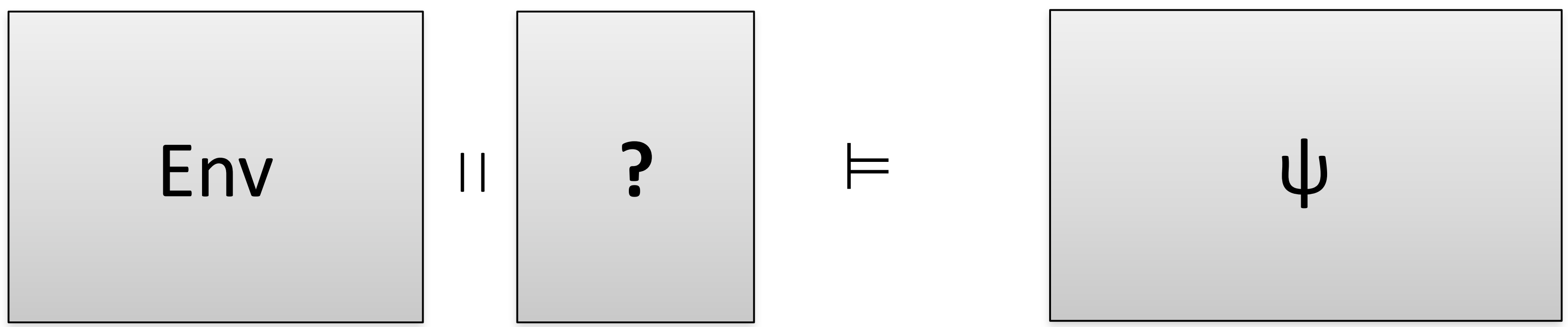
Non-zero Sum Games for Reactive Synthesis

Jean-François Raskin
Université Libre de Bruxelles

Joint works with Romain Brenguier, Véronique Bruyère, Krishnendu Chatterjee, Lorenzo Clemente, Laurent Doyen, Emmanuel Filiot, Paul Hunter, Noémie Meunier, Guillermo Perez, Mickael Randour, Ocan Sankur, Mathieu Sassolas.

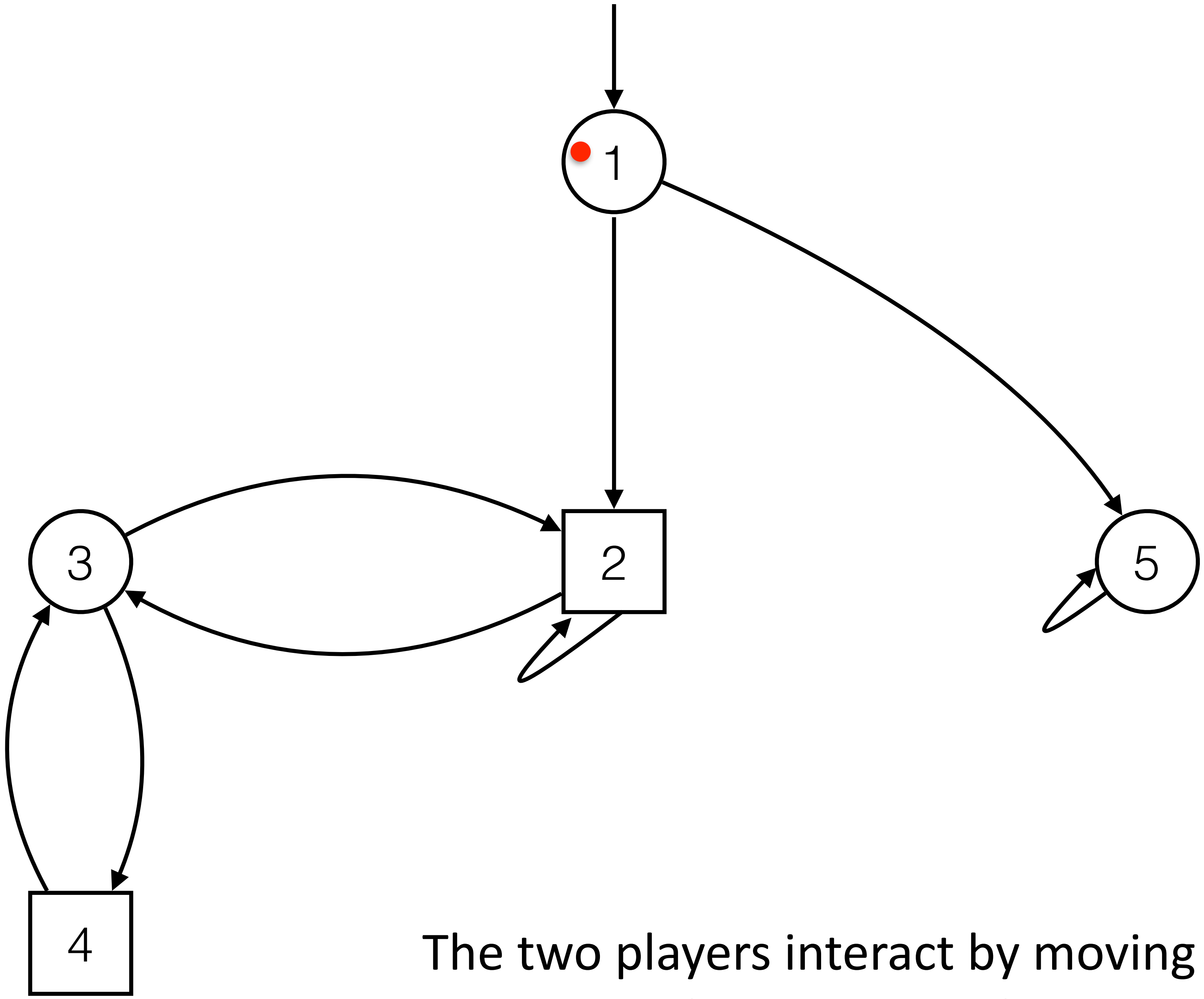
Synthesis of reactive systems

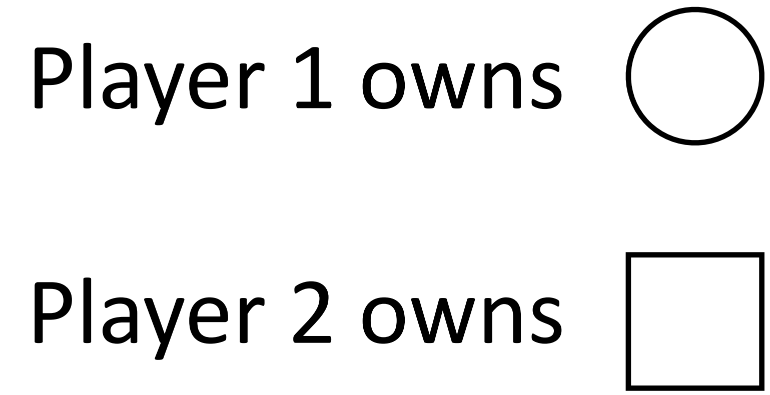
The classical setting



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

Winning strategy = Correct Sys



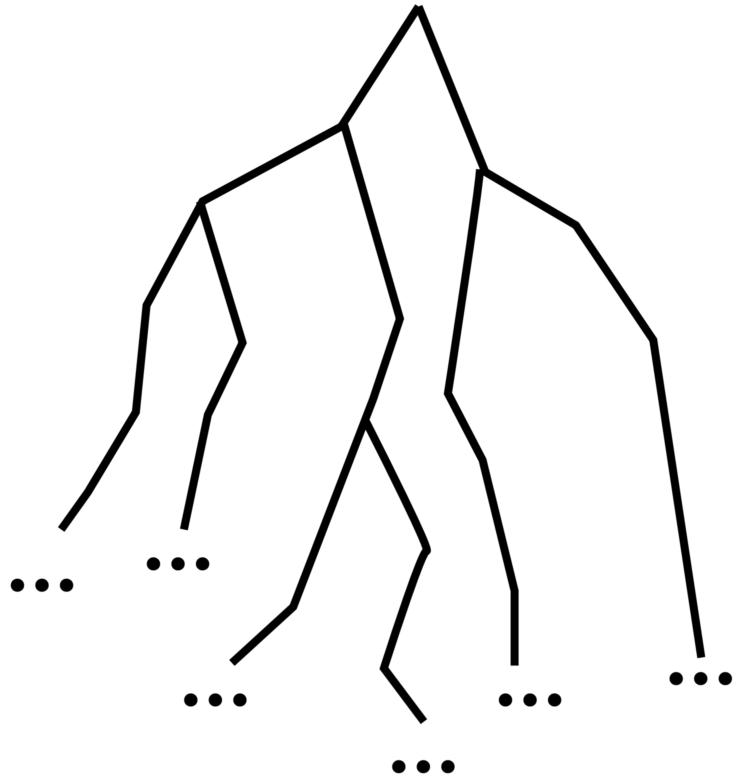


The two players interact by moving a token along the edges of the graph to form an **infinite path**(=outcome)

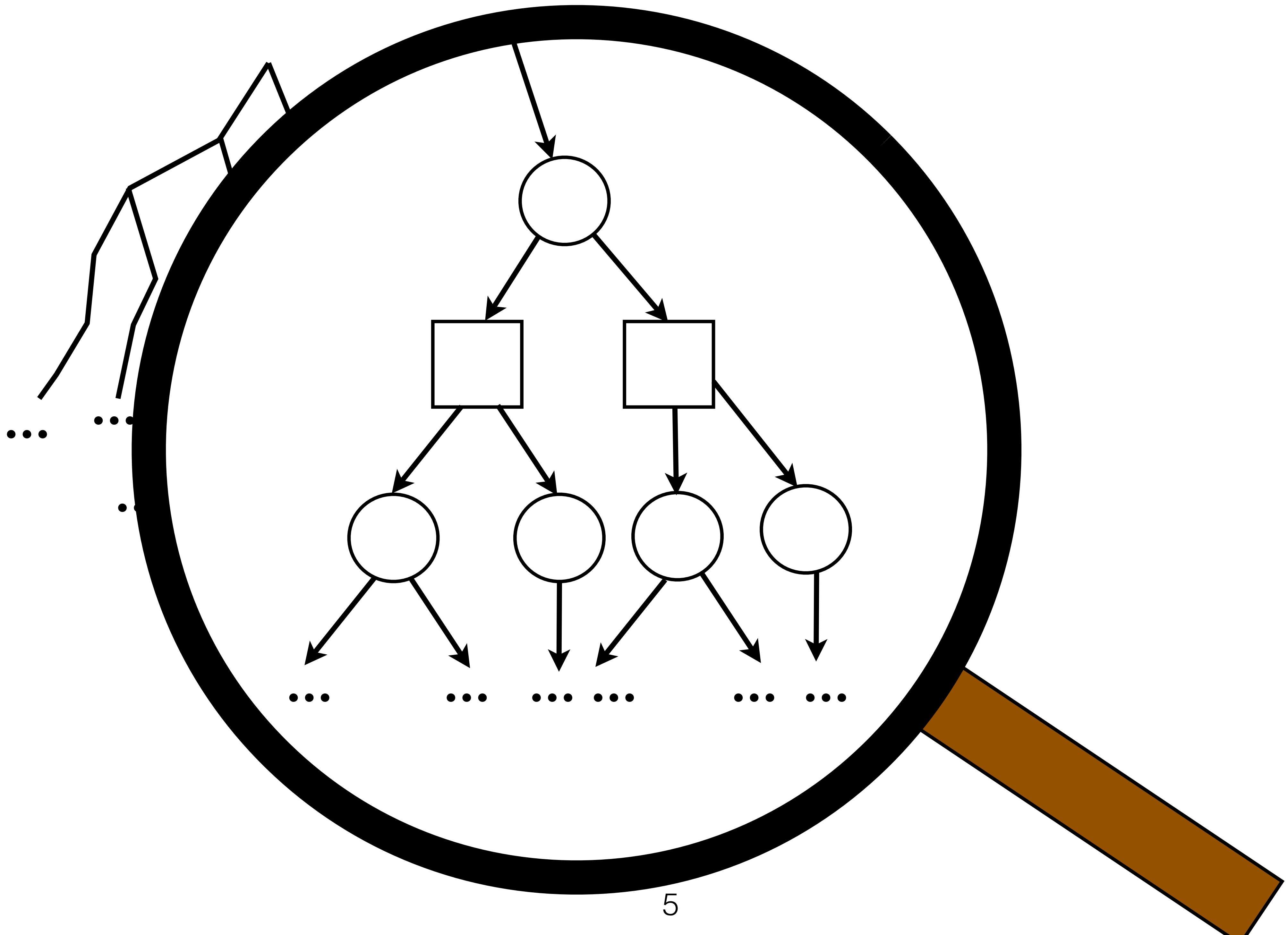
In each round, the owner of the current vertex chooses where to move the token

Strategies

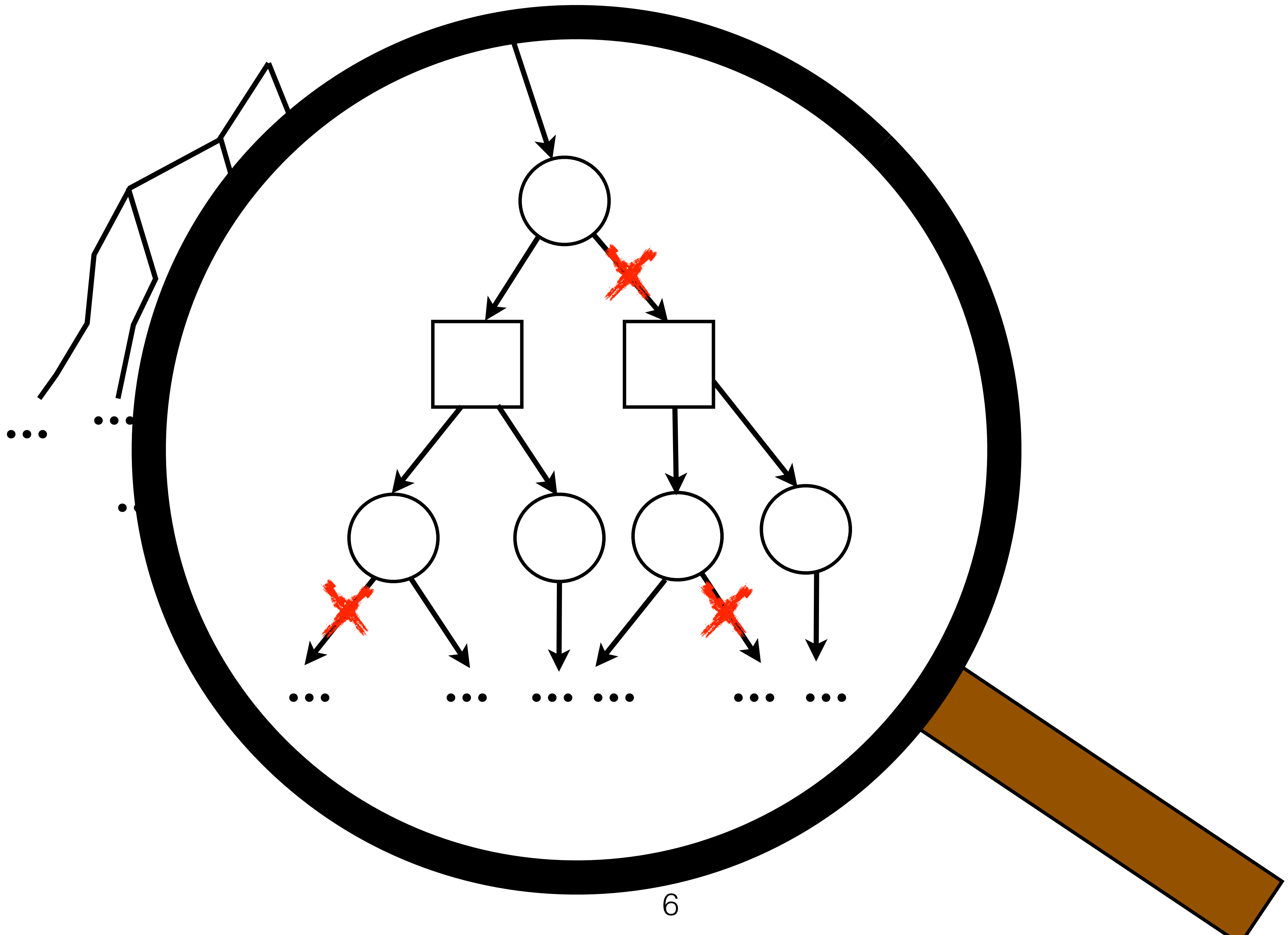
Unfolding of the game graph



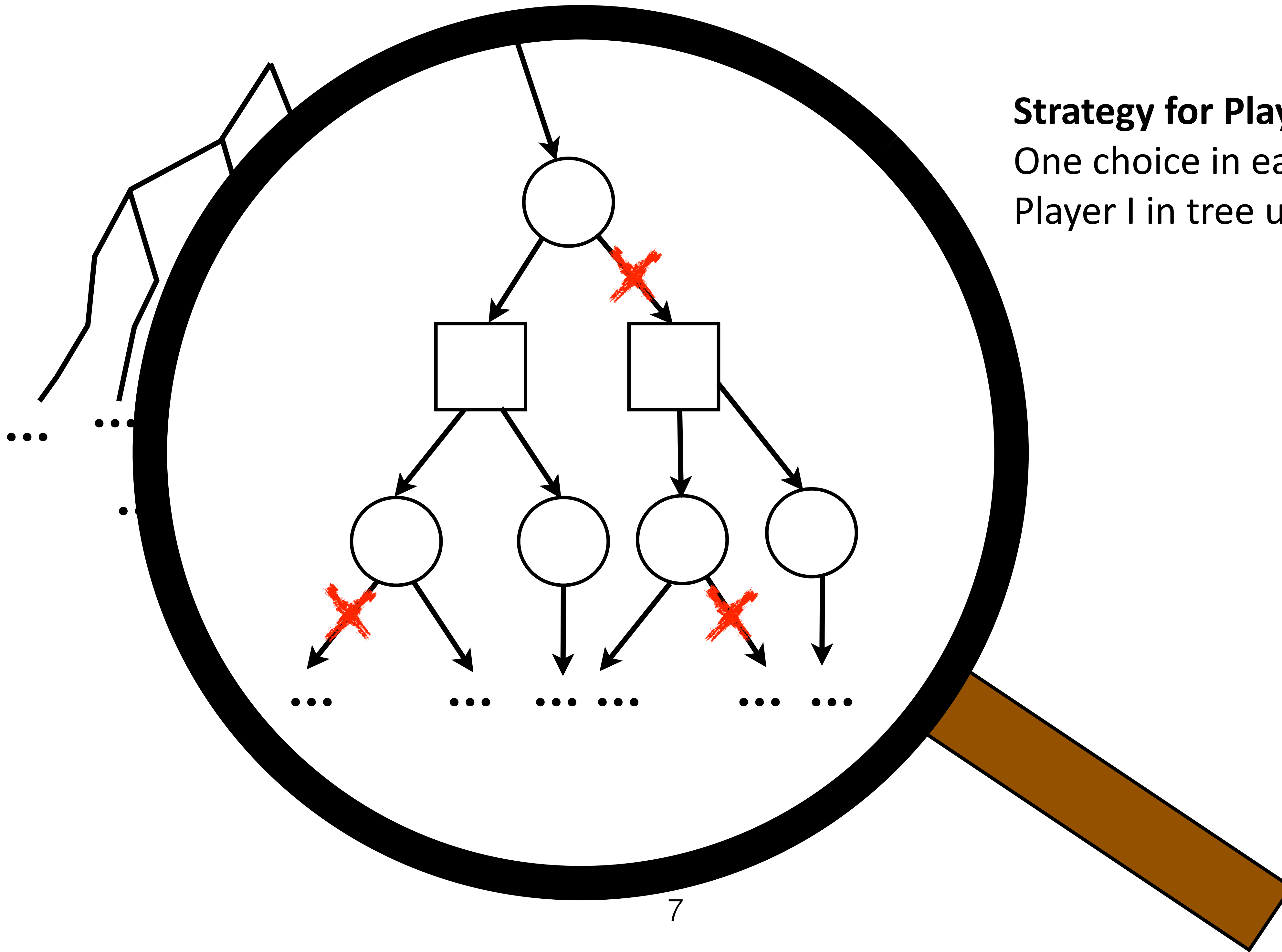
Strategies



Strategies

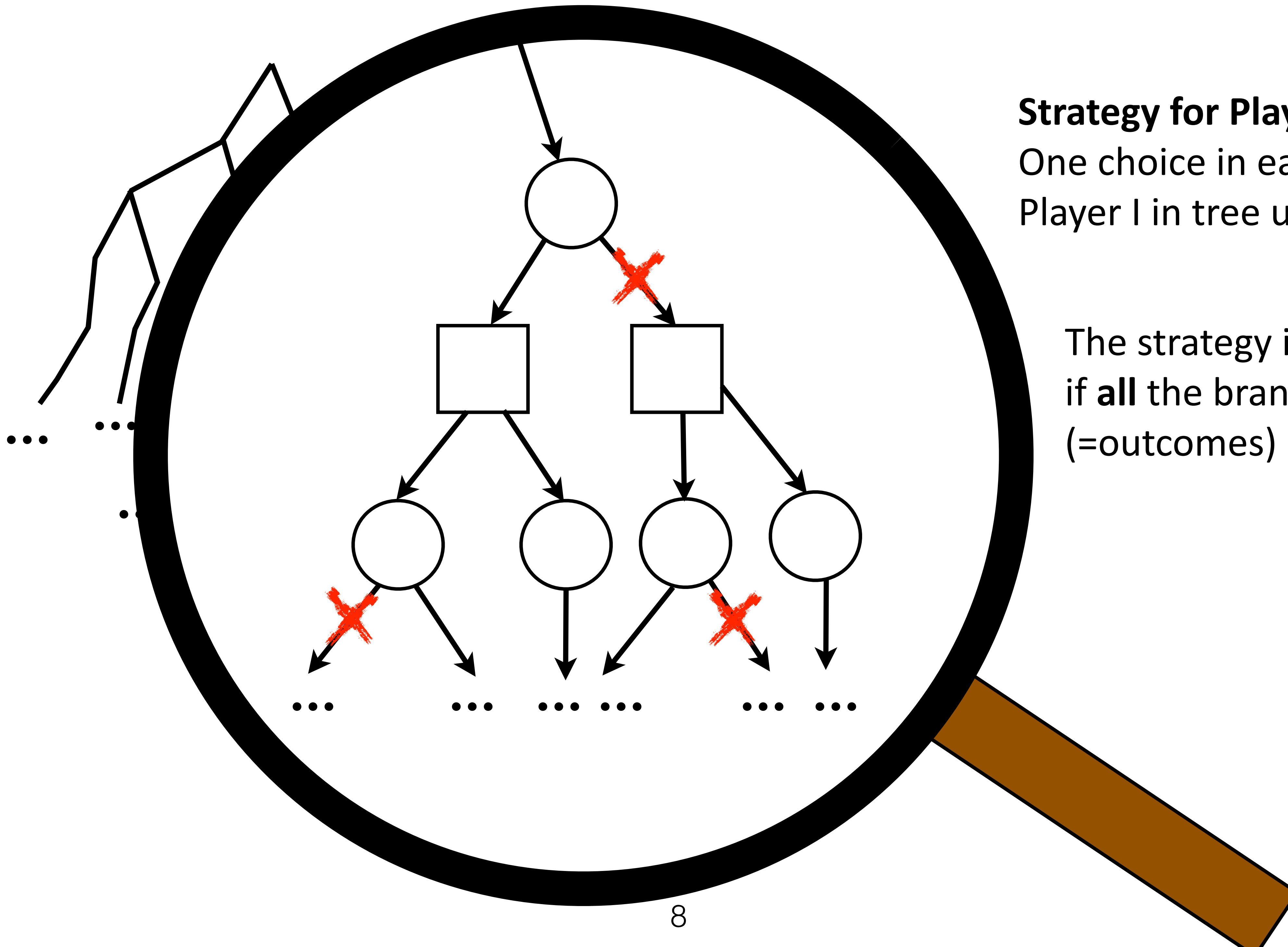


Strategies



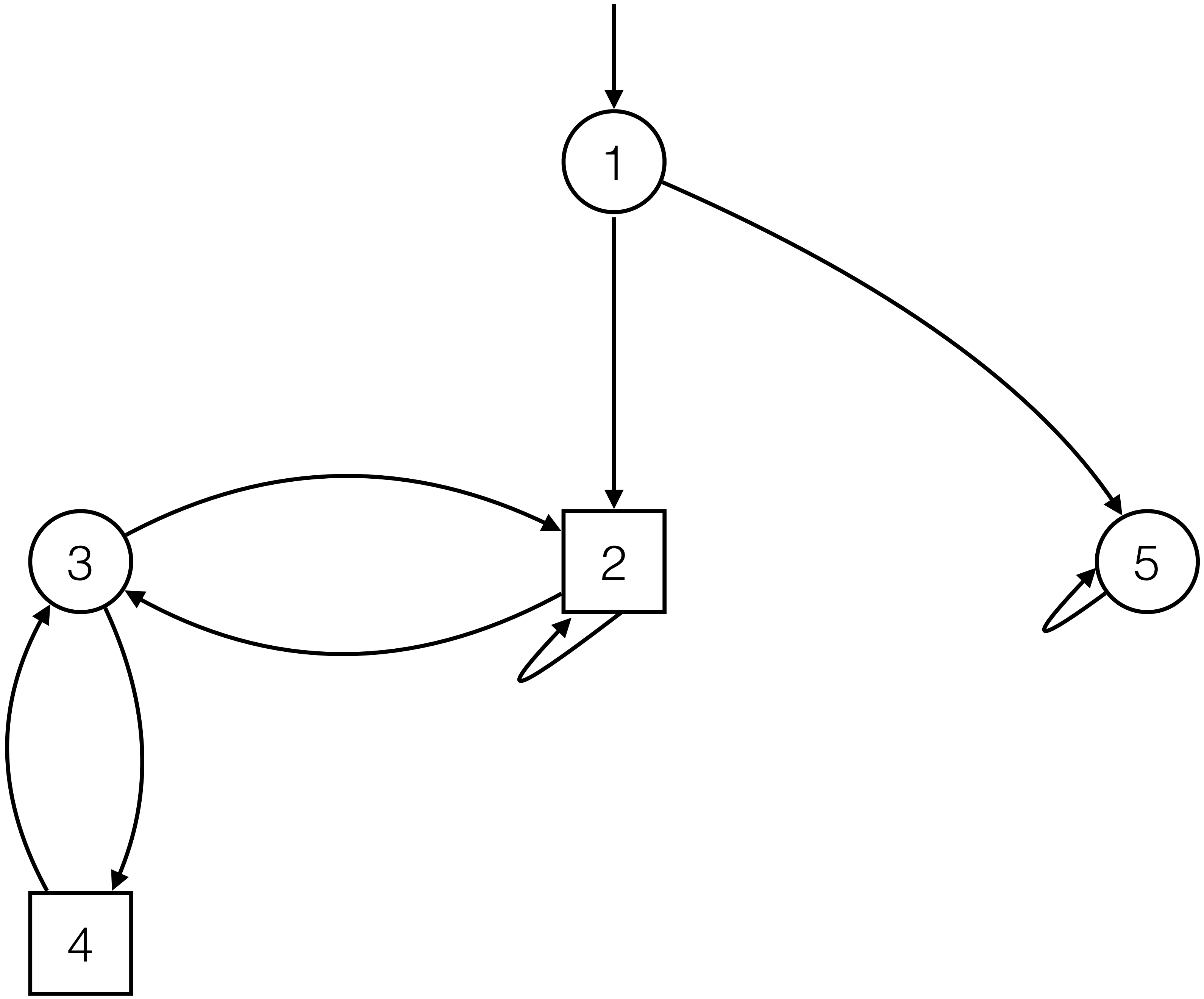
Strategy for Player 1 =
One choice in each node of
Player I in tree unfolding

Strategies

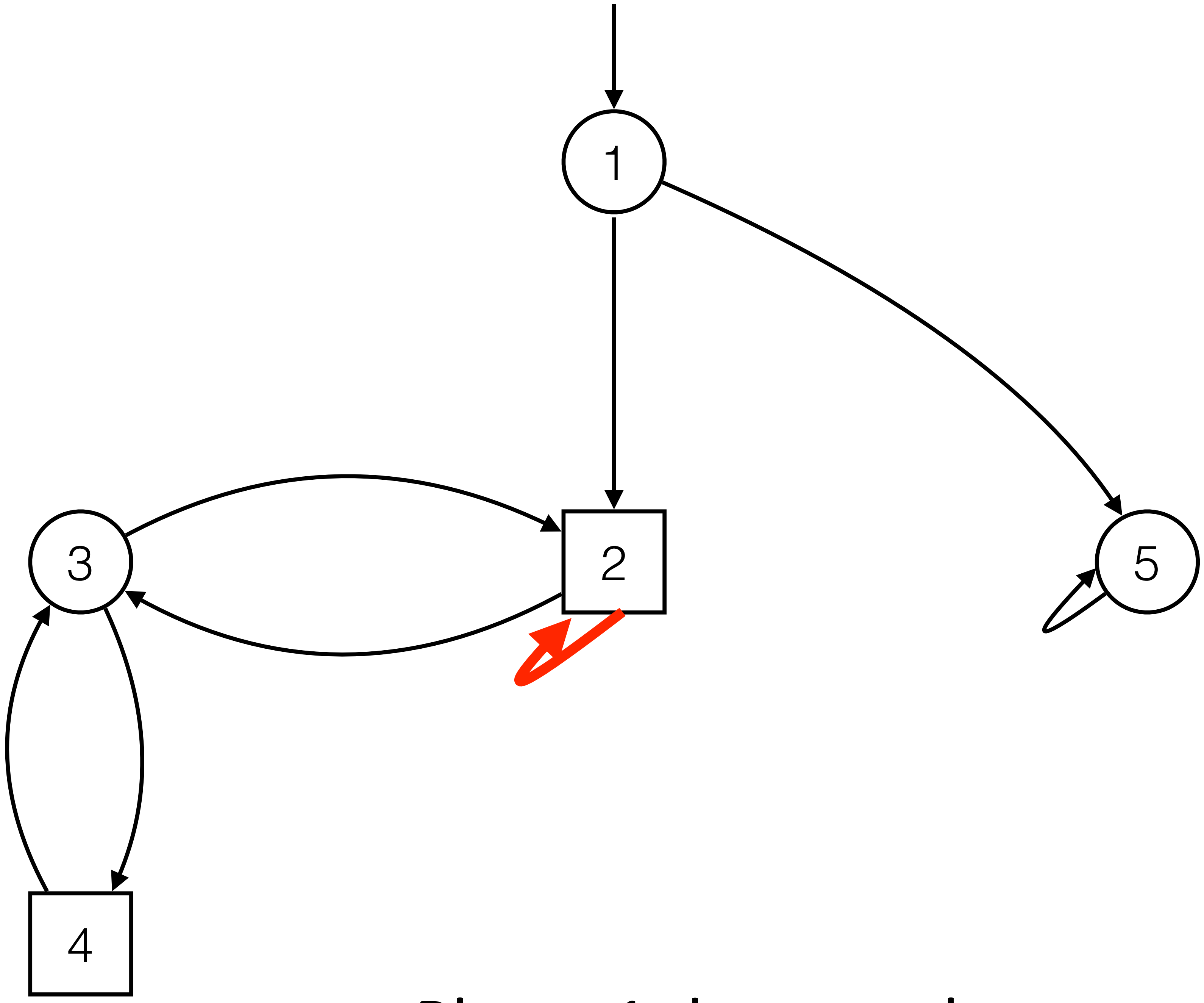


Strategy for Player 1 =
One choice in each node of
Player I in tree unfolding

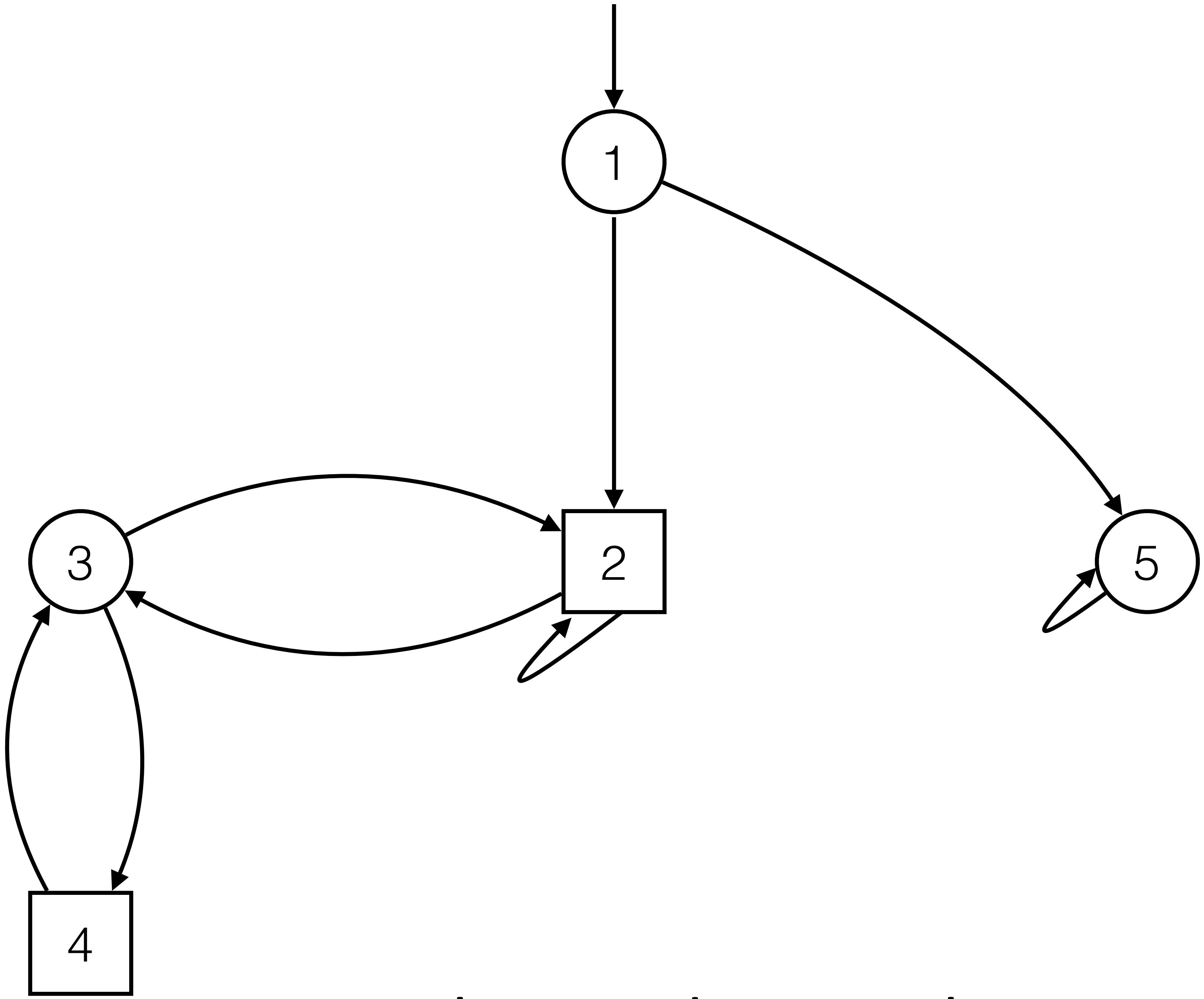
The strategy is **winning**
if **all** the branches
(=outcomes) are winning



$Win_1 = \{ \rho \mid \rho \models \square \lozenge 4 \}$



Player 1 does **not** have a winning strategy for $Win_1 = \{ \rho \mid \rho \models \Box \Diamond 4 \}$

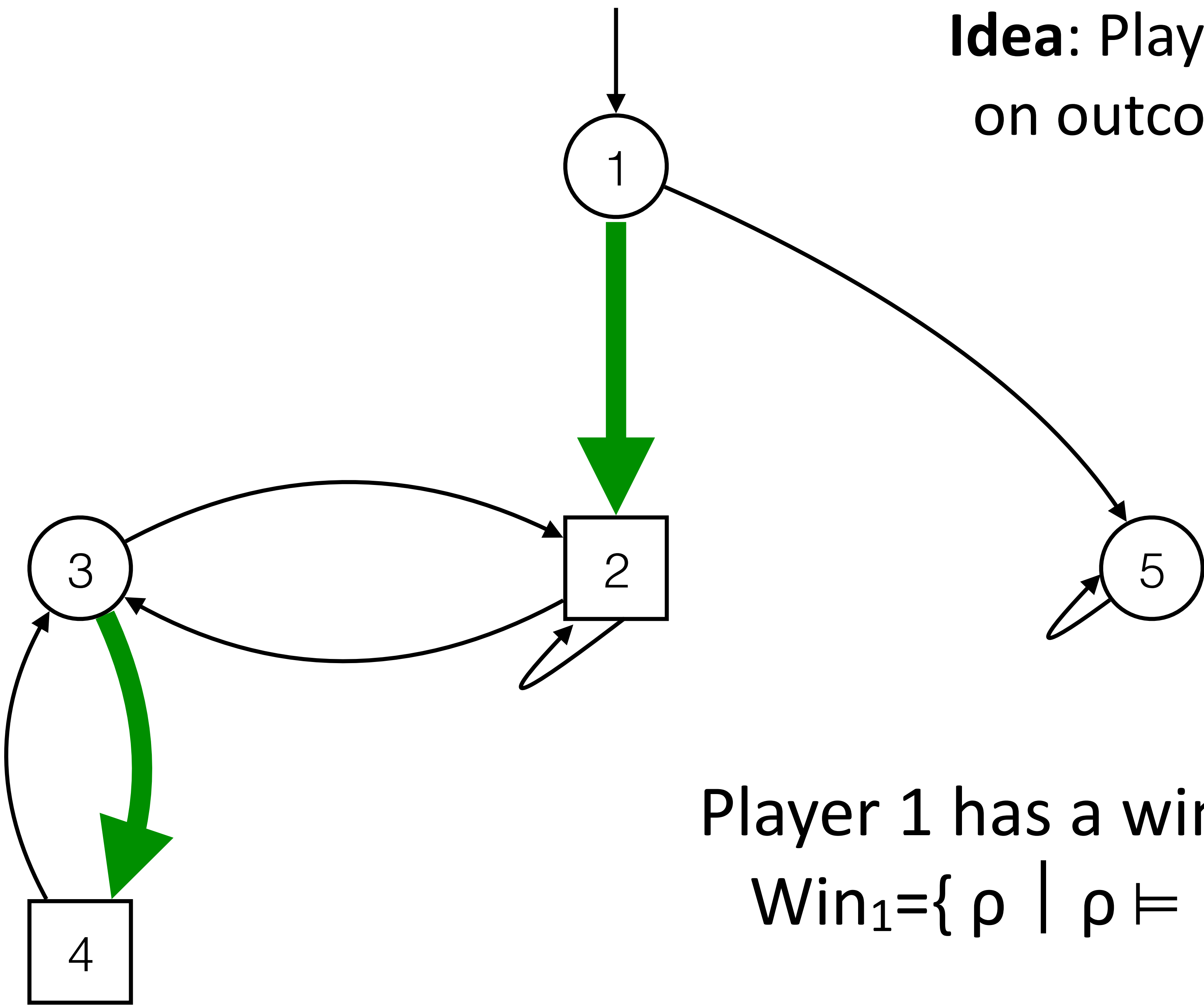


Player 1 does **not** have a winning strategy for Win_1={ ρ | ρ ⊨ □◇4 }

Assume that Win_2={ ρ | ρ ⊨ □◇3 }

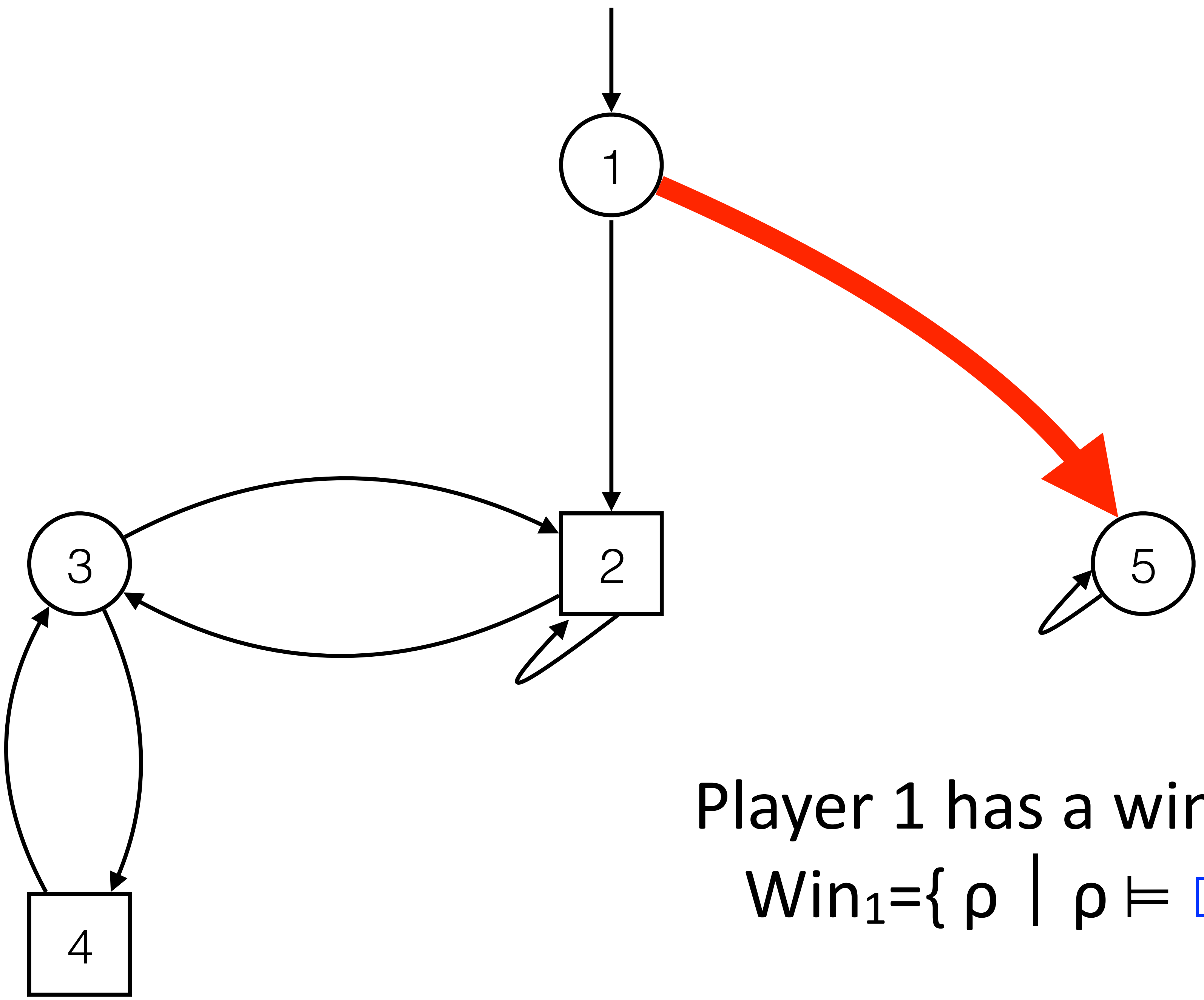
Adding hypothesis …

Idea: Player 1 needs to win only on outcomes that are winning for Player 2



Player 1 has a winning strategy for $Win_1 = \{ \rho \mid \rho \models \Box \Diamond 3 \rightarrow \Box \Diamond 4 \}$

Adding hypothesis …



Player 1 has a winning strategy for
$Win_1 = \{ \rho \mid \rho \models \Box \Diamond 3 \rightarrow \Box \Diamond 4 \}$

But this is also a winning strategy !

We need richer foundations !

1. Synthesis under **hypothesis** about the behavior of Env.,
   i.e. $\psi \equiv \alpha \Longrightarrow \beta$.

   But what if strategy of Sys is to force $\neg\alpha$?

2. A system is made of **components**, i.e. Sys = $C_1 \parallel C_2 \parallel \ldots \parallel C_n \parallel C_{n+1}$(=E)

   The objective of one component is **not** to falsify the objectives of other components: **each component has its own specification** (winning objective).

   But, on the contrary, we may **not hope for collaboration**: components should be considered as **partially adversarial**.

3. … other options if environment is **not** adversarial:
   -we may want to play a « **reasonable »** strategy (regret minimization).
   -expected behavior specified by a **stochastic model**.

We need richer foundations !

1. Synthesis under **hypothesis** about the behavior of Env., i.e. ...

   Research agenda

   B... ce ¬α ?

2. A system is made of **components**, i.e. ...

   Apply/define (new) solution concepts from game theory

   The o... the objectives of other comp... **its own specification** (winning...

   But, on the contrary, we may **not h**... should be considere...

   Study algorithmic support

3. ... other o... **not** adversarial:
   -we ma... play a « **reasonable** » strategy (regret minimization).
   -expected behavior specified by a **stochastic model**.

Plan of the talk

- **Doomsday** equilibria
  (refinement of Nash/secure equilibria)

- Assume **admissible** synthesis

- **Regret** minimization in reactive synthesis

- **Mixing** worst-case and expectation requirements

Secure/Doomsday Equilibria

[VMCAI 2014,LICS2014]

Nash equilibria
Secure equilibria

To predict/analyze how players behave in multi-player games, several notions have been proposed a.o.:

- **Nash equilibria** [Nash51]:
  a strategy profile$(St_1,St_2,\ldots,St_n)$ is a **NE**
  <u>if no player has an incentive to **unitarily** deviate</u>:
  $Out_1(\mathbf{St_1'},St_2,\ldots,St_n) \leq Out_1(St_1,St_2,\ldots,St_n)$

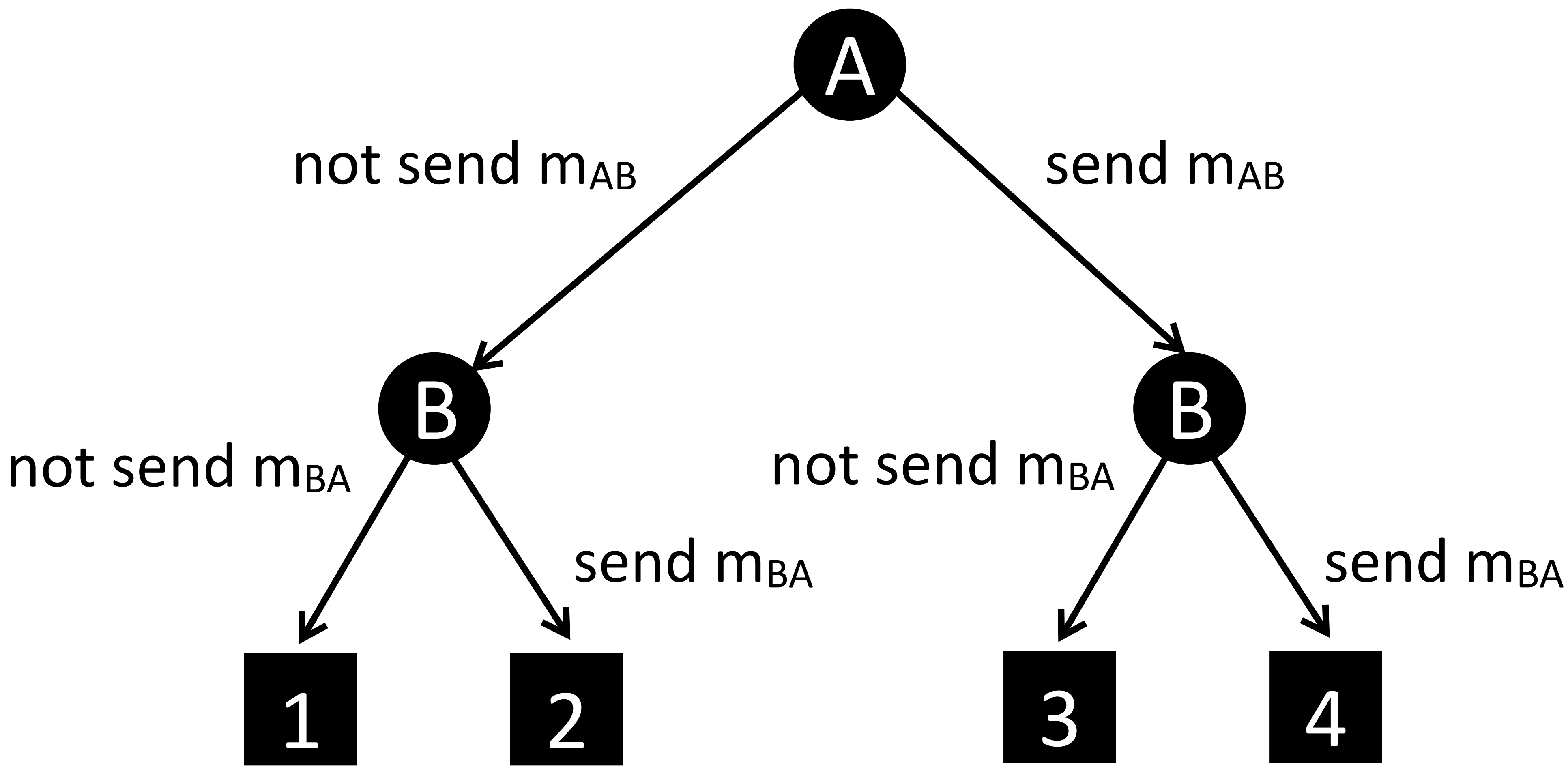
- **Secure equilibria** (2 players) [CHJ06]:
  <u>NE+deviation does **not** harm the other player</u>:
  $Out_1(\mathbf{St'_1},St_2) = Out_1(St_1,St_2) \Longrightarrow Out_2(\mathbf{St'_1},St_2) \geq Out_2(St1,St2)$
  = each player first tries to satisfy his/her goal (**primary** objective) and then his/her goal is to fasilfy the goal of the other player (**secondary** objective)

A simple example

- Alice and Bob want to exchange messages m_{AB} and m_{BA}
- Either
  - **both** have received their message (preferred)
  - or none (sub-optimal)
- If one receives and the other not, this is **not** acceptable (for the one that does not receive)

- ≈ spec. of “Fair Exchange Protocols”
  - no easy solution (e.g. need for a TTP)

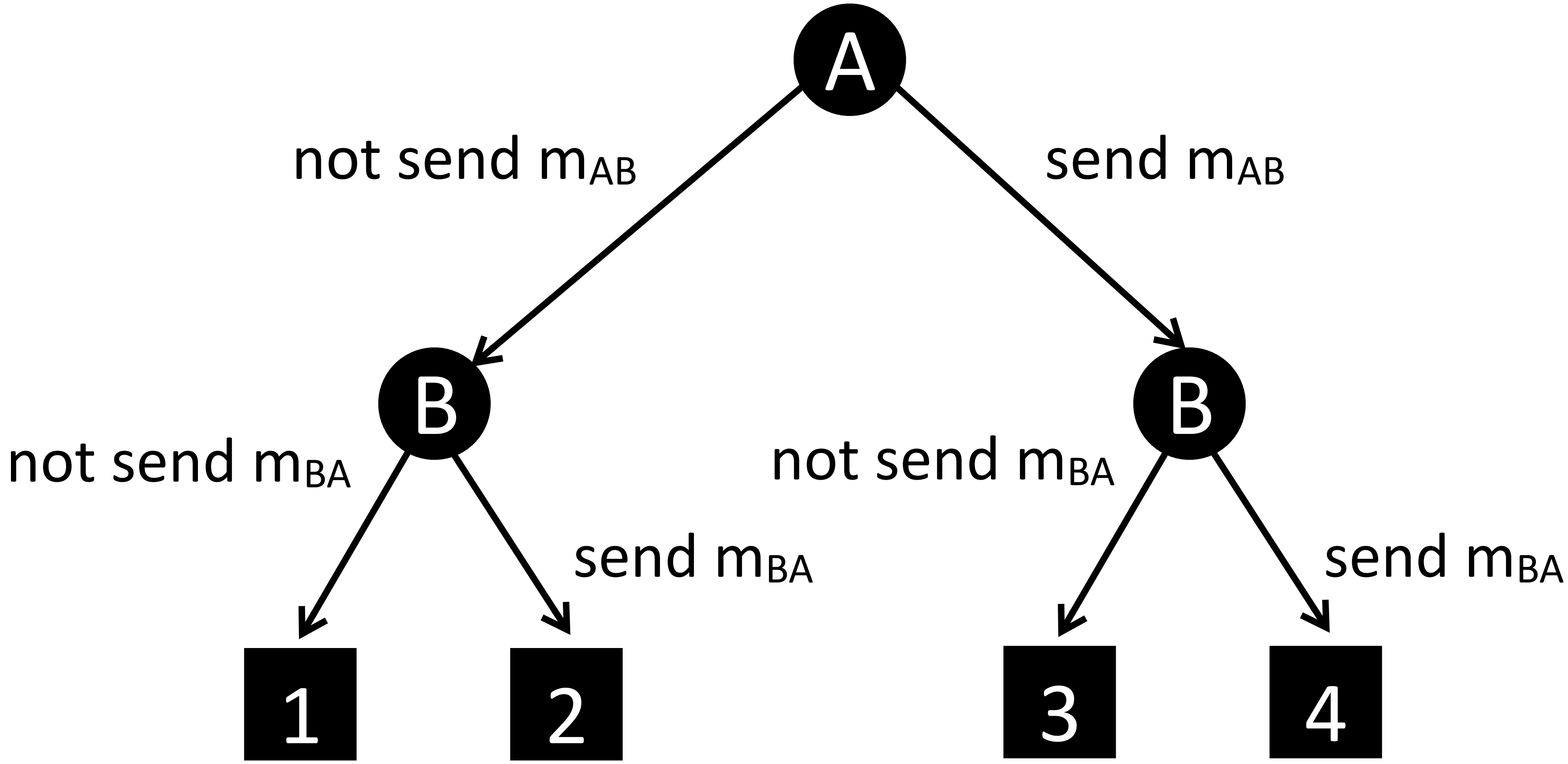
A simple example



A wants to reach {2,4}

B wants to reach {3,4}

A simple example

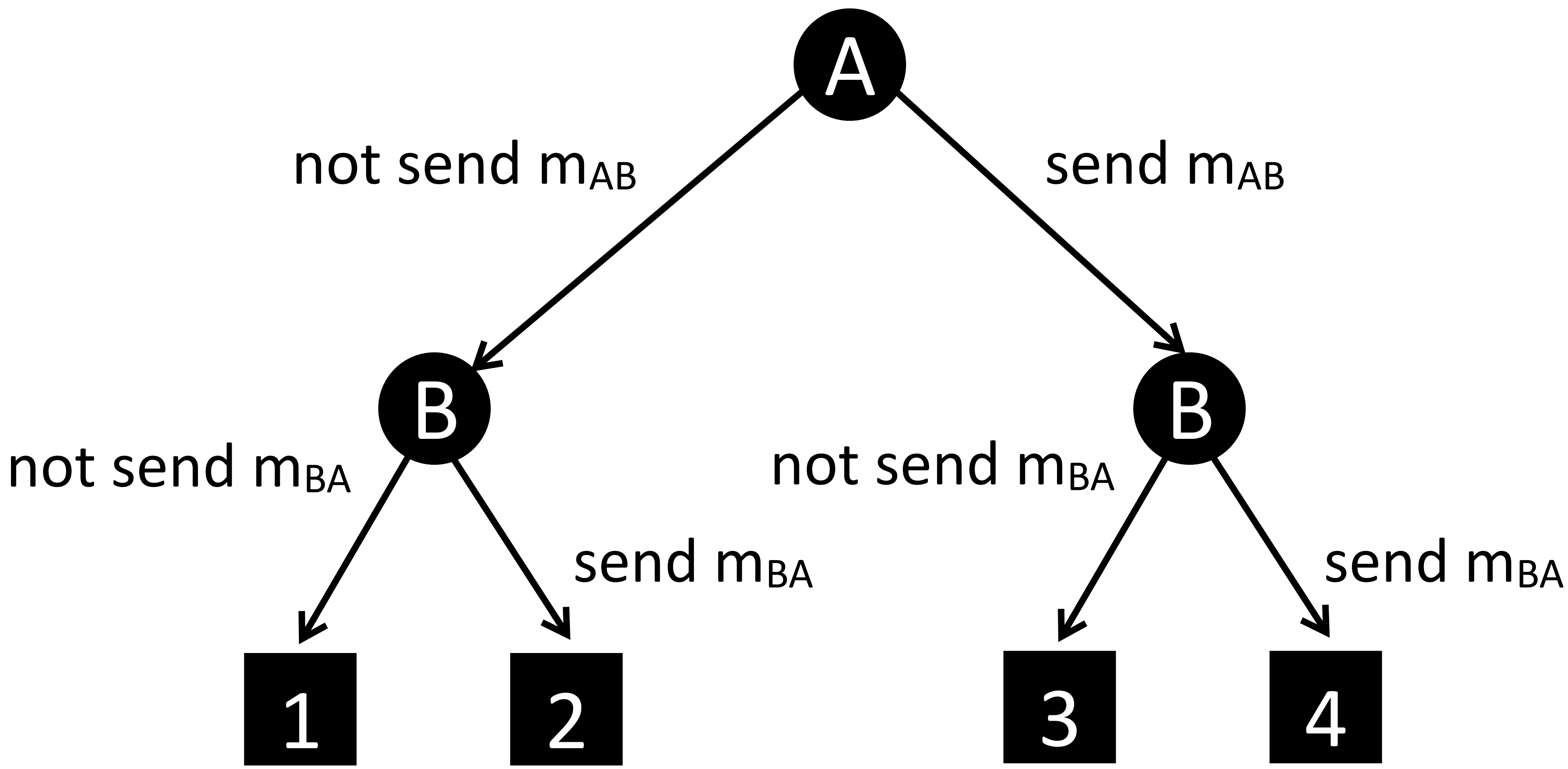


A wants to reach {2,4}

B wants to reach {3,4}

None of the players has a winning strategy for his/her objective

A simple example



A preference: 2>4>1>3

B preference: 3>4>1>2

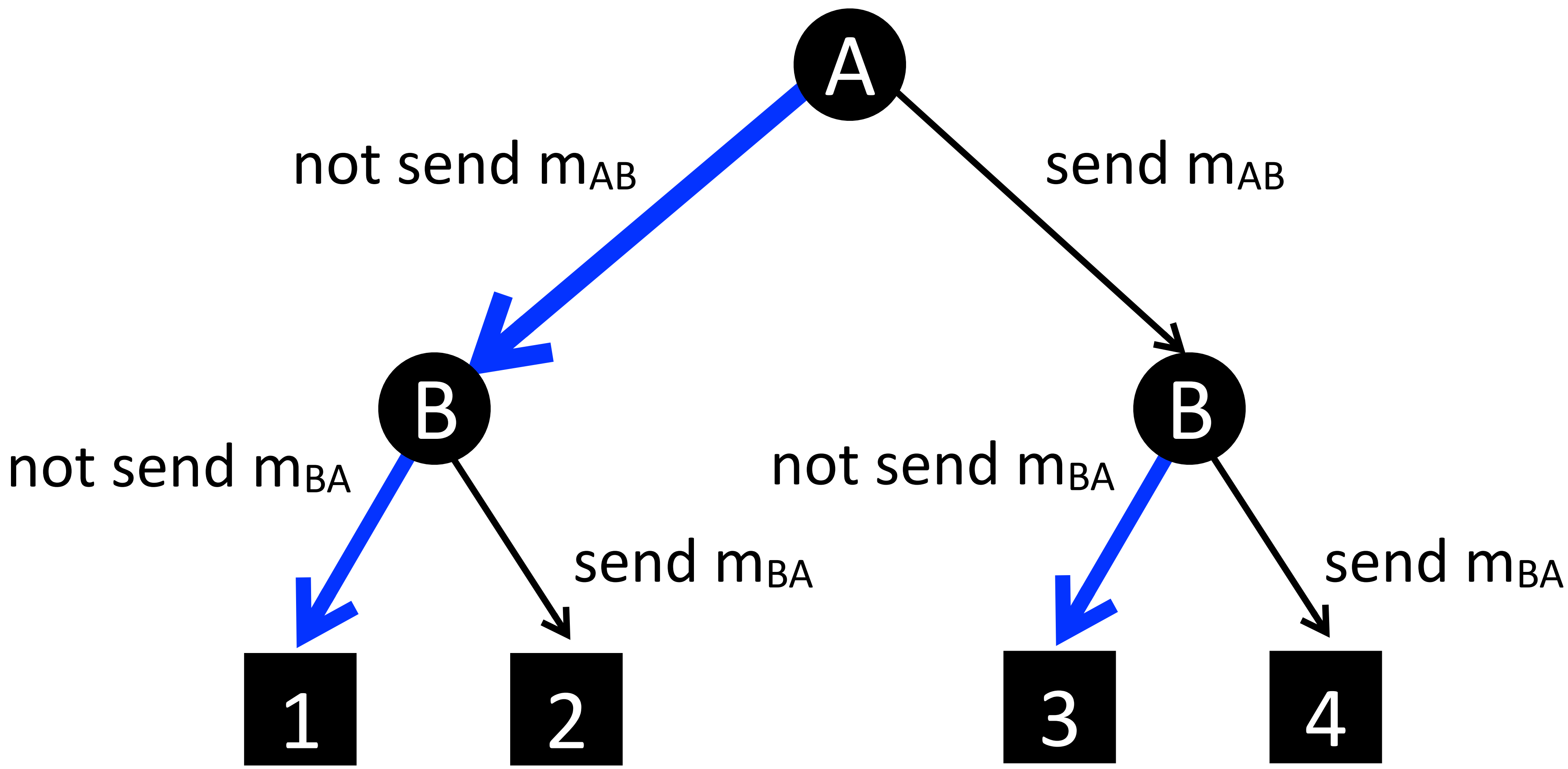
Nash equilibria
Secure equilibria

- **Secure equilibria** (2 players) [CHJ06]:
  NE+deviation does **not** harm the other player:
  $Out_1(\mathbf{St'_1},St_2) = Out_1(St_1,St_2) \Longrightarrow Out_2(\mathbf{St'_1},St_2) \geq Out_2(St1,St2)$

- Objectives of the players: the **preliminary** objective is to satisfy his/her own objective, the **secondary** objective is to falsify the objective of the other player

- Back to the example: A preference: **2>4>1>3**
  2=A wins alone
  4=A and B win
  1=Both A and B fail to win
  3=B wins alone

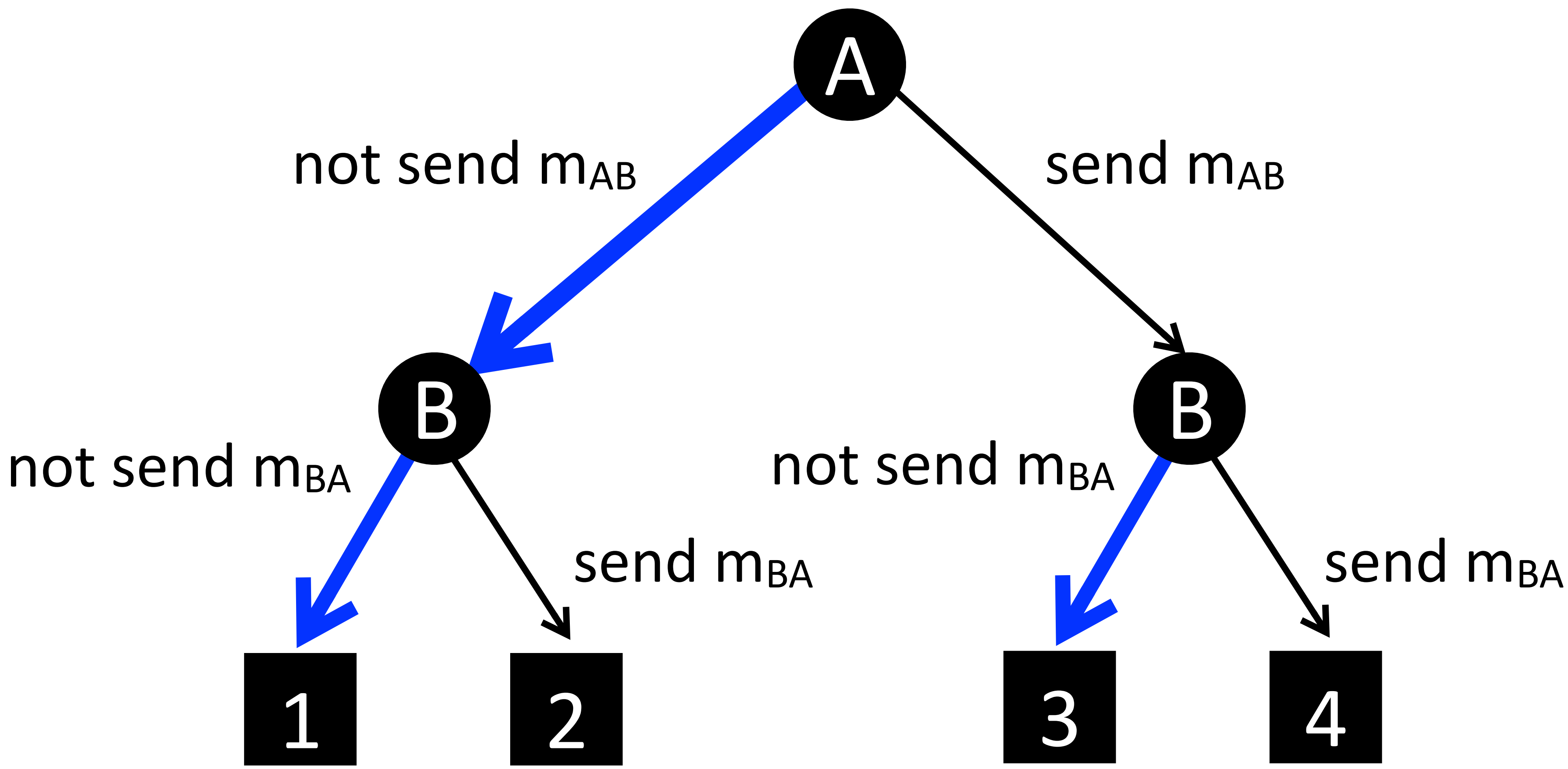
A simple example



A preference: 2>4>1>3

B preference: 3>4>1>2

A simple example



A preference: 2>4>1>3

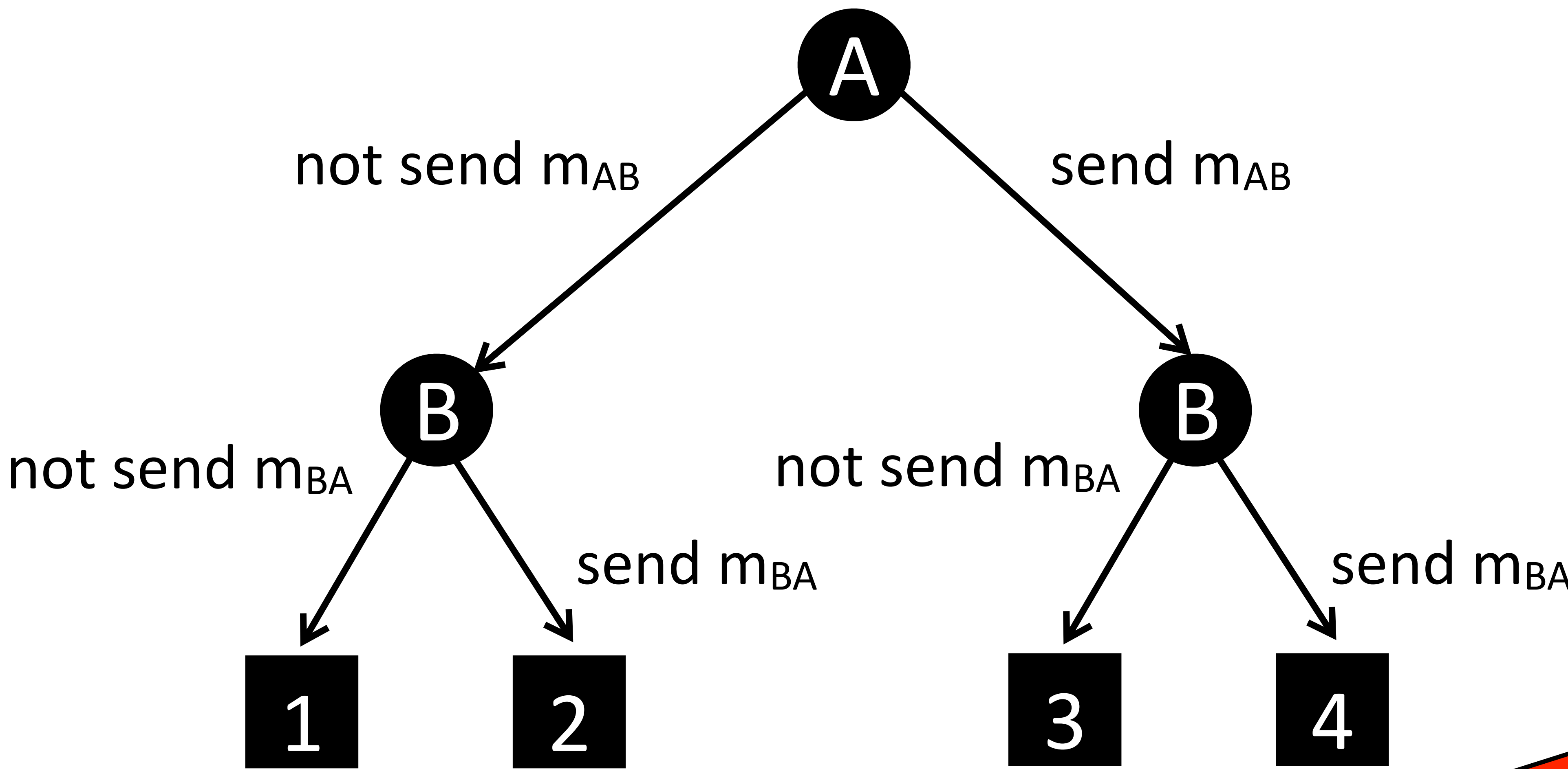
B preference: 3>4>1>2

Unique secure equilibrium:

not send m_{AB}, not send m_{BA}

Not satisfactory !

A simple example



A preference: 2>4>1>3

B preference: 3>4>1>2

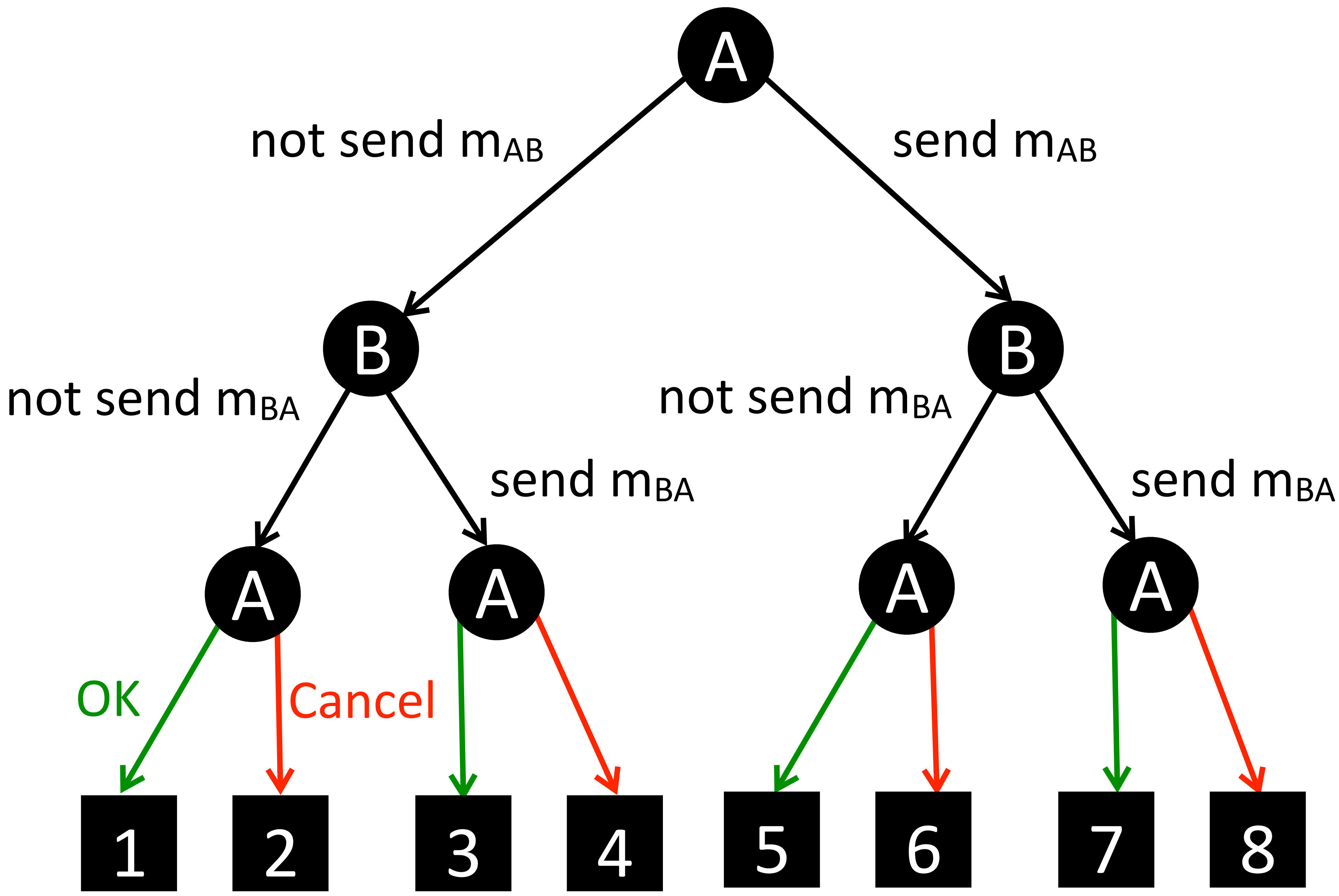
Un… …librium:

…m_{AB}, **not send m_{BA}**

Idea: add cancel (≈TTP)

Not satisfactory !

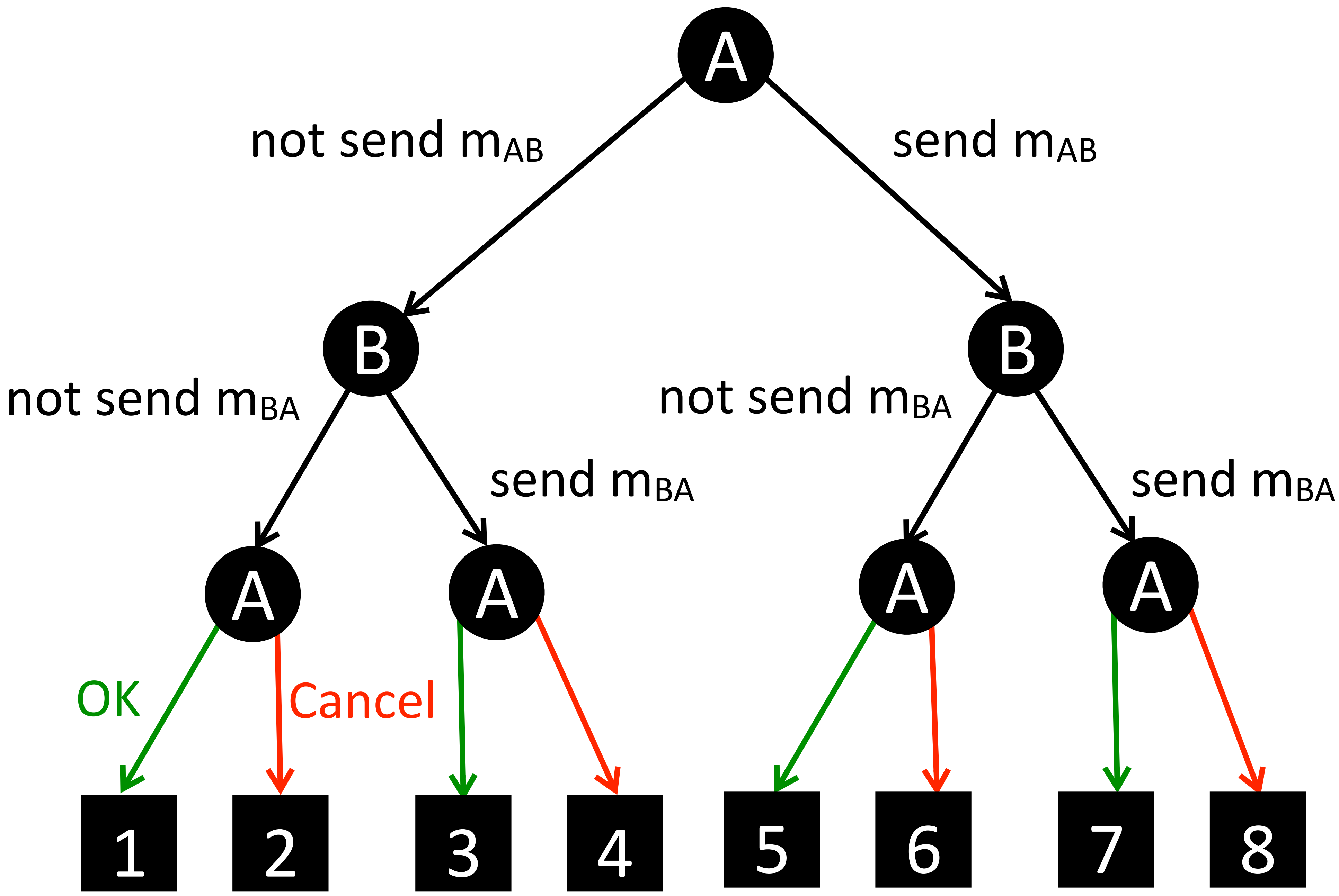
A simple example



A wants to reach {3,7}

B wants to reach {5,7}

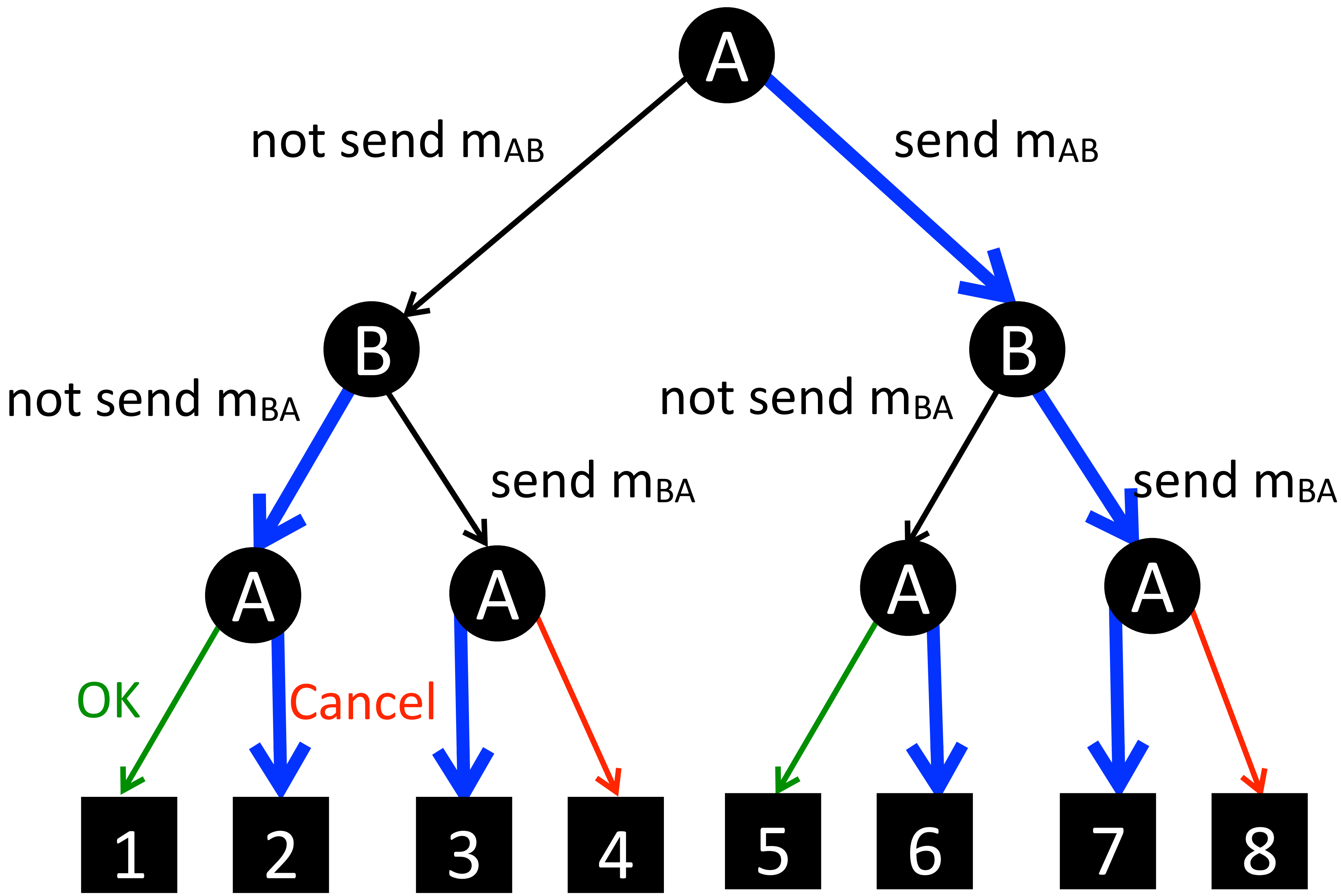
A simple example



A preferences: 3>7>1=2=4=6=8>5

B preferences: 5>7>1=2=4=6=8>3

A simple example



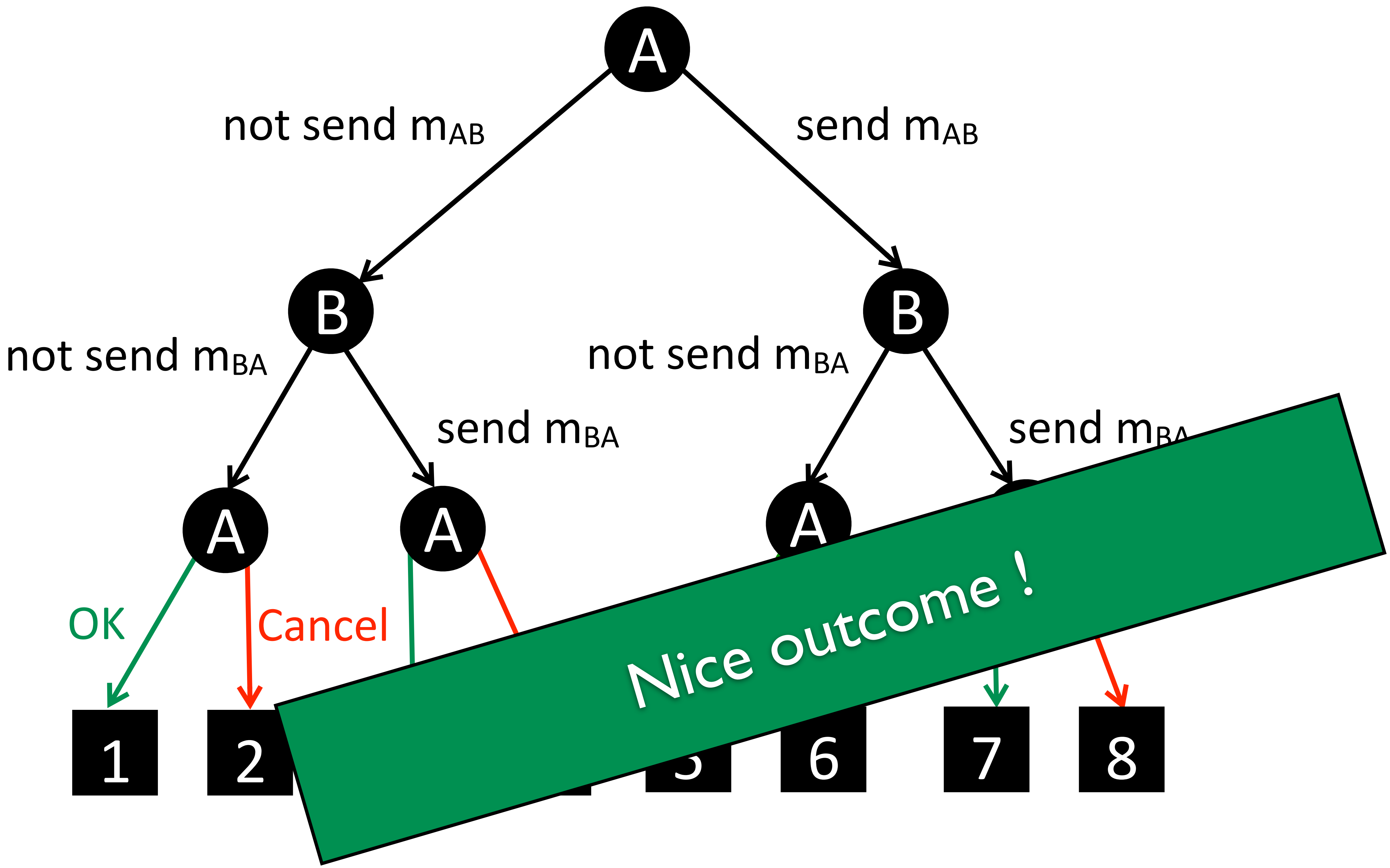
A preferences: 3>7>1=2=4=6=8>5

B preferences: 5>7>1=2=4=6=8>3

Unique secure equilibrium:

send m_{AB}, send m_{BA}, OK

A simple example



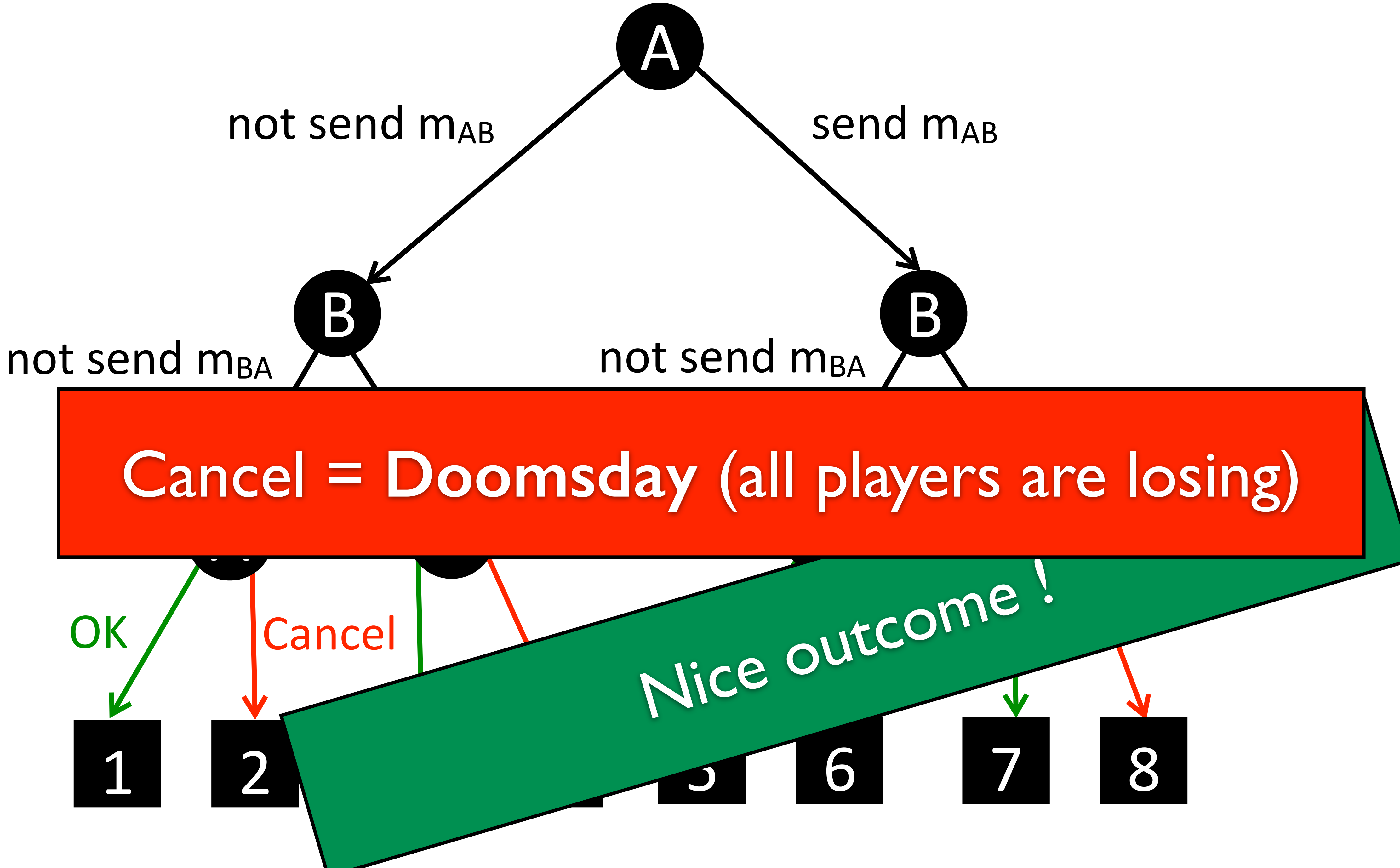
A preferences: 3>7>1=2=4=6=8>5

B preferences: 5>7>1=2=4=6=8>3

Unique secure equilibrium:

send m_{AB}, send m_{BA}, OK

A simple example



A preferences: 3>7>1=2=4=6=8>5

B preferences: 5>7>1=2=4=6=8>3

Unique secure equilibrium:

send m_{AB}, send m_{BA}, OK

Doomsday threatening equilibria

A strategy profile $(St_1,St_2,...,St_n)$ is a **doomsday threatening equilibrium (DE)** if:

1. Outcome$(St_1,St_2,...,St_n)$ is "**winning**" for **all** players

2. For all player i, Outcome(St_i) is such that:
either player i wins **or** all players lose (doomsday)
i.e. St_i is **good for retaliation**

Glimpse on the results

	Safety	Reach	Büchi	coBüchi	Parity	LTL
Perfect info	PSpace-C	PTime-C	PTime-C	PTime-C	in PSpace NP-Hard coNP-Hard	2ExpTimeC
Imperfect info	ExpTime-C	ExpTime-C	ExpTime-C	ExpTime-C	ExpTime-C	2ExpTimeC

Assume Admissible Synthesis

[LICS2014-CONCUR2015]

Models of rationality

- Nash equilibria: no player has an incentive to deviate.
- **Elimination of dominated strategies**: players eliminate "bad" strategies.

Dominated strategies

- Rational players **avoid** to play **bad** strategies.
- a strategy σ is **dominated** by σ' if
  - **for all** profiles of the other players,
    if σ wins then σ' wins,
  - and **for some** profile of the other players,
    σ loses while σ' wins.
- Idea: we may assume that Players **only** play **admissible strategies**.

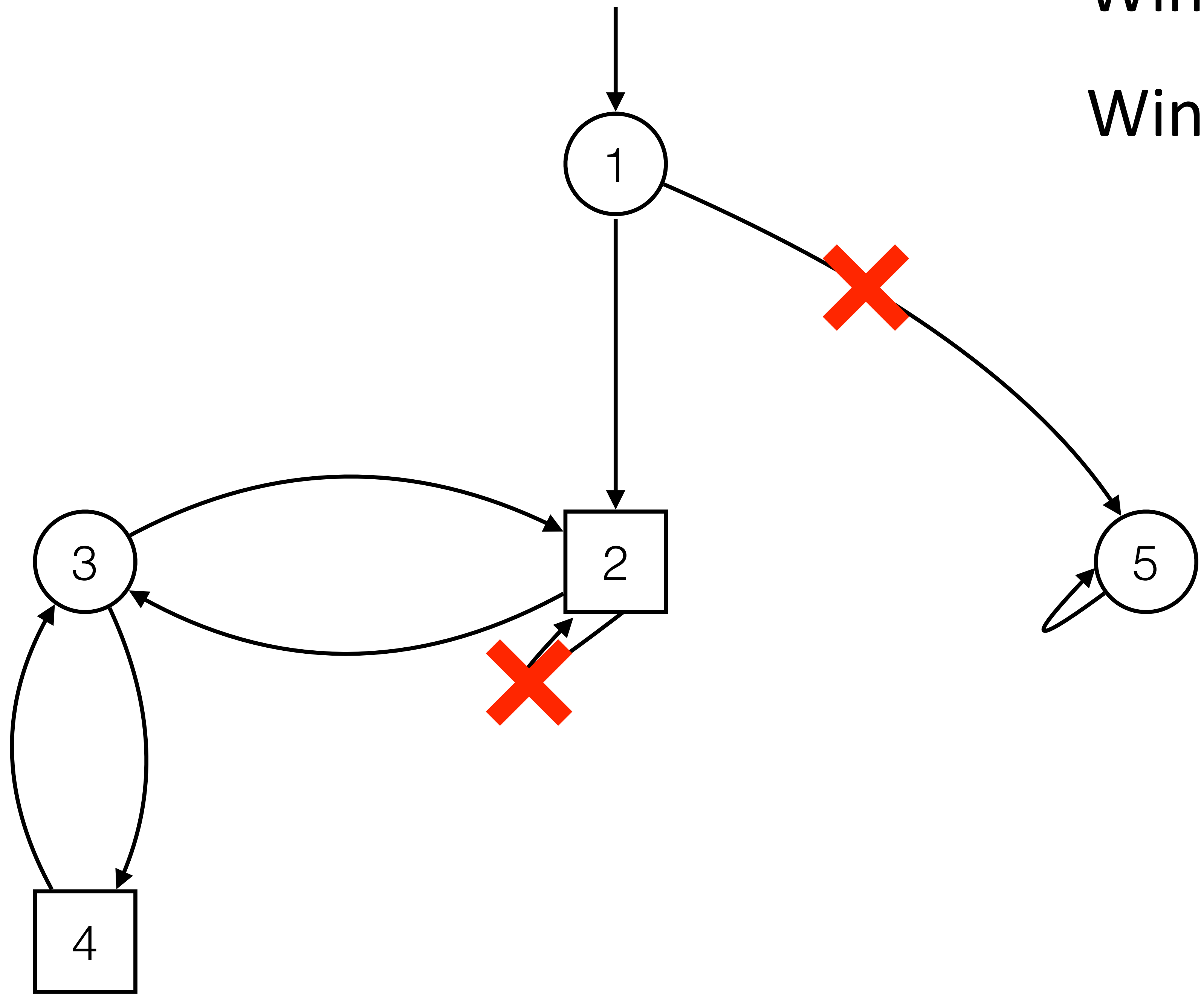
Assume Admissible Synthesis

- Let G be a **n**-player game with objectives $W_1, W_2, \ldots, W_n$
- $W_1, W_2, \ldots, W_n$ can be **achieved** by **assume-admissible strategies** if there exists a strategy profile $(\sigma_1,\sigma_2, \ldots, \sigma_n)$ such that:
  - for all i, $1 \leq i \leq n$, σ_i is **admissible** for W_i
  - for all i, $1 \leq i \leq n$, for all σ_{-i} that is **admissible** for W_{-i}, $Out(\sigma_i,\sigma_{-i}) \models W_i$,

    i.e. each σ_i is **winning** against **all admissible** strategies of the other players
- Such a profile is called **Assume-Admissible (AA)**

$Win_1 = \{ \rho \mid \rho \models \Box \Diamond 4 \}$

$Win_2 = \{ \rho \mid \rho \models \Box \Diamond 3 \}$

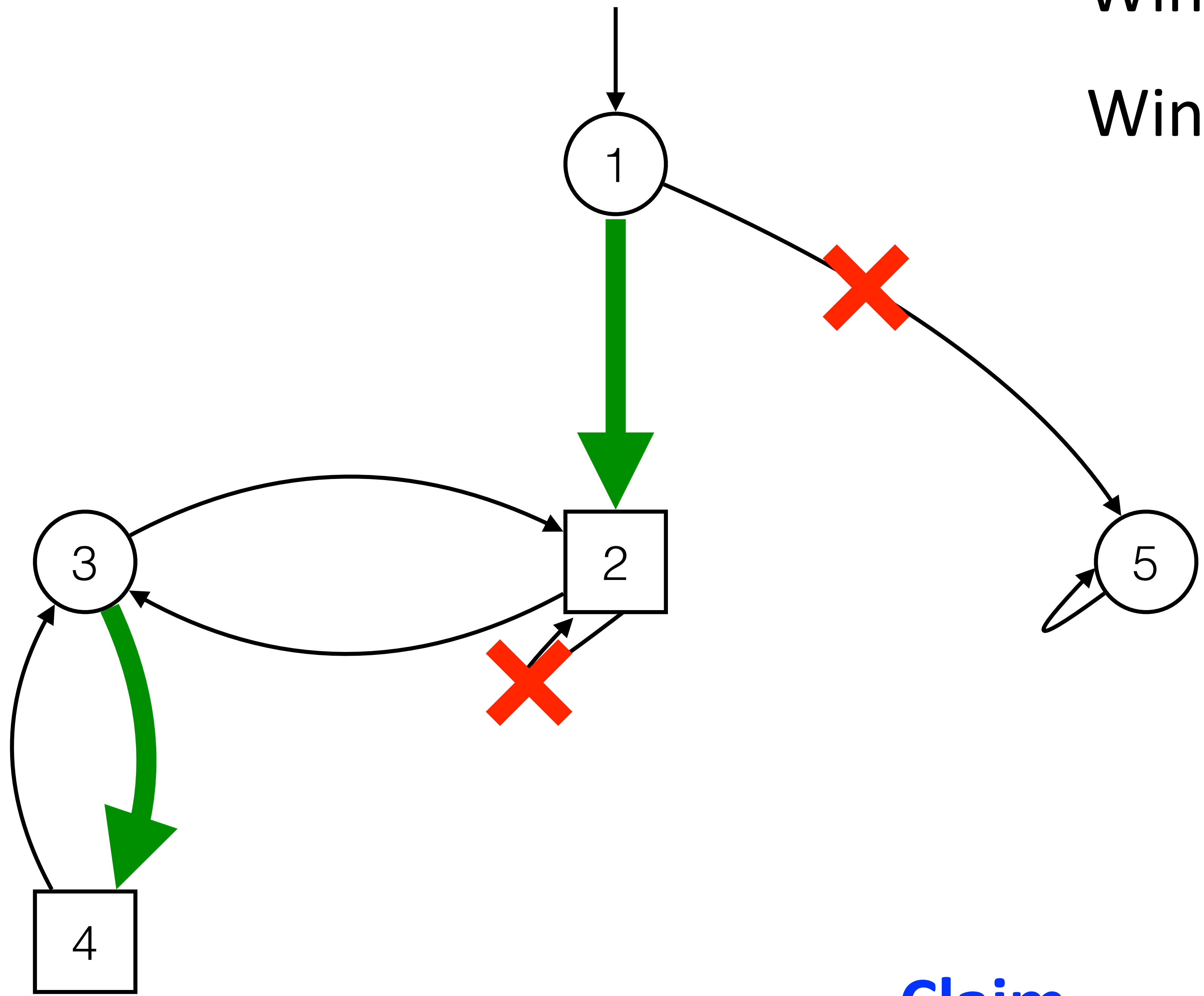


Some facts:

- 1-5 is not admissible for Player 1
- $\Box$(2-2) is not admissible for Player 2

$Win_1 = \{ \rho \mid \rho \models \Box \Diamond 4 \}$

$Win_2 = \{ \rho \mid \rho \models \Box \Diamond 3 \}$

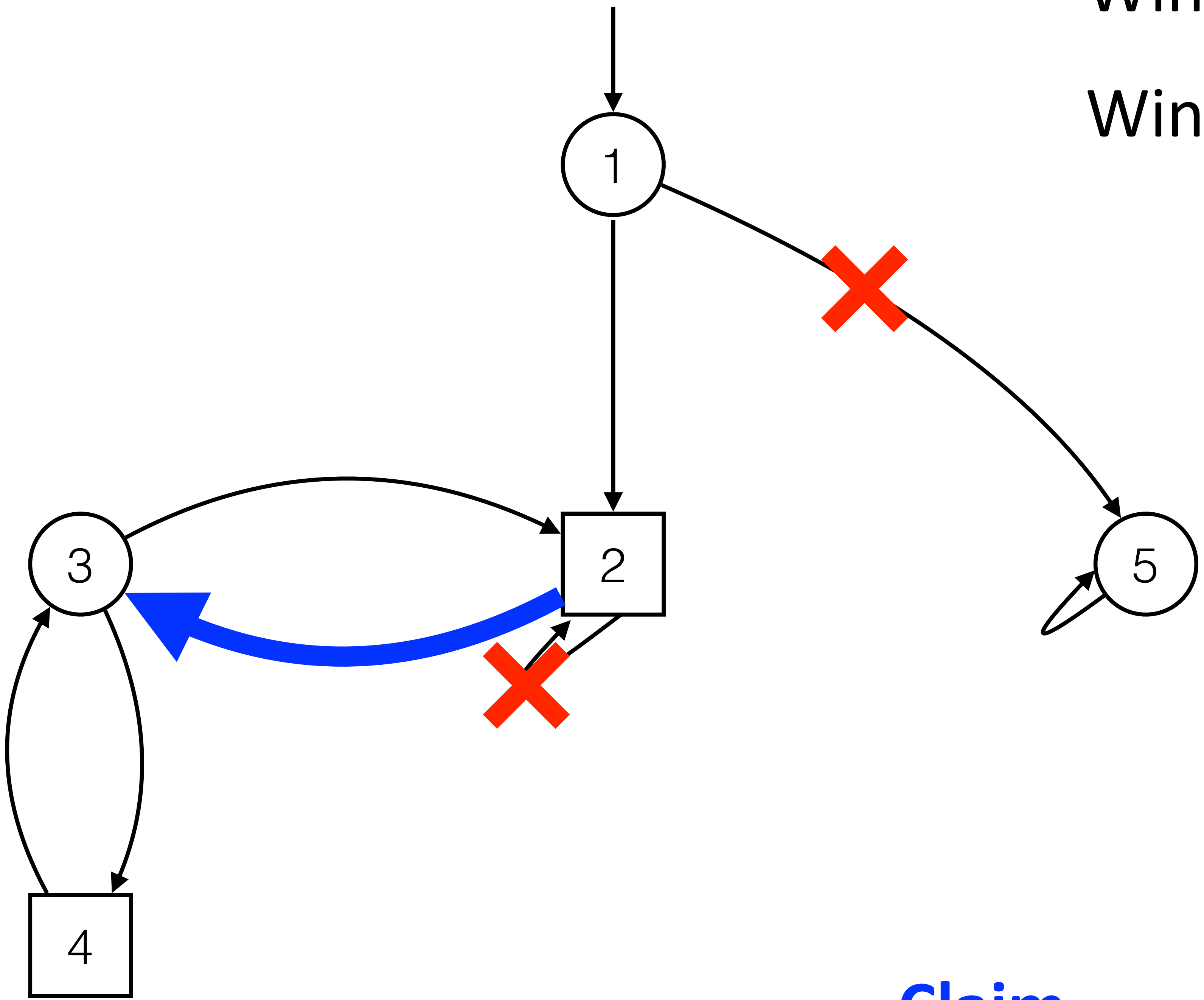


Claim

→ is admissible for Player 1 and winning against all admissible strategies of Player 2

$\text{Win}_1 = \{ \rho \mid \rho \models \Box \Diamond 4 \}$

$\text{Win}_2 = \{ \rho \mid \rho \models \Box \Diamond 3 \}$



Claim

→ is admissible for Player 2 and winning against all admissible strategies of Player 1

Assume Admissible Synthesis

- **Theorem**: for all **AA**-profiles σ_P, $Out(\sigma_P) \models \bigwedge_{i=1,..,n} W_i$
- **Theorem**: the set of **AA**-profiles is **rectangular**.
  → No need for synchronization!
- **Theorem**: **AA-synthesis** is PSPACE-complete for safety, reachability and Müller objectives and PTIME-complete for Büchi objectives.

Regret minimization in reactive synthesis

[CONCUR 2015]

Regret minimization

- **Regret in decision theory**: decision in presence of **uncertainty** → minimize the regret associated with the decision once the uncertainty is **resolved**.

- **… in computer science**: the performances of an **online** algorithm facing **uncertainty** are compared to the performances of an **offline** algorithm. An online algorithm is **better** if its performances are **closer** to the performances of an **optimal offline solution**.

Regret minimization

- **Regret in decision theory**: decision in presence of **uncertainty** → minimize the regret associated with the decision once the uncertainty is **resolved**.

- **… in computer science**: the performances of an **online** algorithm facing **uncertainty** are compared … performances of an **offline** … algorithm is **bette**… to the performances of a… **solution**.

The offline solution serves as a **yardstick**

Regret minimization

- **Regret minimization in game theory**: when a player chooses his strategy, and does not know how the other player(s) will play, he may choose to **minimize his regret** instead of aiming for worst-case optimality.

  The regret of a strategy is equal to difference of the performance of this strategy compared to performance of the **best response** to the strategy chosen by the other player(s).

Regret minimization

- **Regret minimization in game theory**: when a player chooses his strategy, and does not know how the other player(s) will play, he may choose to **minimize his regret** instead of aiming for worst-case optimality.

  The regret of a strategy is equal to difference of the performance of this strategy compared to p... of the **best response** to the st... player(s).

Best-responses serve as **yardsticks**

Regret - definition

- Regret associated to strategy σ (for Player 1)

  $\mathrm{Reg}(\sigma)=\sup_{\pi\in St2} \sup_{\sigma'\in St1} \mathbf{Val}(\sigma',\pi)-\mathbf{Val}(\sigma,\pi)$

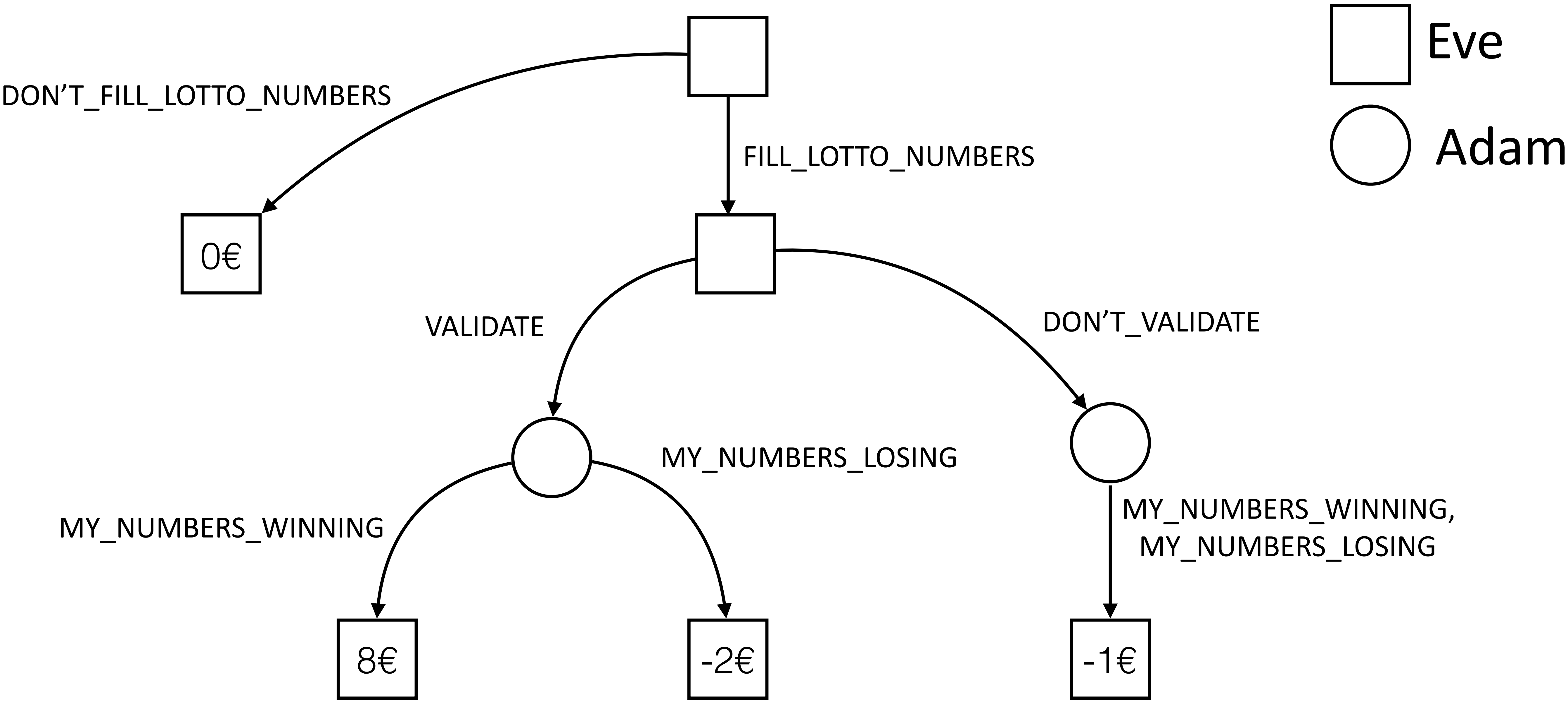
best-response as yardsticks

- Regret minimization

  $\mathrm{Reg}=\inf_{\sigma\in St1} \mathrm{Reg}(\sigma)$

- Three variants:
  1) St_2=All possible strategies
  2) St_2=All memoryless strategies
  3) St_2=All word strategies

Worst-case value for Eve=0



4 strategies for Eve

DON’T_FILL_NUMBERS-VALIDATE: **regret=8** because MY_NUMBERS_WINNING gain=0 and could be 8

DON’T_FILL_NUMBERS-DON’T_VALIDATE: **regret=8** because MY_NUMBERS_WINNING gain=0 and could be 8

→ FILL_NUMBERS-VALIDATE: **regret=2** because MY_NUMBERS_LOSING gain=-2 and could be 0

FILL_NUMBERS-DON’T VALIDATE: **regret=9** because MY_NUMBERS_WINNING gain=-1 and could be 8

Motivations

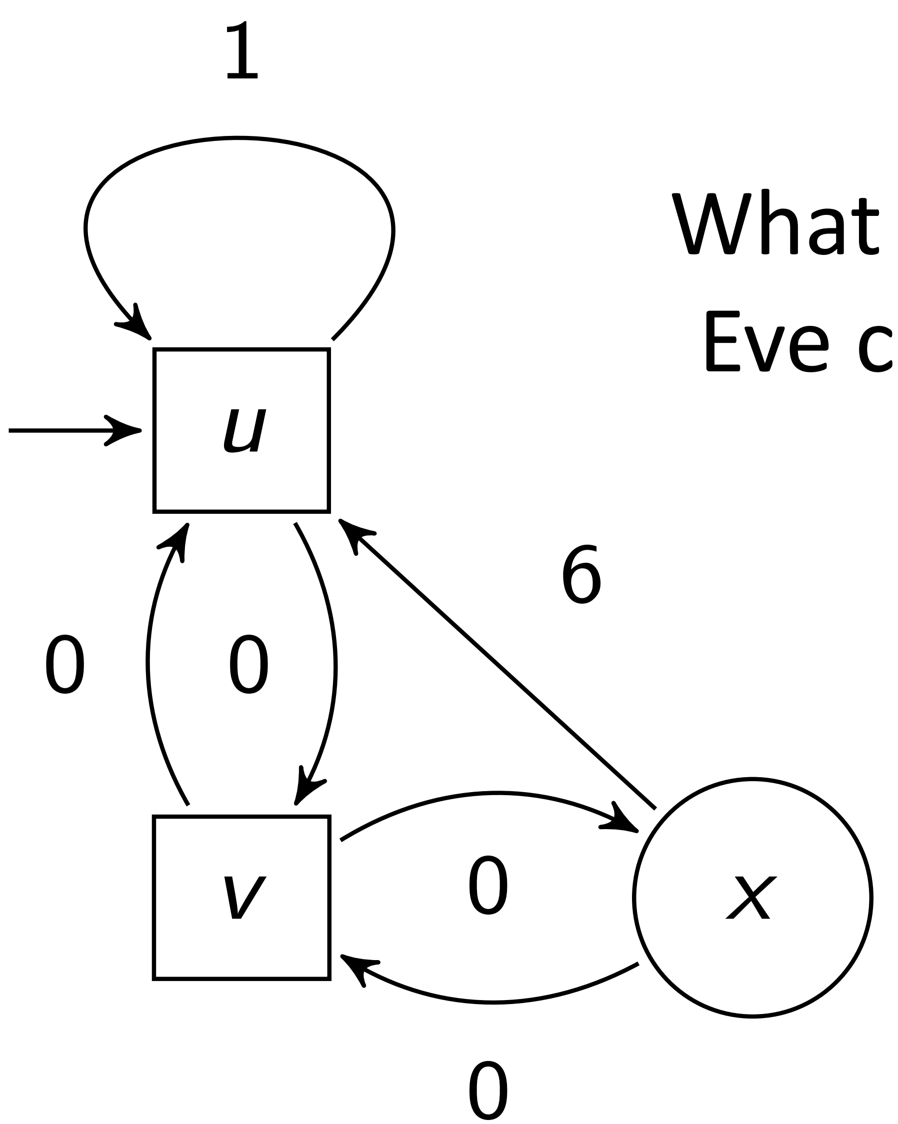
The other player is playing **memoryless**.

Useful for building a system which should accommodate **several usage scenarios** (=memoryless strategies).

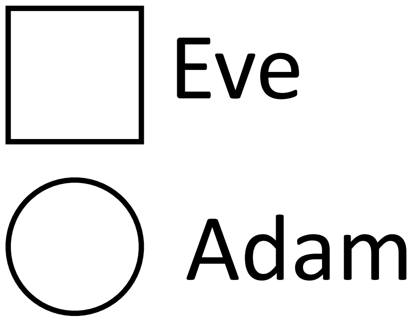
Regret=difference between the answer of the system to the usage scenario versus the best response possible.

If you want to keep users happy: **minimize regret** !

The other player is memoryless



What is the minimal regret that Eve can achieve here for MP ?



Motivations

- The other player is playing **word strategies**:

  Env. produces disturbances independent of Sys behavior
  =sequence of events
  =word on the alphabet of the disturbances
  ≠antagonistic process

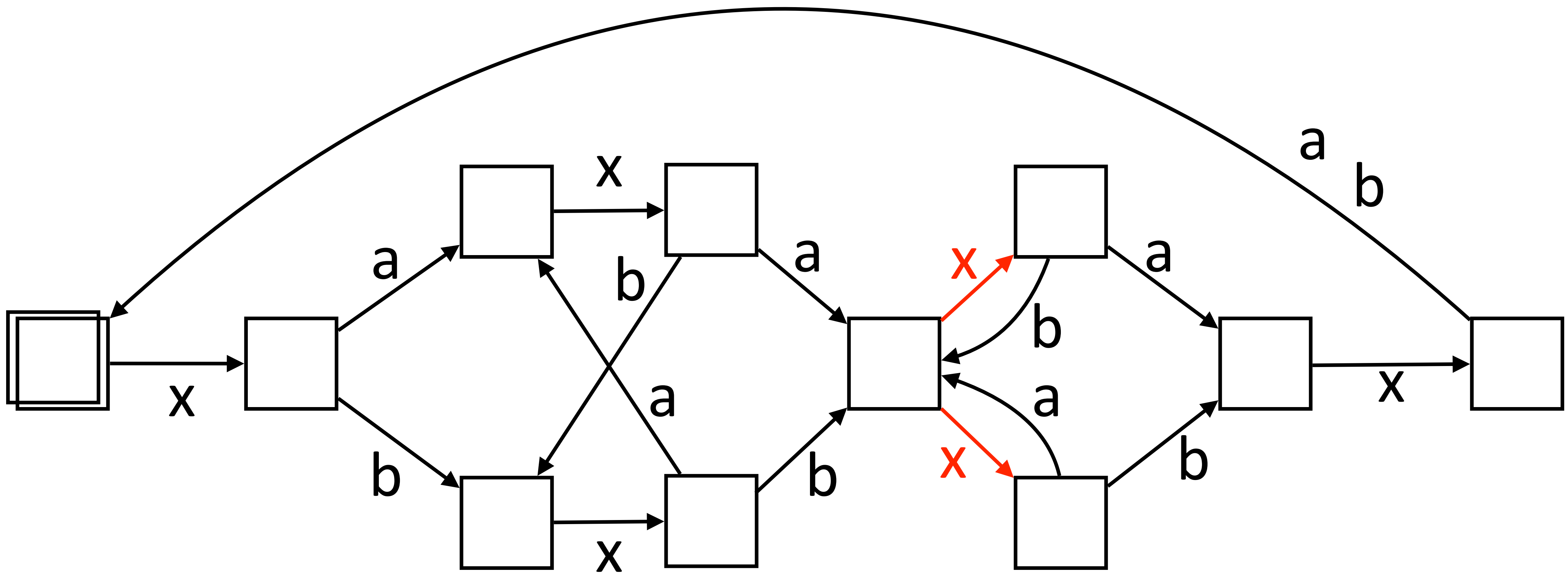
- minimizing regret produces a system that tries to produce a correct behavior when it is possible

Good-for-games

- A nondeterministic automaton A is **good-for-games** if the **nondeterminism** in A can be resolved **online**
- … no need to make A deterministic before game solving

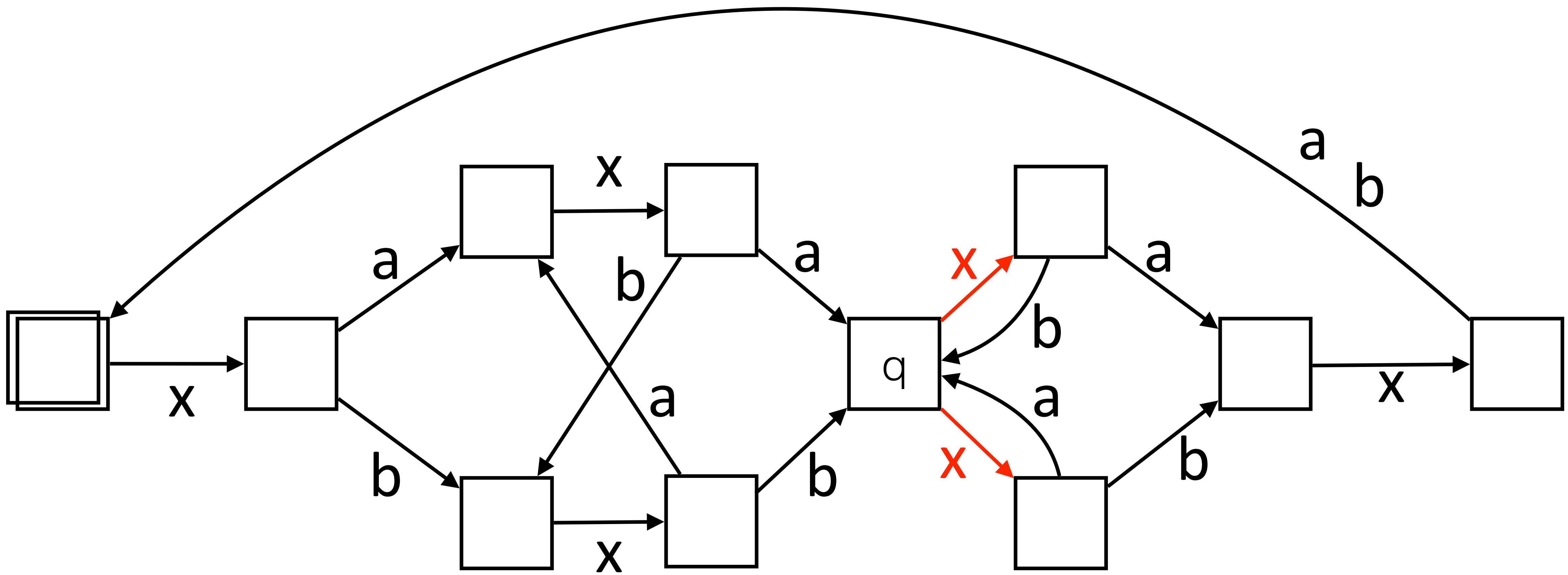
Good-for-games

Example by
Udi Boker



L(A)=words $(x\{a,b\})^\omega$ that contain **infinitely many** xaxa or xbxb

Good-for-games



An automaton is **good-for-game** if
its nondeterminism can be resolved **online**.

strategy: in q, chooses **up** if enter q by reading a
chooses **down** if enter q by reading b

Good-for-games

- A is **good-for-games** if

  $\exists\sigma:\Sigma^*\rightarrow Q: \forall w\in\Sigma^\omega:$

  if $w\in L(A)$ **then** $\sigma^*(w)$ is an accepting run

  accepting run=yardstick

  σ=**online** resolution of nondeterminism

- Relation with **regretless** strategies:

  $\exists\sigma:\Sigma^*\rightarrow Q: \forall w\in\Sigma^\omega: \forall\sigma':\Sigma^*\rightarrow Q: \mathbf{Val}(\sigma',w)-\mathbf{Val}(\sigma,w)\leq 0$

  Eve, against all word strategies of Adam, can resolve nondeterminism online without regret

- **Regret minimization** offers a **quantitative generalization** of good-for-games (useful in the context of **weighted automata**)

  $\exists\sigma:\Sigma^*\rightarrow Q: \forall w\in\Sigma^\omega: \forall\sigma':\Sigma^*\rightarrow Q: \mathbf{Val}(\sigma',w)-\mathbf{Val}(\sigma,w)$

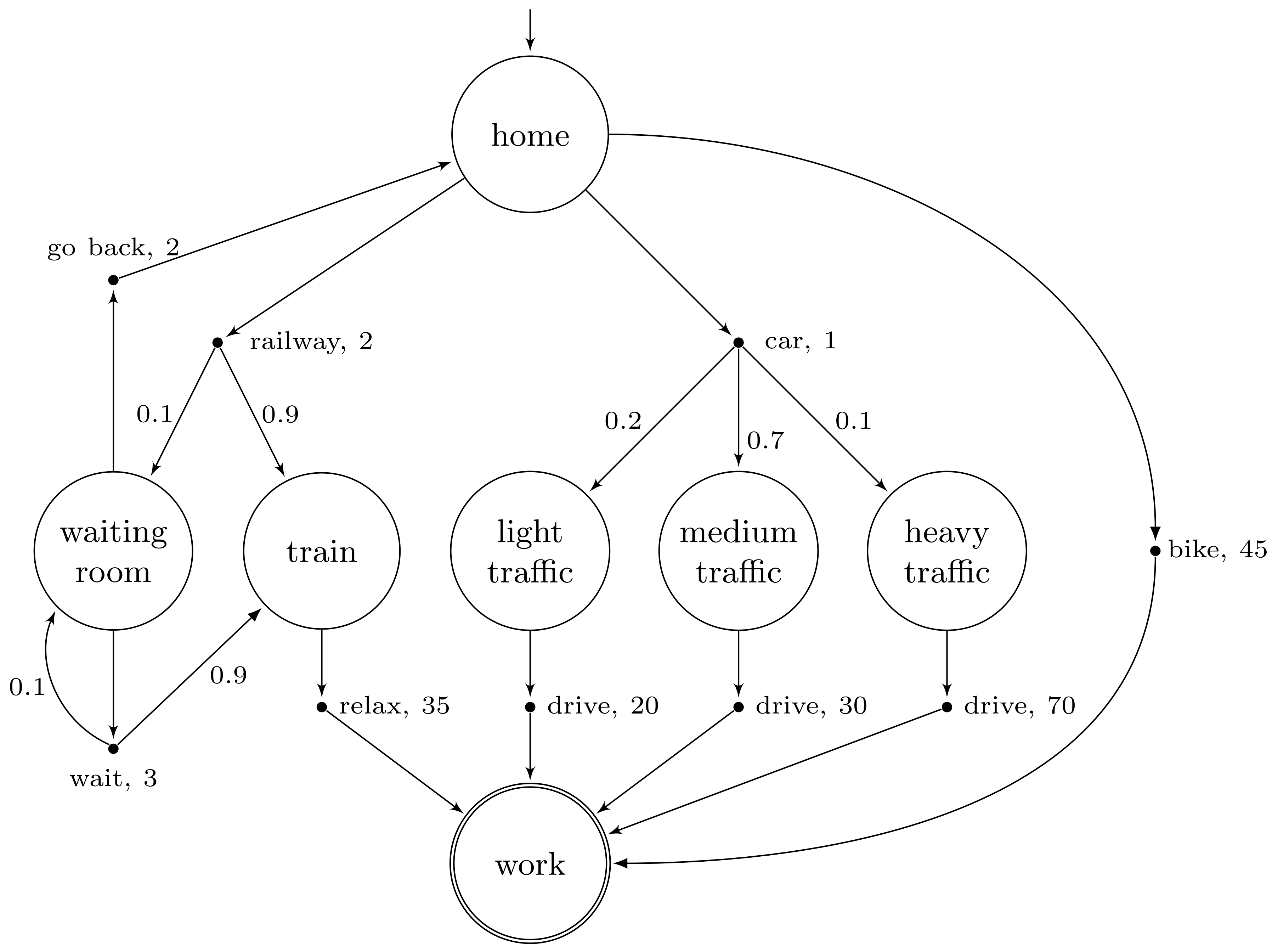
Glimpse on results

	sup	inf	lim sup	lim inf	MP
Any	PTIME-C	PTIME-C	PTIME-C	PTIME-C	NP∩coNP MP-hard
Memoryless	in PSpace NP-hard	in PSpace NP-hard	in PSpace NP-hard	PSpace-C	PSpace-C
Word	ExpTime-C	ExpTime-C	ExpTime-C	ExpTime-C	Undec

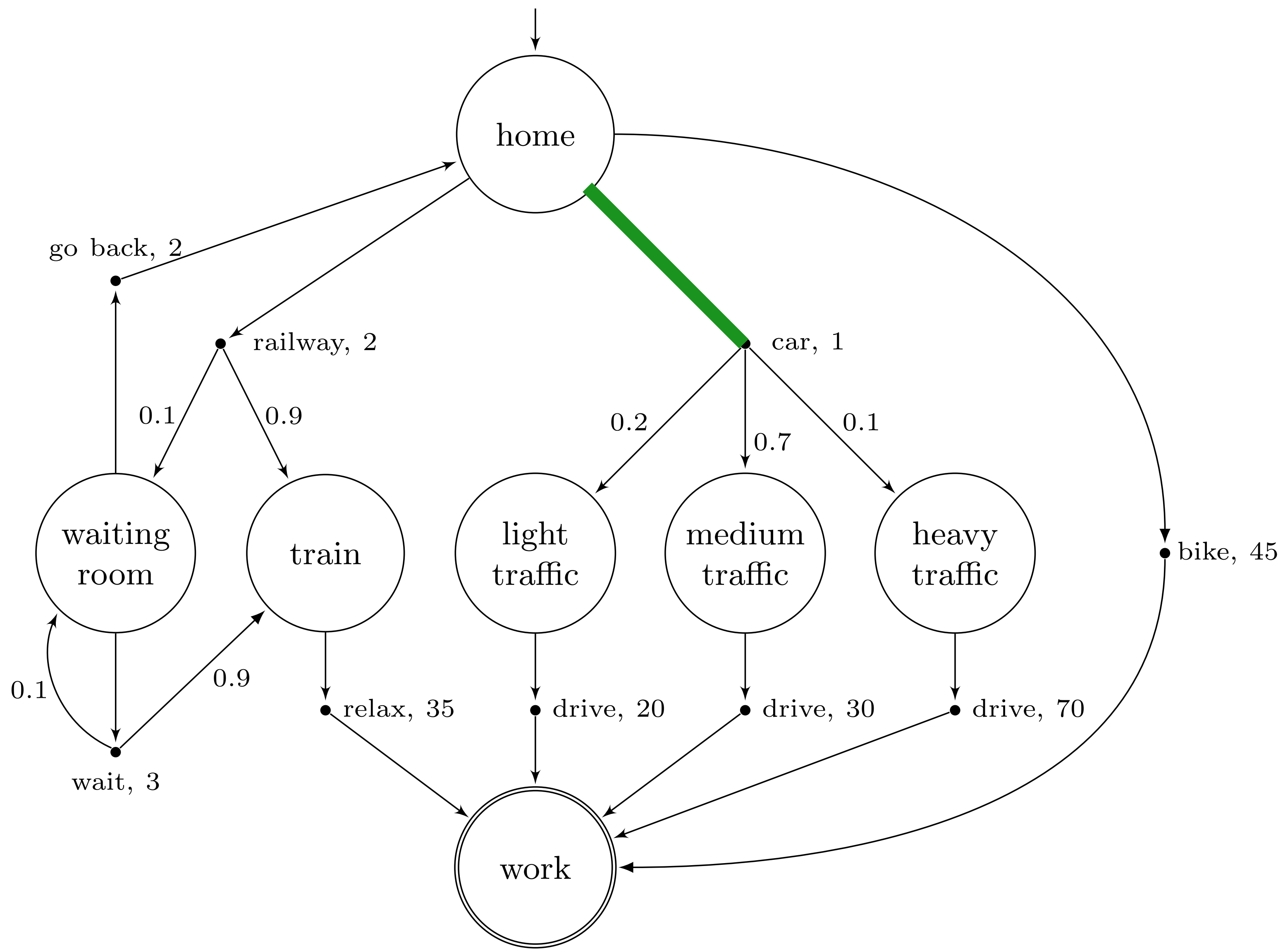
Beyond worst-case

[STACS14-LICS15-VMCAI15]

MDP


home
go back, 2
railway, 2
car, 1
0.1
0.9
0.2
0.7
0.1
waiting room
train
light traffic
medium traffic
heavy traffic
bike, 45
0.1
0.9
relax, 35
drive, 20
drive, 30
drive, 70
wait, 3
work

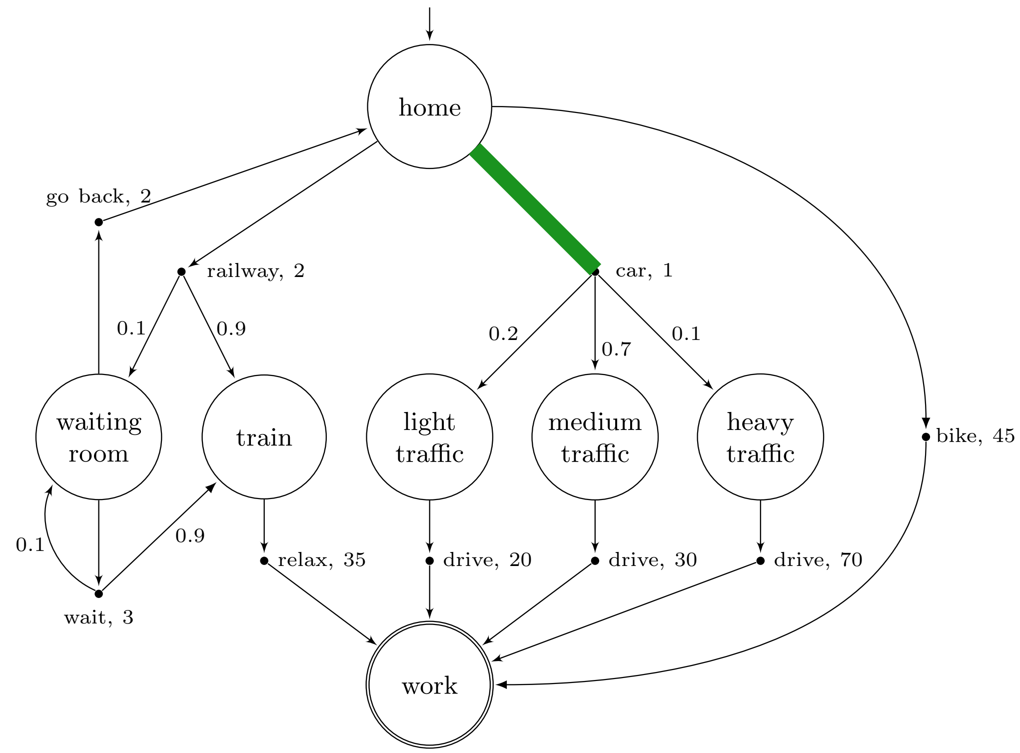
Minimize expected time



Take the **car**:

1 + 0.2x20 + 0.7x30 + 0.1x70 = **33** minutes

Min. Prob. of Outliers

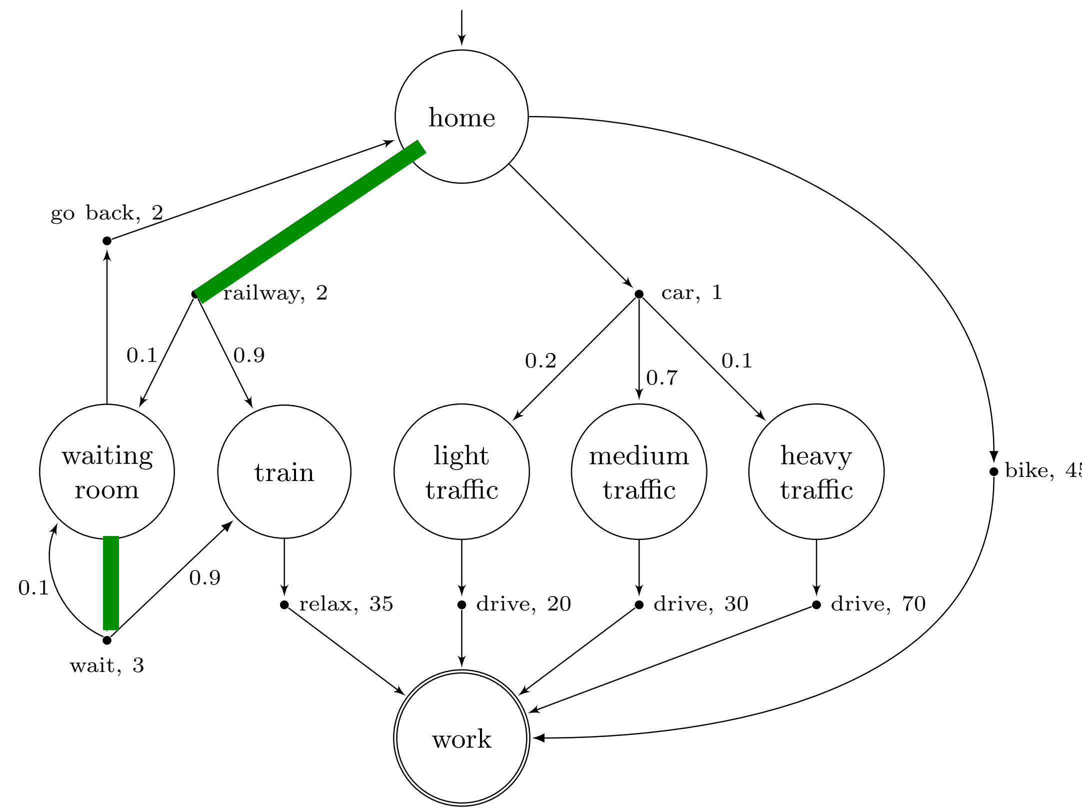


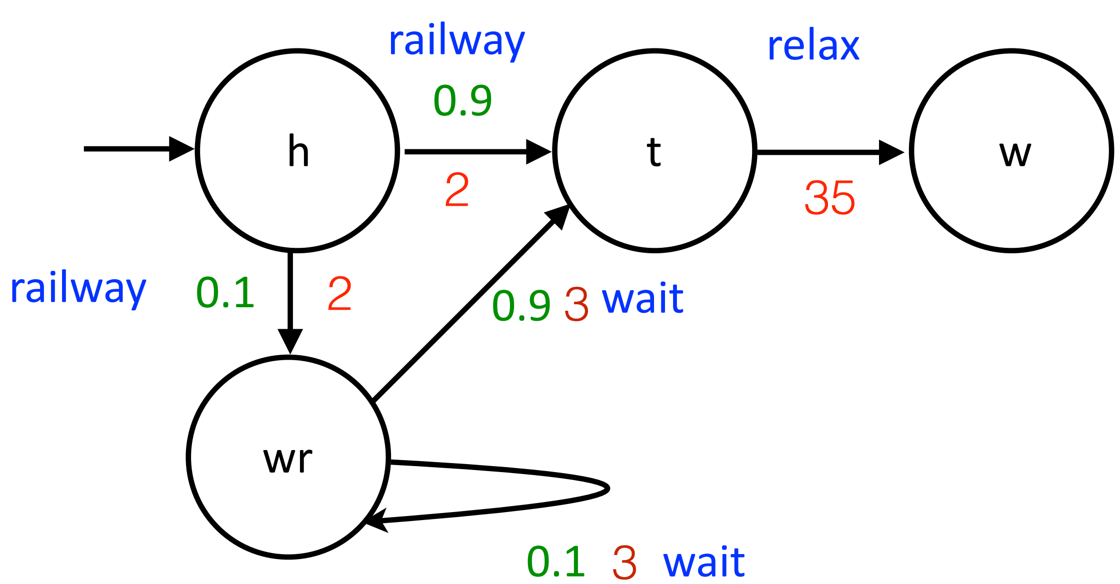
With car, the prob. of **long** runs (e.g. 71 minutes) is not negligible (10%).

What if the employee is **risk-averse** ?

Back to the example

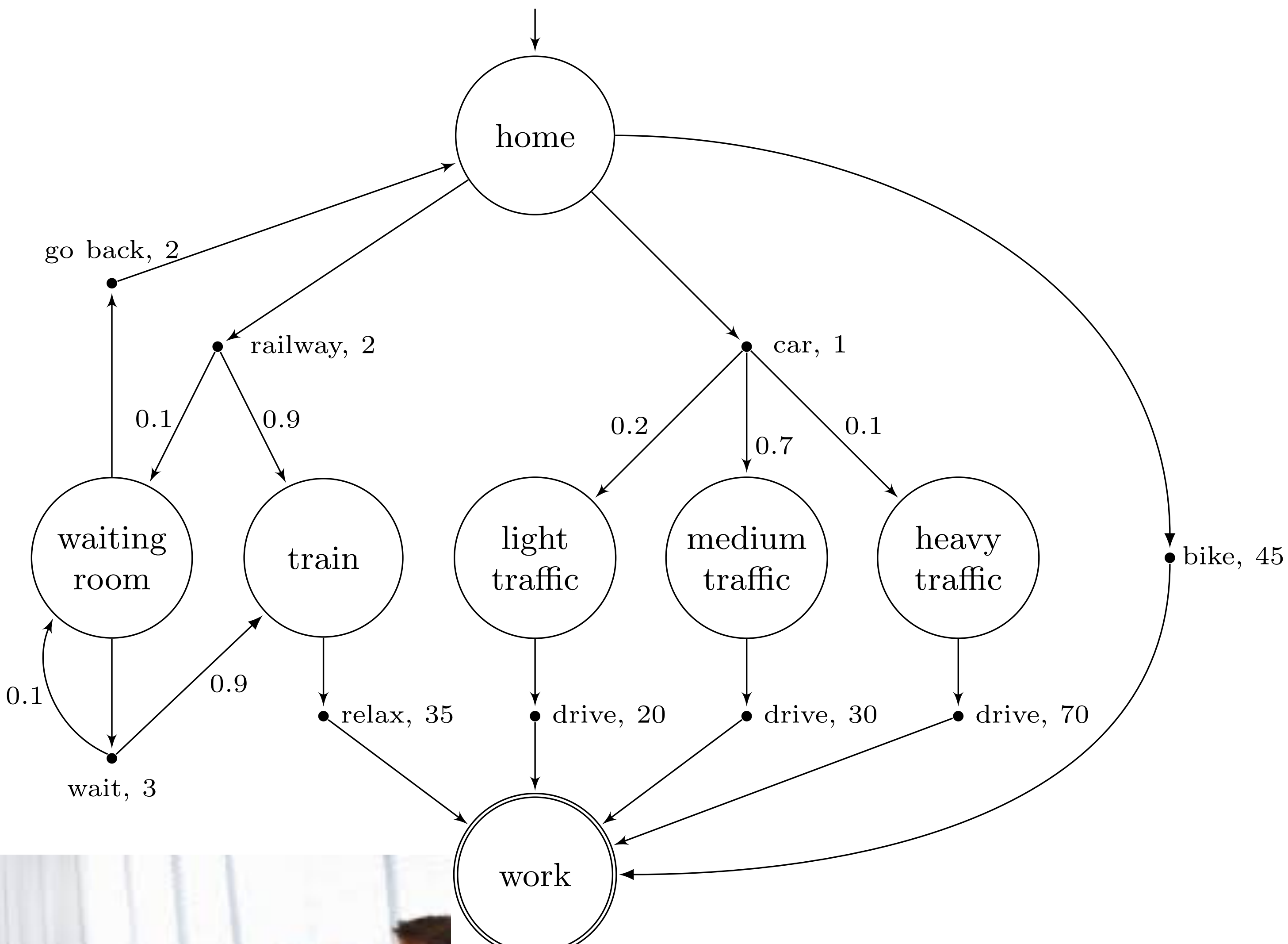
Take the **train** !





$\mathbb{P}(D, \sigma)(TS^T \leq 40) \geq 0.9+0.1\text{x}0.9$
$=0.99$

What if you **ought** to be at work within one hour ?



- **Train** option leaves a small probability of **not** reaching work within 1 hour (i.e. 1%)
- What if this is **unacceptable** ?
- Take your **bike** !
- **But can we do better ?** i.e. be sure to be at work within one hour with a better expectation than 45 min. ?

Worst-case guarantees with good expectation

Given a single-dimensional weighted MDP $D=(S,s_{init},A,\delta,w)$, a set of target states $T \subseteq S$, and two values v_1 and $v_2 \in \mathbb{N}$, decide if there exists a (unique) **strategy σ** such that:

1. [**Worst-case**] **for all** $\boldsymbol{\rho \in Out(D,\sigma): TS^T(\rho) \leq v_1}$

2. [**Expectation**] $\boldsymbol{\mathbb{E}(D, \sigma)(TS^T) \leq v_2}$

It is a **natural problem**:

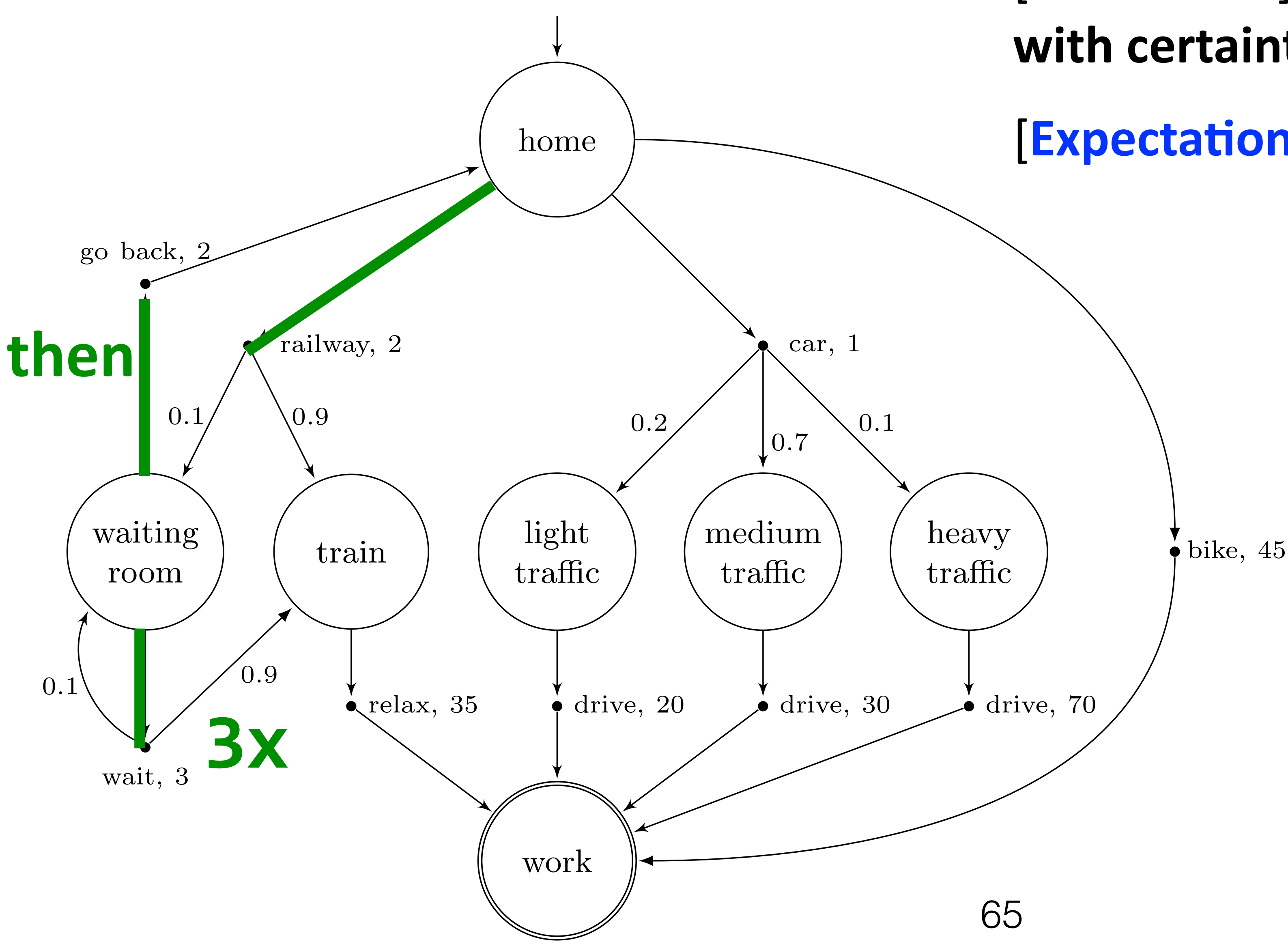
avoid unacceptable outliers at all cost!

Two views

- A **game + an expected behavior** of the env./adversary given as a stochastic Moore machine: you want a winning strategy (worst-case) that **behaves well/better** against the **expected behavior** of the env./adversary

- A **MDP** (expectation) **+** you want to **avoid outliers** at all cost (worst-case)

Back to the example



- Wait for the train
- After three delays, goes back home and bike

[**Worst-case**] **safe**: at work within 58 minutes **with certainty**

[**Expectation**] ≈ 37, 34… minutes (<45 —Bike)

Glimpse on results

	Worst-Case	Expectation	Beyond Worst-case
MP 1 dim	NP∩coNP	PTime-c	NP∩coNP
MP n dim	NP-c	PTime-c	NP-c
Shortest Path	PTime-c	PTime-c	in ExpTime PP-h

Take home message

Non-zero sum games are useful for reactive synthesis !

- Secure equilibria/**Doomsday** equilibria
  (refinements of Nash equilibria)

- **Admissible** strategies

- **Regret** minimization
  (a.o. extensions of good-for-games automata)

- Play good **both** for **worst-case** and **expectation**