ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Games for Controller Synthesis

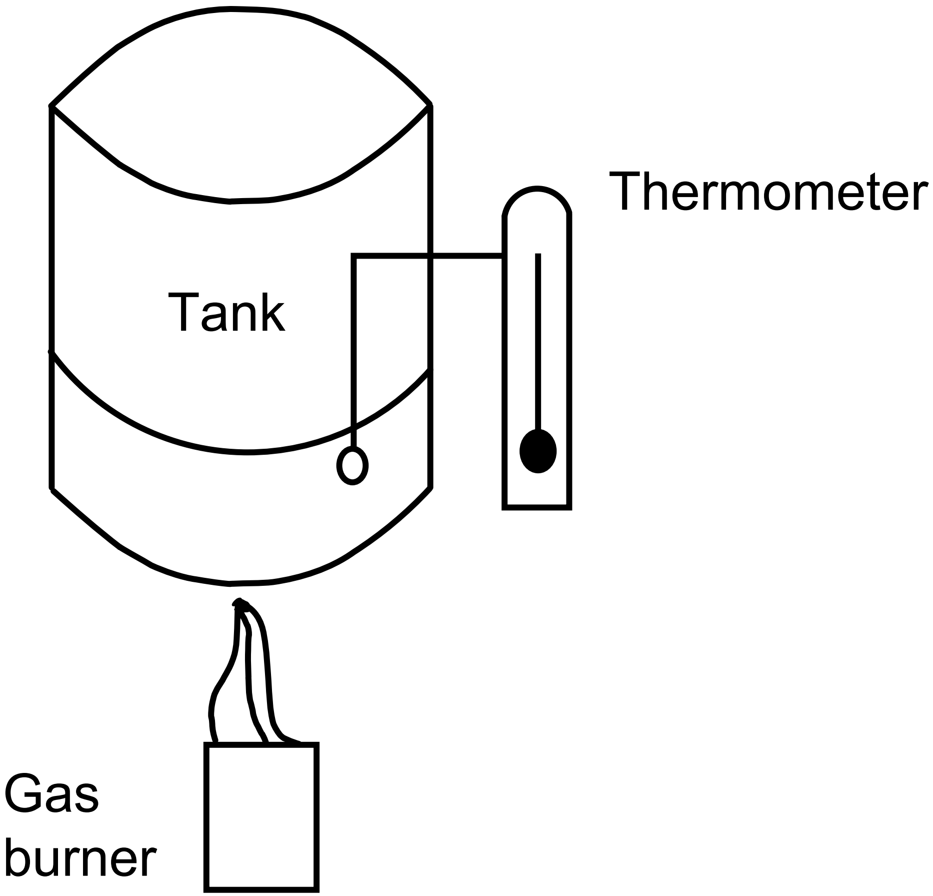
Laurent Doyen

EPFL

MoVeP'08

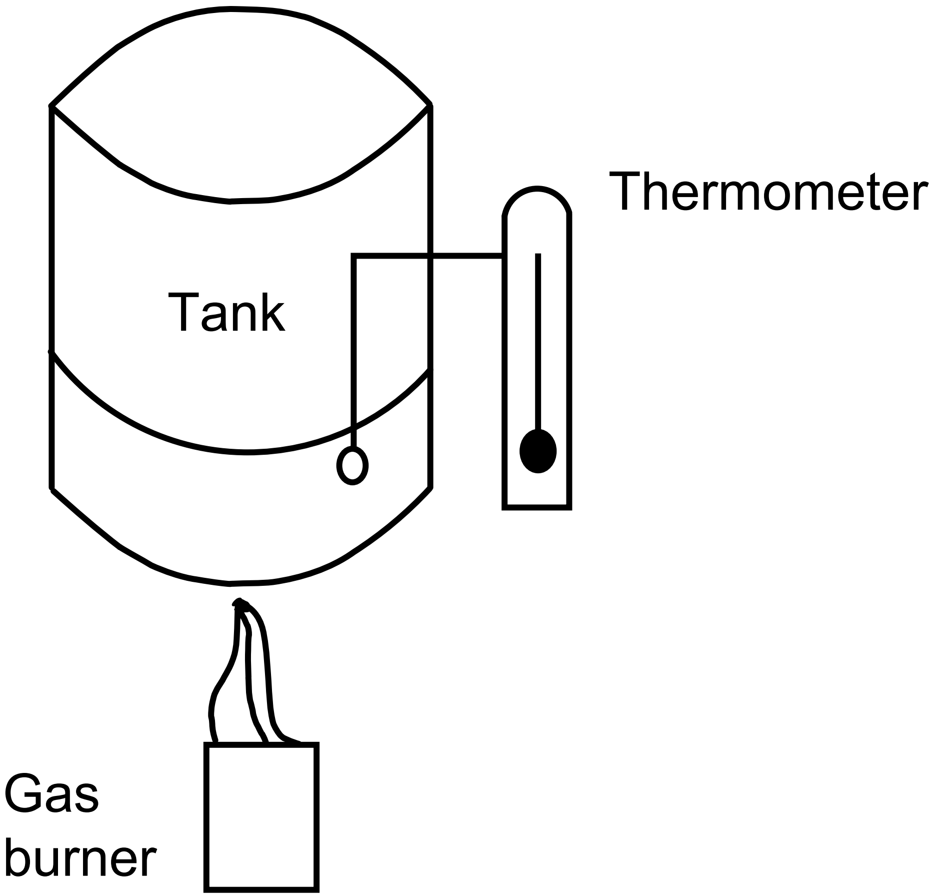
The Synthesis Question

Given a plant P …



The Synthesis Question

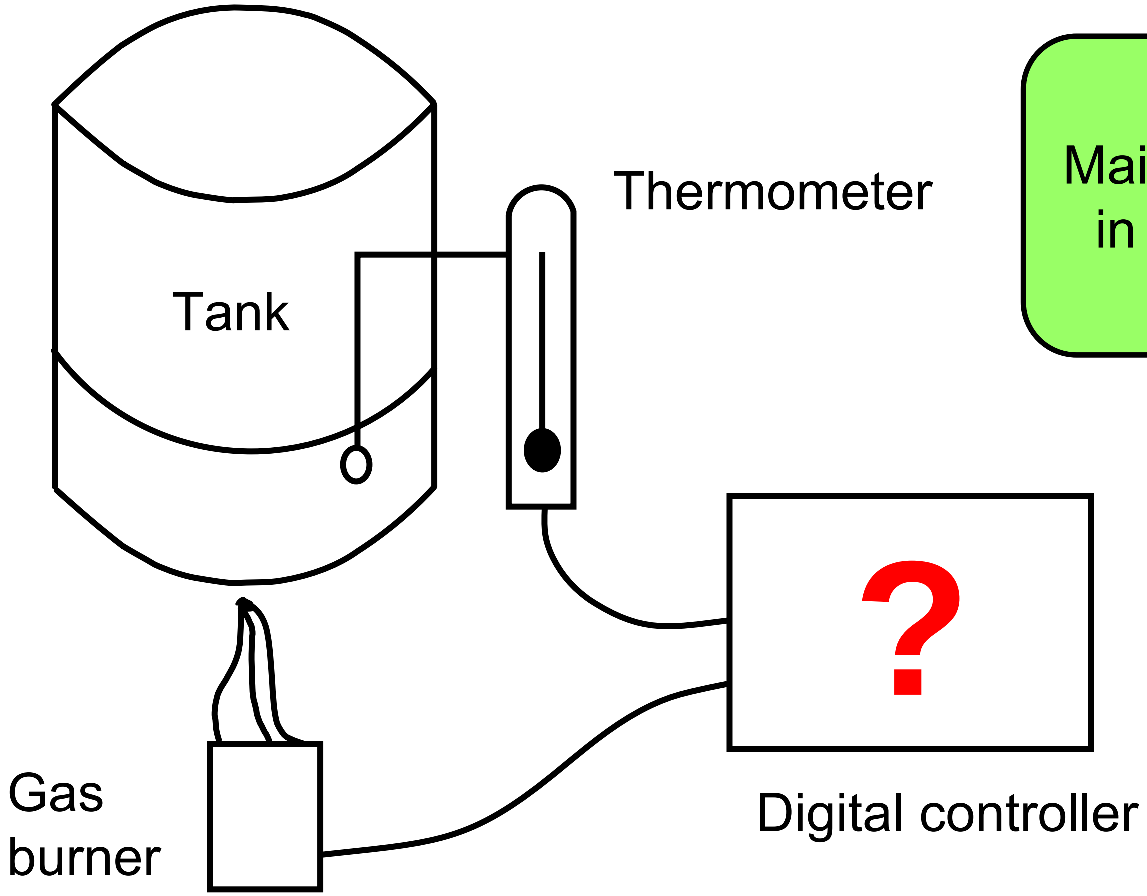
Given a plant P and a specification φ,…



Maintain the temperature in the range $[T_{min}, T_{max}]$.

The Synthesis Question

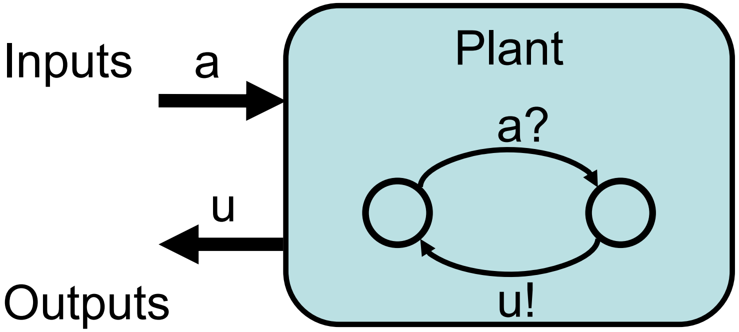
Given a plant P and a specification φ, is there a controller C such that the closed-loop system C ‖ P satisfies φ ?



Maintain the temperature in the range [T_{min},T_{max}].

Synthesis as a game

Given a plant P and a specification φ, is there a controller C such that the closed-loop system C‖P satisfies φ ?



Specification

$$\varphi \equiv \varphi_1(a_1, u_1) \wedge \varphi_2(a_2, u_2)$$

Plant: 2-players game arena

Input (Player 1, System, Controller)

vs.

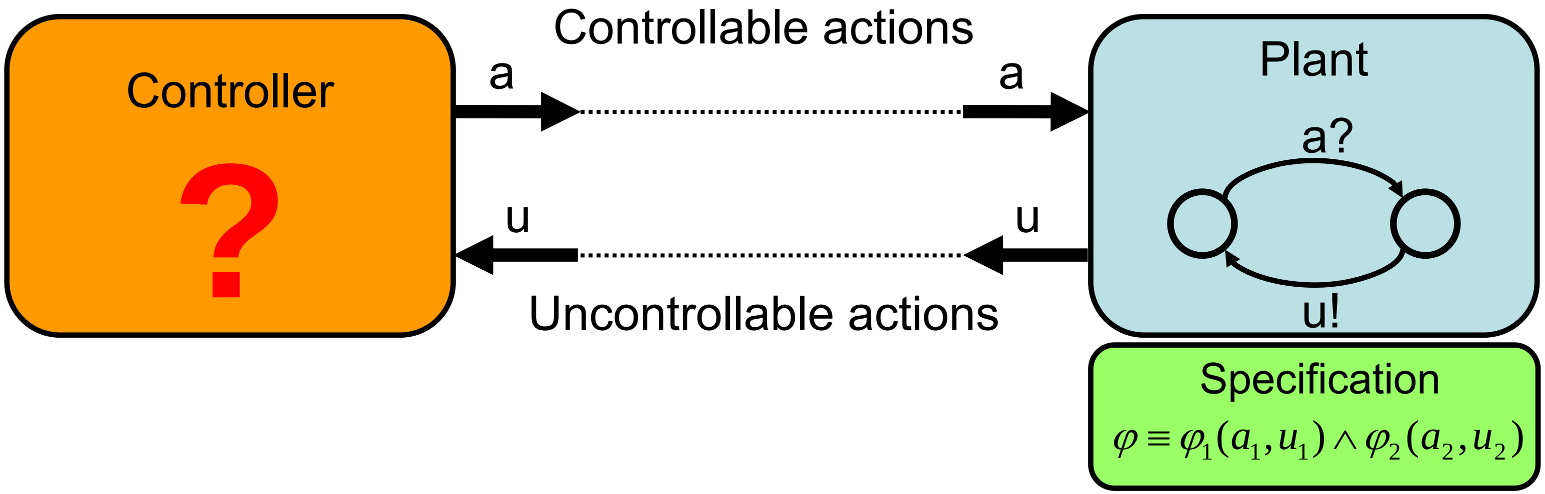
Output (Player 2, Environment, Plant)

Specification: game objective

for Player 1

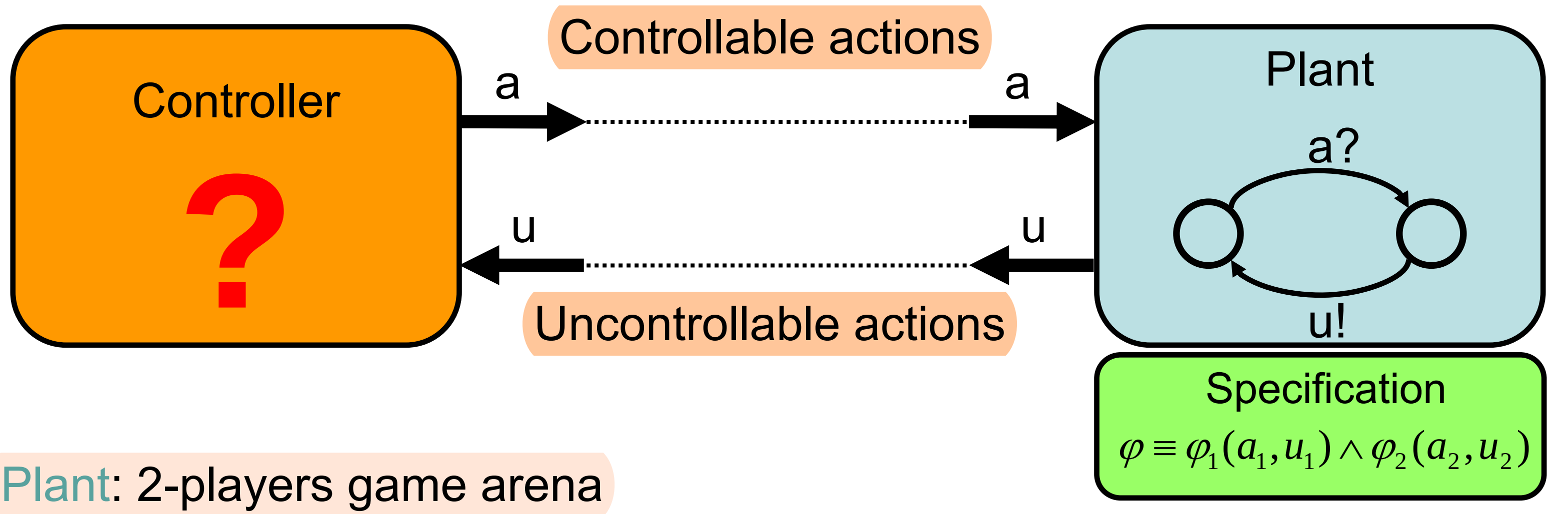
The Synthesis Question

Given a plant P and a specification φ, is there a controller C such that the closed-loop system C‖P satisfies φ ?



If a controller C exists, then construct such a controller.

Synthesis as a game



Plant: 2-players game arena

Specification: game objective for Player 1

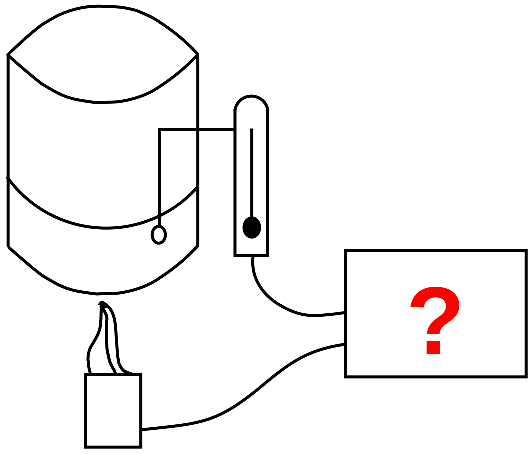
Controller: winning strategy for Player 1

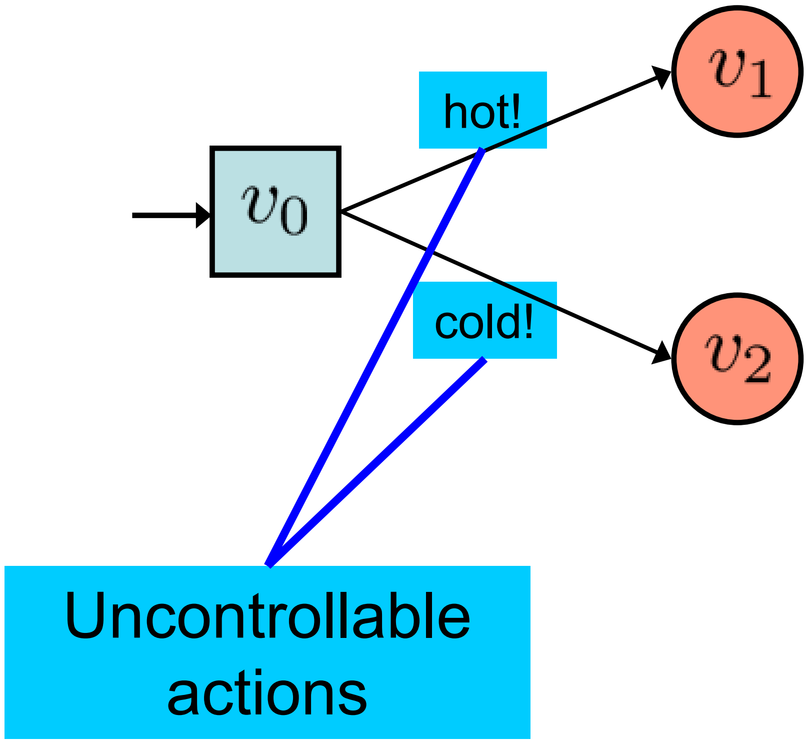
We are often interested in simple controllers:
finite-state, or even stateless (memoryless).

We are also often interested in “least restrictive” controllers.

Example

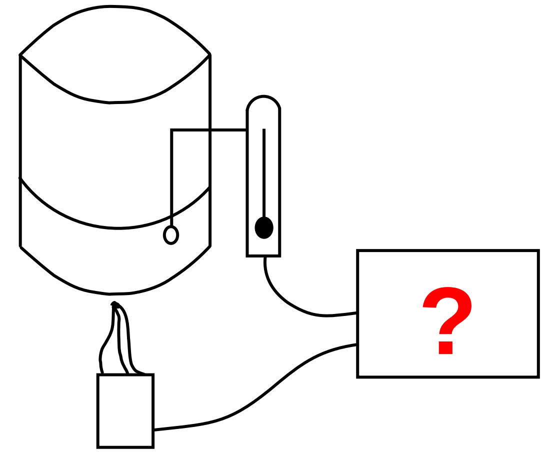
Objective: avoid Bad

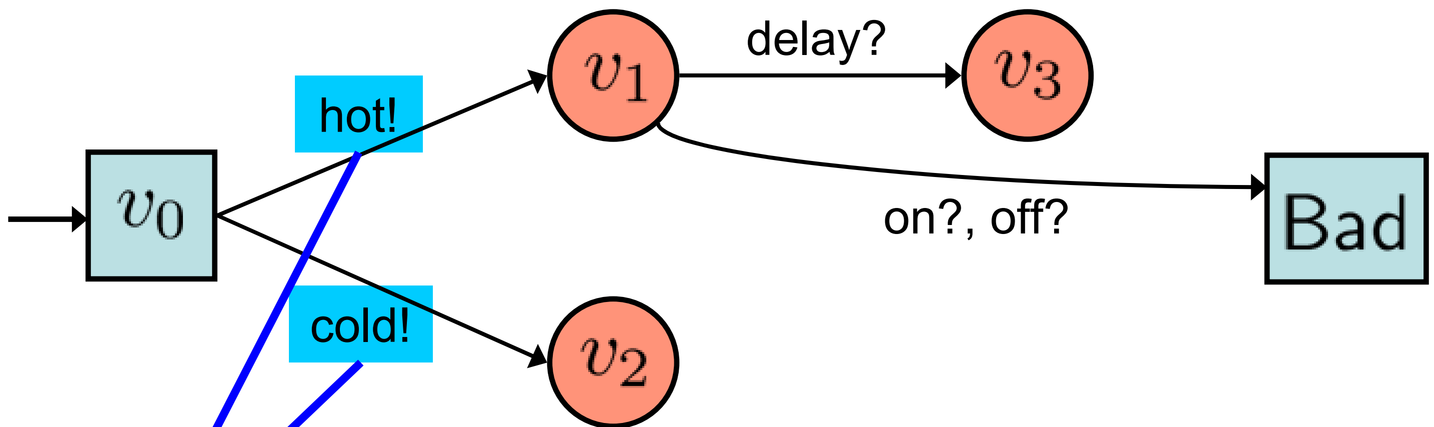




Example

Objective: avoid Bad

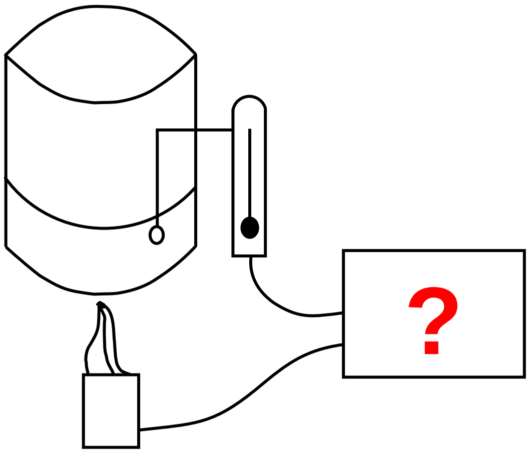


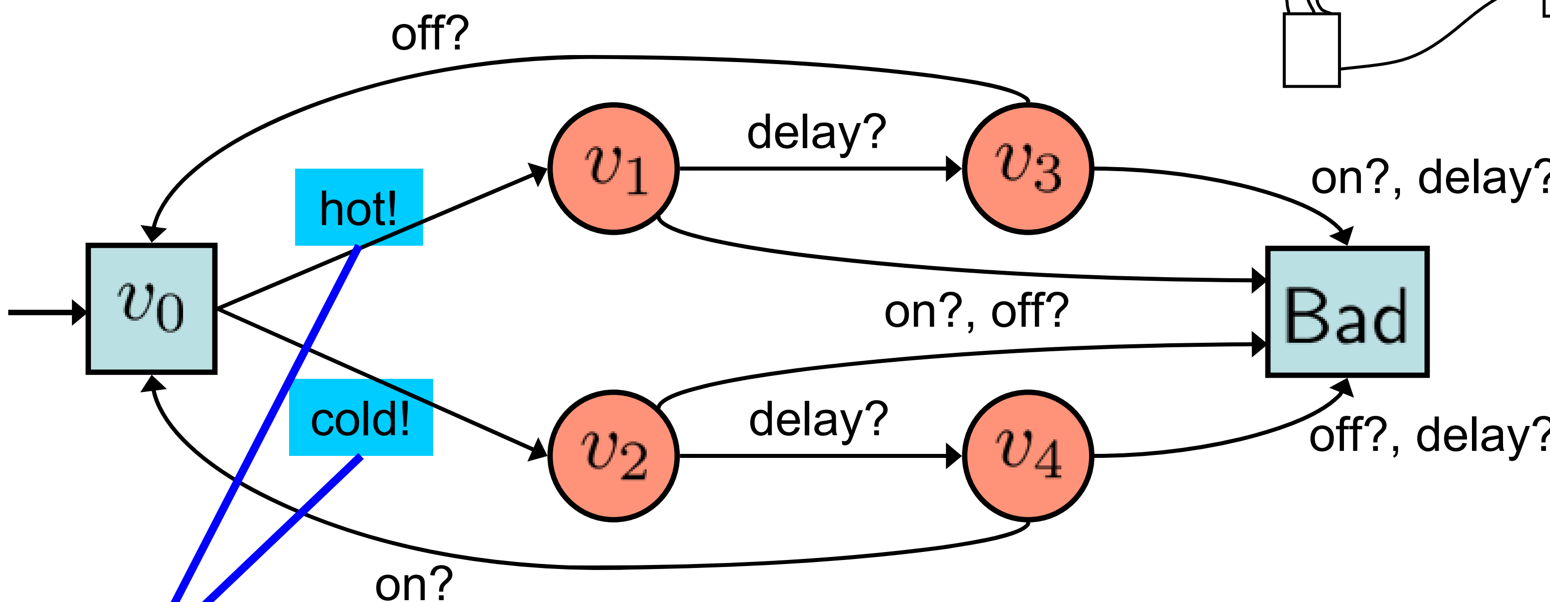


Uncontrollable actions

Example

Objective: avoid Bad

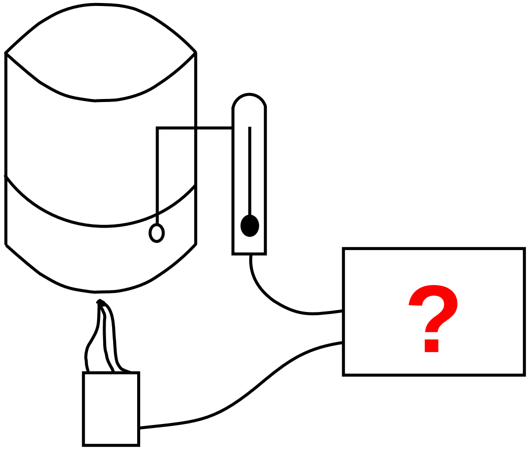


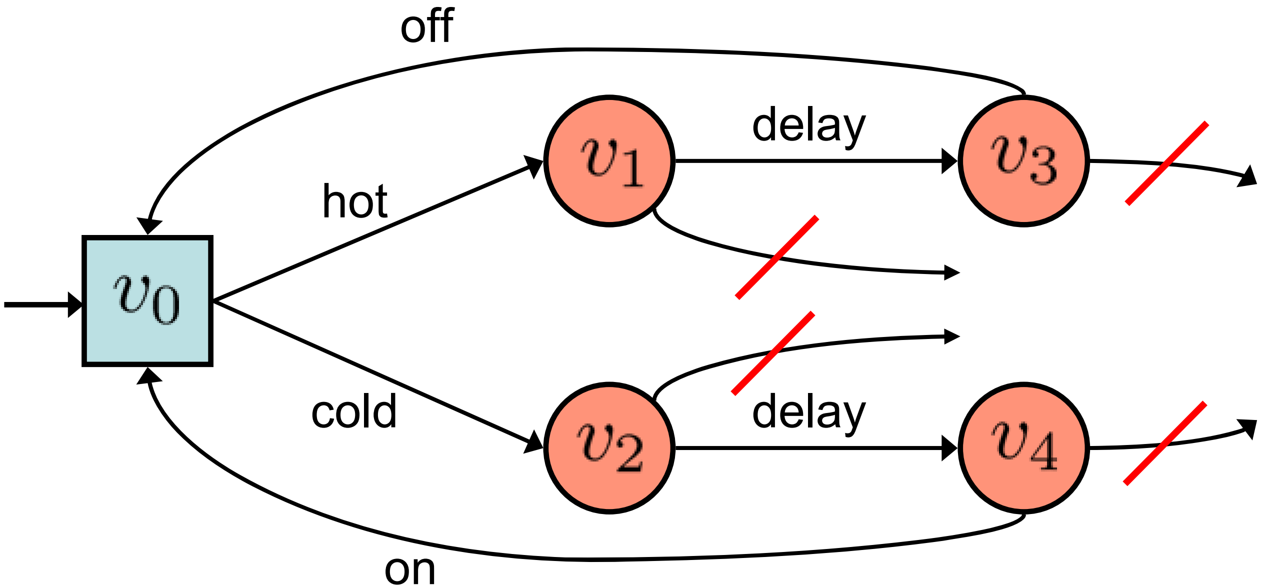


Uncontrollable actions

Example

Objective: avoid Bad





Winning strategy = Controller

Games for Synthesis

Several types of games:

- Turn-based vs. Concurrent
- Perfect-information vs. Partial information
- Sure vs. Almost-sure winning
- Objective: graph labelling vs. monitor
- Timed vs. untimed
- Stochastic vs. deterministic
- etc. …

This tutorial: Games played on graphs, 2 players, turn-based, ω-regular objectives.

Games for Synthesis

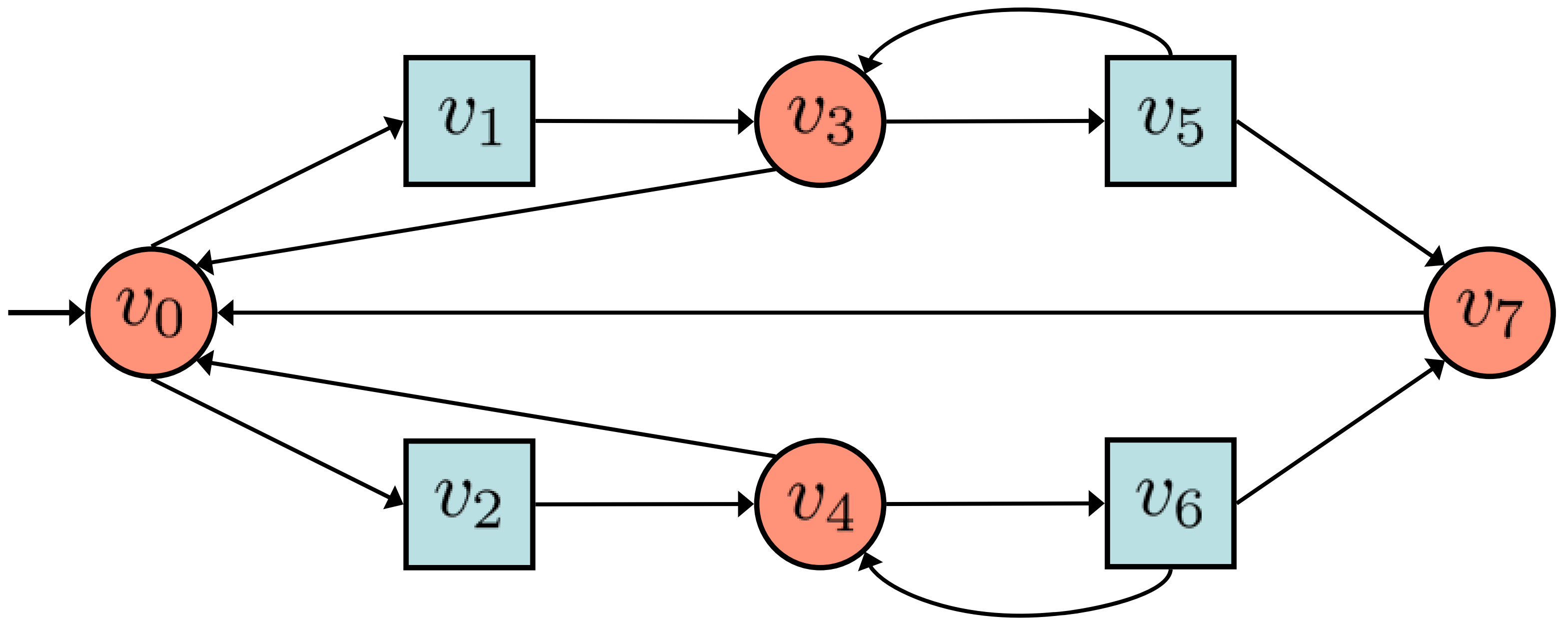
This tutorial: Games played on graphs, 2 players, turn-based, ω-regular objectives.

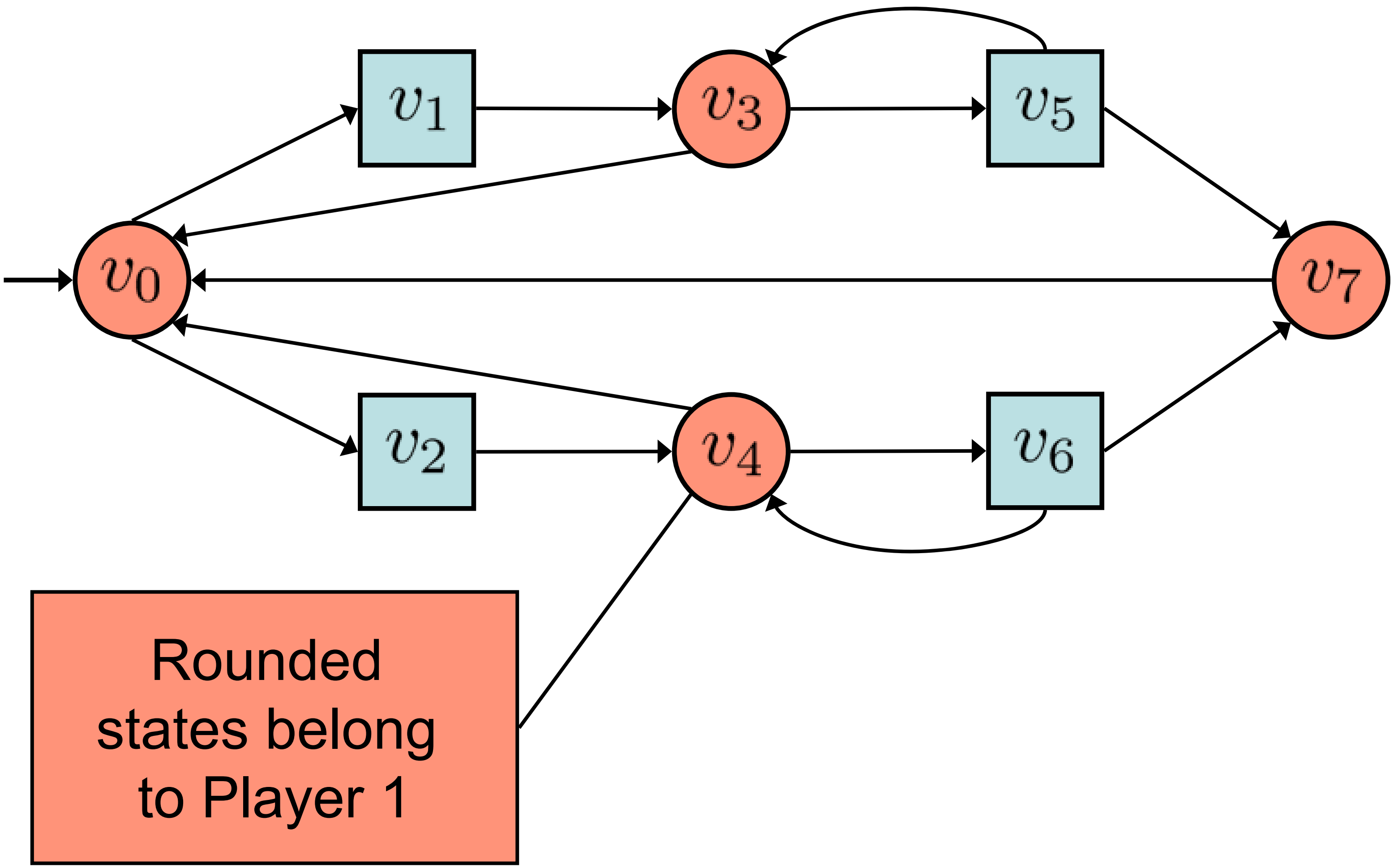
Outline

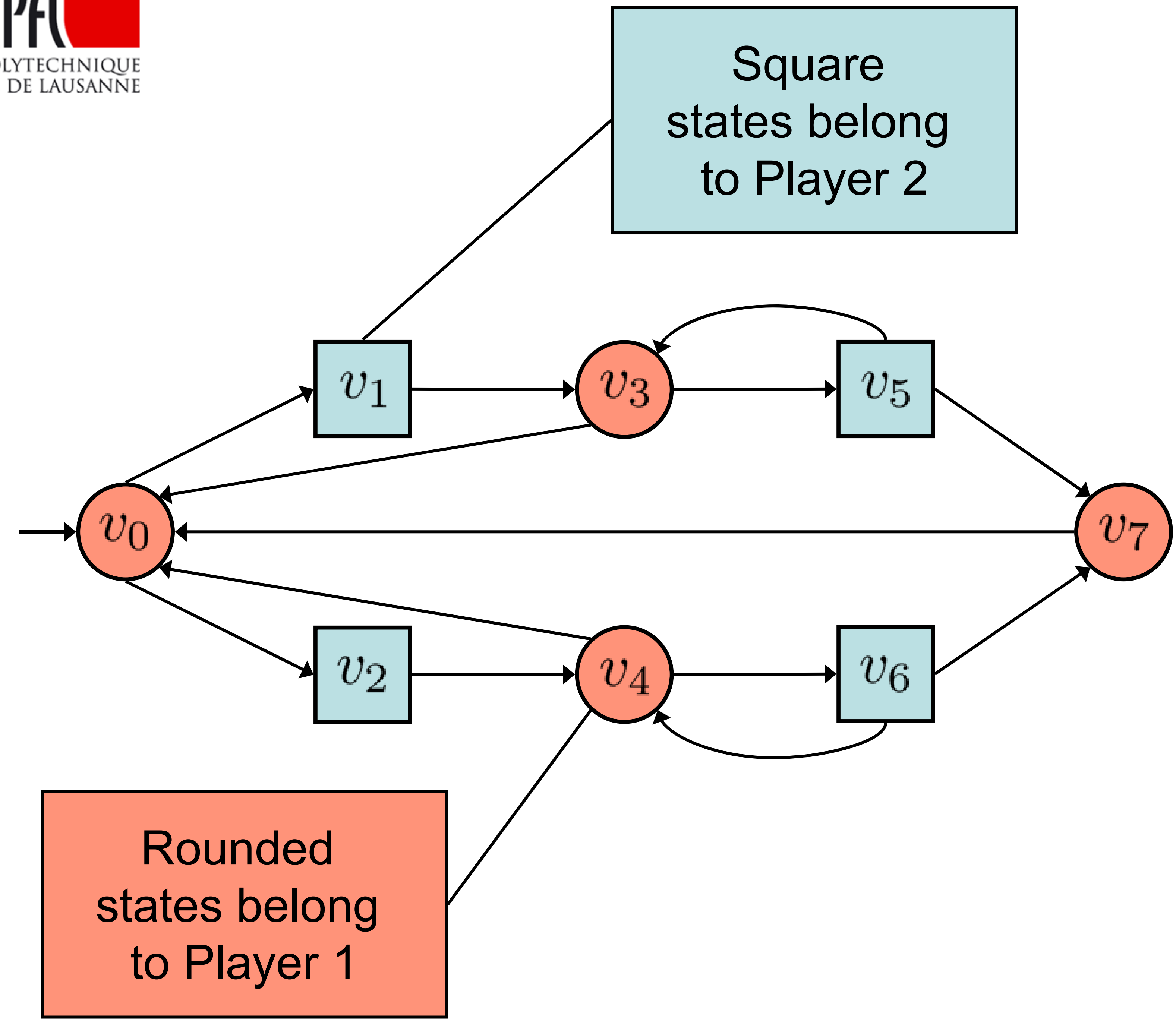
Part #1: perfect-information

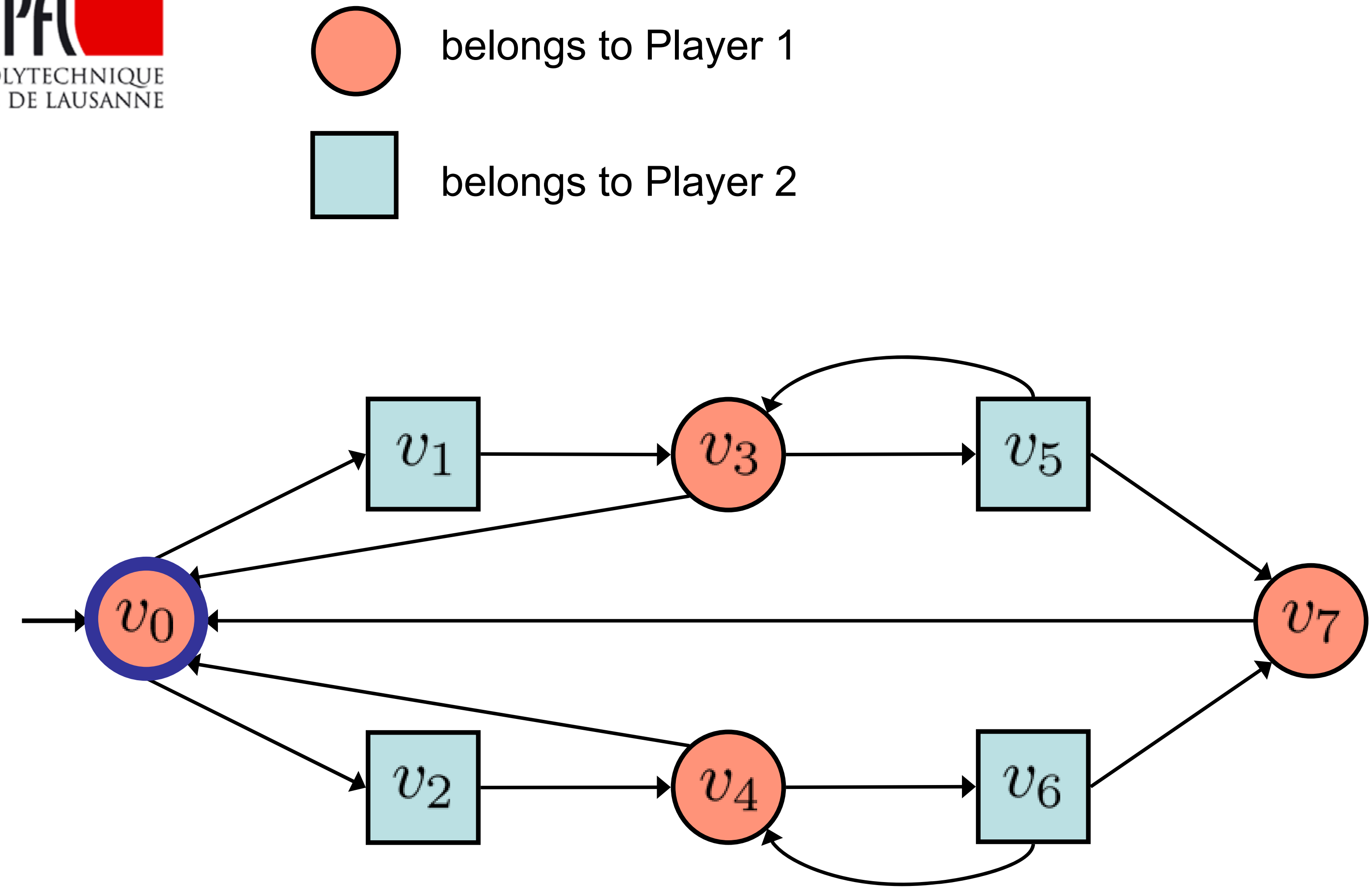
Part #2: partial-information

Two-player game structures


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7


v_0
v_1
v_2
v_3
v_4
v_5
v_6
v_7
Rounded states belong to Player 1


Square states belong to Player 2
v_1
v_3
v_5
v_0
v_7
v_2
v_4
v_6
Rounded states belong to Player 1

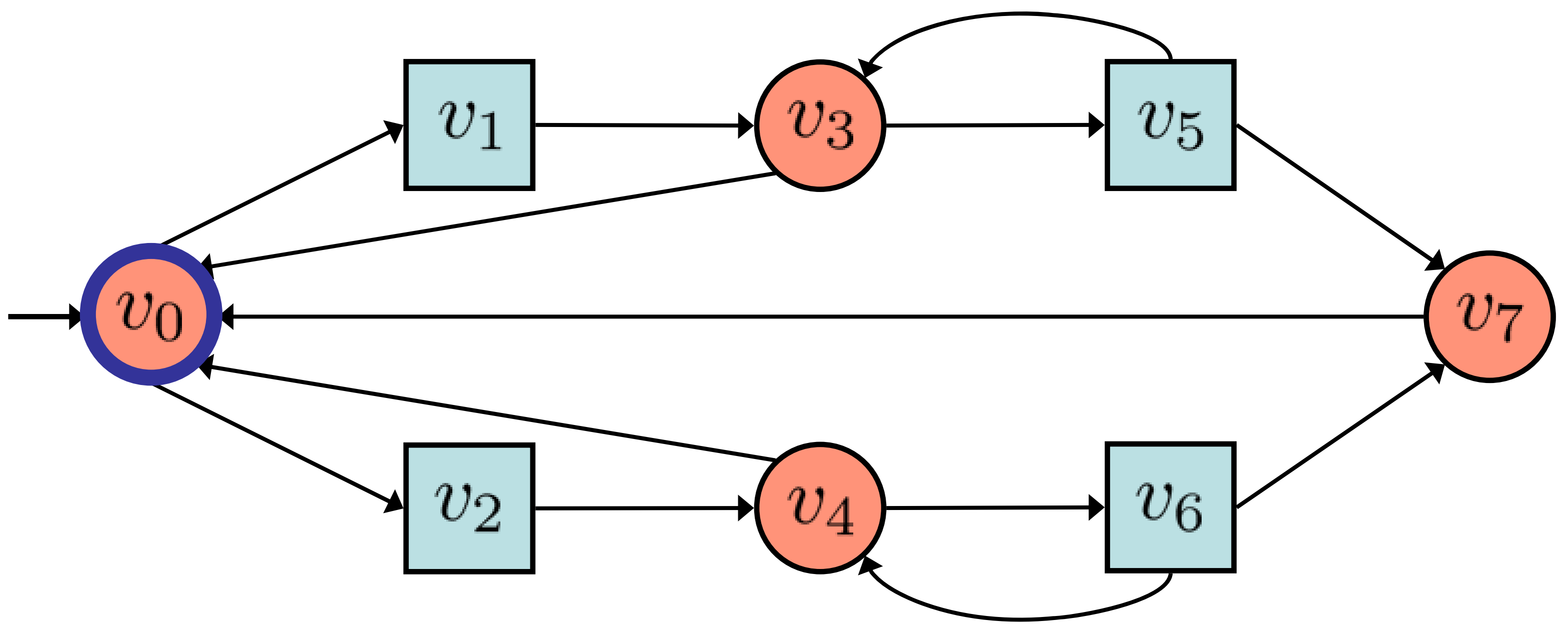


Playing the game: the players move a **token** along the edges of the graph

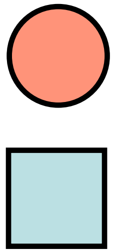
- The token is initially in v_0.
- In rounded states, Player 1 chooses the next state.
- In square states, Player 2 chooses the next state.

belongs to Player 1

belongs to Player 2

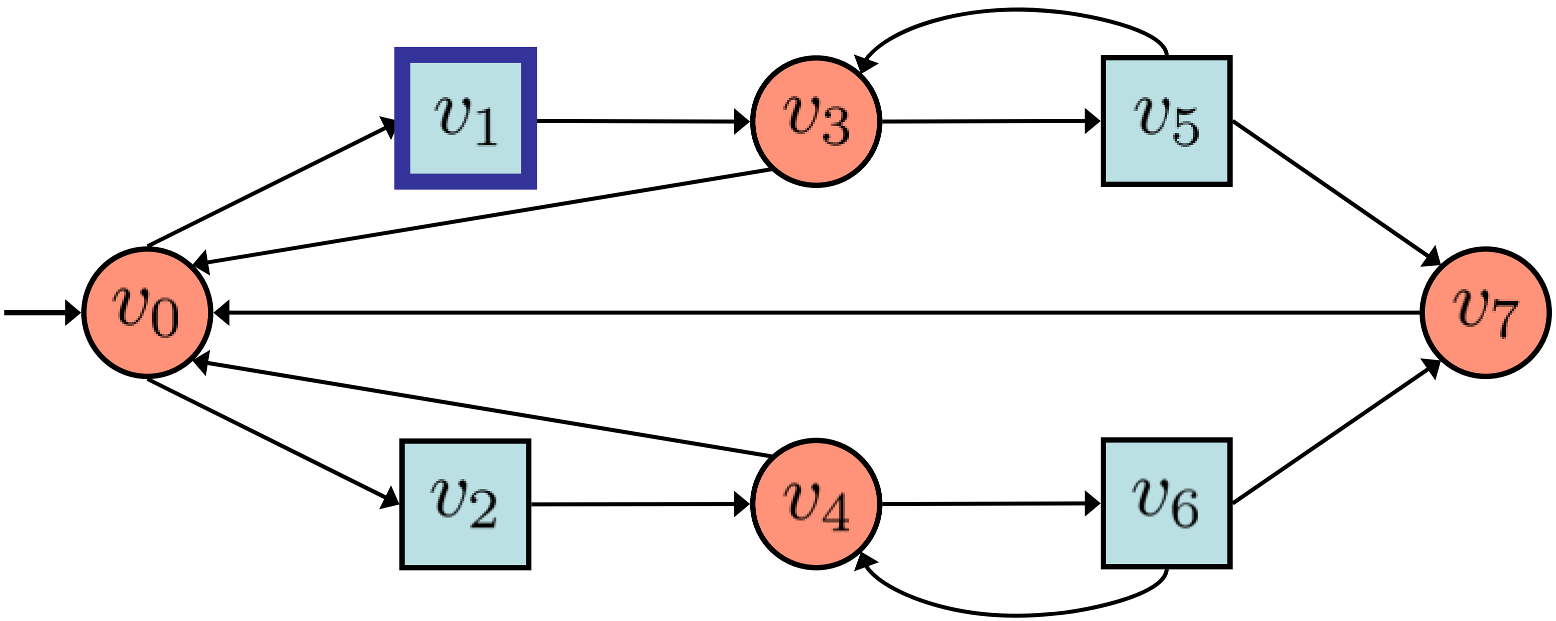


Play: v_0



belongs to Player 1

belongs to Player 2

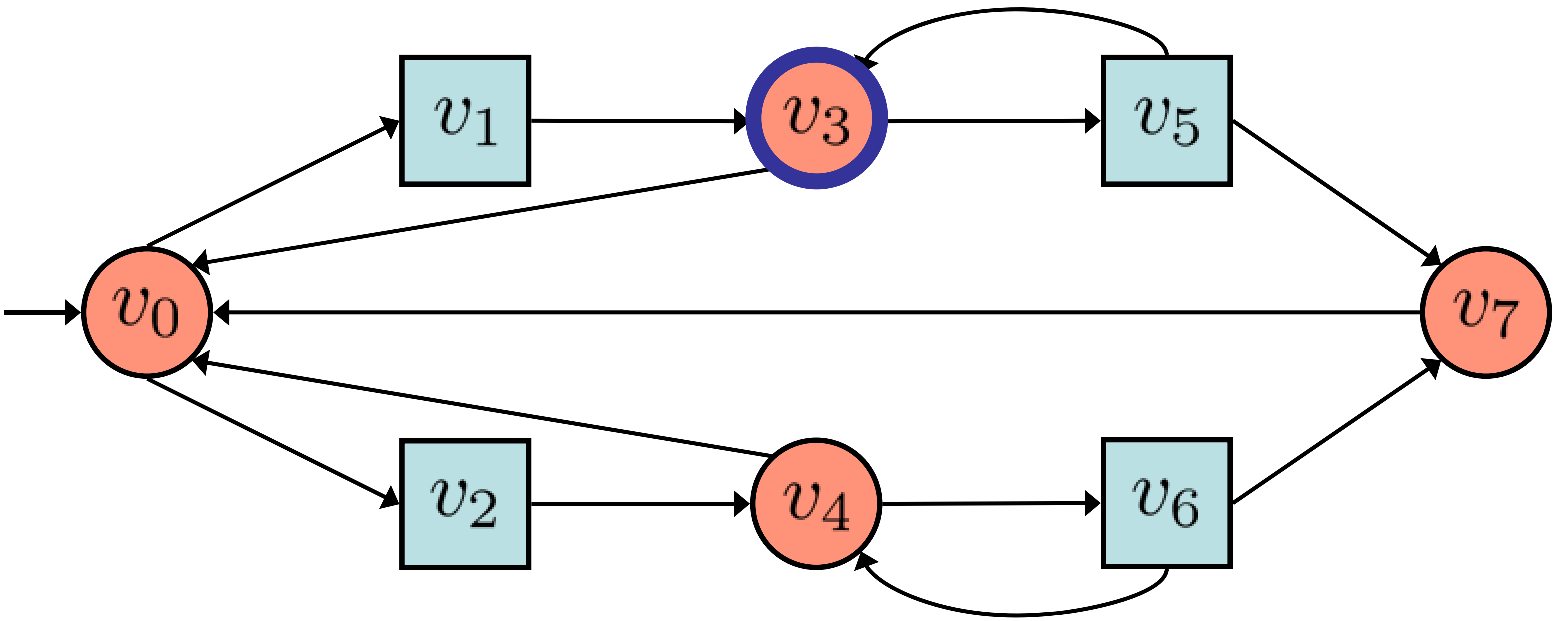


Play: v_0 v_1

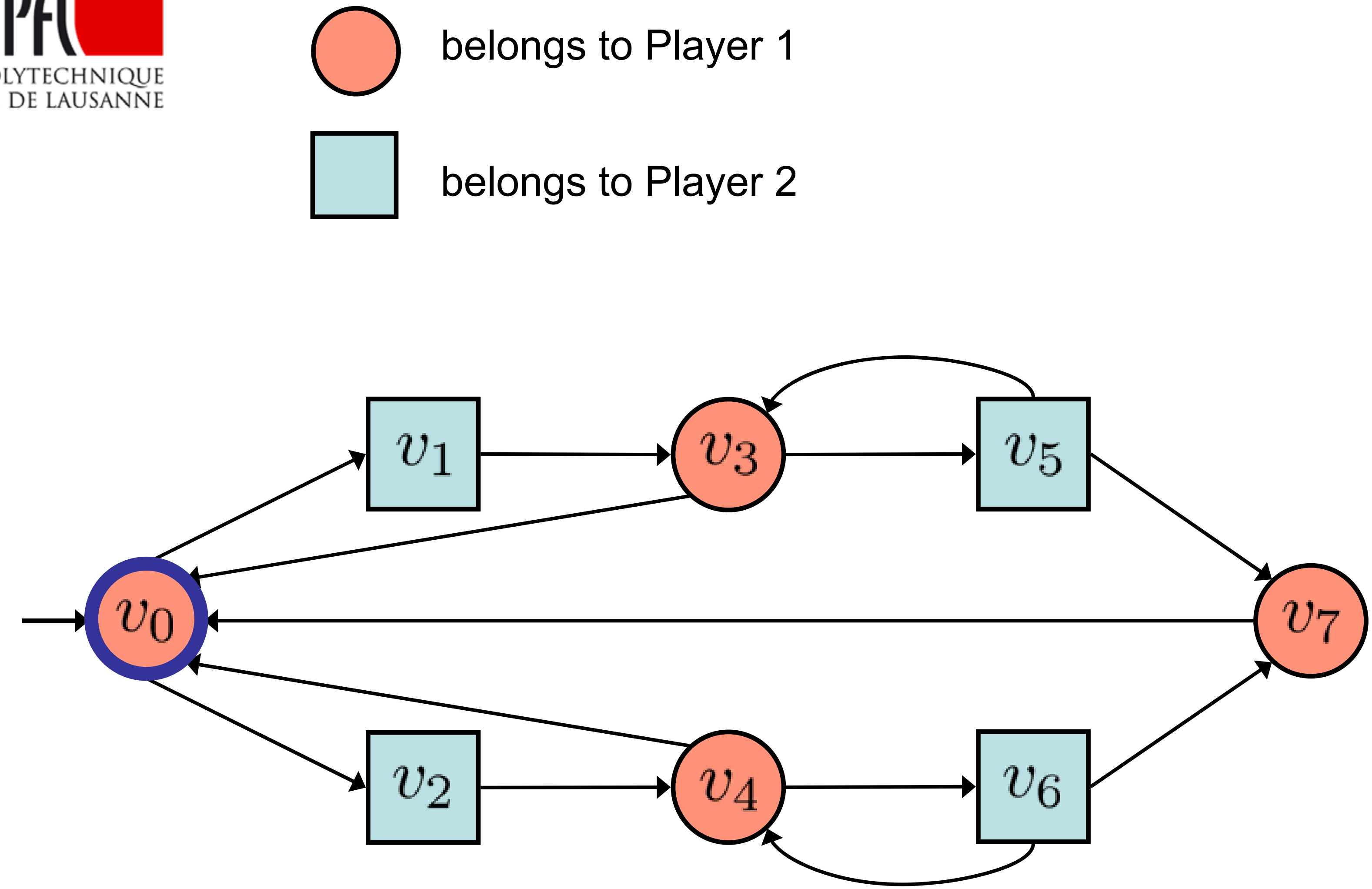
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

belongs to Player 1

belongs to Player 2



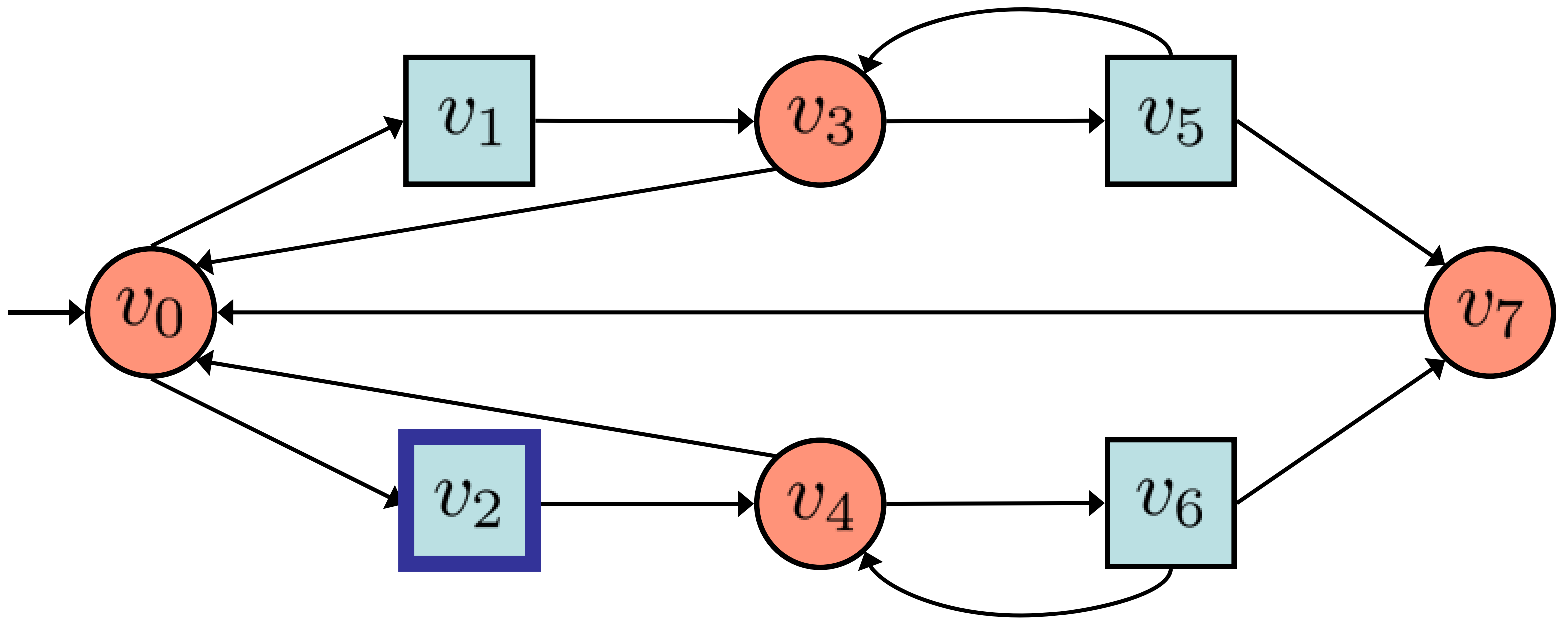
Play: $v_0\ v_1\ v_3$



Play: v_0 v_1 v_3 v_0

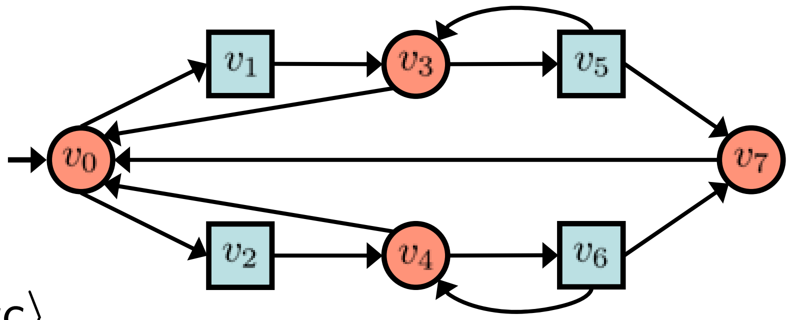
belongs to Player 1

belongs to Player 2



Play: $v_0\ v_1\ v_3\ v_0\ v_2\ \ldots$

Two-player game graphs

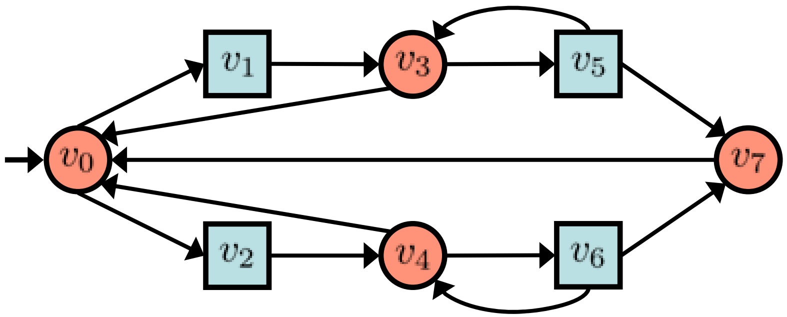


A **2-player game graph** $G = \langle V_1, V_2, \widehat{v}, \mathsf{Succ}\rangle$ consists of:

- V_1 the set of Player 1 states,
- V_2 the set of Player 2 states,

  with $V_1 \cap V_2 = \varnothing$ and $V := V_1 \cup V_2$;
- $\hat{v} \in V$ the initial state,
- $\mathsf{Succ} : V \to 2^V \setminus \varnothing$ the transition relation.

Two-player game graphs



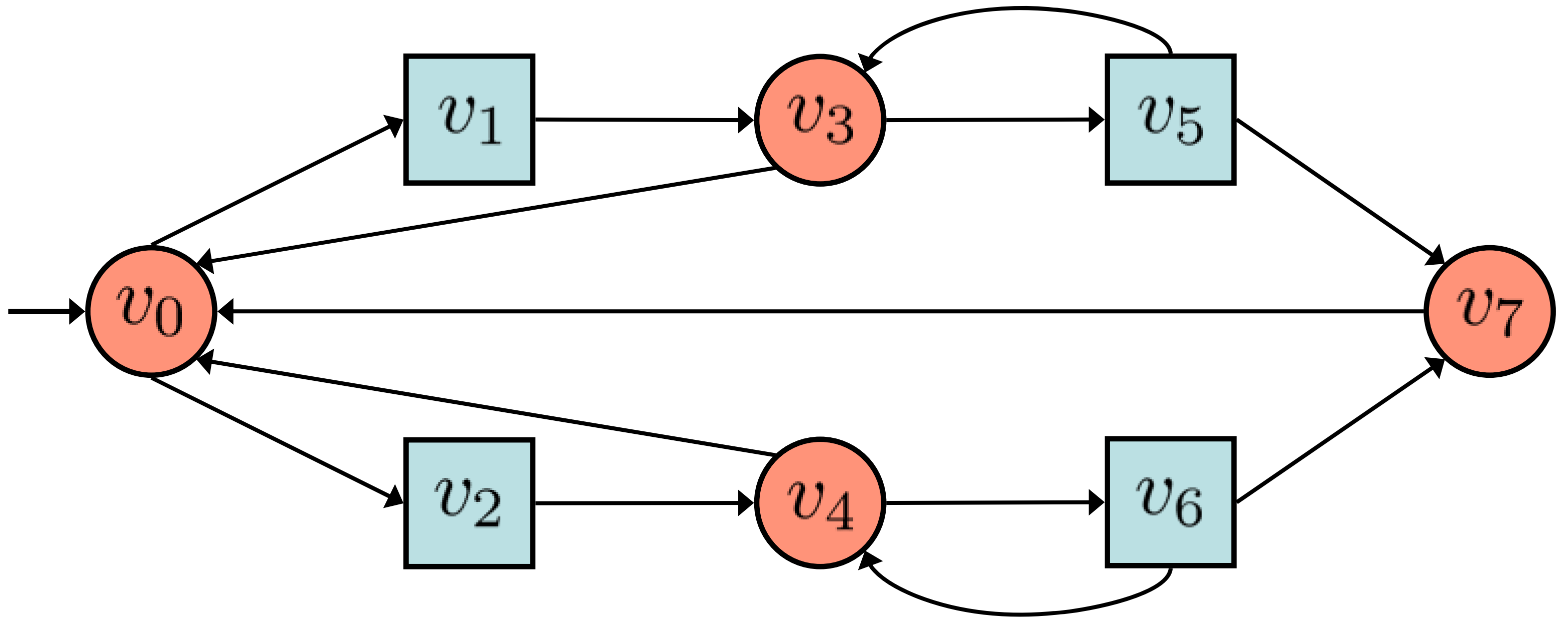
A **play** in $G = \langle V_1, V_2, \hat{v}, \mathsf{Succ} \rangle$ is an infinite sequence $w = v_0 v_1 v_2 \cdots \in V^\omega$ such that:

$V = V_1 \cup V_2$

$\mathsf{Succ} : V \to 2^V \setminus \varnothing$

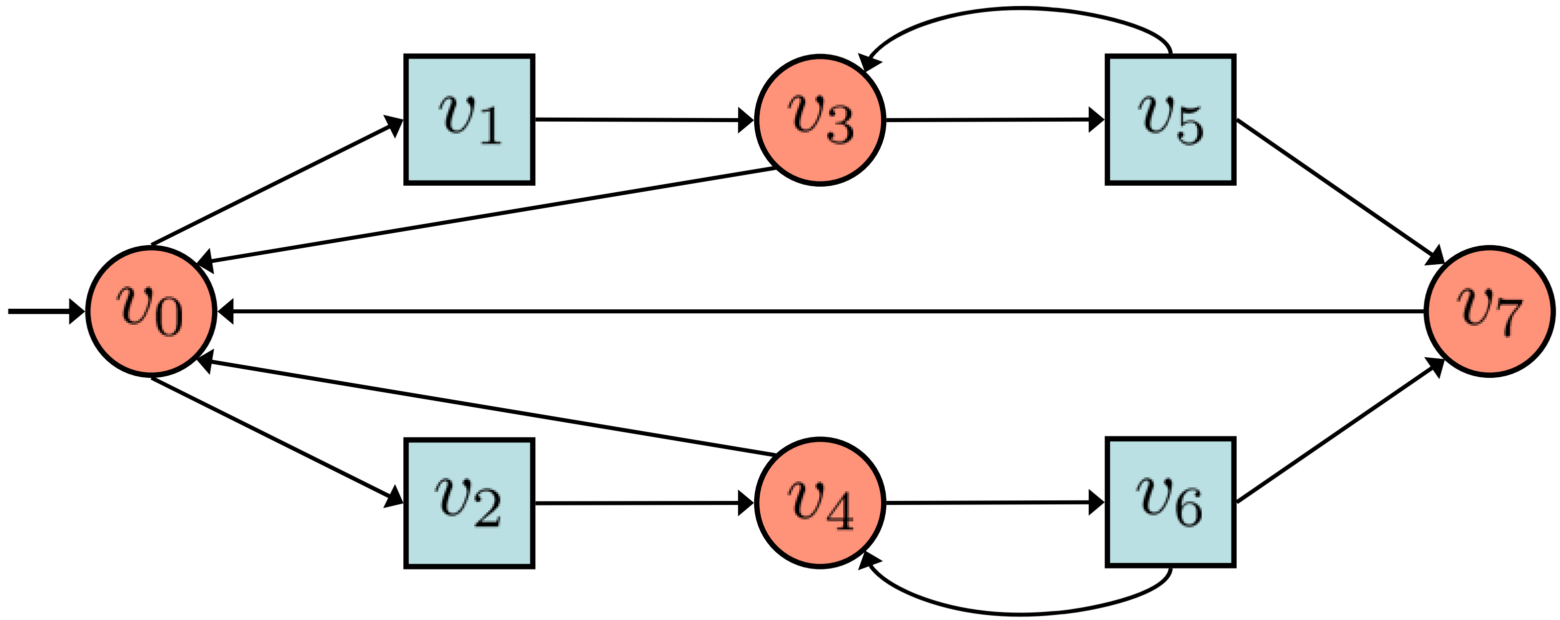
1. $v_0 = \hat{v}$,
2. $v_{i+1} \in \mathsf{Succ}(v_i)$ for all $i \geq 0$.

Who is winning ?



Play: $v_0\ v_1\ v_3\ v_0\ v_2\ \ldots$

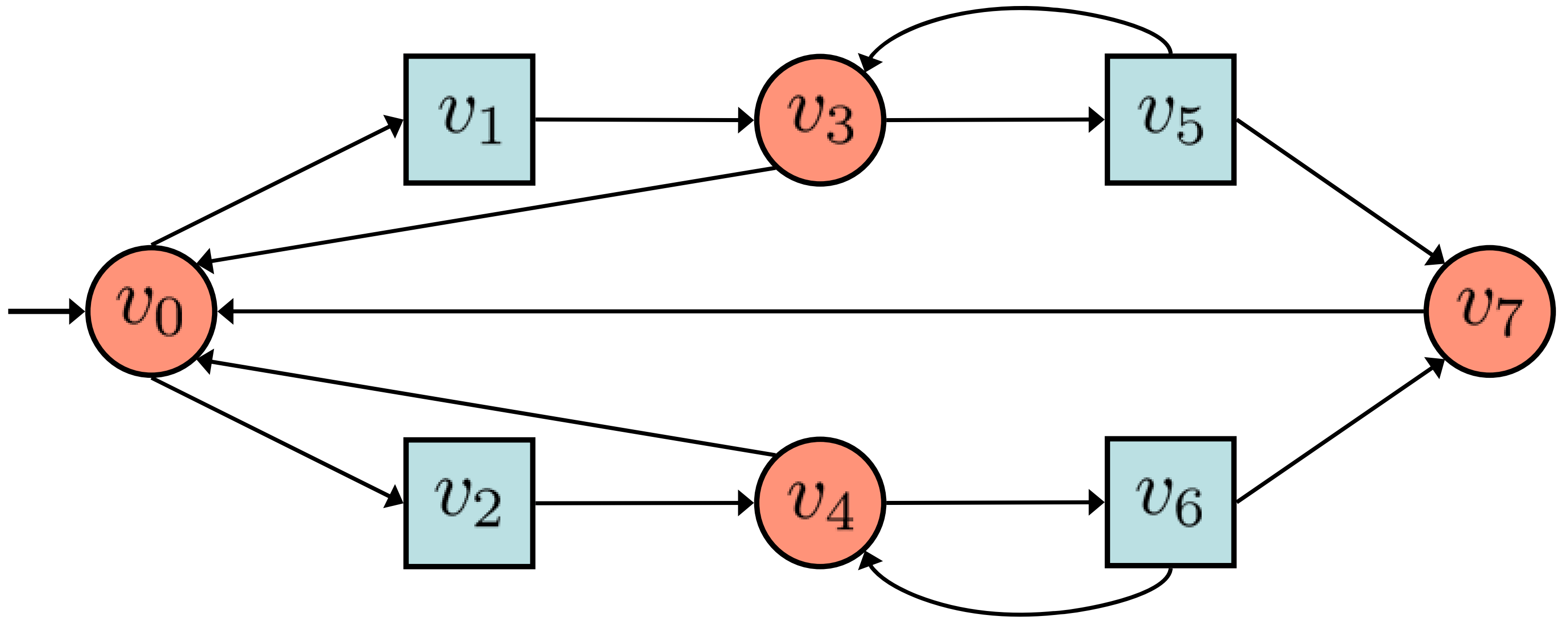
Who is winning ?



Play: $v_0\ v_1\ v_3\ v_0\ v_2\ \ldots$

A **winning condition** for Player k is a set $W_k \subseteq V^\omega$ of plays.

Who is winning ?



A **winning condition** for Player k is a set $W_k \subseteq V^\omega$ of plays.

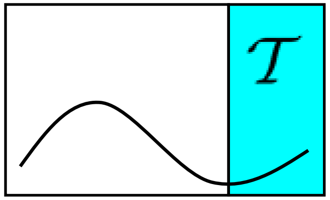
A 2-player game is **zero-sum** if $W_2 = V^\omega \setminus W_1$.

Winning condition

A **winning condition** for Player k is a set $W_k \subseteq V^\omega$ of plays.

Given $\mathcal{T} \subseteq V$, let

- $\mathsf{Reach}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \exists i : v_i \in \mathcal{T}\}$ Touch $\mathcal{T}$ eventually



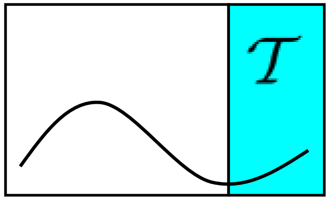
Reachability

Winning condition

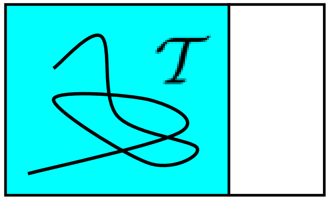
A **winning condition** for Player k is a set $W_k \subseteq V^\omega$ of plays.

Given $\mathcal{T} \subseteq V$, let

- $\mathsf{Reach}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \exists i : v_i \in \mathcal{T}\}$ Touch $\mathcal{T}$ eventually
- $\mathsf{Safe}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \forall i : v_i \in \mathcal{T}\}$ Avoid $V \setminus \mathcal{T}$ forever



Reachability



Safety

Winning condition

A **winning condition** for Player k is a set $W_k \subseteq V^\omega$ of plays.

Given $\mathcal{T} \subseteq V$, let

- $\mathsf{Reach}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \exists i : v_i \in \mathcal{T}\}$ — Touch $\mathcal{T}$ eventually
- $\mathsf{Safe}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \forall i : v_i \in \mathcal{T}\}$ — Avoid $V \setminus \mathcal{T}$ forever
- $\mathsf{Büchi}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \forall j \cdot \exists i \geq j : v_i \in \mathcal{T}\}$ — Visit $\mathcal{T}$ ∞-often
- $\mathsf{coBüchi}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \exists j \cdot \forall i \geq j : v_i \in \mathcal{T}\}$ — Visit $V \setminus \mathcal{T}$ finitely often

Reachability — Safety — Büchi — coBüchi

Remark

A **winning condition** for Player k is a set $W_k \subseteq V^\omega$ of plays.

Reach($\mathcal{T}$), Safe($\mathcal{T}$), Büchi($\mathcal{T}$) and coBüchi($\mathcal{T}$) are subsumed by the **parity** condition:

- Given a priority function $p : V \to \mathbb{N}$, define

$$\text{Parity}(p) = \{v_0 v_1 \cdots \mid \min\{d \mid \forall i \cdot \exists j \geq i : p(v_j) = d\} \text{ is even}\}$$

"Minimal priority seen ∞-often is even"

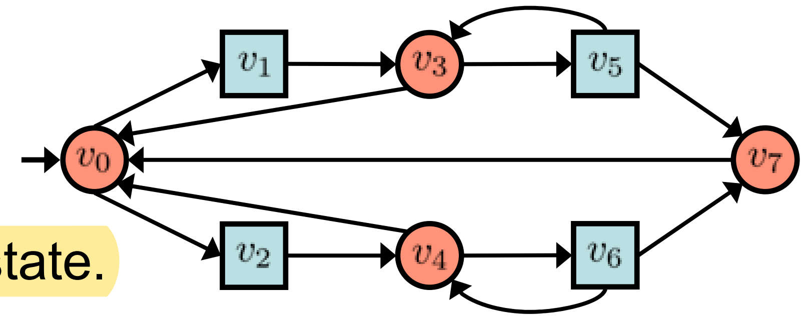
$$\begin{tikzcd} \boxed{p{=}4} \arrow[r] & \boxed{p{=}1} \arrow[r] & \boxed{p{=}3} \arrow[r] & \boxed{p{=}1} \arrow[r] & \boxed{p{=}0} \arrow[l, bend right] \end{tikzcd} \quad \in \text{Parity}(p)$$

$$\begin{tikzcd} \boxed{p{=}2} \arrow[r] & \boxed{p{=}0} \arrow[r] & \boxed{p{=}3} \arrow[r] & \boxed{p{=}1} \arrow[r] & \boxed{p{=}2} \arrow[ll, bend right] \end{tikzcd} \quad \notin \text{Parity}(p)$$

Strategies

Players use strategies to play the game,

i.e. to choose the successor of the current state.



$G = \langle V_1, V_2, \hat{v}, \mathsf{Succ} \rangle$

A **strategy for Player k** is a function:

$$\boxed{\lambda : V^* V_k \to V}$$

such that

$$\boxed{\lambda(v_1 v_2 \ldots v_n) \in \mathsf{Succ}(v_n) \text{ for all } v_1, \ldots, v_{n-1} \in V \text{ and } v_n \in V_k}$$

Strategies outcome

Graph: nondeterministic generator of behaviors.

Strategy: deterministic selector of behavior.

Graph + Strategies for both players $\longrightarrow$ Behavior

Strategies outcome

Given strategies λ_k for Player k ($k = 1, 2$), the **outcome** of $\langle \lambda_1, \lambda_2 \rangle$ is the play $w = v_0 v_1 \ldots$ such that:

$$\boxed{v_i \in V_k \rightarrow v_{i+1} = \lambda_k(v_0 \ldots v_i)}$$

for all $i \geq 0$ and $k \in \{1, 2\}$

This play is denoted $\boxed{\mathsf{Outcome}(G, \lambda_1, \lambda_2)}$

Winning strategies

- Given a game G and winning conditions W_1 and W_2,

  a strategy λ_k is **winning** for Player k in (G,W_k) if for all strategies $\lambda_{3\text{-}k}$ of Player 3-k, the outcome of $\{\lambda_k, \lambda_{3\text{-}k}\}$ in G is a winning play of W_k.

- Player 1 is winning if $\boxed{\exists\lambda_1 \cdot \forall\lambda_2 : \mathsf{Outcome}(G, \lambda_1, \lambda_2) \in W_1}$

- Player 2 is winning if $\boxed{\exists\lambda_2 \cdot \forall\lambda_1 : \mathsf{Outcome}(G, \lambda_1, \lambda_2) \in W_2}$

Winning strategies

=

Controllers that enforce winning plays

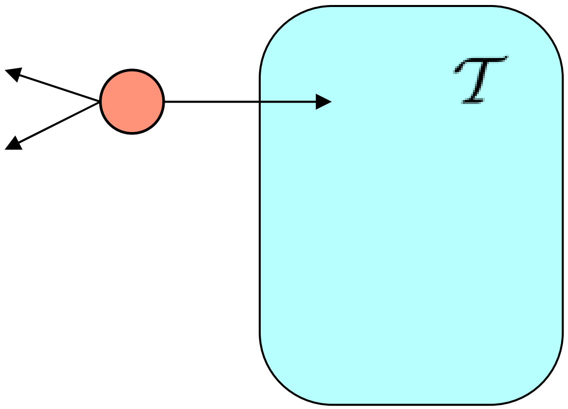
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Symbolic algorithms to solve games

Controllable predecessors

Given $\mathcal{T} \subseteq V$, let

- $\exists\mathsf{CPre}(\mathcal{T}) = \{v \in V \mid \exists v' \in \mathsf{Succ}(v) : v' \in \mathcal{T}\}$

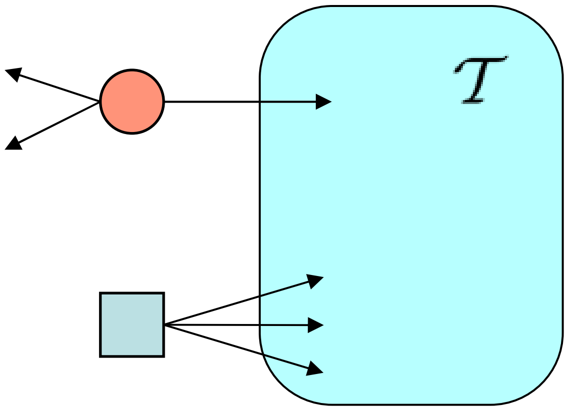


Controllable predecessors

Given $\mathcal{T} \subseteq V$, let

- $\exists \mathsf{CPre}(\mathcal{T}) = \{v \in V \mid \exists v' \in \mathsf{Succ}(v) : v' \in \mathcal{T}\}$

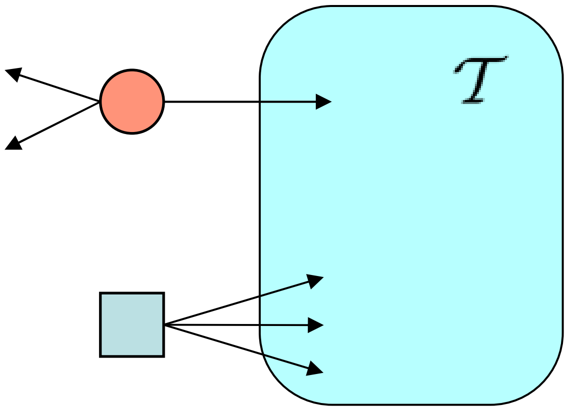
- $\forall \mathsf{CPre}(\mathcal{T}) = \{v \in V \mid \forall v' \in \mathsf{Succ}(v) : v' \in \mathcal{T}\}$



Controllable predecessors

Given $\mathcal{T} \subseteq V$, let

- $\exists \mathsf{CPre}(\mathcal{T}) = \{v \in V \mid \exists v' \in \mathsf{Succ}(v) : v' \in \mathcal{T}\}$
- $\forall \mathsf{CPre}(\mathcal{T}) = \{v \in V \mid \forall v' \in \mathsf{Succ}(v) : v' \in \mathcal{T}\}$



From a state v, Player 1 can **force** the next position of the game to be in $\mathcal{T}$ if:

$$v \in \underbrace{(\exists \mathsf{CPre}(\mathcal{T}) \cap V_1) \cup (\forall \mathsf{CPre}(\mathcal{T}) \cap V_2)}_{1\mathsf{CPre}(\mathcal{T})}$$

Controllable predecessors

$$1\mathsf{CPre}(\mathcal{T}) := (\exists\mathsf{CPre}(\mathcal{T}) \cap V_1) \;\cup\; (\forall\mathsf{CPre}(\mathcal{T}) \cap V_2)$$

and symmetrically

$$2\mathsf{CPre}(\mathcal{T}) := (\forall\mathsf{CPre}(\mathcal{T}) \cap V_1) \;\cup\; (\exists\mathsf{CPre}(\mathcal{T}) \cap V_2)$$

Note: $\mathcal{T}' \subseteq \mathcal{T}$ implies $\begin{cases} 1\mathsf{CPre}(\mathcal{T}') \subseteq 1\mathsf{CPre}(\mathcal{T}) \\ 2\mathsf{CPre}(\mathcal{T}') \subseteq 2\mathsf{CPre}(\mathcal{T}) \end{cases}$

$1\mathsf{CPre}(\cdot)$ and $2\mathsf{CPre}(\cdot)$ are **monotone** functions.

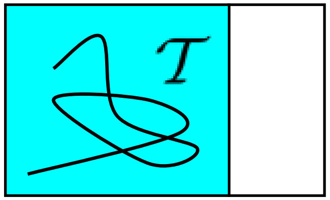
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Symbolic algorithm to solve **safety** games

Solving safety games

$\mathsf{Safe}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \forall i : v_i \in \mathcal{T}\}$

Avoid $V \setminus \mathcal{T}$ forever



To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

Solving safety games

To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

Solving safety games

To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

1 step: $X_1 = \mathcal{T} \cap \mathsf{1CPre}(\mathcal{T})$

Solving safety games

To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

1 step: $X_1 = \mathcal{T} \cap \mathsf{1CPre}(\mathcal{T})$

2 steps: $X_2 = \mathcal{T} \cap \mathsf{1CPre}(\mathcal{T}) \cap \mathsf{1CPre}(\mathcal{T} \cap \mathsf{1CPre}(\mathcal{T}))$

Solving safety games

To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

1 step: $X_1 = \mathcal{T} \cap \mathsf{1CPre}(\mathcal{T})$

2 steps: $X_2 = \mathcal{T} \cap \cancel{\mathsf{1CPre}(\mathcal{T})} \cap \mathsf{1CPre}(\underbrace{\mathcal{T} \cap \mathsf{1CPre}(\mathcal{T})}_{\text{subset of } \mathcal{T}})$

Solving safety games

> To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

1 step: $X_1 = \mathcal{T} \cap \mathsf{1CPre}(\mathcal{T})$

2 steps: $X_2 = \mathcal{T} \cap \mathsf{1CPre}(\mathcal{T} \cap \mathsf{1CPre}(\mathcal{T}))$

Solving safety games

To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

1 step: $X_1 = \mathcal{T} \cap \mathsf{1CPre}(X_0)$

2 steps: $X_2 = \mathcal{T} \cap \mathsf{1CPre}(X_1)$

Solving safety games

To win a safety game, Player 1 should be able to force the game to be in $\mathcal{T}$ at every step.

States in which Player 1 can force the game to stay in $\mathcal{T}$ for the next:

0 step: $X_0 = \mathcal{T}$

1 step: $X_1 = \mathcal{T} \cap \mathsf{1CPre}(X_0)$

2 steps: $X_2 = \mathcal{T} \cap \mathsf{1CPre}(X_1)$

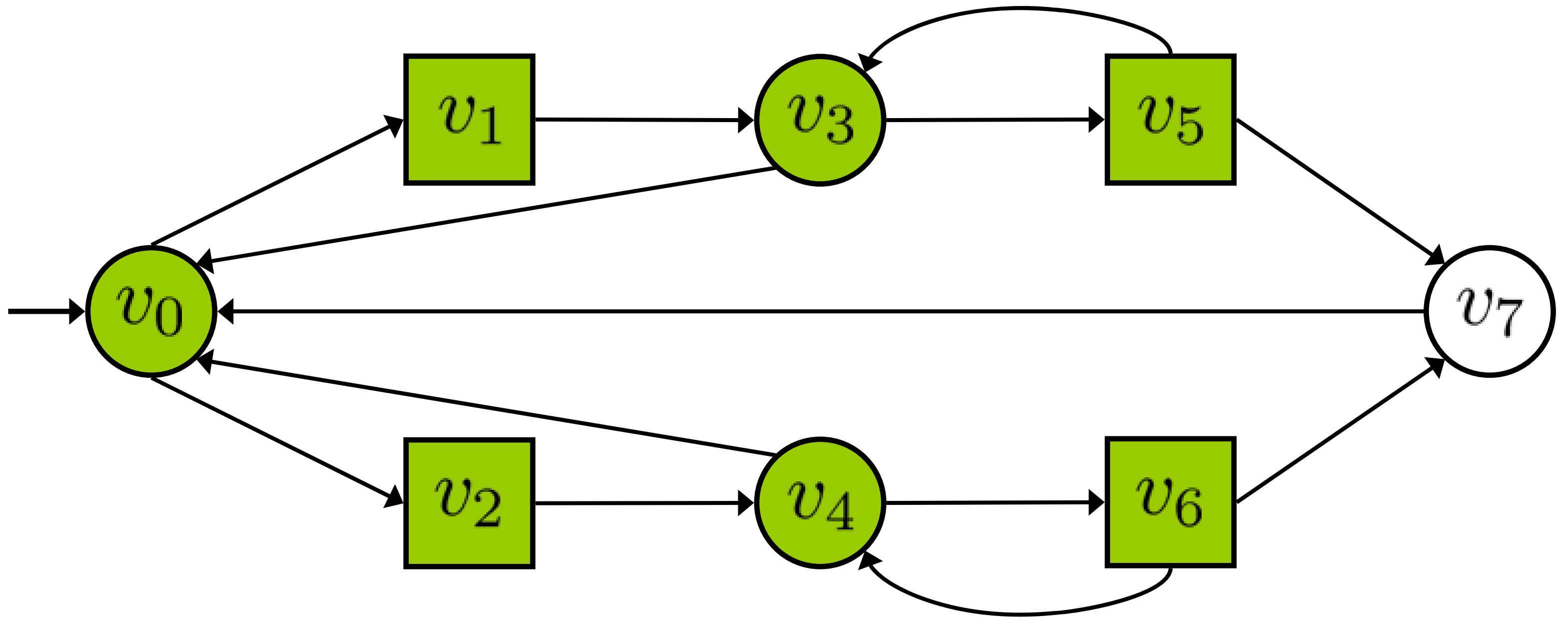
$\vdots$

n steps: $X_n = \mathcal{T} \cap \mathsf{1CPre}(X_{n-1})$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

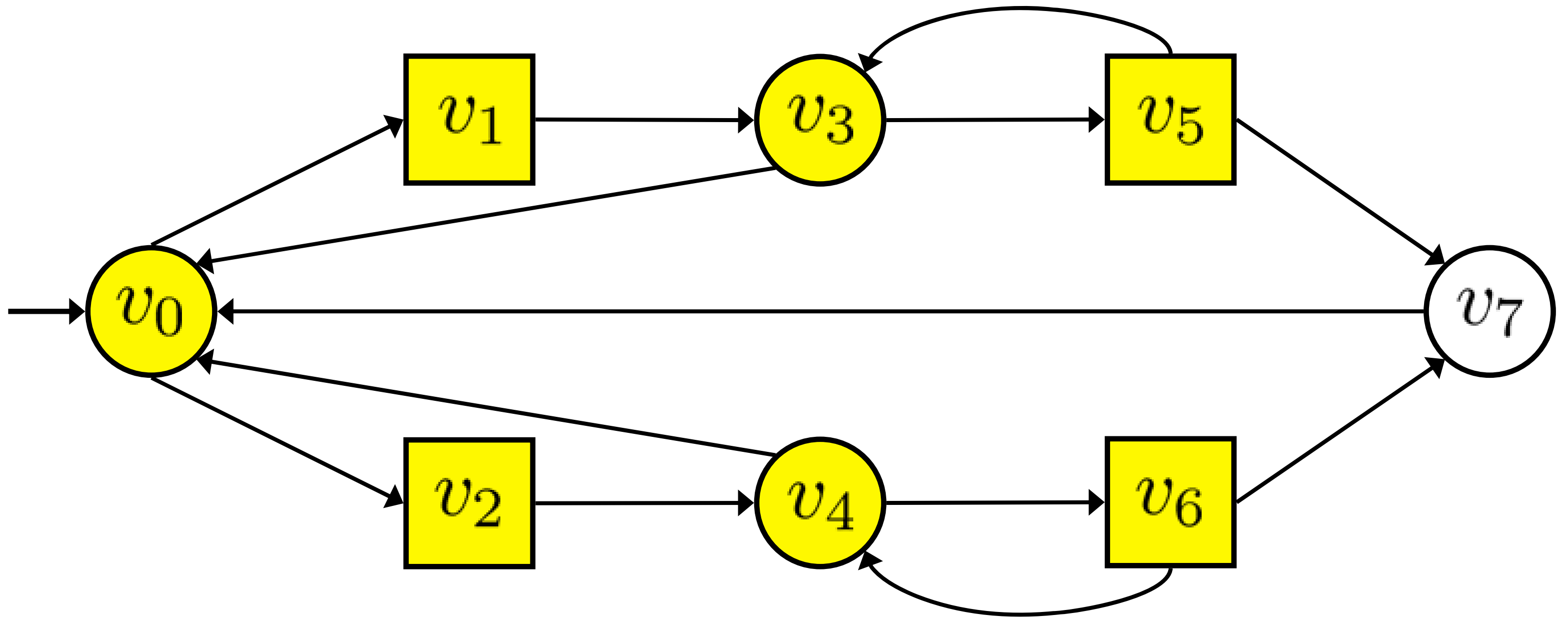
Objective: $\mathsf{Safe}(\mathcal{T})$



Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

Objective: Safe($\mathcal{T}$)

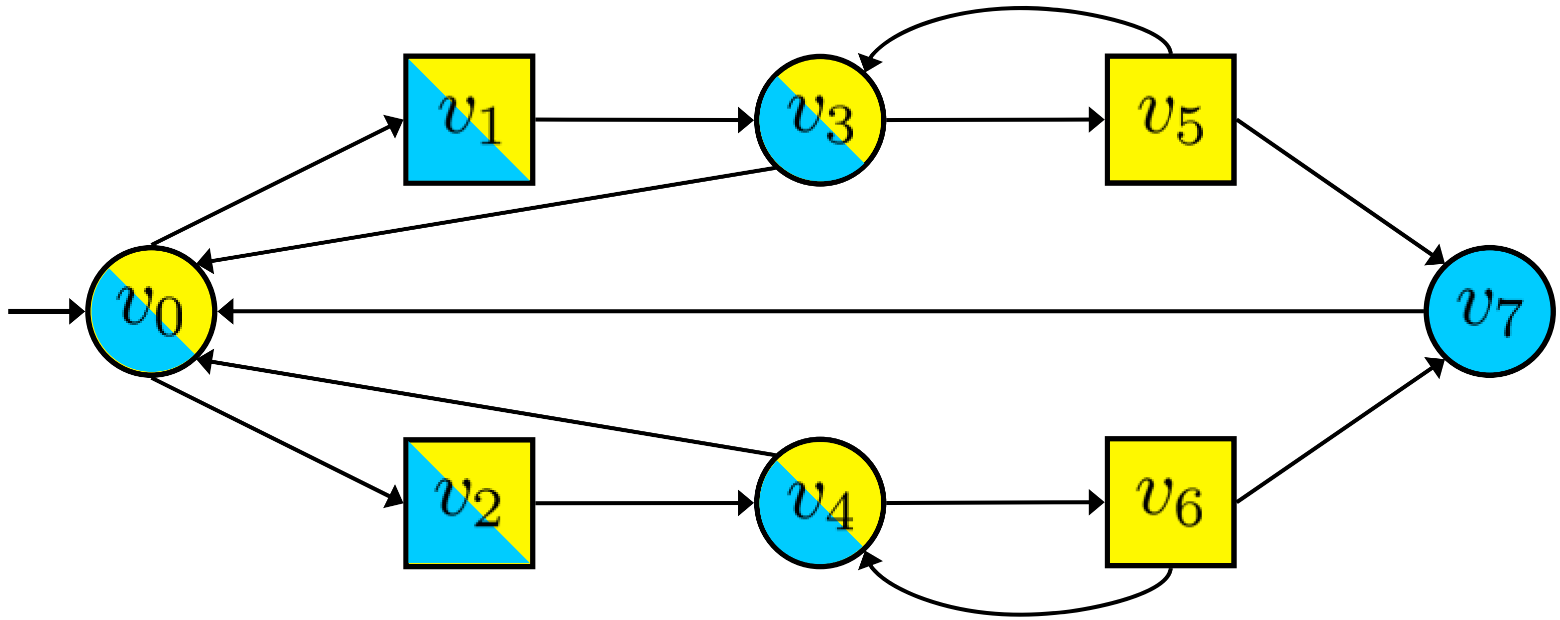


$\boxed{X_0} = \mathcal{T}$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

Objective: $\mathsf{Safe}(\mathcal{T})$

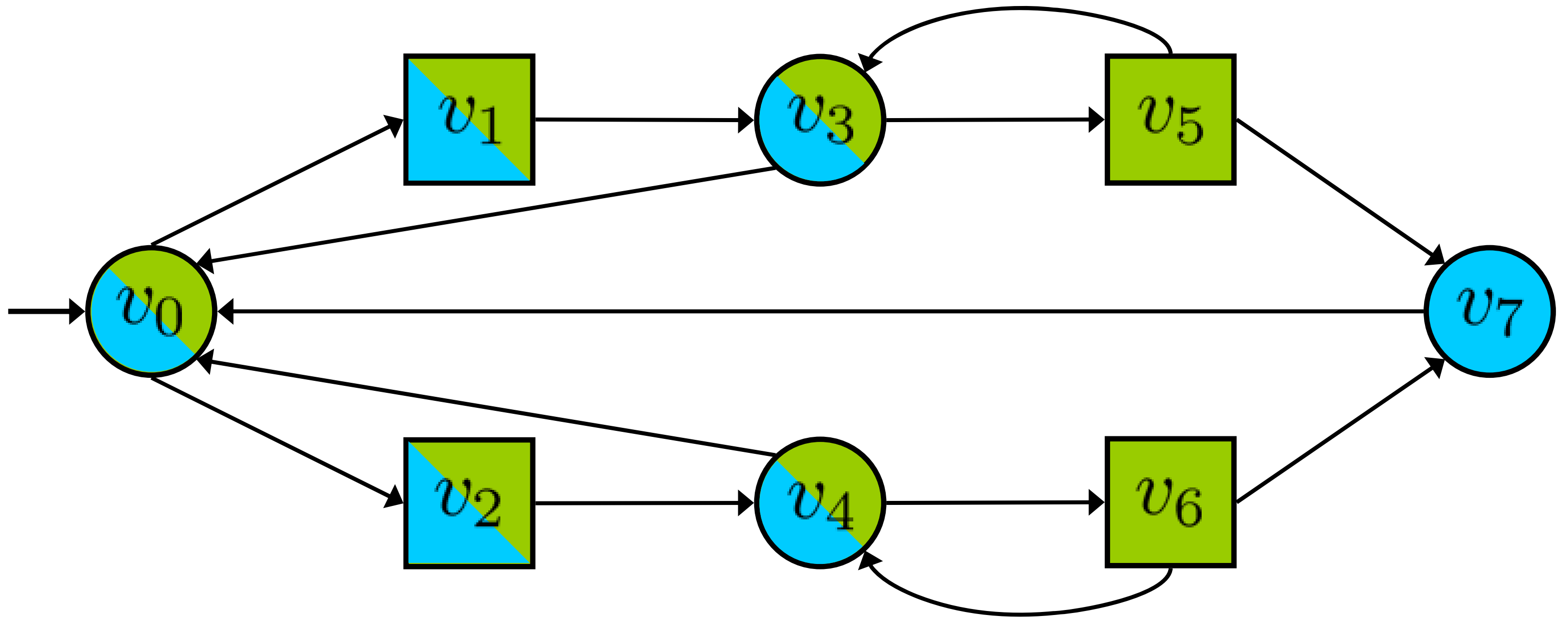


$X_0 = \mathcal{T}$

$X_1 = \mathcal{T} \cap \boxed{\mathsf{1CPre}(X_0)}$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$ Objective: $\mathsf{Safe}(\mathcal{T})$



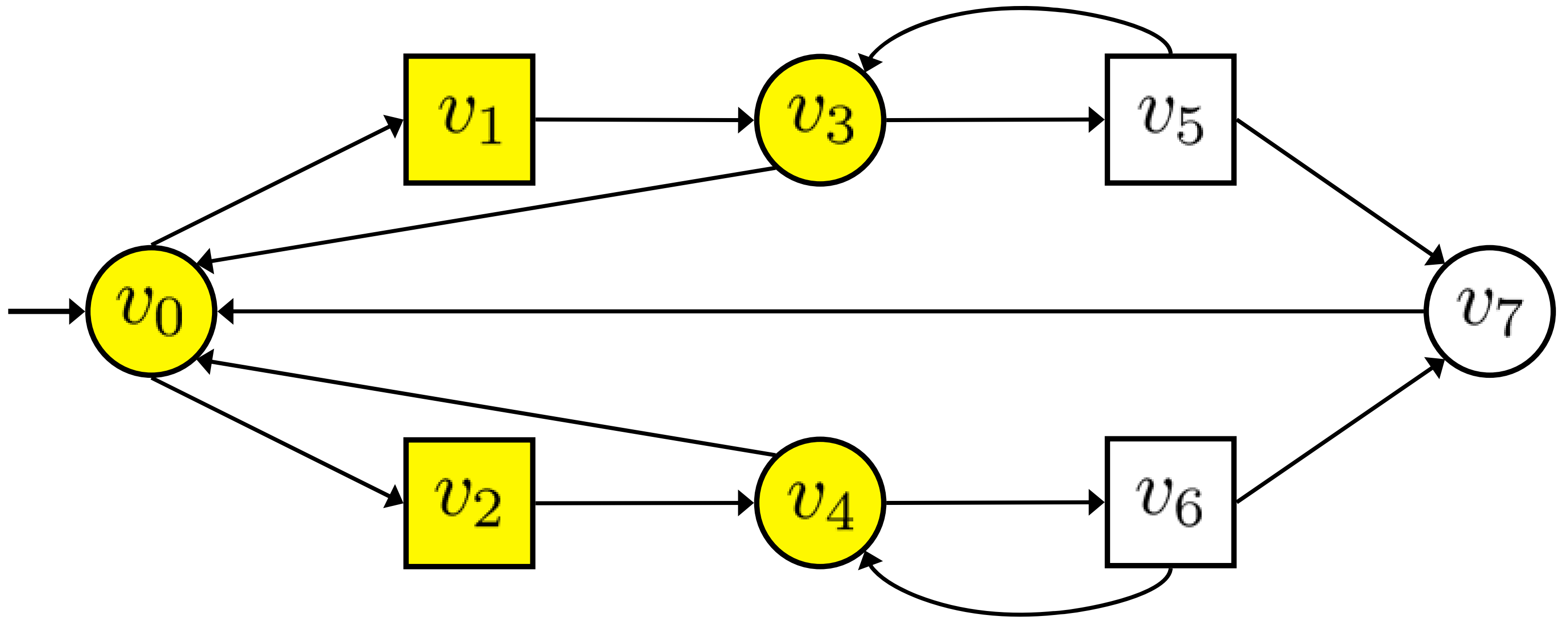
$X_0 = \mathcal{T}$

$X_1 = \boxed{\mathcal{T}} \cap \boxed{\mathsf{1CPre}(X_0)}$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



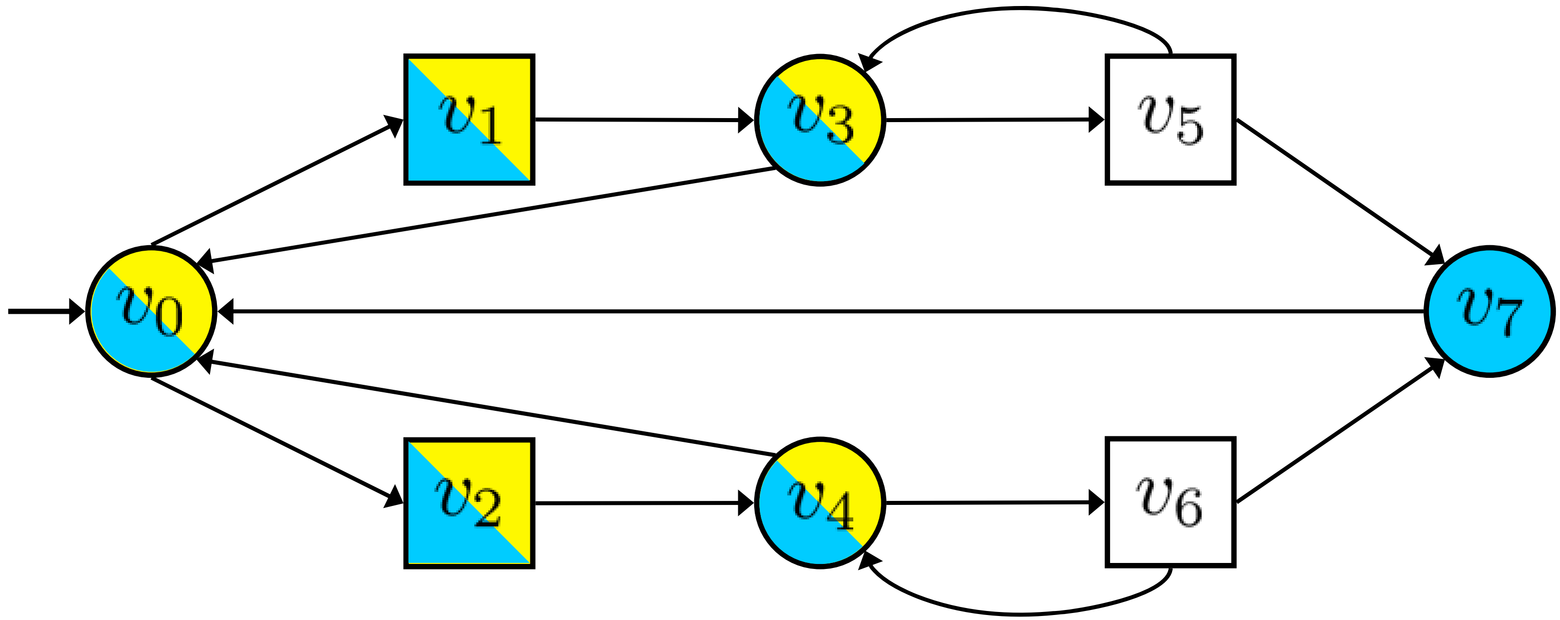
$X_0 = \mathcal{T}$

$\boxed{X_1} = \mathcal{T} \cap \mathsf{1CPre}(X_0)$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$X_0 = \mathcal{T}$

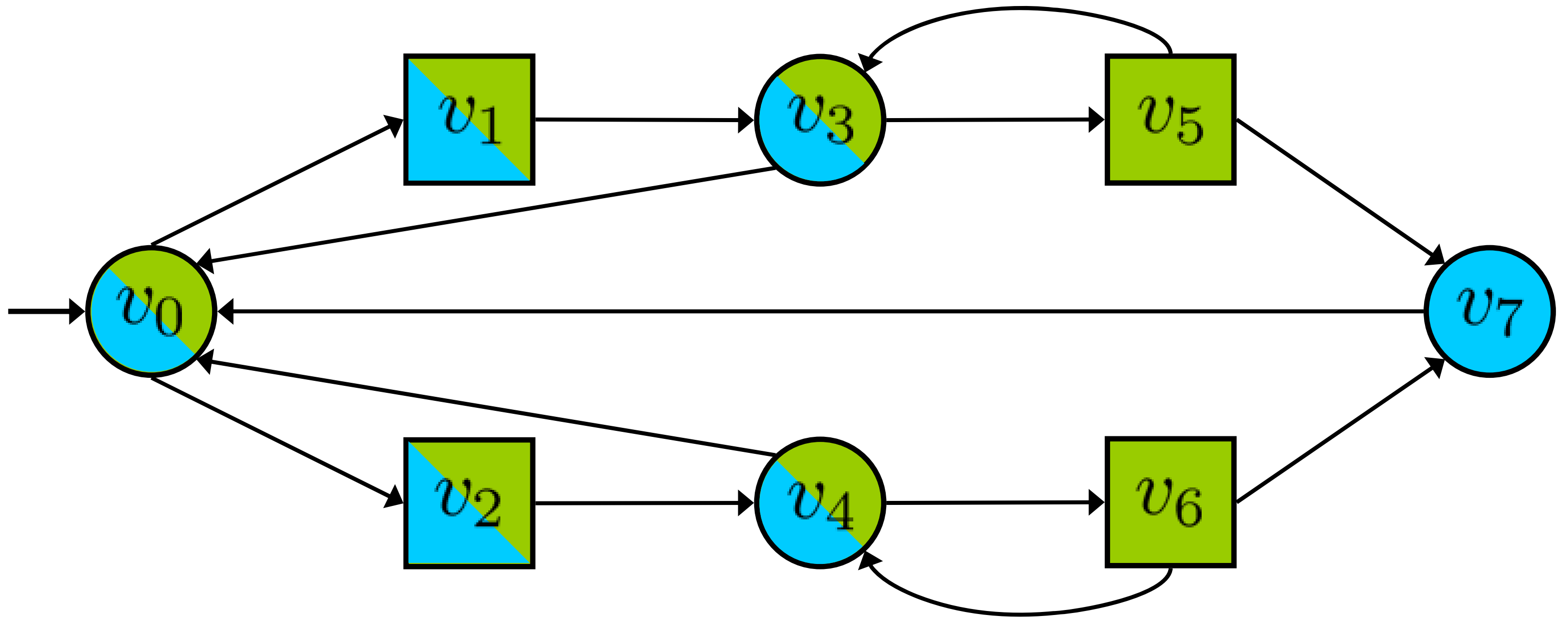
$\boxed{X_1} = \mathcal{T} \cap \mathsf{1CPre}(X_0)$

$X_2 = \mathcal{T} \cap \boxed{\mathsf{1CPre}(X_1)}$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



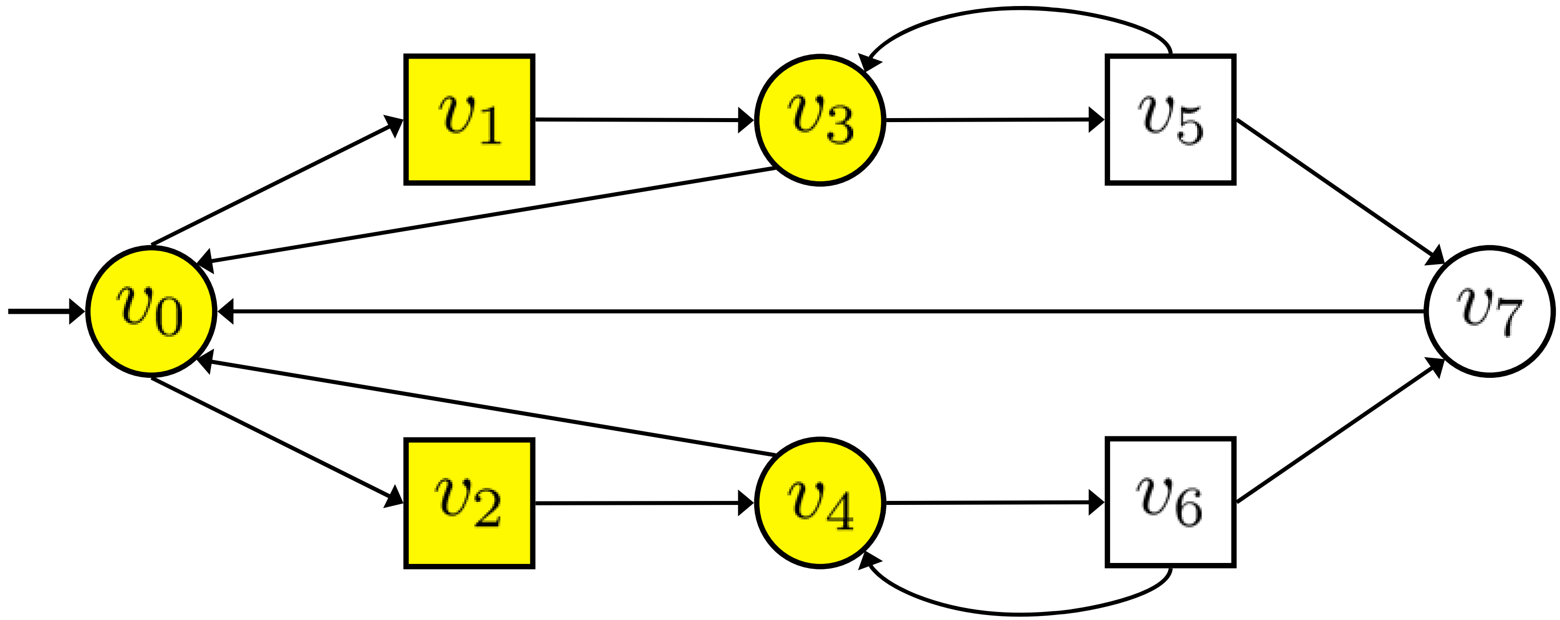
$X_0 = \mathcal{T}$

$X_1 = \mathcal{T} \cap \mathsf{1CPre}(X_0)$

$X_2 = \boxed{\mathcal{T}} \cap \boxed{\mathsf{1CPre}(X_1)}$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$ Objective: Safe($\mathcal{T}$)



$X_0 = \mathcal{T}$

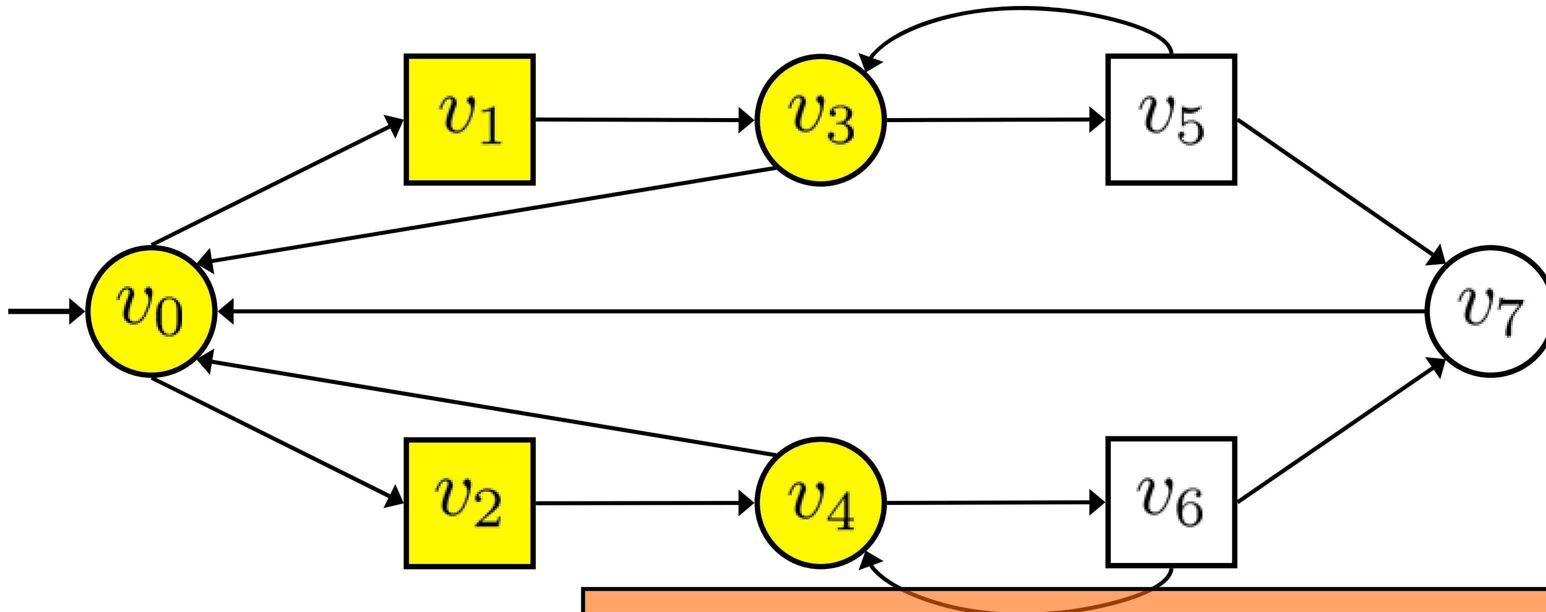
$X_1 = \mathcal{T} \cap 1\mathsf{CPre}(X_0)$

$\boxed{X_2} = \mathcal{T} \cap 1\mathsf{CPre}(X_1)$

Solving safety games

$\mathcal{T} = V \setminus \{v_7\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



This is the set of states from which Player 1 can confine the game in $\mathcal{T}$ forever no matter how Player 2 behaves.

$X_0 = \mathcal{T}$

$X_1 = \mathcal{T} \cap \mathsf{1CPre}(X_0)$

$\boxed{X_2} = \mathcal{T} \cap \mathsf{1CPre}(X_1) = \boxed{X_1}$

Solving safety games

X_2 is a solution of the set-equation $X = \mathcal{T} \cap \mathsf{1CPre}(X)$

and it is the greatest solution.

Solving safety games

X_2 is a solution of the set-equation $X = \mathcal{T} \cap \mathsf{1CPre}(X)$

and it is the greatest solution.

We say that X_2 is the **greatest fixpoint** of the function $\mathcal{T} \cap \mathsf{1CPre}(\cdot)$, written:

$$\boxed{X_2 = \underbrace{\nu X}_{\text{greatest fixpoint operator}} \cdot \mathcal{T} \cap \mathsf{1CPre}(X)}$$

On fixpoint computations

Partial order

A partially ordered set $\langle S, \sqsubseteq \rangle$ is a set S equipped with a **partial order** $\sqsubseteq$, *i.e.* a relation such that:

$\forall x$	$x \sqsubseteq x$	(reflexivity)
$\forall x, y, z$	if $x \sqsubseteq y$ and $y \sqsubseteq z$ then $x \sqsubseteq z$	(transitivity)
$\forall x, y$	if $x \sqsubseteq y$ and $y \sqsubseteq x$ then $x = y$	(anti-symmetry)

$\sqsubseteq$ is not necessarily total, *i.e.* there can be x, y such that $x \not\sqsubseteq y$ and $y \not\sqsubseteq x$.

Partial order

Let $X \subseteq S$.

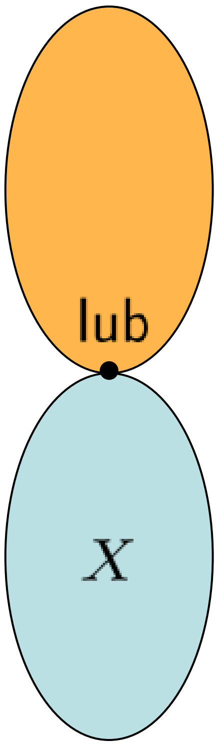
y is an **upper bound** of X if $x \sqsubseteq y$ for all $x \in X$.

y is a **least upper bound** of X if

(1) y is an upper bound of X, and

(2) $y \sqsubseteq y'$ for all upper bounds y' of X.

Note: if X has a least upper bound, then it is unique (by anti-symmetry), and we write $y = \mathsf{lub}(X)$.



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Partial order

Examples: $\langle \mathbb{N}, \leq \rangle$

$X = \{3, 5, 7, 8\}$ $\qquad$ $\mathsf{lub}(X) = 8$

$X = \{1, 3, 5, 7, 9, \ldots\}$ $\qquad$ X has no lub

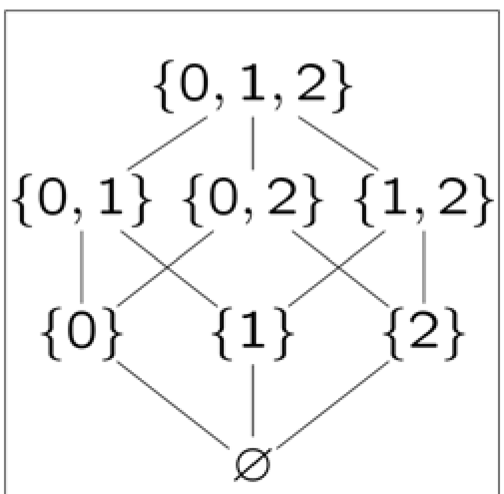
Partial order

Examples: $\langle \mathbb{N}, \leq \rangle$

$X = \{3, 5, 7, 8\}$ $\quad$ $\mathrm{lub}(X) = 8$

$X = \{1, 3, 5, 7, 9, \ldots\}$ $\quad$ X has no lub

$\langle \mathcal{P}(\{0, 1, 2\}), \subseteq \rangle$



$X = \big\{\{0\}, \{2\}\big\}$ $\quad$ $\mathrm{lub}(X) = \{0, 2\}$

Partial order

A set $X = \{x_0, x_1, x_2, \ldots\}$ is a **chain** if $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots$

The partially ordered set $\langle S, \sqsubseteq \rangle$ is **complete** if

(1) $\varnothing$ has a lub, written $\mathsf{lub}(\varnothing) = \perp$, and

(2) every chain $X \subseteq S$ has a lub.

Fixpoints

Let $f : S \to S$ be a function.

f is **monotonic** if $x \sqsubseteq y$ implies $f(x) \sqsubseteq f(y)$.

is **continuous** if(1) f is monotonic, and

(2) $f(\text{lub}(X)) = \text{lub}(f(X))$ for every chain X.

where $f(X) = \{f(x_0), f(x_1), f(x_2), \ldots\}$

Note: $f(X)$ is a chain (*i.e.* $f(x_0) \sqsubseteq f(x_1) \sqsubseteq f(x_2) \sqsubseteq \ldots$)

by monotonicity, and therefore $\text{lub}(f(X))$ exists.

Fixpoints

Let $f : S \rightarrow S$ be a function.

x is a **fixpoint** of f if $x = f(x)$

x is a **least fixpoint** of f if

(1) x is a fixpoint of f, and

(2) $x \sqsubseteq x'$ for all fixpoints x' of f.

Kleene-Tarski Theorem

Let $\langle S, \sqsubseteq \rangle$ be a partially ordered set.

If $\sqsubseteq$ is a complete partial order, and $f : S \to S$ is a continuous function, then

f has a least fixpoint, denoted $\mathsf{lfp}(f)$

and $\mathsf{lfp}(f) = \mathsf{lub}(\{\bot, f(\bot), f^2(\bot), f^3(\bot), \ldots\})$

Proof: exercise.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Kleene-Tarski Theorem

Let $\langle S, \sqsubseteq \rangle$ be a partially ordered set.

If $\sqsubseteq$ is a complete partial order, and $f : S \to S$ is a continuous function, then

f has a least fixpoint, denoted $\mathsf{lfp}(f)$

and $\mathsf{lfp}(f) = \mathsf{lub}(\{\bot, f(\bot), f^2(\bot), f^3(\bot), \dots\})$

Proof: exercise.

Over finite sets S, all monotonic functions are continuous.

Kleene-Tarski Theorem

The greatest fixpoint of f can be defined dually by

$$\mathsf{gfp}(f) = \mathsf{glb}(\{\top, f(\top), f^2(\top), f^3(\top), \ldots\})$$

where $\mathsf{glb}(\cdot)$ is the greatest lower bound operator (dual of $\mathsf{lub}(\cdot)$) and $\mathsf{glb}(\emptyset) = \top$

and $\mathsf{lfp}(f) = \mathsf{lub}(\{\perp, f(\perp), f^2(\perp), f^3(\perp), \ldots\})$

Proof: exercise.

Over finite sets S, all monotonic functions are continuous.

Safety game

Winning states of a safety game: $\boxed{\nu X \cdot \mathcal{T} \cap \mathsf{1CPre}(X)}$

$\boxed{\mathsf{gfp}(\mathcal{T} \cap \mathsf{1CPre}(X))}$

Limit of the iterations:

$$X_0 = \mathcal{T} \cap \mathsf{1CPre}(V)$$
$$X_1 = \mathcal{T} \cap \mathsf{1CPre}(X_0)$$
$$X_2 = \mathcal{T} \cap \mathsf{1CPre}(X_1)$$
$$\vdots$$

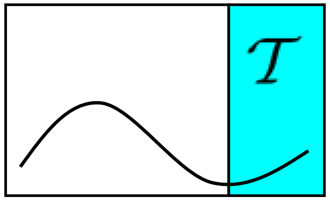
Partial order: $\langle 2^V, \subseteq \rangle$ with $\top = V$, $\bot = \varnothing$.

Symbolic algorithm to solve **reachability** games

Solving reachability games

$\mathsf{Reach}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \exists i : v_i \in \mathcal{T}\}$

Visit $\mathcal{T}$ eventually

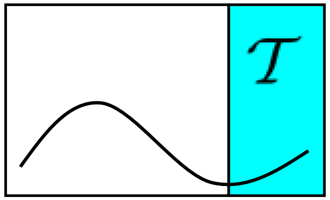


To win a reachability game, Player 1 should be able to force the game be in $\mathcal{T}$ after finitely many steps.

Solving reachability games

$\mathsf{Reach}(\mathcal{T}) = \{v_0 v_1 \cdots \in V^\omega \mid \exists i : v_i \in \mathcal{T}\}$

Visit $\mathcal{T}$ eventually



To win a reachability game, Player 1 should be able to force the game be in $\mathcal{T}$ after finitely many steps.

Let X_i be the set of states from which Player 1 can force the game to be in $\mathcal{T}$ within at most i steps:

$$X_0 = \mathcal{T}$$
$$X_{i+1} = X_i \cup \mathsf{1CPre}(X_i) \quad \text{for all } i \geq 0$$

Solving reachability games

Tthe limit of this iteration is the **least fixpoint** of the function $\mathcal{T} \cup \mathsf{1CPre}(\cdot)$, written:

$$\boxed{\underbrace{\mu X}_{\textbf{least fixpoint operator}} \cdot \mathcal{T} \cup \mathsf{1CPre}(X)}$$

Symbolic algorithms

Let $G = \langle V_1, V_2, \widehat{v}, \mathsf{Succ} \rangle$ be a 2-player game graph.

Theorem

Player 1 has a winning strategy

$$\begin{array}{lll} \text{in } \langle G, \mathsf{Reach}(\mathcal{T}) \rangle & \text{iff} & \hat{v} \in \mu X \cdot \mathcal{T} \cup \mathsf{1CPre}(X) \\ \text{in } \langle G, \mathsf{Safe}(\mathcal{T}) \rangle & \text{iff} & \hat{v} \in \nu X \cdot \mathcal{T} \cap \mathsf{1CPre}(X) \\ \text{in } \langle G, \mathsf{B\ddot{u}chi}(\mathcal{T}) \rangle & \text{iff} & \hat{v} \in \nu Y \cdot \mu X \cdot \mathsf{1CPre}(X) \cup (\mathcal{T} \cap \mathsf{1CPre}(Y)) \\ \text{in } \langle G, \mathsf{coB\ddot{u}chi}(\mathcal{T}) \rangle & \text{iff} & \hat{v} \in \mu Y \cdot \nu X \cdot \mathsf{1CPre}(X) \cap (\mathcal{T} \cup \mathsf{1CPre}(Y)) \end{array}$$

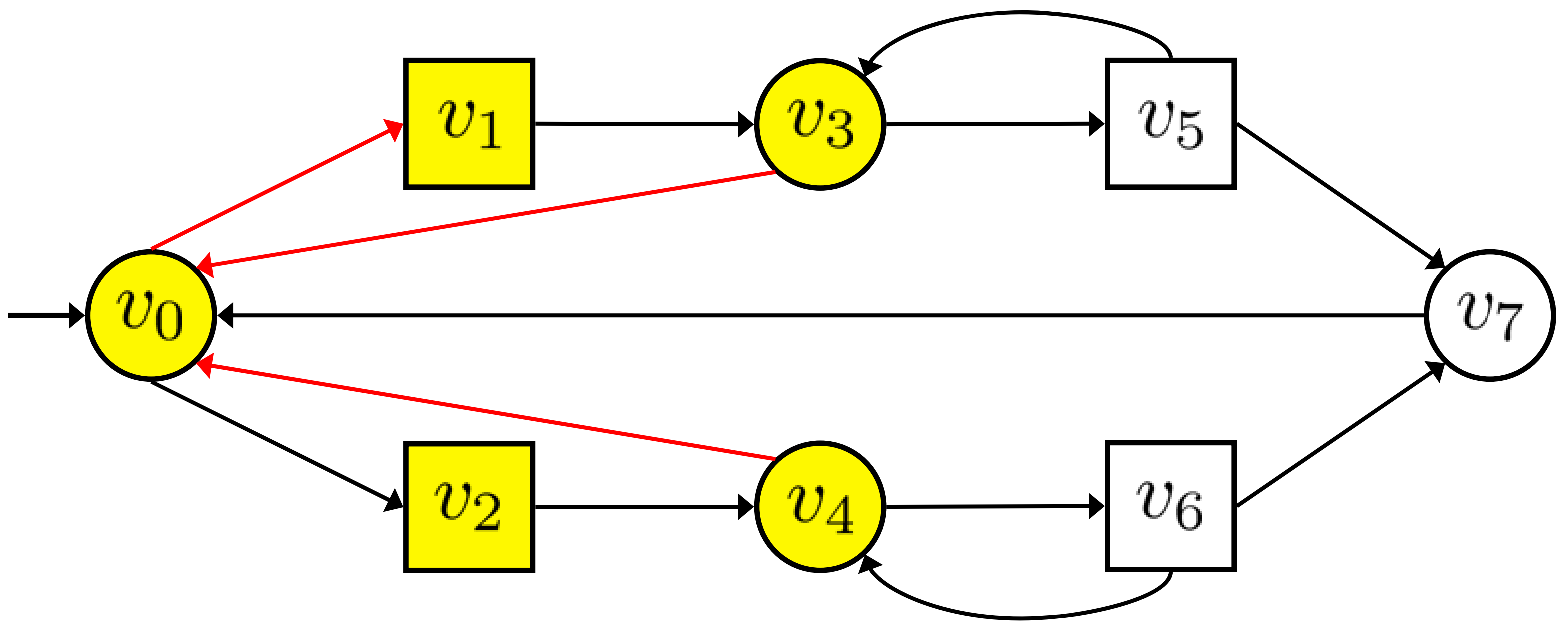
Remarks (I)

Memoryless strategies are always sufficient to win parity games, and therefore also for safety, reachability, Büchi and coBüchi objectives.

Remarks (I)

$\mathcal{T} = V \setminus \{v_7\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



A memoryless winning strategy

Remarks (II)

Parity games are **determined**:
in every state, either Player 1 or Player 2 has a winning strategy.

Remarks (II)

Parity games are **determined**:
in every state, either Player 1 or Player 2 has a winning strategy.

$$\phi_1 \equiv \exists \lambda_1 \cdot \forall \lambda_2 : \mathsf{Outcome}(G, \lambda_1, \lambda_2) \in \mathsf{Parity}(p)$$

$$\phi_2 \equiv \exists \lambda_2 \cdot \forall \lambda_1 : \mathsf{Outcome}(G, \lambda_1, \lambda_2) \notin \mathsf{Parity}(p)$$

Determinacy says: $\phi_1 \vee \phi_2$

More generally, zero-sum games with Borel objectives are determined [Martin75].

Remarks (II)

For instance, since $V^\omega \setminus \mathsf{Safe}(\mathcal{T}) = \mathsf{Reach}(V \setminus \mathcal{T})$,

Player 1 does not win $\langle G, \mathsf{Safe}(\mathcal{T})\rangle$

iff Player 2 wins $\langle G, \mathsf{Reach}(V \setminus \mathcal{T})\rangle$.

$$\boxed{X_* = \nu X \cdot \mathcal{T} \cap \mathsf{1CPre}(X)} \qquad \boxed{X'_* = \mu X' \cdot \mathcal{T}' \cup \mathsf{2CPre}(X')}$$

Claim: if $\mathcal{T}' = V \setminus \mathcal{T}$, then $X'_* = V \setminus X_*$

Proof: exercise

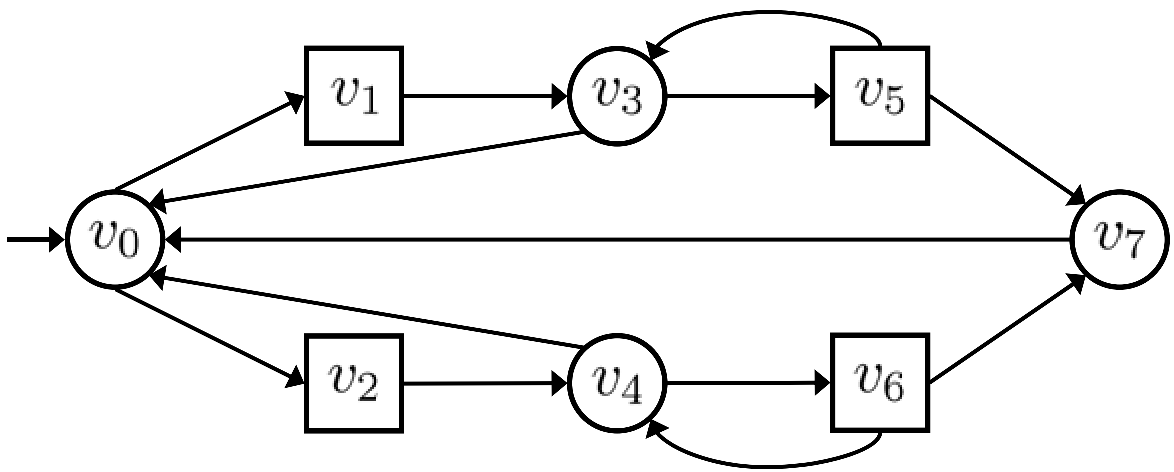
Hint: show that $V \setminus \mathsf{1CPre}(X) = \mathsf{2CPre}(V \setminus X)$

Remarks (II)

$\mathcal{T} = V \setminus \{v_7\}$

Objective for Player 1: Safe($\mathcal{T}$)

for Player 2: Reach($\{v_7\}$)



$X_0 = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6\}$

$X_1 = \{v_0, v_1, v_2, v_3, v_4\}$

$X_2 = \{v_0, v_1, v_2, v_3, v_4\}$

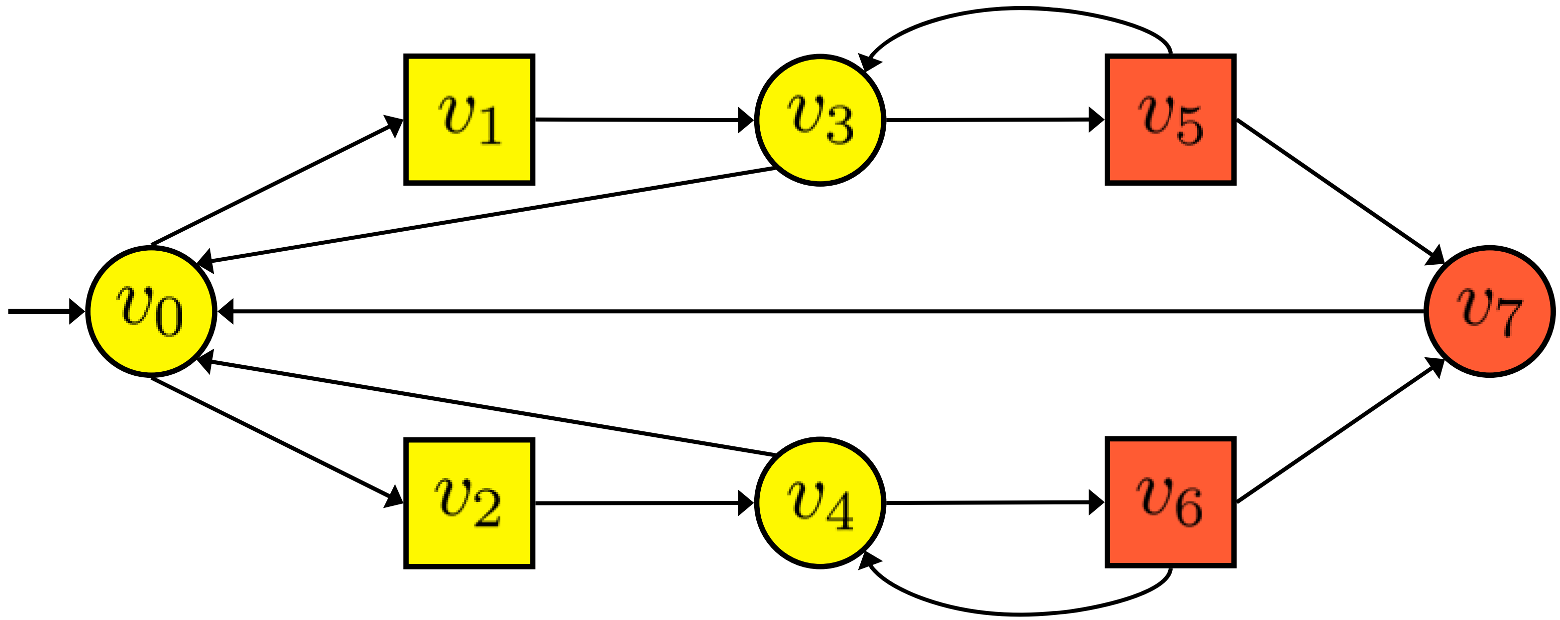
$X'_0 = \{v_7\}$

$X'_1 = \{v_5, v_6, v_7\}$

$X'_2 = \{v_5, v_6, v_7\}$

Remarks (II)

$\mathcal{T} = V \setminus \{v_7\}$



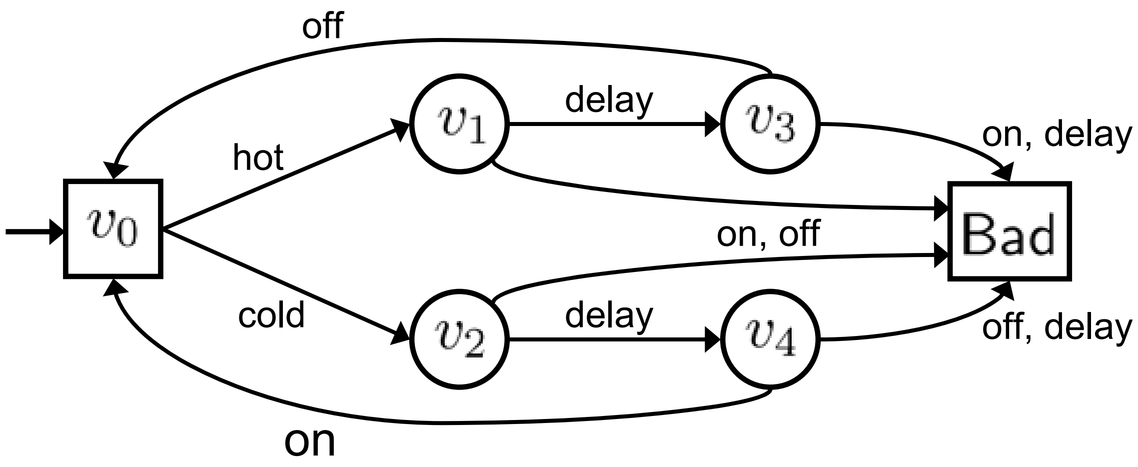
States in which Player 1 wins for $\mathsf{Safe}(\mathcal{T})$.

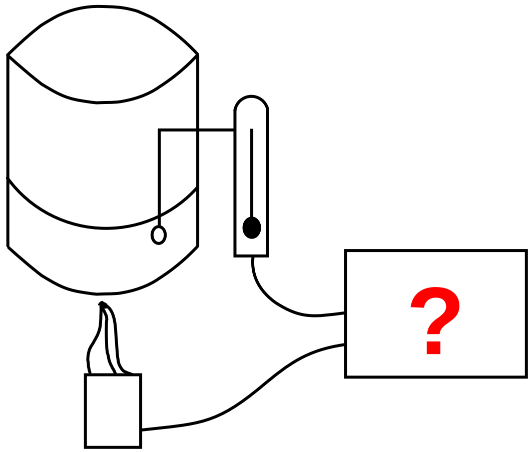
States in which Player 2 wins for $\mathsf{Reach}(V \setminus \mathcal{T})$.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Games of imperfect information

The Synthesis Question



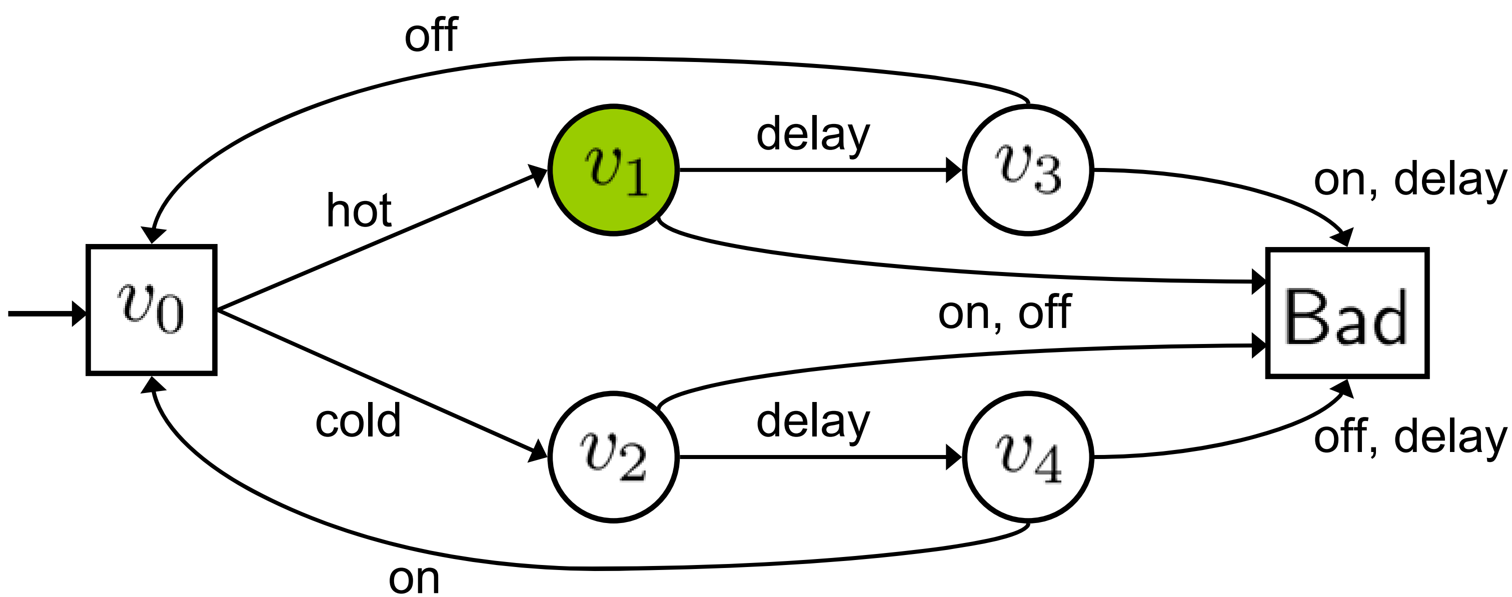


The controller knows the state of the plant ("perfect information").
This, however, is often unrealistic.

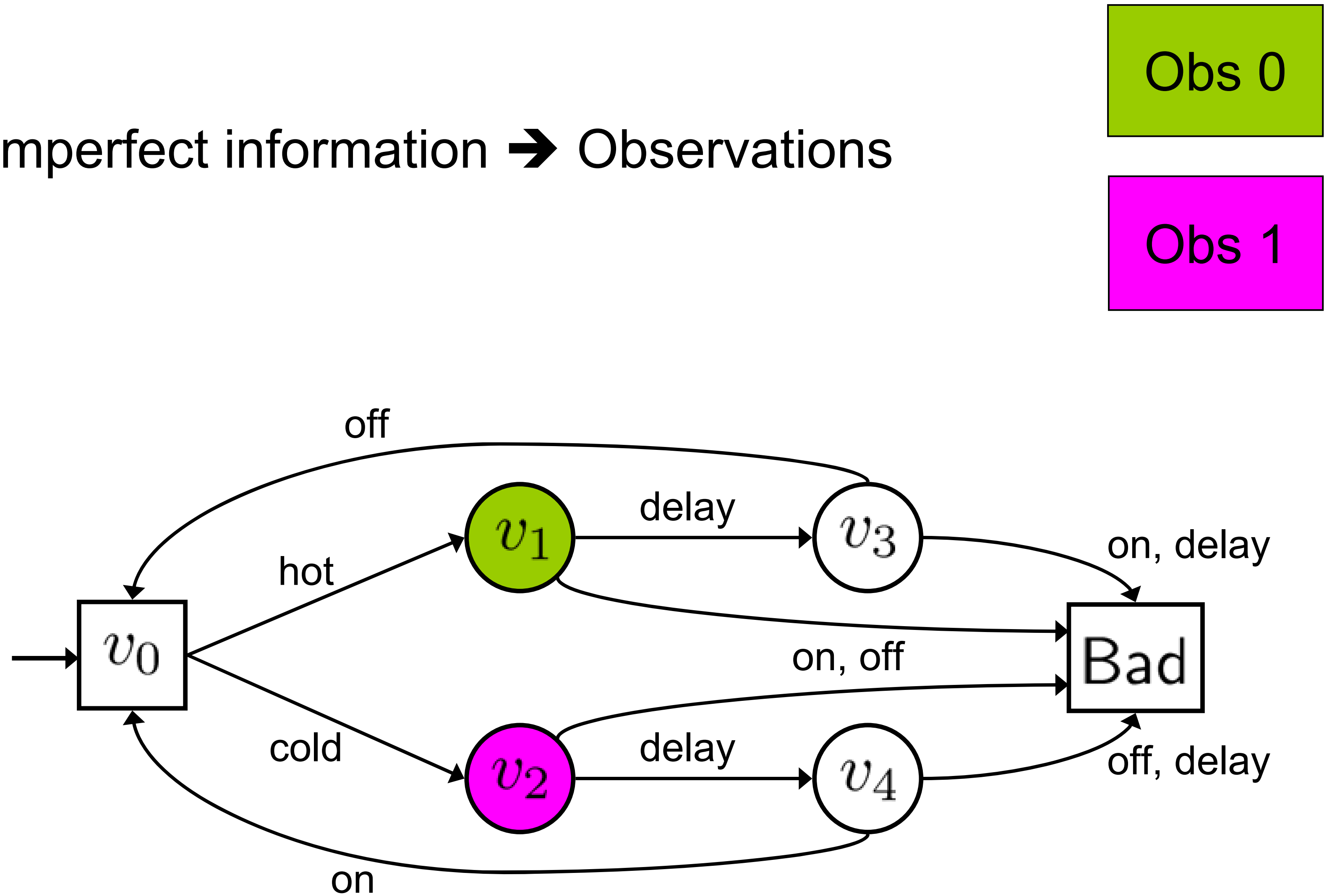
- Sensors provide partial information (imprecision),
- Sensors have internal delays,
- Some variables of the plant are invisible,
- etc.…

Obs 0

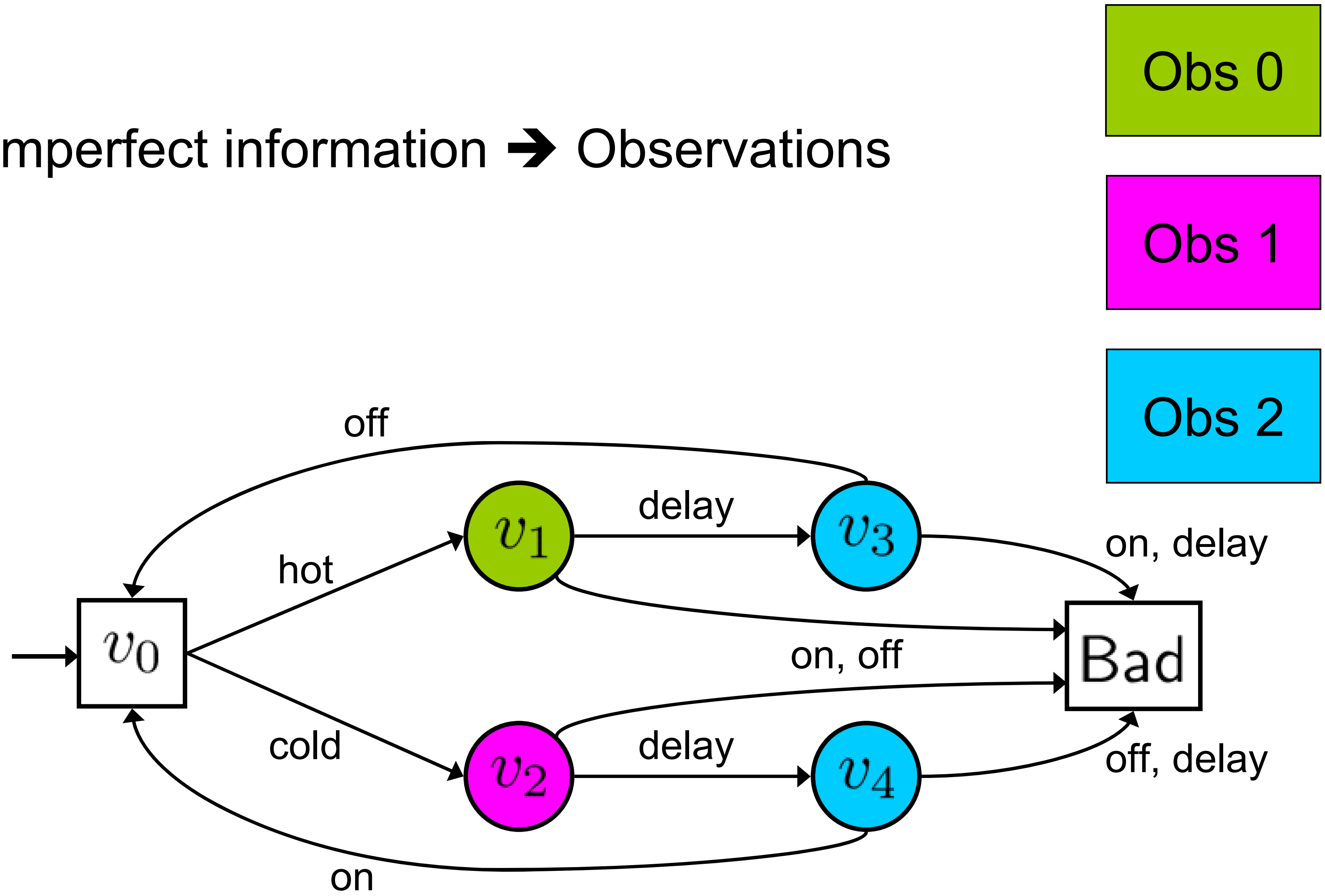
Imperfect information ➔ Observations



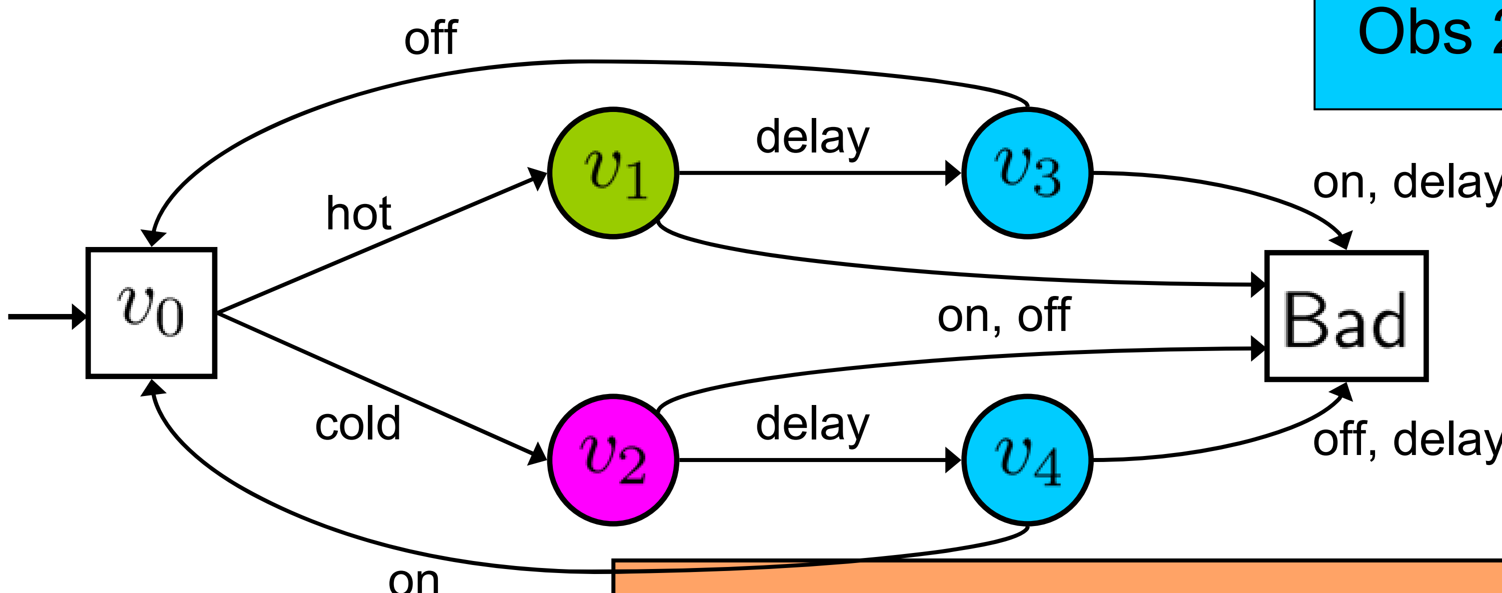
Imperfect information ➔ Observations



Imperfect information ➔ Observations

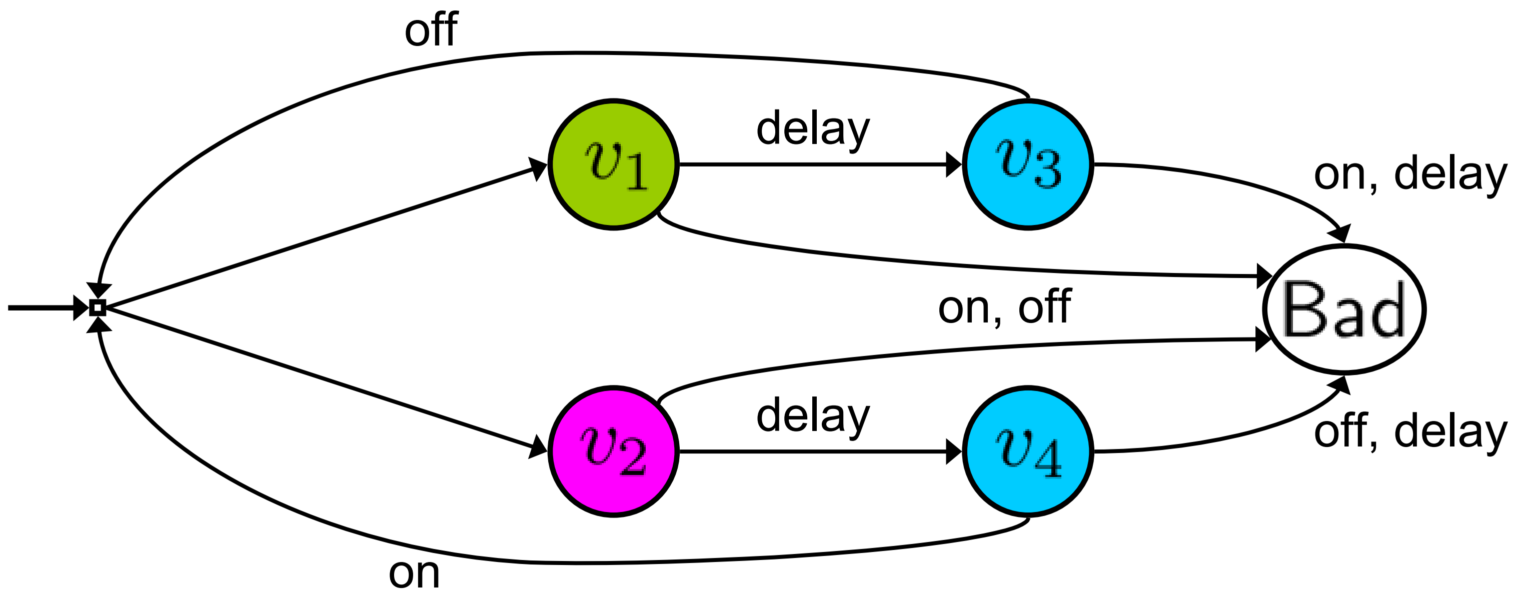


Imperfect information ➔ Observations



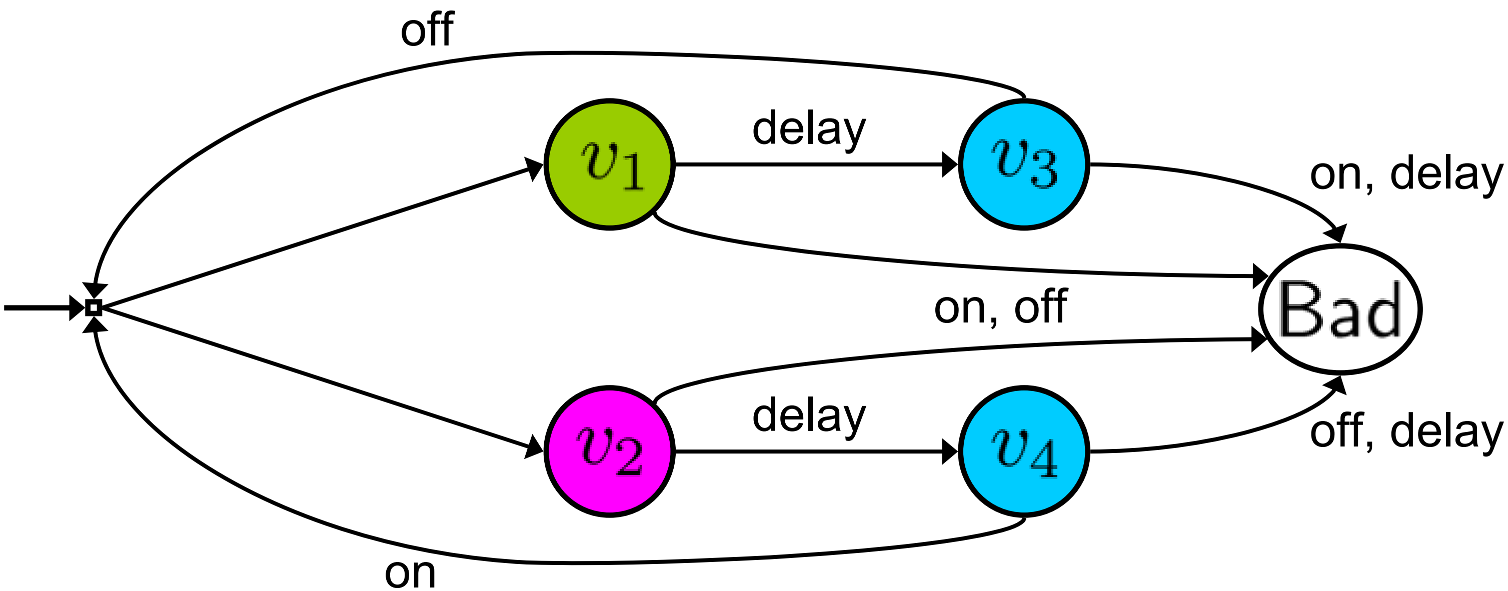
When observing Obs 2,
there is no unique good choice:
memory is necessary

Player 2 states ➔ Nondeterminism



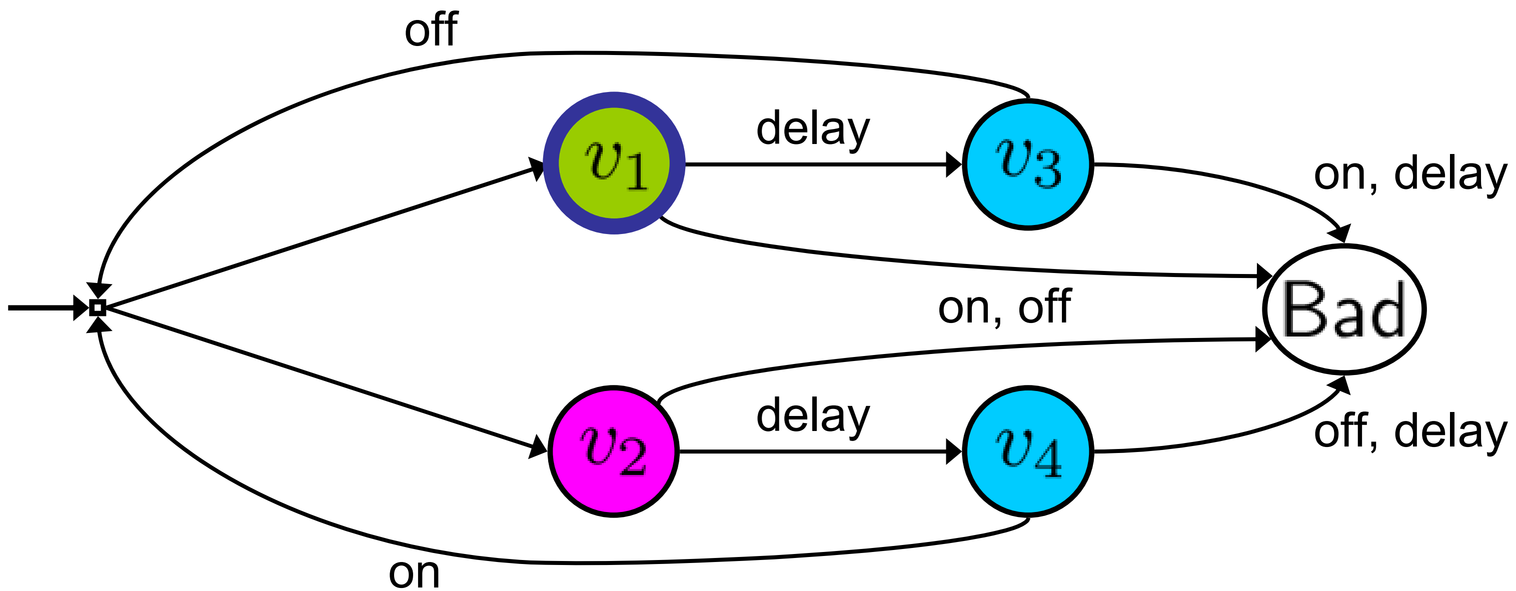
Playing the game: Player 2 moves a **token** along the edges of the graph,
Player 1 does not see the position of the token.

- Player 1 chooses an action (on, off, delay), and then
- Player 2 resolves the nondeterminism and announces the color of the state.



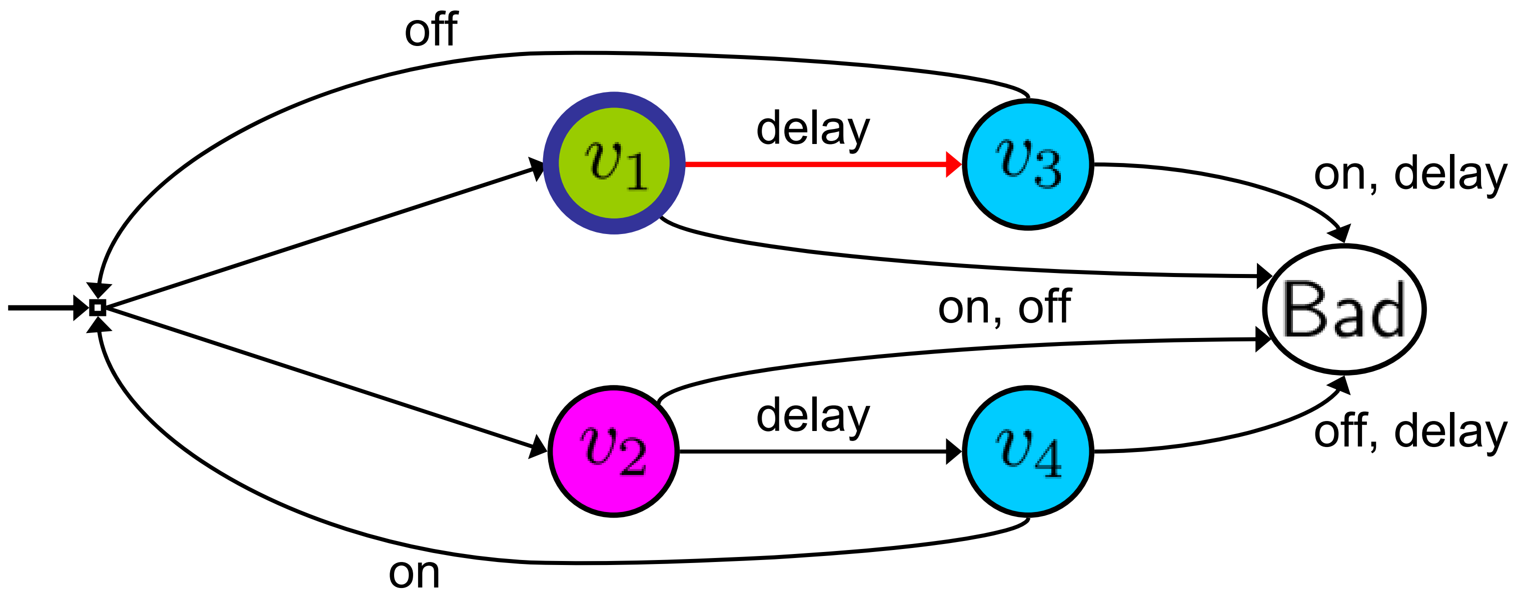
Player 2:

Player 1:



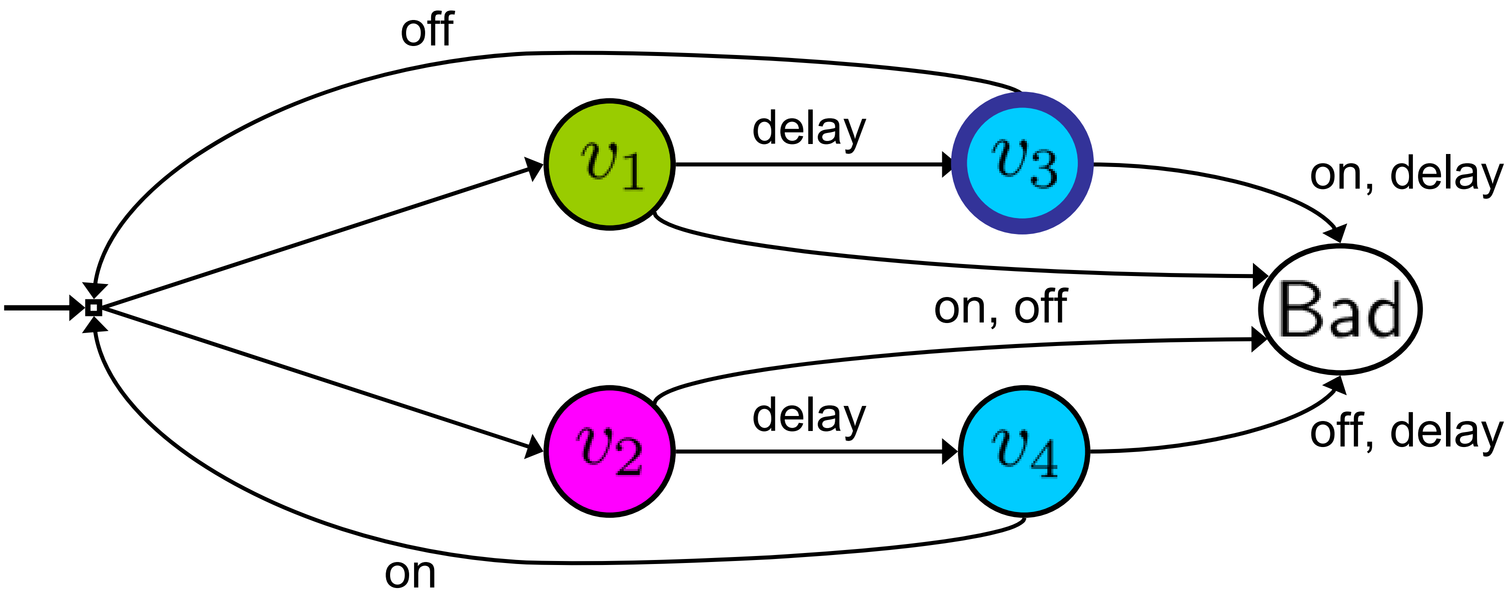
Player 2: v_1 chooses v_1, announces Obs 0

Player 1:



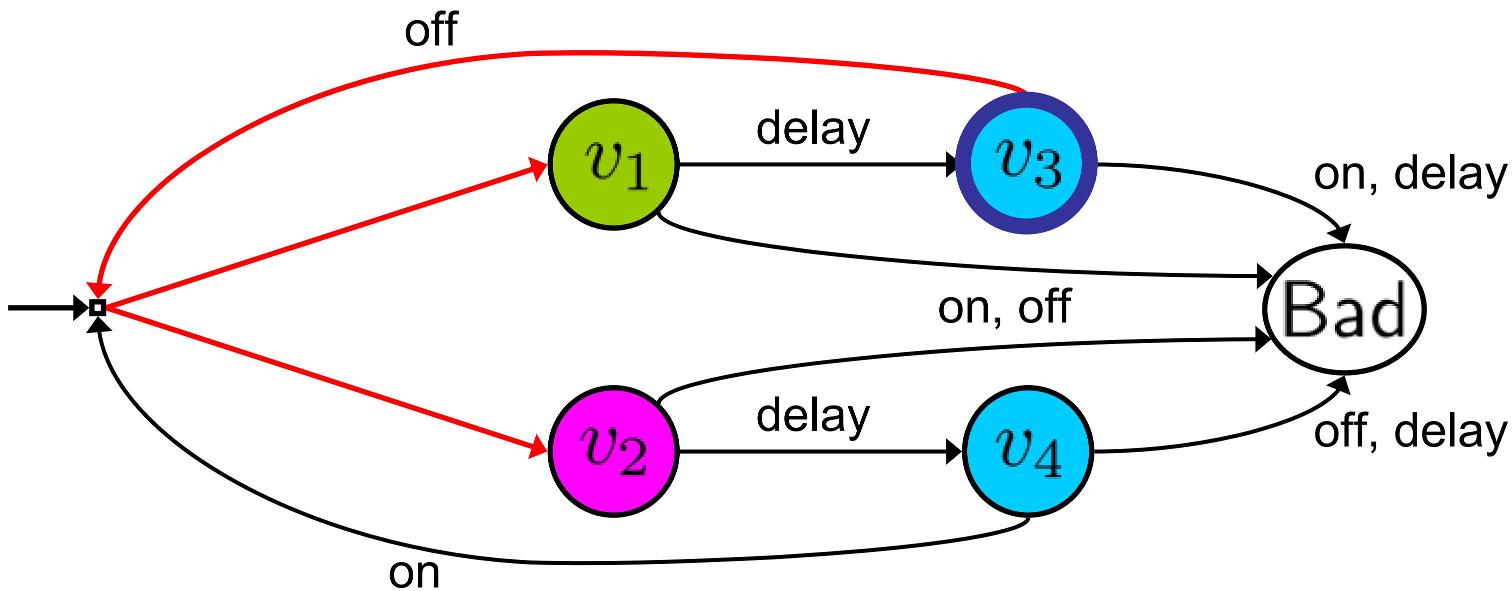
Player 2: v_1 delay

Player 1: delay plays action *delay*

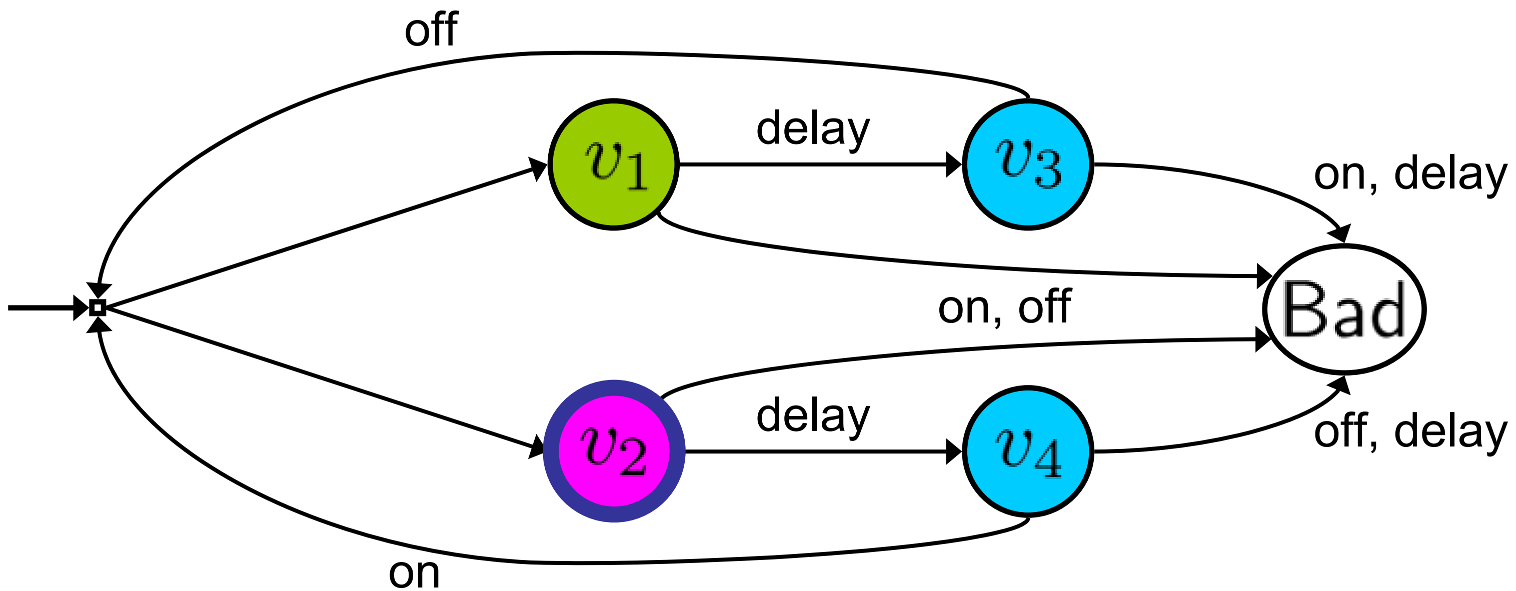


Player 2: v_1 delay v_3 chooses v_3, announces Obs 2

Player 1: delay



Player 2:	v_1	delay	v_3	off
Player 1:		delay		off



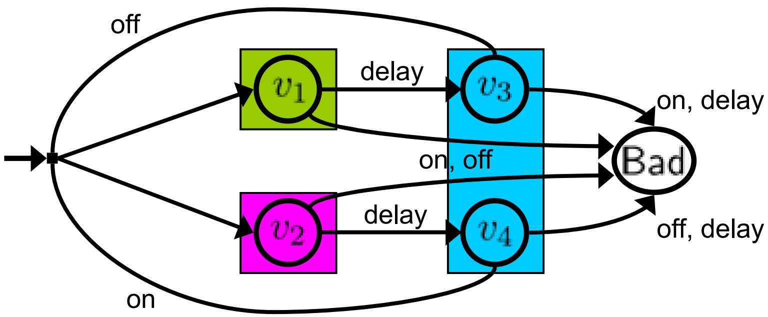
Player 2:	v_1	delay	v_3	off	v_2	…
Player 1:		delay		off		…

Imperfect information

A **game graph + Observation structure**

$G = \langle V, \hat{v}, \mathsf{Succ} \rangle$ $\langle \Sigma, \mathsf{Obs} \rangle$

- Σ is a finite alphabet,
- Obs is a partition of V,
- $\mathsf{Succ} : V \times \Sigma \to 2^V \setminus \varnothing$.



$\Sigma = \{delay, on, off\}$

$\mathsf{Obs} = \{\{v_1\}, \{v_2\}, \{v_3, v_4\}\}$

$\mathsf{Post}_\sigma(s) = \{v' \in \mathsf{Succ}(v, \sigma) \mid v \in s\}$

Indistinguishable states belong to the same observation.

Let $\mathsf{obs}(v) \in \mathsf{Obs}$ be the (unique) observation containing v.

Strategies

Player 1 chooses a letter in Σ,

Player 2 resolves nondeteminisim.

An **observation-based strategy for Player 1** is a function:

$$\boxed{\lambda_1 : \mathsf{Obs}^+ \to \Sigma}$$

A strategy for Player 2 is a function:

$$\boxed{\lambda_2 : V^+ \times \Sigma \to V}$$

such that

$$\boxed{\lambda_2(v_1 \dots v_n, \sigma) \in \mathsf{Succ}(v_n, \sigma) \text{ for all } v_1, \dots, v_n \in V \text{ and } \sigma \in \Sigma}$$

Outcome

$\lambda_1 : \mathsf{Obs}^+ \to \Sigma$

$\lambda_2 : V^+ \times \Sigma \to V$

The **outcome** of $\langle \lambda_1, \lambda_2 \rangle$ is the play
$w = v_0 v_1 \dots$ such that:

$$v_{i+1} = \lambda_2(v_0 \dots v_i, \sigma) \text{ where } \sigma = \lambda_1(\mathsf{obs}(v_0) \dots \mathsf{obs}(v_i))$$

for all $i \geq 0$.

This play is denoted $\mathsf{Outcome}(G, \lambda_1, \lambda_2)$

Winning strategies

A **winning condition** for Player 1 is a set $\boxed{U_1 \subseteq \mathsf{Obs}^\omega}$ of sequences of observations. The set U_1 defines the set of winning plays:

$$W_1 = \{v_0 v_1 \cdots \mid \mathsf{obs}(v_0)\mathsf{obs}(v_1) \cdots \in U_1\}$$

Player 1 is winning if

$$\boxed{\exists \lambda_1 \cdot \forall \lambda_2 : \mathsf{Outcome}(G, \lambda_1, \lambda_2) \in W_1}$$

Solving games of imperfect information

Imperfect information

Games of imperfect information can be solved by a reduction to games of perfect information.

G,Obs → G' → Winning region

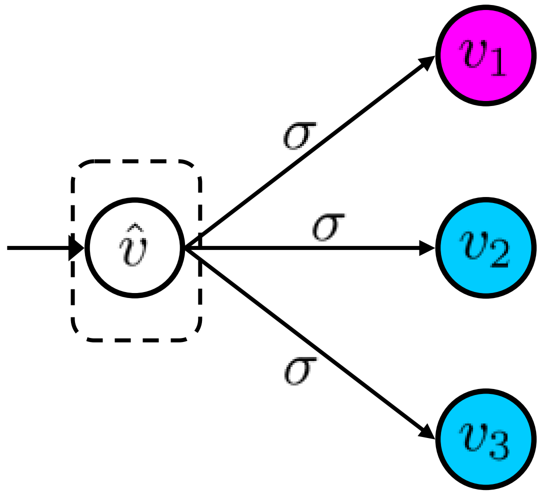
G,Obs		G'		Winning region
Imperfect information	subset construction	Perfect information	classical techniques	

Subset construction

After a finite prefix of a play, Player 1 has a partial knowledge of the current state of the game: **a set of states**, called a **cell**.

Subset construction

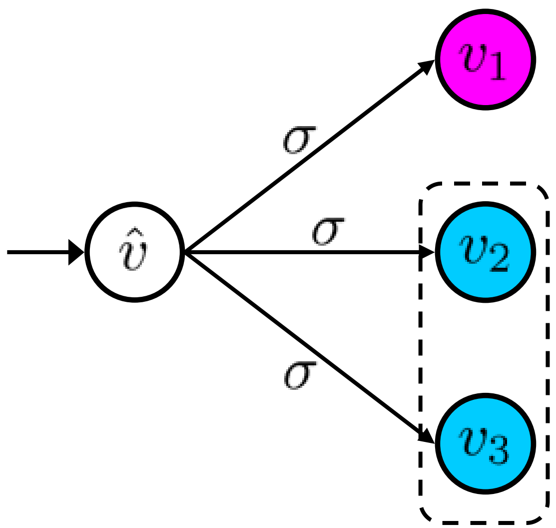
After a finite prefix of a play, Player 1 has a partial knowledge of the current state of the game: **a set of states**, called a **cell**.



Initial knowledge: cell $\{\widehat{v}\}$

Subset construction

After a finite prefix of a play, Player 1 has a partial knowledge of the current state of the game: **a set of states**, called a **cell**.



Initial knowledge: cell $\{\hat{v}\}$

Player 1 plays σ,

Player 2 chooses v_2.

Current knowledge: cell $\{v_2, v_3\}$

$\mathsf{Post}_\sigma(\{\hat{v}\}) \cap o_2$

Subset construction

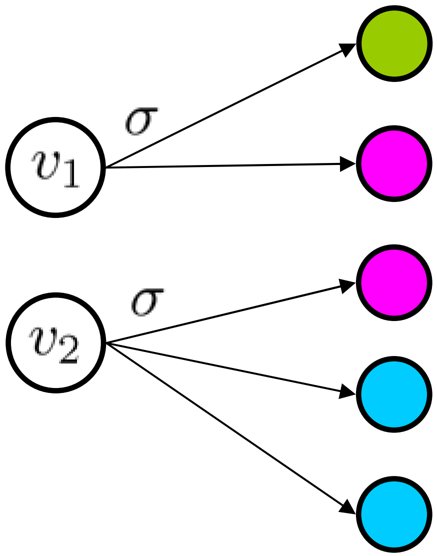
	Imperfect information	Perfect information
	$G = \langle V, \widehat{v}, \mathsf{Succ} \rangle$	$G' = \langle V_1', V_2', \widehat{v}', \mathsf{Succ}' \rangle$
	$\langle \Sigma, \mathsf{Obs} \rangle$	
State space	V	$V_1' = 2^V$ $V_2' = 2^V \times \Sigma$
Initial state	$\widehat{v}$	$\widehat{v}' = \{\widehat{v}\}$

Subset construction

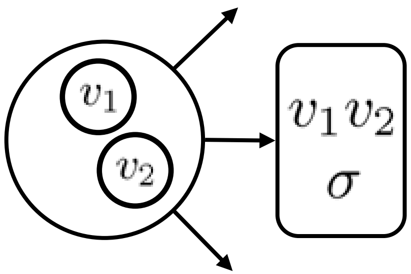
$$G = \langle V, \hat{v}, \mathsf{Succ} \rangle$$

$$\langle \Sigma, \mathsf{Obs} \rangle$$

Transitions



$$G' = \langle V_1', V_2', \hat{v}', \mathsf{Succ}' \rangle$$



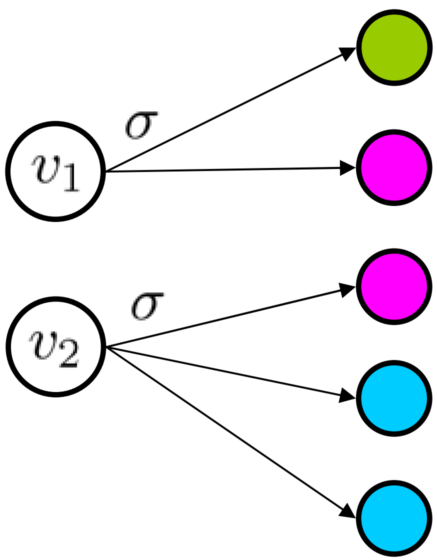
$$\mathsf{Succ}'(s) = \{(s, \sigma) \mid \sigma \in \Sigma\}$$

Subset construction

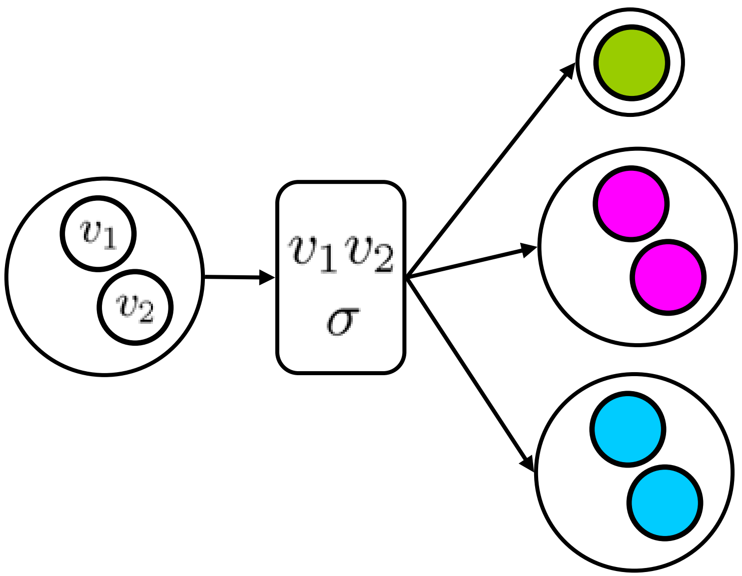
$G = \langle V, \widehat{v}, \mathsf{Succ} \rangle$

$\langle \Sigma, \mathsf{Obs} \rangle$

Transitions



$G' = \langle V_1', V_2', \widehat{v}', \mathsf{Succ}' \rangle$



$$\mathsf{Succ}'(s) = \{(s, \sigma) \mid \sigma \in \Sigma\}$$
$$\mathsf{Succ}'(s, \sigma) = \{\mathsf{Post}_\sigma(s) \cap o \mid o \in \mathsf{Obs}\}$$

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Subset construction

$$G = \langle V, \widehat{v}, \mathsf{Succ} \rangle \qquad G' = \langle V_1', V_2', \widehat{v}', \mathsf{Succ}' \rangle$$

$$\langle \Sigma, \mathsf{Obs} \rangle$$

Parity condition

$$p : \mathsf{Obs} \rightarrow \mathbb{N}$$

$$p' : V_1' \cup V_2' \rightarrow \mathbb{N}$$

$$p'(s) = p'(s, \sigma) = p(o)$$

where $s \subseteq o$.

Subset construction

$G = \langle V, \hat{v}, \mathsf{Succ} \rangle$

$\langle \Sigma, \mathsf{Obs} \rangle$

$G' = \langle V'_1, V'_2, \hat{v}', \mathsf{Succ}' \rangle$

Parity condition

$p : \mathsf{Obs} \to \mathbb{N}$

$p' : V'_1 \cup V'_2 \to \mathbb{N}$

$p'(s) = p'(s, \sigma) = p(o)$
where $s \subseteq o$.

Theorem

Player 1 is winning in G,p if and only if Player 1 is winning in G',p'.

Imperfect information

G,Obs ➔ G' ➔ Winning region

Imperfect information

Perfect information

subset construction

Exponential blow-up

classical techniques

Imperfect information

G,Obs ➔ G' ➔ Winning region

implicit

Imperfect information

Perfect information

Direct symbolic algorithm

Symbolic algorithm

Controllable predecessor: $\mathsf{1CPre} : 2^{V_1'} \to 2^{V_1'}$

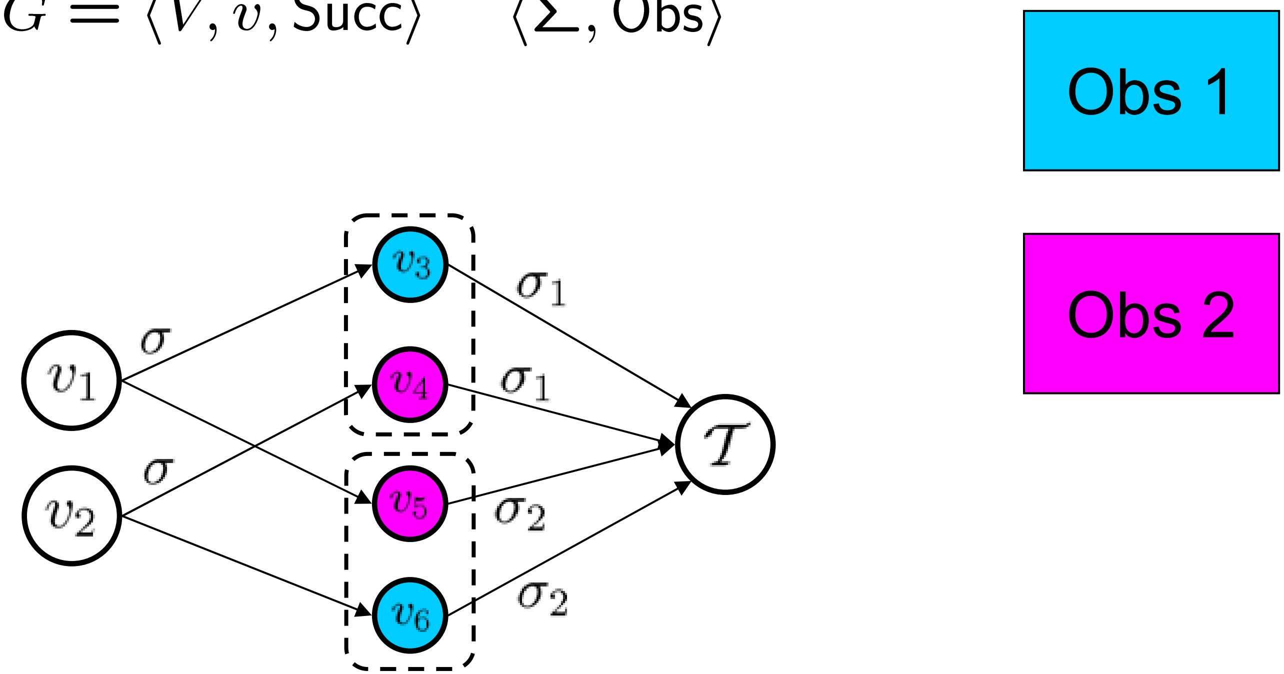
set of cells

$$\mathsf{1CPre}(q) = \{s \mid \exists (s,\sigma) \in \mathsf{Succ}'(s) \cdot \forall s' \in \mathsf{Succ}'(s,\sigma) : s' \in q\}$$

$$= \underbrace{\{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} : \mathsf{Post}_\sigma(s) \cap o \in q\}}_{\text{set of cells}}$$

Symbolic algorithm

$$G = \langle V, \hat{v}, \mathsf{Succ} \rangle \qquad \langle \Sigma, \mathsf{Obs} \rangle$$



$$\mathsf{1CPre}\big(\{\{v_3, v_4\}, \{v_5, v_6\}\}\big) = \{\{v_1\}, \{v_2\}\}$$

$$\neq \{\{v_1, v_2\}\}$$

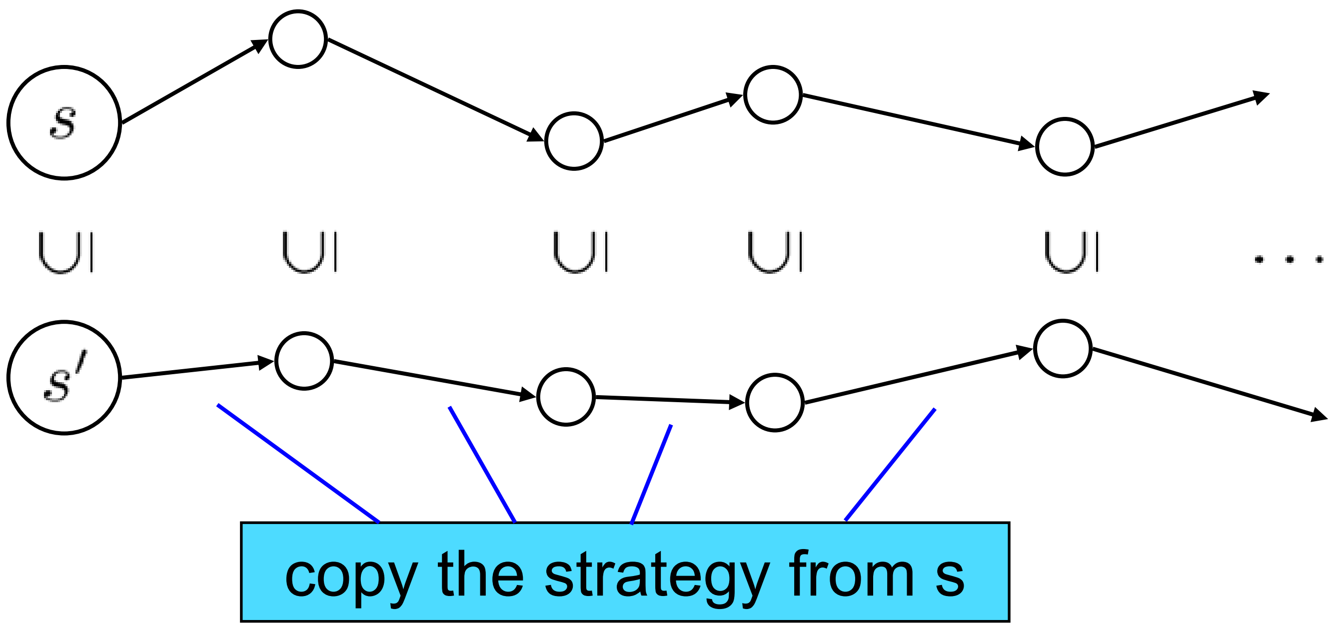
The union of two controllable cells is not necessarily controllable,

but…

Symbolic algorithm

$$1\mathsf{CPre}(q) = \{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} : \mathsf{Post}_\sigma(s) \cap o \in q\}$$

If a cell s is controllable (i.e. winning for Player 1), then all sub-cells s' $\subseteq$ s are controllable.



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Symbolic algorithm

$$\mathsf{1CPre}(q) = \{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} : \mathsf{Post}_\sigma(s) \cap o \in q\}$$

The sets of cells computed by the fixpoint iterations are **downward-closed**.

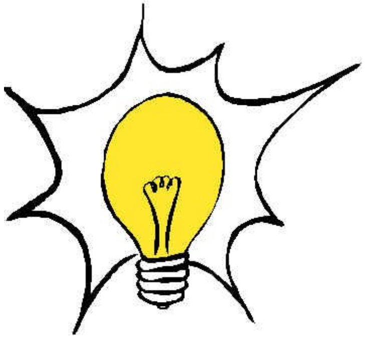
A set q of cells is downward-closed
if $s \in q$ and $s' \subseteq s$ implies $s' \in q$.

Symbolic algorithm

$$\mathsf{1CPre}(q) = \{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} : \mathsf{Post}_\sigma(s) \cap o \in q\}$$

The sets of cells computed by the fixpoint iterations are **downward-closed**.

A set q of cells is downward-closed
if $s \in q$ and $s' \subseteq s$ implies $s' \in q$.

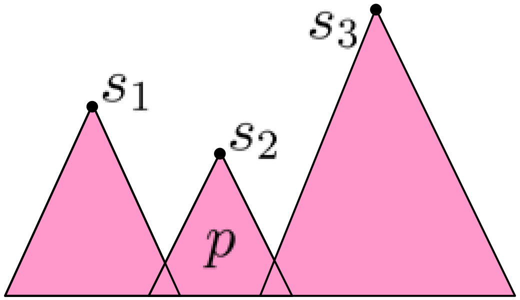


It is sufficient to keep only the **maximal cells**.

Antichains

Maximal cells in p: $\boxed{\lceil p \rceil = \{s \in p \mid \forall s' \in p : s \not\subset s'\}}$

$\lceil p \rceil$ is an **antichain**, *i.e.* a set of $\subseteq$-incomparable cells.



$\lceil p \rceil = \{s_1, s_2, s_3\}$

Antichains

Maximal cells in p: $\boxed{\lceil p \rceil = \{s \in p \mid \forall s' \in p : s \not\subset s'\}}$

$\lceil p \rceil$is an **antichain**, *i.e.* a set of $\subseteq$-incomparable cells.

For downward-closed set p, we have:

$$\begin{aligned} \mathsf{1CPre}(p) &= \{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} : \mathsf{Post}_\sigma(s) \cap o \in p\} \\ &= \{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} \cdot \exists s' \in \lceil p \rceil : \mathsf{Post}_\sigma(s) \cap o \subseteq s'\} \end{aligned}$$

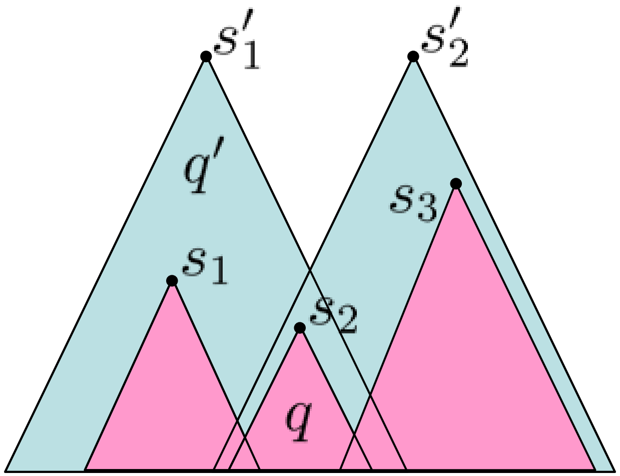
Hence, over antichains we define:

$$\boxed{\mathsf{1CPre}^{\mathcal{A}}(q) = \lceil \{s \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathsf{Obs} \cdot \exists s' \in q : \mathsf{Post}_\sigma(s) \cap o \subseteq s'\} \rceil}$$

Antichains

$\mathsf{1CPre}(\cdot)$ is monotone with respect to the following order:

$q \sqsubseteq q'$ iff $\forall s \in q \cdot \exists s' \in q' : s \subseteq s'$



$\boxed{\langle \mathcal{A}, \sqsubseteq \rangle \text{ is a complete partial order.}}$

Least upper bound and greatest lower bound are defined by:

$$q \sqcup q' = \lceil \{ s \mid s \in q \vee s \in q' \} \rceil$$

$$q \sqcap q' = \lceil \{ s \cap s' \mid s \in q \wedge s' \in q' \} \rceil$$

Symbolic algorithms

Let $G = \langle V, \hat{v}, \mathsf{Succ}, \Sigma, \mathsf{Obs} \rangle$ be a 2-player game graph of imperfect information, and $\mathcal{T} \subseteq \mathsf{Obs}$ a set of observations.

Games of imperfect information can be solved by the same fixpoint formulas as for perfect information, namely:

Theorem

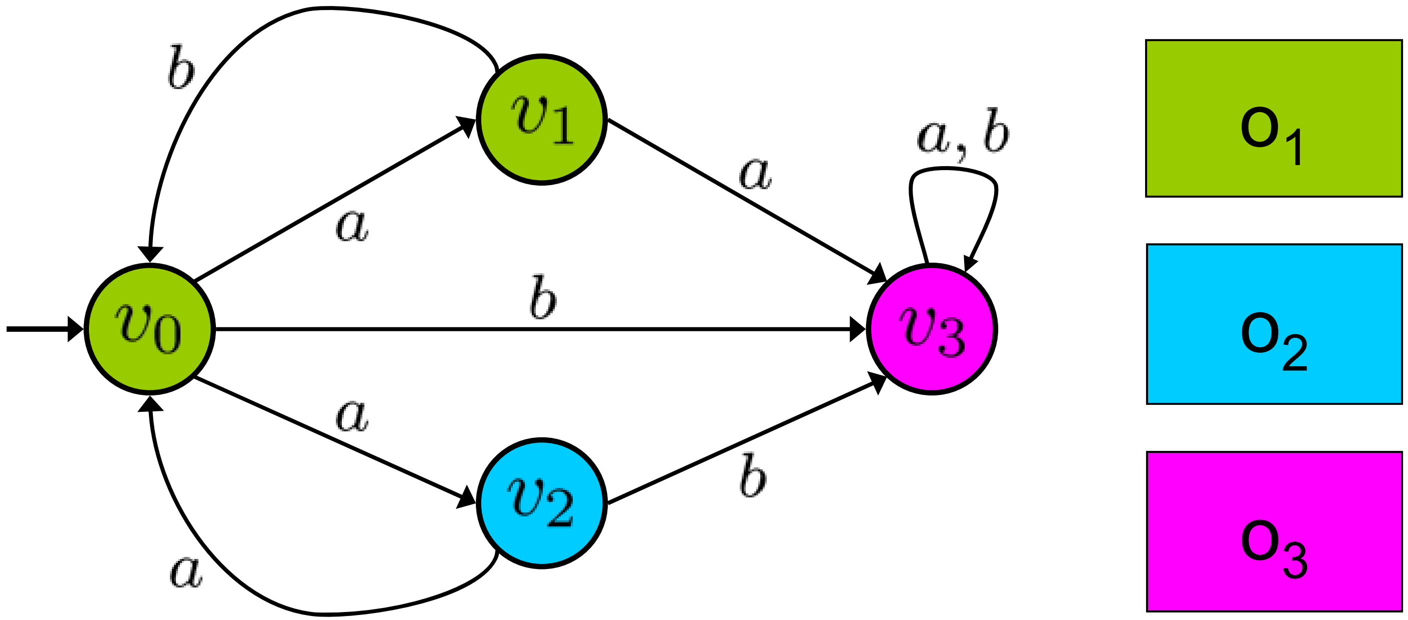
Player 1 has a winning strategy

$$
\begin{array}{lll}
\text{in } \langle G, \mathsf{Reach}(\mathcal{T}) \rangle & \text{iff} & \{\hat{v}\} \sqsubseteq \mu X \cdot \mathcal{T} \sqcup \mathsf{1CPre}(X) \\
\text{in } \langle G, \mathsf{Safe}(\mathcal{T}) \rangle & \text{iff} & \{\hat{v}\} \sqsubseteq \nu X \cdot \mathcal{T} \sqcap \mathsf{1CPre}(X) \\
\text{in } \langle G, \mathsf{Büchi}(\mathcal{T}) \rangle & \text{iff} & \{\hat{v}\} \sqsubseteq \nu Y \cdot \mu X \cdot \mathsf{1CPre}(X) \sqcup (\mathcal{T} \sqcap \mathsf{1CPre}(Y)) \\
\text{in } \langle G, \mathsf{coBüchi}(\mathcal{T}) \rangle & \text{iff} & \{\hat{v}\} \sqsubseteq \mu Y \cdot \nu X \cdot \mathsf{1CPre}(X) \sqcap (\mathcal{T} \sqcup \mathsf{1CPre}(Y))
\end{array}
$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

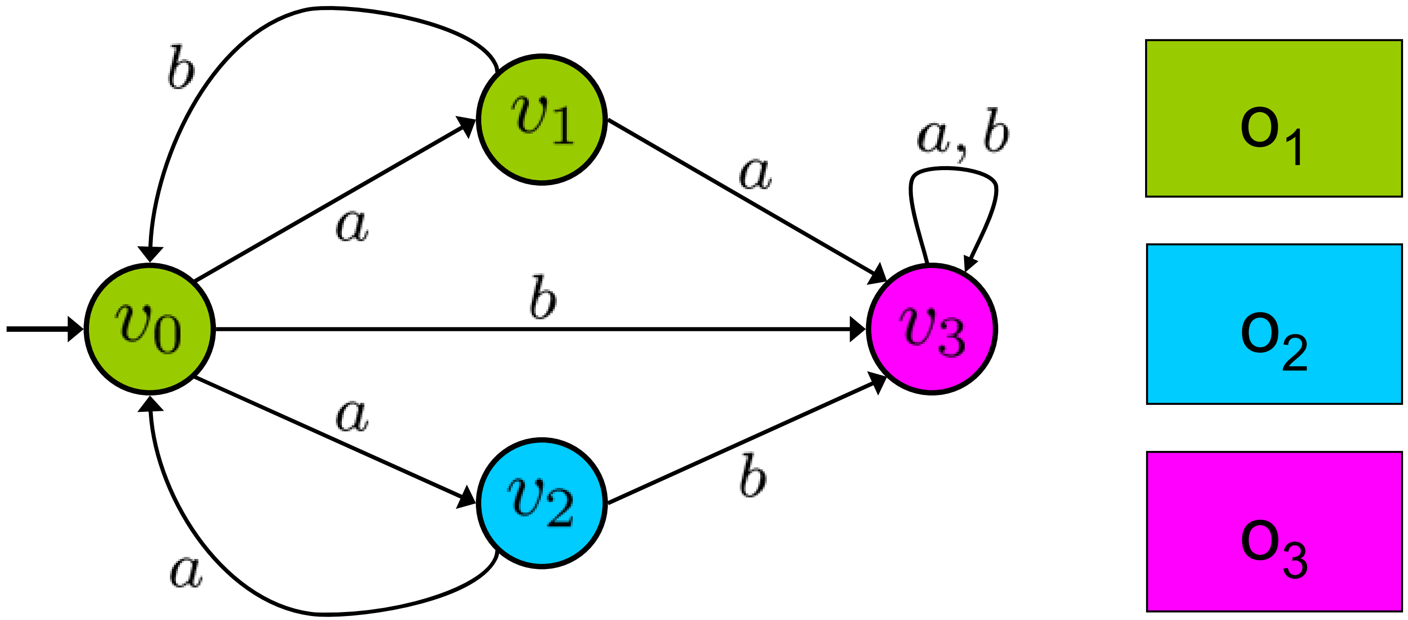
Objective: $\mathsf{Safe}(\mathcal{T})$



Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



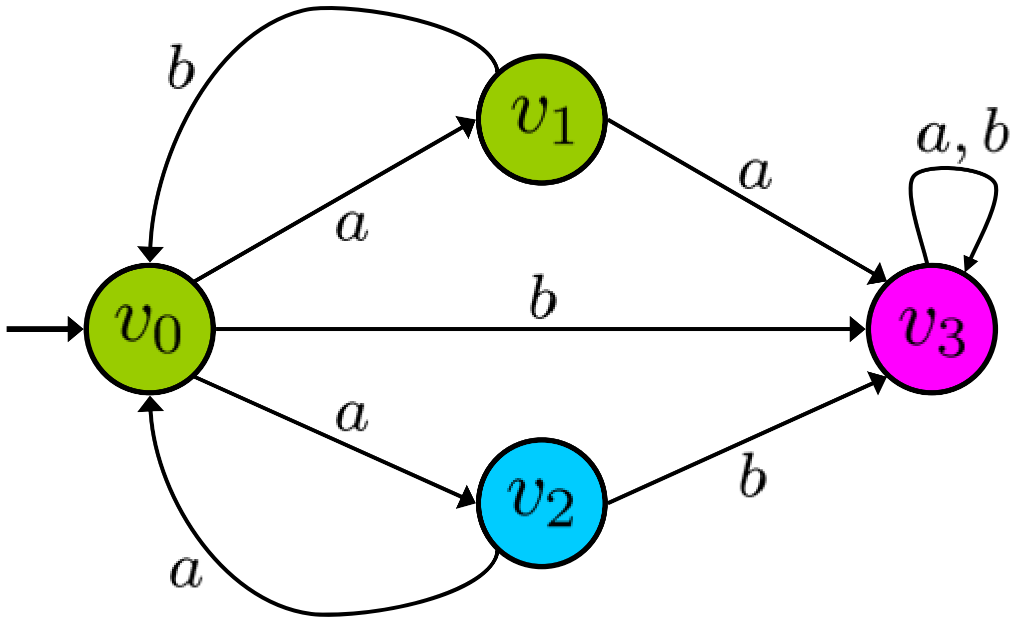
Has Player 1 an observation-based strategy to avoid v_3 ?

We compute the fixpoint $\nu X \cdot \mathcal{T} \sqcap \mathsf{1CPre}(X)$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$

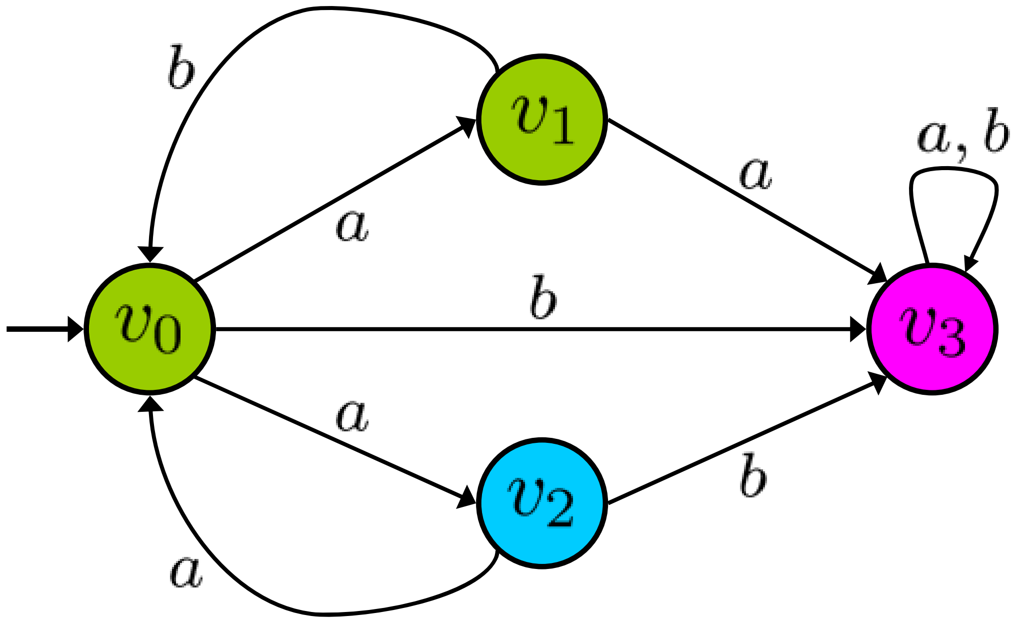


$$X_0 = \mathcal{T} = \big\{\{v_0, v_1\}, \{v_2\}\big\}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



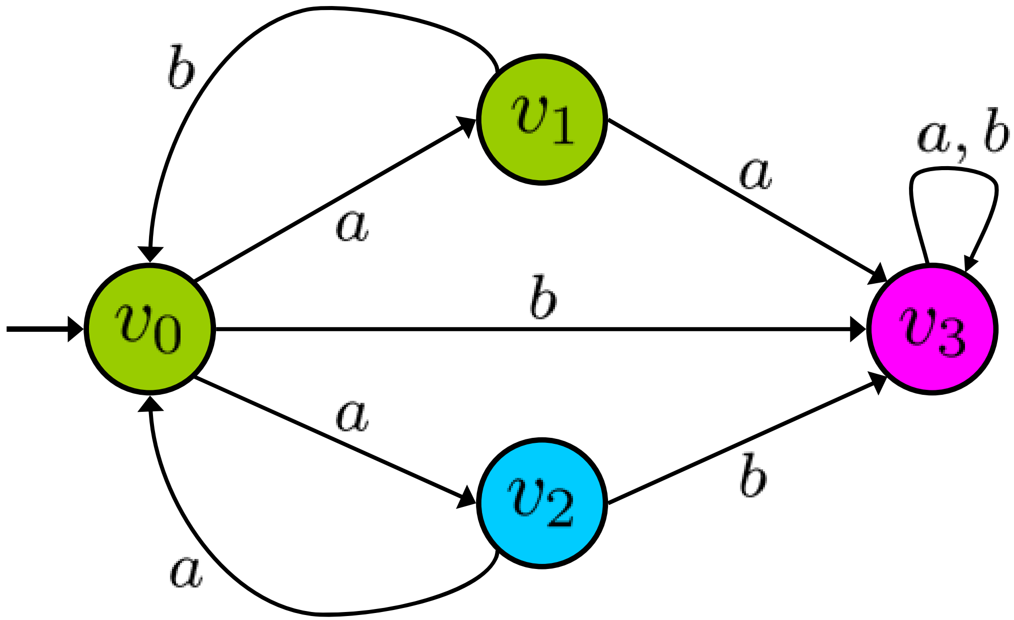
$X_0 = \big\{\{v_0, v_1\}, \{v_2\}\big\}$

$X_1 = \mathsf{CPre}(X_0) \sqcap \mathcal{T} = \big\{\{v_1\}_b, \{v_0, v_2\}_a\big\} \sqcap \mathcal{T}$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



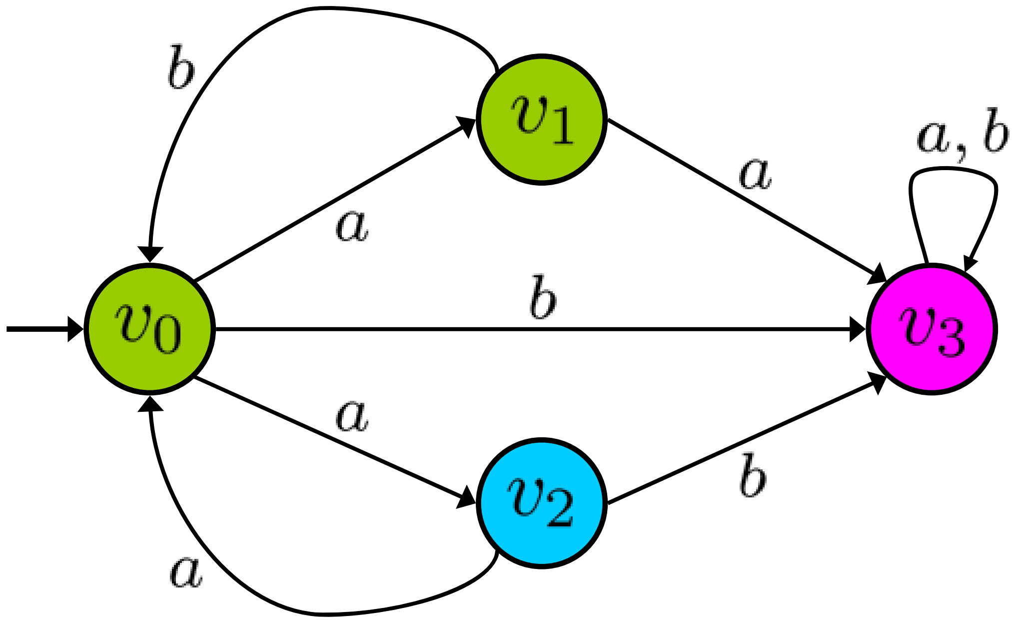
$$X_0 = \Big\{\{v_0, v_1\}, \{v_2\}\Big\}$$

$$X_1 = \mathsf{CPre}(X_0) \sqcap \mathcal{T} = \Big\{\{v_1\}_b, \{v_0, v_2\}_a\Big\} \sqcap \mathcal{T}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



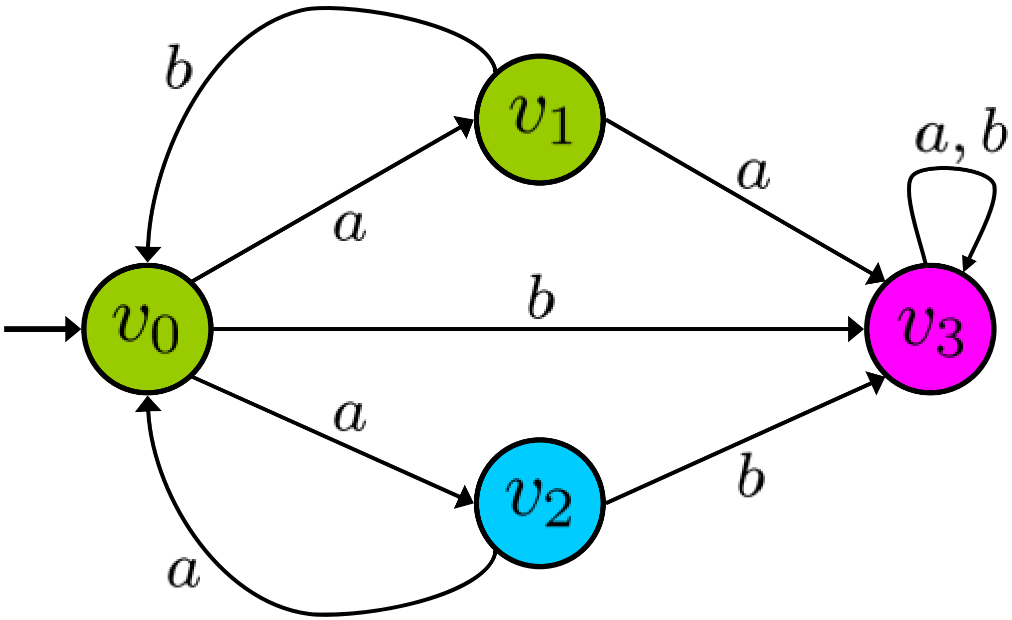
$$X_0 = \big\{\{v_0, v_1\}, \{v_2\}\big\}$$

$$X_1 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$$X_0 = \Big\{\{v_0, v_1\}, \{v_2\}\Big\}$$

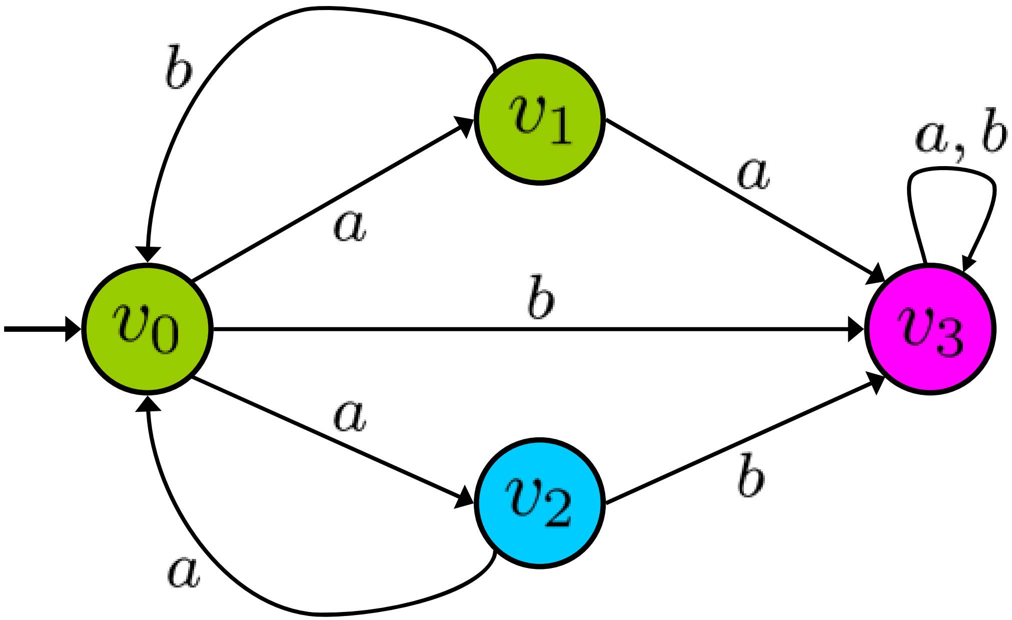
$$X_1 = \Big\{\{v_0\}, \{v_1\}, \{v_2\}\Big\}$$

$$X_2 = \mathsf{CPre}(X_1) \sqcap \mathcal{T} = \Big\{\{v_0\}_a, \{v_1\}_b, \{v_2\}_a\Big\} \sqcap \mathcal{T}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$$X_0 = \big\{\{v_0, v_1\}, \{v_2\}\big\}$$

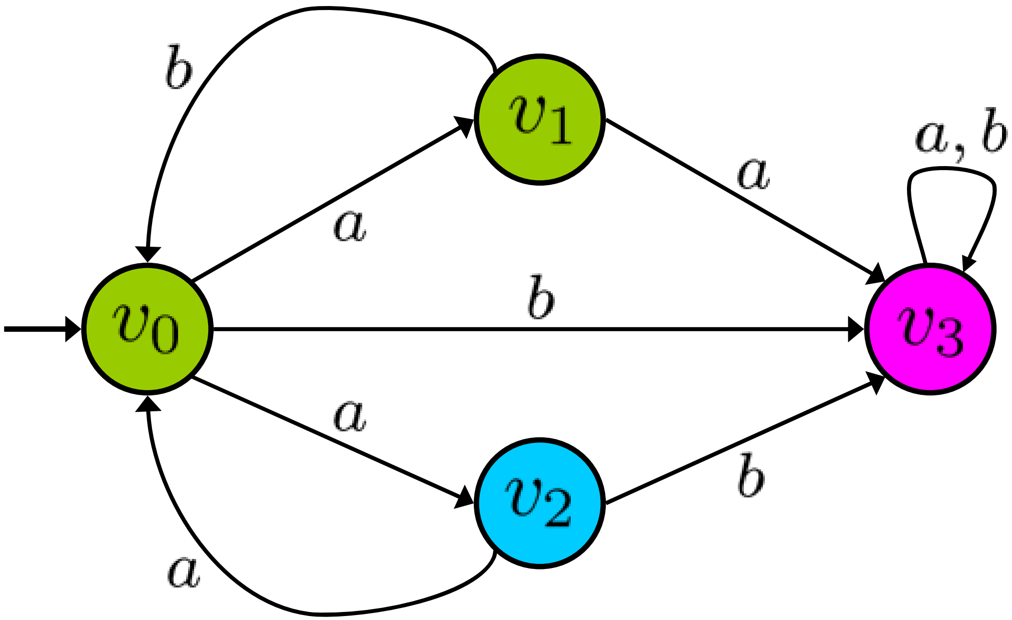
$$X_1 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

$$X_2 = \mathsf{CPre}(X_1) \sqcap \mathcal{T} = \big\{\{v_0\}_a, \{v_1\}_b, \{v_2\}_a\big\} \sqcap \mathcal{T}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$$X_0 = \big\{\{v_0, v_1\}, \{v_2\}\big\}$$

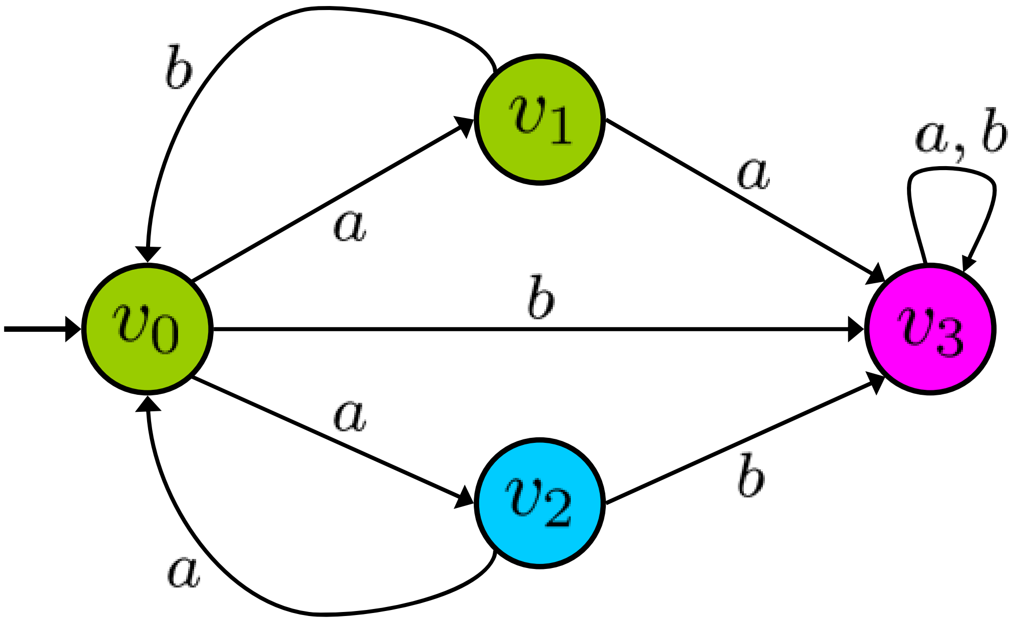
$$X_1 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

$$X_2 = \mathsf{CPre}(X_1) \sqcap \mathcal{T} = \big\{\{v_0\}_a, \{v_1\}_b, \{v_2\}_a\big\} \sqcap \mathcal{T}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$$X_0 = \big\{\{v_0, v_1\}, \{v_2\}\big\}$$

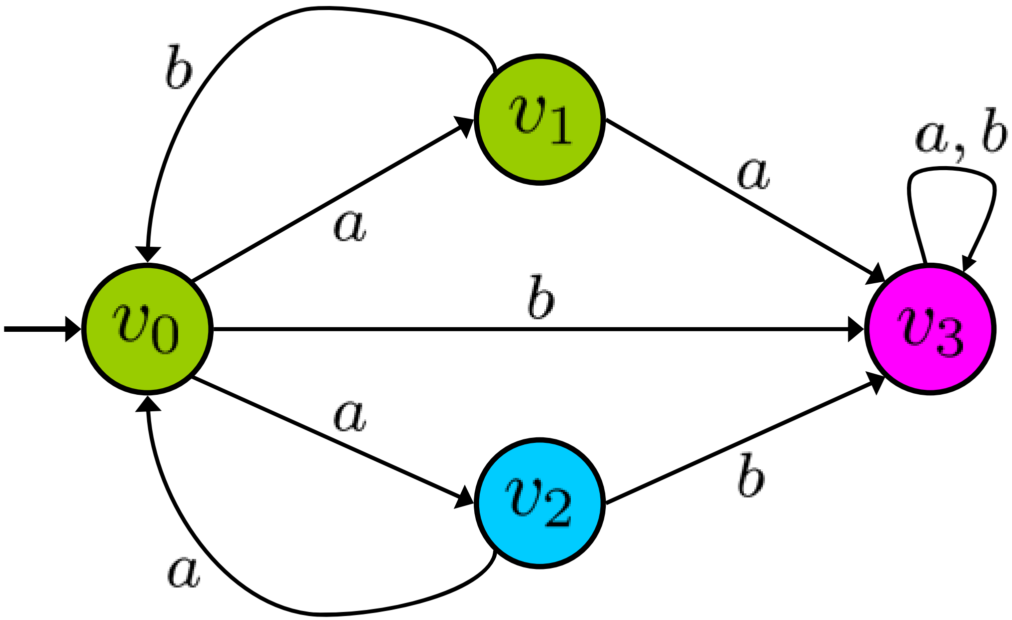
$$X_1 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

$$X_2 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$$X_0 = \big\{\{v_0, v_1\}, \{v_2\}\big\}$$

$$X_1 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

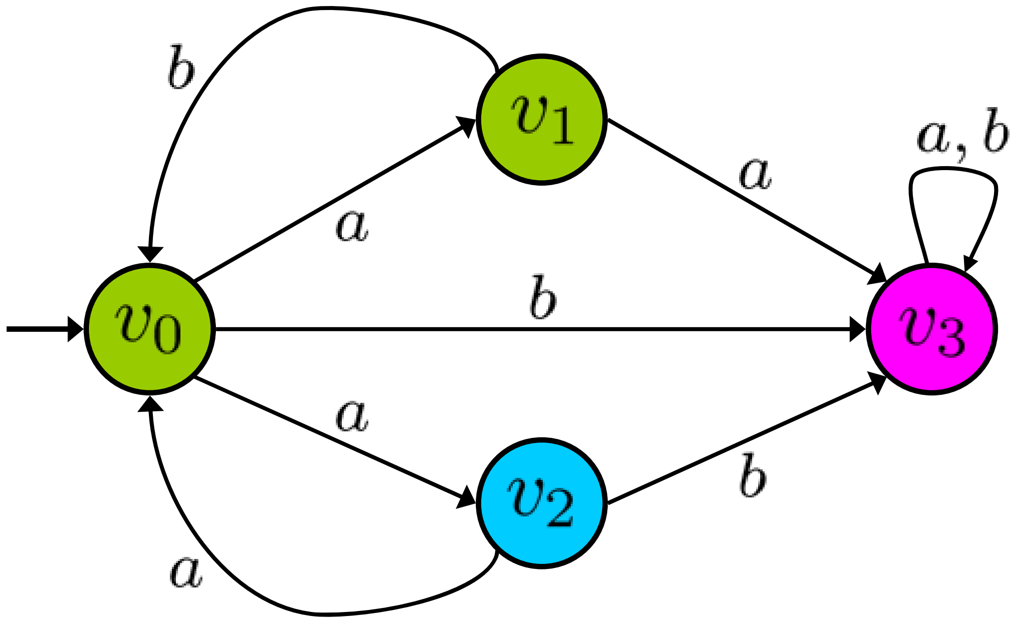
$$X_2 = \big\{\{v_0\}, \{v_1\}, \{v_2\}\big\}$$

Fixed point

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



$$X_0 = \{\{v_0, v_1\}, \{v_2\}\}$$

$$X_1 = \{\{v_0\}, \{v_1\}, \{v_2\}\}$$

$$X_2 = \{\{v_0\}, \{v_1\}, \{v_2\}\}$$

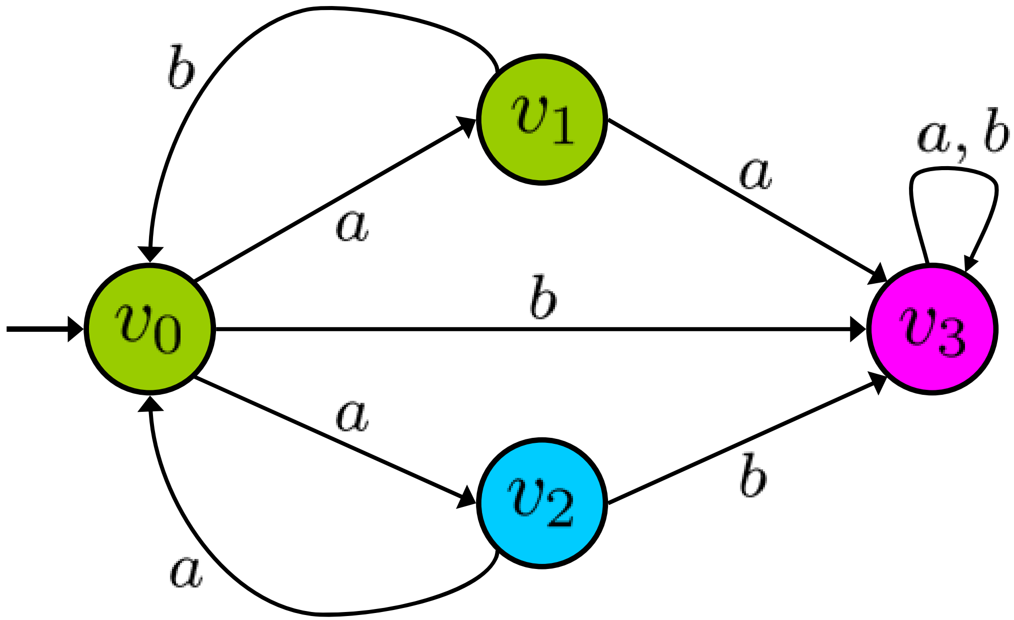
Fixed point

Player 1 is winning since $\{v_0\} \in X_2$

Solving safety games

$\mathcal{T} = \mathsf{Obs} \setminus \{o_3\}$

Objective: $\mathsf{Safe}(\mathcal{T})$



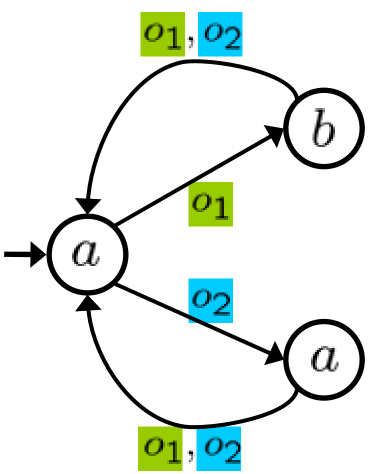
$X_0 = \{\{v_0, v_1\}, \{v_2\}\}$

$X_1 = \{\{v_0\}, \{v_1\}, \{v_2\}\}$

$X_2 = \{\{v_0\}, \{v_1\}, \{v_2\}\}$

Fixed point

A winning strategy:



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Remarks

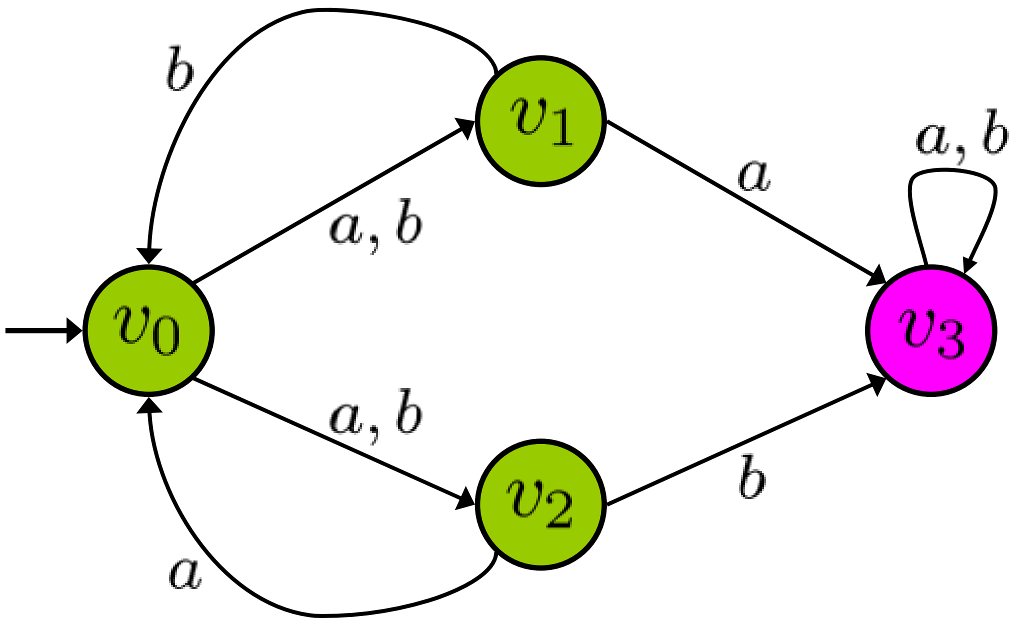
1. **Finite memory** may be necessary to win safety and reachability games of imperfect information, and therefore also for Büchi, coBüchi, and parity objectives.

Remarks

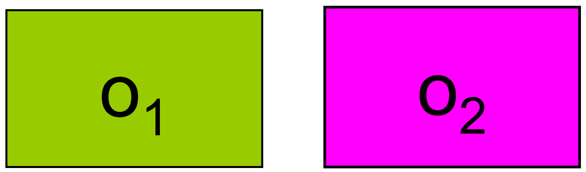
1. **Finite memory** may be necessary to win safety and reachability games of imperfect information, and therefore also for Büchi, coBüchi, and parity objectives.

2. Games of imperfect information are **not determined**.

Non determinacy



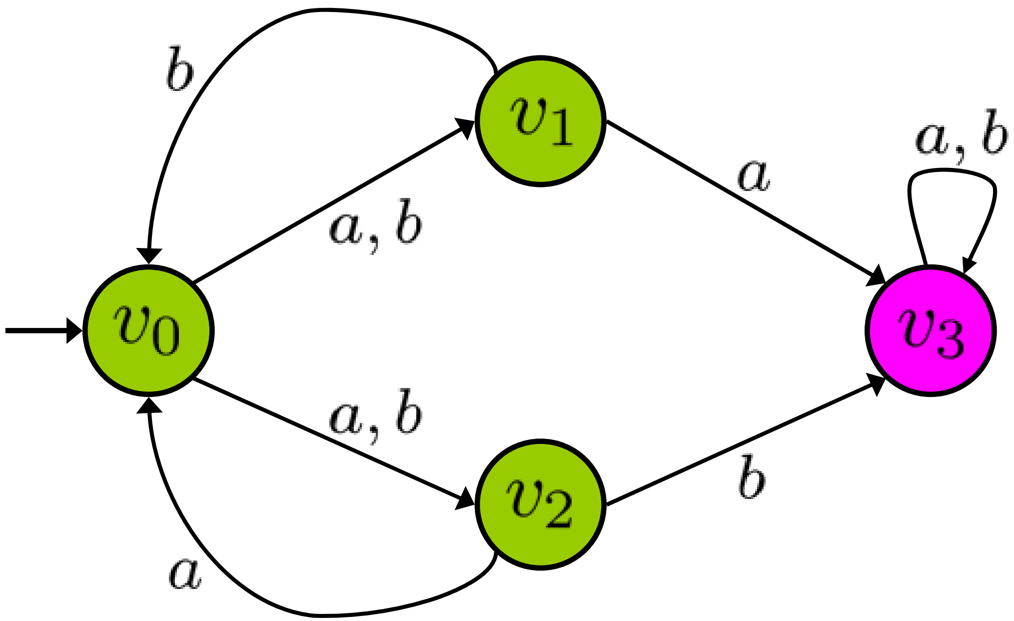
Objective: Reach($\{v_3\}$)



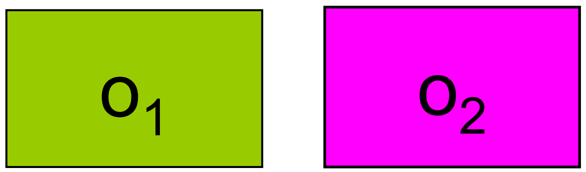
Any fixed strategy λ_1 of Player 1 can be spoiled by a strategy λ_2 of Player 2 as follows:

In v_0: λ_2 chooses v_1 if in the next step λ_1 plays b, and
λ_2 chooses v_2 if in the next step λ_1 plays a.

Non determinacy



Objective: $\mathsf{Reach}(\{v_3\})$



Player 1 cannot enforce $\mathsf{Reach}(\{v_3\})$.

Similarly, Player 2 cannot enforce $\mathsf{Safe}(\{v_0, v_1, v_2\})$.

because when a strategy λ_2 of Player 2 is fixed, either $\lambda_1(o_1 o_1) = a$ or $\lambda'_1(o_1 o_1) = b$ is a spoiling strategy for Player 1.

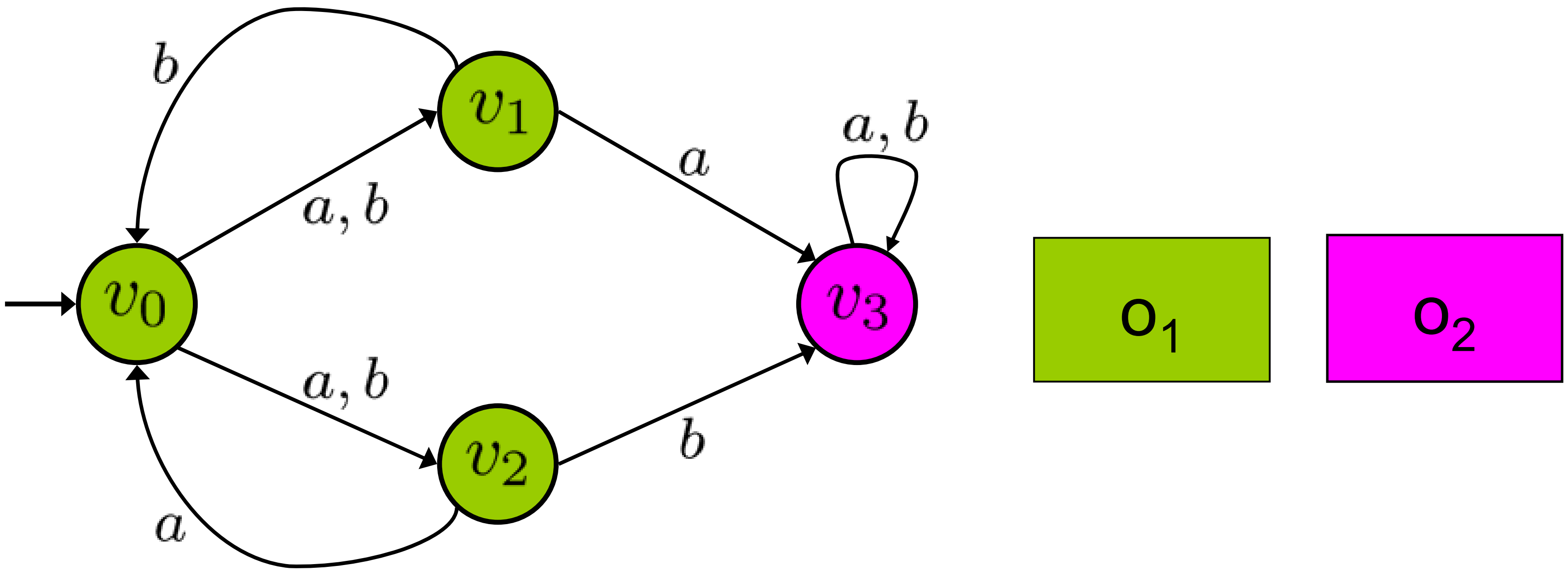
Remarks

1. **Finite memory** may be necessary to win safety and reachability games of imperfect information, and therefore also for Büchi, coBüchi, and parity objectives.

2. Games of imperfect information are **not determined**.

3. **Randomized** strategies are more powerful, already for reachability objectives.

Randomization



The following strategy of Player 1 wins with probability 1:

At every step, play a and b uniformly at random.

After each visit to $\{v_1, v_2\}$, no matter the strategy of Player 2, Player 1 has probability $\frac{1}{2}$ to win (reach v_3).

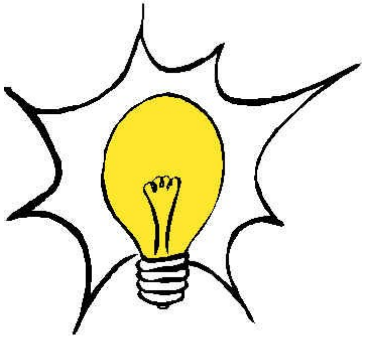
Summary

Conclusion

- Games for controller synthesis: symbolic algorithms using fixpoint formulas.
- Imperfect information is more realistic, gives more robust controllers; but exponentially harder to solve.
- Antichains: exploit the structure of the subset construction.

Conclusion

- Games for controller synthesis: symbolic algorithms using fixpoint formulas.
- Imperfect information is more realistic, gives more robust controllers; but exponentially harder to solve.
- Antichains: exploit the structure of the subset construction.



It is sufficient to keep only the **maximal elements**.

Conclusion

- The antichain principle has applications in other problems where subset constructions are used:
  - Finite automata: language inclusion, universality, etc. [De Wulf,D,Henzinger,Raskin 06]
  - Alternating Büchi automata: emptiness and language inclusion. [D,Raskin 07]
  - LTL: satisfiability and model-checking. [De Wulf,D,Maquet,Raskin 08]

Alaska

Antichains for Logic, Automata and Symbolic Kripke Structure Analysis

http://www.antichains.be

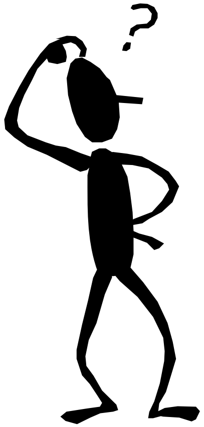
Acknowledgments

Credits

Antichains for games is a joint work with Krishnendu Chatterjee, Martin De Wulf, Tom Henzinger and Jean-François Raskin.

Special thanks to Jean-François Raskin for slides preparation.

Thank you !



Questions ?