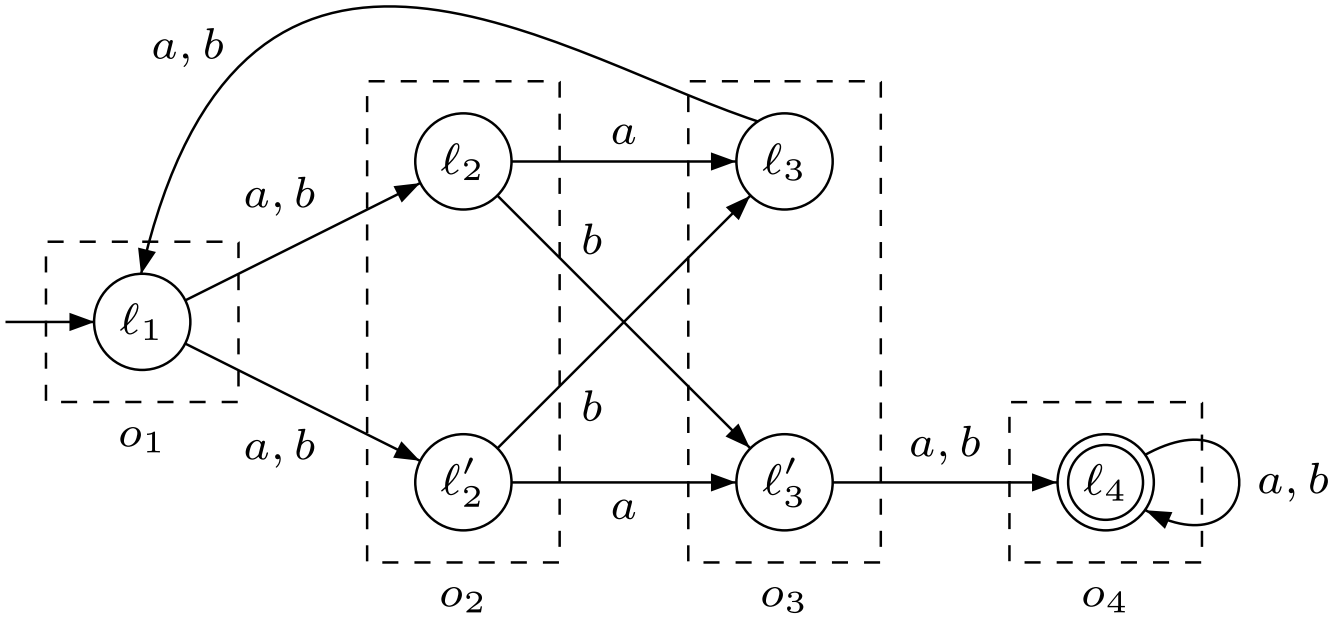
  $$\alpha : L^{+} \rightarrow \Sigma^{\omega}$$

  is **surely-winning** for an objective $\varphi \in Obs^{\omega}$ in G if

  $$obs(\mathbf{Outcome_I}(G,\alpha)) \subseteq \varphi$$

- Note that games with perfect information are clearly a special case, take $Obs=\{\{l\} \mid l \in L\}$.

Game with imperfect information: an example



φ=**Reach**({o_4})

Can Player 1 surely-win with an observation-based strategy ?

NO

Let α be an arbitrary observation-based strategy. Consider the strategy β for Player 2:

for all ρ•*l*∈L^+ and **Last**(*obs*(ρ·*l*))=o_2,

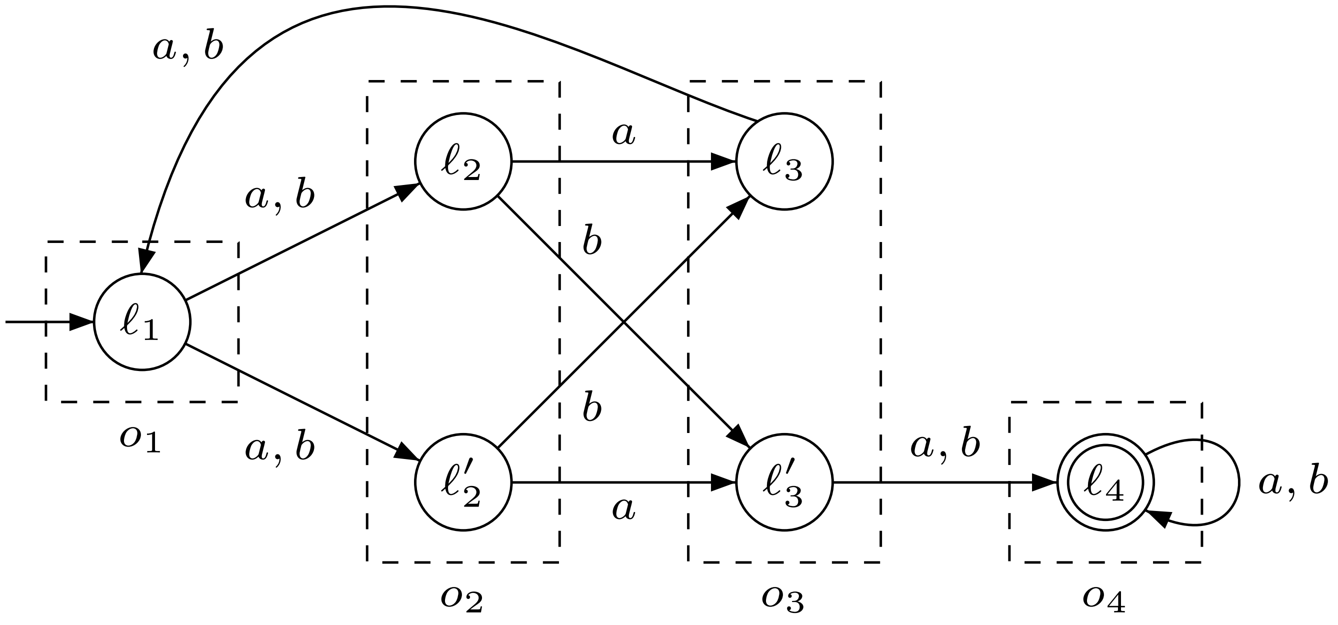
-if α(*obs*(ρ·*l*))=*a* then β(ρ,•)=l_2, and

-if α(*obs*(ρ·*l*))=*b* then β(ρ,•)=l_2'.

as α is fixed, this is possible to choose β as described !

β is clearly a **spoiling strategy** against α for φ.

Game with imperfect information: an example



φ=**Reach**($\{o_4\}$)

Can Player 2 surely-win the objective Safe($o_1 \cup o_2 \cup o_3$) ?

NO

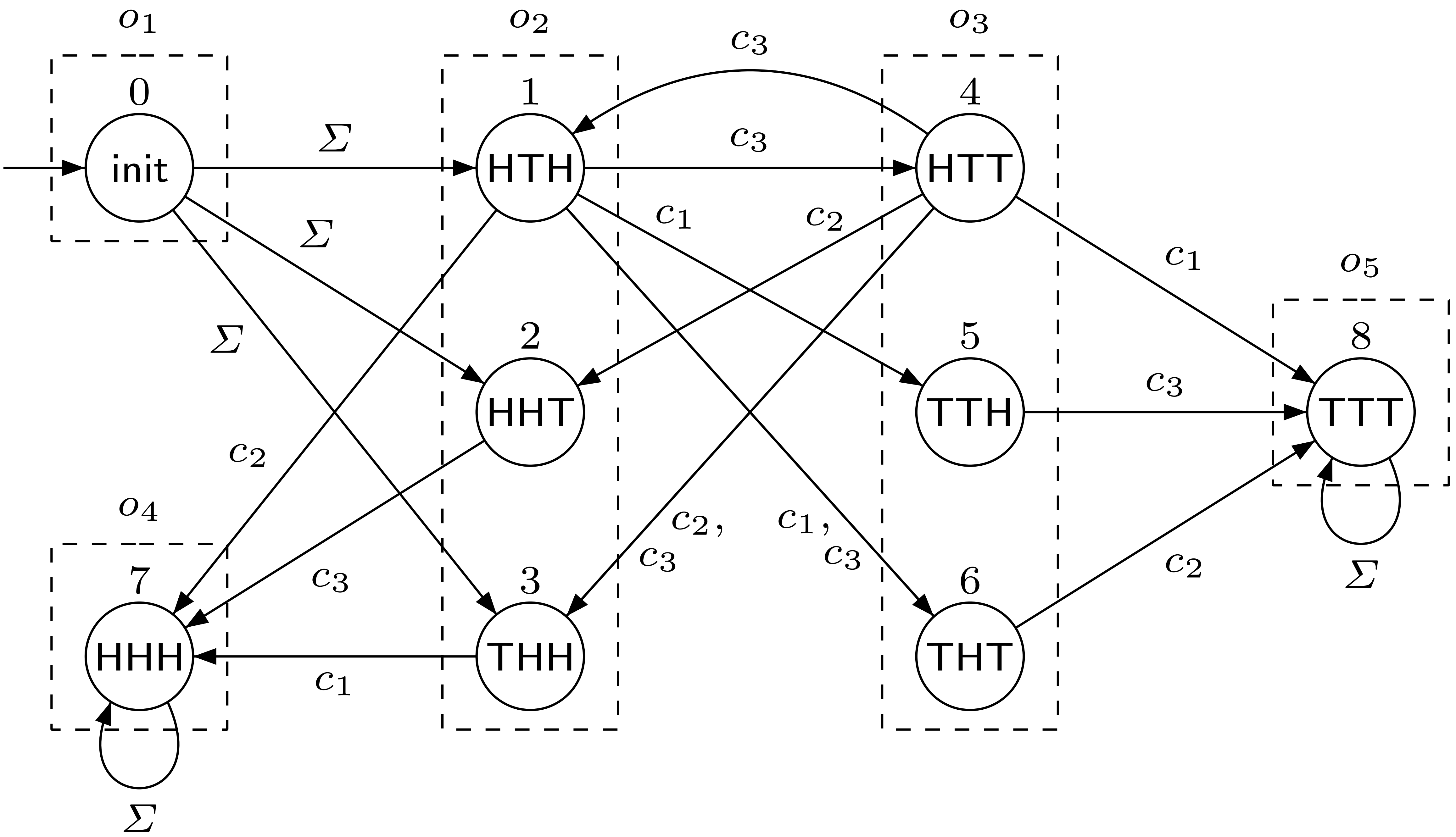
Note that Player 2 does not have a deterministic strategy to ensure **Safe**($o_1 \cup o_2 \cup o_3$).

As **Reach**(o_4) and **Safe**($o_1 \cup o_2 \cup o_3$) are complementary objectives, this shows that games with imperfect information are **not** determined.

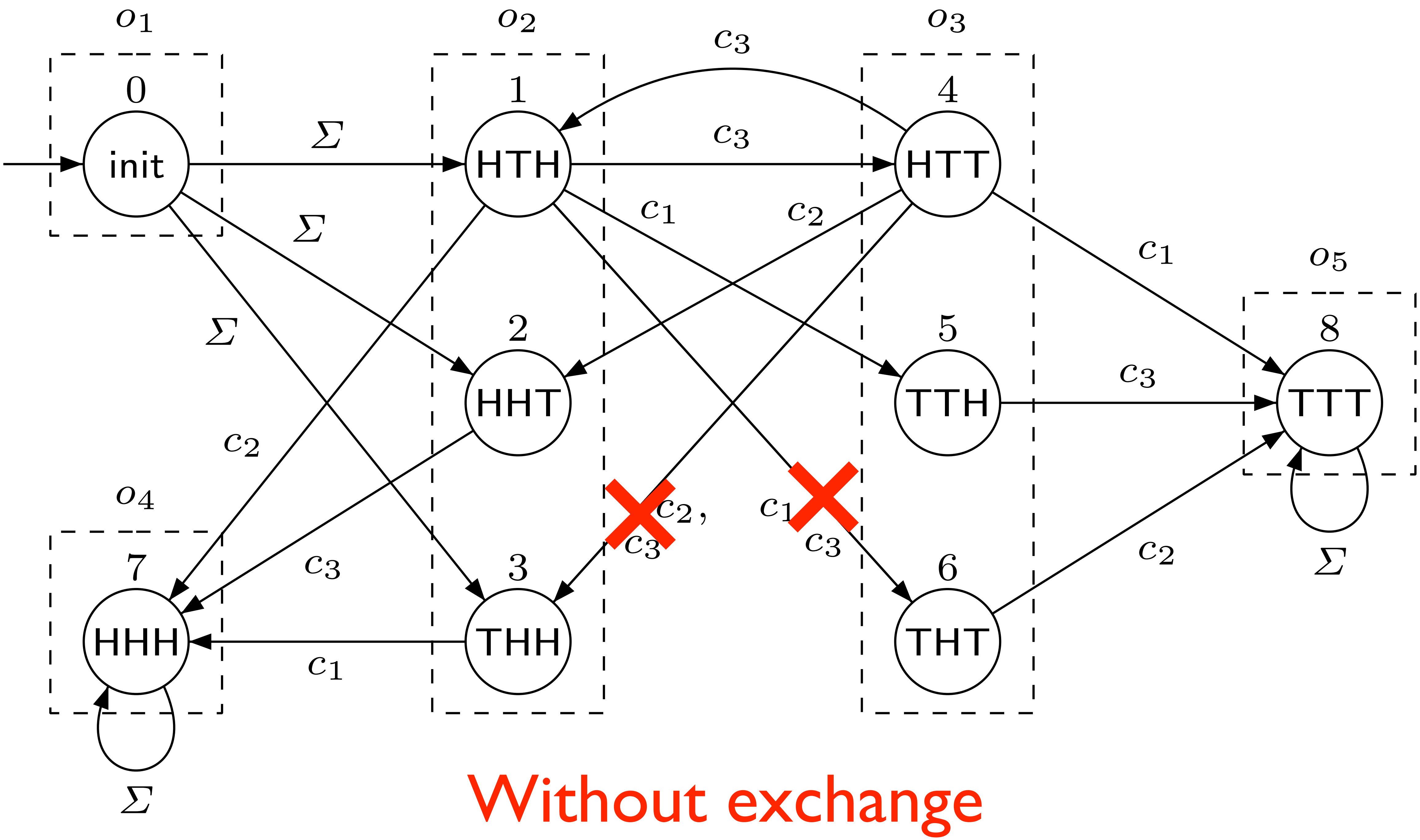
Game with imperfect information: discussion

- The games that we consider sounds **asymmetric**.
- Indeed, Player 1 has **imperfect** information while Player 2 has **perfect** information.
- Nevertheless, making Player 2 weaker (with imperfect information) would not help Player 1 to **surely** win.
- Indeed, it can be shown that **counting strategies** are sufficient for spoiling deterministic strategies.

Game with imperfect information
3-coin example

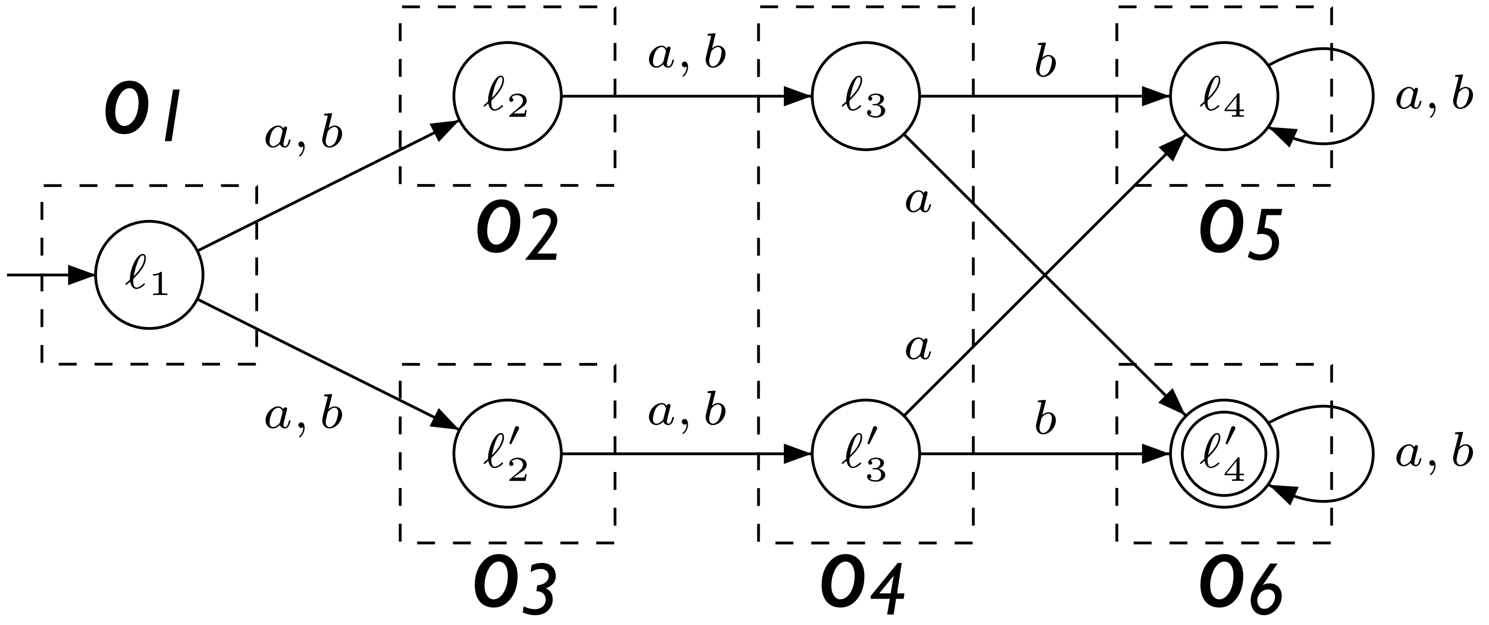


Game with imperfect information
3-coin example



Without exchange

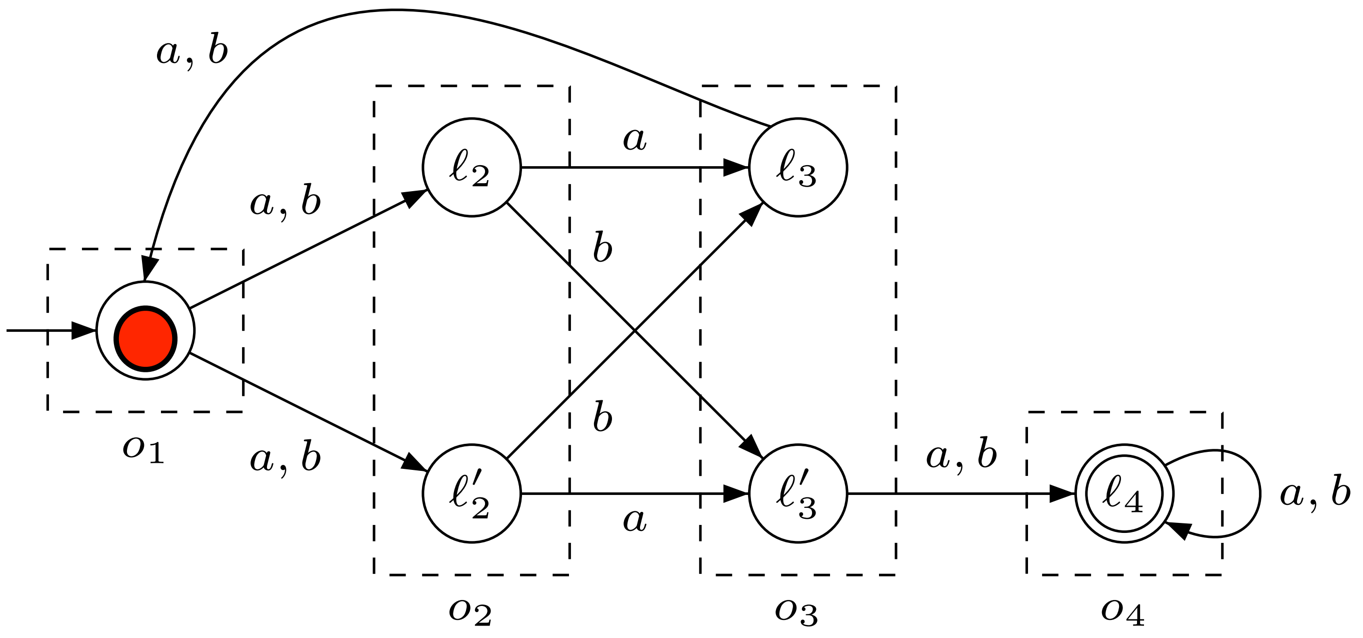
Game with imperfect information
Memory



π=	l_1	*a*	l_2	*b*	l_3	***a***	l_4‘	....
Obs(π)=	o_1		o_2		o_4		o_6	....
π’=	l_1	*a*	l_2’	*b*	l_3‘	***b***	l_4‘	....
Obs(π’)=	o_1		o_3		o_4		o_6	....

Player I needs memory to surely-winning **Reach**({o_6}).

Game with imperfect information
Knowledge



φ=**Reach**({*o₄*})

Can Player 1 win with an observation-based strategy ?

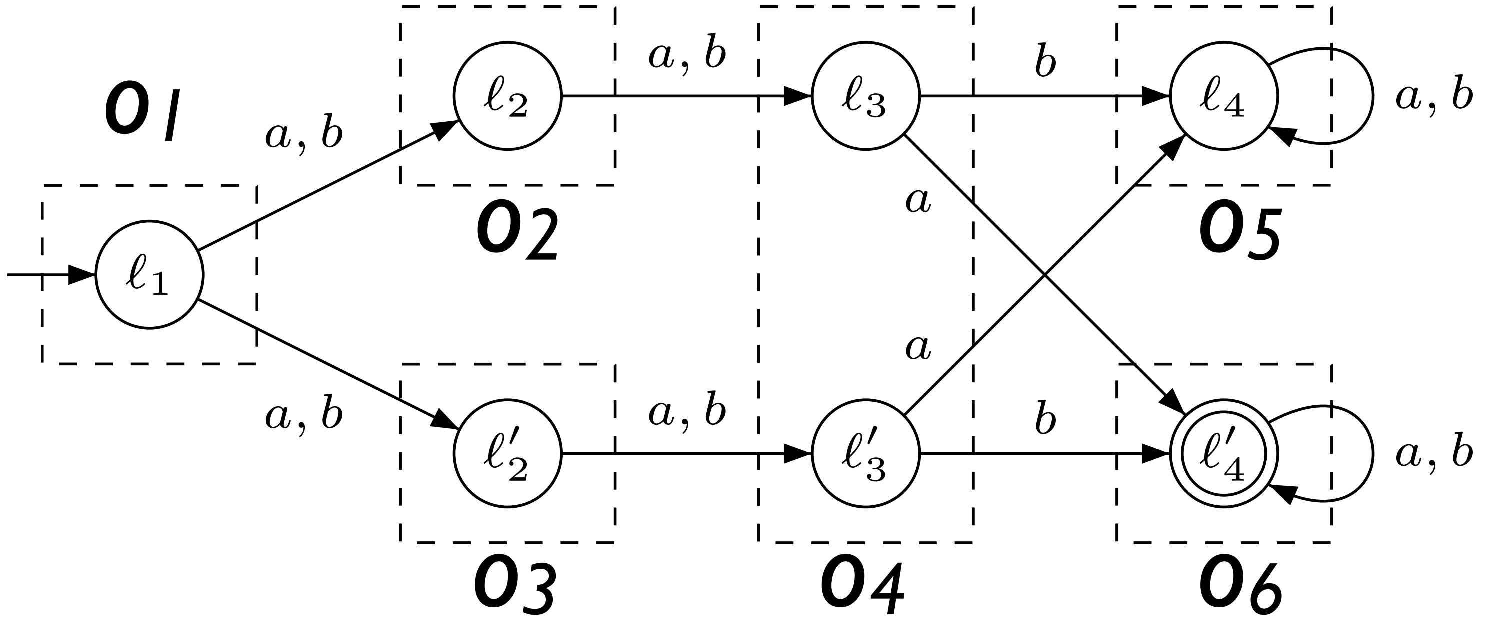
$\pi \quad = \quad l_1 \quad a \quad l_2 \quad a \quad l_3 \quad a \quad l_4 \quad ....$

$obs(\pi) \quad = \quad o_1 \quad o_2 \quad o_3 \quad o_1 \quad ....$

$\Longrightarrow \quad K(\pi) \quad = \quad \{l_1\} \quad \{l_2,l_2'\} \quad \{l_3,l_3'\} \quad \{l_1,\cancel{l_4}\}=\{l_1\}$ ...

$=\mathbf{post}_{G,a}(\{l_1\})\cap o_2$ ....

Game with imperfect information
Knowledge provides memory



Player I needs memory to surely-winning **Reach**($\{o_6\}$).

π=	l_1	a	l_2	b	l_3	a	l_4‘	....
Obs(π)=	o_1		o_2		o_4		o_6	....
K(π)=	$\{l_1\}$		$\{l_2\}$		$\{l_3\}$		$\{l_4$’$\}$	....

The **knowledge** of Player I provides information about the past of the play.

π’=	l_1	a	l_2’	b	l_3‘	b	l_4‘	....
Obs(π’)=	o_1		o_3		o_4		o_6	....
K(π’)=	$\{l_1\}$		$\{l_2\}$		$\{l_3$’$\}$		$\{l_4$’$\}$	....

Reduction to games with perfect information

- Let $G=(L,l_i,\Sigma,\Delta,Obs)$ be a game structure with imperfect information.
- The **knowledge subset construction** of G is the game structure with perfect information $G^K=(S,s_i,\Sigma,\Delta^K)$ where:

  (*i*) $S=\{\ s \in 2^{L\setminus\{\varnothing\}} \mid \exists o \in Obs : s \subseteq o\ \}$,
  elements of S are called **cells**, we note this set **Cells**(Obs).

  (*ii*) $s_i=\{l_i\}$.

  (*iii*) $\Delta^K \subseteq S \times \Sigma \times S$ contains all pairs (s,σ,s') such that
  $\exists o \in Obs \cdot\ s' = \mathbf{post}_{G,\sigma}(s) \cap o$.

  where $\mathbf{post}_{G,\sigma}(s)=\{\ l' \mid \exists\ l \in s : (l,\sigma,l') \in \Delta\ \}$

Reduction to games with perfect information Objectives

- **Observable reachability objectives**. Let T be a union of observations, and φ=**Reach**(T) be an observable reachability objective. Let T^K denotes the set of knowledges { $s \in S$ | $\exists o \in T \cdot s \subseteq o$ }. Then φ^K is defined as **Reach**(T^K).

- **Observable safety objectives**. Let S be a be a union of observations, and φ=**Safe**(S) be an observable safety objective. Let S^K denotes the set of knowledges { $s \in S$ | $\exists o \in S \cdot s \subseteq o$ }. Then φ^K is defined as **Safe**(T^K).

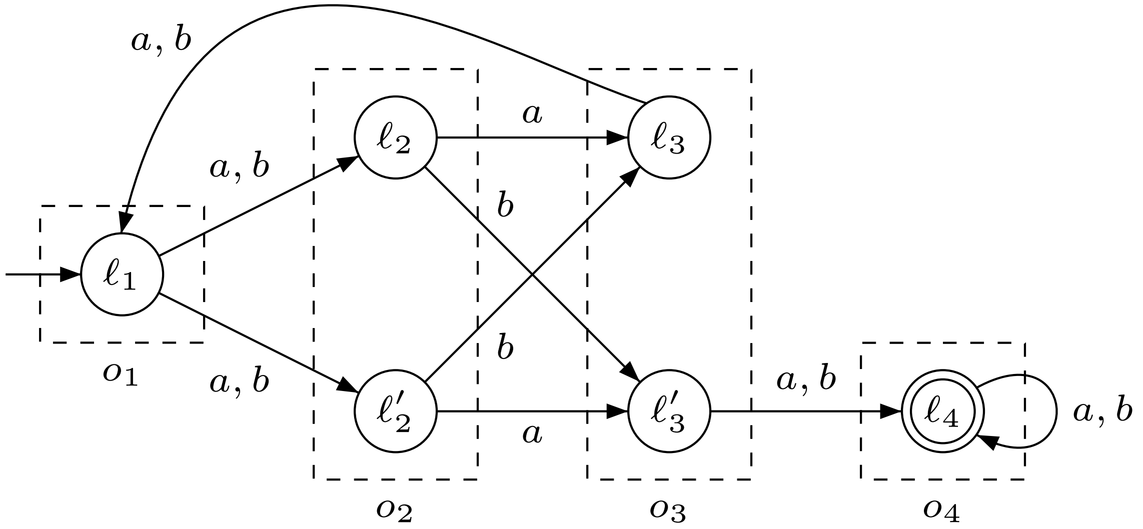
- **Observable parity objectives**. Let $pr:Obs \rightarrow \{1,...,d\}$ be a parity function defining φ=**Parity**(pr) an observable parity objective. Let $pr^K:S \rightarrow \{1,...,d\}$ be the function such that $pr^K(s)=p$ iff $pr(o)=p$ for the observation o such that $s \subseteq o$. Then φ^K is defined as **Parity**(pr^K).

Reduction to games with perfect information Correctness

Theorem. Let $G=(L,l_i,\Sigma,\Delta,Obs)$ be a game structure of imperfect information, let φ be a **observable parity objective**.

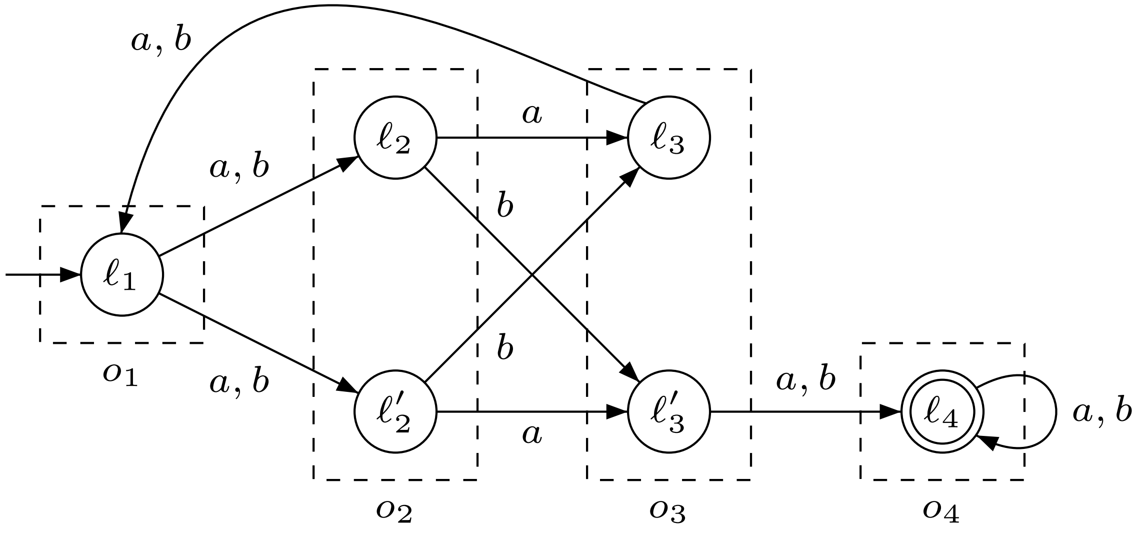
Player I has an observation-based surely-winning strategy in (G,φ) **iff** Player I has a surely-winning in the game with perfect information (G^K,φ^K).

Reduction to games with perfect information
An example

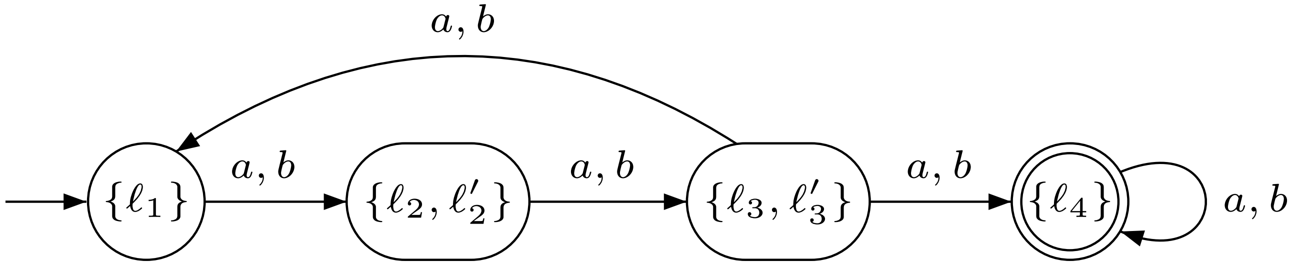


Does Player I have a surely-winning strategy for the objective φ=**Safe**(*L\o₄*) ?

Reduction to games with perfect information
An example



G^K (limited to reachable cells)



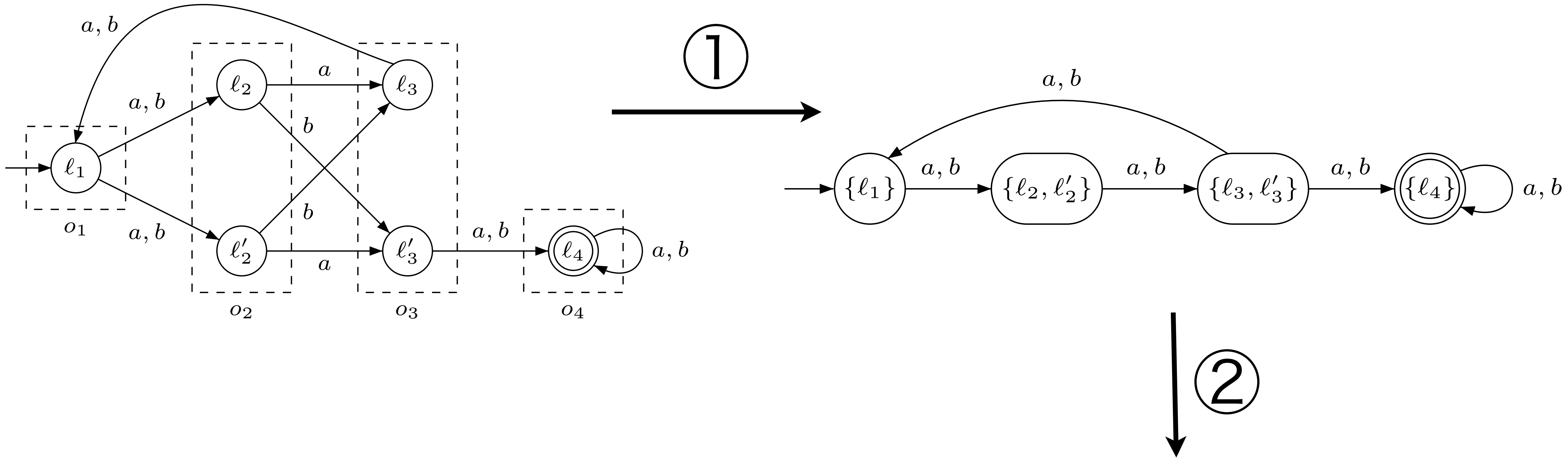
Does Player I have a surely-winning strategy for the objective φ=**Safe**($L\backslash o_4$) ?

Can Player I win φ^K=**Safe**($\{l_1\},\{l_2,l_2'\},\{l_3,l_3'\}$) ?

$X^0=\{\{l_1\},\{l_2,l_2'\},\{l_3,l_3'\}\}$
$X^1=\{\{l_1\},\{l_2,l_2'\}\}$
$X^2=\{\{l_1\}\}$
$X^3=\{\}=X^4$

No !

Reduction to games with perfect information
Algorithm



Does Player 1 have a observation-based surely-winning strategy in *G* for the observable objective φ ?

Does Player 1 have a surely-winning strategy in G^K for the objective φ^K ?

Cpre for the knowledge subset construction

- We want to keep the knowledge-based subset construction implicit !
- For that, we need to define the operator "controllable predecessors" for the knowledge-based subset construction.
- Let $q \subseteq S$ be a set of cells,

  Cpre(q)

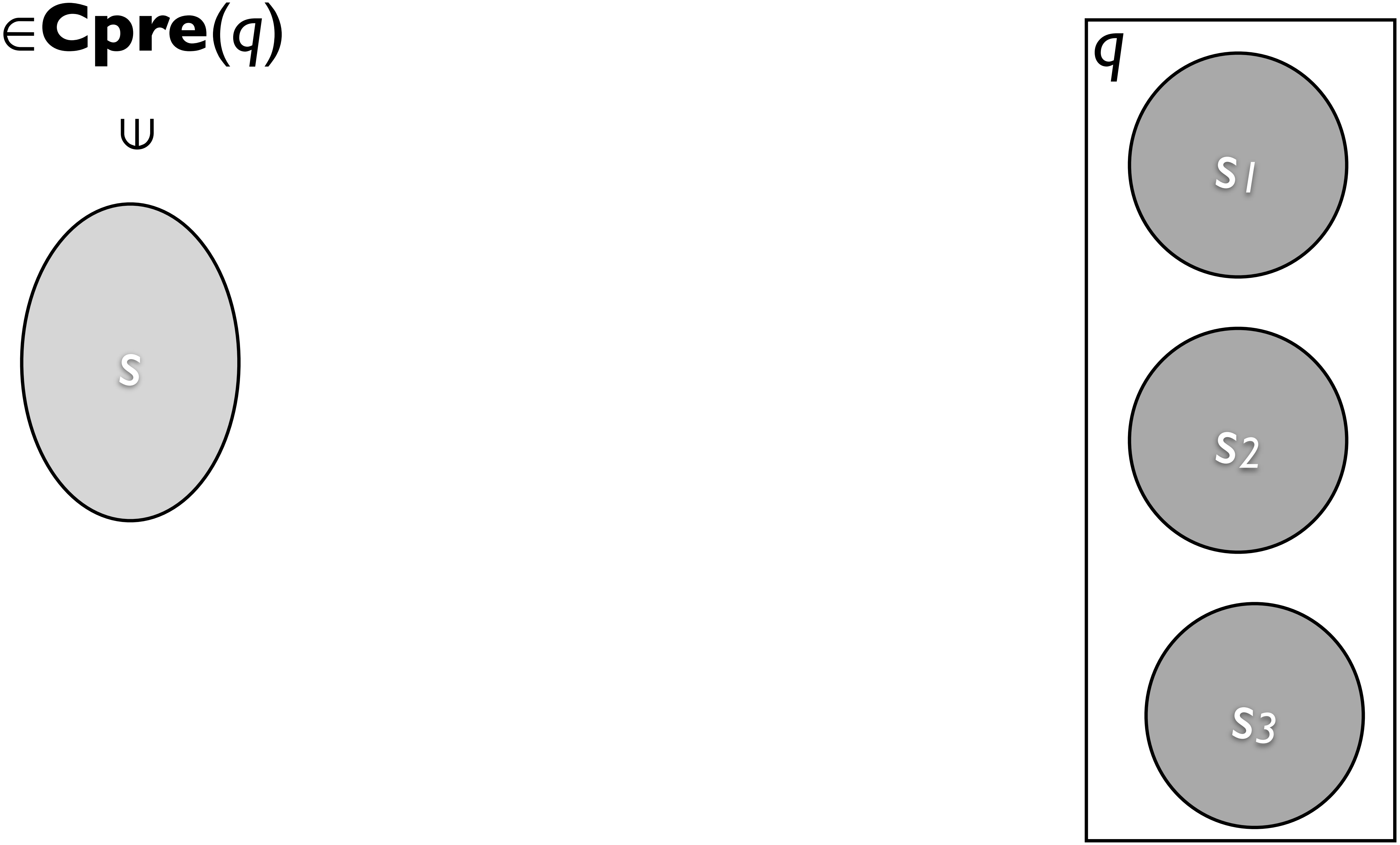
  $=\{ s \mid \exists \sigma \in \Sigma \cdot \forall s' \in S \cdot (s,\sigma,s') \Longrightarrow s' \in q \}$

  $=\{ s \mid \exists \sigma \in \Sigma \cdot \forall o \in Obs \cdot \forall s' \cdot s' = \mathbf{post}_{G,\sigma}(s) \neq \emptyset \Longrightarrow s' \in q \}$

Cpre for the knowledge subset construction

$s \in \mathbf{Cpre}(q)$

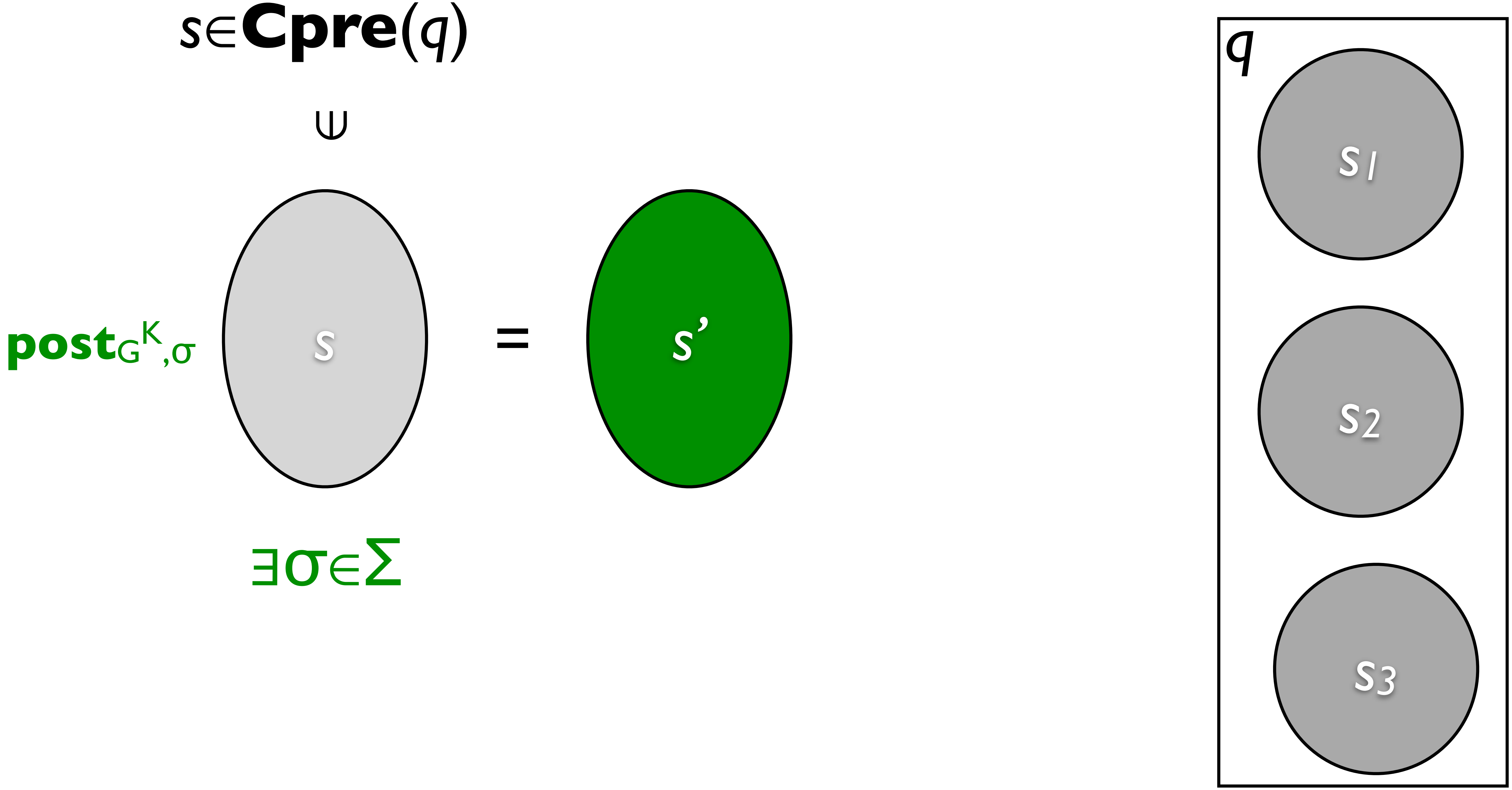
⟒



Cpre for the knowledge subset construction

$s \in \mathbf{Cpre}(q)$

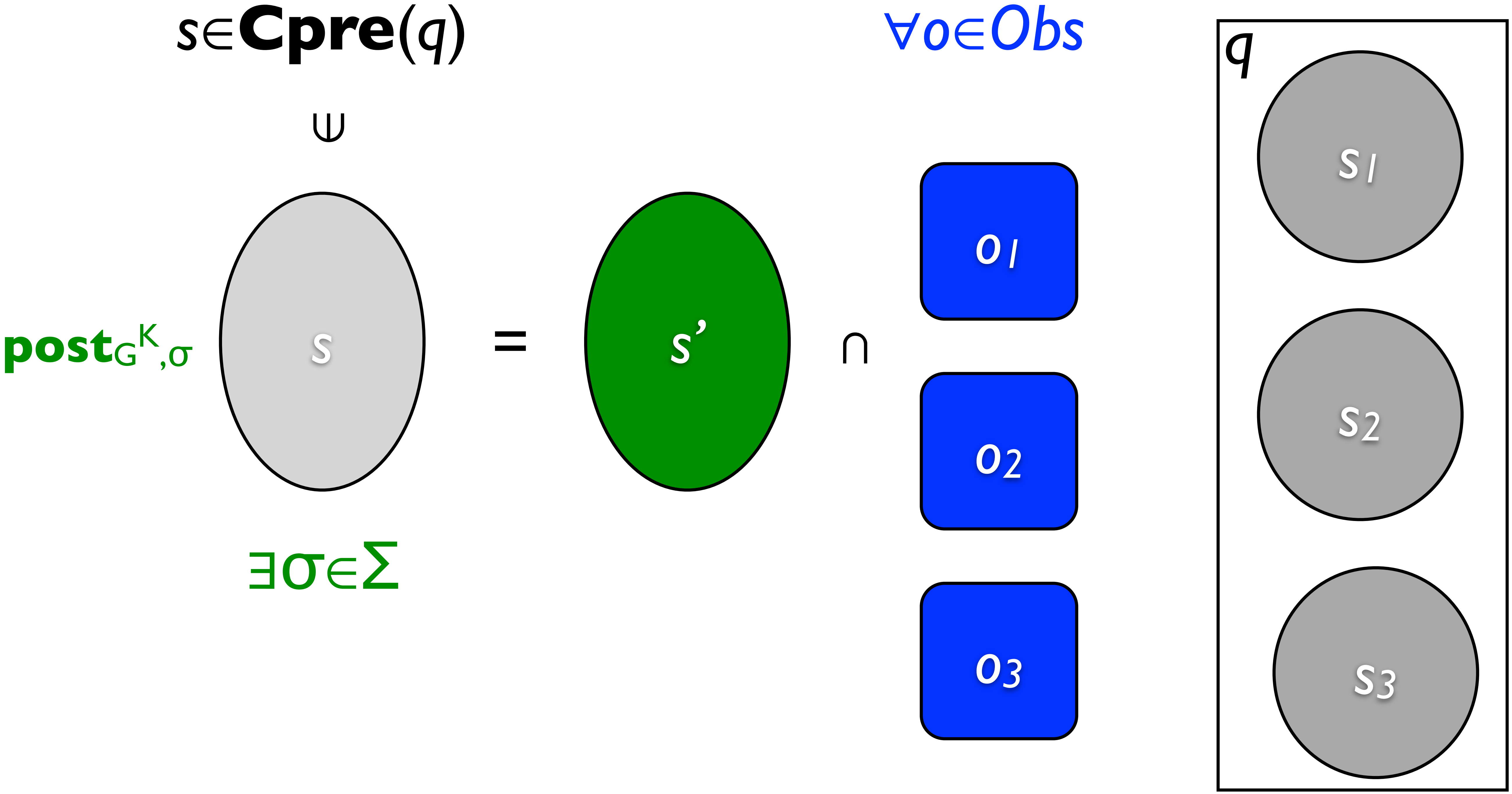
⟒



$\mathbf{post}_{G^K,\sigma}$ (s) = s'

$\exists \sigma \in \Sigma$

Cpre for the knowledge subset construction



$s \in \mathbf{Cpre}(q)$

$\forall o \in Obs$

q

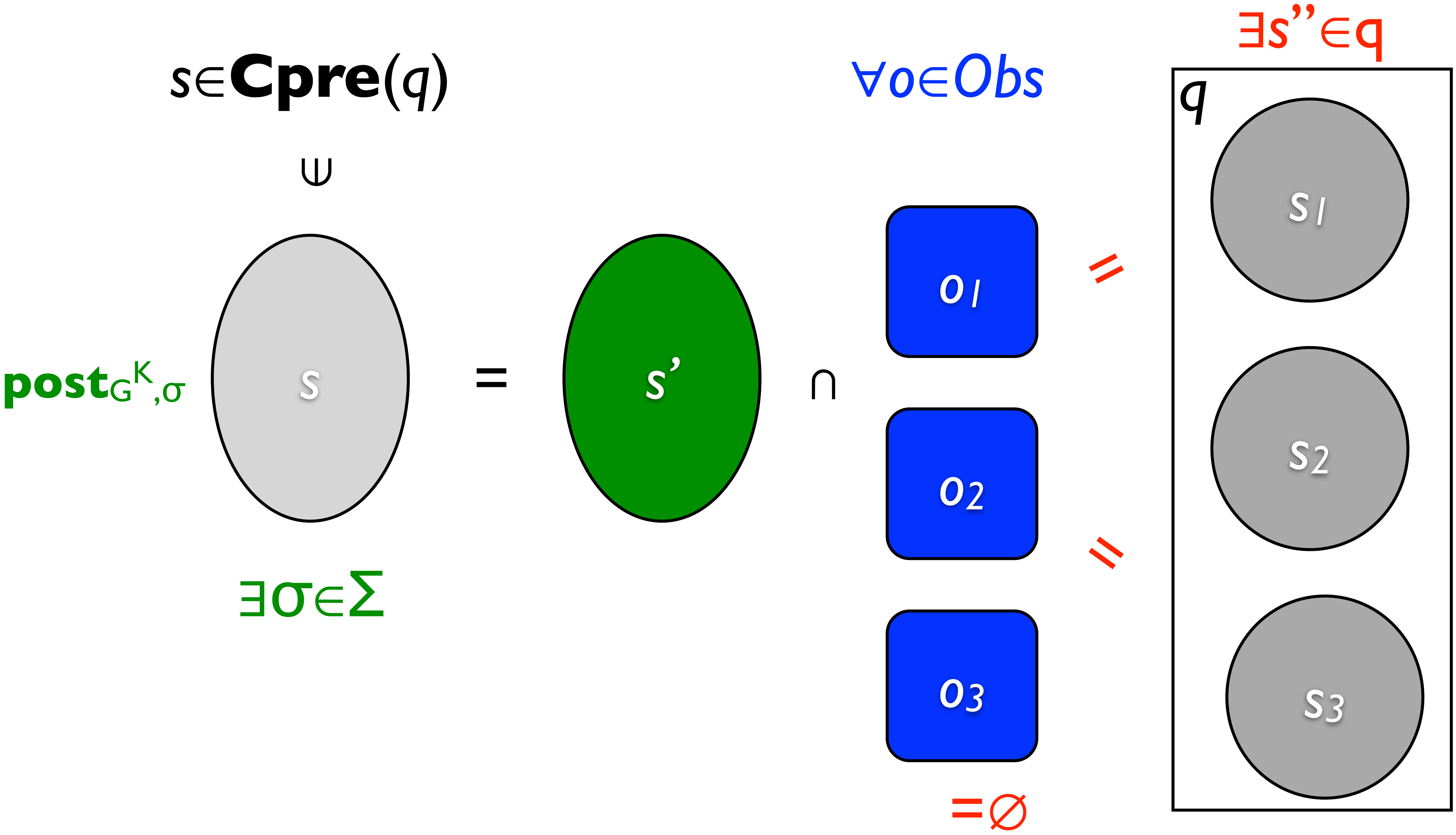
⋓

$\mathbf{post}_{G^K,\sigma}$ s = s' $\cap$ o_1 o_2 o_3

$\exists \sigma \in \Sigma$

s_1 s_2 s_3

Cpre for the knowledge subset construction



$s \in \mathbf{Cpre}(q)$

⋓

$\mathbf{post}_{G^K,\sigma}$ s $=$ s' $\cap$

$\exists \sigma \in \Sigma$

$\forall o \in Obs$

o_1 $=$

o_2 $=$

o_3 $= \varnothing$

$\exists s'' \in q$

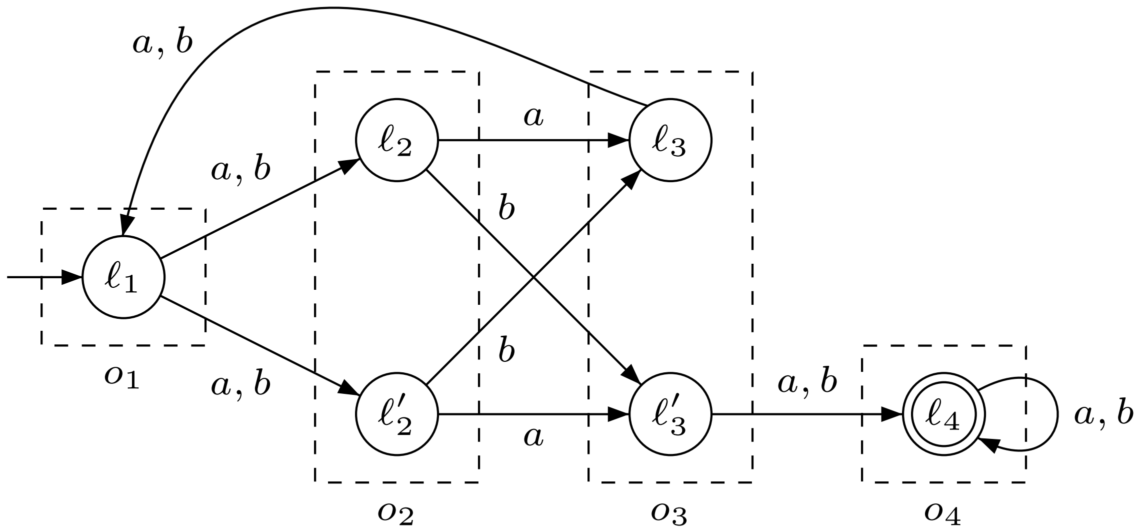
q

s_1

s_2

s_3

Cpre for the knowledge subset construction An example



$q=\{\{l_2,l_2'\},\{l_4\},\{l_1\}\}$

$\{l_3,l_3'\}\in$**Cpre**(q) because if Player 1 chooses action *a*, we verify that:

- ☞ for o_1, $\mathbf{post}_{G,a}(\{l_3,l_3'\}) \cap o_1 = \{l_1\}$
- ☞ for o_2, $\mathbf{post}_{G,a}(\{l_3,l_3'\}) \cap o_2 = \varnothing$
- ☞ for o_3, $\mathbf{post}_{G,a}(\{l_3,l_3'\}) \cap o_3 = \varnothing$
- ☞ for o_4, $\mathbf{post}_{G,a}(\{l_3,l_3'\}) \cap o_4 = \{l_4\}$

Cpre for the knowledge subset construction Downward-closed sets

- A set of cells q is ⊆**-downward closed**

  iff $\forall s \in q \cdot \forall s' \subseteq s \cdot s' \neq \varnothing \Rightarrow s' \in q$.

- The ⊆**-downward closure** of a set of cells q is the set of cells

  $\downarrow q = \{ s \neq \varnothing \mid \exists s' \in q \cdot s \subseteq s' \}$.

- **Proposition**. For any set of cells q, **Cpre**($\downarrow q$) is ⊆-downward closed.

  Proof.This is a direct consequence of the fact that for all cells s_1, s_2, for all $\sigma \in \Sigma$, we have that $s_1 \subseteq s_2 \Rightarrow \mathbf{post}_{G^K,\sigma}(s_1) \subseteq \mathbf{post}_{G^K,\sigma}(s_2)$, and that intersections and unions of ⊆-downward closed sets are ⊆-downward closed sets.

- **Corollary.** All the sets manipulated during fixed point computations for observable safety and reachability objectives are ⊆-downward closed.

Antichains for representing downward-closed sets

- An ⊆**-antichain** q over 2^L is a set of sets of locations such that

  $$\forall s_1, s_2 \in q \cdot s_1 \subseteq s_2 \Longrightarrow s_1 = s_2$$

- We note **A** the set of antichains.

- Note that elements within an antichain are not necessarily cells.

- A ⊆**-antichain** q compactly represents the downward closed set of cells

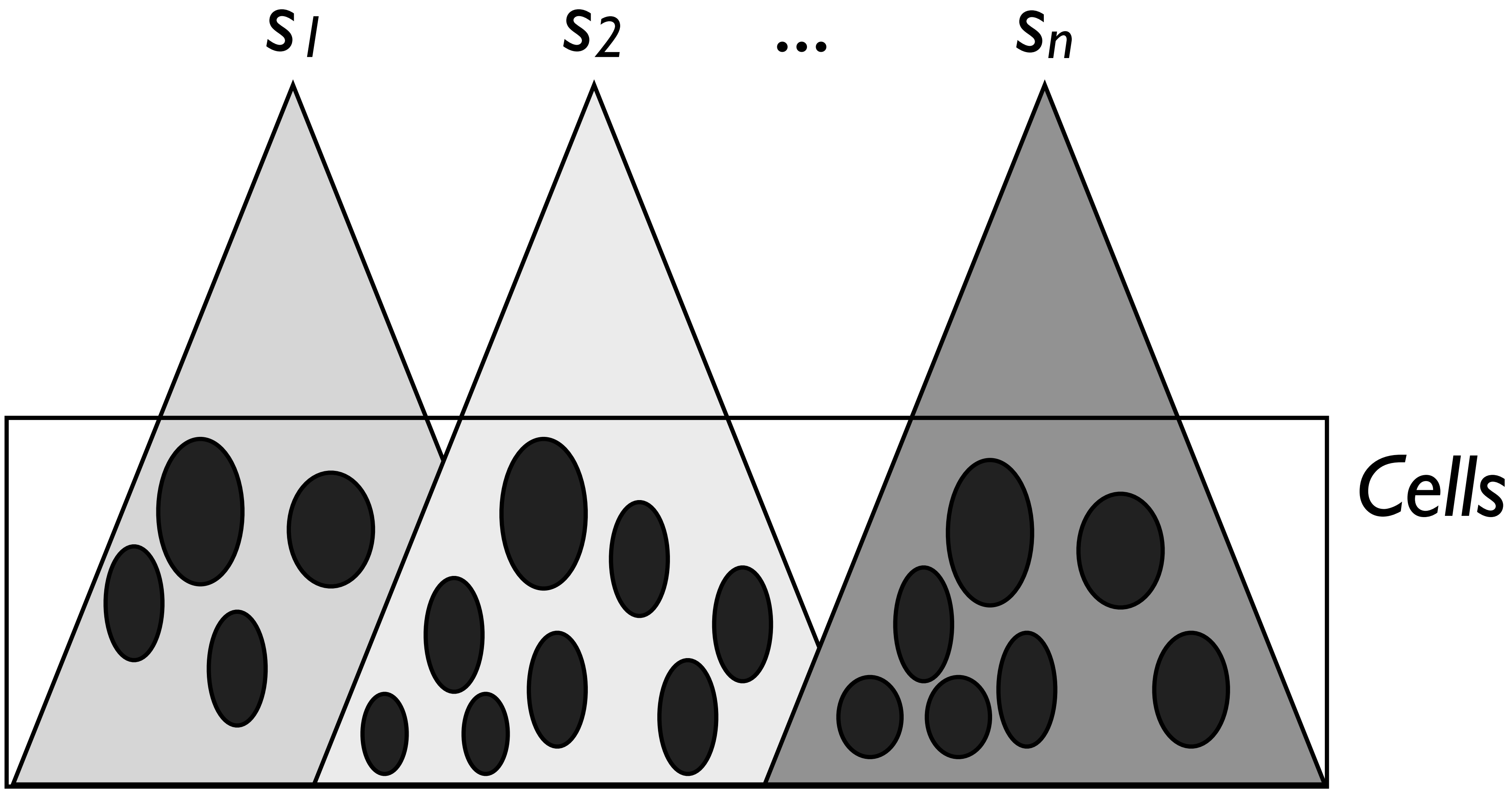
  $$\downarrow q = \{ s \in \mathbf{Cells}(Obs) \mid \exists s' \in q \cdot s \subseteq s' \}$$

- The set **A** is partially ordered as follows:

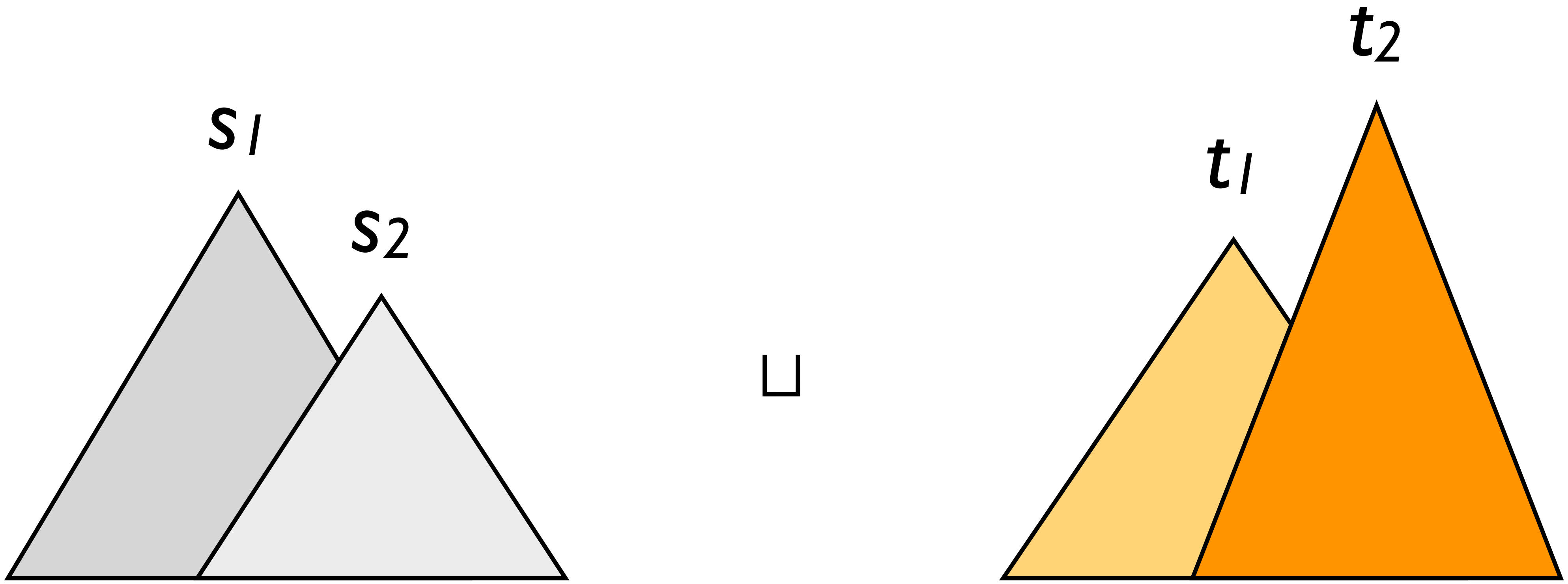
  $$q \sqsubseteq q' \text{ iff } \forall s \in q \cdot \exists s' \in q' \cdot s \subseteq s'$$

- We note $q \sqcap q'$ the **greatest lower bound** of q and q' in **A**, and $q \sqcup q'$ the **least upper bound** of q and q' in **A**.

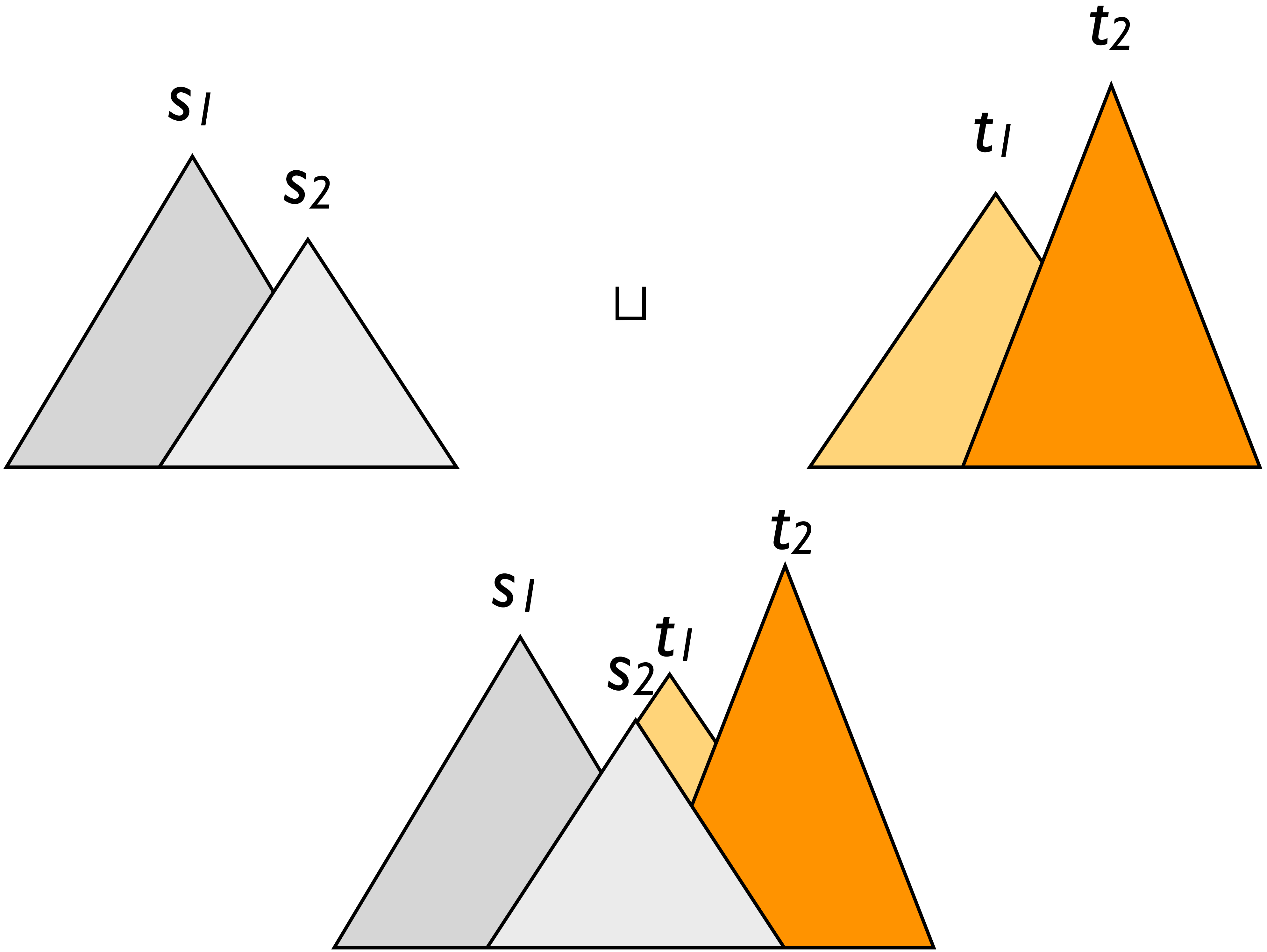
Antichains for representing downward-closed sets



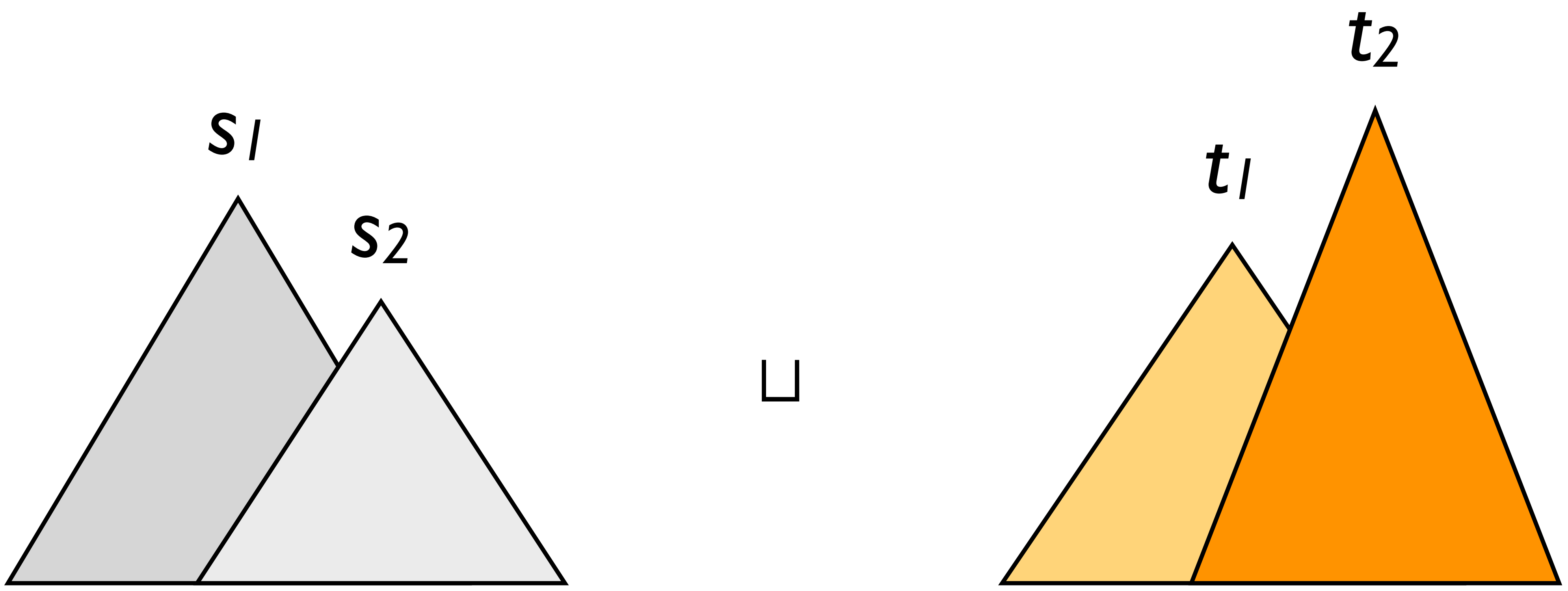
Antichains for representing downward-closed sets

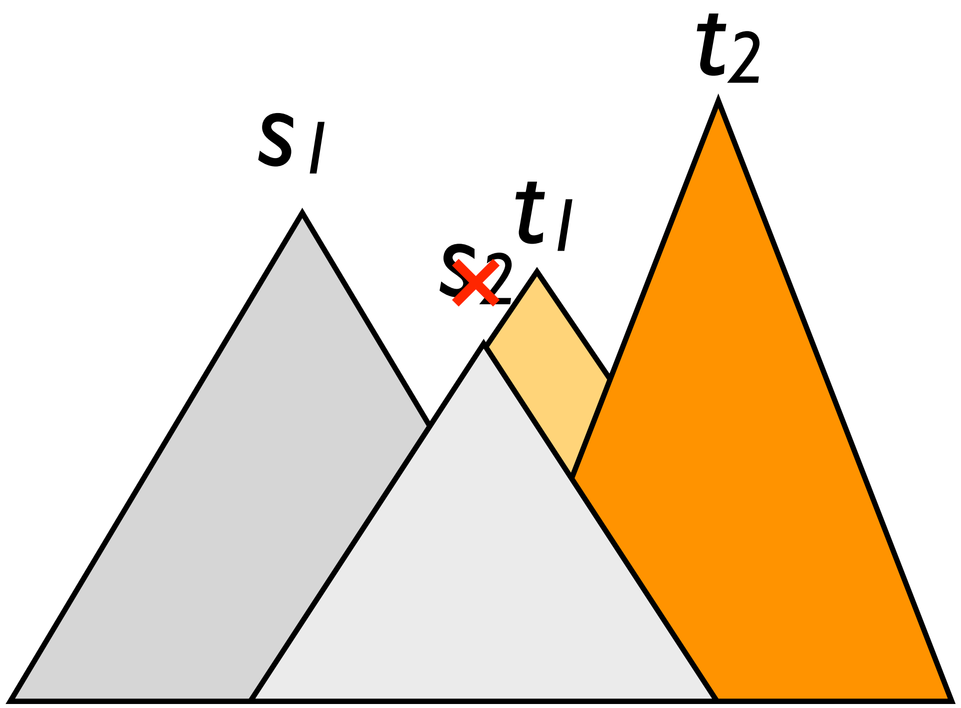


Antichains for representing downward-closed sets



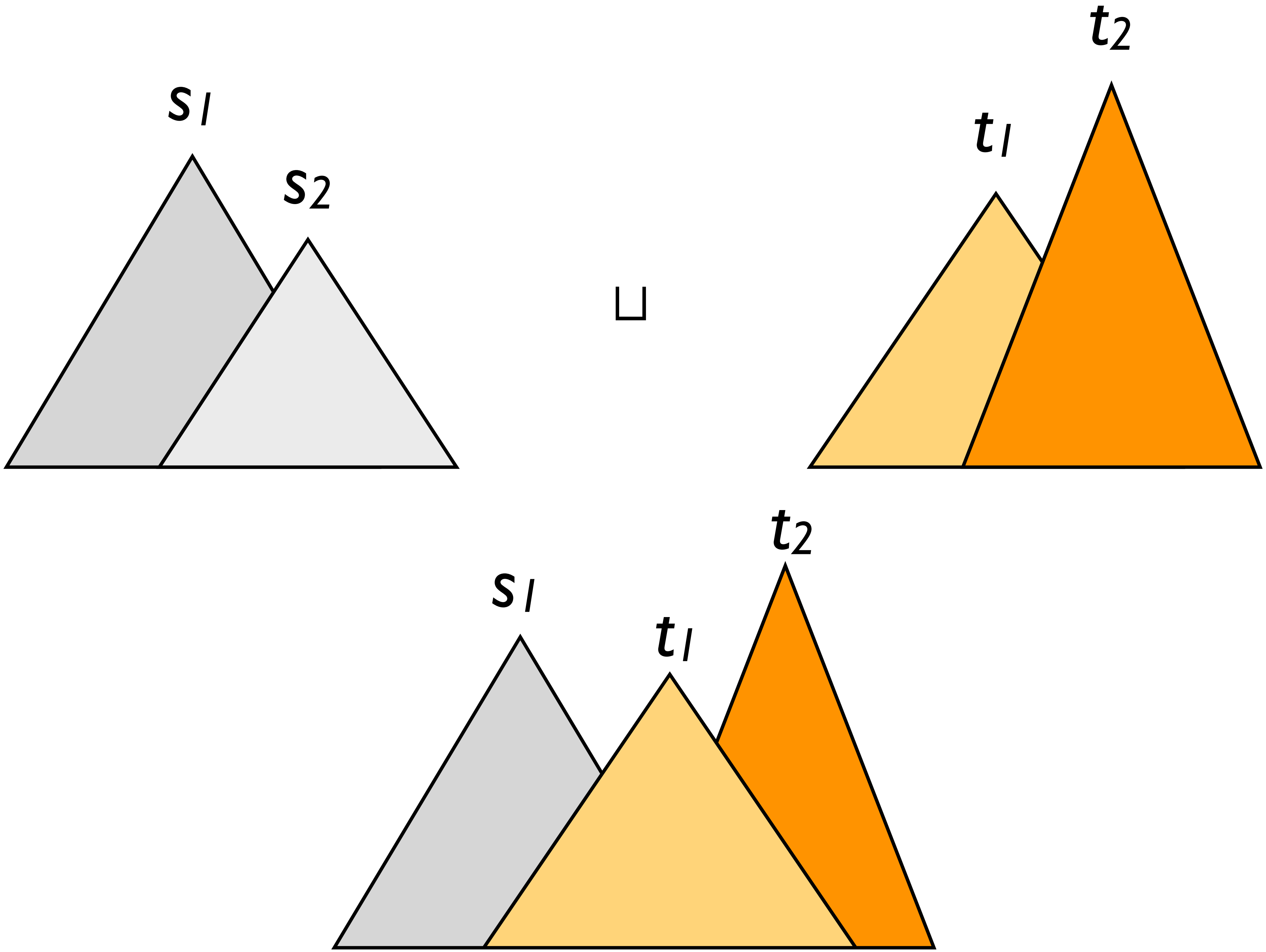
Antichains for representing downward-closed sets



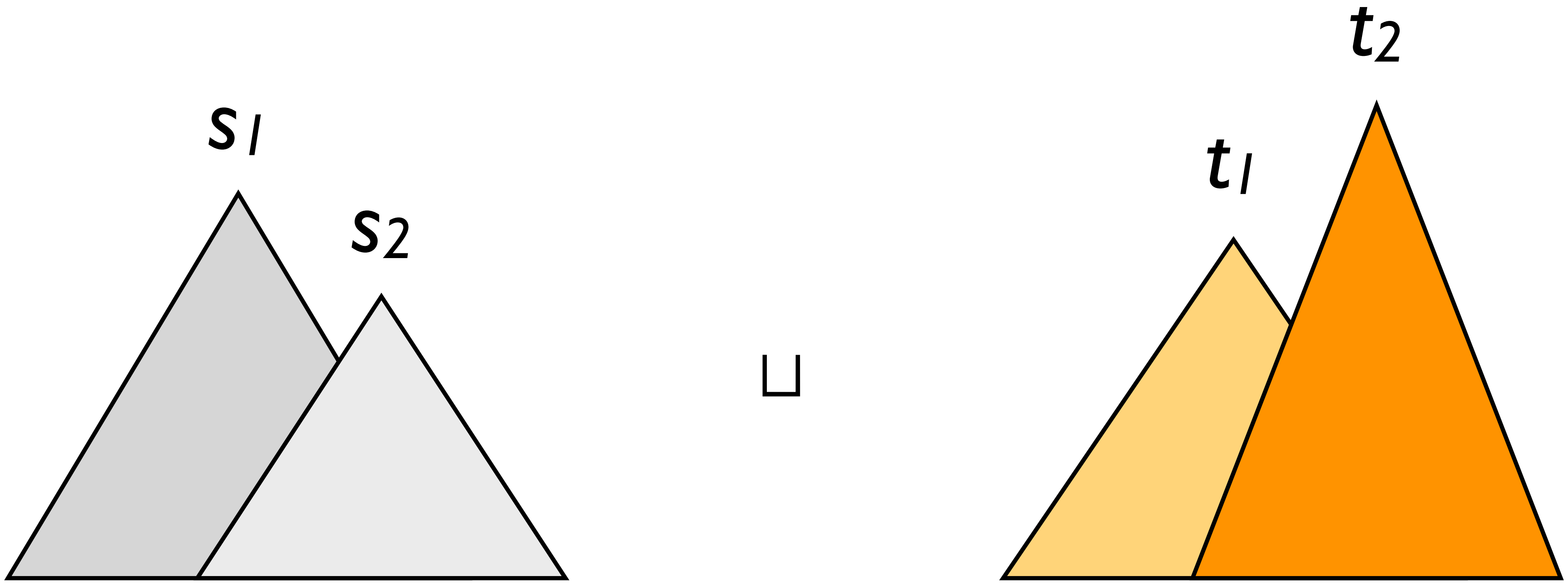


keep only maximal elements

Antichains for representing downward-closed sets

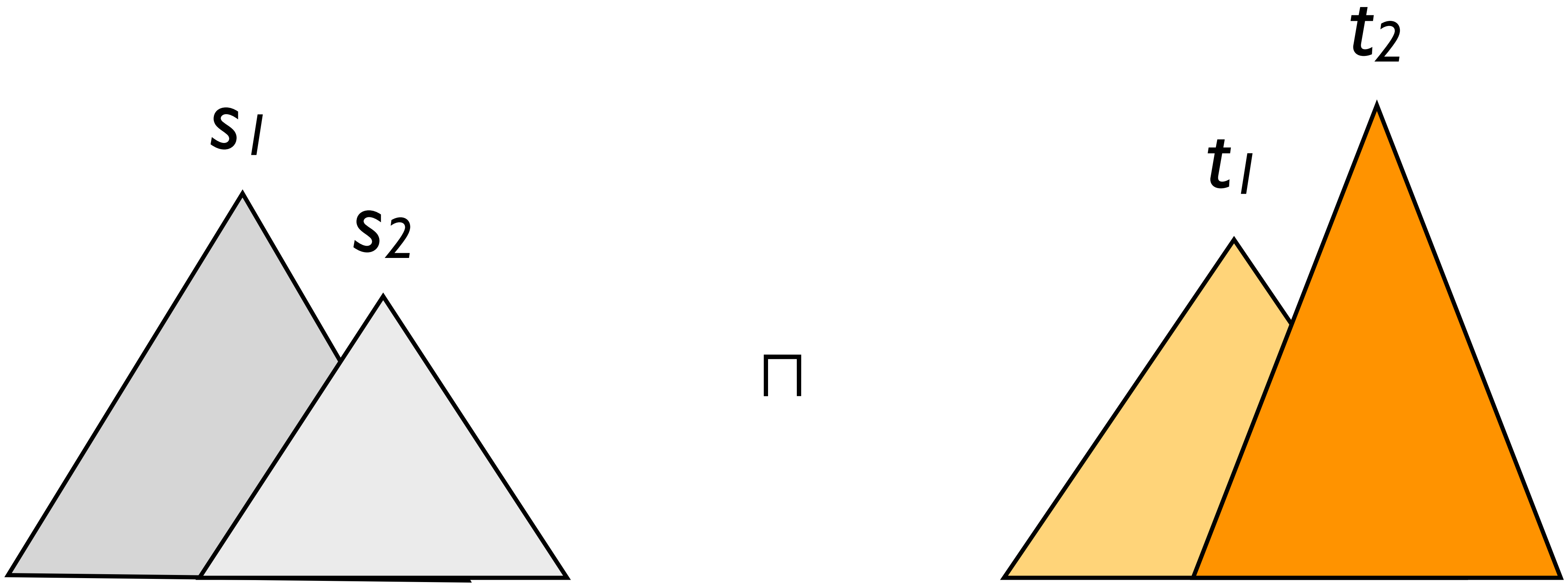


Antichains for representing downward-closed sets

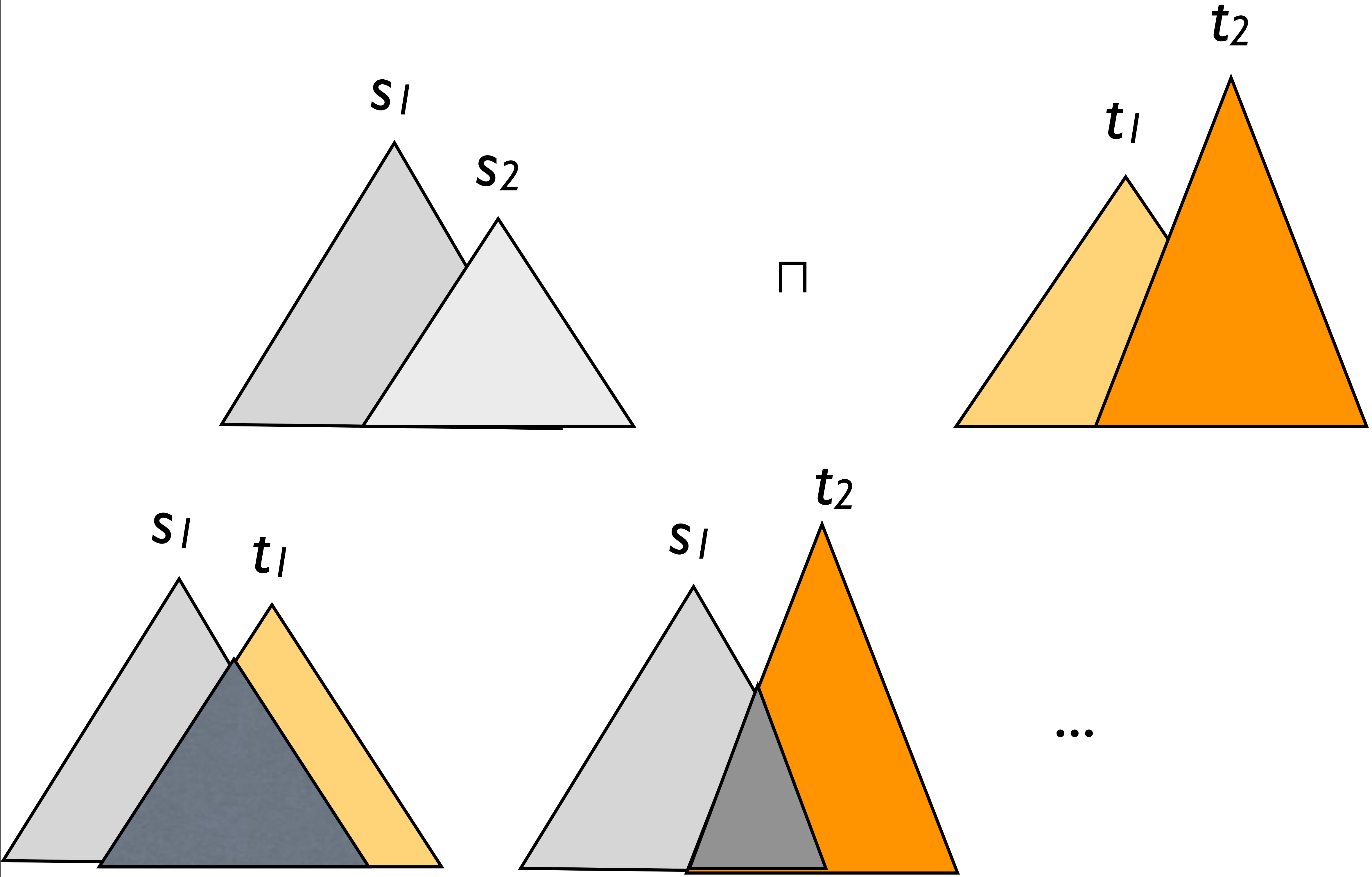


$$q_1 \sqcup q_2 = \mathrm{Max}_{\subseteq}\{\, s \mid s \in q_1 \vee s \in q_2 \,\}$$

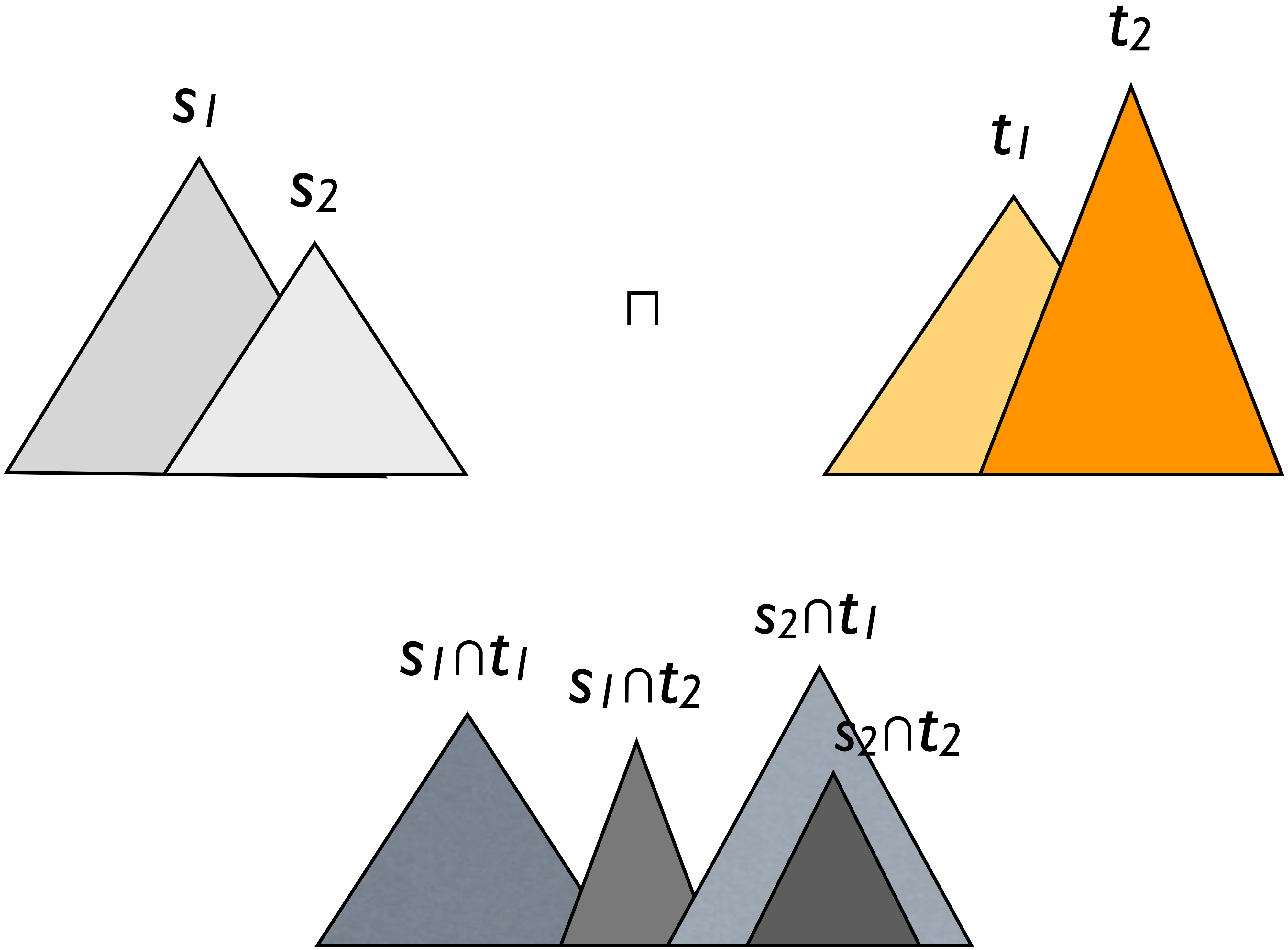
Antichains for representing downward-closed sets



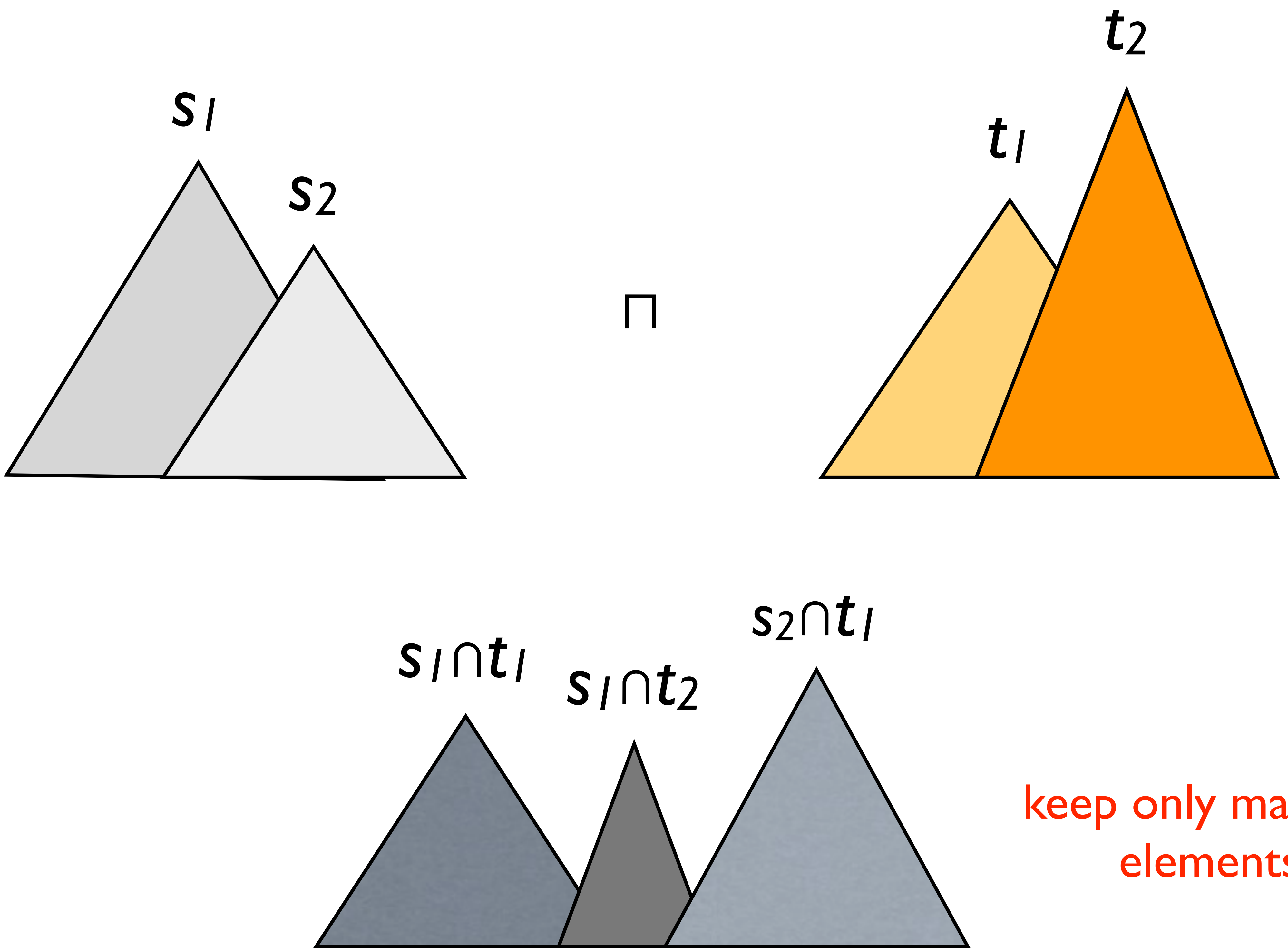
Antichains for representing downward-closed sets



Antichains for representing downward-closed sets

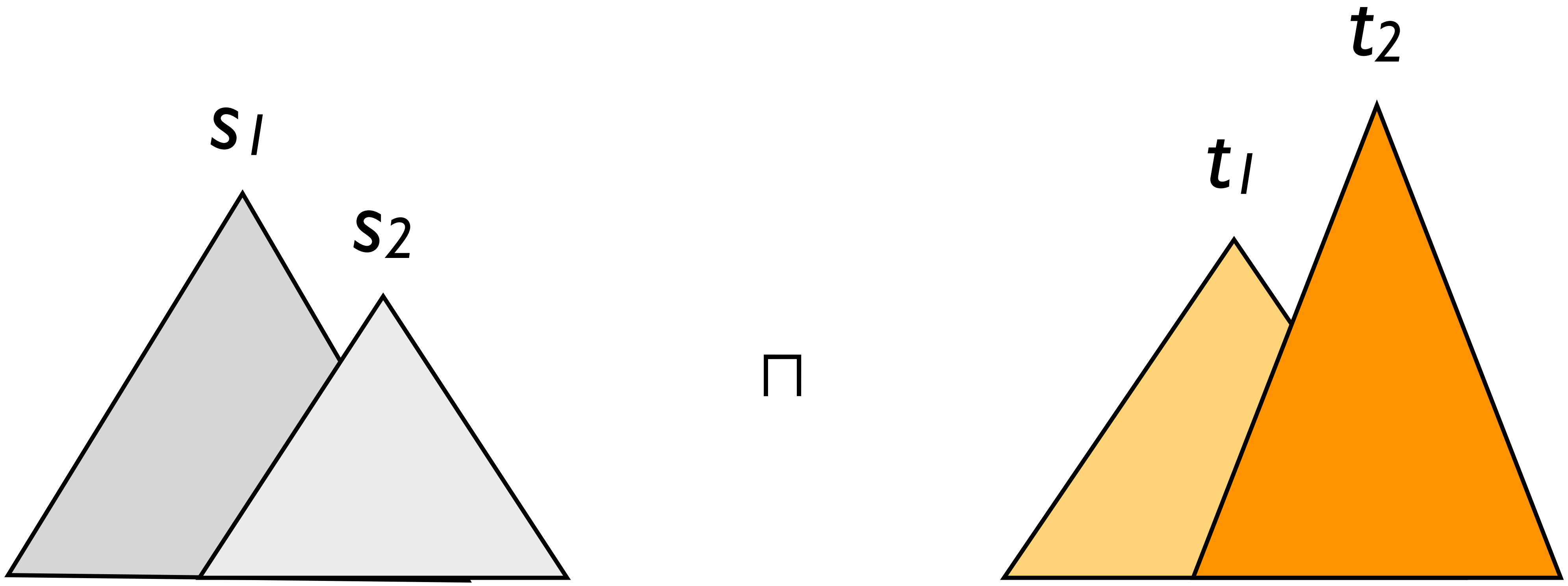


Antichains for representing downward-closed sets



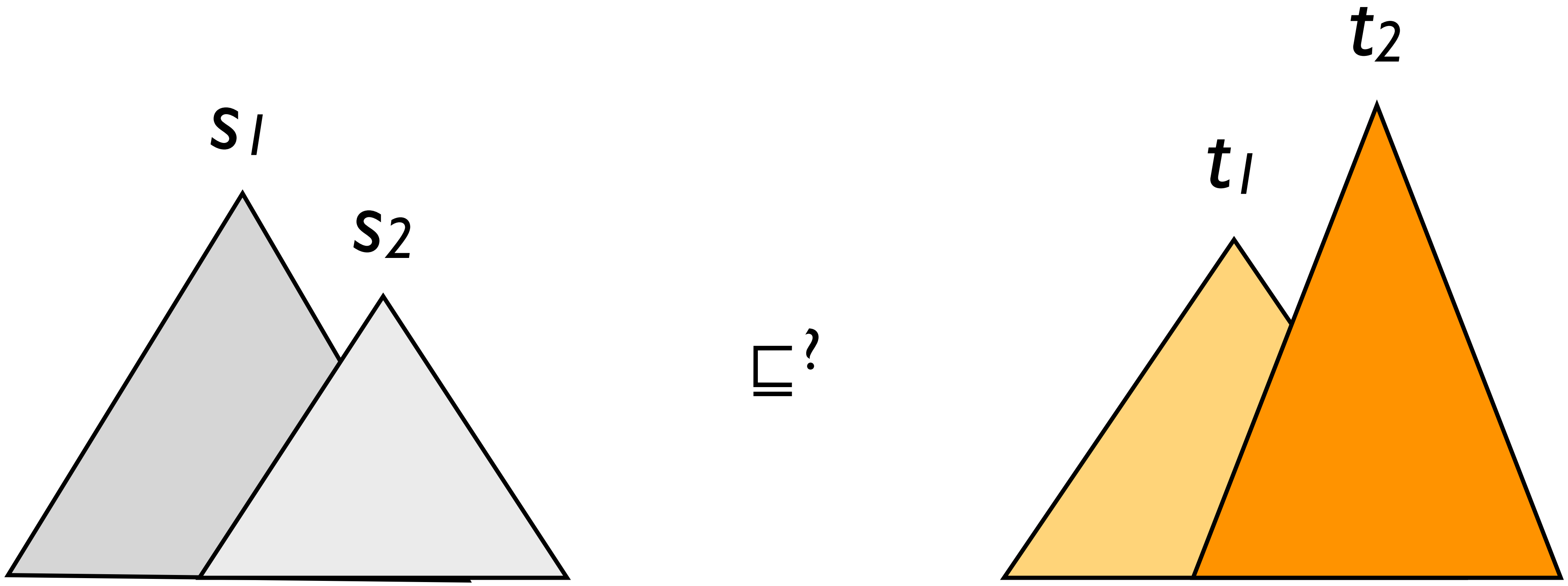
keep only maximal elements

Antichains for representing downward-closed sets



$$q_1 \sqcap q_2 = \mathrm{Max}_{\subseteq}\{\, s_1 \cap s_2 \mid s_1 \in q_1 \wedge s_2 \in q_2 \,\}$$

Antichains for representing downward-closed sets



$q_1 \sqsubseteq q_2$ iff $\forall s_1 \in q_1 \cdot \exists s_2 \in q_2 \cdot s_1 \subseteq s_2$

Cpre of an $\subseteq$-antichain

- As we can **compactly** represent set of cells as antichains, we want to compute directly the **Cpre** of an antichain.

- **Cpre**($\downarrow q$)

  $=\{\, s \in S \mid \exists \sigma \in \Sigma \cdot \forall o \in Obs \cdot \exists s' \in q \cdot \mathbf{post}_{G,\sigma}(s) \cap o \subseteq s' \}$

  $=\{\, s \in S \mid \exists \sigma \in \Sigma \cdot \forall o \in Obs \cdot \exists s' \in q \cdot s \subseteq \mathbf{apre}_{G,\sigma}(s' \cup (L \backslash o)) \}$

  where $\mathbf{apre}_{G,\sigma}(s)=\{\, l \in L \mid \mathbf{post}_{G,\sigma}(\{l\}) \subseteq s \}$

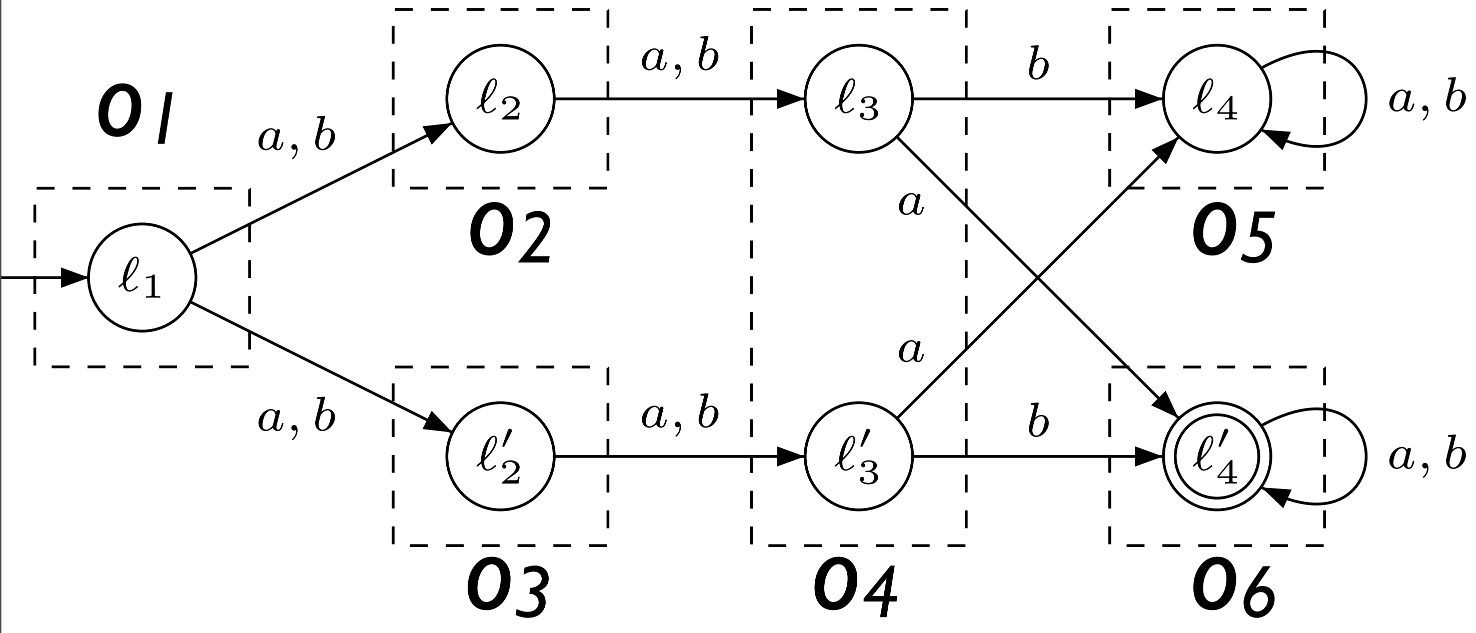
- $\mathbf{Cpre}^{\mathbf{A}}(\downarrow q)=\bigsqcup_{\sigma \in \Sigma} \bigsqcap_{o \in Obs} \bigsqcup_{s' \in q} \left\{\, \mathbf{apre}_{G,\sigma}(s' \cup (L \backslash o)) \,\right\}$

Cpre of an ⊆-antichain

- All the operations on antichains for the $\mathbf{Cpre^{A}}$ can be implemented in polynomial time except $\sqcap_{o \in Obs}$.
- **Theorem.** There is no polynomial time algorithm to compute $\sqcap_{i \in I} q_i$ unless P=NP.

  Proof. Consider a graph (V,E), the set of its independent sets is $\left(\sqcap_{(v,w) \in E} \{V \setminus \{v\}, V \setminus \{w\}\}\right)\downarrow$.

Solving reachability: an example



Does Player 1 have an observation-based strategy to force *o₆* ?

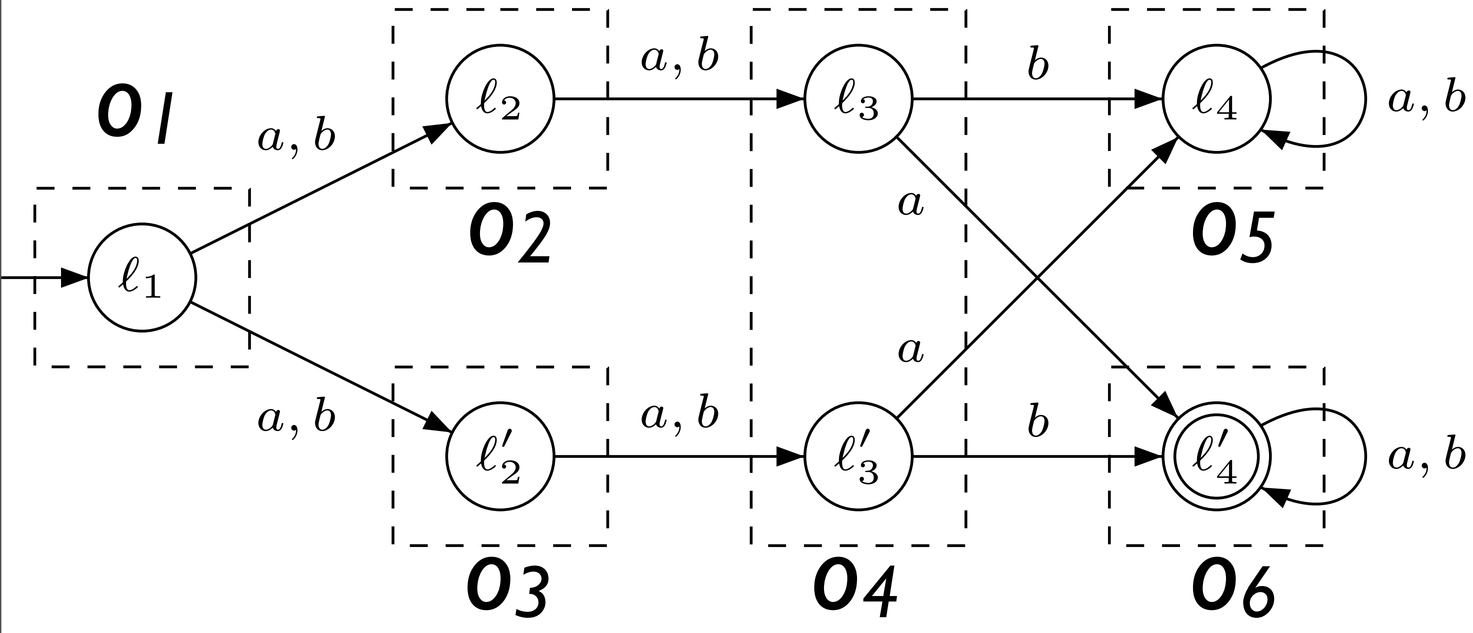
$X^0=\{\{l_4'\}\}$

$X^1=\{\{l_4'\}\} \sqcup \mathbf{Cpre^A}(\{\{l_4'\}\})$
$\quad = \{\{l_4'\}\} \sqcup \{\{l_3\},\{l_3'\}\}= \{\{l_4'\},\{l_3\},\{l_3'\}\}$

$X^2= \{\{l_4'\}\} \sqcup \mathbf{Cpre^A}(\{\{l_4'\},\{l_3\},\{l_3'\}\})$
$\quad = \{\{l_4'\}\} \sqcup \{\{l_4'\},\{l_3\},\{l_3'\},\{l_2\},\{l_2'\}\}= \{\{l_4'\},\{l_3\},\{l_3'\},\{l_2\},\{l_2'\}\}$

$X^3= \{\{l_4'\}\} \sqcup \mathbf{Cpre^A}(\{\{l_4'\},\{l_3\},\{l_3'\},\{l_2\},\{l_2'\}\})$
$\quad = \{\{l_4'\}\} \sqcup \{\{l_4'\},\{l_3\},\{l_3'\},\{l_2\},\{l_2'\},\{l_1\}\}= \{\{l_4'\},\{l_3\},\{l_3'\},\{l_2\},\{l_2'\},\{l_1\}\}=X^4$

Solving safety: an example



Does Player 1 have a observation-based strategy to avoid o_6 ?

$X^0=\{L\setminus\{l_4'\}\}$

$X^1=\{L\setminus\{l_4'\}\} \sqcap \mathbf{Cpre^A}(\{L\setminus\{l_4'\}\})$

$=\{L\setminus\{l_4'\}\} \sqcap \{L\setminus\{l_3,l_4'\}, L\setminus\{l_3',l_4'\}\}=\{L\setminus\{l_3,l_4'\}, L\setminus\{l_3',l_4'\}\}$

$X^2=\{L\setminus\{l_4'\}\} \sqcap \mathbf{Cpre^A}(\{L\setminus\{l_3,l_4'\}, L\setminus\{l_3',l_4'\}\})$

$=\{L\setminus\{l_4'\}\} \sqcap \{L\setminus\{l_2,l_3,l_4'\}, L\setminus\{l_2',l_3',l_4'\}, L\setminus\{l_2',l_3,l_4'\}, L\setminus\{l_2,l_3',l_4'\}\}$

$=\{L\setminus\{l_2,l_3,l_4'\}, L\setminus\{l_2',l_3',l_4'\}, L\setminus\{l_2',l_3,l_4'\}, L\setminus\{l_2,l_3',l_4'\}\}=X^3$

Antichains in other applications

- Those techniques can be applied with success to LTL model-checking (see [DDMR08] - TACAS08 paper)
- To timed games with imperfect information (see [CDLR07] - ATVA07 paper)
- ... and LTL synthesis (see [FJR09] - CAV09 paper).
- **Antichains**: symbolic data-structure to handle huge state spaces in games with imperfect information and in several important problems from automata theory.

Practical evaluation
Universality

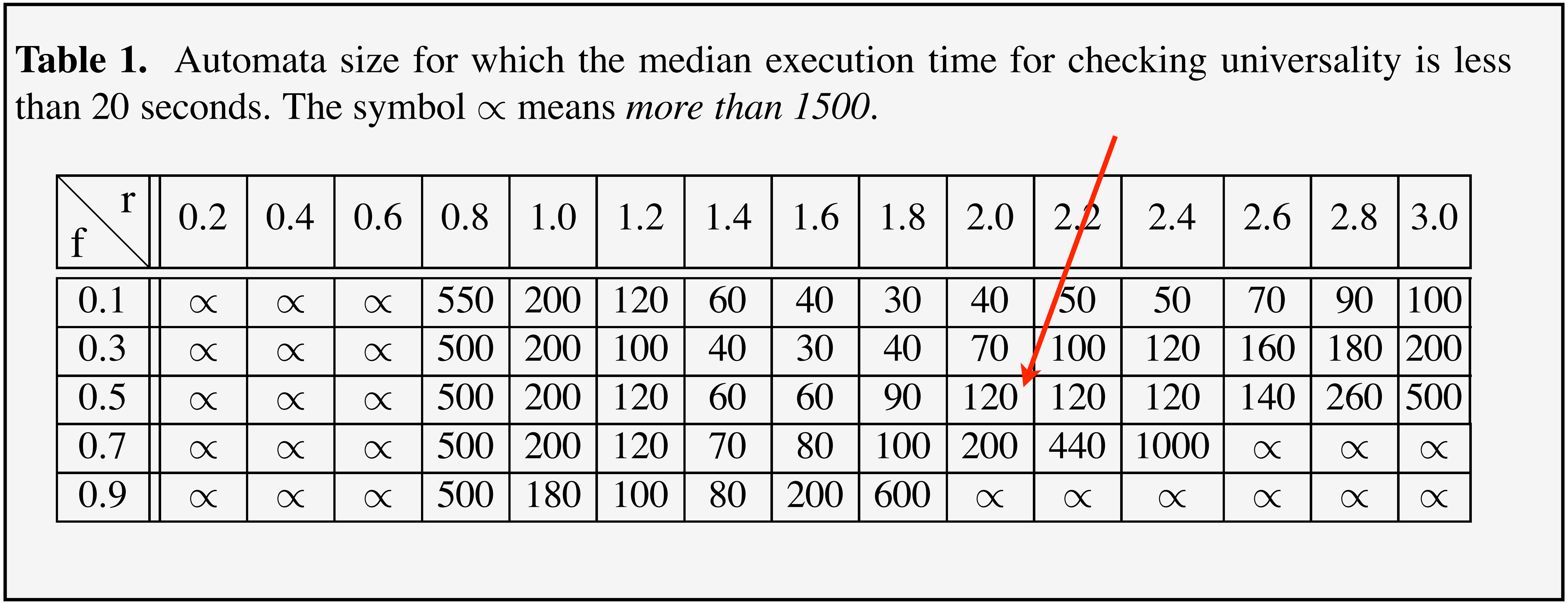


Table 1. Automata size for which the median execution time for checking universality is less than 20 seconds. The symbol ∝ means *more than 1500*.

f \ r	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
0.1	∝	∝	∝	550	200	120	60	40	30	40	50	50	70	90	100
0.3	∝	∝	∝	500	200	100	40	30	40	70	100	120	160	180	200
0.5	∝	∝	∝	500	200	120	60	60	90	120	120	120	140	260	500
0.7	∝	∝	∝	500	200	120	70	80	100	200	440	1000	∝	∝	∝
0.9	∝	∝	∝	500	180	100	80	200	600	∝	∝	∝	∝	∝	∝

For r=2, f=0.5, Tabakov can handle **8** states while our algorithm handles **120** states in less than 20s.

Practical evaluation Universality

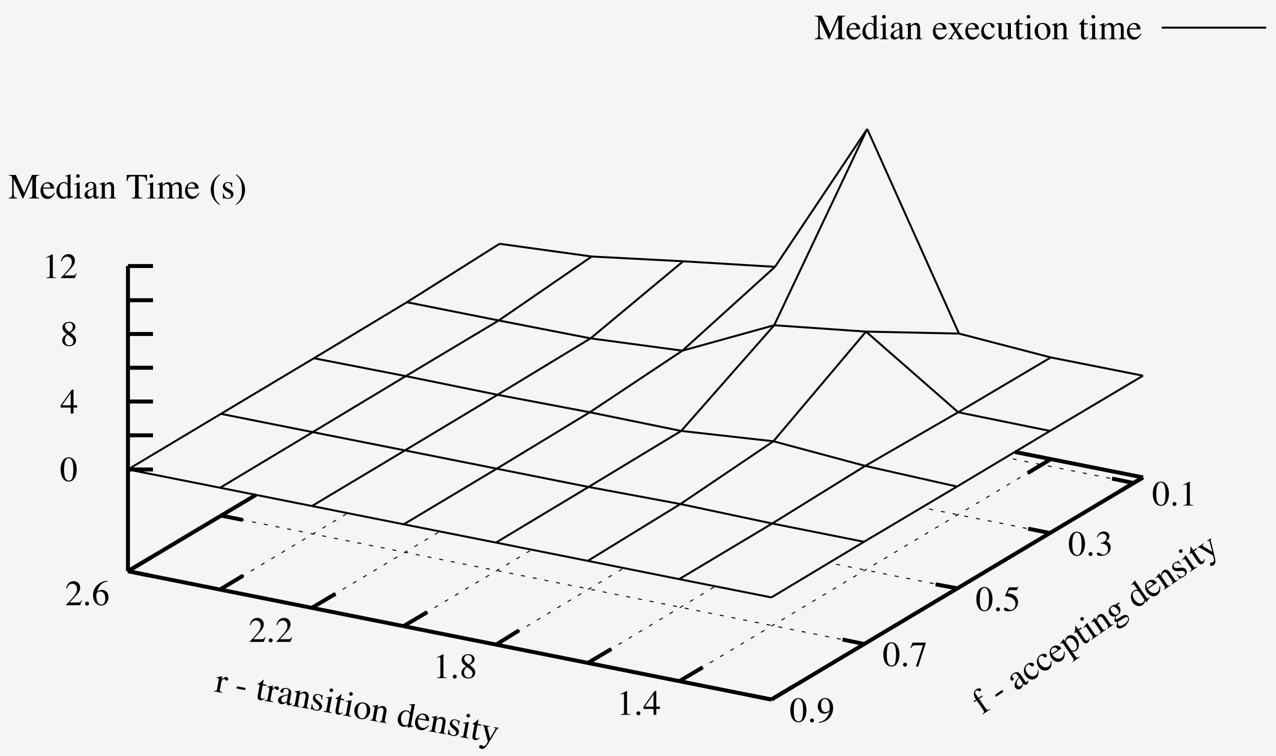


Fig. 1. Median time to check universality of 100 automata of size 30 for each sample point.

To compare, Tabakov's BDD implementation was able to handle automata of size **6** on the entire state space (within 20s as in our experiments).

Game with imperfect information
Strategy construction

- We have shown how to compute **efficiently** using antichains the set of winning cells of a game with imperfect information.

  Is it possible to extract an observation-based winning strategy from this computation ?

- The answer is **yes** for all parity objectives. Nevertheless it may be costly, i.e. *there are games for which the strategy is exponentially larger than the size of the antichain representation of the fixed point.*

- It is easy for safety objectives, more intricate for reachability and for parity objectives. We concentrate here on safety and refer to [BCD+08] for the other cases.

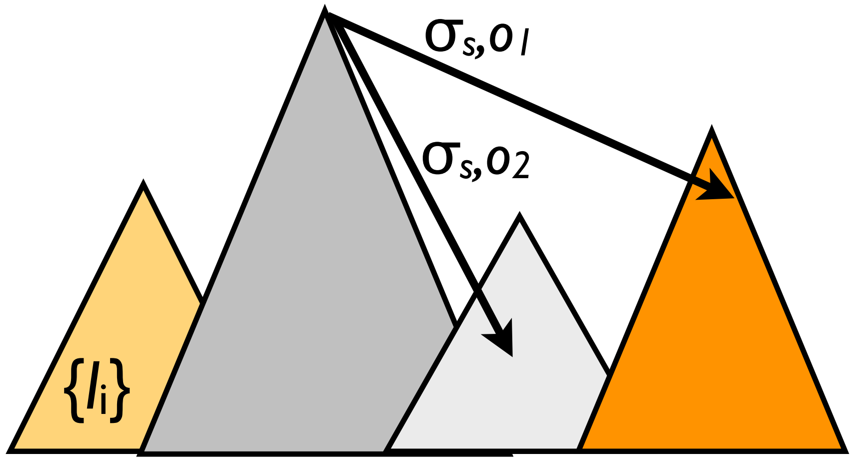
Game with imperfect information
Strategy construction

- Let $W = q_{win}{\downarrow}$ be the set of winning cells for a safety games with imperfect information that are compactly represented by the antichain q_{win}.

- By definition of the fixed point equation, we know that

$$\forall s \in q_{win} \cdot \exists \sigma_s \in \Sigma \cdot \forall o \in Obs \cdot \exists s' \in q_{win} \cdot \mathbf{post}_{G,\sigma}(s) \cap o \subseteq s'$$

It is easy to see that the strategy that plays the action σ_s in any cell $s'' \subseteq s$ is surely-winning.



Game with imperfect information Strategy construction - Safety

- Let $W=q_{win}\downarrow$ be the set of winning cells.
- We construct a Moore machine $(M,m_i,update,\mu)$ where:

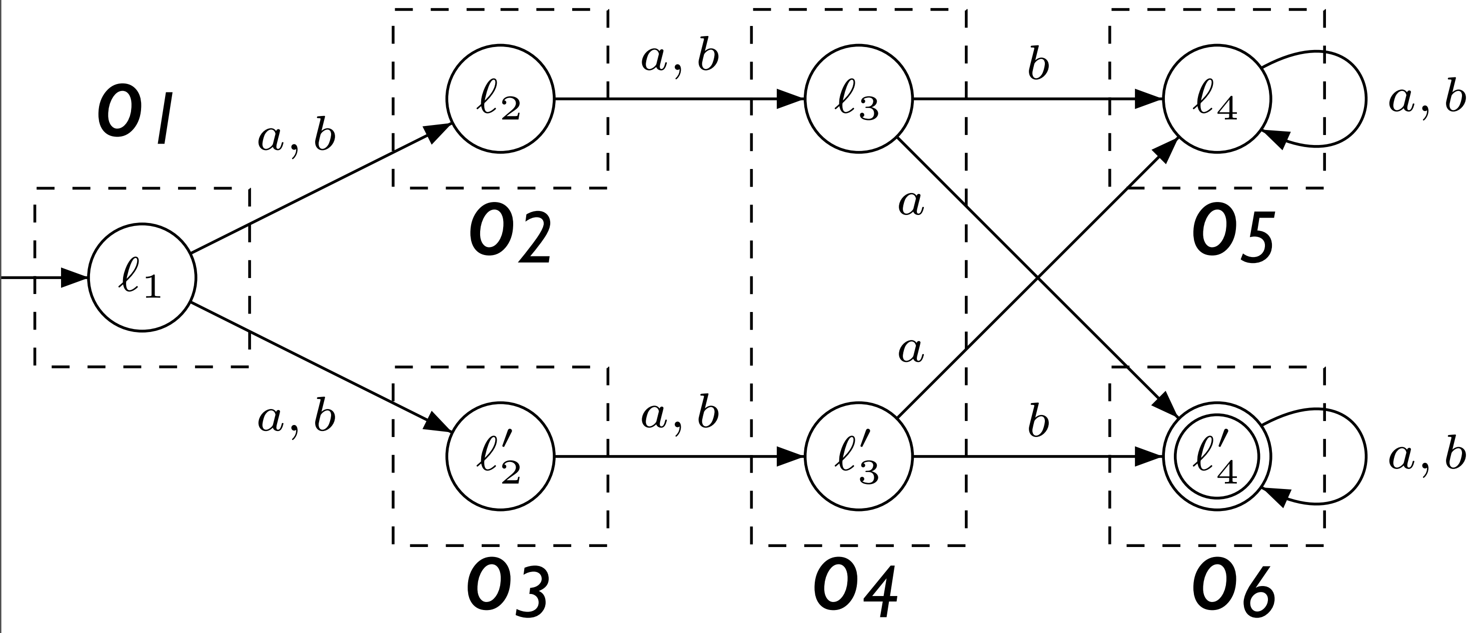
(*i*) $M=q_{win}$

(*ii*) $m_i=s$ for some $s\in q_{win}$ such that $s_i\subseteq s$.

(iii) $update{:}M\times Obs\rightarrow M$ such that
$update(s,o)=s'$ for some s' such that $\mathbf{post}_{G,\sigma s}(s)\cap o\subseteq s'$.

(iv) $\mu(s)=\sigma_s$.

Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$

post$_{G,a}(\{l_1,l_2',l_3',l_4\})=\{l_2,l_2',l_3',l_4\}$

$\{l_2,l_2',l_3',l_4\}\cap o_1=\{l_1\}\subseteq\{l_1,l_2',l_3',l_4\}$

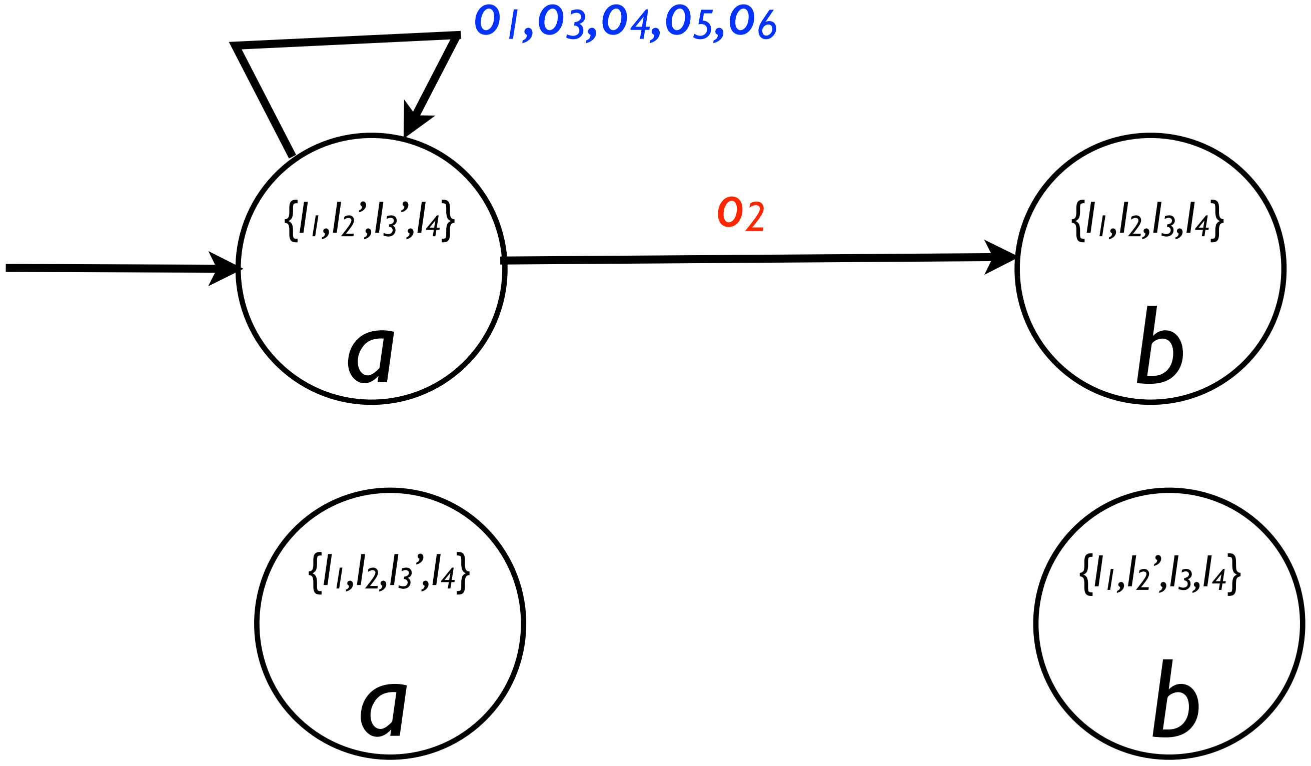
$\{l_2,l_2',l_3',l_4\}\cap o_2=\{l_2\}\subseteq\{l_1,l_2,l_3,l_4\}$

$\{l_2,l_2',l_3',l_4\}\cap o_3=\{l_2'\}\subseteq\{l_1,l_2',l_3',l_4\}$

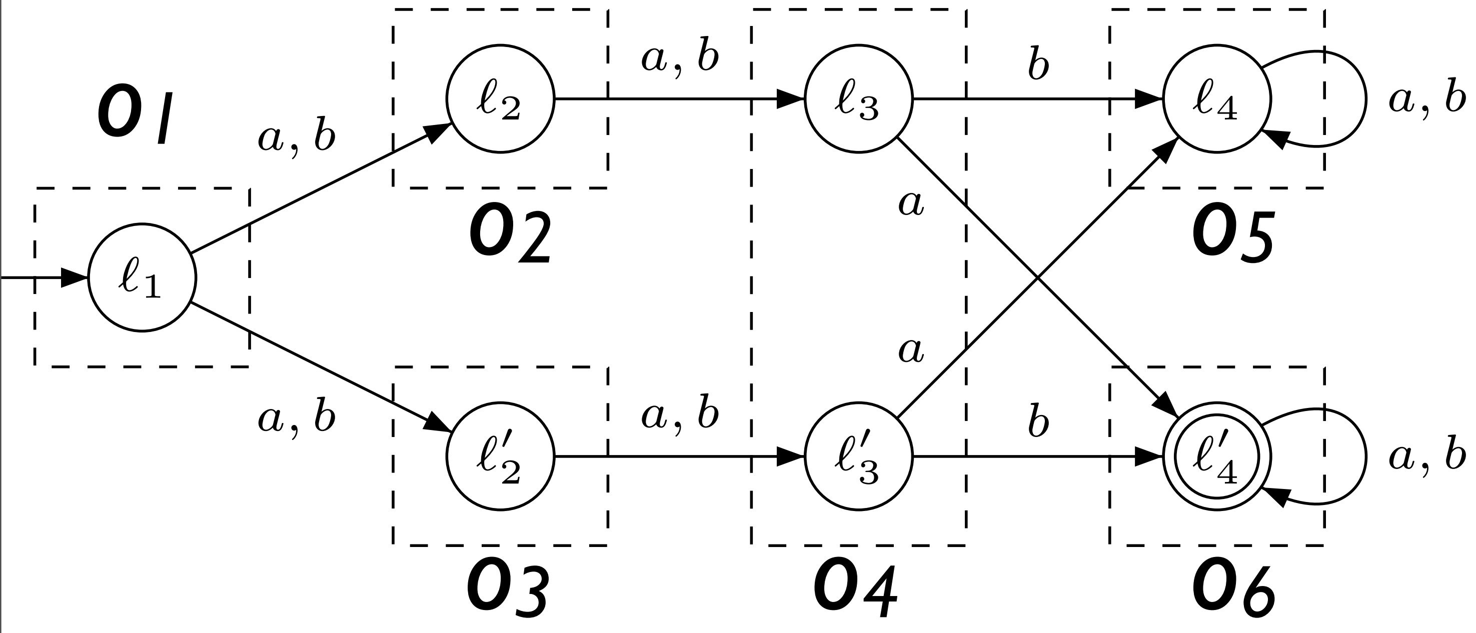
$\{l_2,l_2',l_3',l_4\}\cap o_4=\{l_3'\}\subseteq\{l_1,l_2',l_3',l_4\}$

$\{l_2,l_2',l_3',l_4\}\cap o_5=\{l_4\}\subseteq\{l_1,l_2',l_3',l_4\}$

$\{l_2,l_2',l_3',l_4\}\cap o_6=\varnothing\subseteq\{l_1,l_2',l_3',l_4\}$



Game with imperfect information
Strategy construction - Safety



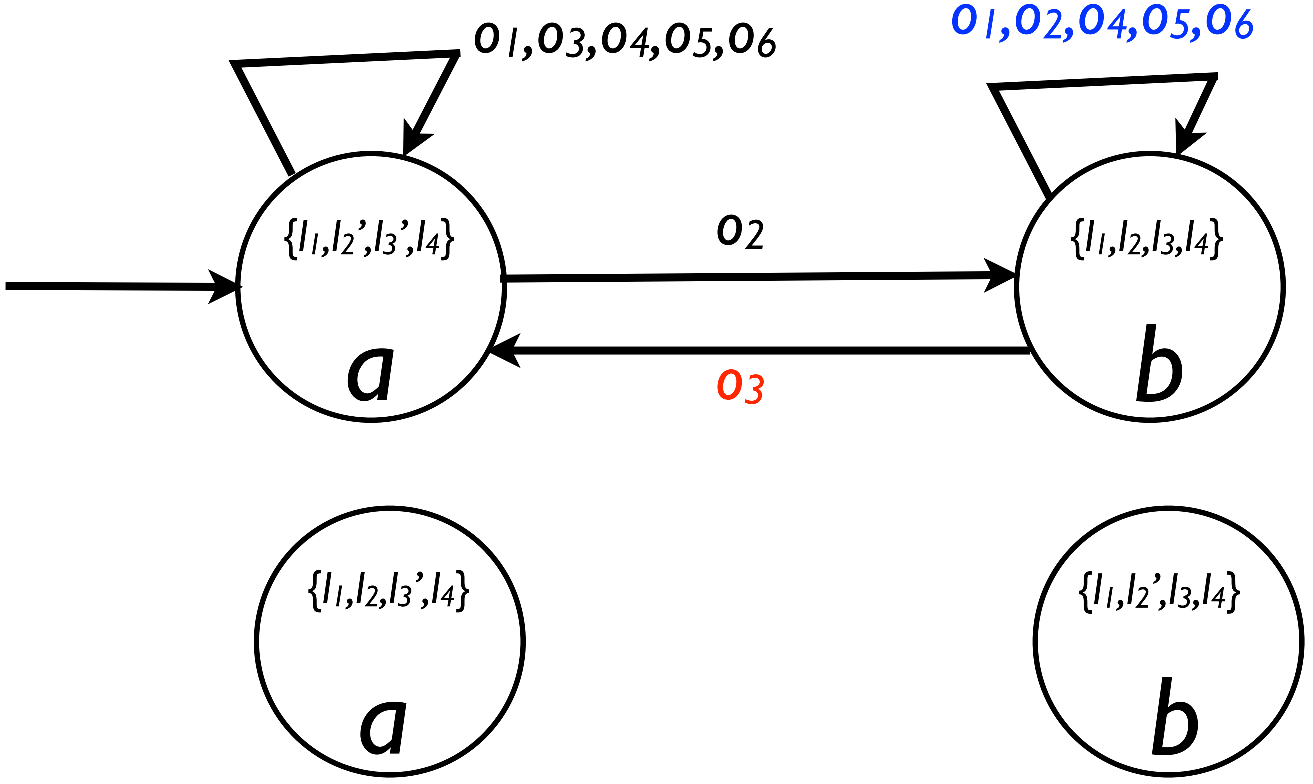
Safety objective **Safe**($L\setminus\{o_6\}$).

q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

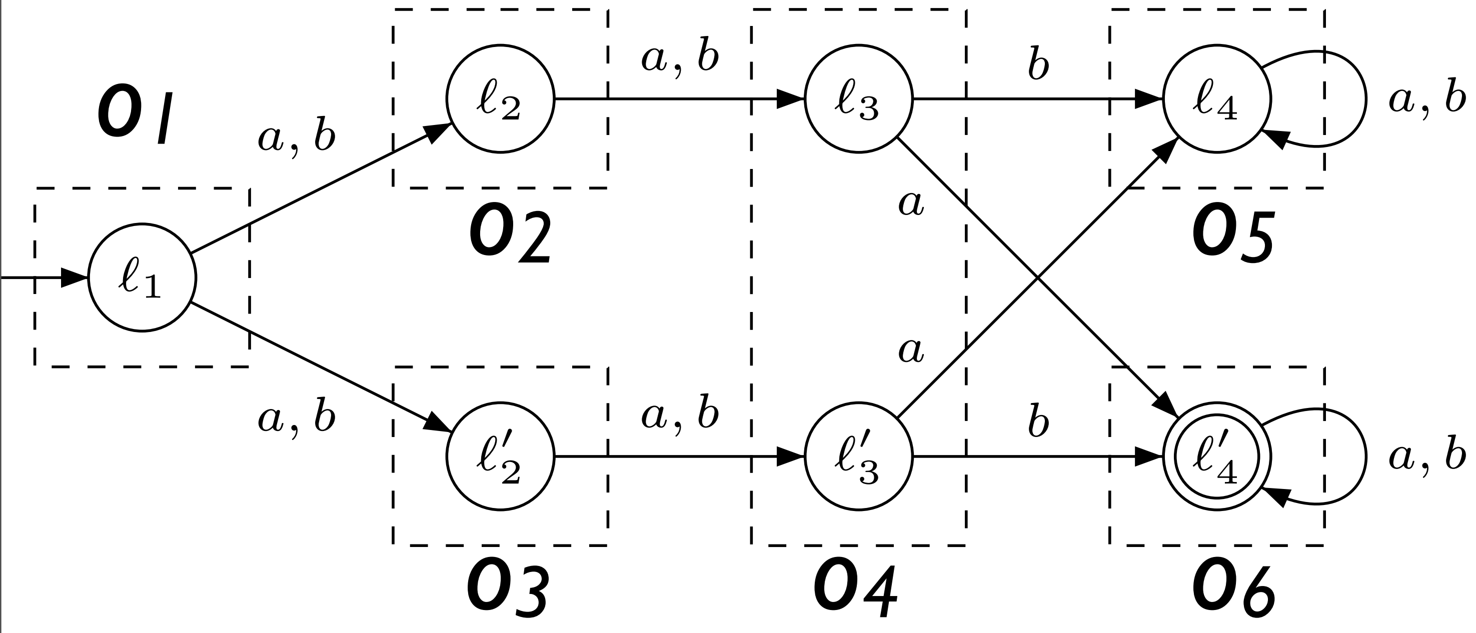
$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$

post$_{G,a}(\{l_1,l_2,l_3,l_4\})=\{l_2,l_2',l_3,l_4\}$

$\{l_2,l_2',l_3,l_4\}\cap o_1=\varnothing\subseteq\{l_1,l_2,l_3,l_4\}$
$\{l_2,l_2',l_3,l_4\}\cap o_2=\{l_2\}\subseteq\{l_1,l_2,l_3,l_4\}$
$\{l_2,l_2',l_3,l_4\}\cap o_3=\{l_2'\}\subseteq\{l_1,l_2',l_3',l_4\}$
$\{l_2,l_2',l_3,l_4\}\cap o_4=\{l_3\}\subseteq\{l_1,l_2,l_3,l_4\}$
$\{l_2,l_2',l_3,l_4\}\cap o_5=\{l_4\}\subseteq\{l_1,l_2,l_3,l_4\}$
$\{l_2,l_2',l_3,l_4\}\cap o_6=\varnothing\subseteq\{l_1,l_2,l_3,l_4\}$



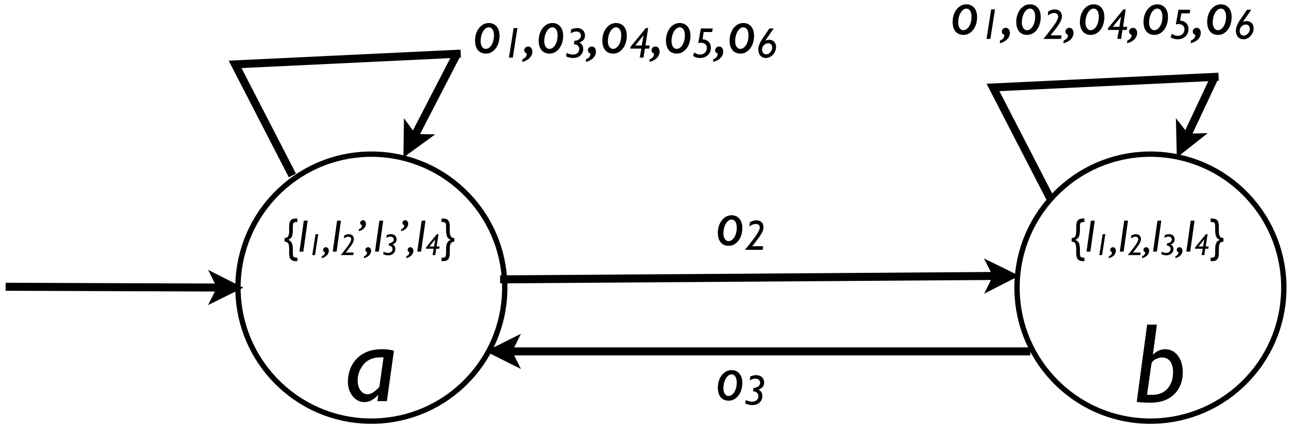
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

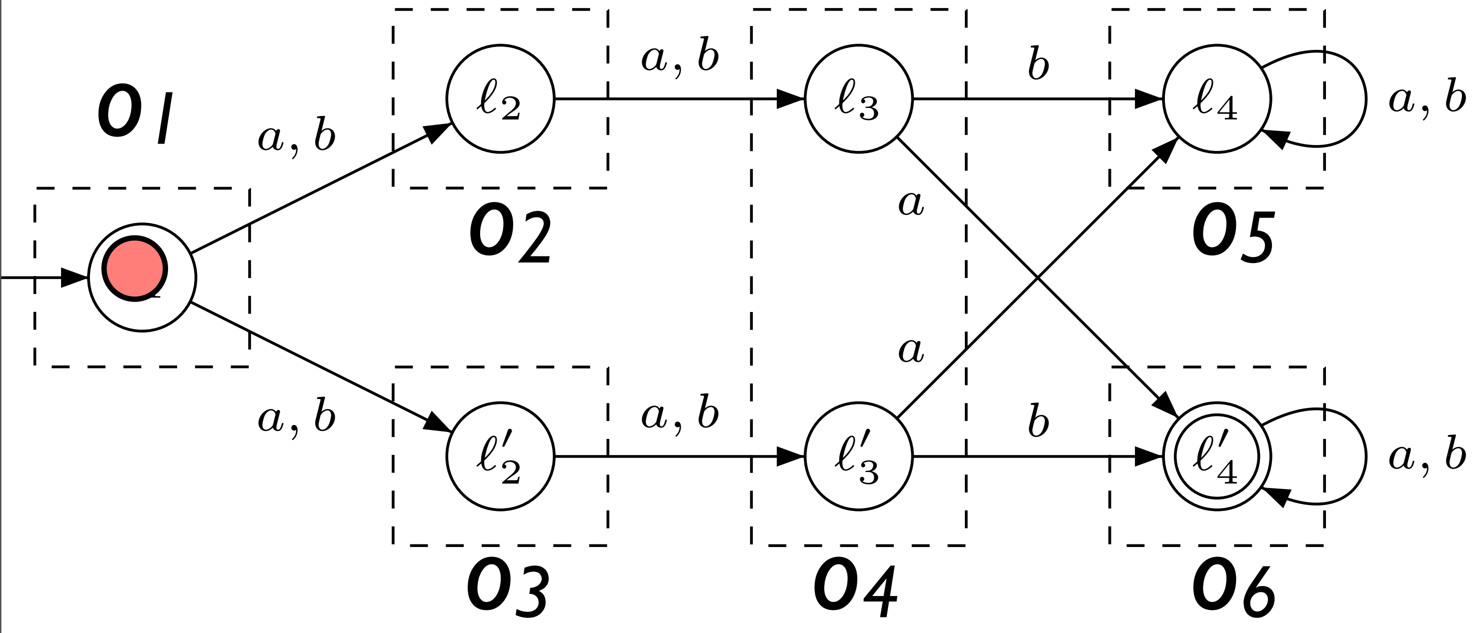
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

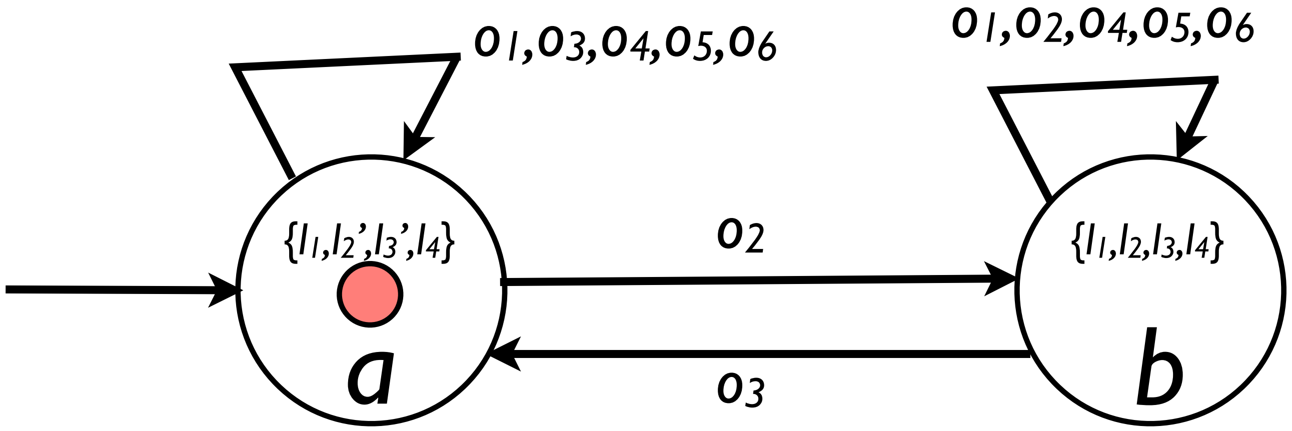
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**(*L*\{*o*₆}).

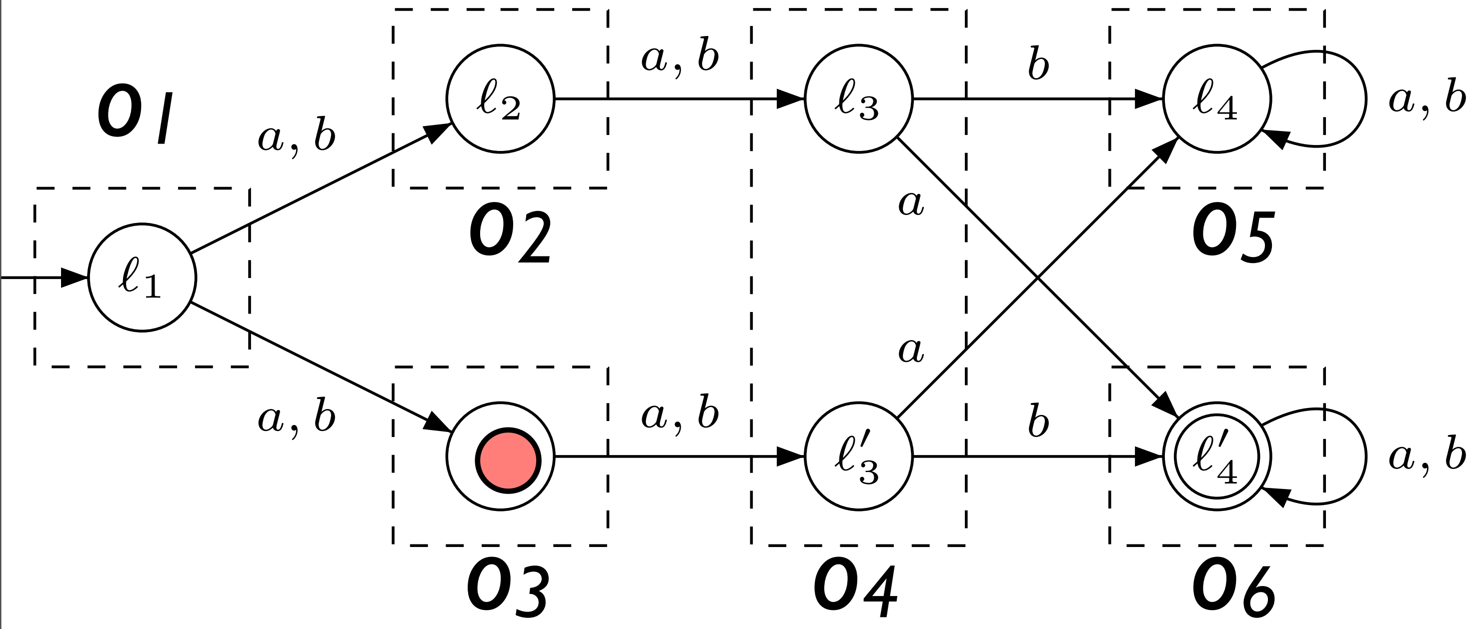
q$_{win}$
={L\{l₂,l₃,l₄'},L\{l₂',l₃',l₄'},L\{l₂',l₃,l₄'},L\{l₂,l₃',l₄'}}

={{l₁,l₂',l₃',l₄},{l₁,l₂,l₃,l₄},{l₁,l₂,l₃',l₄},{l₁,l₂',l₃,l₄}}



Let us play the strategy

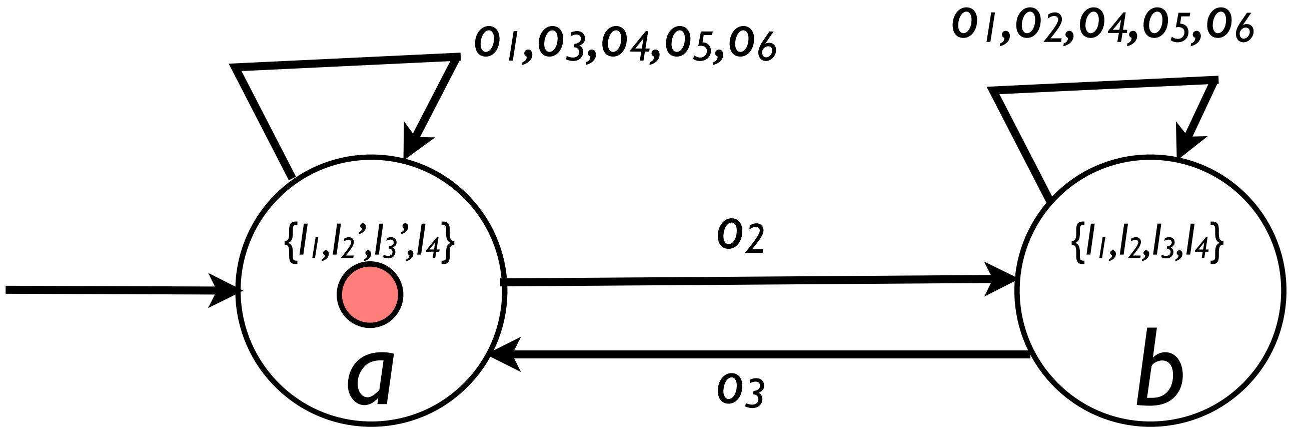
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

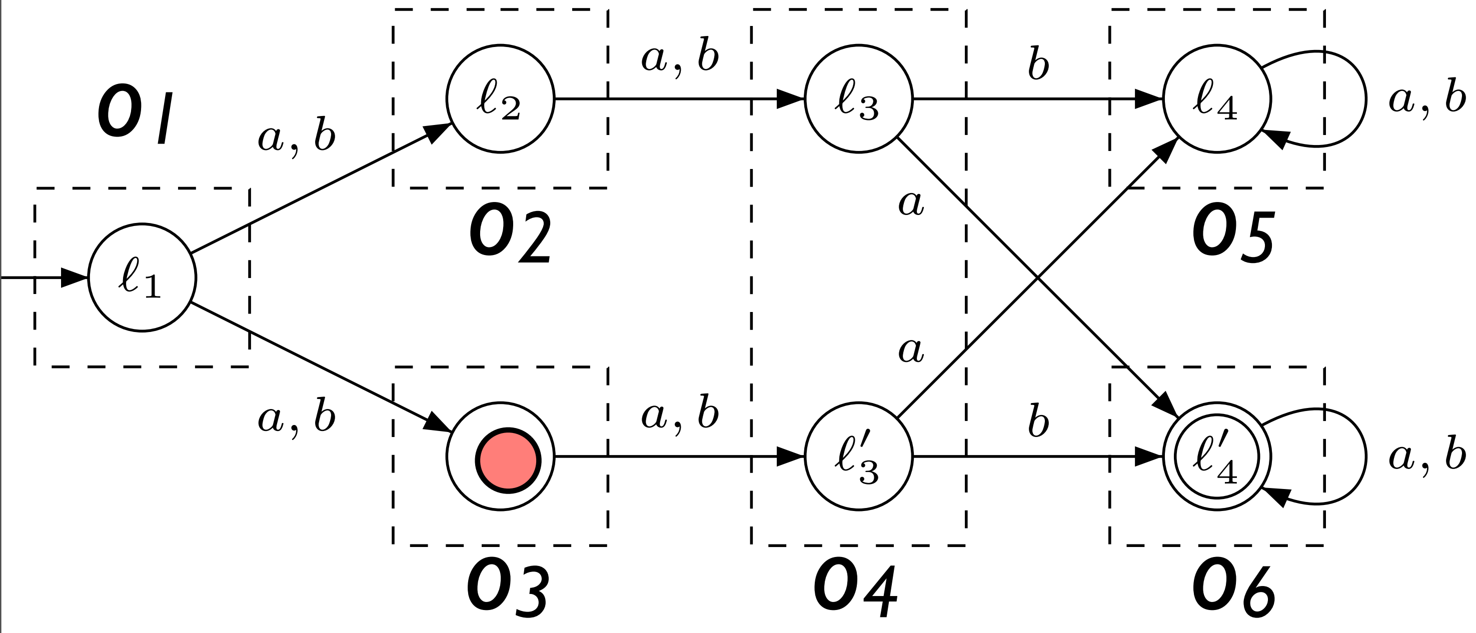
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

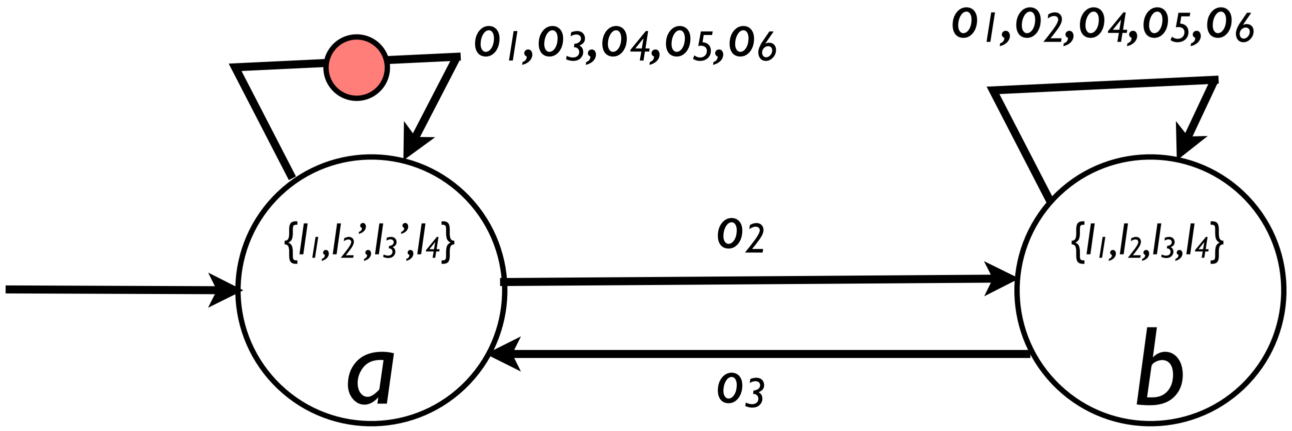
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

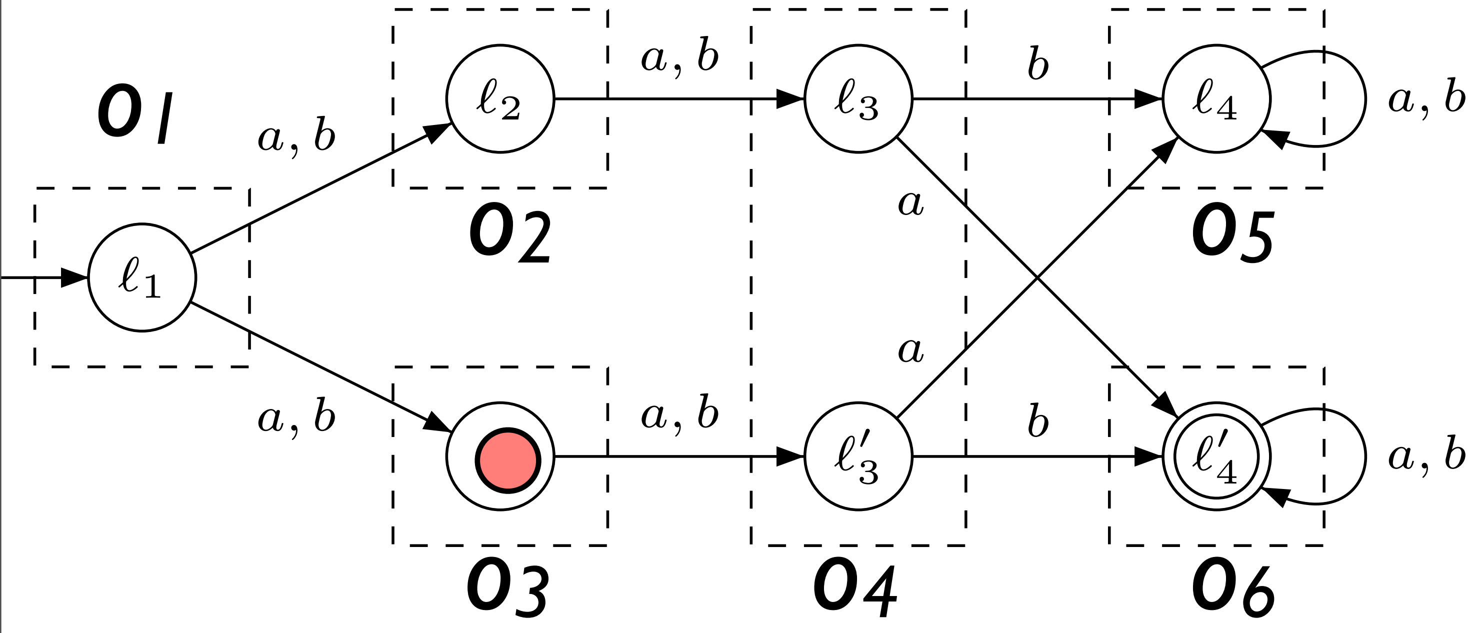
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

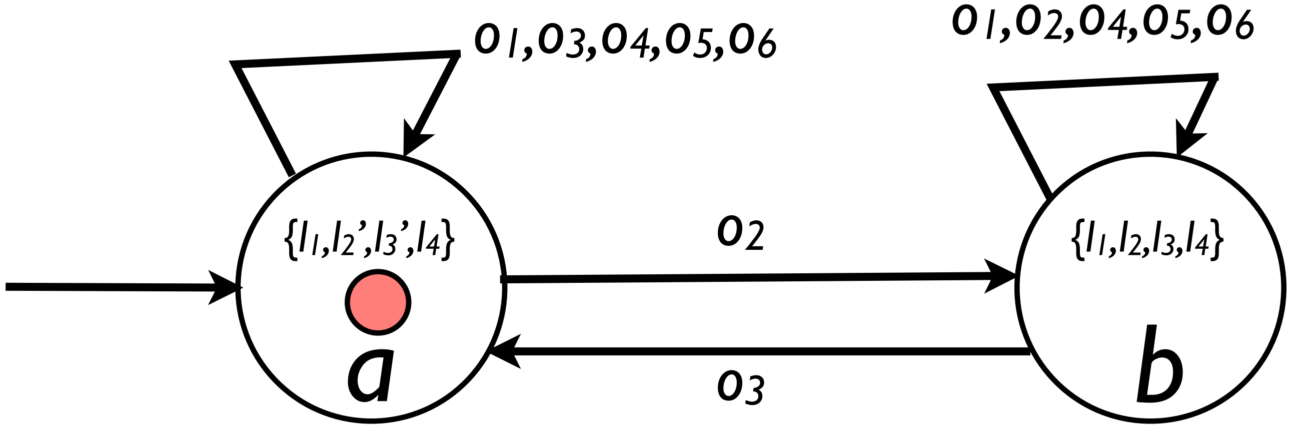
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

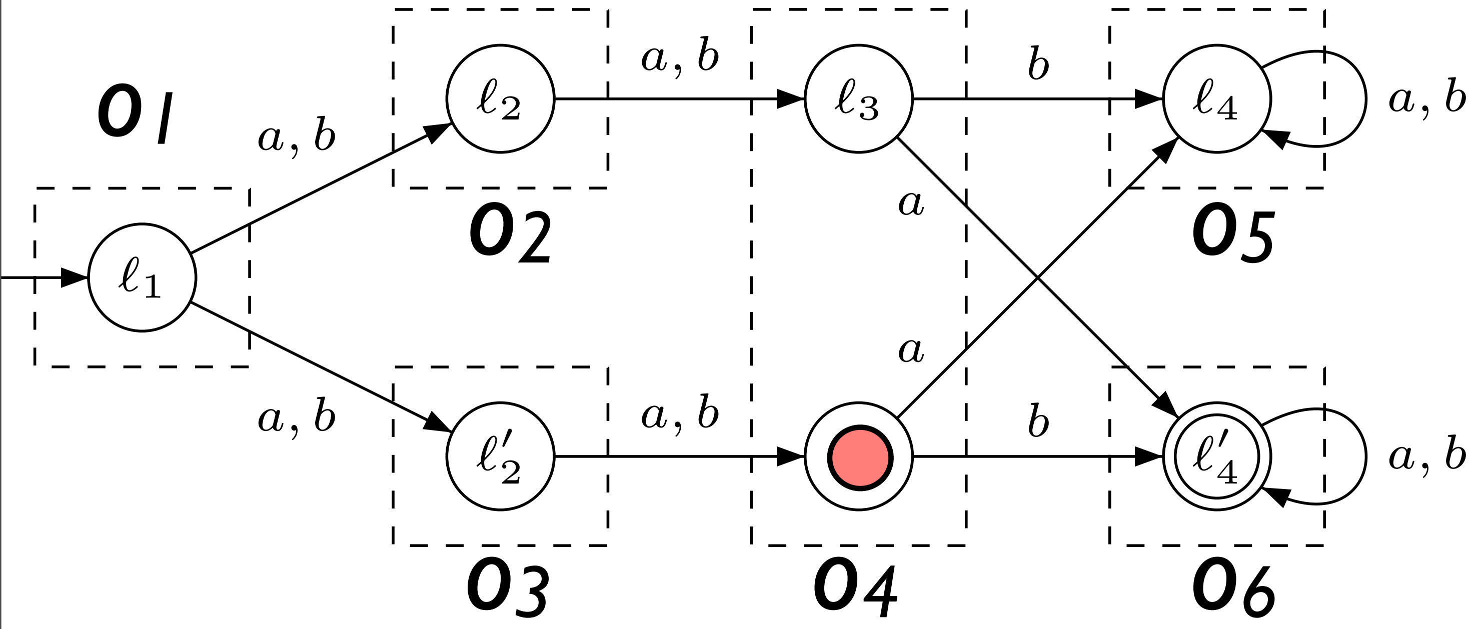
q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

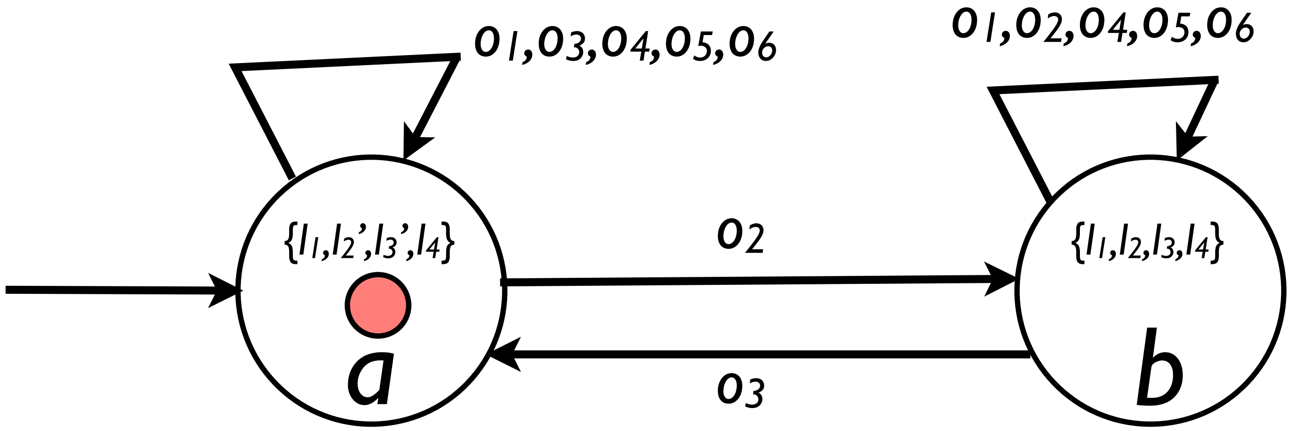
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L \backslash \{o_6\}$).

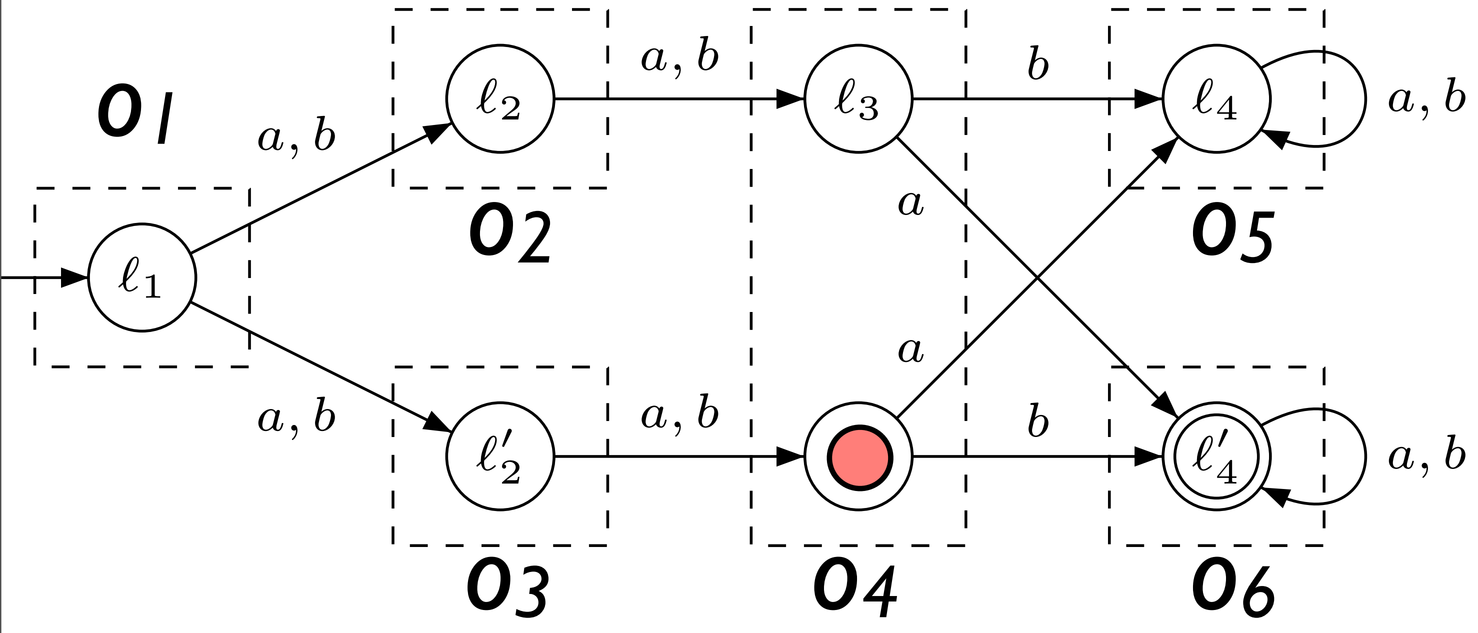
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

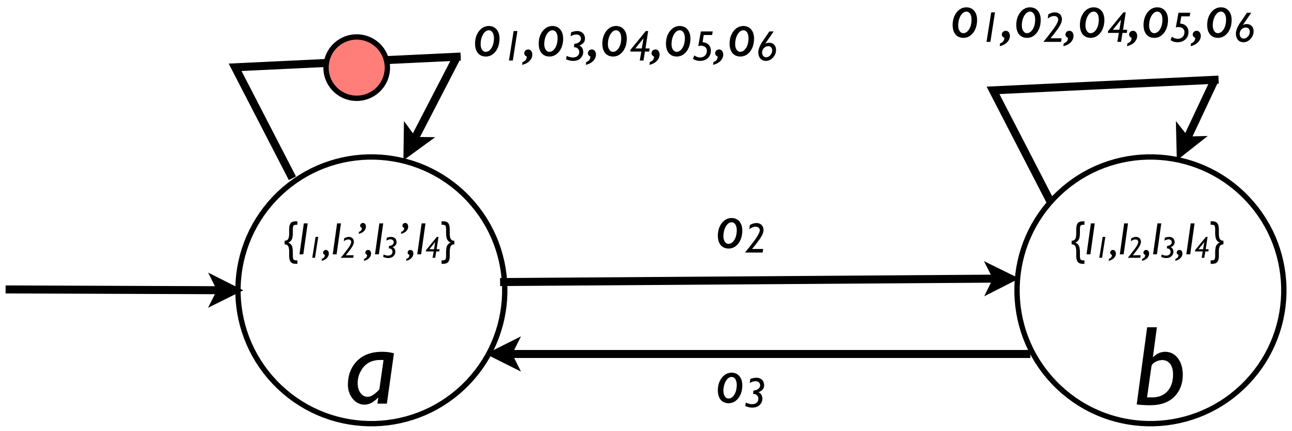
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

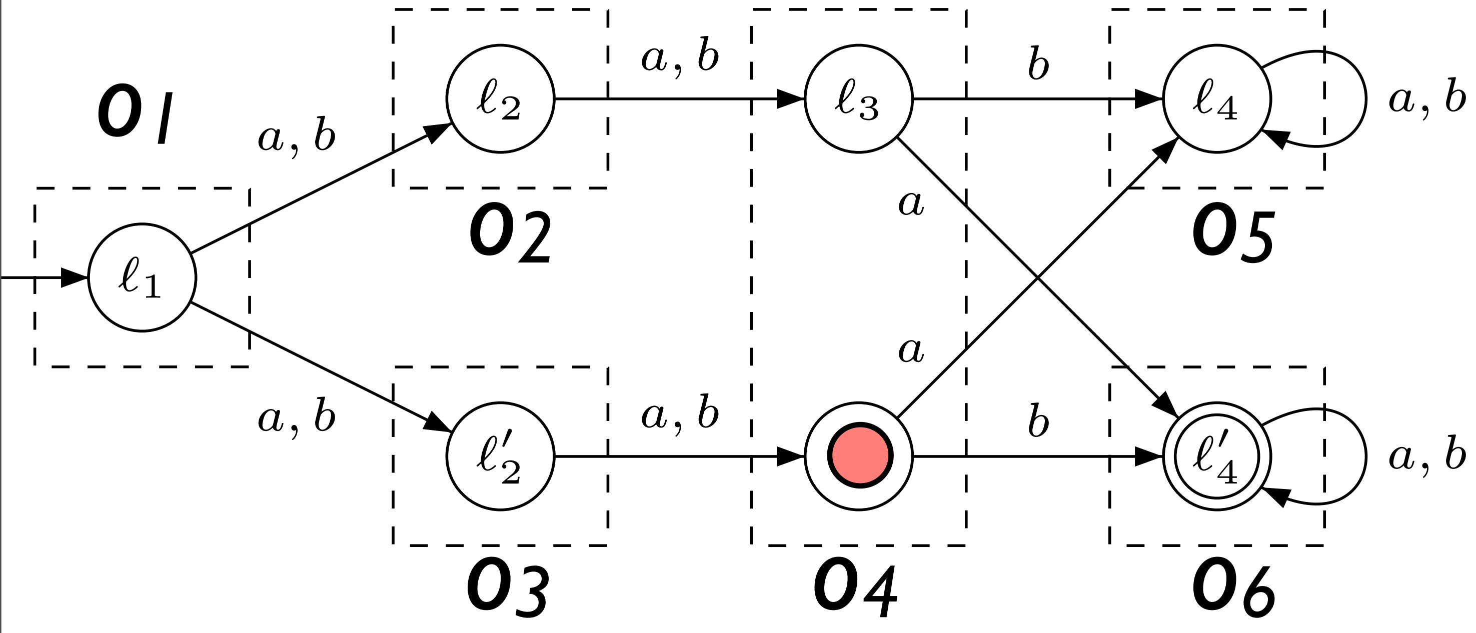
q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

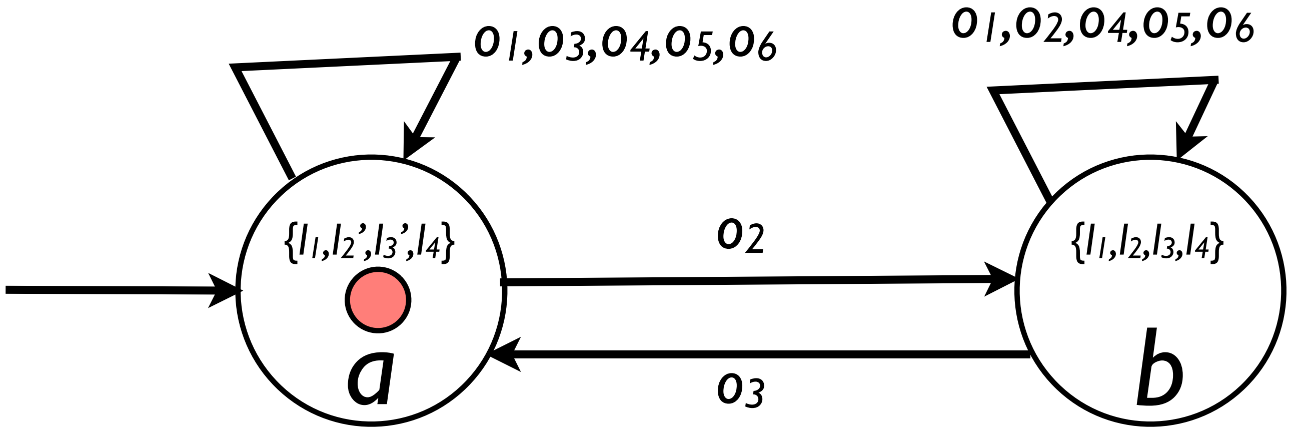
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

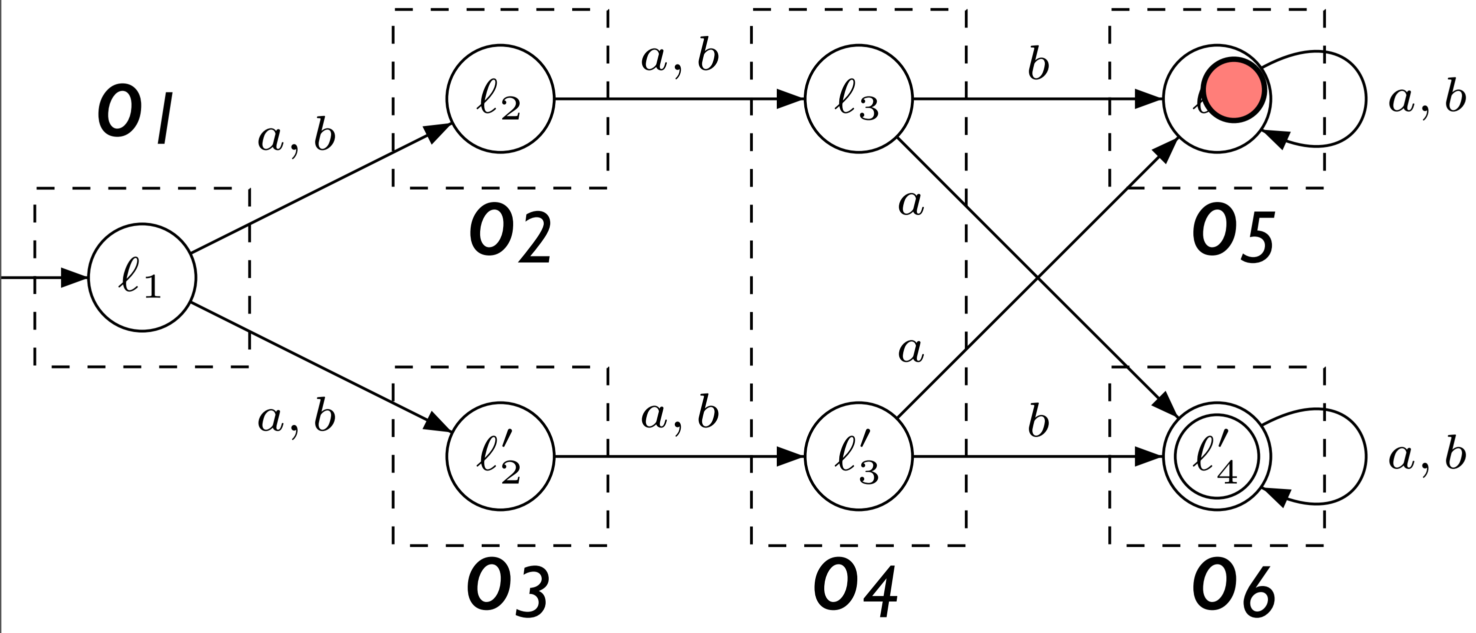
q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

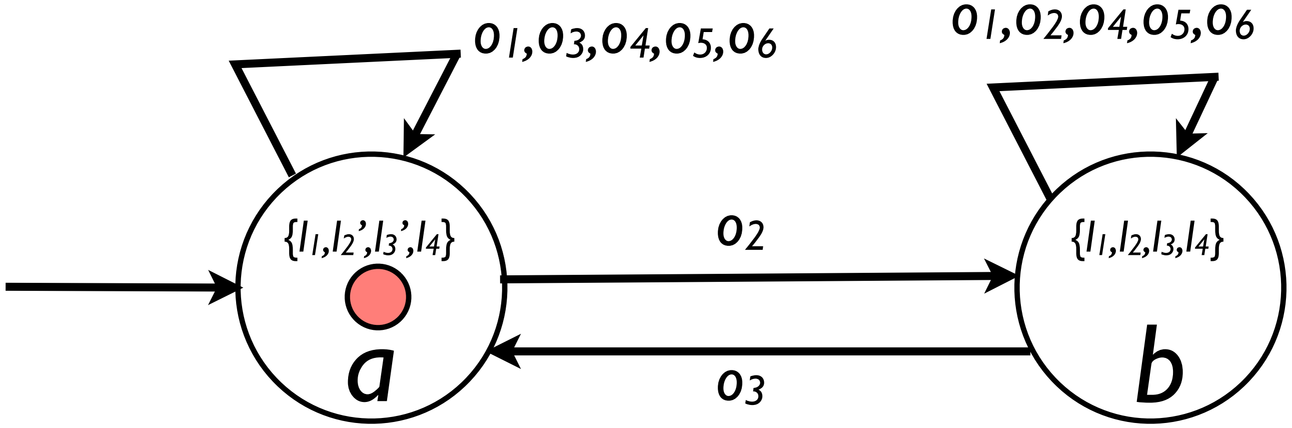
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

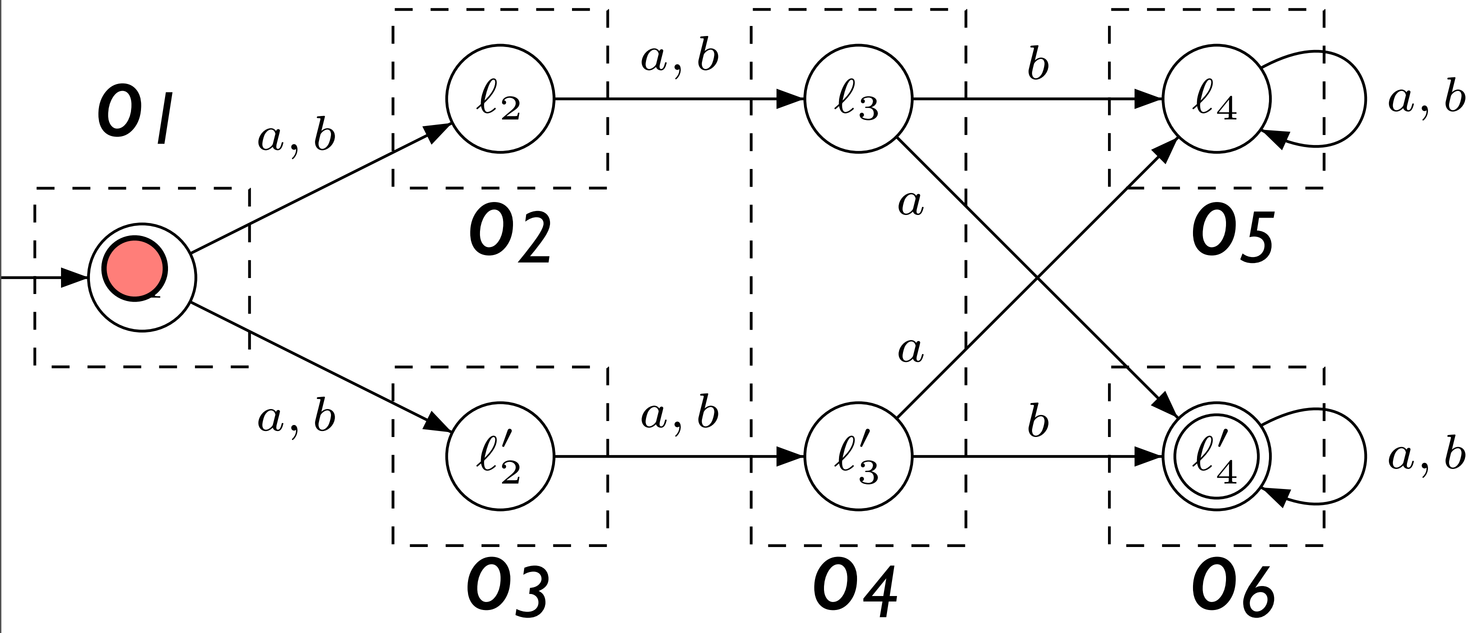
$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Player I wins

us play the strategy

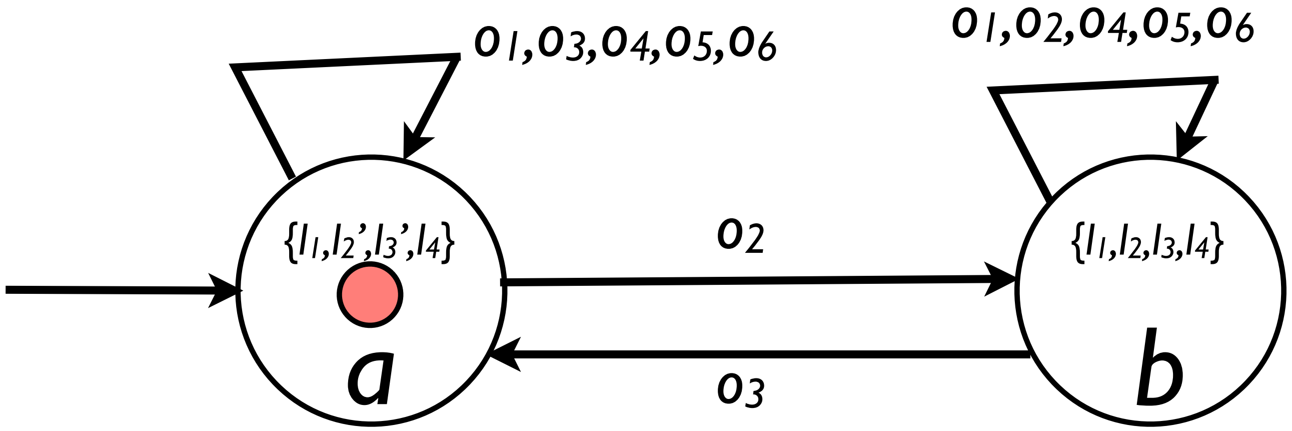
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

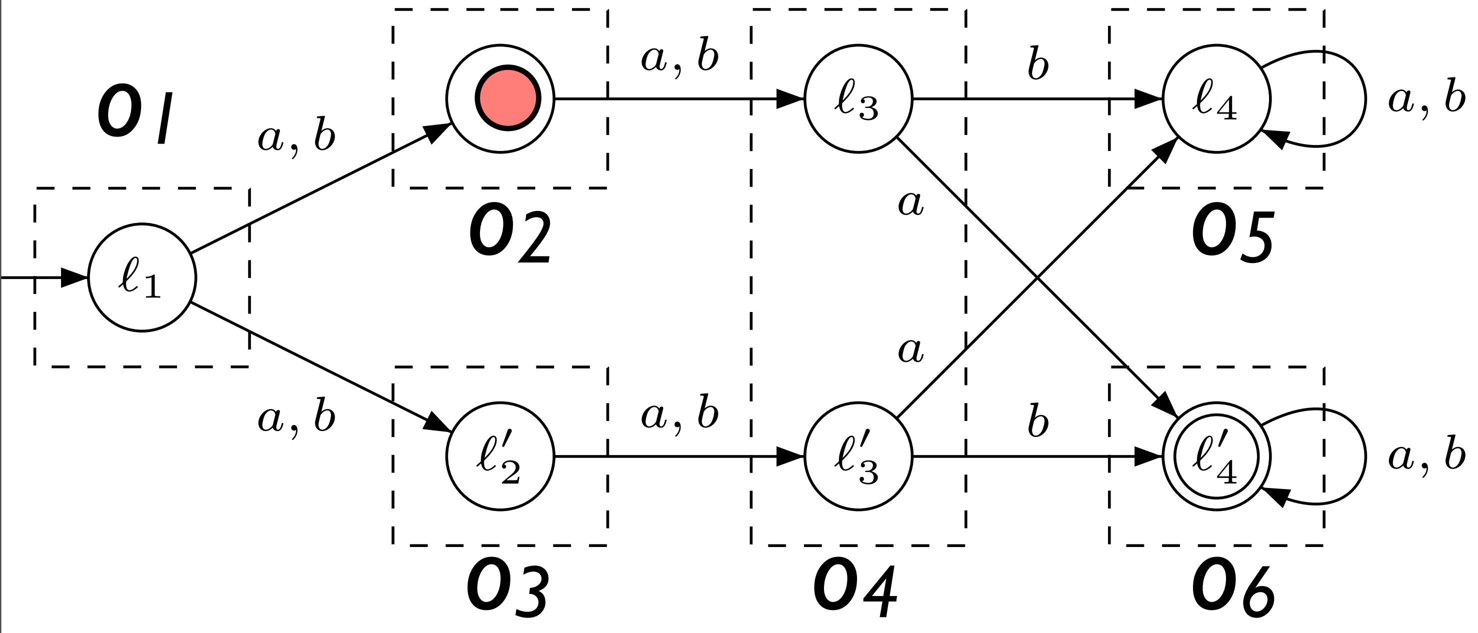
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

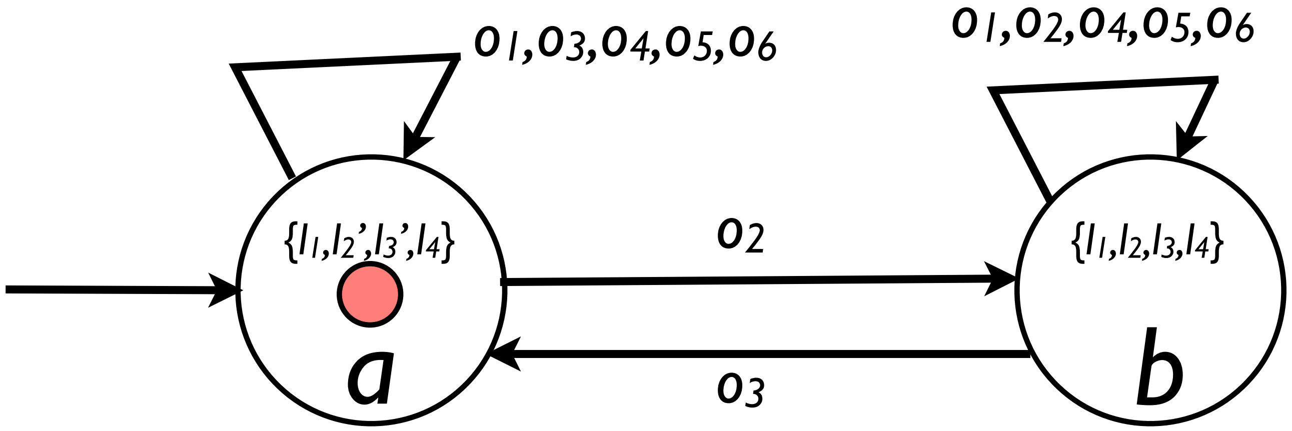
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

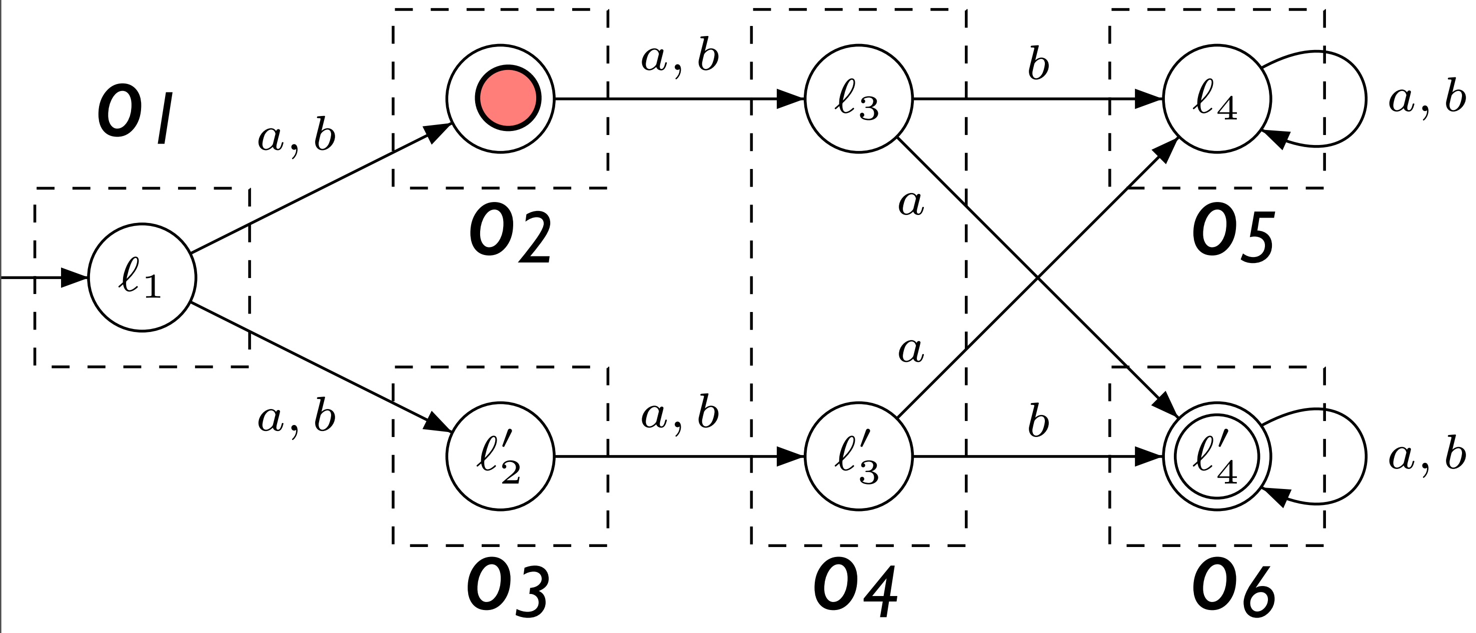
q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

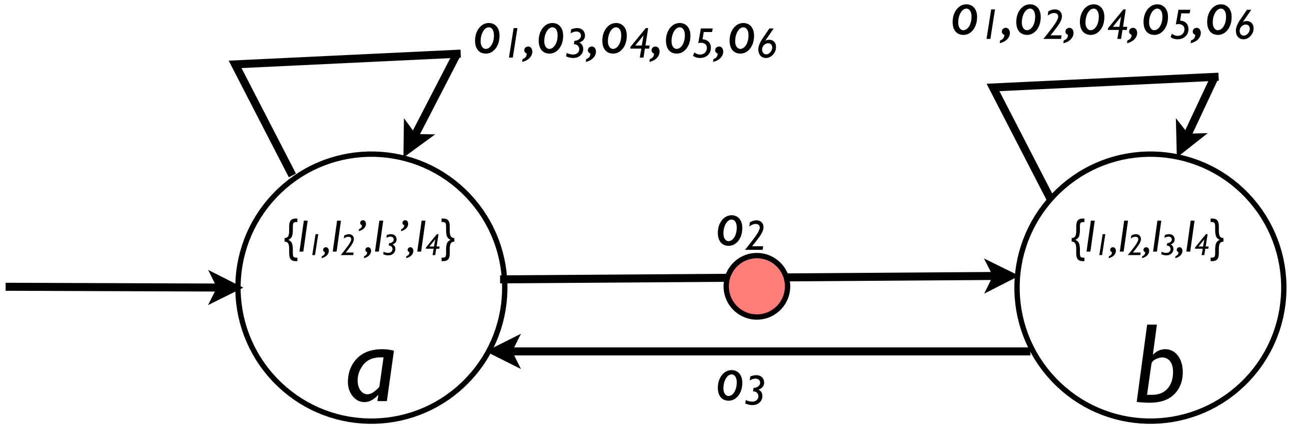
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

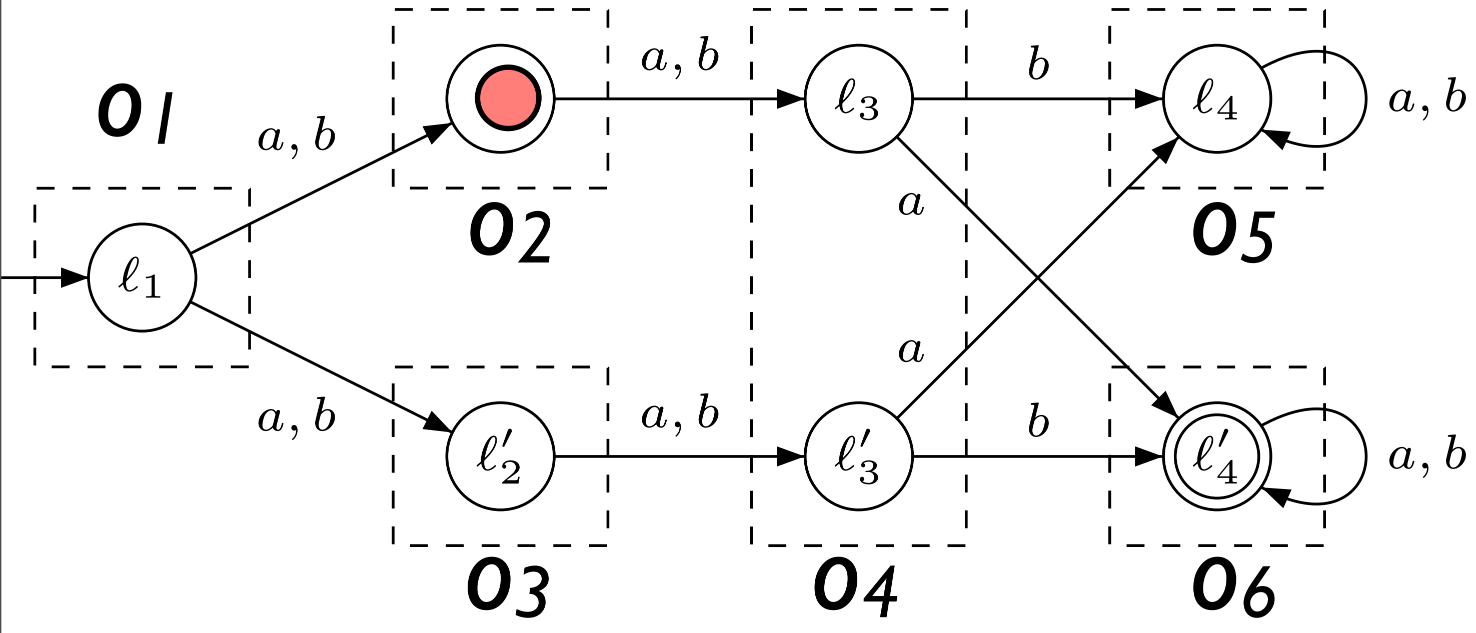
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

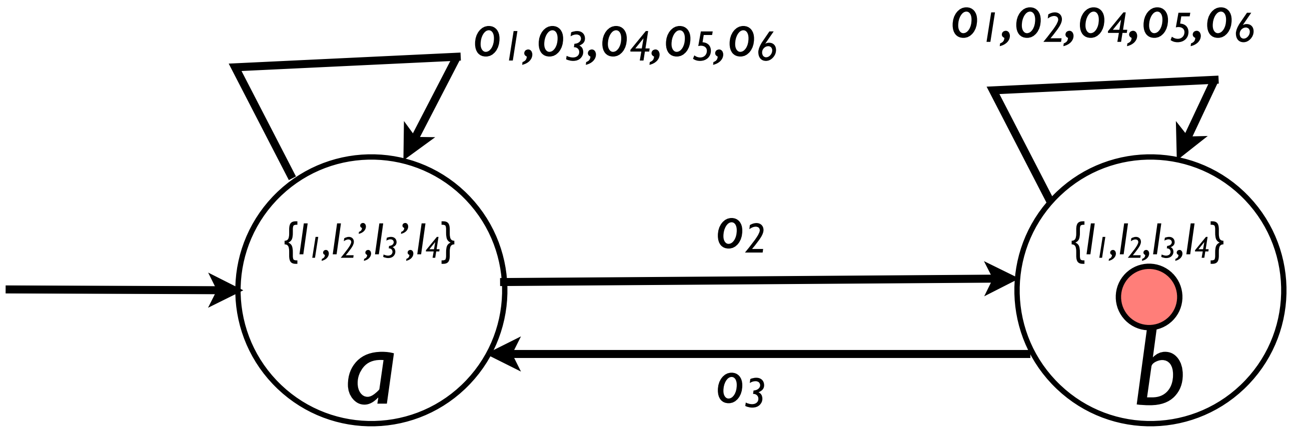
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

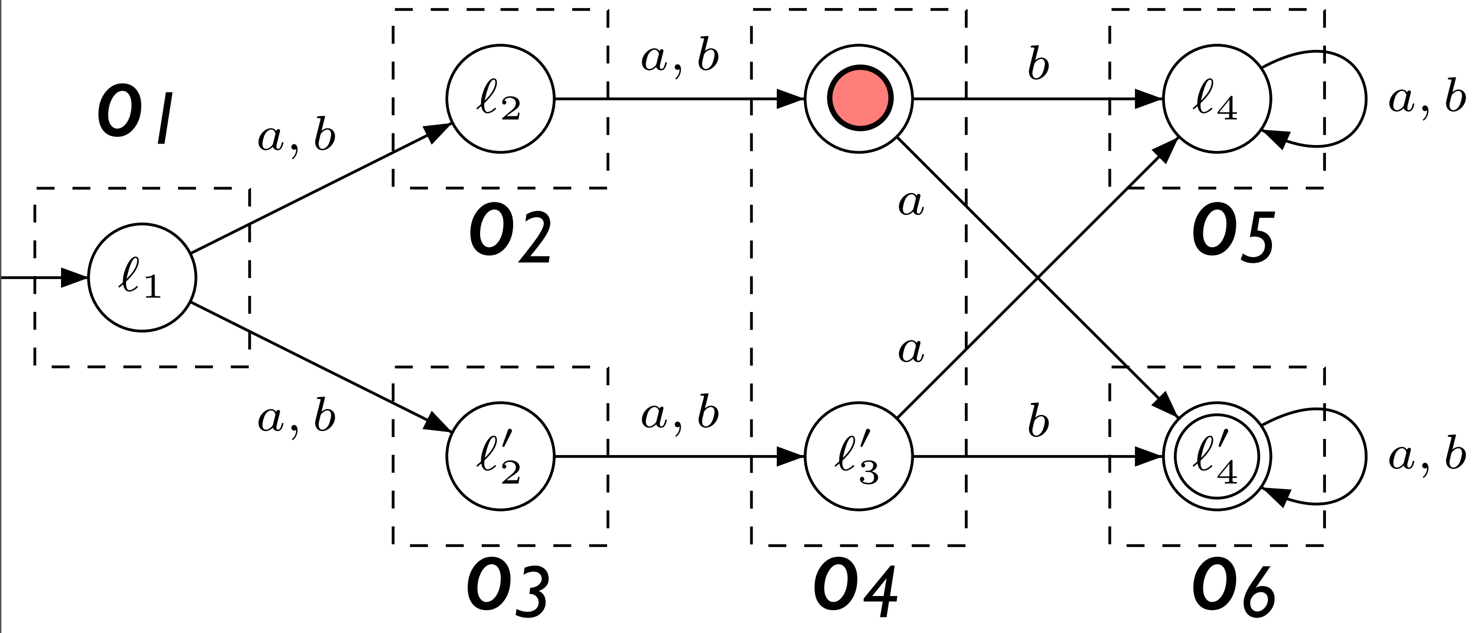
q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

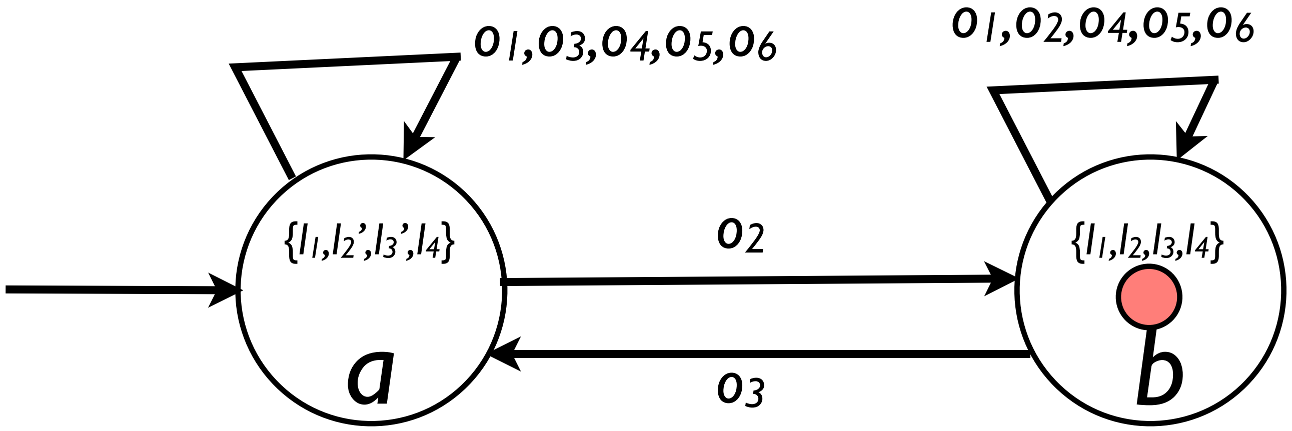
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

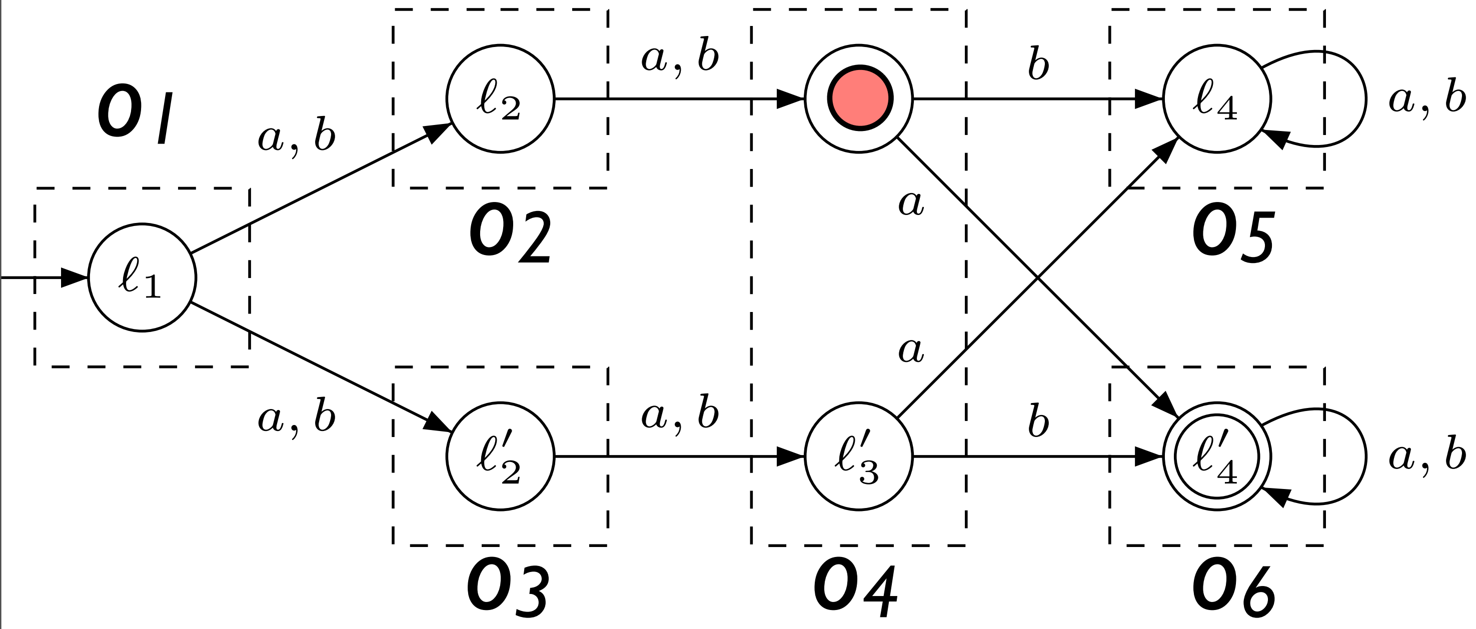
q_{win}
$=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

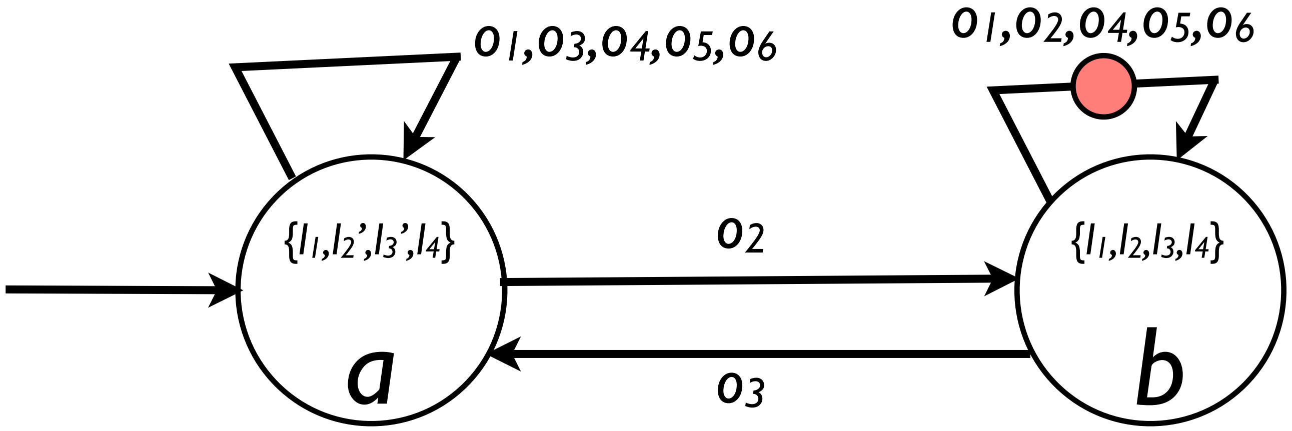
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

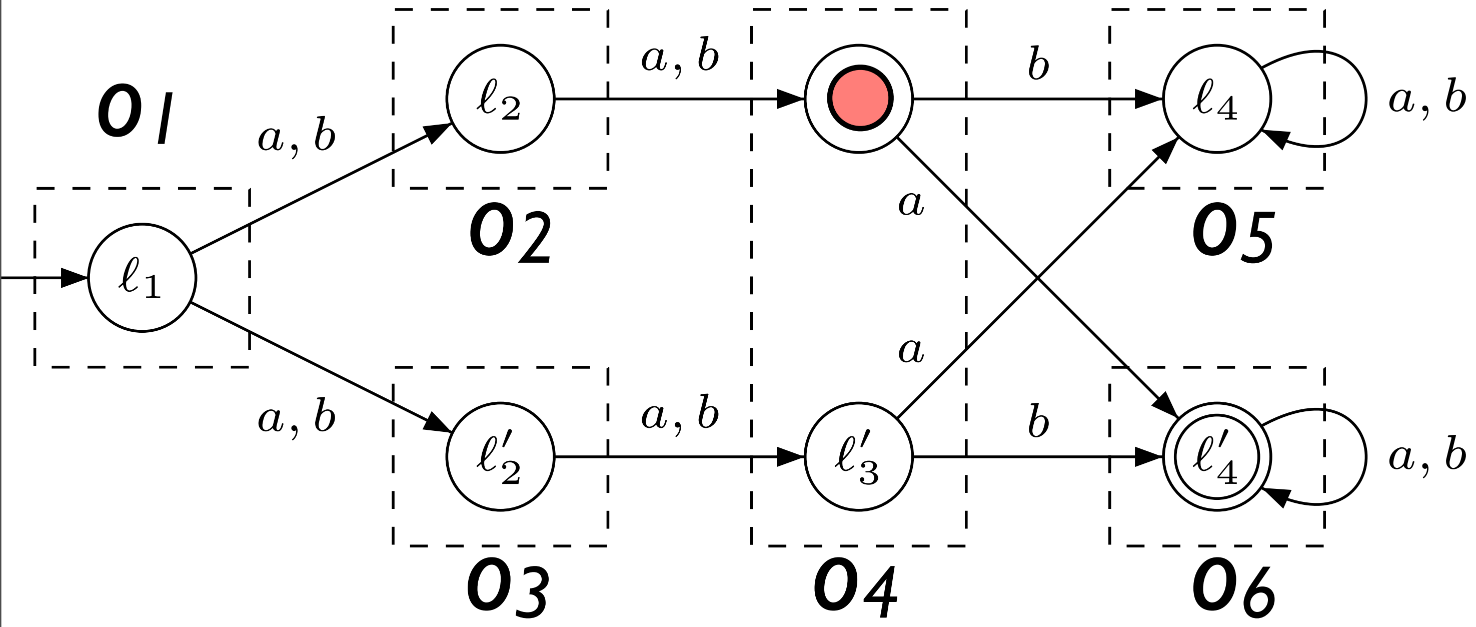
q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

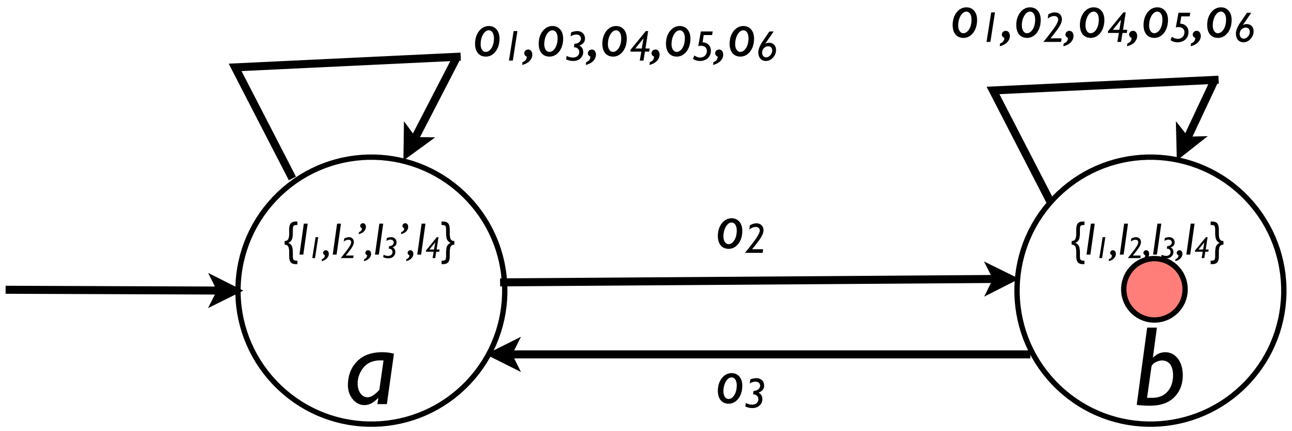
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\backslash\{o_6\}$).

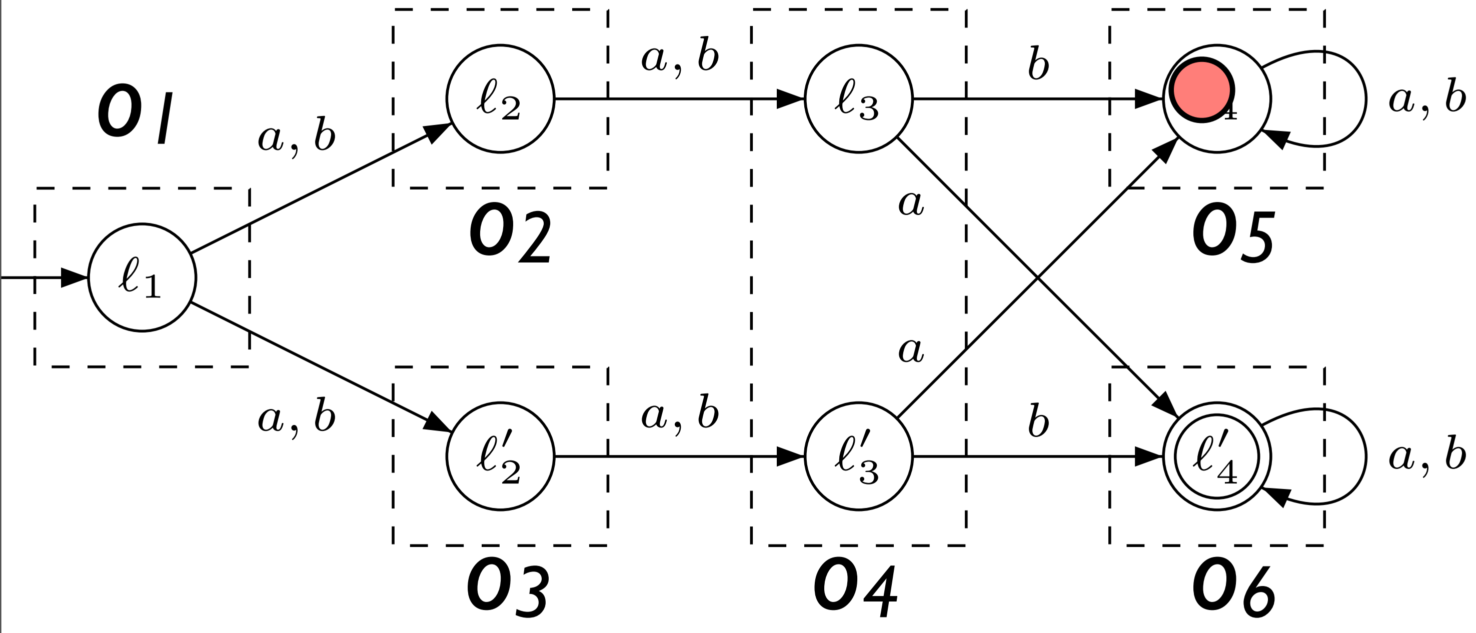
q_{win} $=\{L\backslash\{l_2,l_3,l_4'\},L\backslash\{l_2',l_3',l_4'\},L\backslash\{l_2',l_3,l_4'\},L\backslash\{l_2,l_3',l_4'\}\}$

$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Let us play the strategy

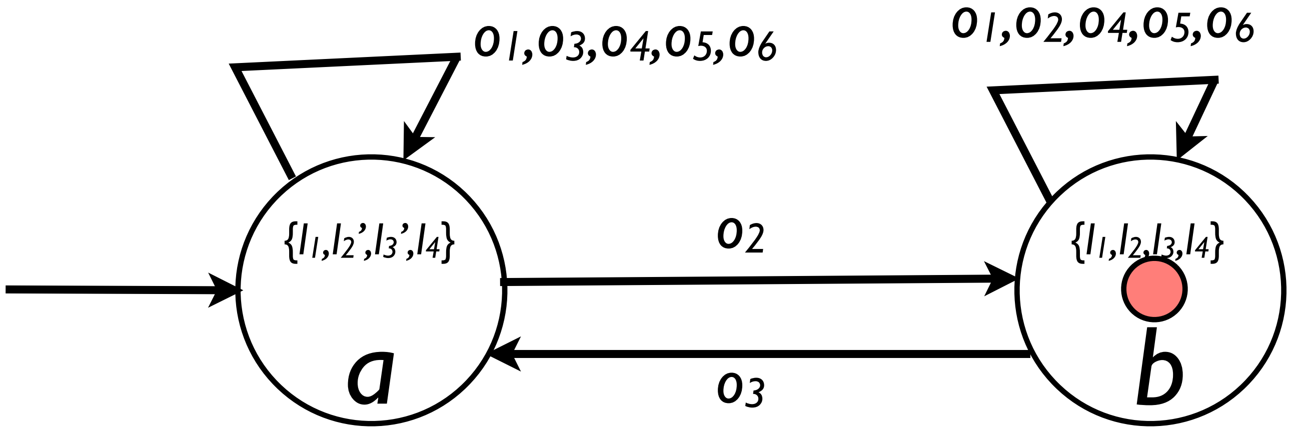
Game with imperfect information
Strategy construction - Safety



Safety objective **Safe**($L\setminus\{o_6\}$).

q_{win}
$=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

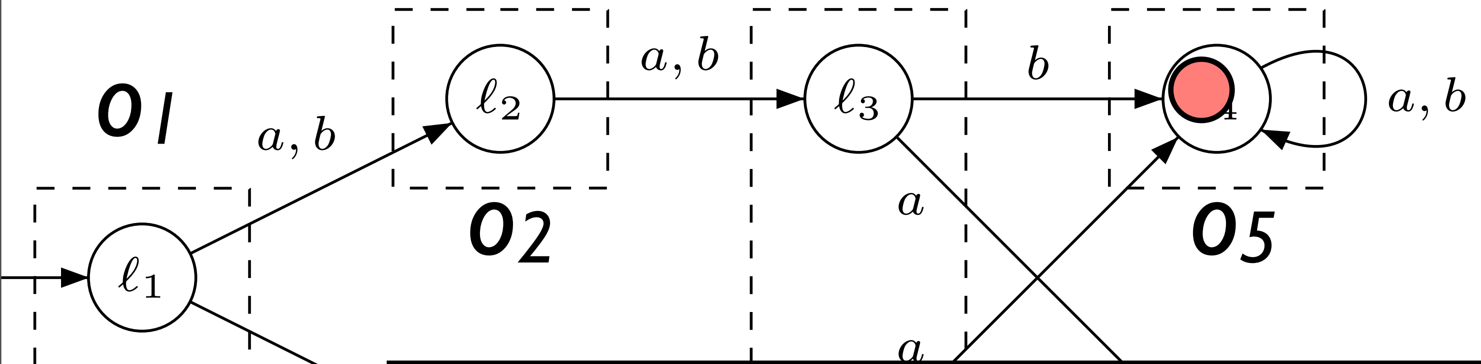
$=\{\{l_1,l_2',l_3',l_4\},\{l_1,l_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$



Player I wins

us play the strategy

Game with imperfect information
Strategy construction - Safety



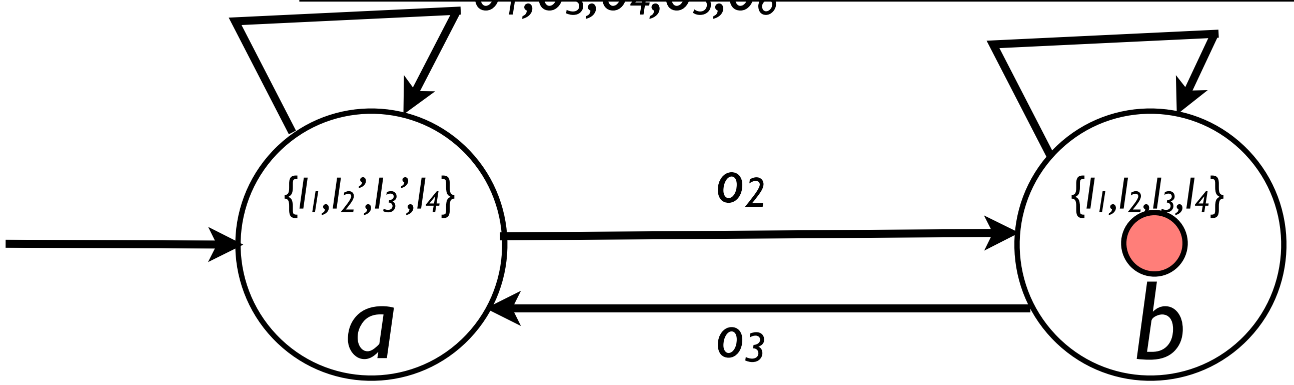
Safety objective **Safe**($L\setminus\{o_6\}$).

q_{win} $=\{L\setminus\{l_2,l_3,l_4'\},L\setminus\{l_2',l_3',l_4'\},L\setminus\{l_2',l_3,l_4'\},L\setminus\{l_2,l_3',l_4'\}\}$

$_2,l_3,l_4\},\{l_1,l_2,l_3',l_4\},\{l_1,l_2',l_3,l_4\}\}$

This has been implemented in a tool called Alpaga

(see [BCDHR09] - TACAS09)



Player I wins

s play the strategy

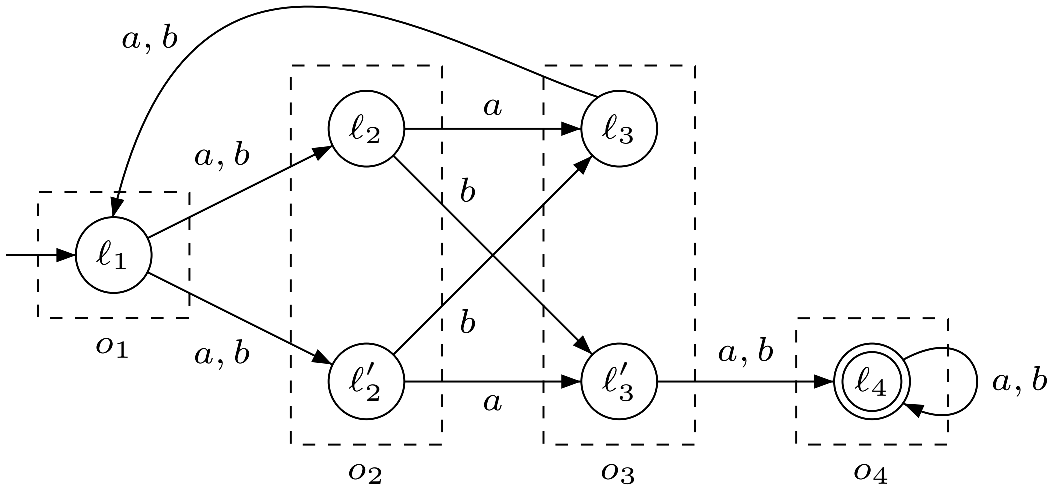
Games with imperfect information Surely-winning - Summary

- Games with imperfect information are **EXPTIME complete** (even for reachability objective [CDHR07] and for safety objective [BD08])
- Games with imperfect information with the notion of surely-winning are **not determined**;
- **Memory** (finite) is needed even for simple reachability objectives.
- **Knowledge-based subset construction** allows us to construct equivalent games with perfect information.
- **Antichains** are adequate data-structures to handle underlying state spaces.
- **Observation-based strategies can be extracted** from the fixed point computations (computed over the lattice of antichains), see [BCDHR08] for details.
- Blind games and antichains are useful to obtain **new efficient algorithms for classical automata-theoretic problems**.

Games of imperfect information

Almost surely-winning

Almost-surely winning
An example

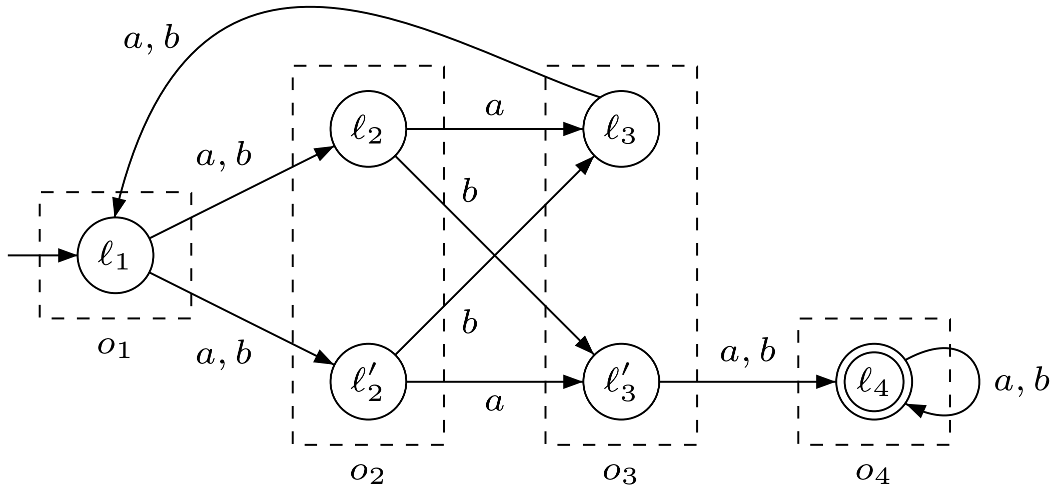


We have seen that Player I can not surely-win the objective **Reach**(o_4) in this game structure with imperfect information.

This is because when Player I has **fixed** his deterministic strategy α, Player 2 can decide how to resolve nondeterminism when entering observation o_2 in order to avoid reaching *l₃'*.

⟹ Player 2 foresee how Player 1 **will** play ! This is not reasonable.

Almost-surely winning
An example



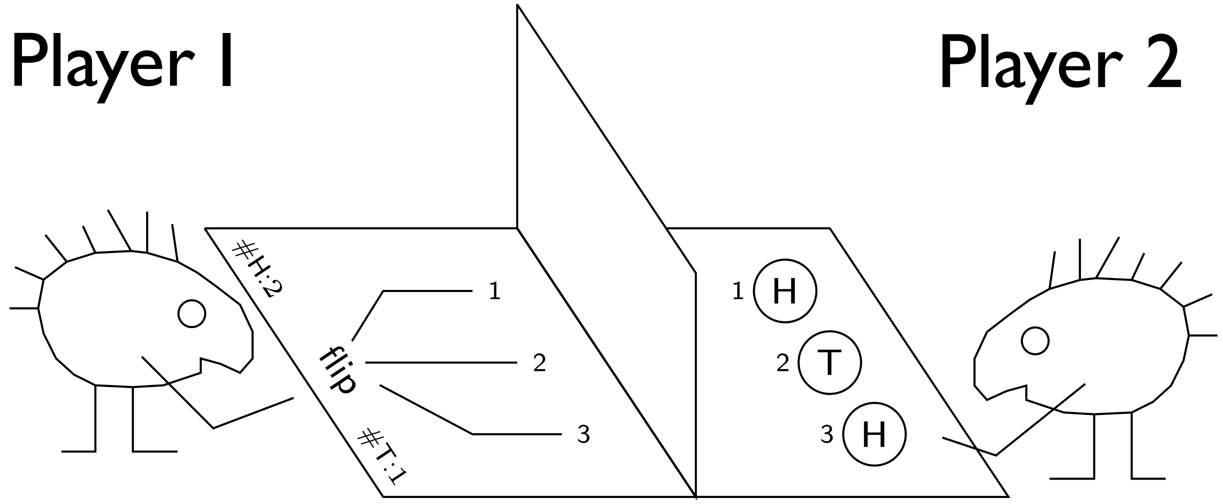
Consider Player I playing this simple following **randomized** strategy:
when receiving observation *o_2*, play uniformly at random *a* and *b*.

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.

This example shows that randomized strategies are **more powerful** than deterministic strategies for winning reachability games with imperfect information.

Almost-surely winning 3-coin game

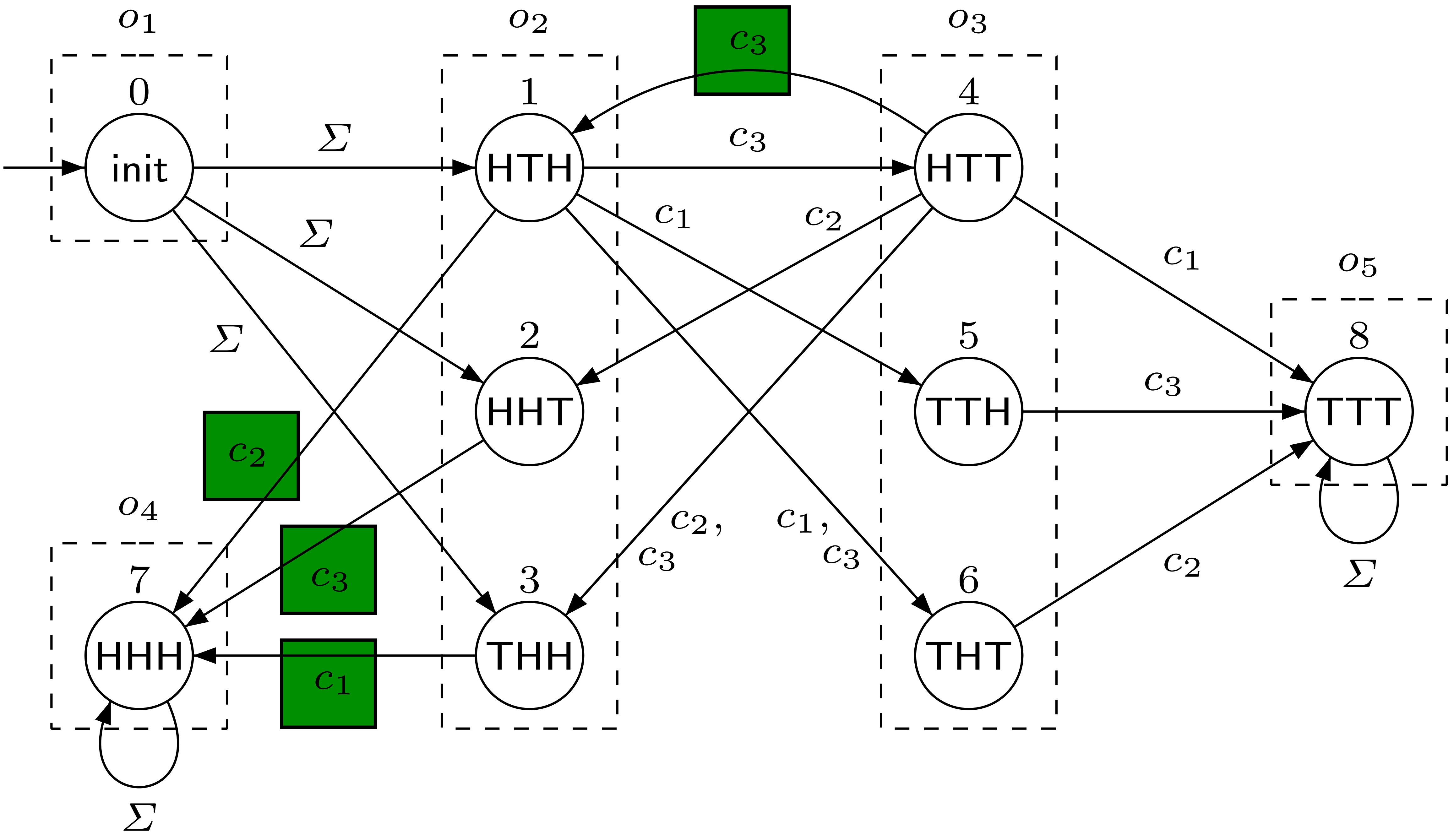


The following randomized strategy is **almost-surely winning** for Player 1 in the 3-coin game.

① Select uniformly at random a coin $c \in \{1,2,3\}$ and ask to flip it.

② If the 3H configuration is not reached, then play c again, and go in ①

Game with imperfect information
3-coin example



Almost-sure winning Randomized strategies

- A randomized strategy for Player 1 is a function $\alpha:(L\times\Sigma)^*L\rightarrow\textbf{Dist}(\Sigma)$.

- A randomized strategy for Player 2 is a function $\beta:(L\times\Sigma)^+\rightarrow\textbf{Dist}(L)$ such that for all finite plays $l_0\sigma_0l_1\sigma_1...l_n\sigma_n$, and location $l\in L$ such that $\beta(l_0\sigma_0l_1\sigma_1...l_n\sigma_n)(l)>0$, we have $(l_n,\sigma_n,l)\in\Delta$.

- Given strategies α and β, an initial location l_0, the probability of a finite play $l_0\sigma_0l_1\sigma_1...l_{n-1}\sigma_{n-1}l_n$, is $\prod_{i\in\{1,...,n\}} p_i$ where

$$p_i=\sum_{\sigma\in\Sigma}\alpha(l_0\sigma_0l_1\sigma_1...l_{n-1})(\sigma_{n-1})\cdot\beta\,(l_0\sigma_0l_1\sigma_1...l_{n-1})(\sigma_{n-1})(l_n).$$

With this measure, the probability of measurable sets is uniquely defined.

- Notions like observation-based strategies are adapted in the expected way.

Almost-sure winning
Randomized strategies

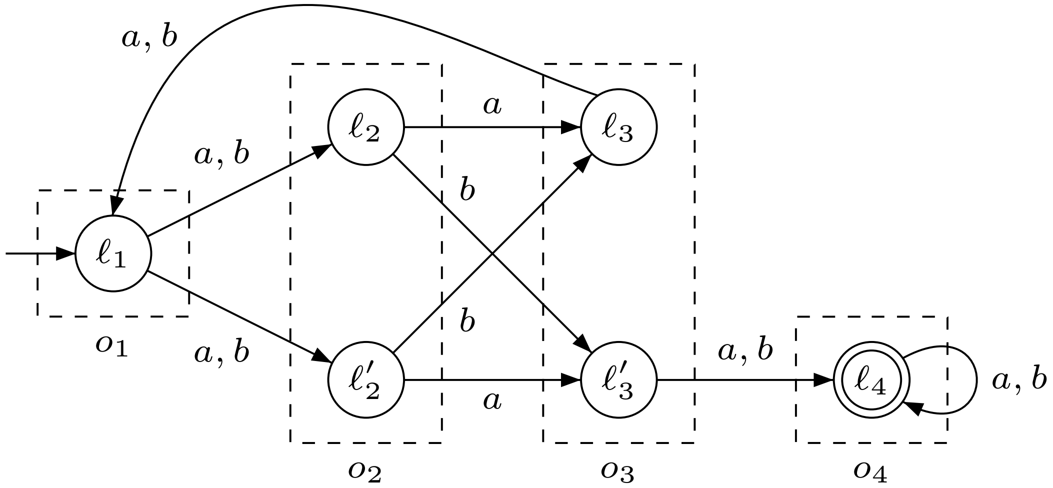
- If φ is a measurable set (any omega-regular set of plays is measurable) then we note **Pr**$_{\alpha\beta}$(l,φ) the probability that the objective φ is satisfied by a play starting in *l* when Player I plays strategy α and Player 2 plays β.

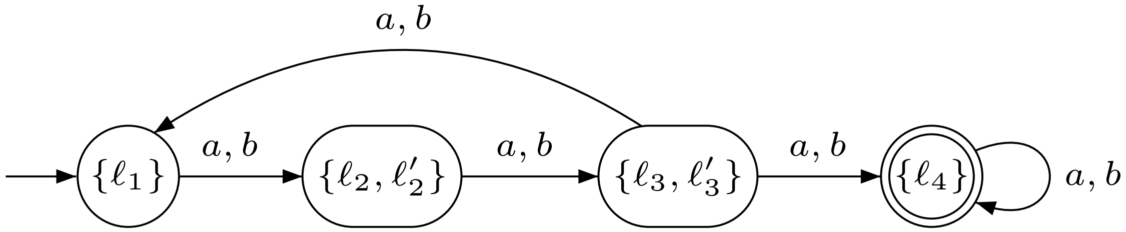
- A randomized strategy α for Player I in G is **almost-surely winning** for the objective φ if

  for all randomized strategy β for Player 2, we have **Pr**$_{\alpha\beta}$(l_i,φ)=1.

- Note that our definition is again asymmetric. While having perfect information does not help Player 2 in the case of surely-winning, it makes Player 2 **stronger** in this probabilistic setting. See [BGG09,GS09] for a symmetric model.

Almost-surely winning
Knowledge-subset construction





Clearly, the knowledge-based subset construction does **not** preserve the **almost-sure winning** strategies.

We need to **extend** the knowledge-based subset construction.

In the new construction, we will consider pairs ***(s,l)***, that we call states:

- *s* models the **knowledge** of Player I
- *l* is the **current location**, i.e. this keeps track of the choices of Player 2.

Almost-surely winning
Extended knowledge-based subset construction

- Given a game structure with imperfect information G=(L,l_i,Σ,Δ,Obs), we construct the **extended knowledge-based subset construction** of G as the game structure **Knw**(G) =H=(Q,q_i,Σ,Δ_H) where:

  -Q={ $(s,l) \mid \exists o \in Obs \cdot s \subseteq o \wedge l \in s$ }
  -q_i=($\{l_i\}$,l_i)
  -$\Delta_H \subseteq Q \times \Sigma \times Q$ is defined by

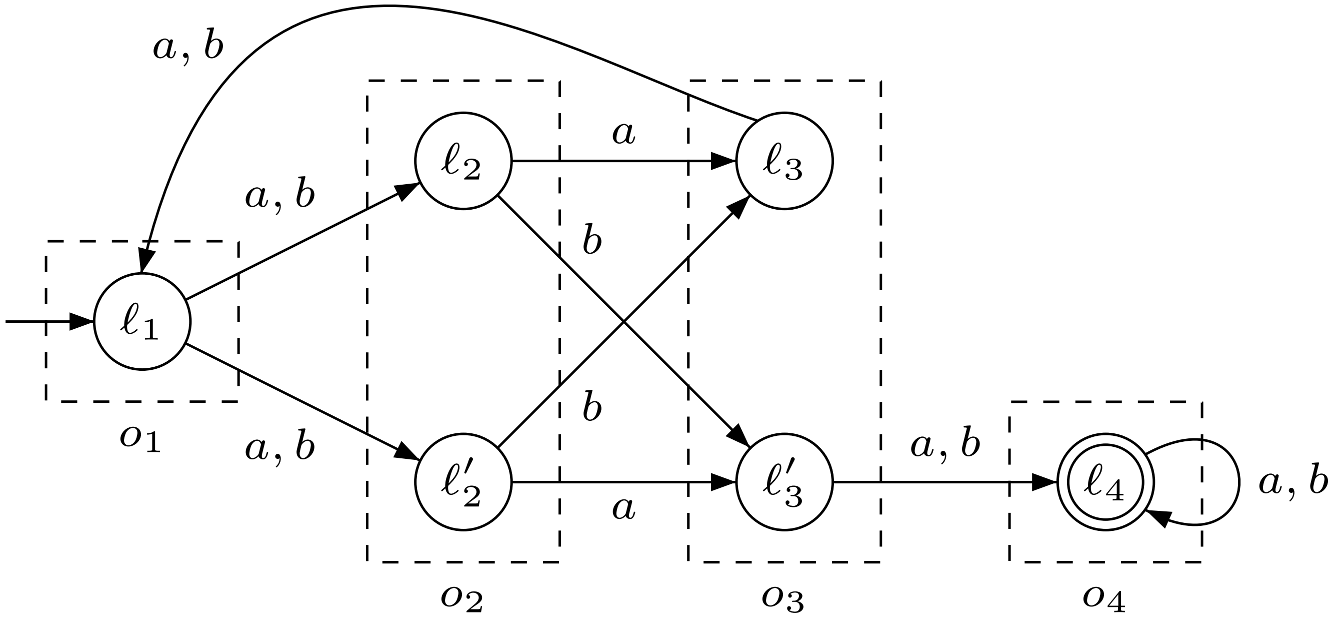
$$((s,l),\sigma,(s',l')) \in \Delta_H$$

iff

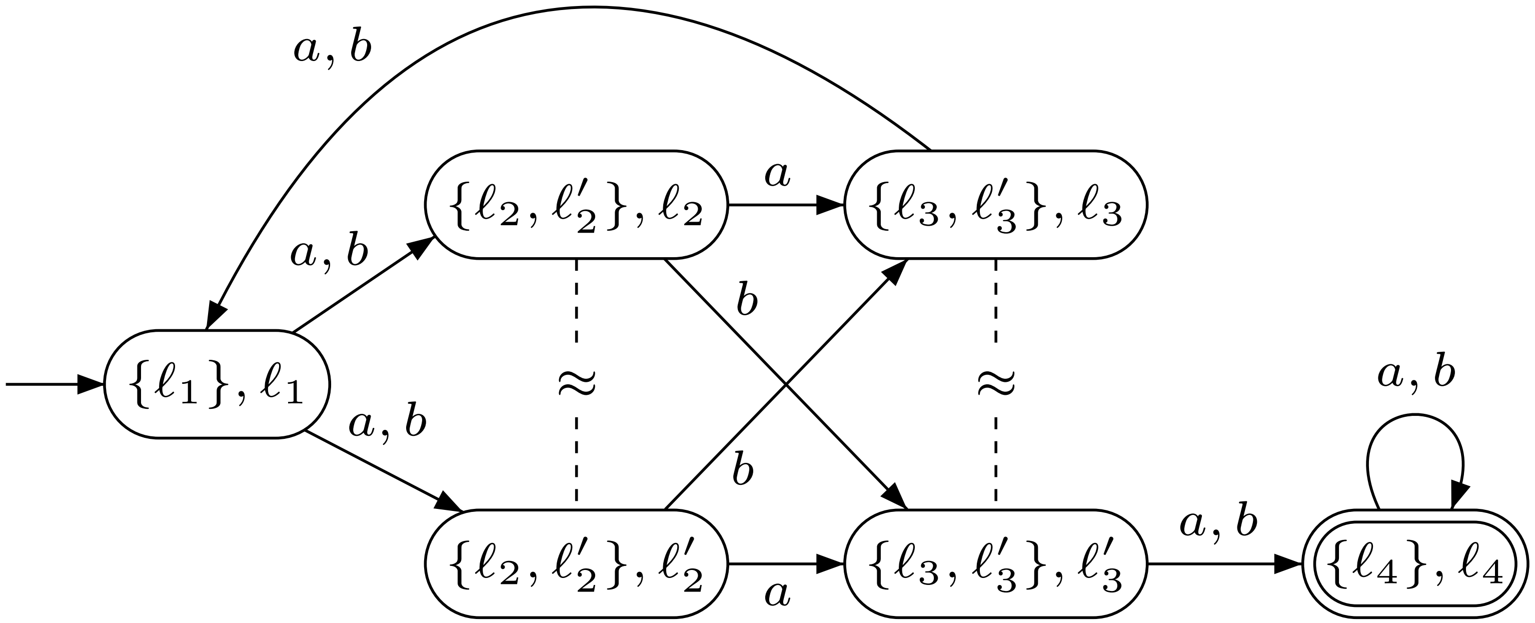
$$\exists o \in Obs \cdot s' = \mathbf{post}_{G,\sigma}(s) \cap o \wedge (l,\sigma,l') \in \Delta$$

Almost-surely winning
Extended knowledge-based subset construction

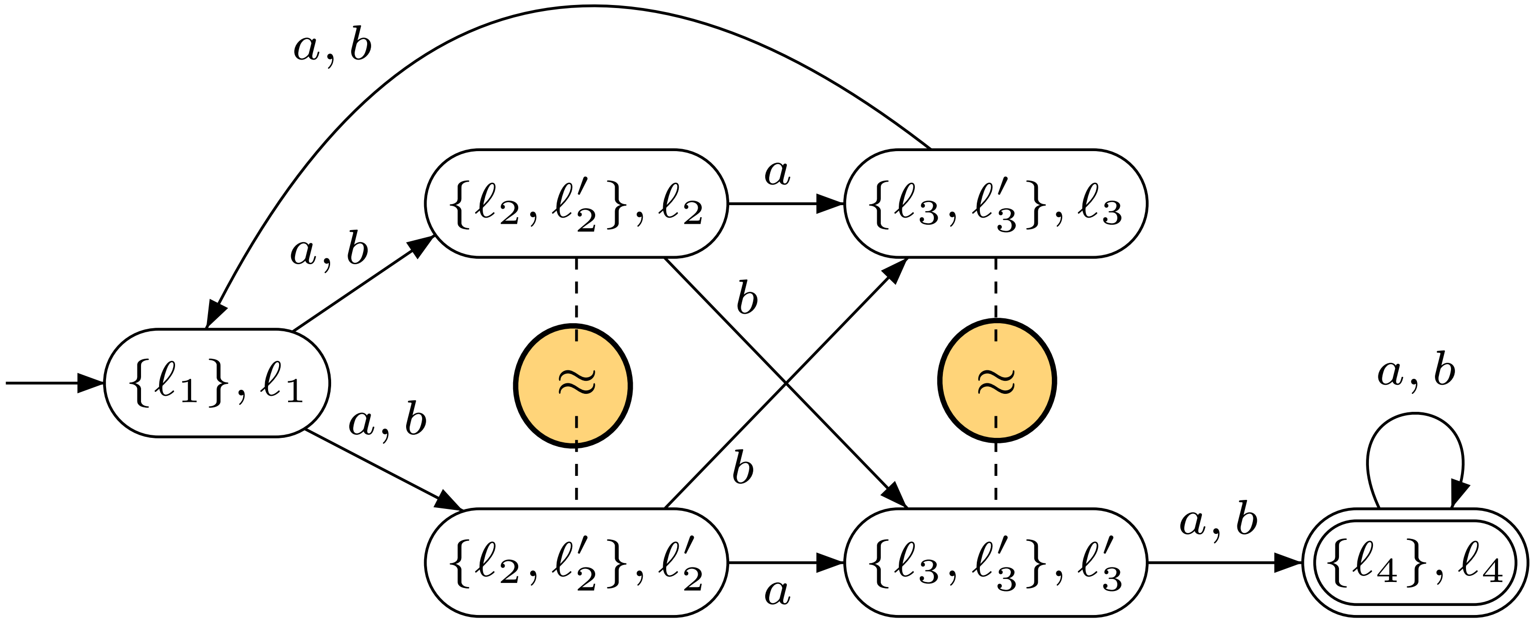
G



H


a, b
a
$\{\ell_2, \ell'_2\}, \ell_2$
$\{\ell_3, \ell'_3\}, \ell_3$
a, b
b
$\{\ell_1\}, \ell_1$
$\approx$
$\approx$
a, b
a, b
b
a, b
$\{\ell_2, \ell'_2\}, \ell'_2$
$\{\ell_3, \ell'_3\}, \ell'_3$
$\{\ell_4\}, \ell_4$
a

Almost-surely winning
Extended knowledge-based subset construction


a, b
$\{\ell_2, \ell'_2\}, \ell_2$
a
$\{\ell_3, \ell'_3\}, \ell_3$
a, b
b
$\{\ell_1\}, \ell_1$
$\approx$
$\approx$
a, b
a, b
b
a, b
$\{\ell_2, \ell'_2\}, \ell'_2$
a
$\{\ell_3, \ell'_3\}, \ell'_3$
$\{\ell_4\}, \ell_4$

- Clearly, in a state (*s*,*l*) we have to make sure that the decisions of Player 1 do not depend on the location *l* but only on cell *s*.
- To solve this problem, we introduce a notion of:
  $\Longrightarrow$ **equivalence** between states, and of
  $\Longrightarrow$ **equivalence-preserving strategies**.

Almost-surely winning
Extended knowledge-based subset construction

- Two states (s,l), (s',l') are **equivalent**, written $(s,l)\approx(s',l')$, if $s=s'$, i.e. when they share the **same knowledge**.
- Let $q\in Q$, $[q]_\approx$ denotes the $\approx$-equivalence class of q.
- Equivalence and equivalence classes for plays and prefixes of plays are defined in the expected way.
- A strategy α is **equivalence-preserving** if $\alpha(\rho)=\alpha(\rho')$ for any two prefixes ρ,ρ' that are equivalent.

Almost-surely winning
Extended knowledge-based subset construction

- Let $pr{:}Obs \rightarrow \{1,..d\}$ be a parity function defining φ=**Parity**(pr) an observable parity objective.
  Let $pr^K{:}Q \rightarrow \{1,...,d\}$ be the function such that $pr^K((s,l))=p$ iff $pr(o)=p$ for the observation o such that $s \subseteq o$.
  Then φ^K is defined as **Parity**(pr^K).

- **Theorem**. For all game structures with imperfect information G,

  Player I has an observation-based **almost-surely** winning strategy in G for a parity objective φ

  iff

  Player I has an **equivalence-preserving** **almost-surely** winning strategy in H for the parity objective φ^K.

Almost-surely winning
An algorithm for reachability

- First, note that for safety objectives, almost-surely winning and surely-winning are **equivalent** notions.

  This is because any violation of a safety objective by a finite prefix of play would make the probability of being safe **strictly less than 1**.

- In [CDHR07], we have given algorithms for solving reachability and Büchi games, we concentrate here on **reachability** objectives.

- We provide an algorithm for the reachability objectives using the extended knowledge-based subset construction.

Almost-surely winning
An algorithm for reachability

- First, it can be shown that **memoryless strategies** are sufficient for Player I to almost-surely win the game of perfect information H=Knw(G) for reachability (and Büchi objectives).

- Let H=**Kwn**(G)=(Q,q_i,Σ,Δ_H), let **Reach**(T) with $T \subseteq Q$ be an observable reachability objective in H (we assume T to be absorbing), and ≈ the equivalence relation that declares equivalent two states with the same knowledge.

  Player I almost surely-win with an equivalence preserving strategy in the set $W \subseteq Q$

  iff

  there exists two functions **Allow**: $W \rightarrow 2^{\Sigma}$ and **Good**:$W \rightarrow \Sigma$ such that $\forall q \in W$:

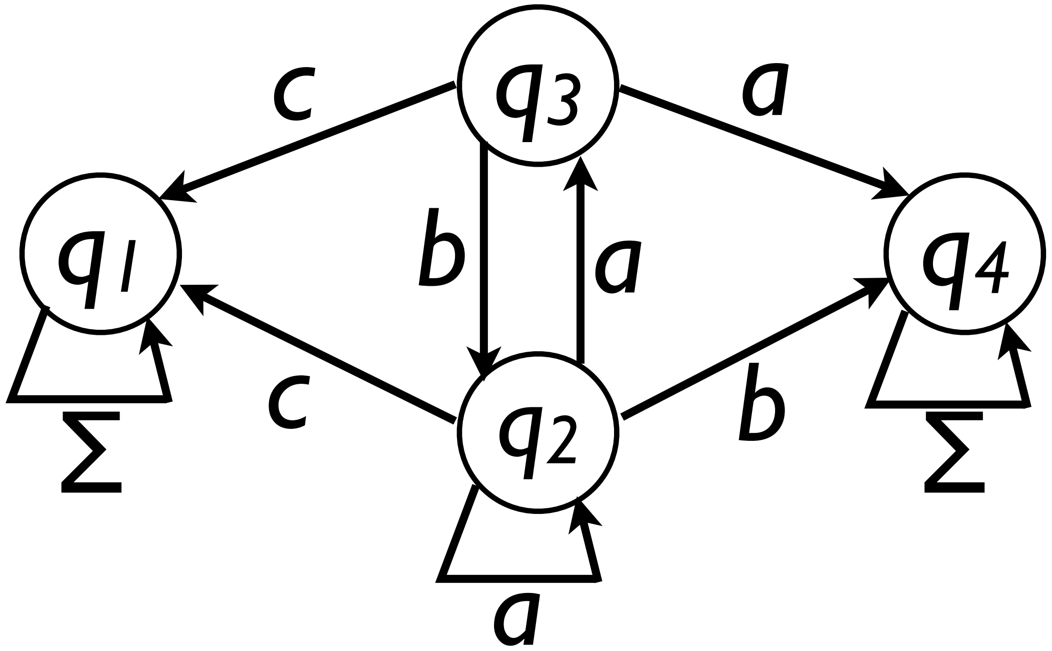
  (i) **Good**(q)∈**Allow**(q)

  (ii) for all $q \approx q'$ and for all σ∈**Allow**(q), **post**$_{H,\sigma}(q') \subseteq W$;

  (iii) on the graph (W,E) with E={ (q,q')∈$W \times W$ | (q,**Good**(q),q')∈Δ_H}, all infinite paths visit a state in T.

Almost-surely winning
An algorithm for reachability

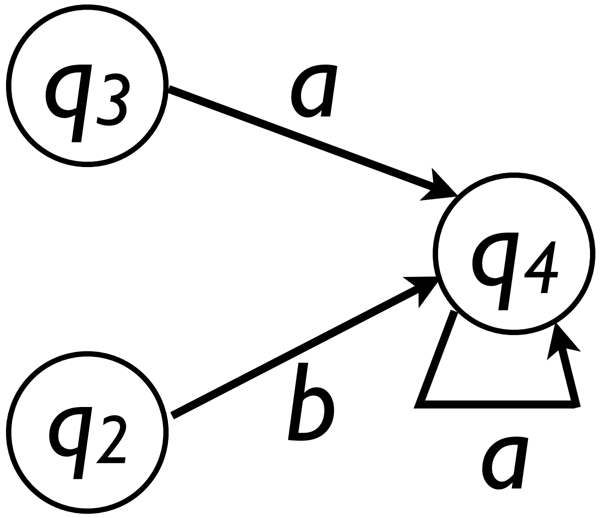
$q_2 \approx q_3$ **Reach**({q$_4$})



$W=\{q_2,q_3,q_4\}$ is witnessed by the following functions:

Allow(q_2)=$\{a,b\}$ **Good**(q_2)=b
Allow(q_3)=$\{a,b\}$ **Good**(q_3)=a
Allow(q_4)=Σ **Good**(q_4)=a

$(V,E)=$



Almost-surely winning
An algorithm for reachability

- How to compute the set W ? We need a double fixed point computation.

  Intuitively, the greatest fixed point is used to determine the safe region together with the **Allow** actions, and a least fixed point to ensure that progress toward the target is possible thanks to the **Good** actions.

- The set W is the limit of the following computation:

  W^0=Q
  W^{i+1}=**PosReach**(W^i) for all $i \geq 0$

  where **PosReach**(W^i) is the limit of the following iteration:

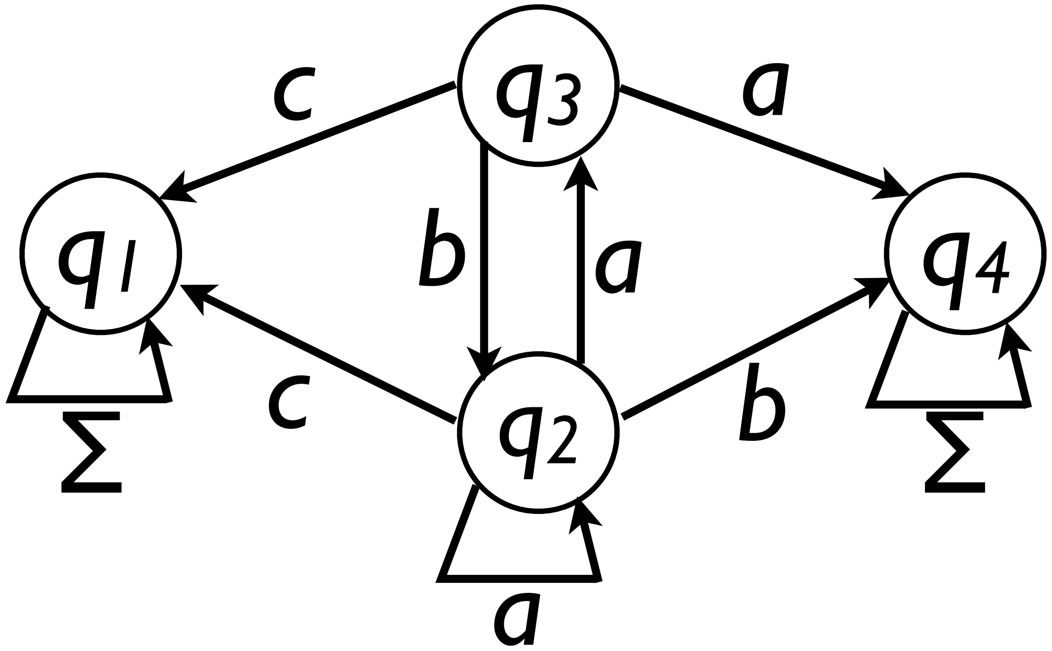
  $X^0=T$
  $X^{j+1}=X^j \cup$**Apre**(W^i,X^j) for all $j \geq 0$.

  where **Apre**(W,X)={ $q \in W \mid \exists \sigma \in \Sigma \cdot \mathbf{post}_{\sigma,H}(q) \subseteq X \wedge \forall q \approx q' \cdot \mathbf{post}_{\sigma,H}(q') \subseteq W$}

  Note that **good** σ can differ from states to states but should be **allow**ed in all equivalent states !

Almost-surely winning
An algorithm for reachability

$q_2 \approx q_3$ $\textbf{Reach}(\{l_4\})$



Reach T while staying in W^0.

$W^0=\{q_1,q_2,q_3,q_4\}$

$W^1=\textbf{PosReach}(W^0)$

$X^0=\{q_4\}$

$X^1=\{q_4\}\cup\textbf{Apre}(\{q_1,q_2,q_3,q_4\},\{q_4\})$

$=\{q_4\}\cup\{q_2,q_3,q_4\}=\{q_2,q_3,q_4\}=X^2=W^1$

$W^2=\textbf{PosReach}(\{q_2,q_3,q_4\})$

$X^0= \{q_4\}\cup\textbf{Apre}(\{q_2,q_3,q_4\},\{q_4\})$

$X^1=\{q_4\}\cup\textbf{Apre}(\{q_2,q_3,q_4\},\{q_4\})$

$=\{q_4\}\cup\{q_2,q_3,q_4\}=\{q_2,q_3,q_4\}=X^2=W^2$

$W^2=W^1$

(fixed point is reached)

$\textbf{Apre}(W,X)=\{\ q\in W \mid \exists\sigma\in\Sigma\cdot\textbf{post}_{\sigma,H}(q)\subseteq X\wedge\forall q\approx q'\cdot\textbf{post}_{\sigma,H}(q')\subseteq W\}$

Almost-surely winning
Beyond this introduction

- It can be shown that the algorithm presented here (that uses the extended subset construction) is worst case optimal. The problem of deciding if a location is almost-sure winning for a reachability objective in a game with imperfect information is **ExpTime-Complete**.
- Antichains can be extended to compute efficiently the set of almost-surely winning states for reachability and Büchi objectives.
- The algorithm can be applied to compute the almost-sure winning strategy for the 3-coin example (see written notes).
- co-Büchi has been shown undecidable recently [CD2010] !
- Models with two players with imperfect observation [BGG09,GS09].

Conclusion

- Games of imperfect information are useful to model **faithfully** practical synthesis problems.
- **Memory** and **randomization** are necessary to win games with imperfect information even for reachability objectives.
- Reductions to games of perfect information are possible... but more complex in the case of “almost-surely winning”.
- Direct algorithms that uses **tailored data-structures** (antichains) are useful to obtain practical algorithms.