An introduction to game theory

(with applications to computer science and embedded systems design)

ULB

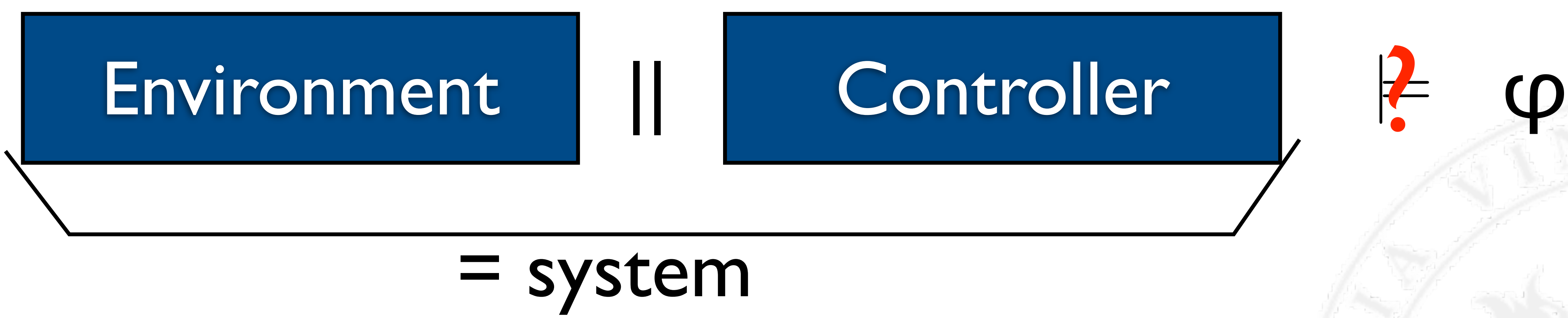
UNIVERSITÉ LIBRE DE BRUXELLES

Motivation

- One of the most important constraint in Controller design is **correctness**
- To ensure correctness, a first approach consists in:
  - Devising a (**model** of the) controller
  - Using a **verification tool** to prove that the controller is correct

Verification

- Verification problem:
  - *Given a* ***model*** *of a system made up of and* ***environment*** *and a* ***controller****, we want to prove that the system respects a given* ***property***



Synthesis

- Instead of this error-prone, trial-and-error process, we would like to perform **synthesis** of **correct controllers**
- Cfr. chemical synthesis:
  « ... *a purposeful execution of chemical reactions to obtain a product, or several products.* » (Wikipedia)

Synthesis

- Synthesis problem:
  - *Given a model of the environment, we want to* ***compute*** *a (model of) a controller that will enforce the property*
- The synthesised controller is **correct by construction**.

Environment || Contr?oller $\models \varphi$

Synthesis and games

- Seeing the synthesis problem through a **game metaphor** will be very useful
  - The **environment** is a player.
  - The **controller** is another player.
  - They **compete** against each other: the controller wants to enforce the property, while the environment wants to falsify the property.
  - A correct controller is one that implements a **strategy** that guarantees him to win **whatever the environment does**.
- But game theory has other applications !

Game - Intuition

- Consider the classical **4-in-a-row** game

Game - Intuition

- Players play by turn: they **alternate** one after the other.
  - This is a **turn-based** game

  There are finitely many positions: at most $3^{(6x7)} \simeq 1.1 \times 10^{20}$

Game - Intuition

- It has been shown that **the first player to play can always win the game.**
  - There exists a winning strategy for the first player.
  - This strategy can be finitely described as a function that assigns the optimal move to each positions. In theory this strategy can be implemented as an algorithm.

Game - Intuition

- Both players have a **complete view** on the the current state of the game, at all times
  - This is a game of **perfect information**.
- This is a **zero-sum game**: either player 1 win, or player 2 win, or there is a draw
  - It is not possible that both win or loose

Game - Intuition

- Other examples of games:
  - –**Poker**: Unlike 4-in-a-row, players do not see the complete state of the game (some cards are hidden).
    - This is a game of **imperfect information**
  - –**Penalty kick**: The kicker decides to kick either left or right of the goal. The goal keeper decides *simultaneously* to jump left of right.
    - The game is **concurrent**: players choose their move at the same time

Games - Intuition

- Games are best used to describe situations where different entities **compete** with each other:
  - **Synthesis problem**: controller vs. environment
  - **Network routing**: each ISP wants to minimise the amount of traffic on its network
  - **File sharing protocols**: with bittorrent, all participants want to get the whole file asap, while minimising bandwidth for upload.
    - These last two examples are non zero-sum games
  - **Real-time scheduling**: tasks vs. scheduler.

Game Theory

- Historically, game theory has been studied mainly by **economists**.
- During the last 10 years, game theory has started to pervade computer science
- We will be mainly interested in **algorithmic game theory**, with questions like:
  - Can we **compute** winning strategies ?
  - What is the **complexity** of computing those strategies ?
  - How can we **implement** those strategies ?

Strategic games

Strategic games

- Strategic games are a very simple form of games, where each player **chooses a strategy** (independently of the others), and gets a payoff that depends on **all the strategies**
  - The vector of strategies for all the players is called a **strategy profile**.
- The payoff for each player is given by a **matrix**

Prisoner's dilemma

- Two gangsters get arrested by the police. They are given two options during the trial:
  - Either they **confess (C)** their offense
  - Or they remain **silent (S)**
- They following matrix gives the **number of years in jail** they get in all cases:

		2	
		C	S
1	C	4 4	1 5
	S	5 1	2 2

Prisoner's dilemma

- Instead of a matrix that gives the **cost** of each strategy (less is better), we want a matrix that gives the **payoff** (more is better)

		2			
		C		S	
1	C	4	4	1	5
	S	5	1	2	2

Prisoner's dilemma

- Instead of a matrix that gives the **cost** of each strategy (less is better), we want a matrix that gives the **payoff** (more is better)

		2			
		C		S	
1	C	1	1	4	0
	S	0	4	3	3

Prisoner's dilemma

- There is no notion of winner/looser here: each prisoner wants to **minimise** his number of years in jail.

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Prisoner’s dilemma

- If both prisoners can coordinate, they better choose to remain **both silent**.
- But, knowing that 1 will remain silent, 2 might have an **incentive** to **deviate** and confess to get 1 year instead of 2 (and vice-versa)

		2			
		C		S	
1	C	1	1	4	0
	S	0	4	3	3

Prisoner’s dilemma

- If both prisoners can coordinate, they better choose to remain **both silent**.
- But, knowing that 1 will remain silent, 2 might have an **incentive** to **deviate** and confess to get 1 year instead of 2 (and vice-versa)

		2			
		C		S	
1	C	1	1	4	0
	S	0	4	3	3

Prisoner’s dilemma

- If both prisoners can coordinate, they better choose to remain **both silent**.
- But, knowing that 1 will remain silent, 2 might have an **incentive** to **deviate** and confess to get 1 year instead of 2 (and vice-versa)

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Prisoner's dilemma

- In this case, 1 better **confesses** too, to save one year in jail.

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Prisoner's dilemma

- In this case, 1 better **confesses** too, to save one year in jail.

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Prisoner's dilemma

- In this case, 1 better **confesses** too, to save one year in jail.

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Prisoner's dilemma

- In this case, 1 better **confesses** too, to save one year in jail.
- So, if both players are **selfish** and **rational** (as we assumed), the only stable solution is not the optimal one.

		2			
		C		S	
1	C	1	1	4	0
	S	0	4	3	3

Notations

- Let **s** denote a strategy profile. It is a vector of strategies for each player.
- We note s_i the strategy of player **i**
- We note s_{-i} the strategy profile of all players but **i**
- For a strategy profile **s**, we note **u(s)** the payoff of each player under the profile **s**
- We note $u_i(s)$ the payoff for **i**

Notations - examples

- Let $s = (\mathbf{C},\mathbf{S})$ -- player 1 chooses **C**, and player 2 chooses **S**
- $s_1 = C$
- $s_{-1} = (S)$
- $u(s) = u(s_1, s_{-1}) = (1,5)$
- $u_1(s) = 1$

		2: C	2: S
1	C	4 4	1 5
1	S	5 1	2 2

Battle of the sexes

- A couple wants to spend the evening together, but they must **pick an activity**
  - The boy prefers to stay home to watch the soccer game and have beer (**G**).
  - The girl wants to go out to the movies (**M**).
  - Doing different activities is worse than anything else for both.

		G			
		S		M	
B	S	4	2	1	1
	M	1	1	2	4

ULB
UNIVERSITÉ LIBRE DE BRUXELLES

Battle of the sexes

- In this case, it is easy to observe that there are **two stable situations**, which are **equivalent**

		G	
		S	M
B	S	4 2	1 1
	M	1 1	2 4

Penalty kicks

- The kicker and the goal keeper choose **simultaneously** between left or right

		G	
		L	R
K	L	1, -1	-1, 1
	R	-1, 1	1, -1

Penalty kicks

- The kicker and the goal keeper choose **simultaneously** between left or right
- Here, there is no **stable situation**

		G			
		L		R	
K	L	1	-1	-1	1
	R	-1	1	1	-1

How to play in such games ?

- What would be a notion of «best strategy» in such games ?
- First attempt: each player picks a strategy that **maximises his worst case payoff**.

How to play in such games ?

- What would be a notion of «best strategy» in such games ?
- First attempt: each player picks a strategy that **maximises his worst case outcome**.

Prisoner's dilemma:

		2			
		C		S	
1	C	1	1	4	0
	S	0	4	3	3

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

How to play in such games ?

- What would be a notion of «best strategy» in such games ?
- First attempt: each player picks a strategy that **maximises his worst case outcome**.

Worst-case payoff is 1

Worst-case payoff is 0

Prisoner's dilemma:

		2			
		C		S	
1	C	1	1	4	0
	S	0	4	3	3

How to play in such games ?

- What would be a notion of «best strategy» in such games ?
- First attempt: each player picks a strategy that **maximises his worst case outcome**.

The players choose a **stable** strategy profile that **does not minimise** the sum of the payoffs.

Prisoner's dilemma:

		2: C		2: S	
1	C	1	1	4	0
1	S	0	4	3	3

How to play in such games ?

- What would be a notion of «best strategy» in such games ?
- First attempt: each player picks a strategy that **maximises his worst case outcome**.

The players choose an **unstable** strategy profile that **does not maximise** the sum of the payoffs.

Battle of the sexes:

		G			
		S		M	
B	S	4	2	1	1
	M	1	1	2	4

Dominant strategy

- Observe that in the case of the prisoner's dilemma, each prisoner has a **dominant strategy**.
- A strategy is dominant if it gives a better payoff than all other strategies **no matter what the other player does**
  - –In some sense, dominant strategies allow one player to play independently of the other player

Dominant strategy

- **Definition**: A strategy profile **s** is **dominant** iff for all player i, for all strategy profile **t**:

$$u_i(s_i, t_{-i}) \geq u_i(t)$$

- In the prisoner's dilemma (S,S) is the unique dominant profile.
  - Check it !

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Dominant strategy

- **Definition**: A strategy profile **s** is **dominant** iff for all player i, for all strategy profile **t**:

$$u_i(s_i, t_{-i}) \geq u_i(t)$$

- In the prisoner's dilemma (S,S) is the unique dominant profile.
  - Check it !

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

≤

Dominant strategy

- **Definition**: A strategy profile **s** is **dominant** iff for all player i, for all strategy profile **t**:

$$u_i(s_i, t_{-i}) \geq u_i(t)$$

- In the prisoner's dilemma (S,S) is the unique dominant profile.
  - –Check it !

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Dominant strategy

- **Definition**: A strategy profile **s** is **dominant** iff for all … **t**:

- In the pr… unique d…
  - Check it !

In general, we cannot hope for the existence of unique dominant strategies

		2	
		C	S
1	C	1 1	4 0
	S	0 4	3 3

Dominant strategy

- Are there dominant strategies in the battle of the sexes, and the penalty ?

Battle of the sexes:

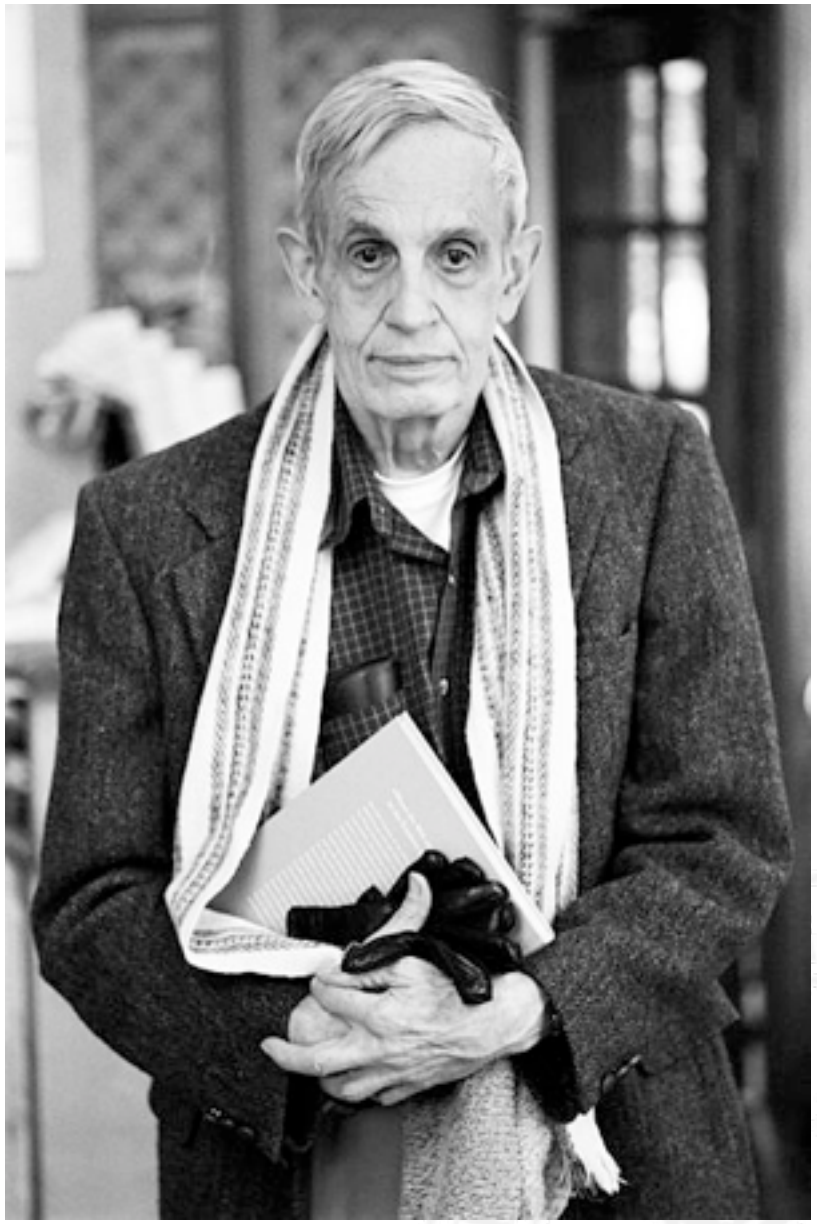
		G			
		S		M	
B	S	4	2	1	1
	M	1	1	2	4

Penalty:

		G			
		L		R	
K	L	1	-1	-1	1
	R	-1	1	1	-1

Nash equilibrium

- In order to capture the notion of «stability», one usually relies on the notion of **Nash equilibrium**, introduced by John F.Nash in 1951
- A strategy profile is an N.E. iff **no player has an incentive to deviate**



Nash equilibrium

- **Definition**: A strategy profile **s** is a **Nash equilibrium** iff for all player i, for all player i's strategy t_i:

$$u_i(s) \geq u_i(t_i, s_{-i})$$

- In the prisoner's dilemma, (C,C) is an N.E.

Prisoner's dilemma:

		2: C		2: S	
1	C	1	1	4	0
1	S	0	4	3	3

Nash equilibrium

- What are the N.E. in the two other games we have considered ?

Battle of the sexes:

		G: S	G: M
B	S	4, 2	1, 1
B	M	1, 1	2, 4

Penalty:

		G: L	G: R
K	L	1, -1	-1, 1
K	R	-1, 1	1, -1

Mixed strategies

- In some cases, there is no N.E. in games.
  - The penalty game is a typical example
- Intuitively, in those cases, one wants to play by flipping a coin to choose the strategy
- Such strategies are called **mixed strategy** (opposed to **pure strategies** seen so far)

Mixed strategies

- **Definition**: a **mixed strategy** for player i is a probability distribution over his possible choices.

Mixed strategies

- **Definition**: a **mixed strategy** for player i is a probability distribution over his possible choices.
- Example, for player **G**: s(L)=0.4, s(R)=0.6

		G			
		L		R	
K	L	1	-1	-1	1
	R	-1	1	1	-1

Mixed strategies

- **Notation**: let **A** be the matrix that associates, to each pair of choices of the players, the payoff of player 1. Let **B** be the symmetric for player 2.
- Example:

		G			
		L		R	
K	L	1	-1	-1	1
	R	-1	1	1	-1

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$$

Mixed strategies

- **Definition**: a best response to the mixed strategy **y** of player 2 is a mixed strategy **x** of player 1 s.t. $\mathbf{x}A\mathbf{y}^T$ is maximal

Mixed strategies

- **Definition**: a best response to the mixed strategy **y** of player 2 is a mixed strategy **x** of player 1 s.t. $\mathbf{xAy^T}$ is maximal
- **Example**: Let y=(0.4, 0.6)

$$\mathbf{A} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$\mathbf{Ay^T = (-0.2, 0.2)^T}$$

Mixed strategies

- **Definition**: a best response to the mixed strategy **y** of player 2 is a mixed strategy **x** of player 1 s.t. xAy^T is maximal
- **Example**: Let x=(0.4, 0.6)

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$Ay^T = (-0.2, 0.2)^T$

Now, let x=(0.9,0.1). Then, $xAy^T = -0.16$

Mixed strategies

- **Definition**: a best response to the mixed strategy **y** of player 2 is a mixed strategy **x** of player 1 s.t. $\mathbf{xAy^T}$ is maximal
- **Example**: Let x=(0.4, 0.6) $\quad \mathbf{A} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$

  $\mathbf{Ay^T = (-0.2, 0.2)^T}$

  Now, let x=(0.9,0.1). Then, $\mathbf{xAy^T = -0.16}$

  Consider x'=(0,1). Then, $\mathbf{x'Ay^T = 0.2}$
- Clearly, **x'** is a best response to **y**

Mixed strategies

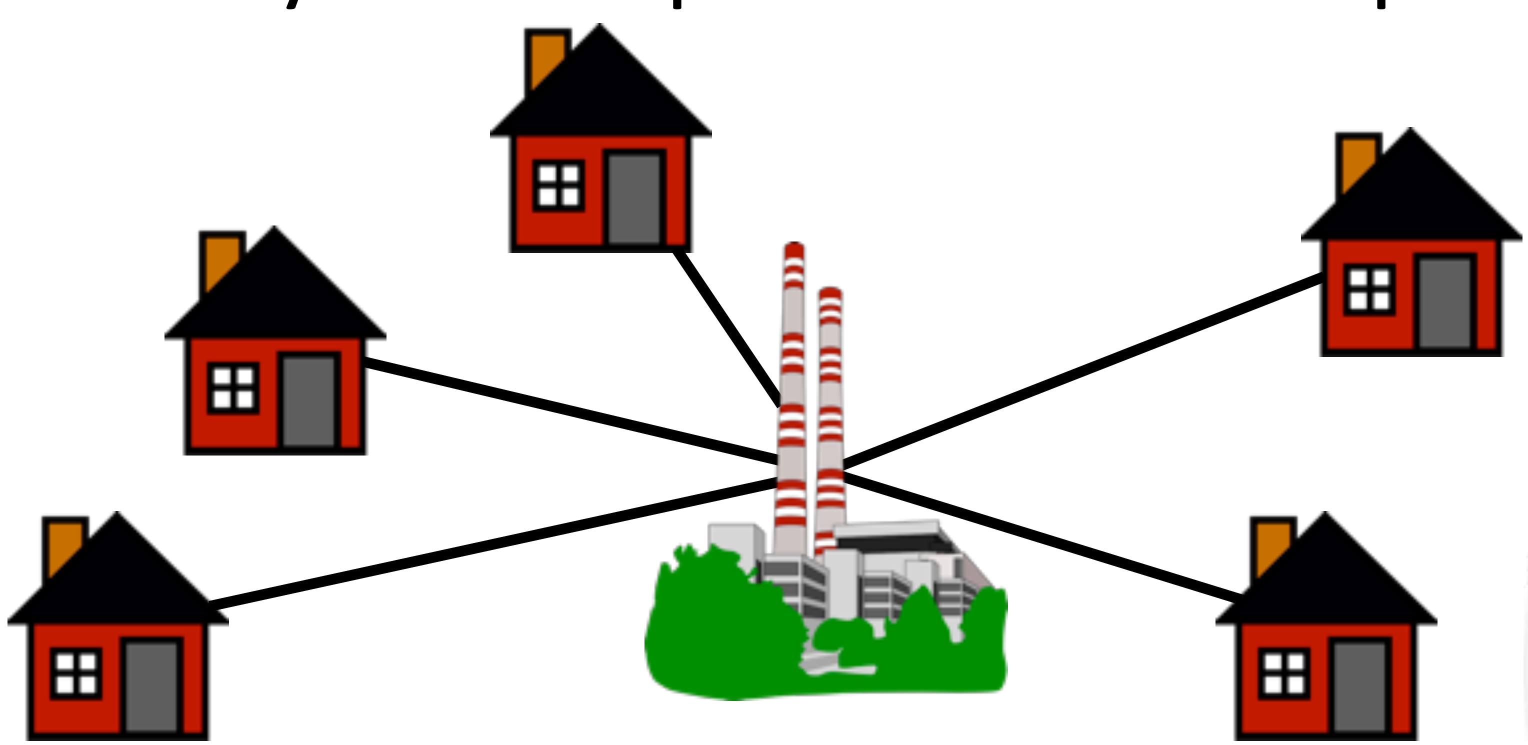
- **Definition**: a pair of mixed strategies (x,y) is a **Nash equilibrium** iff they are a best response to each other.

Mixed strategies

- **Definition**: a pair of mixed strategies (x,y) is a **Nash equilibrium** iff they are a best response to each other.
- **Example** with the penalty game: choosing (0.5, 0.5) for both players is a Nash equilibrium
  - Prove it !
  - Prove that the pure N.E. we had computed before in the prisoner's dilemma respect the def. of best response.

Nash equilibrium - applications

- **Microgrid management**
  - The system consists of N households connected to a single Distribution Manager (DM).
  - The system models a small neighbourhood.
  - Houses must collaborate to balance the electricity consumptions and avoid peak.



Nash equilibrium - applications

- **Microgrid management**
  - An algorithm has been proposed for the houses:
    - When a house generates a load, it evaluates its cost.
    - The cost depends on the current total load of the system.
    - If the cost is below a fixed threshold t, the house executes the load
    - Otherwise, it executes the load with some fixe probability

H. Hildmann and F. Saffre. Influence of variable supply and load flexibility on demand- side management. In Proc. 8th International Conference on the European Energy Market (EEM'11), pages 63–68, 2011.

Nash equilibrium - applications

- **Microgrid management**
  - Obviously, each house wants to **maximise** its value, defined as:

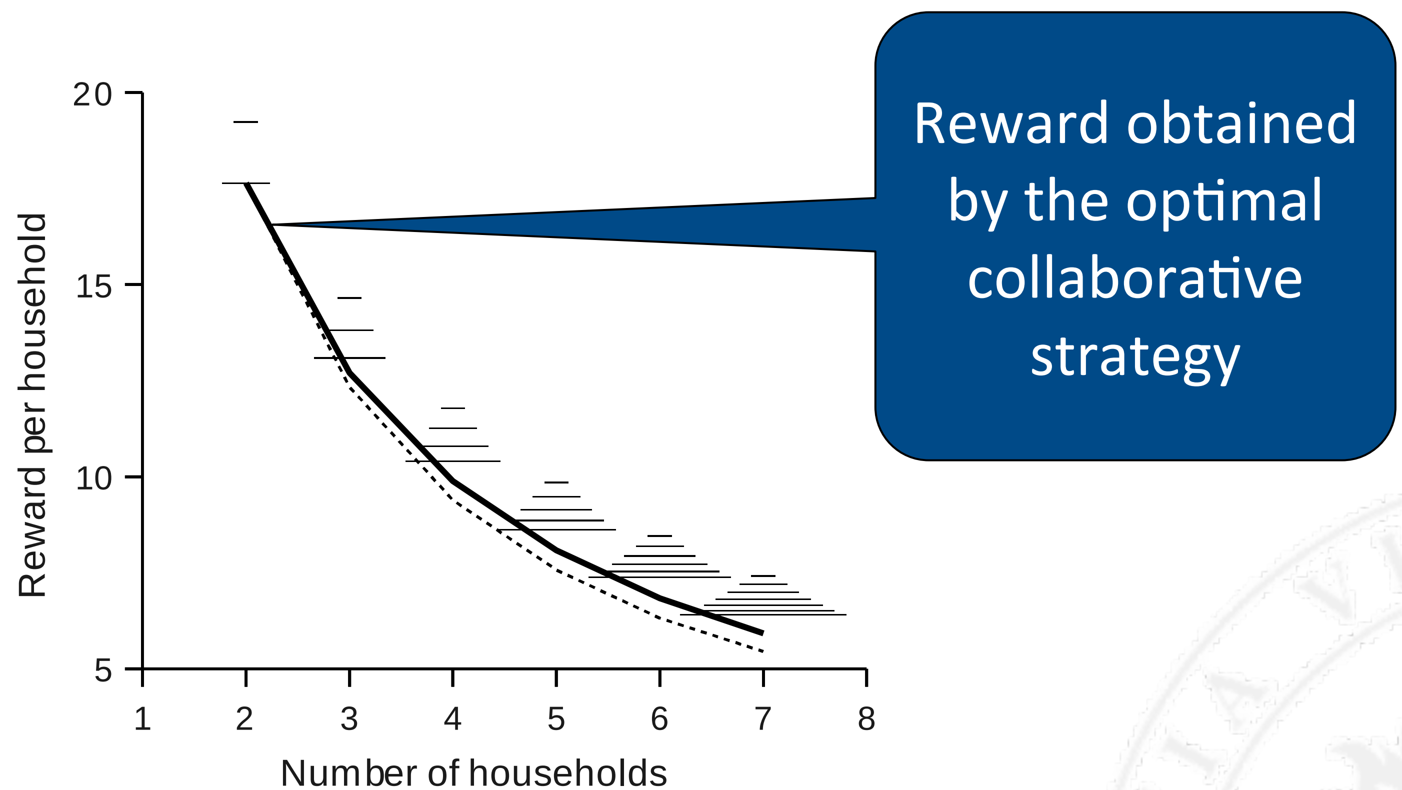
    V = loads executing / cost of execution

  - A desirable property of the system is that **no house has an incentive to deviate** from the agreed algorithm
    - In this case the possible strategies of the players are to deviate (or not) from the algorithm
    - The profile in which no house deviates should be an N.E.

H. Hildmann and F. Saffre. Influence of variable supply and load flexibility on demand- side management. In Proc. 8th International Conference on the European Energy Market (EEM'11), pages 63–68, 2011.

Nash equilibrium - applications

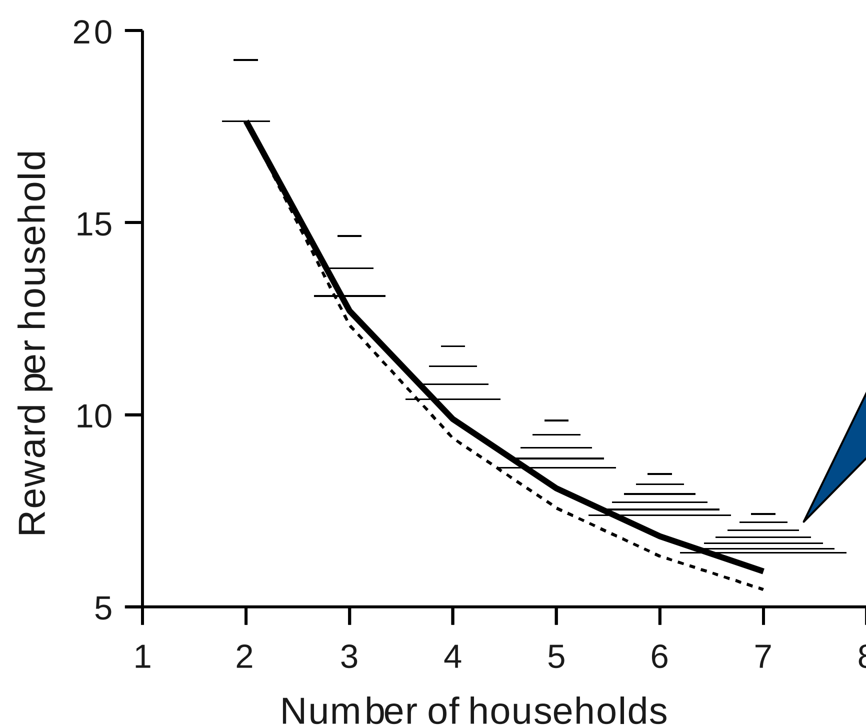
- **Microgrid management**
  - A team from Oxford has shown that a deviation consisting in **ignoring the threshold** might be profitable for individual houses.



T. Chen, V. Forejt, M. Kwiatkowska, D. Parker and A. Simaitis. Automatic Verification of Competitive Stochastic Systems. Formal Methods in System Design, pages 1-32, Springer. February 2013.

Nash equilibrium - applications

- **Microgrid management**
  - A team from Oxford has shown that a deviation consisting in **ignoring the threshold** might be profitable for individual houses.

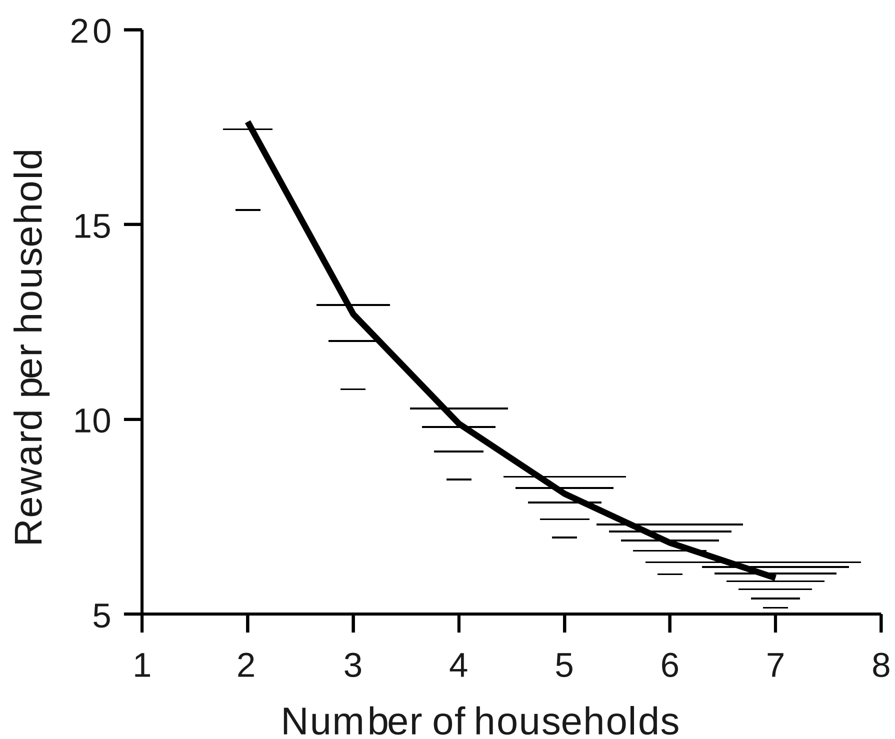


Each line represents the reward obtained when a coalition deviates (width = size of coalition)

T. Chen, V. Forejt, M. Kwiatkowska, D. Parker and A. Simaitis. Automatic Verification of Competitive Stochastic Systems. Formal Methods in System Design, pages 1-32, Springer. February 2013.

Nash equilibrium - applications

- **Microgrid management**
  - One possible solution: allow the DM to cancel one job per step each the cost exceeds the threshold



T. Chen, V. Forejt, M. Kwiatkowska, D. Parker and A. Simaitis. Automatic Verification of Competitive Stochastic Systems. Formal Methods in System Design, pages 1-32, Springer. February 2013.

Nash equilibrium - applications

- **Managing power of wireless devices**:
  - Consider a set of wireless devices that communicate to a base station
  - The higher the emitting power of the device, the higher the bandwidth
  - If a protocol fixes a maximal emitting power, each device has an incentive to deviate, unless the protocol punishes it.

Nash equilibrium - applications

- **File sharing in peer-to-peer systems**:
  - –Each peer owns some parts of the file
  - –All peers want to acquire the file
  - –In the bittorrent protocol, each peer uploads only to the other peers that have contributed most
  - –Do peers have an incentive to deviate ?
  - –Yes ! bittorrent is not a Nash equilibrium
    - Is it an epsilon-N.E. ?

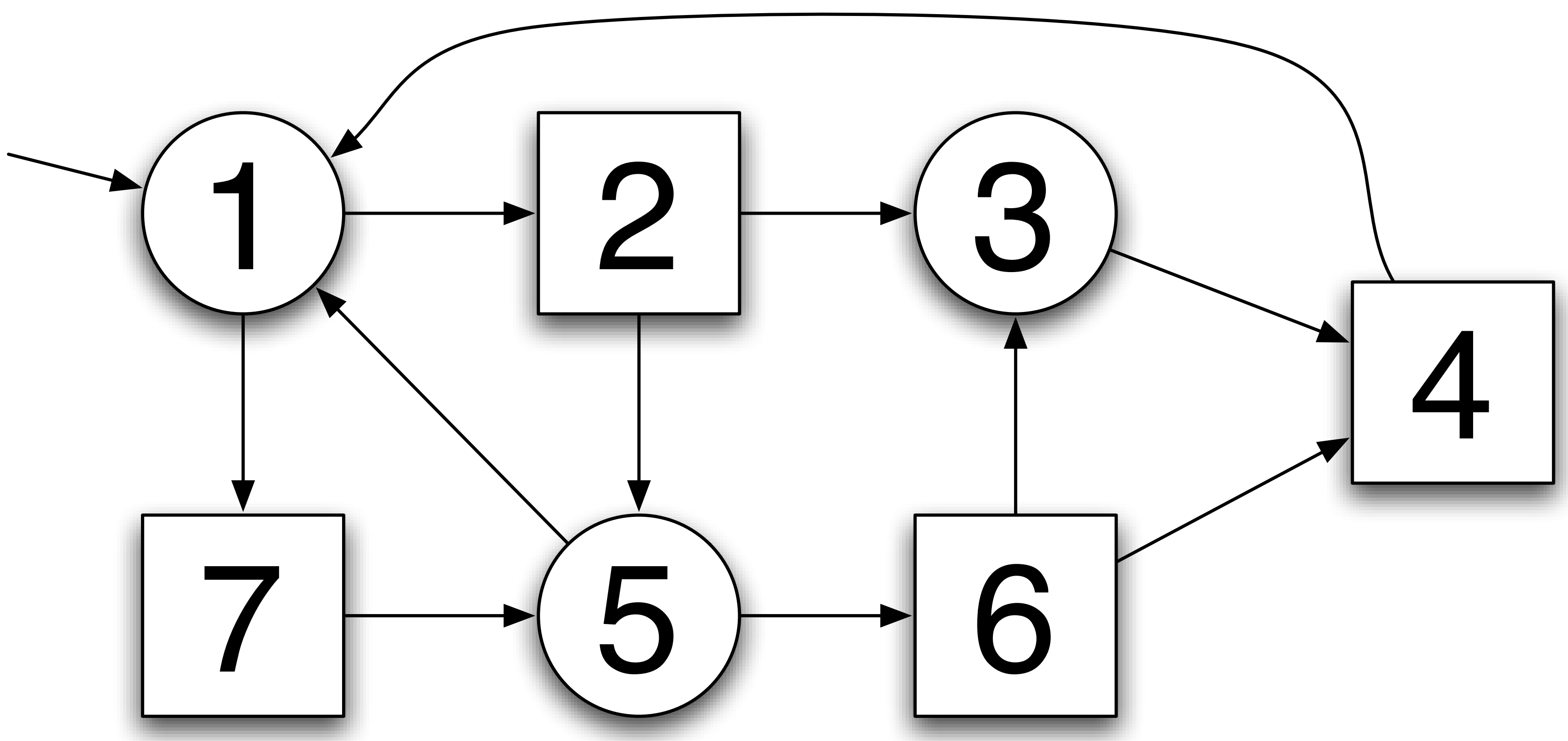
Games played on graphs

Games graphs

- We consider again games with two players
- We will us **graphs** with two types of nodes
  - –Some nodes are controlled by player A
  - –The other nodes are controlled by B
- A **play** will be a **path** in the graph
- Deciding where to move next is the responsibility of the player who controls the node

Game graphs

- **Example**:
  - B plays with rond nodes
  - A plays with square nodes



ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Game graphs

- **Definition**: An **arena** is a tuple $\langle Q, q_0, E\rangle$

  where:

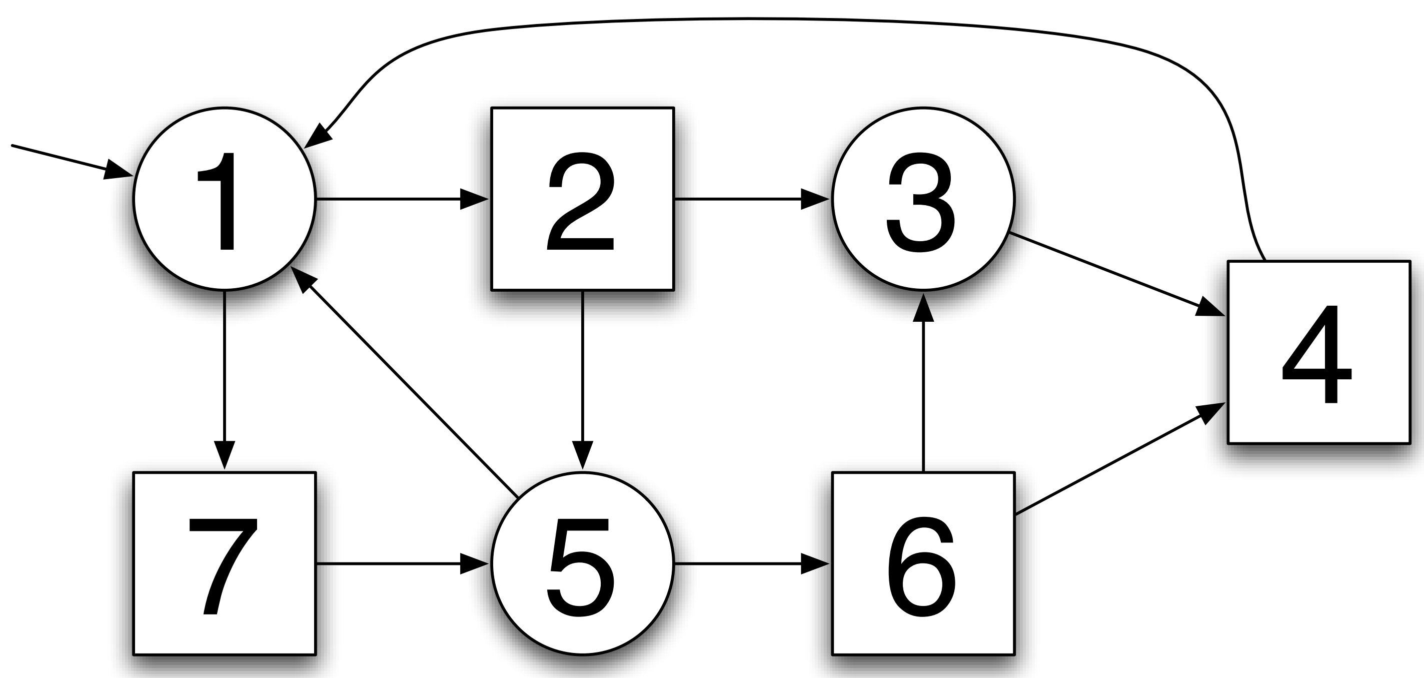
  – $Q=Q_A \cup Q_B$ (with $Q_A \cap Q_B = \varnothing$) is the set of **nodes**. Nodes in Q_A (resp. Q_B) are **controlled** by player A (B)

- $q_0 \in Q$ is the **initial node**
- $E \subseteq Q \times Q$ is the set of **edges**.

Game graphs

- **Definition**: A **play** in an arena ⟨Q, q_0, E⟩

  is an **infinite sequence** $r_1r_2r_3$... s.t. $r_1=q_0$ et $\forall i \geq 1$: $(r_i, r_{i+1}) \in E$. It is thus an infinite path in the graph, starting from q_0.



1 2 3 4 1 7 5 1 7 5 1 7 5 1...

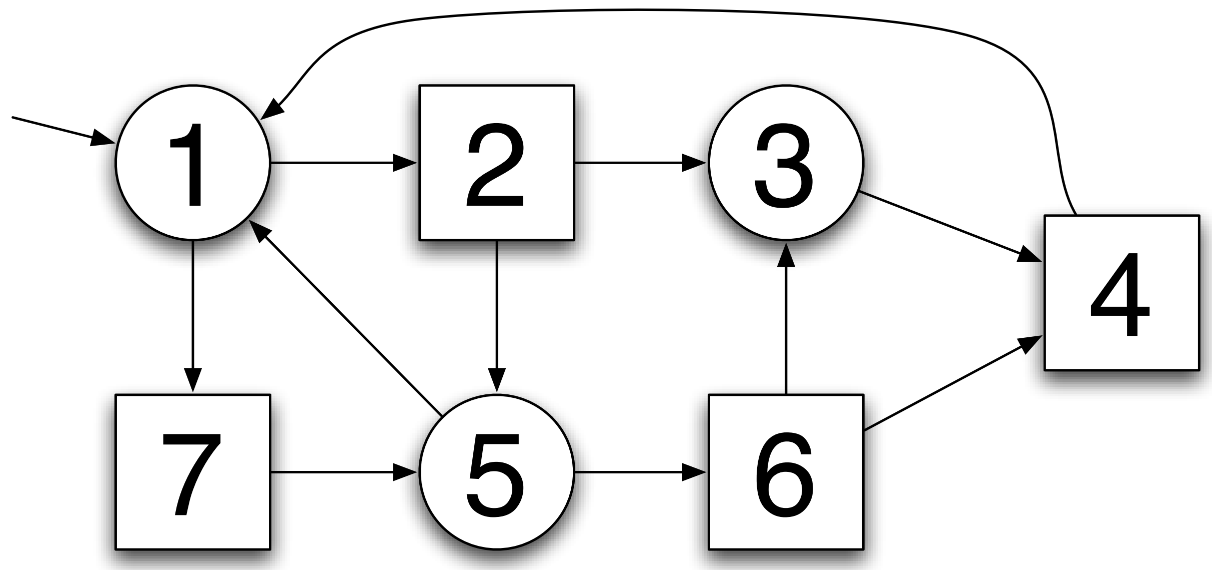
Winning conditions

- To determine who **wins** the play, we will use so-called **Muller conditions**:
- Let p be a play in an arena:
  - **Inf(p)** = set of nodes that appear infinitely often in p
  - **Occ(p)** = set of nodes that appear in p
- **Example**: for
  p= 1 2 3 4 1 7 5 1 7 5 1 $(7\ 5\ 1)^{\omega}$
  - Inf(p)={1,5,7}
  - Occ(p)= {1,2,3,4,5,7}

Winning conditions

- Let us fix a set **F** of sets of nodes of the arena, and a play **p**
- In general, there are two kinds of Muller conditions:
  - –**Weak conditions**: p is winning iff Occ(p)∈F
  - –**Strong conditions**: p is winning iff Inf(p)∈F
- We will focus on certain kinds of weak conditions, i.e. **safety** and **reachability** conditions.

Winning conditions



- **Example**:
  - 1234(175)$^{\omega}$ **wins** for the **weak** condition {{1,2,3,4, 5, 7},{1,5,7}} and for the **strong** one {{1,5,7},{1,2}}
  - 1234(175)$^{\omega}$ **looses** for the **strong** condition {{1,2,3},{1,2,3,5,7}} and for the **weak** condition {{1,4}}

Games on graphs

- **Definition**: An infinite game is a pair $\langle G, \phi \rangle$ where:
  - G is an arena
  - ϕ is a Muller condition for one of the players

Safety

- If the Muller condition is a weak cond. of the form {S' | S' ⊆ S} for a given set S, we have a **safety game** (S = safe states).
  - e.g.: F={{1,2,3},{1,2},{1,3},{2,3},{1},{2},{3}}, with Q={1,2,3,4,5}. We win **if we visit only states 1, 2 or 3** (we we don't have to see them all)
- **Example**: A pump has to **maintain** a certain level of liquid in a tank. The safe level is specified by an **upper and a lower bound** (no under or overflow).

ULB

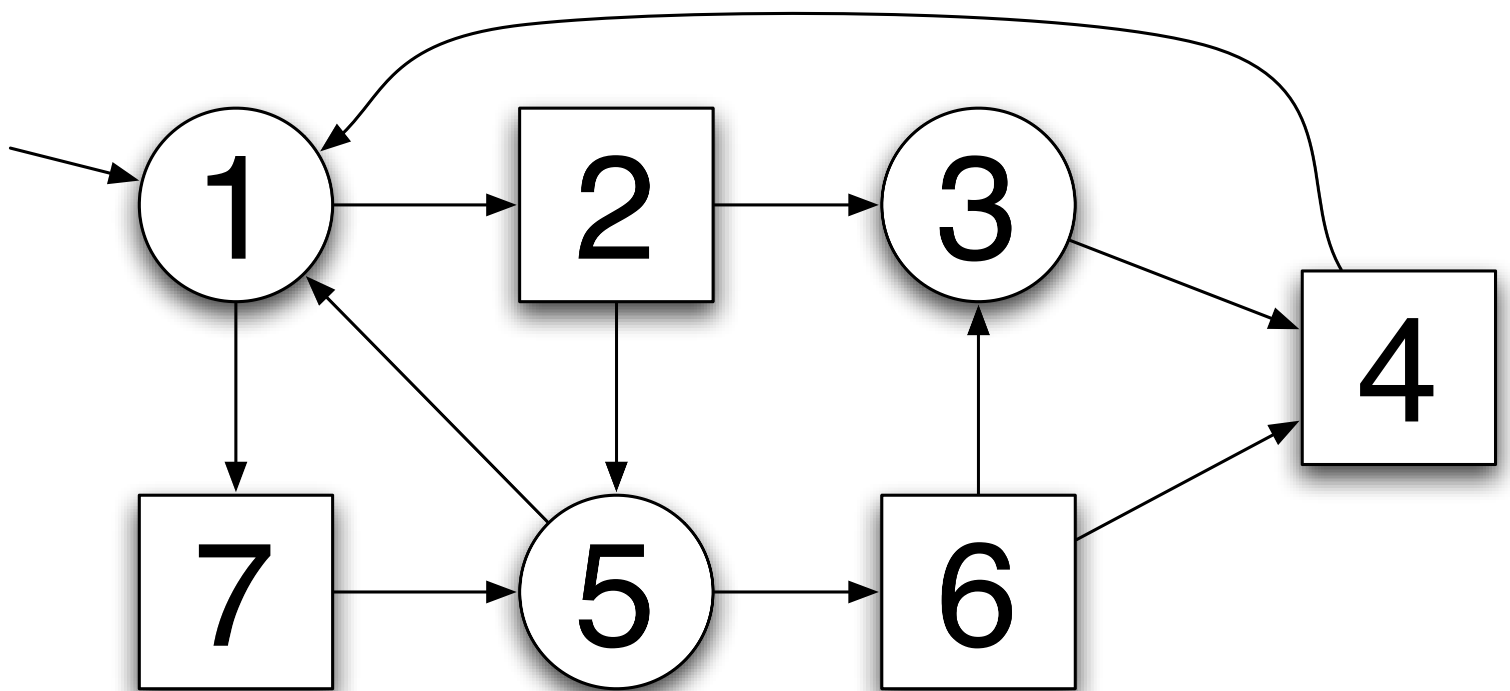
UNIVERSITÉ LIBRE DE BRUXELLES

Reachability

- If the Muller condition is a weak cond. of the form **{S|q∈S}** for some vertex q, then we have a **reachability game**.
  - –e.g.: F={{1},{1,2},{1,3}, {1,2,3}}, with Q={1,2,3}. We win if we **force the game to reach 1**.
- **Example**: A system has to initialise, and we should ensure that it **visits at least once** an “init completed” state, to make sure it does not deadlock during the initialisation phase.

Example

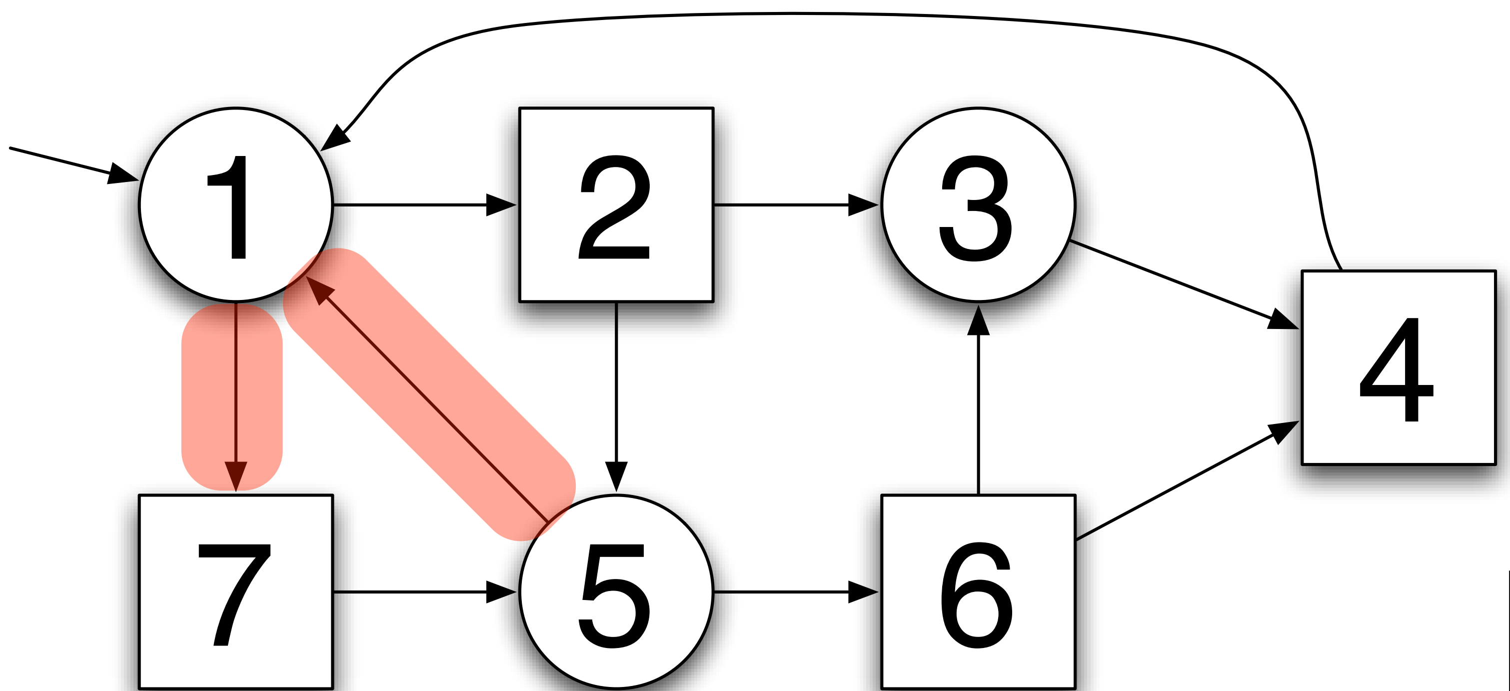
- With the following arena and the strong condition **{{1,5,7}}** for player B



Does B have winning strategy ?

Example

- With the following arena and the strong condition **{{1,5,7}}** for player B

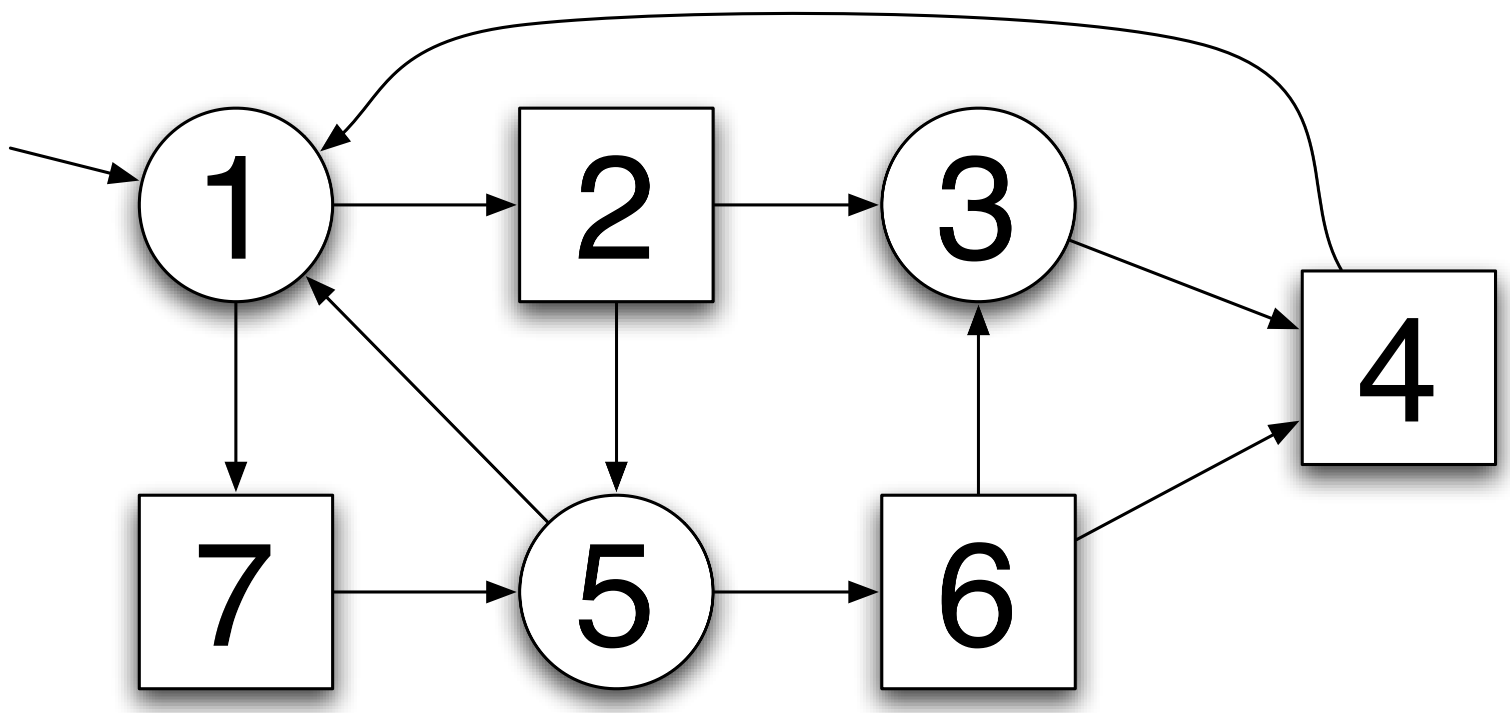


yes !

Does B have winning strategy ?

Example

- With this arena and the strong condition **$\{S \mid \{2,7\} \subseteq S\}$** for B

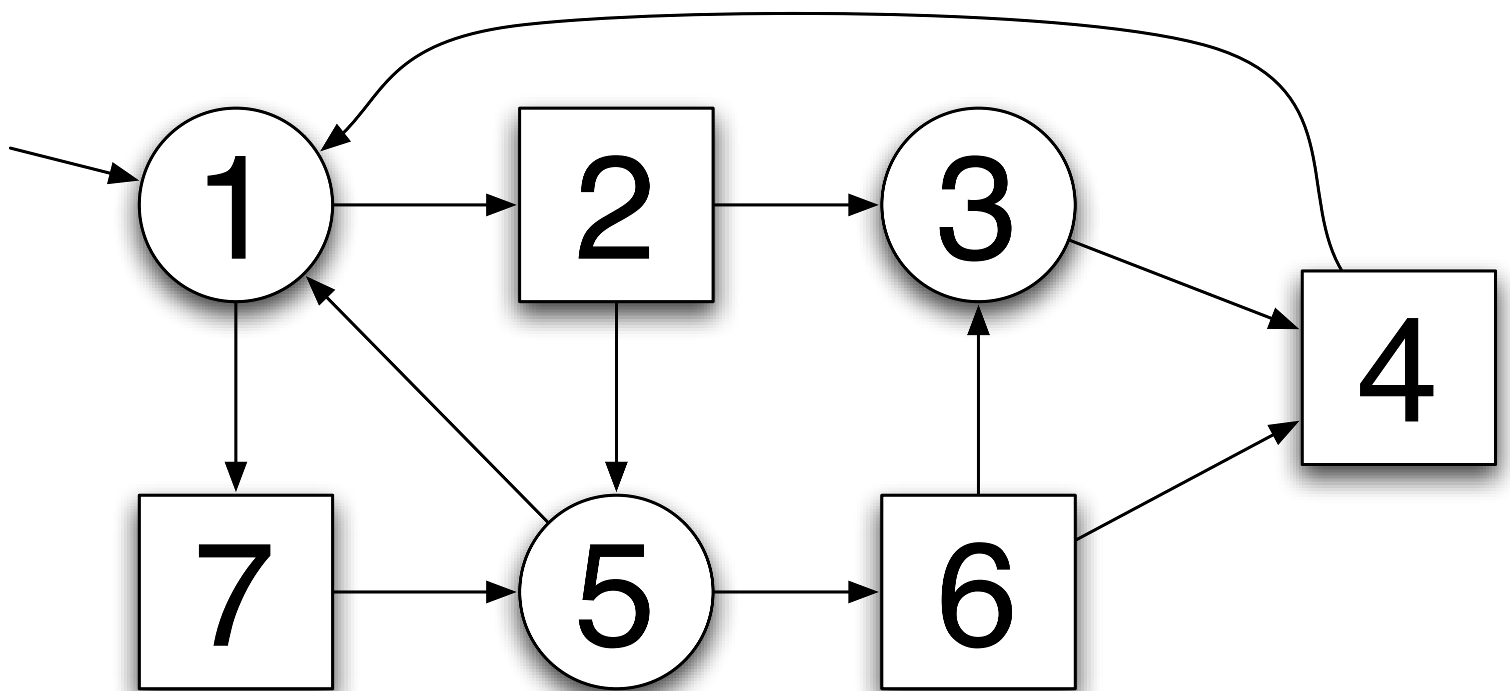


Does B have winning strategy ?

Example

- With this arena and the str
{S|{2,7}⊆S} for B

Yes !
from 1: alternate between 2 and 7



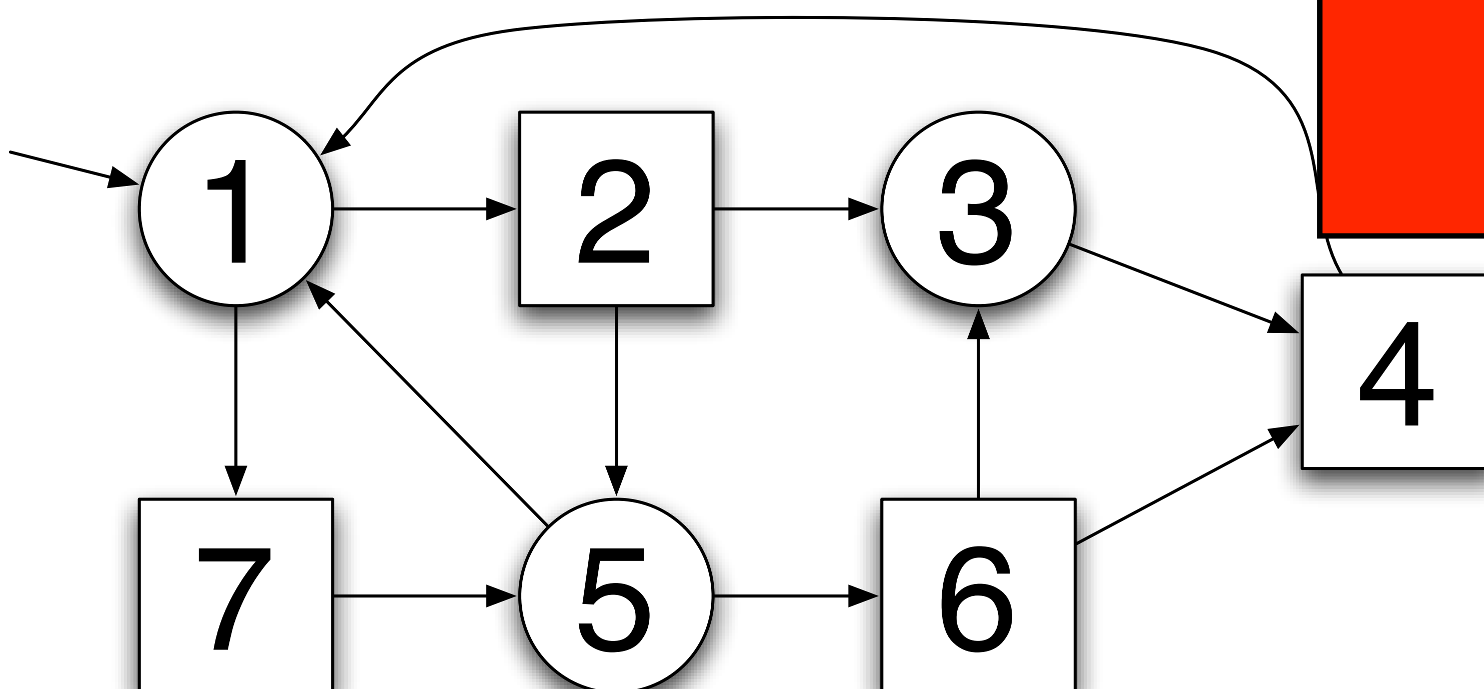
Does B have winning strategy ?

Example

- With this arena and the str
{S|{2,7}⊆S} for B

Yes !
from 1: alternate between 2 and 7

We need memory !



Does B have winning strategy ?

vendredi 7 mars 14

Strategies

- **Definition**: A **strategy** for player X in an arena $\langle Q, q_0, E\rangle$ is a function

  $f: Q^*Q_X \rightarrow Q$ s.t. for all $\sigma q \in Q^*Q_X$: $(q, f(\sigma q)) \in E$.

- **Intuitively**, for all play prefix σq ending in a node controlled by X, f(σq) gives the **next location to play**.
- This possible only if the edge (q, f(σq)) exists

Strategies

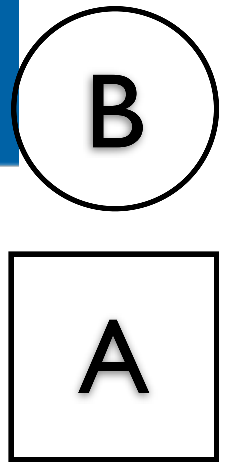
- **Definition**: A play $p=r_1r_2r_3\ldots$ **respects** a strategy **f** (for X) iff: for all **i**: $r_i \in Q_X$ implies $r_{i+1} = f(r_1r_2\ldots r_i)$
- Intuitively, anytime we visit an X location, we chose the **successor given by the strategy**.

Strategies

- **Definition**: A strategy f (for X) in an arena A is **winning** for X in the game G= ⟨A,ϕ⟩

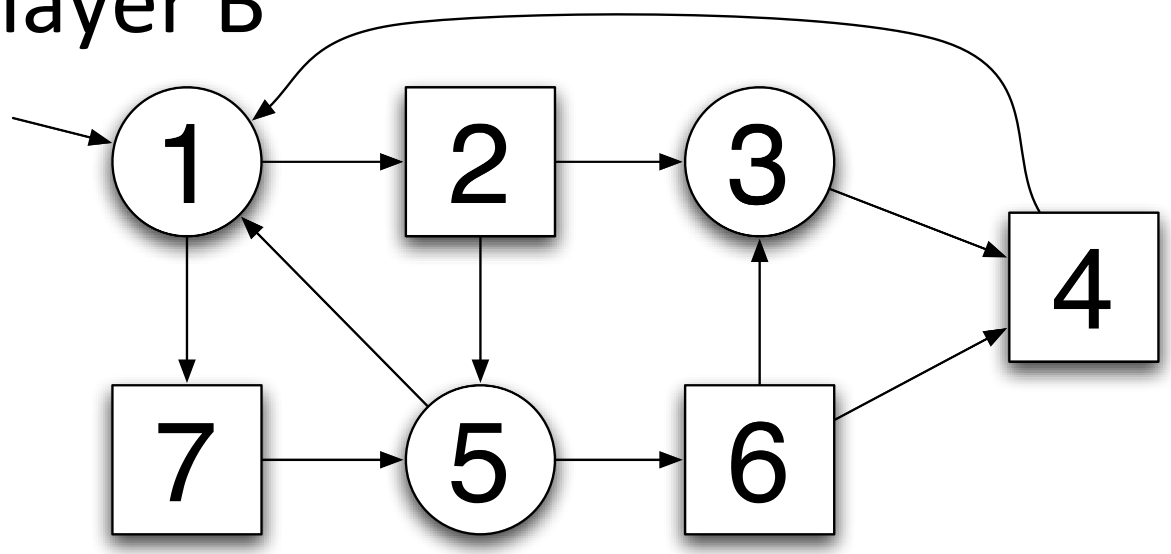
  iff

  for all play p of G: **if** p is played according to f, **then** p is winning for ϕ.
- **Whatever the adversary of X does, X is certain to win**, because the objective ϕ is fulfilled in the resulting play.



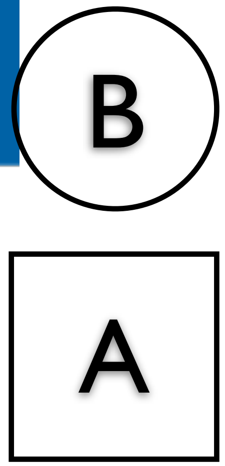
Example

- **Example**: For this arena and the strong condition {{1,5,7}} for player B



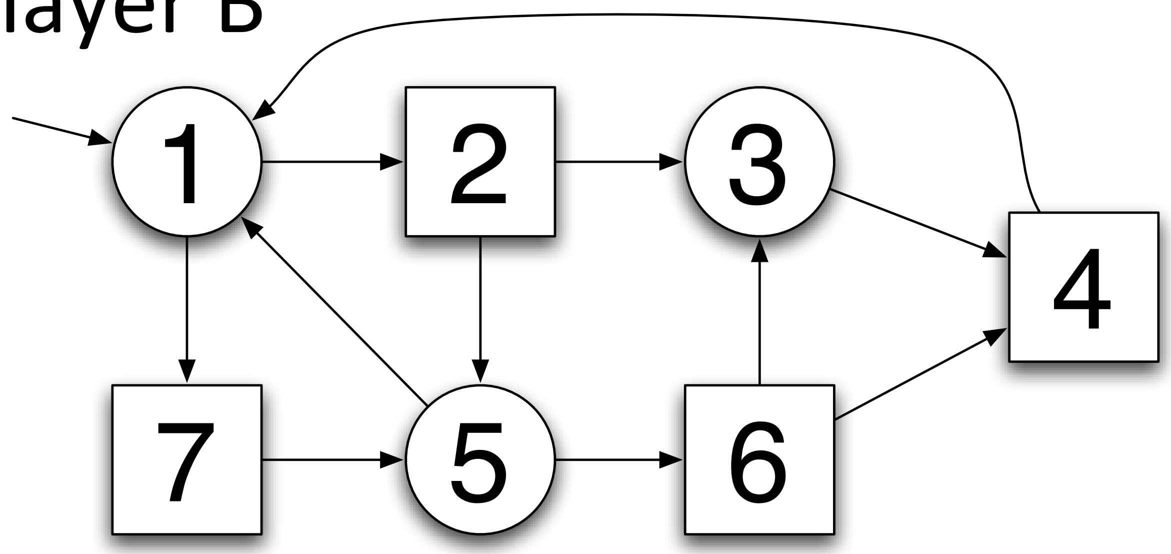
- Winning strategy:
  – $\forall \sigma \in Q^*: f(\sigma 1)=7$
  – $\forall \sigma \in Q^*: f(\sigma 5)=1$
  – $\forall \sigma \in Q^*: f(\sigma 3)=4$

We could have chosen any successor here



Example

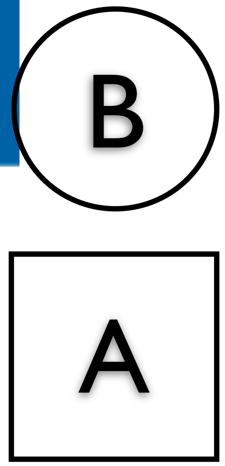
- **Example**: For this arena and the strong condition {{1,5,7}} for player B



- Winning strategy:
  - $\forall \sigma \in Q^*: f(\sigma 1)=7$
  - $\forall \sigma \in Q^*: f(\sigma 5)=1$
  - $\forall \sigma \in Q^*: f(\sigma 3)=4$

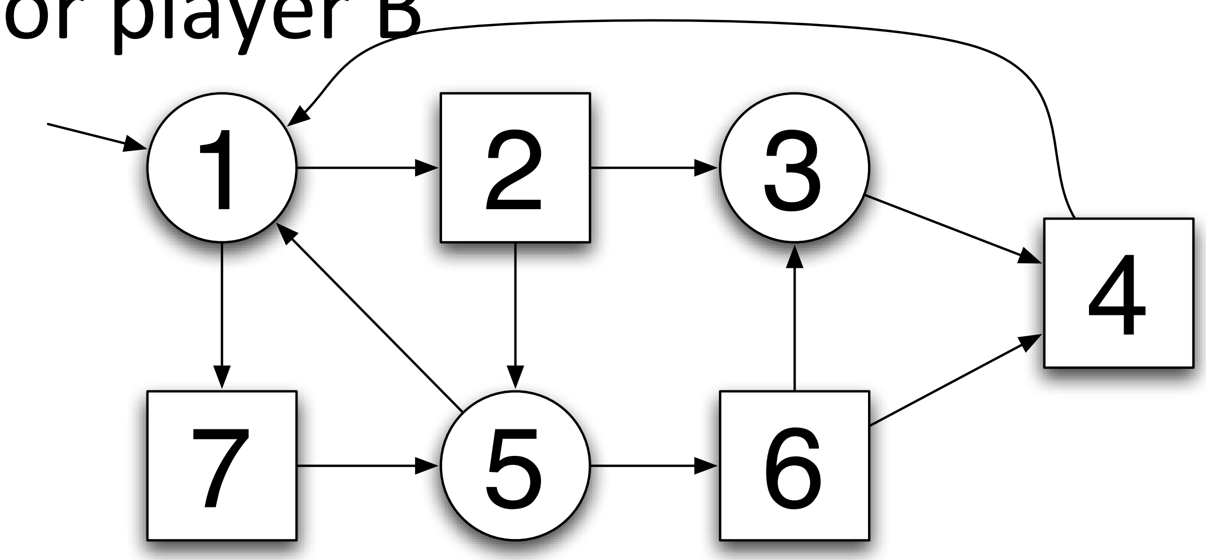
The strategy depends only on the current state

We could have chosen any successor here

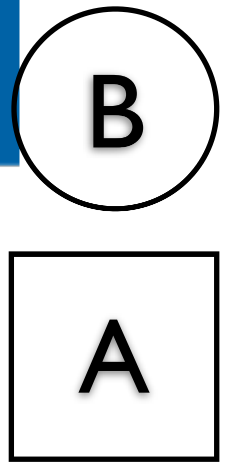


Example

- **Example**: With this arena and the strong condition $\{S \mid \{2,7\} \subseteq S\}$ for player B

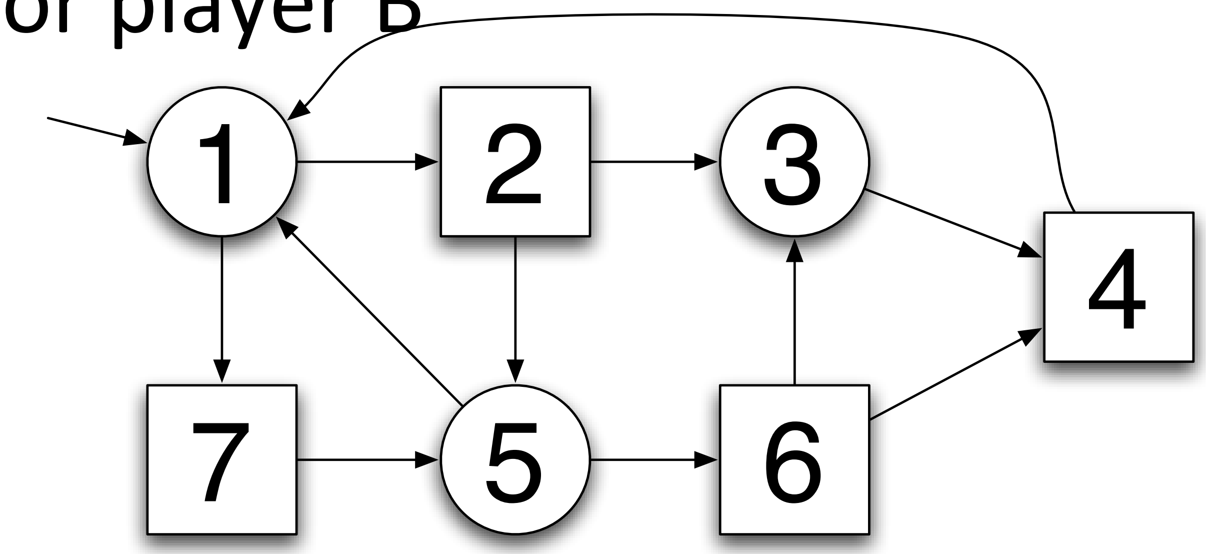


- Winning strategy for B:
  - $\forall \sigma \in Q^*: f(\sigma 1)=7$ if $\exists\, i: \sigma_i=2$ and $\forall\, j>i: \ \sigma_j \notin \{2,7\}$; $f(\sigma 1)=2$ otherwise
  - $\forall \sigma \in Q^*: f(\sigma 5)=1$
  - $\forall \sigma \in Q^*: f(\sigma 3)=4$



Example

- **Example**: With this arena and the strong condition $\{S \mid \{2,7\} \subseteq S\}$ for player B

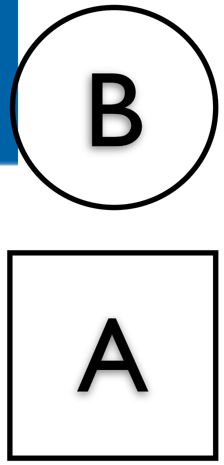


- Winning strategy for B:
  - $\forall \sigma \in Q^*: f(\sigma 1)=7$ if $\exists\ i: \sigma_i=2$ and $\forall\ j>i: \sigma_j \notin \{2,7\}$; $f(\sigma 1)=2$ otherwise
  - $\forall \sigma \in Q^*: f(\sigma 5)=1$
  - $\forall \sigma \in Q^*: f(\sigma 3)=4$

The strategy depends on the history !

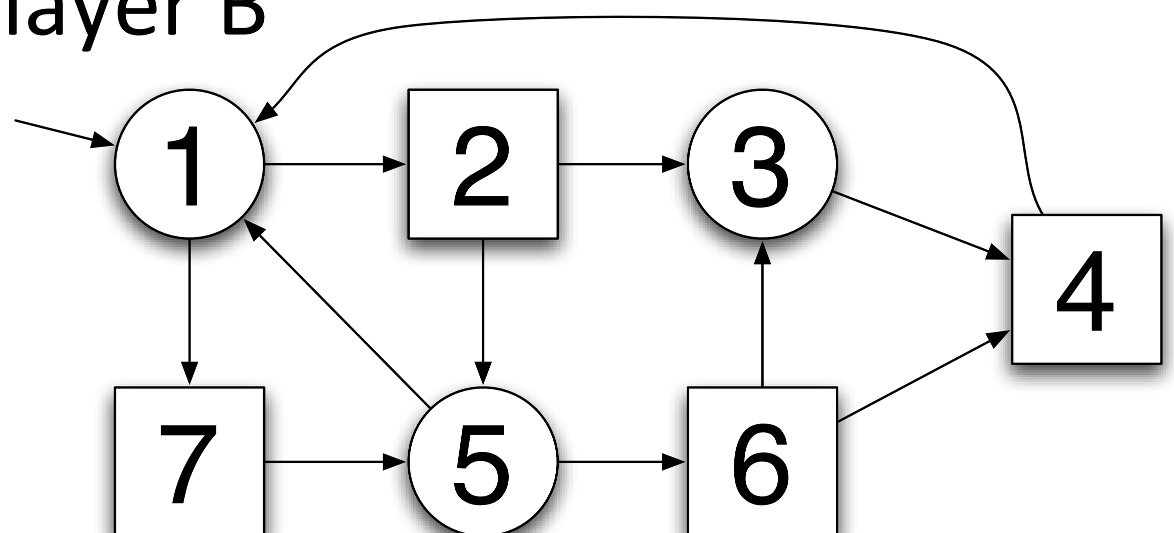
Strategies

- In general, a strategy can use **all the information given by the prefix** played so far.
- We want at least a **computable** strategy, but some (simple) cases are more interesting in practice:
  - If the strategy can be computed by a **finite automaton**, we have a **finite state** strategy
  - If the strategy depends on the **current location** only, we have a **positional strategy**

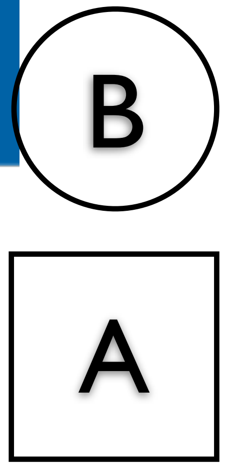


Example

- **Example**: For this arena and the strong condition {{1,5,7}} for player B

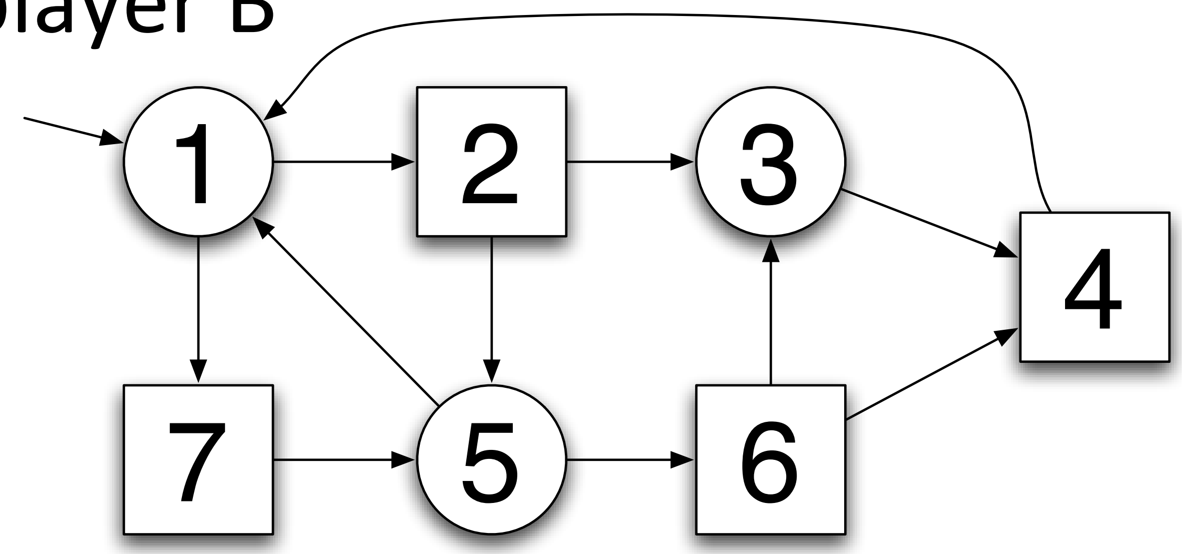


- Winning strategy:
  - $\forall \sigma \in Q^*: f(\sigma 1)=7$
  - $\forall \sigma \in Q^*: f(\sigma 5)=1$
  - $\forall \sigma \in Q^*: f(\sigma 3)=4$



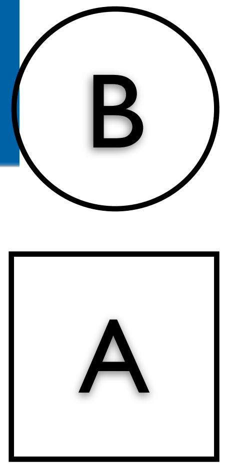
Example

- **Example**: For this arena and the strong condition {{1,5,7}} for player B



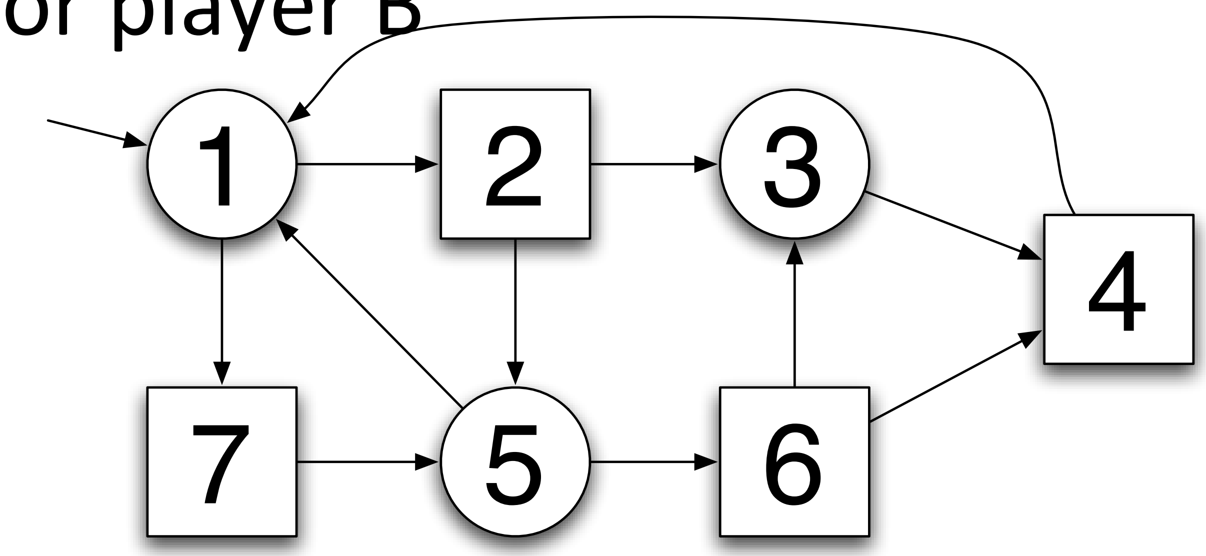
- Winning strategy:
  - –∀σ∈Q*:f(σ1)=7
  - –∀σ∈Q*:f(σ5)=1
  - –∀σ∈Q*:f(σ3)=4

Positional strategy !

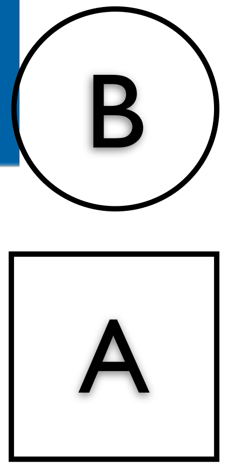


Example

- **Example**: With this arena and the strong condition $\{S \mid \{2,7\} \subseteq S\}$ for player B

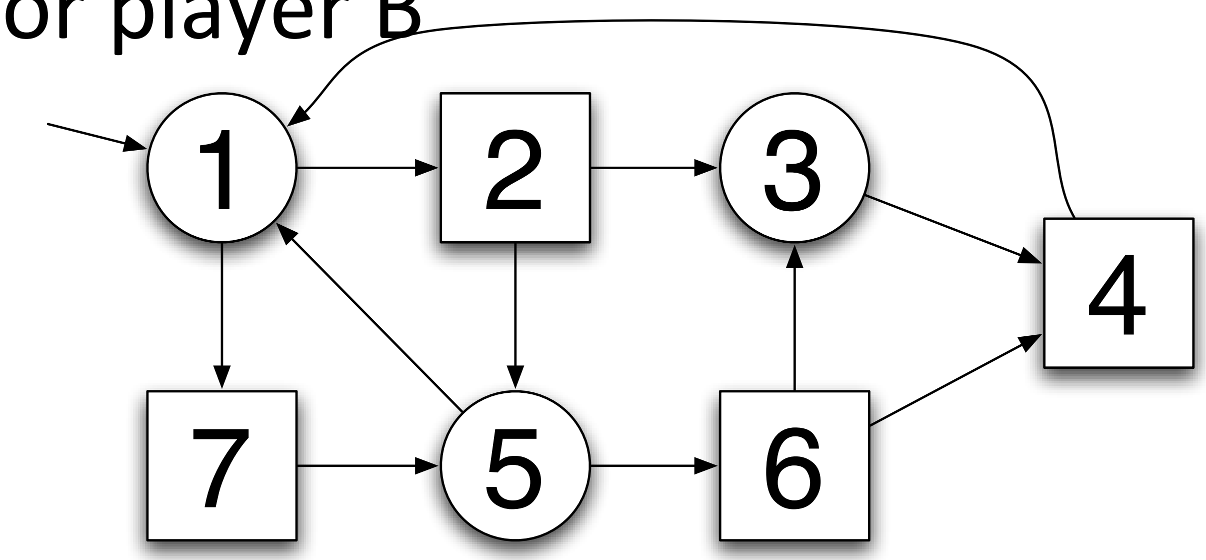


- Winning strategy for B:
  - $\forall \sigma \in Q^*: f(\sigma 1)=7$ if $\exists\ i: \sigma_i=2$ and $\forall\ j>i: \sigma_j \notin \{2,7\}$; $f(\sigma 1)=2$ otherwise
  - $\forall \sigma \in Q^*: f(\sigma 5)=1$
  - $\forall \sigma \in Q^*: f(\sigma 3)=4$



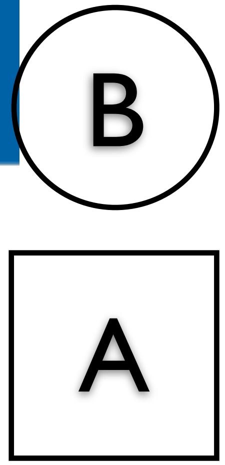
Example

- **Example**: With this arena and the strong condition $\{S \mid \{2,7\} \subseteq S\}$ for player B



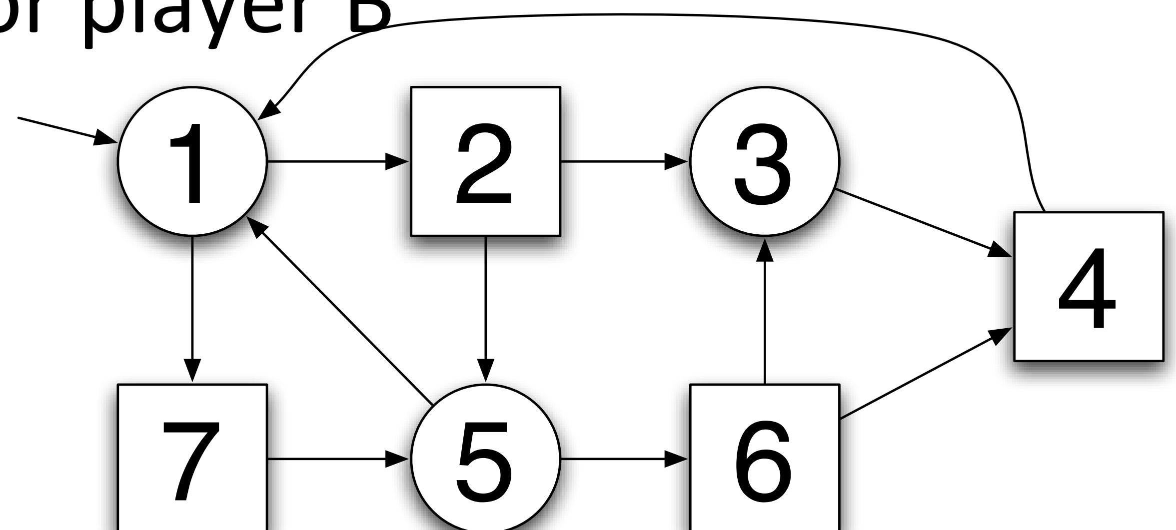
- Winning strategy for B:
  - $\forall \sigma \in Q^*: f(\sigma 1)=7$ if $\exists\, i: \sigma_i=2$ and $\forall\, j>i: \sigma_j \notin \{2,7\}$; $f(\sigma 1)=2$ otherwise
  - $\forall \sigma \in Q^*: f(\sigma 5)=1$
  - $\forall \sigma \in Q^*: f(\sigma 3)=4$

Finite state strategy !

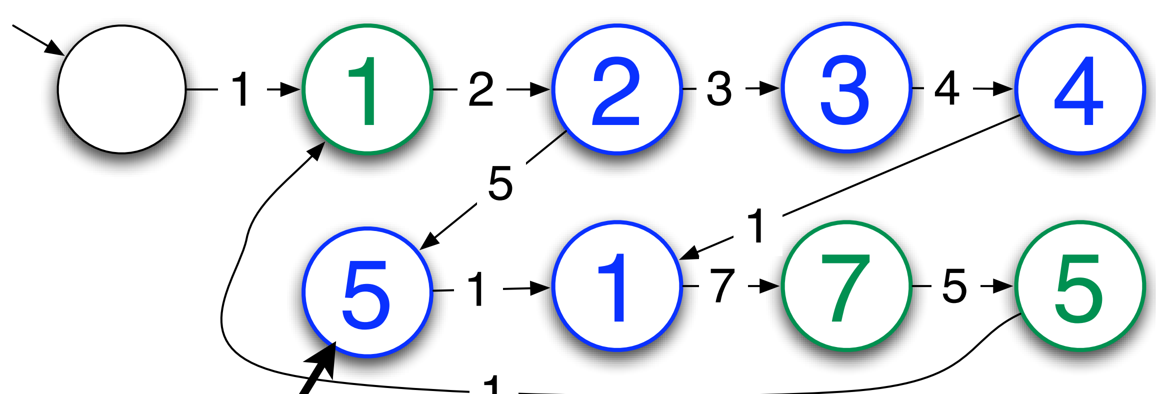


Example

- **Example**: With this arena and the strong condition {S|{2,7}⊆S} for player B



- Winning strategy for B:



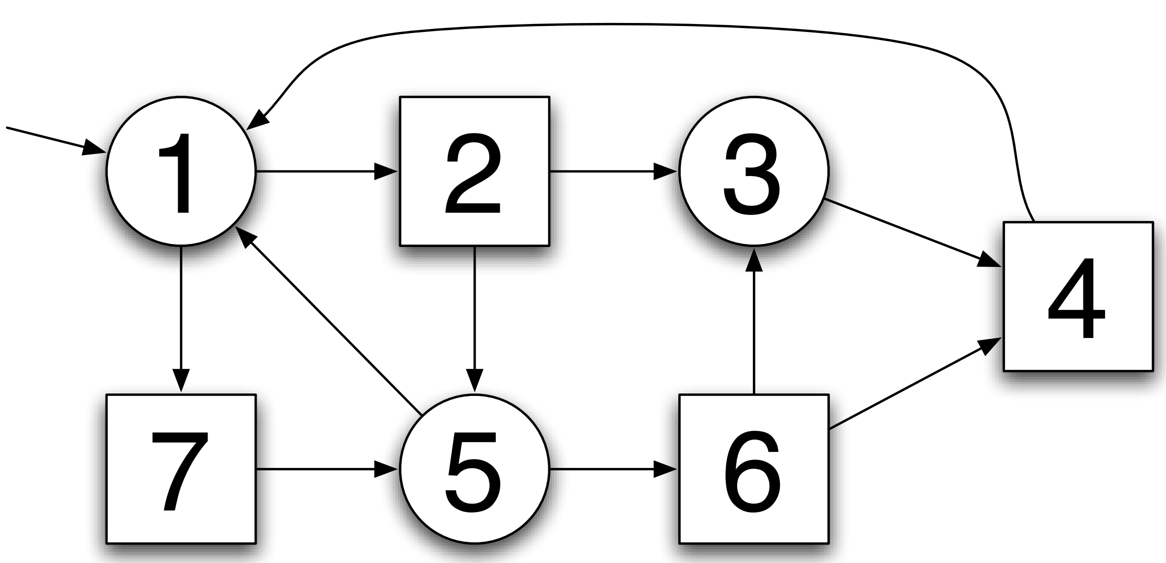
green: B plays 2 after next 1

blue: B plays 7 after next 1

Remembers current location

Positional strategies

- A **positional strategy** f for player X is a function that associates to each node of X **a successor node** (no need to remember the whole history)
- A positional strategy can thus be regarded as **a selection of the game edges**: for all node q of X, we keep only the edge (q,f(q))

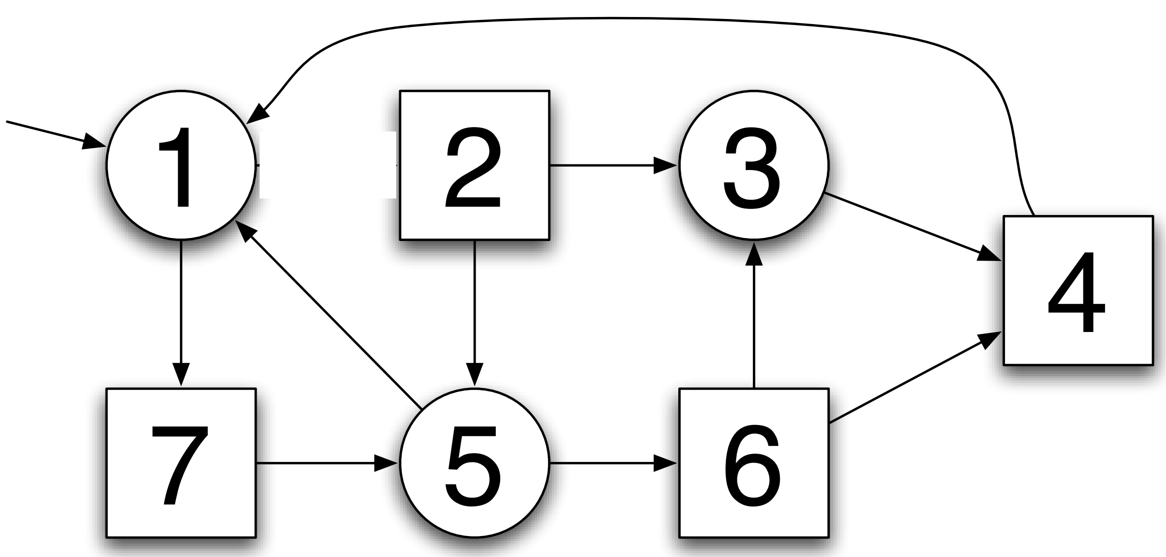


- $\forall \sigma \in Q^*: f(\sigma 1)=7$
- $\forall \sigma \in Q^*: f(\sigma 5)=1$
- $\forall \sigma \in Q^*: f(\sigma 3)=4$

ULB

Positional strategies

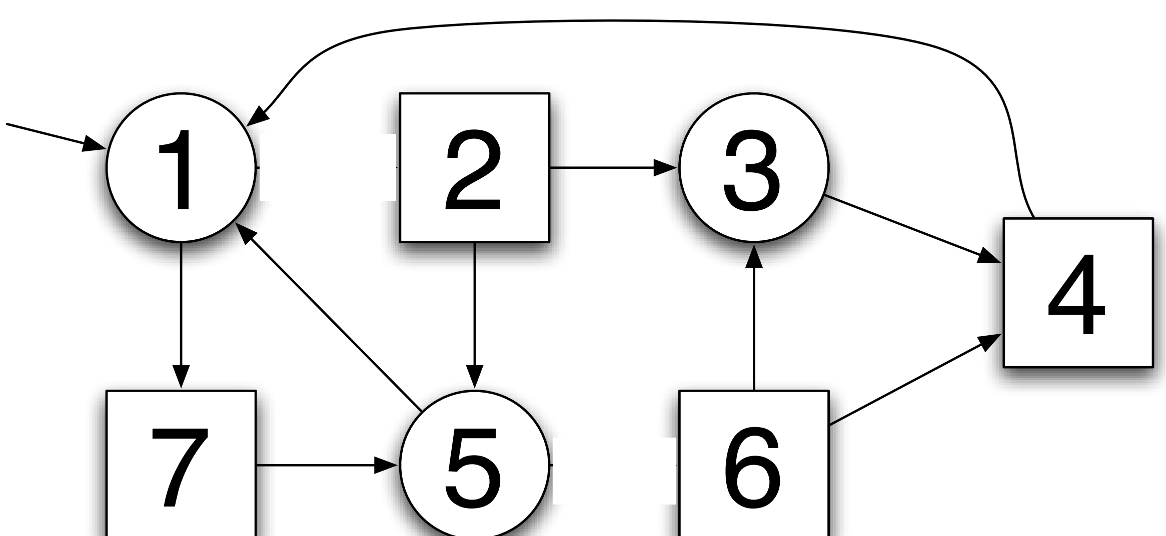
- A **positional strategy** f for player X is a function that associates to each node of X **a successor node** (no need to remember the whole history)
- A positional strategy can thus be regarded as **a selection of the game edges**: for all node q of X, we keep only the edge (q,f(q))



- $\forall \sigma \in Q^*: f(\sigma 1)=7$
- $\forall \sigma \in Q^*: f(\sigma 5)=1$
- $\forall \sigma \in Q^*: f(\sigma 3)=4$

Positional strategies

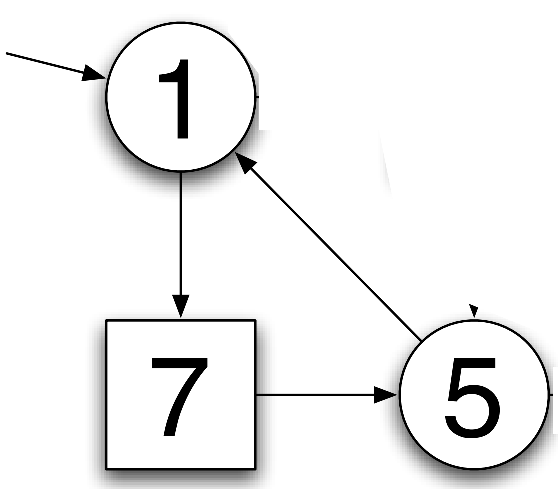
- A **positional strategy** f for player X is a function that associates to each node of X **a successor node** (no need to remember the whole history)
- A positional strategy can thus be regarded as **a selection of the game edges**: for all node q of X, we keep only the edge (q,f(q))



- $\forall \sigma \in Q^*: f(\sigma 1)=7$
- $\forall \sigma \in Q^*: f(\sigma 5)=1$
- $\forall \sigma \in Q^*: f(\sigma 3)=4$

Positional strategies

- A **positional strategy** f for player X is a function that associates to each node of X **a successor node** (no need to remember the whole history)
- A positional strategy can thus be regarded as **a selection of the game edges**: for all node q of X, we keep only the edge (q,f(q))



- $\forall \sigma \in Q^*: f(\sigma 1) = 7$
- $\forall \sigma \in Q^*: f(\sigma 5) = 1$
- $\forall \sigma \in Q^*: f(\sigma 3) = 4$

Determined games

- To solve those games we compute **two sets**:
  - **W_A** = the set of locations of the game from where A has a winning strategy
  - **W_B** = the set of locations of the game from where B has a winning strategy
- Clearly $W_A \cap W_B = \emptyset$
- But we could imagine games where in some positions **neither player has a winning strategy**

Determined games

- **Definition**: A game (with set of locations Q) is **determined** iff $W_A \cup W_B = Q$.
- **Theorem** (Borel - Martin): games with Muller objectives are **determined**.

E. Borel (1871-1956)

D. Martin (1940-)

Reachability games

- To define **reachability games** in a simple fashion, we consider an arena ⟨Q, q_0, E⟩

  and a set **T** of target nodes
- We want to compute a strategy for player A that **guarantees to reach T in all plays.**

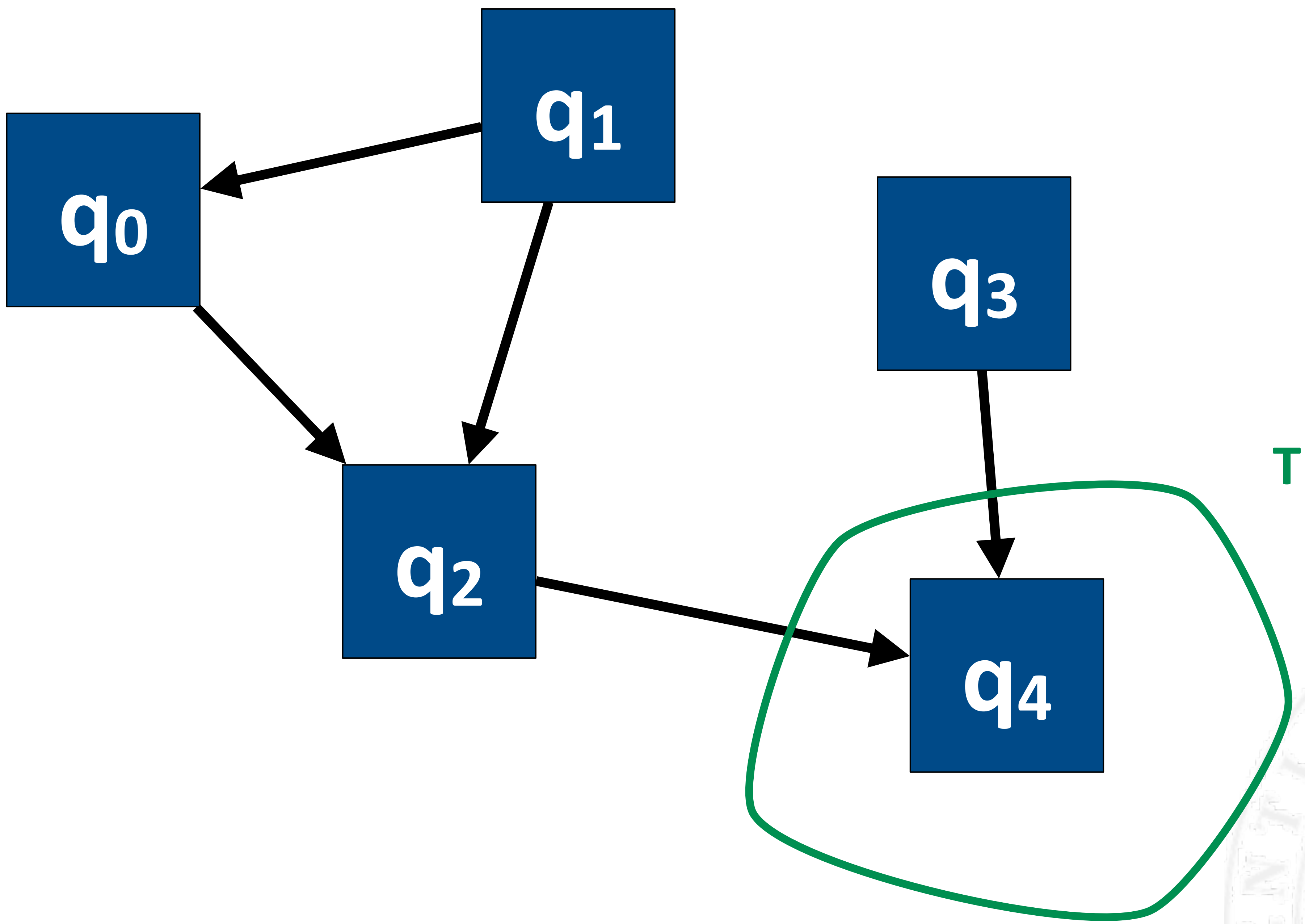
Reachability in 1-player games

- Let $\langle Q_A, q_0, E \rangle$ be a **1-player arena** (i.e., a plain graph)
- Let T be a set of **target nodes** that the player wants to reach
- In 1-player games, a simple solution is the (forward) **breadth-first search**
- It consists in computing the sets $\mathbf{R_i}$ defined as:
  - $\mathbf{R_0} = \{q_0\}$
  - $\mathbf{R_{i+1}} = R_i \cup \{q' \mid \exists\, q \in \mathbf{R_i}: (q,q') \in E\}$

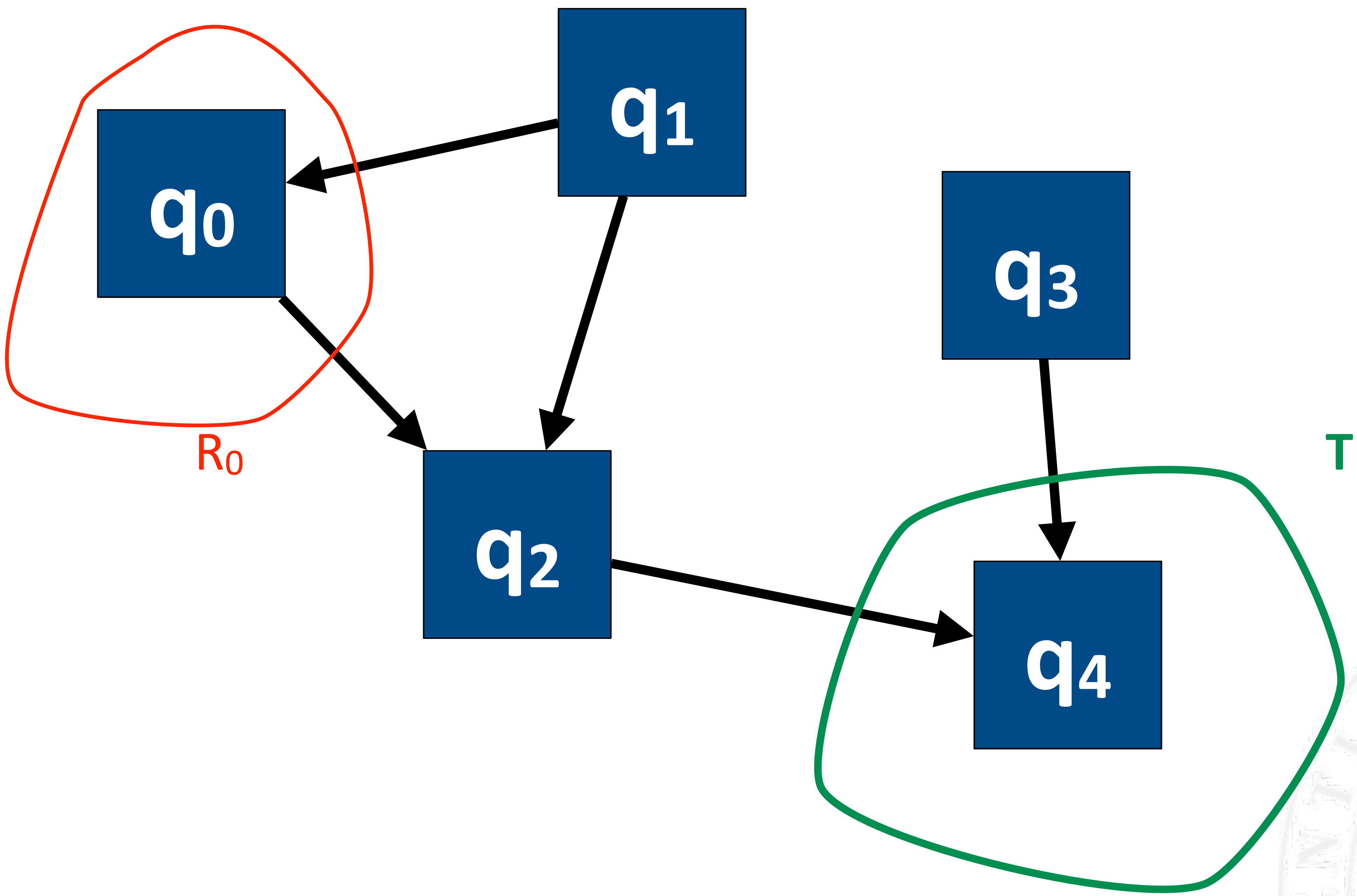
Reachability in 1-player games

- Intuitively, each set R_i contains all the vertices that can be reached from q_0 in **at most i steps**.
- This sequence **eventually stabilises**
  - Prove it !
  - Let **R*** denote the set obtained at stabilisation
- Then, the player has a strategy to reach **T** from q_0 iff $T \cap R^* \neq \emptyset$

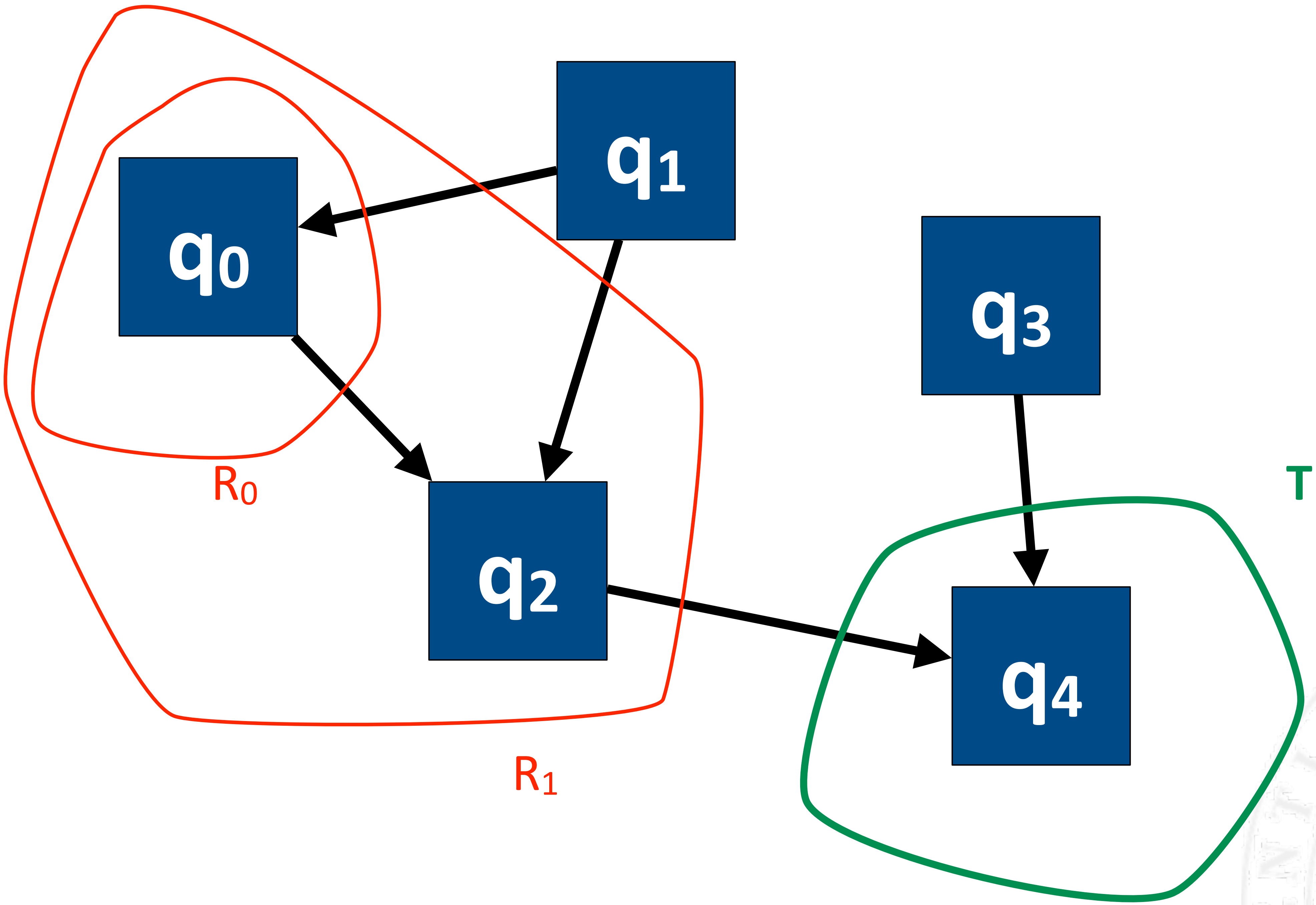
Reachability in 1-player games



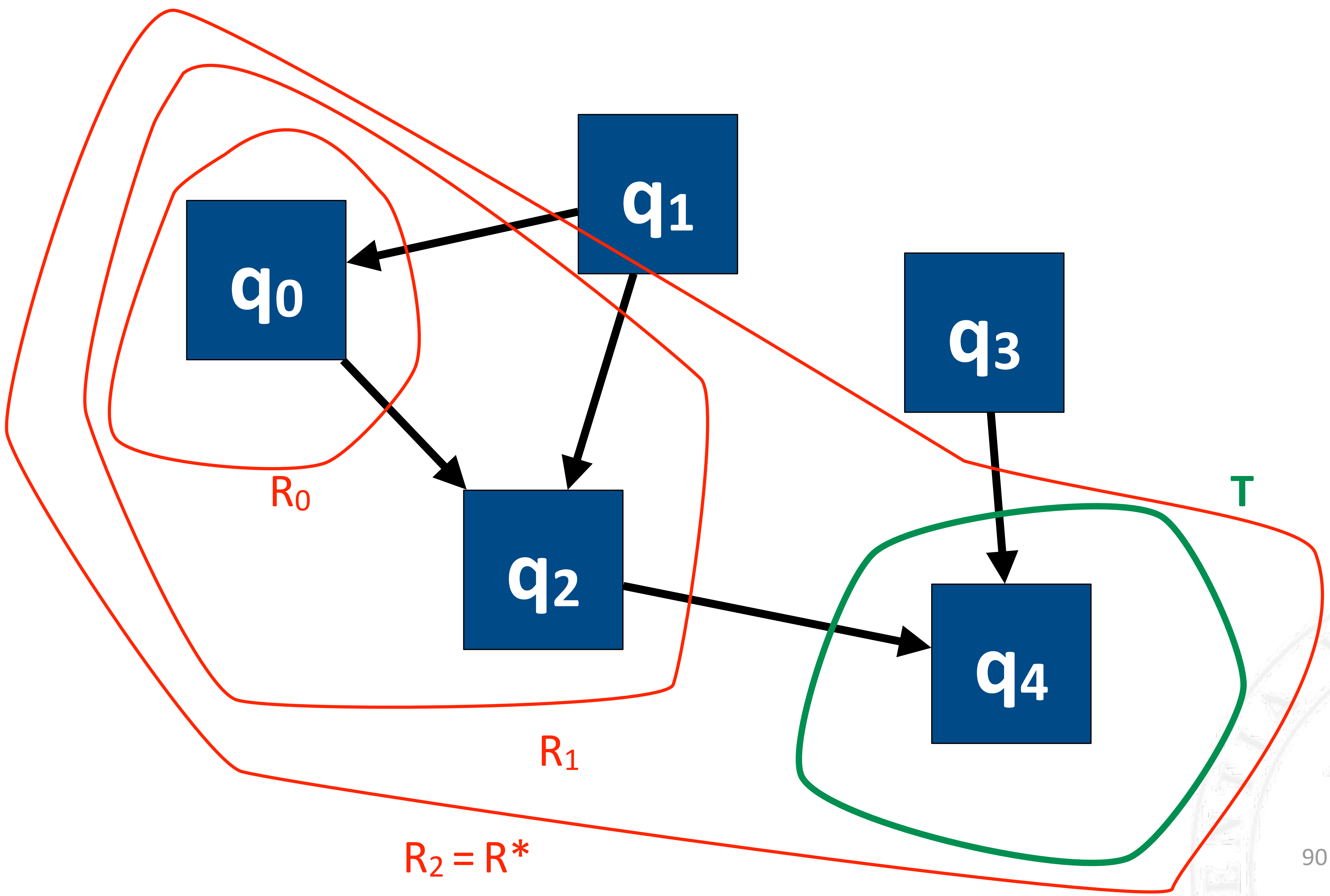
Reachability in 1-player games



Reachability in 1-player games



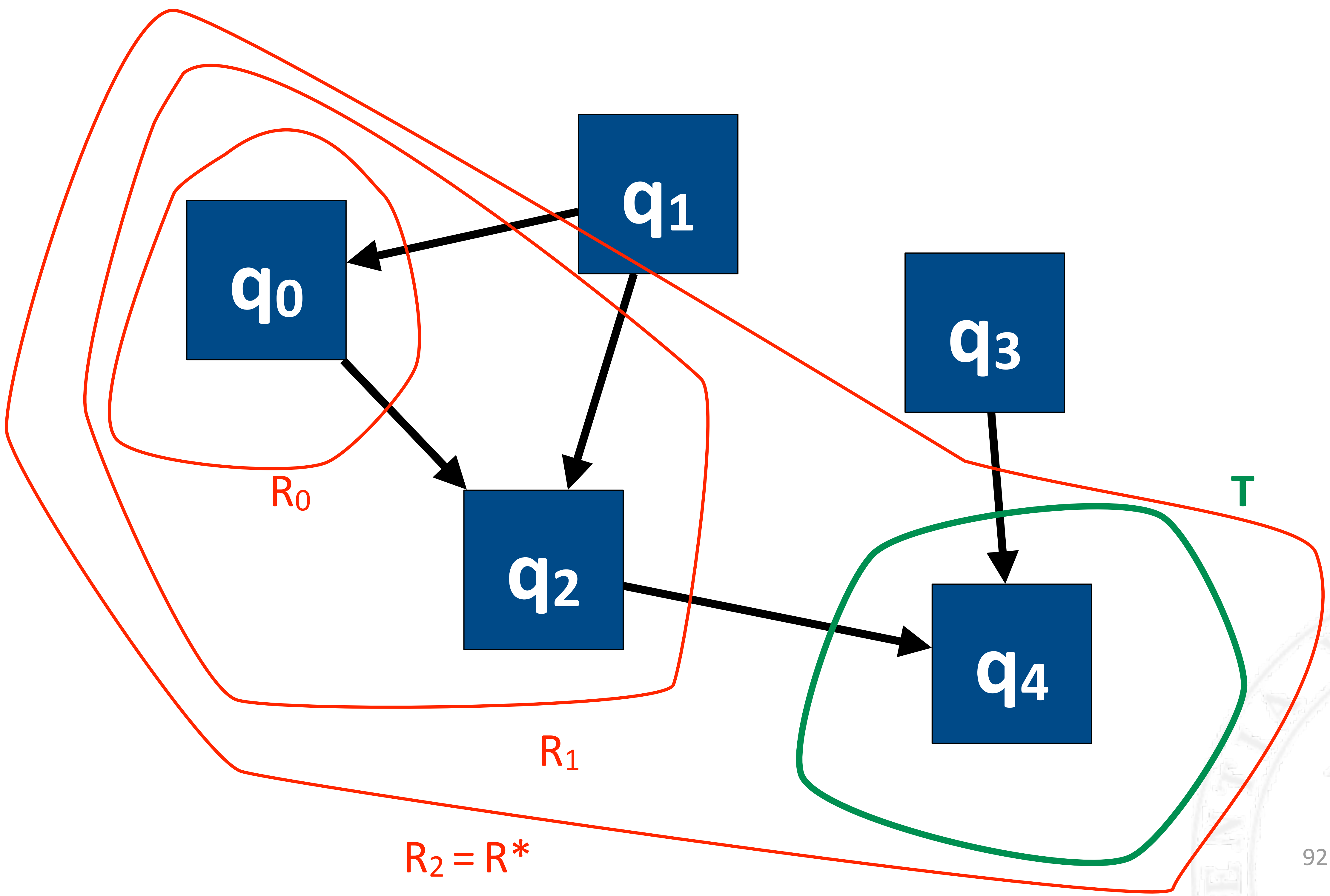
Reachability in 1-player games



Reachability in 1-player games

- Then, the strategy can be **extracted** from the sequence R_0, R_1,...
  - If a node q' has been added at step k, then, there is a node $q \in R_{k-1}$ and an edge (q,q').
  - The strategy from q is to go to q'.
  - Observe that this is a **positional strategy** !

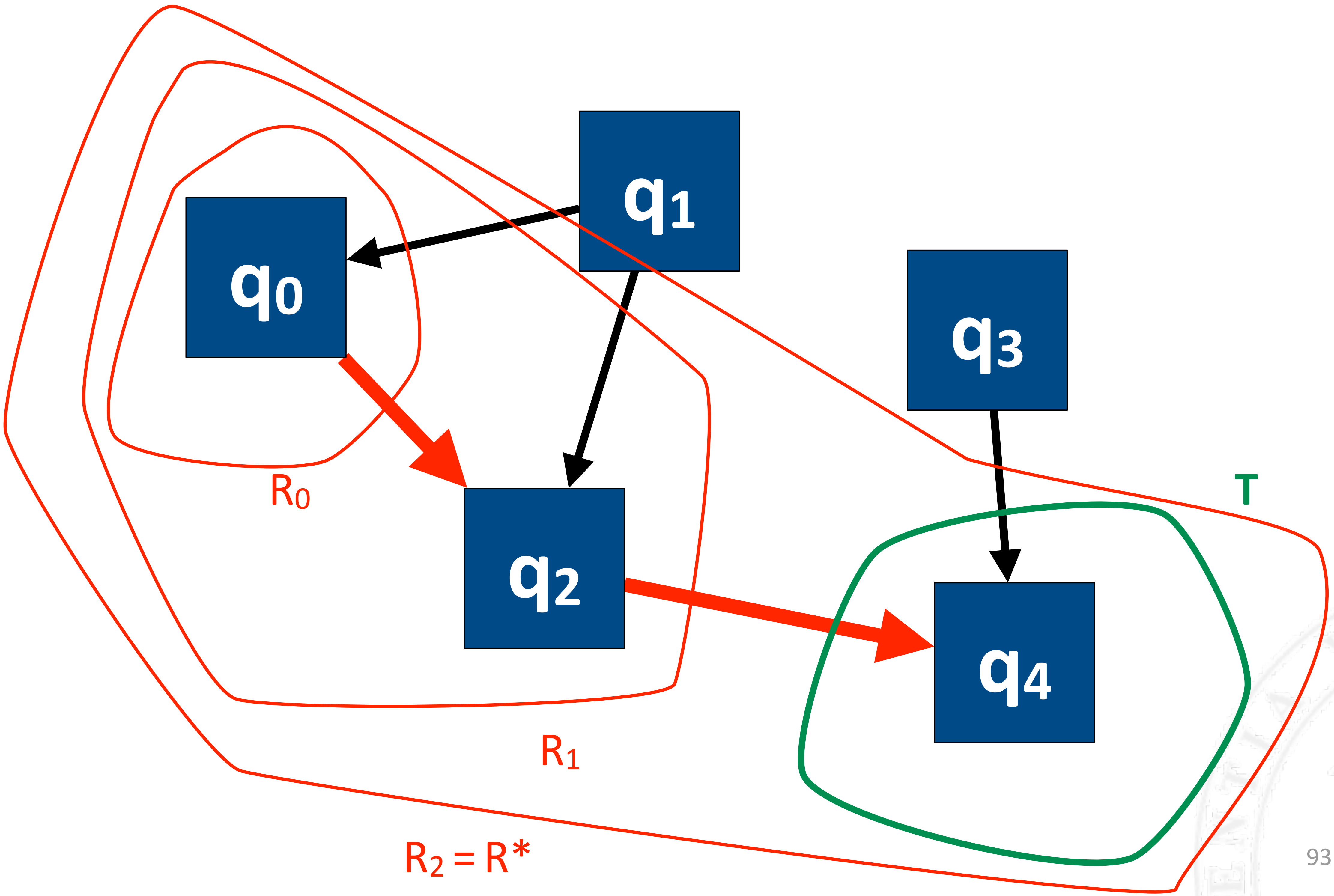
Reachability in 1-player games



ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Reachability in 1-player games



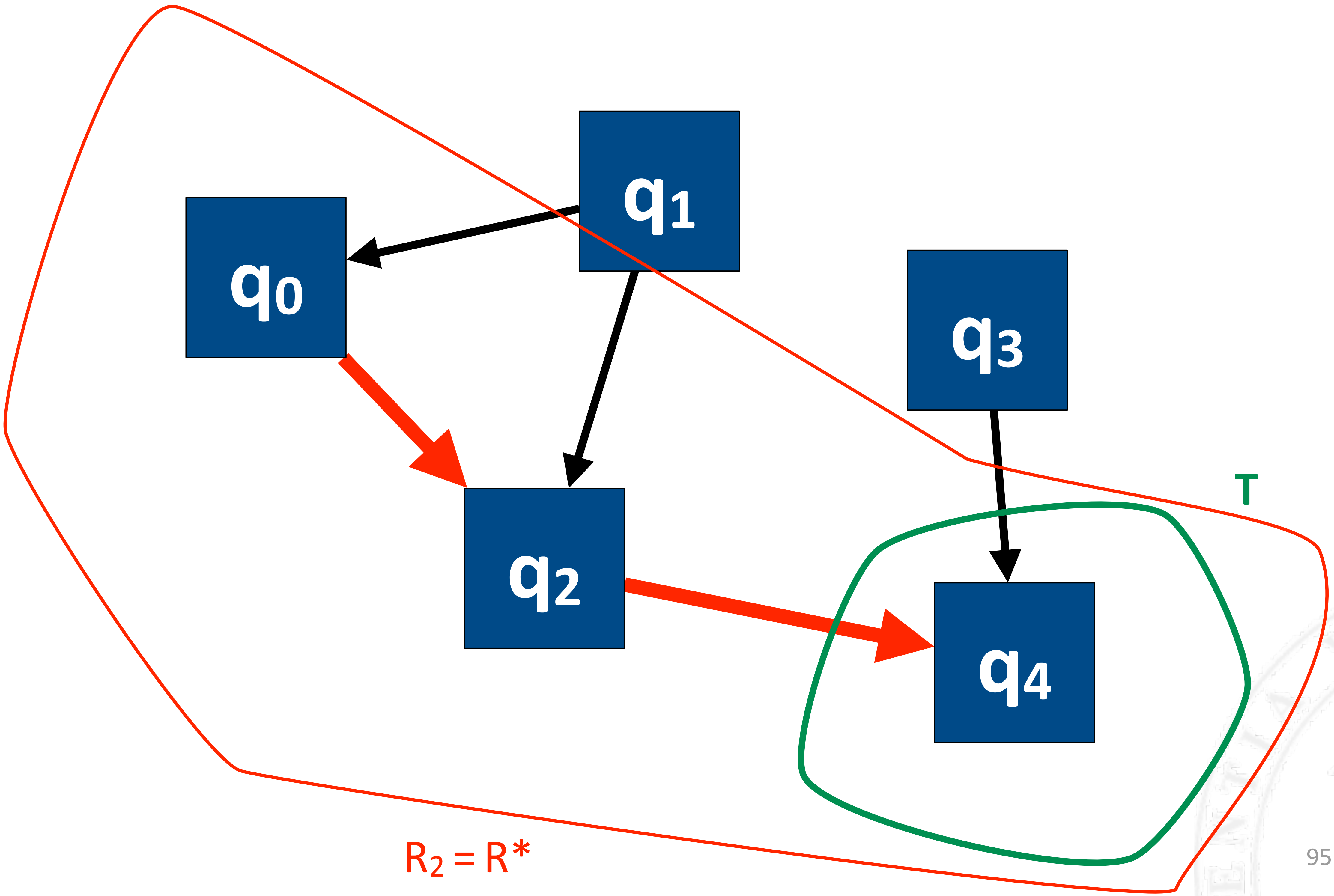
Drawback of the forward approach

- Unfortunately, this technique does not allow us to characterise **W_A**
- In the previous example, **all nodes are winning**, but we only compute those that are **reachable** from q_0

ULB

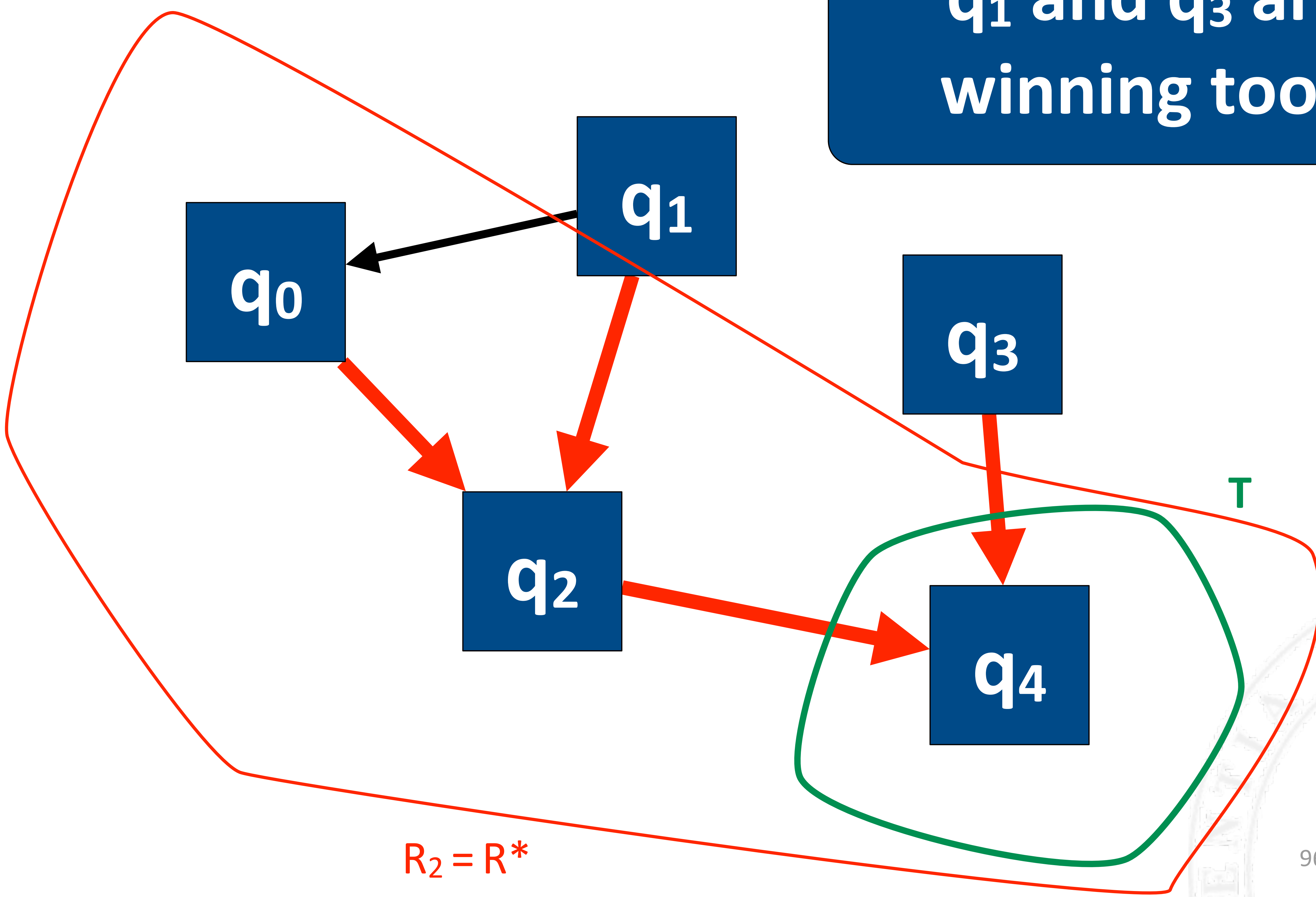
UNIVERSITÉ LIBRE DE BRUXELLES

Reachability in 1-player games



Reachability in 1-player games

q_1 and q_3 are winning too !



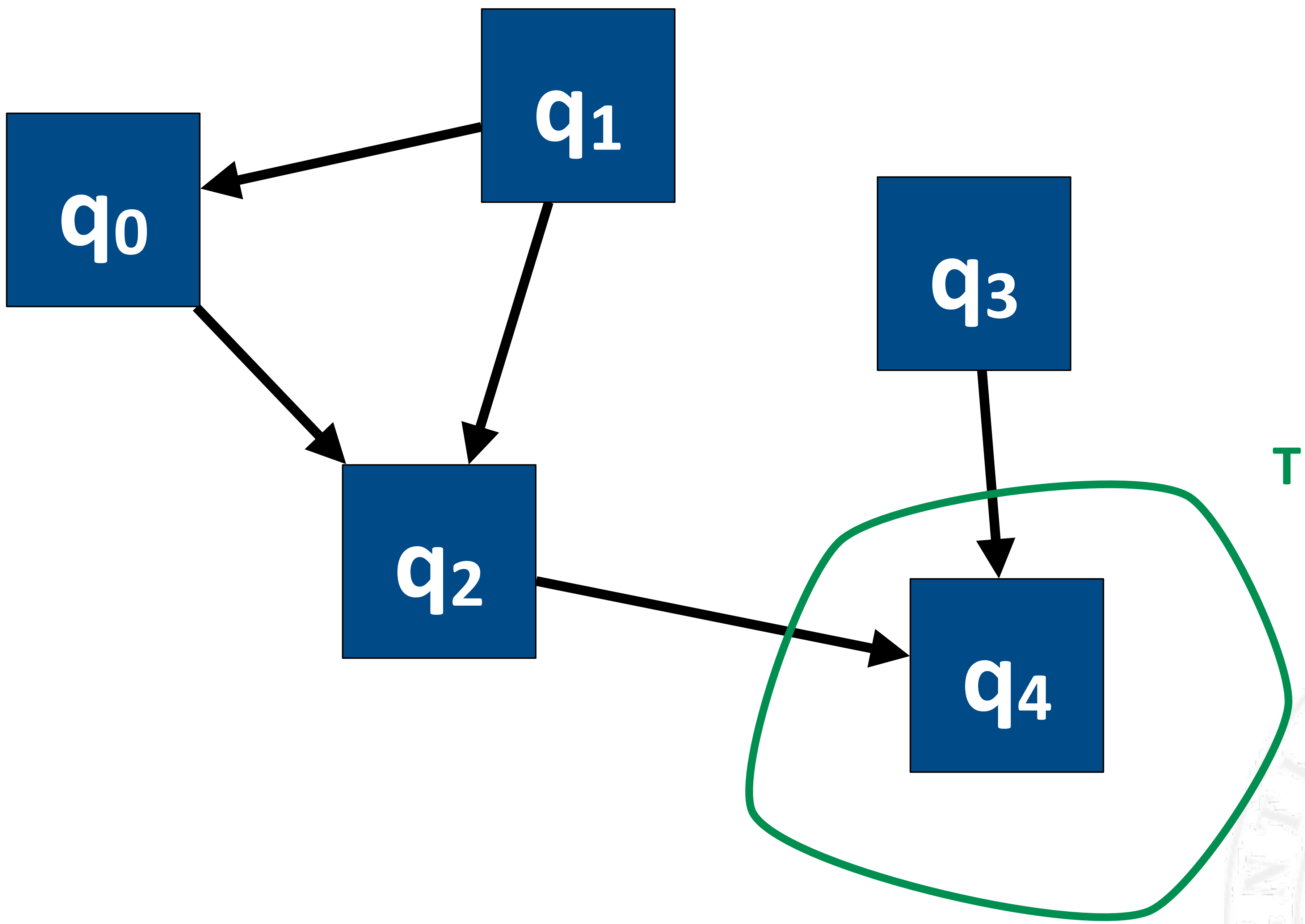
Backward approach

- Instead of computing the nodes reachable from the initial one, we compute the nodes that re **co-reachable from the target.**
  - It is a **backward** approach
- We compute the sequence **B_i:**
  - $B_0 = T$
  - $B_{i+1} = B_i \cup \{q \mid \exists\, q' \in B_i : (q,q') \in E\}$

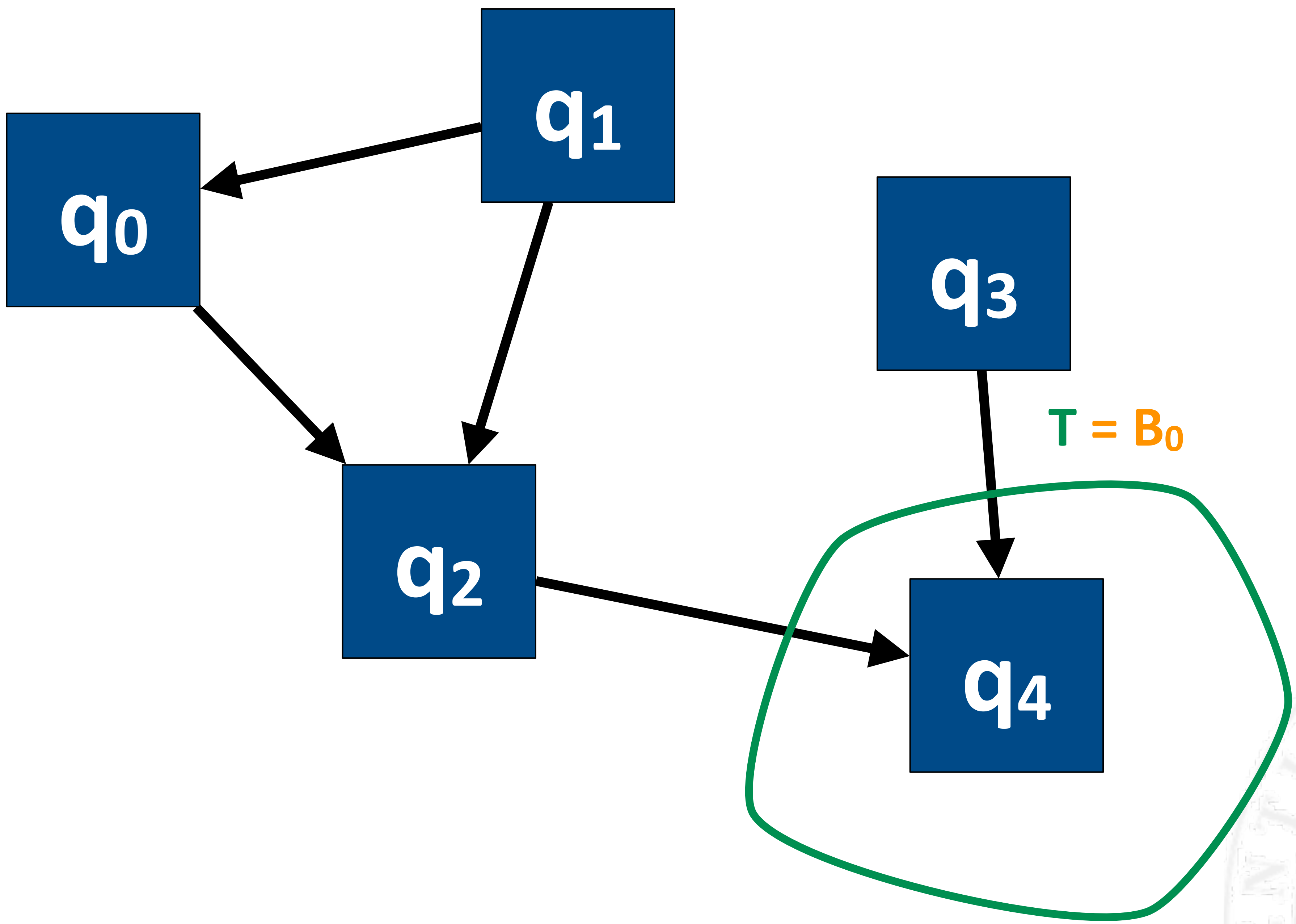
Backward approach

- Intuitively, **B_i** is the set of all nodes that can **reach the target within i steps.**
- This sequence **eventually stabilises**
  - Prove it !
  - Let **B*** denote the set obtained at stabilisation
- Then, player A has a strategy to reach **T** from any node $q \in$ **B***, and from those nodes only.

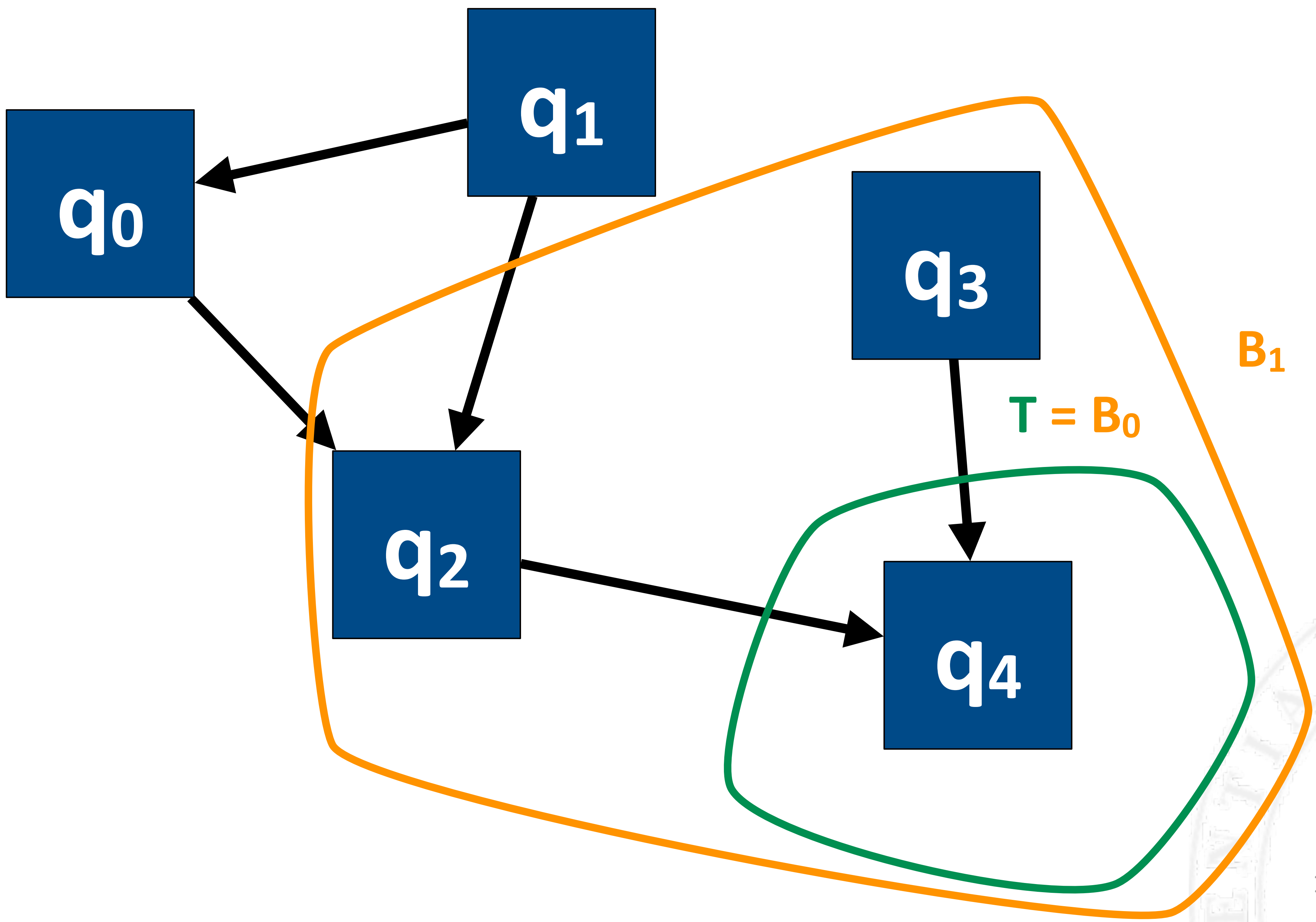
Reachability in 1-player games



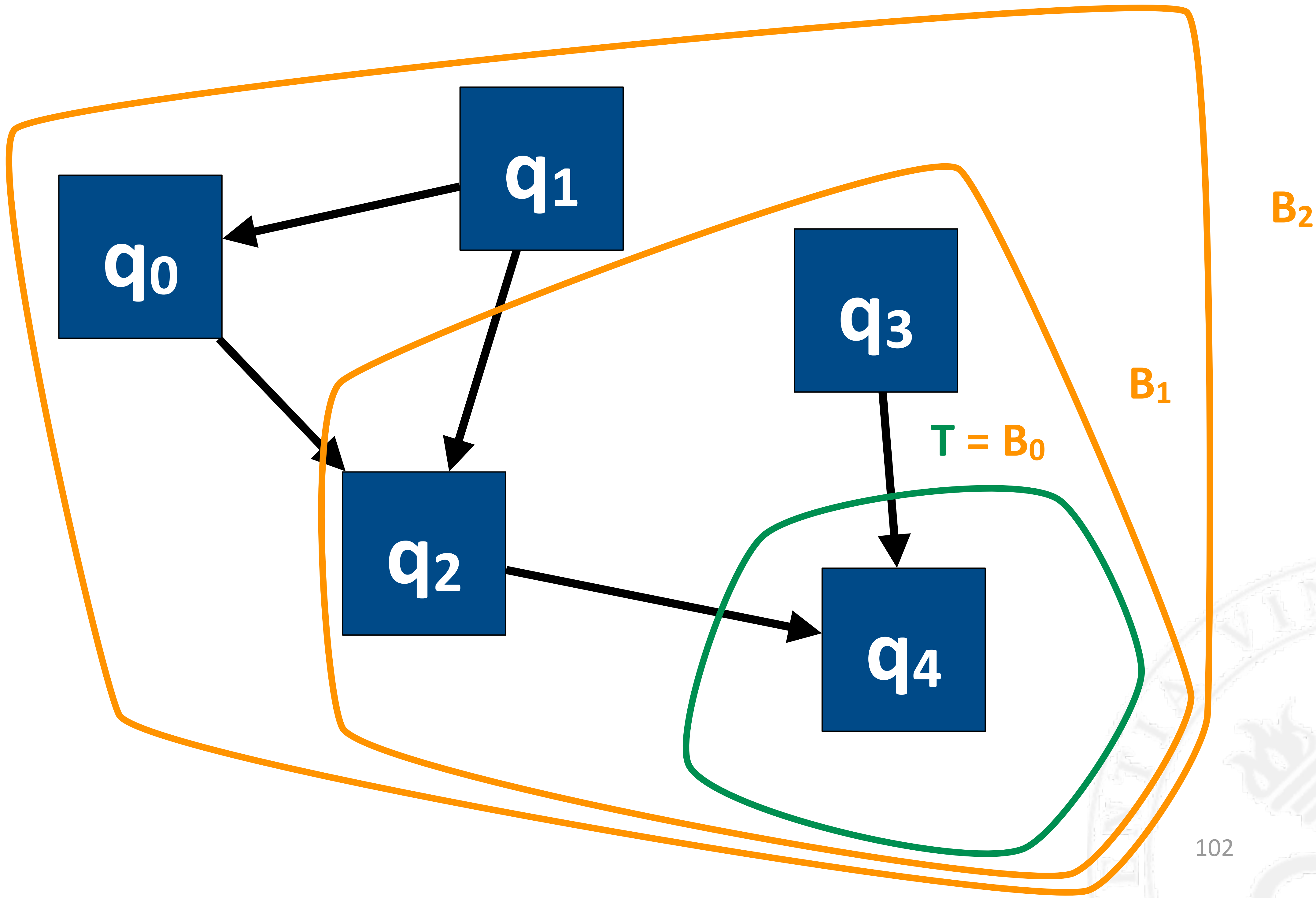
Reachability in 1-player games



Reachability in 1-player games



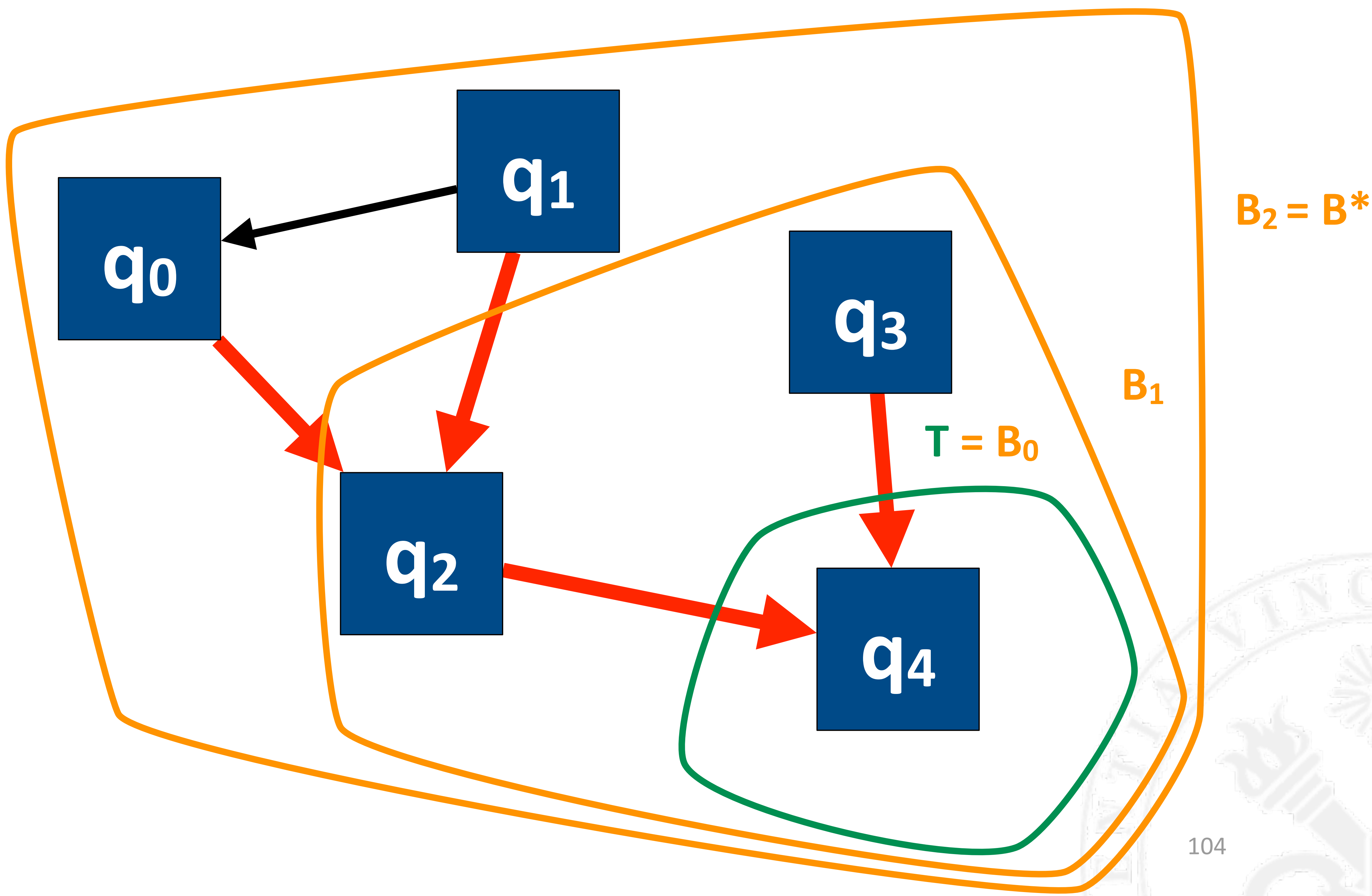
Reachability in 1-player games



Backward approach

- Again, the strategy can be **extracted** from the sequence B_0, B_1,...
- It is also a **positional strategy**.

Reachability in 1-player games



Reachability games

- **Theorem**: reachability games are **positionally determined**.
  - «positionally» means that positional strategies suffices for each player
  - Thus, the set of nodes Q can be **partitioned** into W_A and W_B s.t.
    - from each node in W_A, player A has a positional strategy that guarantees to **eventually reach T** and
    - from each node in WB, player B has a positional winning strategy that guarantees **never to visit T**

Attractor set

- Let us now adapt the idea of the **backward algorithm** to cope with the **interaction** with the second player
- We will compute a sequence of sets A_i s.t. from any node in A_i, the player can **force the game to eventually visit the target** within at most i moves.

Attractor

- **Definition**: For a set T of locations and a player X, the **attractor** of T for X $Attr^X(T)$ is the set of locations from where X can force the game to reach T
- From those nodes, X has a winning strategy for the objective «reach T»
- **Definition**: $Attr^X_i(T)$ is the set of locations from where X can force the game to reach T in at most i steps

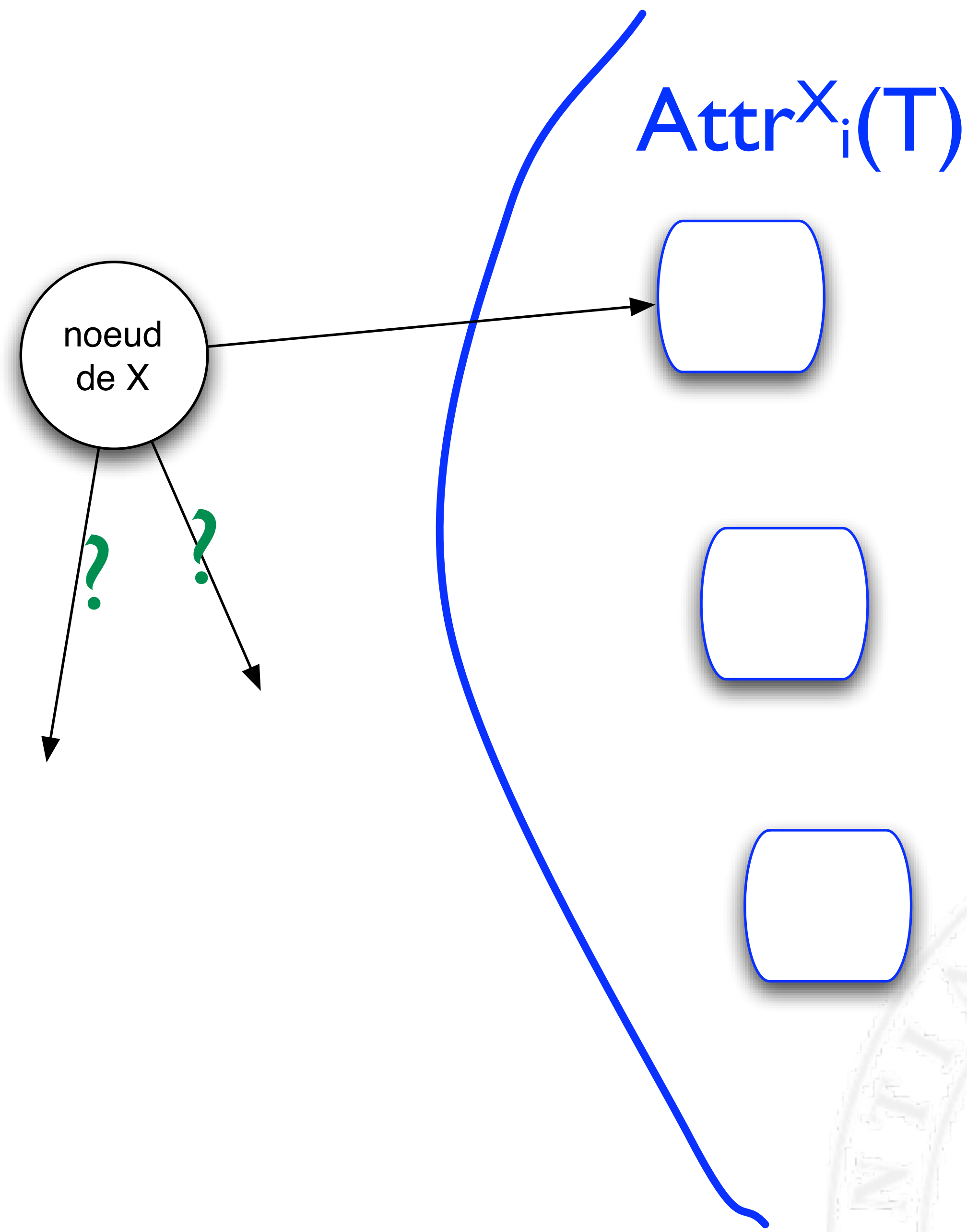
Attractor

- **Definition**: $Attr^X_i(T)$ is the set of locations from where X can force the game to reach T in at most i steps
- Clearly $Attr^X_0(T) = T$.
- How can we compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?
- Clearly $Attr^X_i(T) \subseteq Attr^X_{i+1}(T)$, but what are the locations that should be **added** ?

Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Case 1:

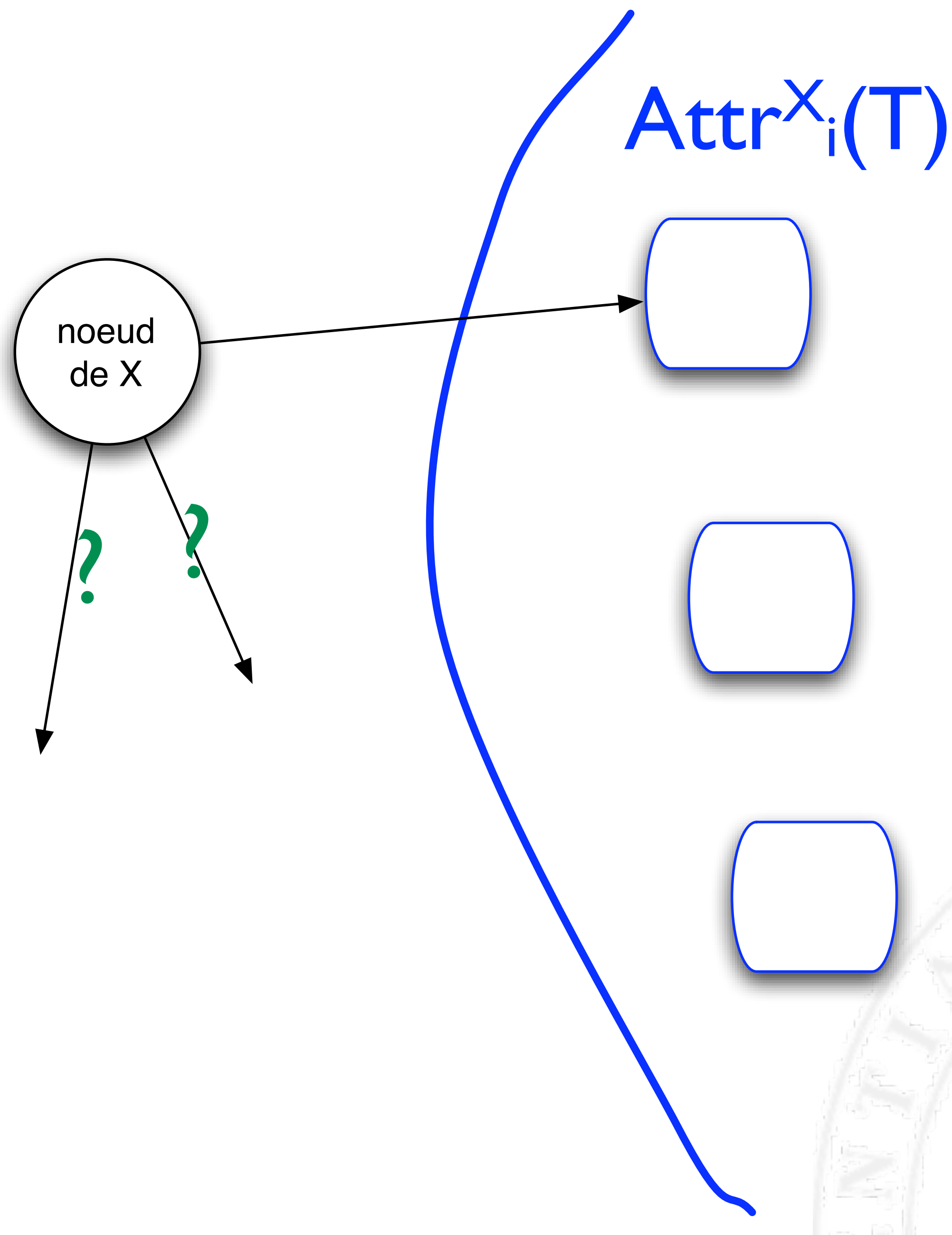


Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Case 1:

Since X defines the strategy, it can always choose to go to $Attr^X_i(T)$

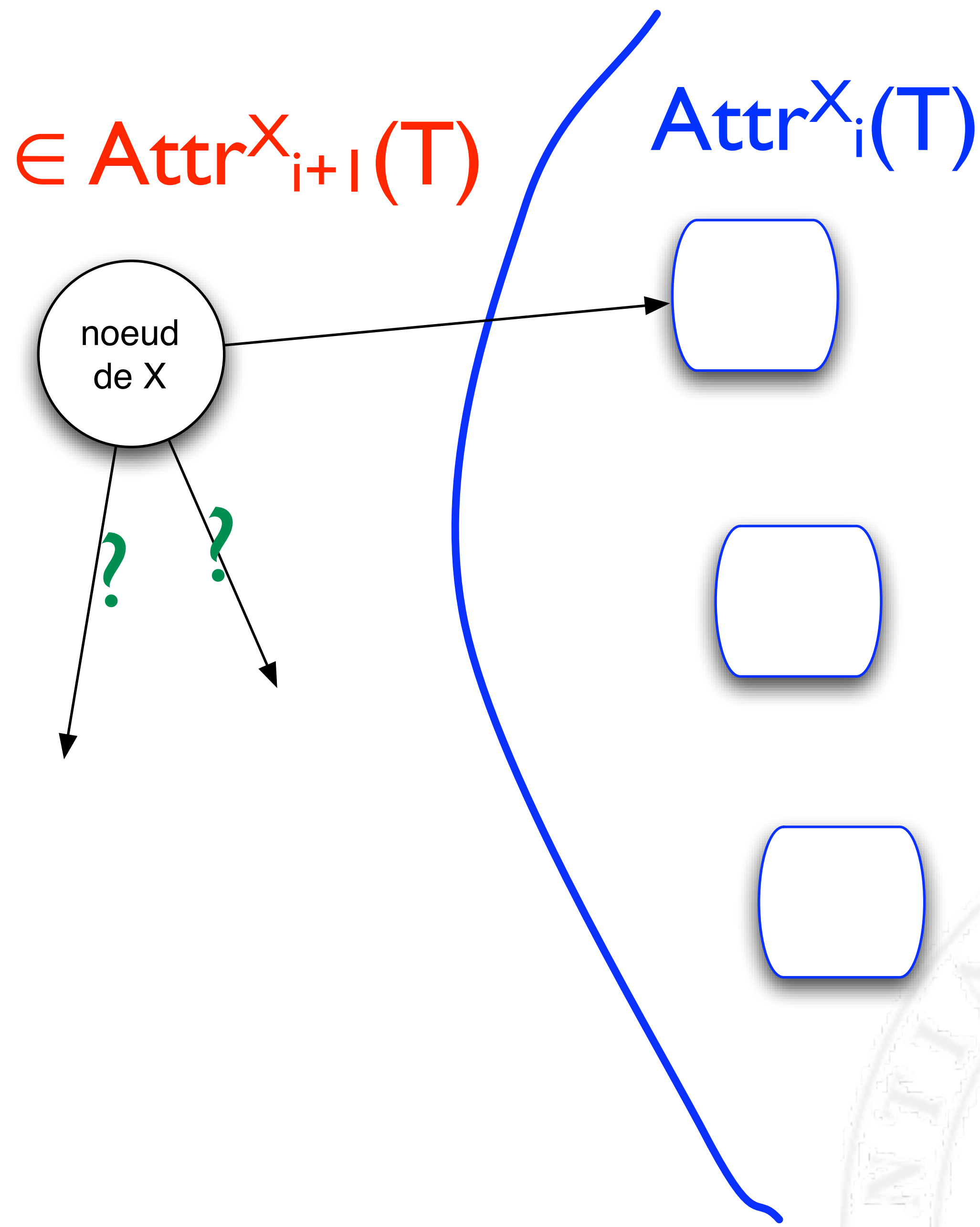


Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Case 1:

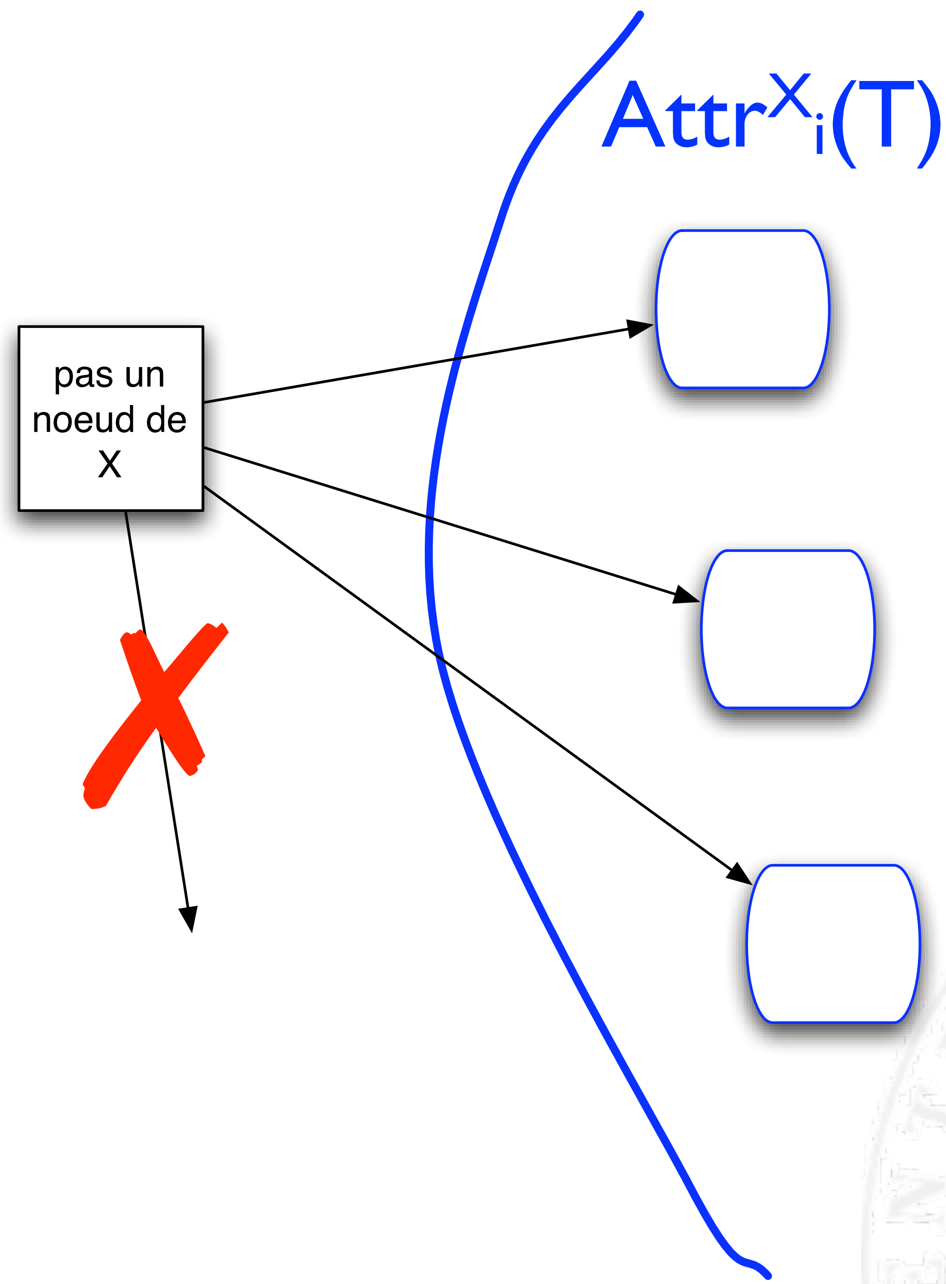
Since X defines the strategy, it can always choose to go to $Attr^X_i(T)$



Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Case 2:

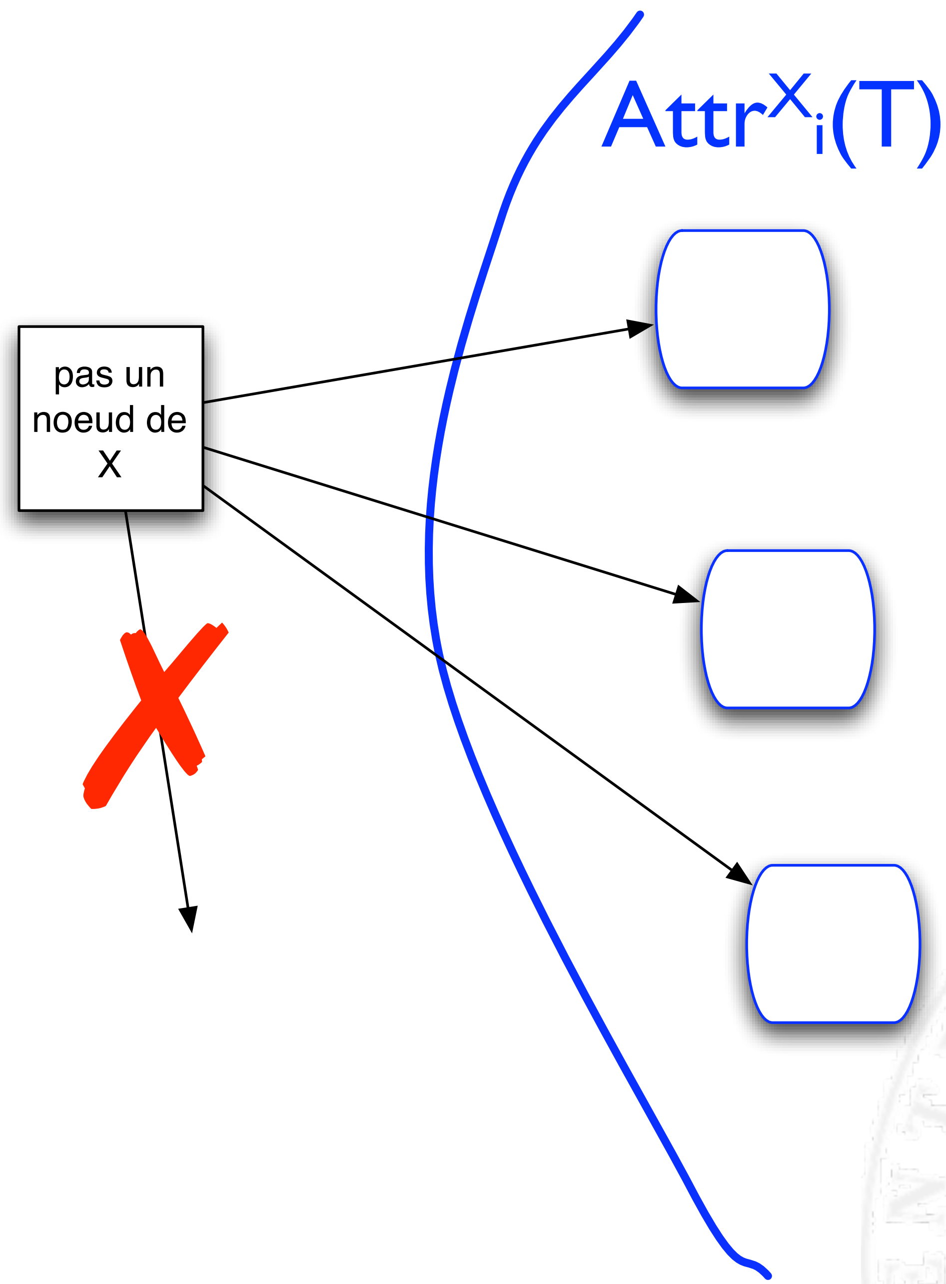


Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Case 2:

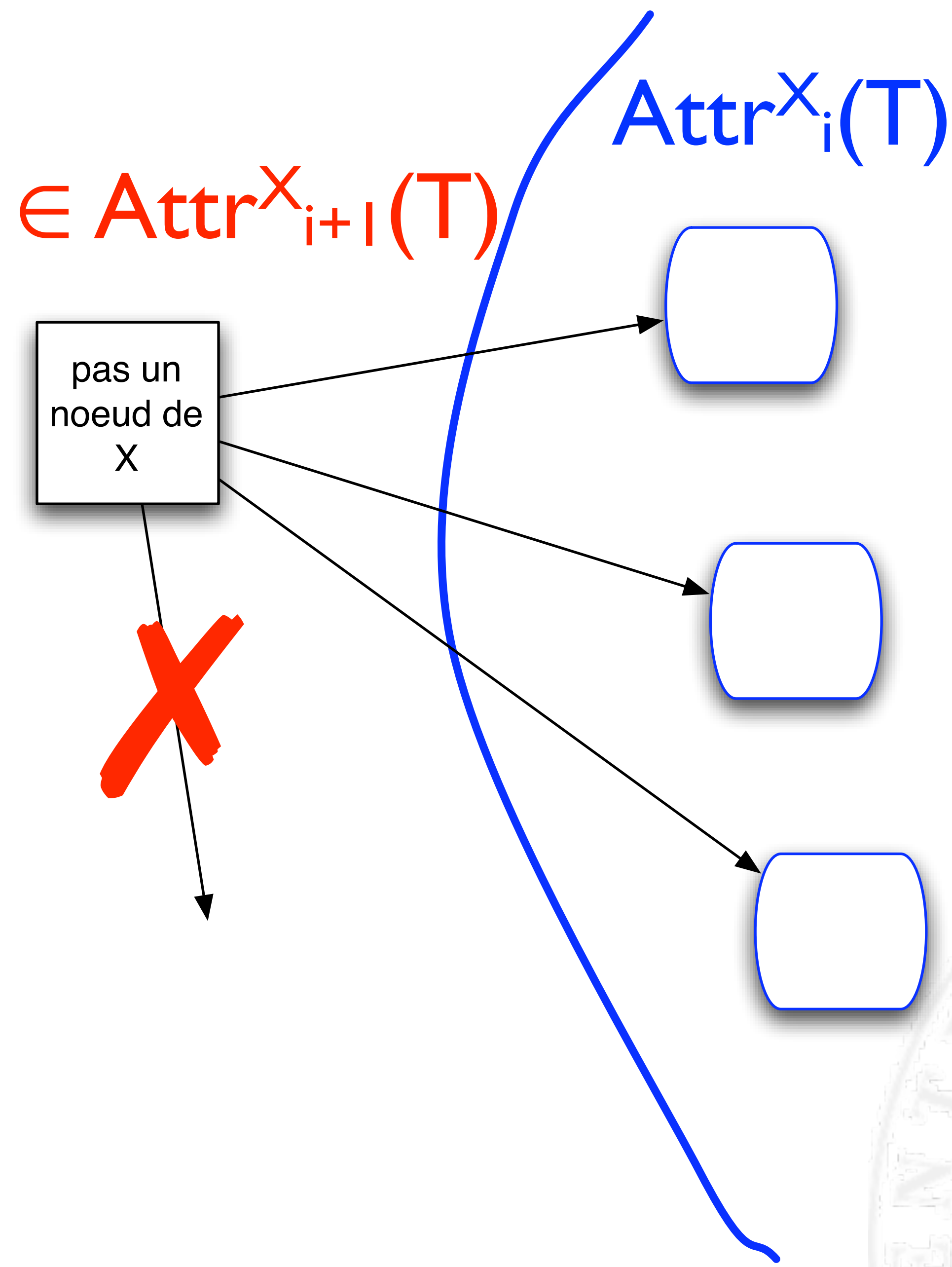
The adversary can choose nodes in $Attr_X^i(T)$ only



Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Case 2:

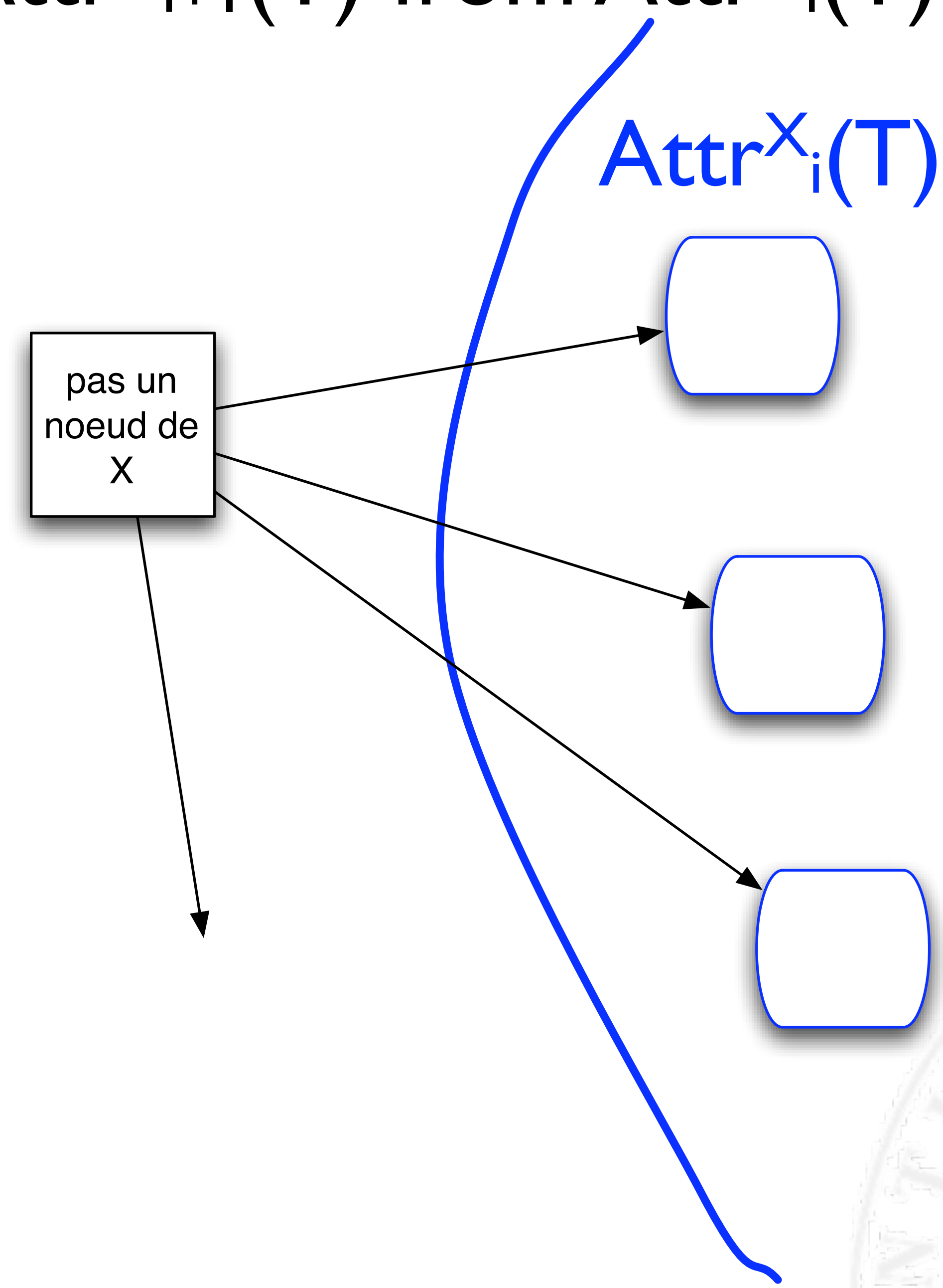


The adversary can choose nodes in $Attr_X^i(T)$ only

Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Cas 3:

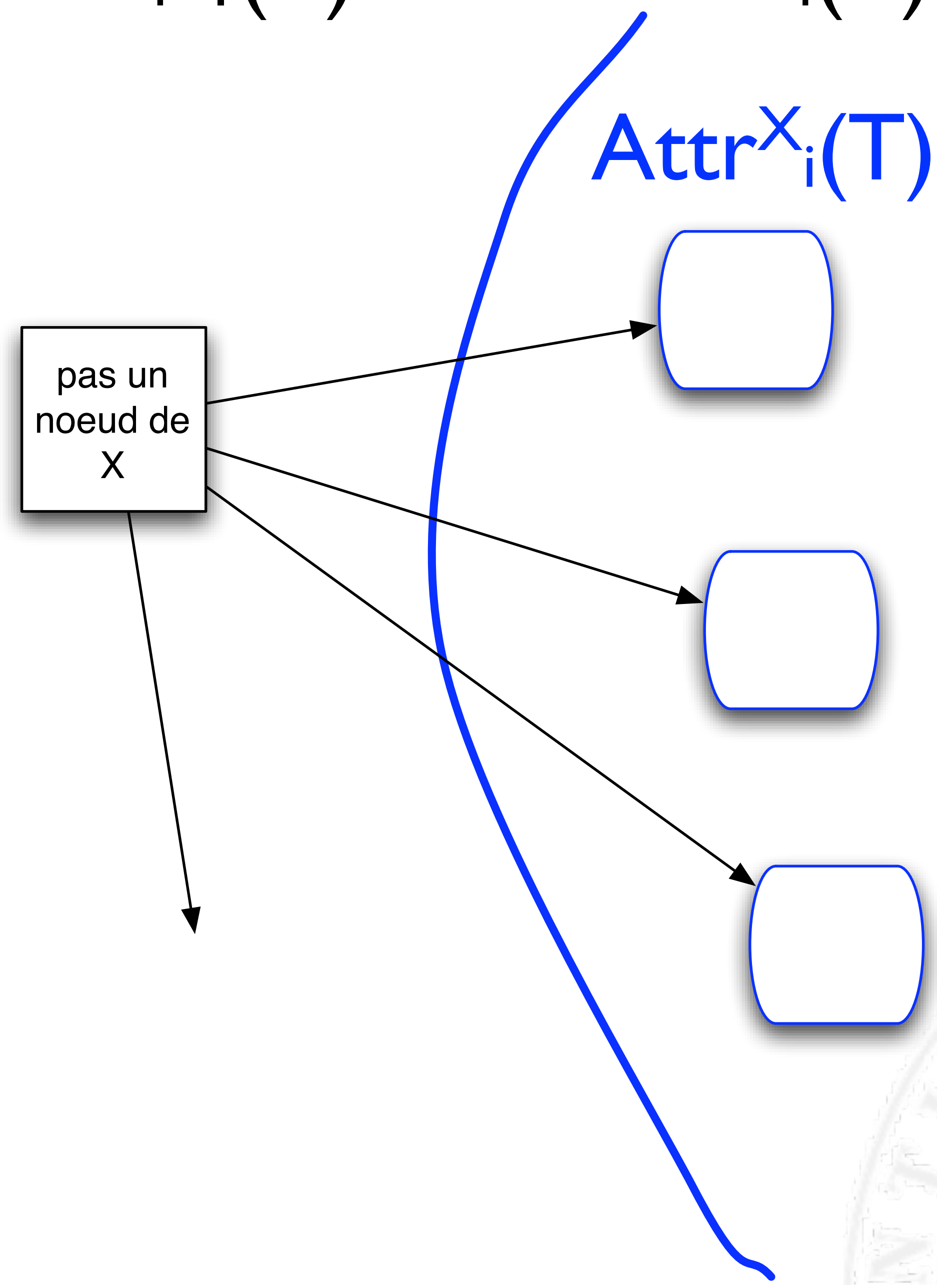


Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Cas 3:

The adversary can choose a successor outside $Attr^X_i(T)$

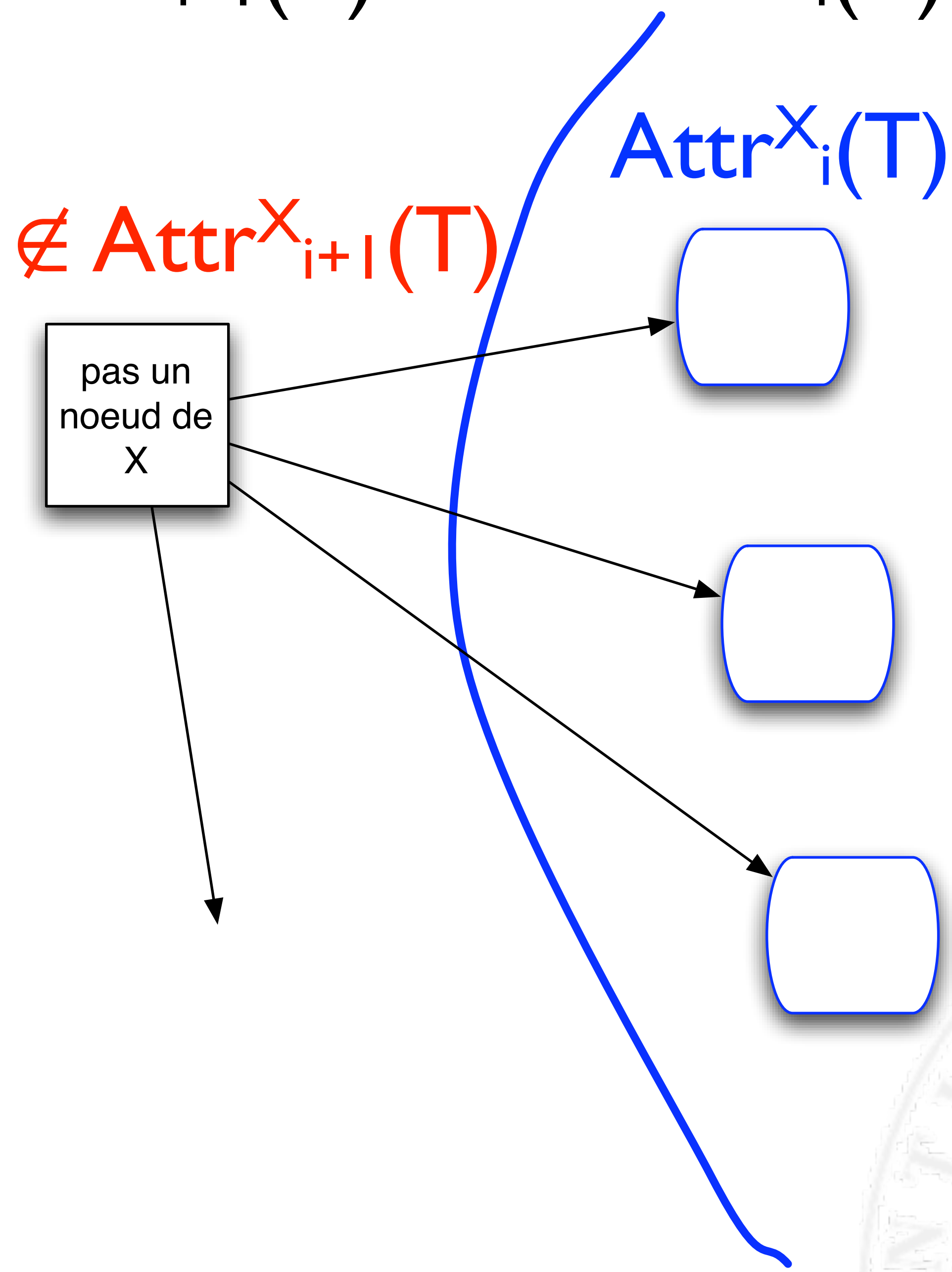


Attractor

How to compute $Attr^X_{i+1}(T)$ from $Attr^X_i(T)$?

Cas 3:

The adversary can choose a successor outside $Attr^X_i(T)$



Attractor

- Thus:

$$\mathbf{Attr}^X_0(\mathbf{T}) = \mathbf{T}$$

$$\mathbf{Attr}^X_{i+1}(\mathbf{T}) = \mathbf{Attr}^X_i(\mathbf{T})$$

$$\cup \{q \in Q_x \mid \exists (q,r) \in E : r \in \mathbf{Attr}^X_i(\mathbf{T})\}$$

$$\cup \{q \in Q \setminus Q_x \mid \forall (q,r) \in E : r \in \mathbf{Attr}^X_i(\mathbf{T})\}$$

But this is an infinite sequence of sets !

Attractor

$Attr^X{}_0(T) \subseteq Attr^X{}_1(T) \subseteq Attr^X{}_2(T) \subseteq \ldots \subseteq Attr^X{}_k(T) \ldots$

- **<u>Theorem</u>**: The sequence $Attr^X{}_i(T)$ **converges**
- **<u>Proof</u>**: The sequence is **increasing**, and each $Attr^X{}_i(\mathbf{T})$ is included in Q, which is finite.
- Let us thus consider the first position k s.t. $Attr^X{}_k(\mathbf{T}) = Attr^X{}_{k+1}(\mathbf{T})$
- We have: $Attr^X(\mathbf{T}) = Attr^X{}_k(\mathbf{T})$

Attractor

$Attr^X_0(T)$ $Attr^X_1(T)$ $Attr^X_2(T)$ … $Attr^X_k(T)$ …

- **Theorem**: The sequence $Attr^X_i(T)$ **converges**
- **Proof**: The sequence is **increasing**, and each $Attr^X_i(T)$ is included in Q, which is finite.
- Let us thus consider the first position k s.t. $Attr^X_k(T) = Attr^X_{k+1}(T)$
- We have: $Attr^X(T) = Attr^X_k(T)$

Attractor

$$Attr^X_0(T) \subset Attr^X_1(T) \subset Attr^X_2(T) \subset \ldots \subset Attr^X_k(T) = \ldots$$

- **Theorem**: The sequence $Attr^X_i(T)$ **converges**
- **Proof**: The sequence is **increasing**, and each $Attr^X_i(T)$ is included in Q, which is finite.
- Let us thus consider the first position k s.t. $Attr^X_k(T) = Attr^X_{k+1}(T)$
- We have: $Attr^X(T) = Attr^X_k(T)$

Attractor

$Attr^X_0(T) \subset \quad Attr^X_1(T) \subset \quad Attr^X_2(T) \subset \cdots \quad \subset \quad Attr^X_k(T) \doteqdot$

- **Theorem**: $W_X = Attr^X(T)$
  - –**Proof (1)**: $Attr^X(T) \subseteq W_X$ (**there are only winning locations in the attractor**)
    Clearly, we have added to $Attr^X(T)$ only winning positions for X (by def of Attr). This is thus trivial.

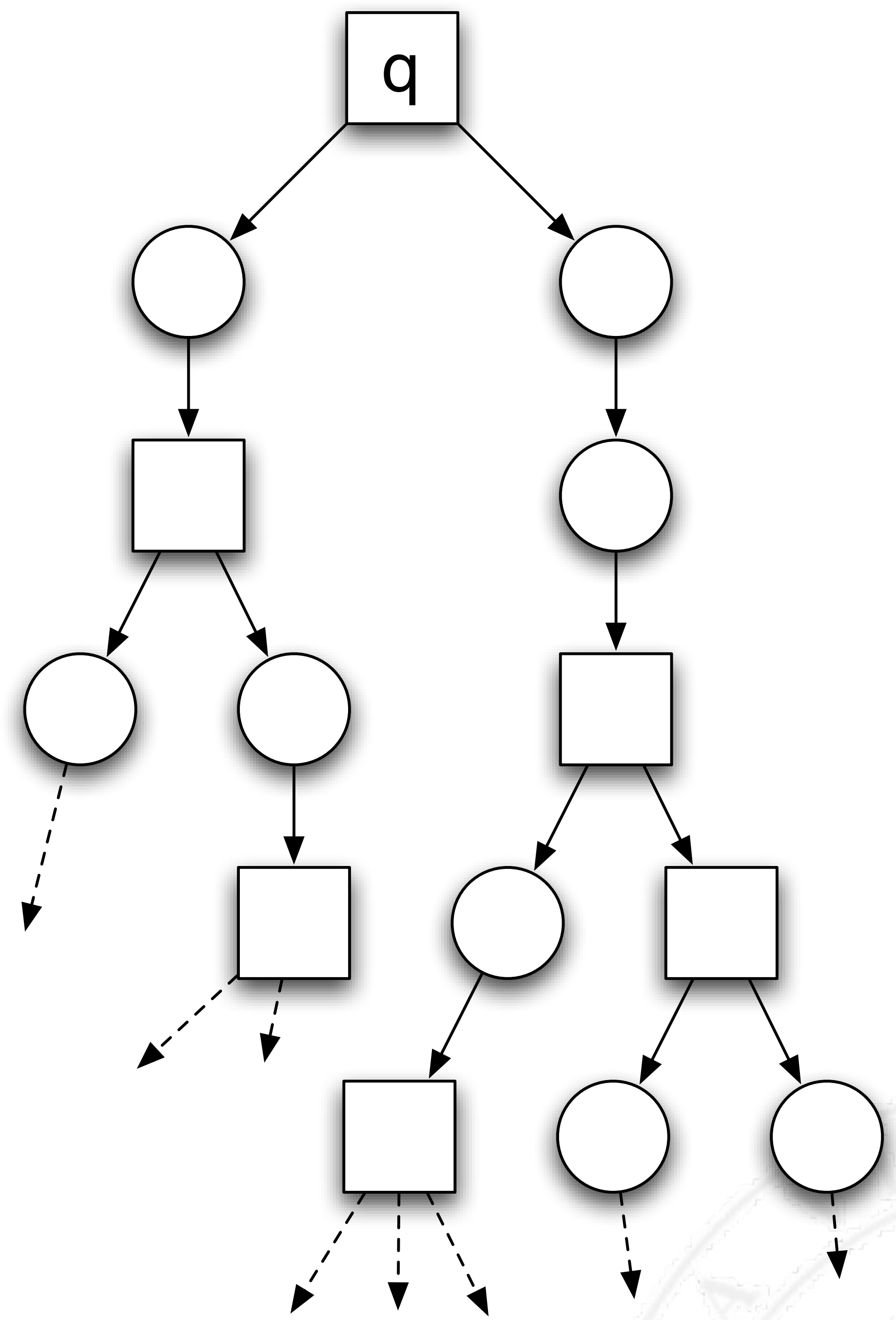
Attractor

$Attr^X_0(T) \subset \quad Attr^X_1(T) \subset \quad Attr^X_2(T) \subset \cdots \quad \subset \quad Attr^X_k(T) = \ldots$

- **Theorem**: $W_X = Attr^X(\mathbf{T})$
  - –**Proof (2)**: $Attr^X(\mathbf{T}) \supseteq W_X$ (All the wining locations are in the attractor)
    - **By contradiction**: assume that some winning position q of X ($q \in W_X$) is not in $Attr^X(\mathbf{T}) = Attr^X_k(\mathbf{T})$.
    - Since $q \in W_X$, X has a **winning strategy** f
    - Let us consider the tree representing **all the possible plays** from q, following f

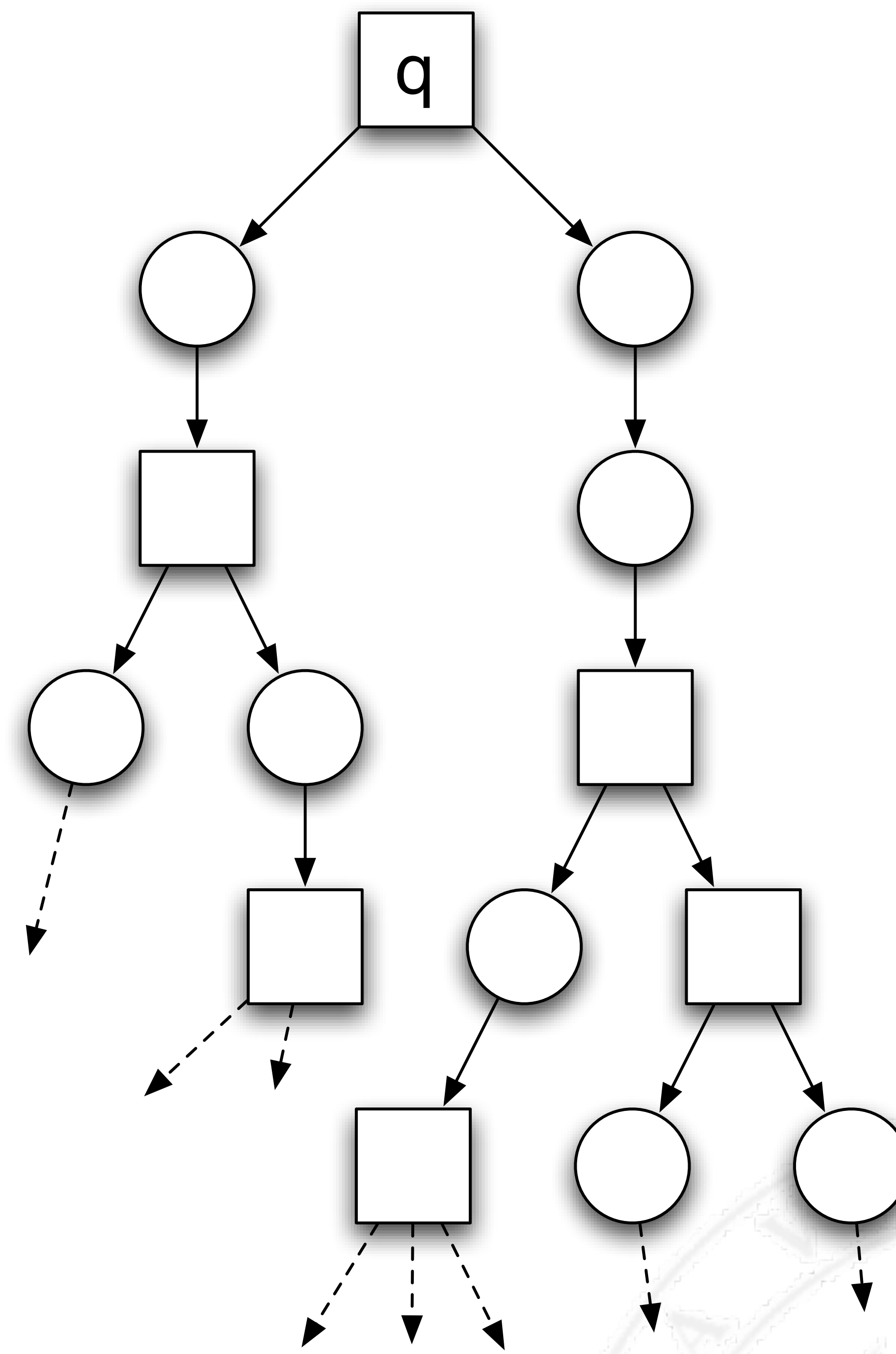
Attractor

- Each position q' of X has **one and only one son**: f(q')
- The **set of sons** of each adversary position q' is is **the set of successors** of q' in the arena
- The tree is **infinite**



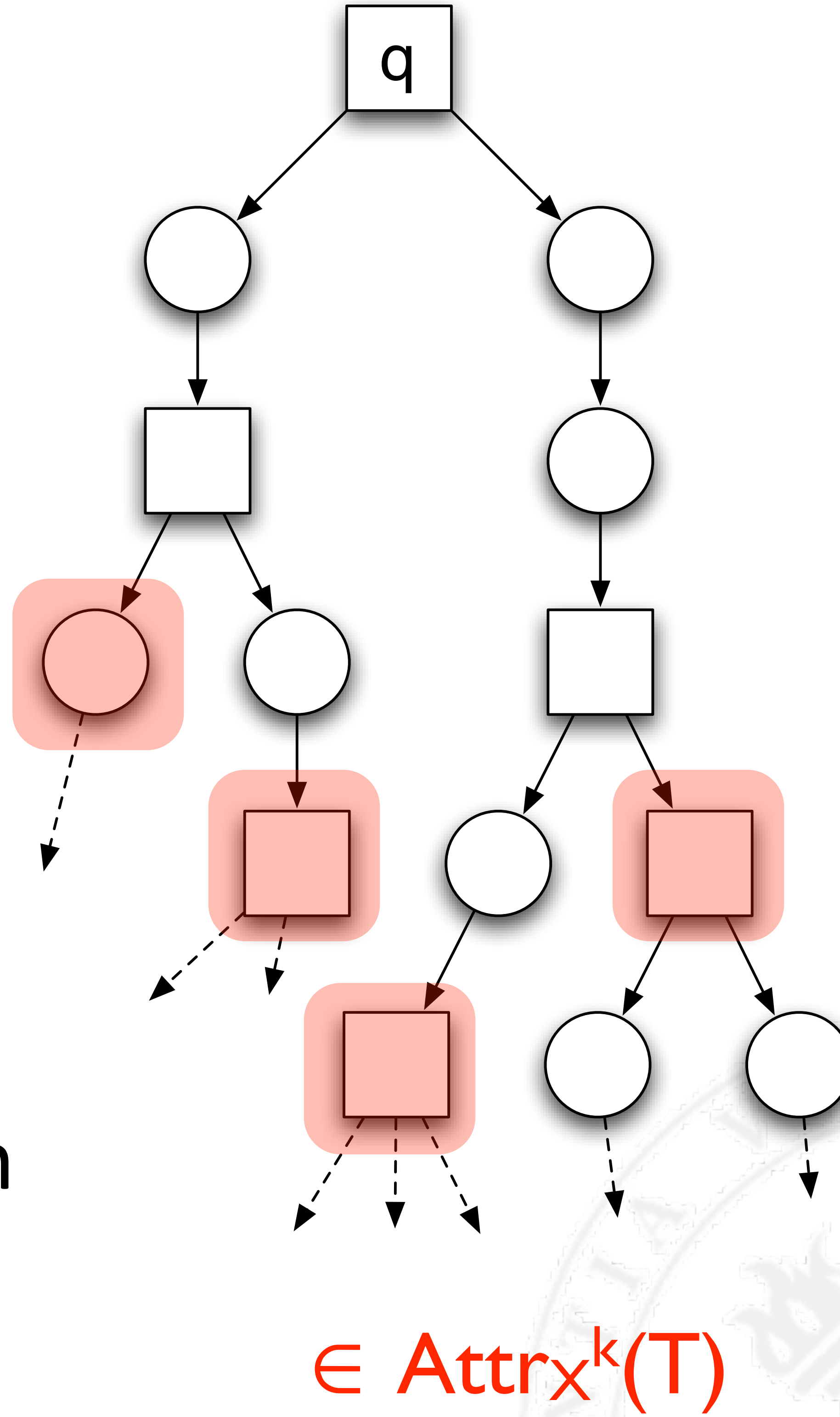
Attractor

- Each branch goes through a location $\in Attr^X_k(T)$ because:
  - f is a **winning strategy**
  - $R \subseteq Attr^X_k(T)$
- We can thus cut the tree in **two parts**:
  - The nodes **above** node in $\in Attr^X_k(T)$
  - Those **under**

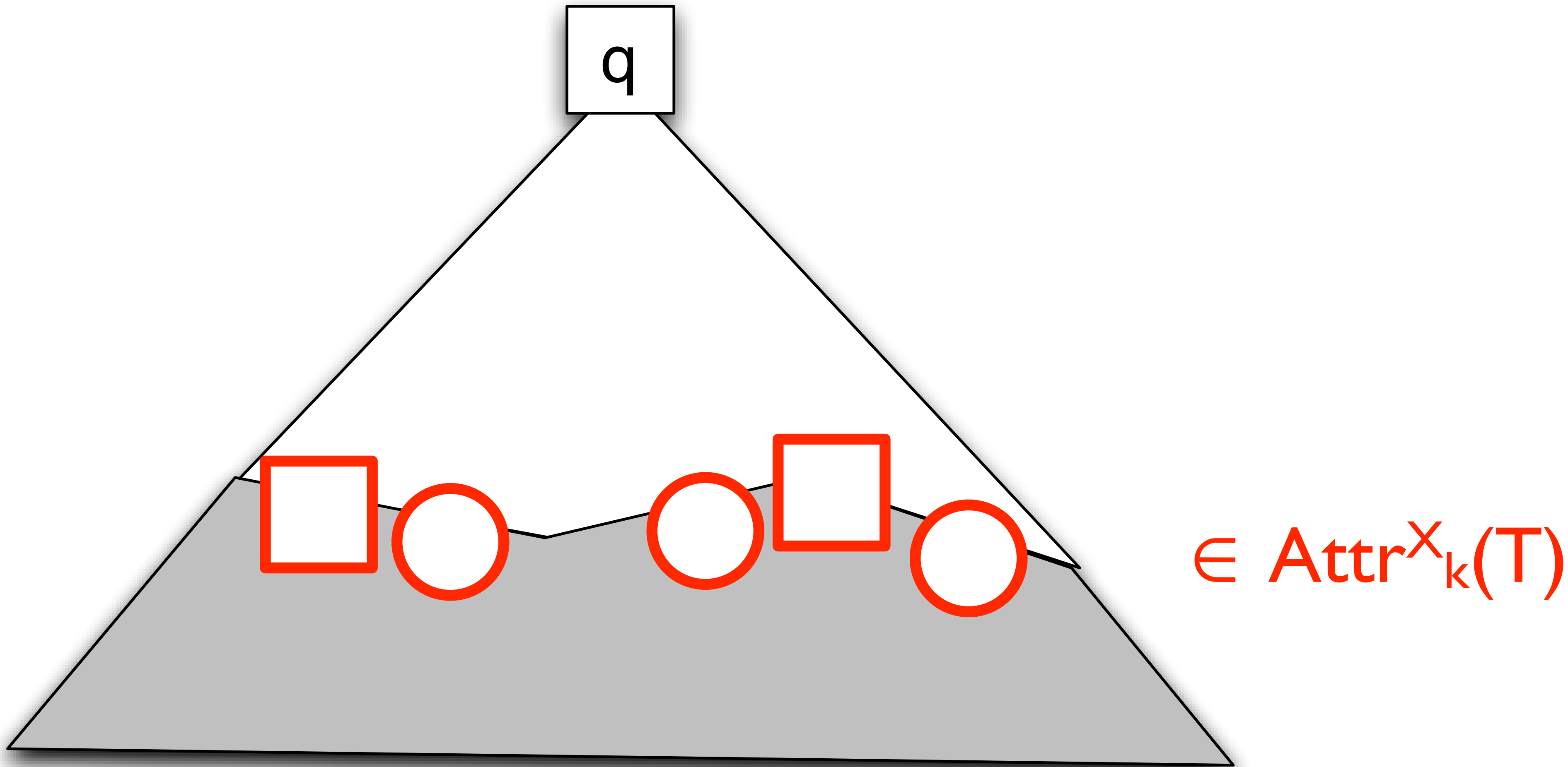


Attractor

- Each branch goes through a location ∈ $Attr^X_k(T)$ because:
  - f is a **winning strategy**
  - $R \subseteq Attr^X_k(T)$
- We can thus cut the tree in **two parts**:
  - The nodes **above** node in ∈ $Attr^X_k(T)$
  - Those **under**



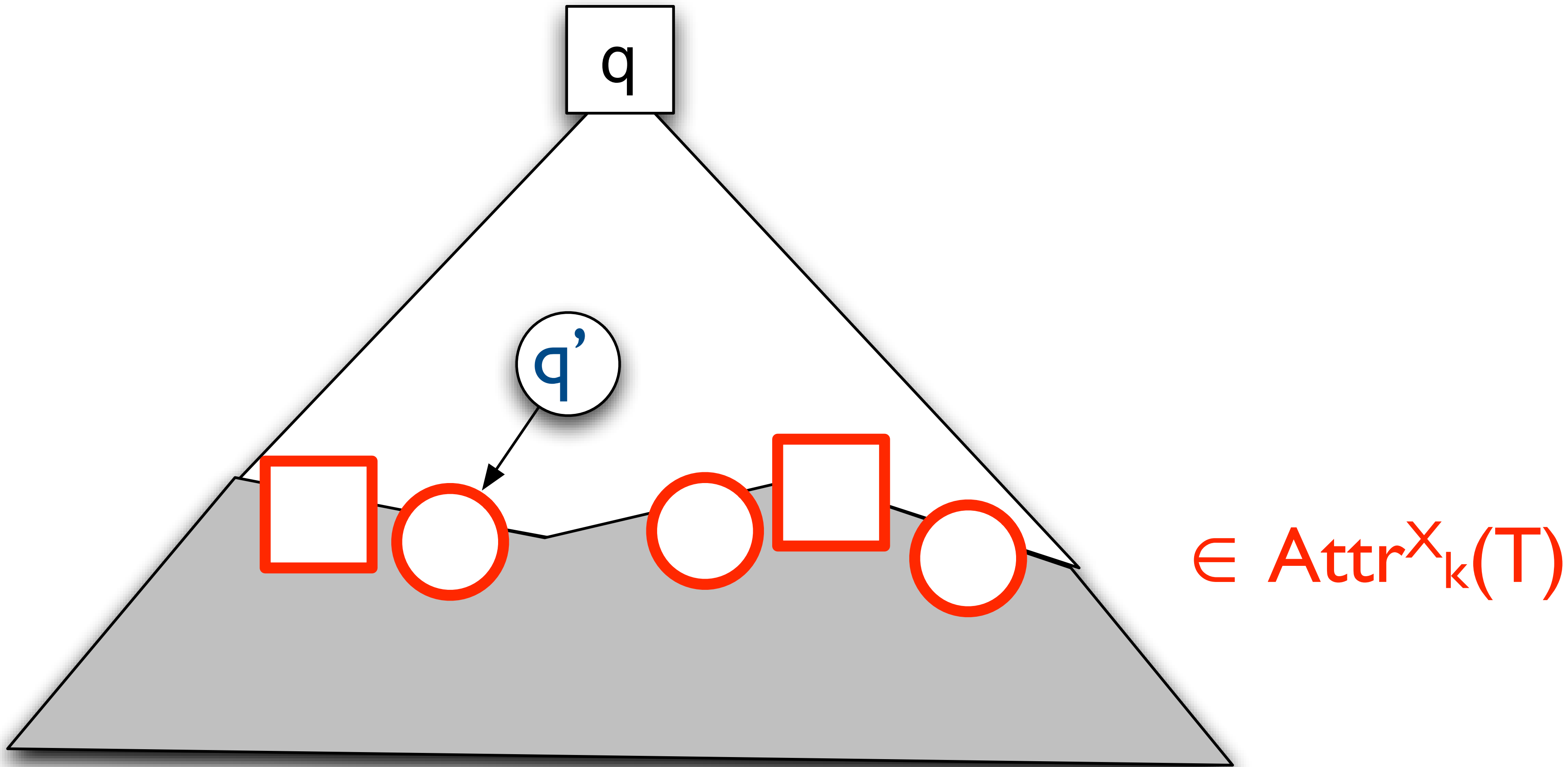
Attractor



- Let us consider the fathers of the red nodes $\in Attr^X_k(T)$.
  Those fathers $\notin Attr^X_k(T)$

- If there is a father q' of X, then q' is in $Attr^X_{k+1}(T)$. Since $q' \notin Attr^X_k(T)$, we have $Attr^X_k(T) \subset Attr^X_{k+1}(T)$. Contradiction.

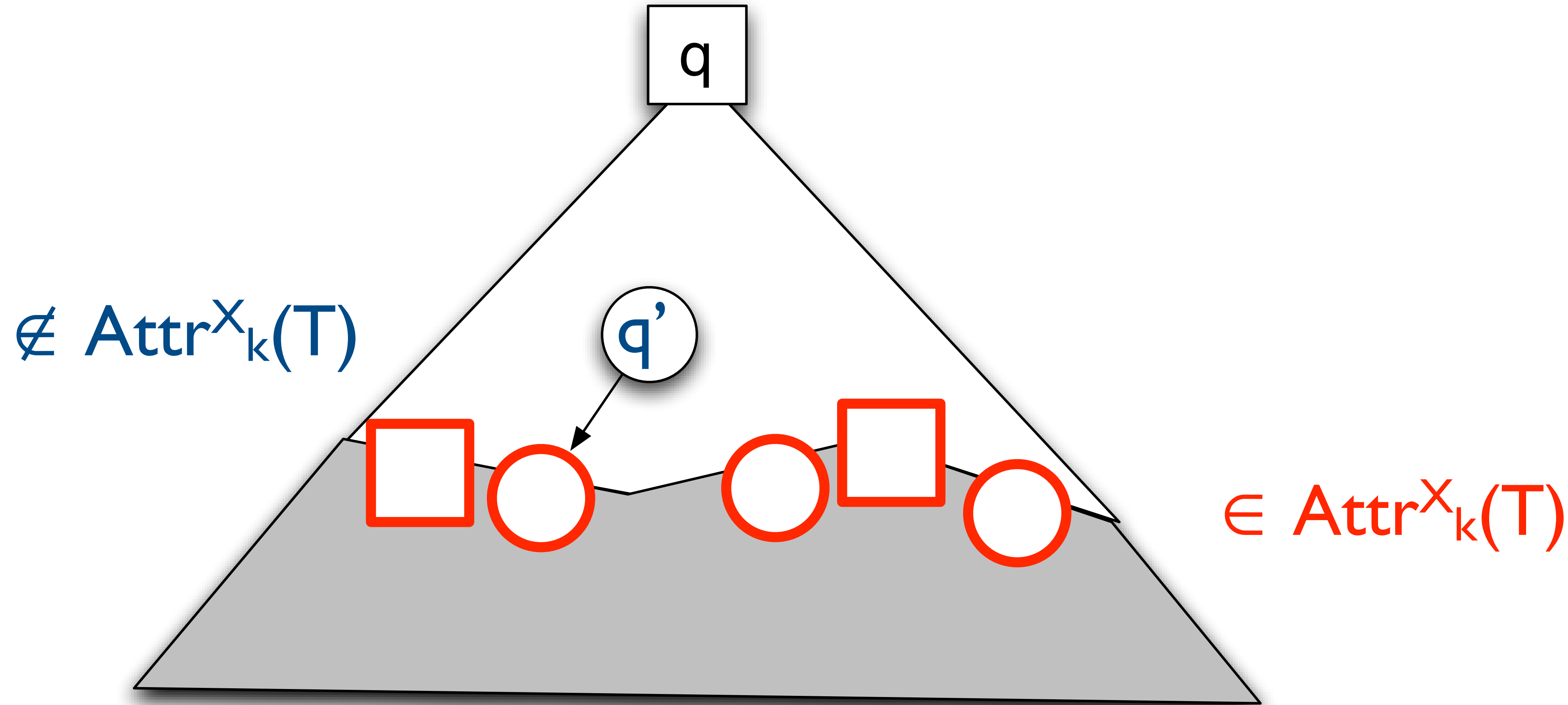
Attractor



- Let us consider the fathers of the red nodes $\in Attr^X_k(T)$.
  Those fathers $\notin Attr^X_k(T)$

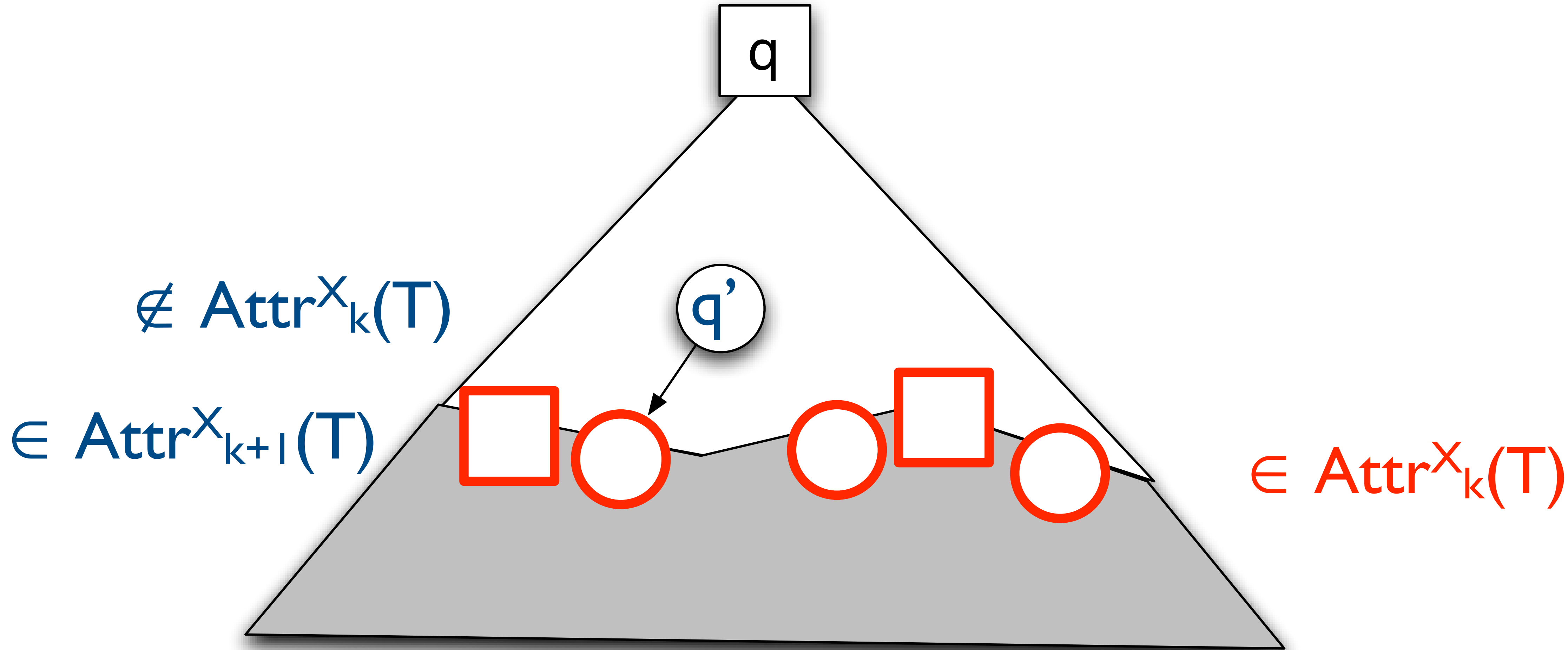
- If there is a father q' of X, then q' is in $Attr^X_{k+1}(T)$. Since $q' \notin Attr^X_k(T)$, we have $Attr^X_k(T) \subset Attr^X_{k+1}(T)$. Contradiction.

Attractor



- Let us consider the fathers of the red nodes $\in Attr^X_k(T)$. Those fathers $\notin Attr^X_k(T)$
- If there is a father q' of X, then q' is in $Attr^X_{k+1}(T)$. Since $q' \notin Attr^X_k(T)$, we have $Attr^X_k(T) \subset Attr^X_{k+1}(T)$. Contradiction.

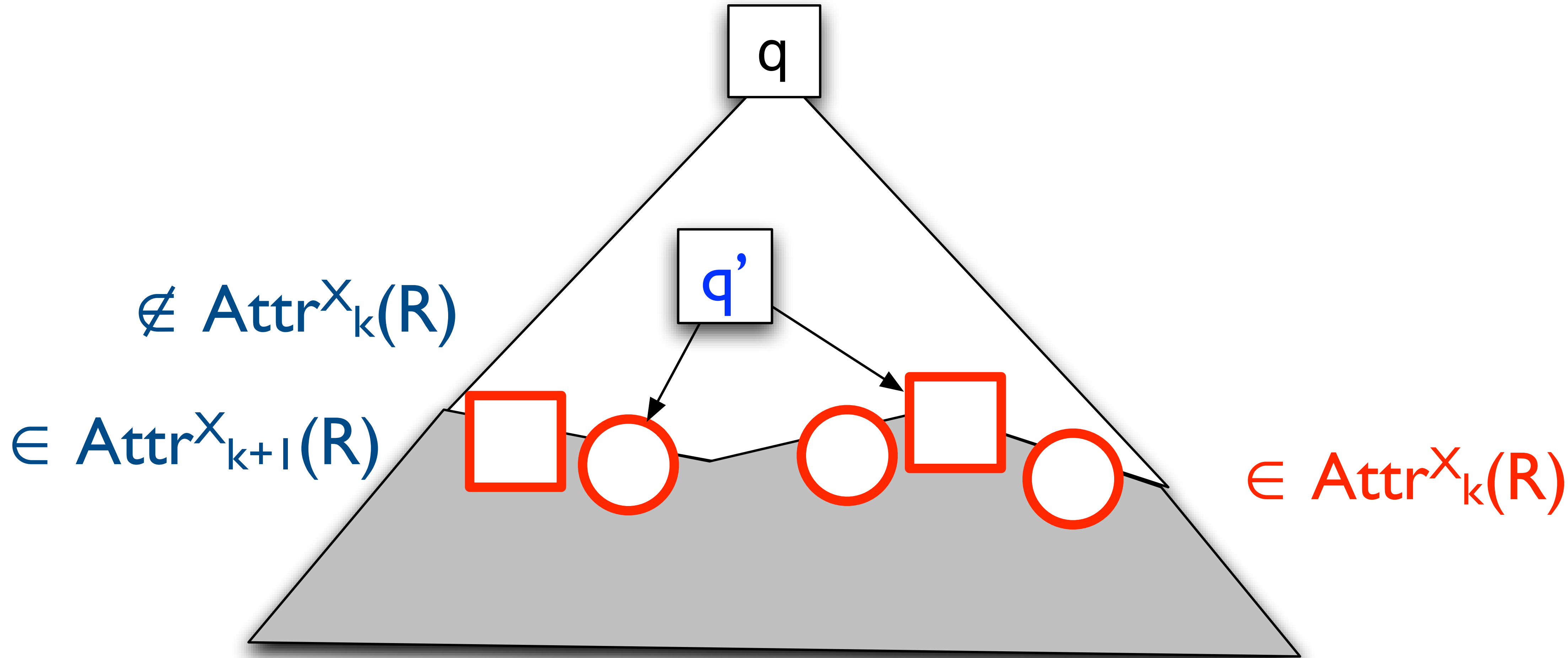
Attractor



- Let us consider the fathers of the red nodes $\in \mathrm{Attr}^X_k(T)$. Those fathers $\notin \mathrm{Attr}^X_k(T)$

- If there is a father q' of X, then q' is in $\mathrm{Attr}^X_{k+1}(T)$. Since $q' \notin \mathrm{Attr}^X_k(T)$, we have $\mathrm{Attr}^X_k(T) \subset \mathrm{Attr}^X_{k+1}(T)$. Contradiction.

Attractor



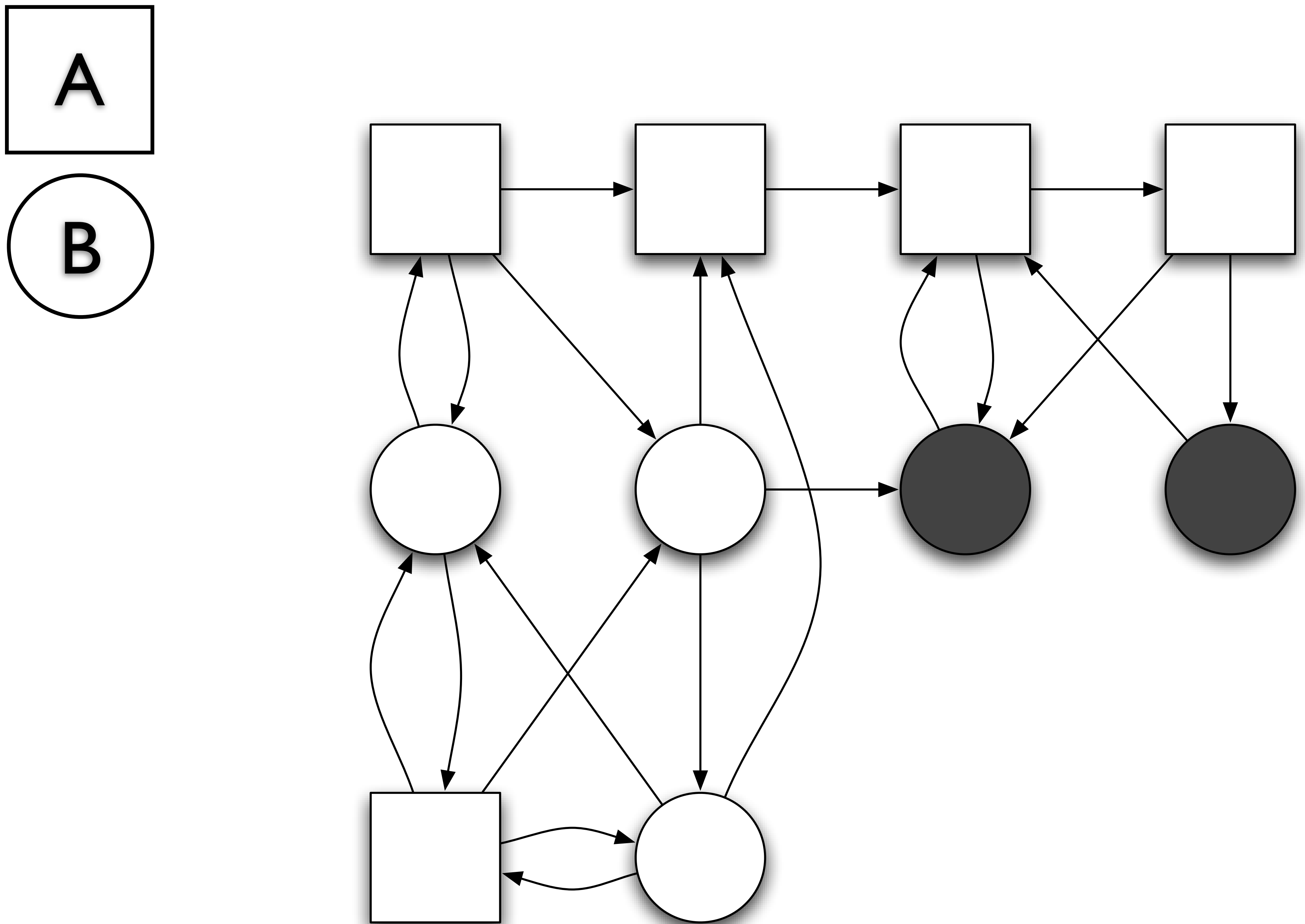
- Otherwise, all the fathers belong to the opponent and have all their sons in $Attr^X_k(R)$. They are thus in $Attr^X_{k+1}(R)$. Again, $Attr^X_k(R) \subset Attr^X_{k+1}(R)$. Contradiction.

QED

Reachability

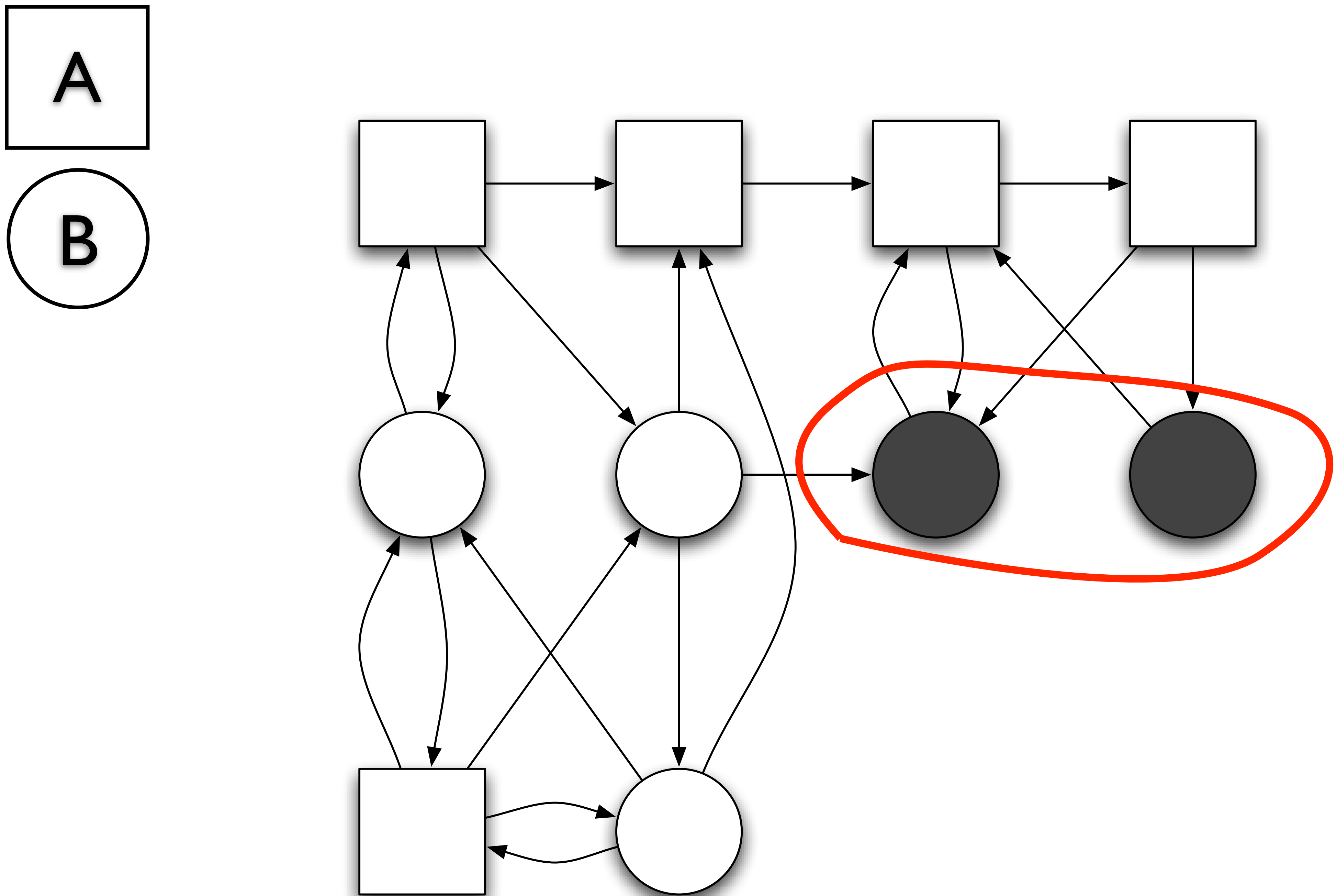
- We can now compute the set of **winning positions** W_X of a player X for reachability objective T:
  - Compute $Attr^X(T)$ (fixed point)
  - X thus has a winning strategy from q_0 iff $q_0 \in W_X$
- How to **compute** that strategy ?
  - For all position $q \in W_X$, we choose f(q) among the successors of q that are «one step closer in the attractor».
  - The fixed point characterises a family of **positional strategies**.

Example



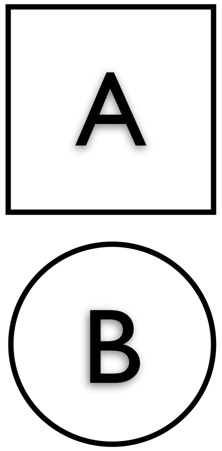
Grey nodes are the objective for B

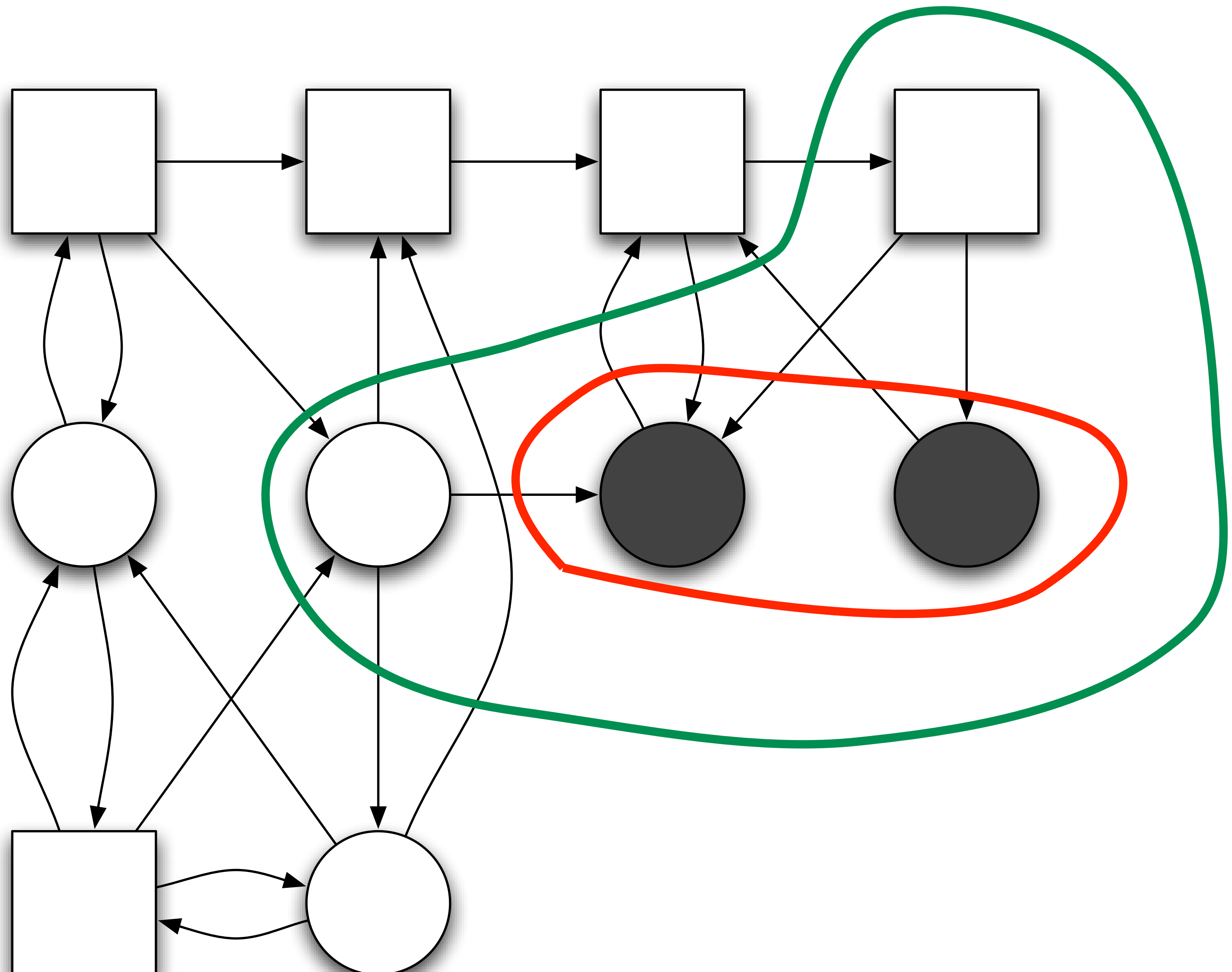
Example



Grey nodes are the objective for B

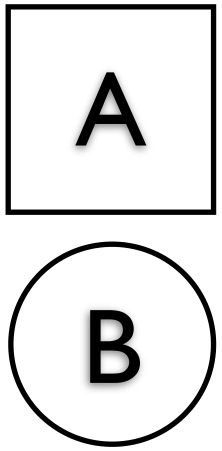
Example

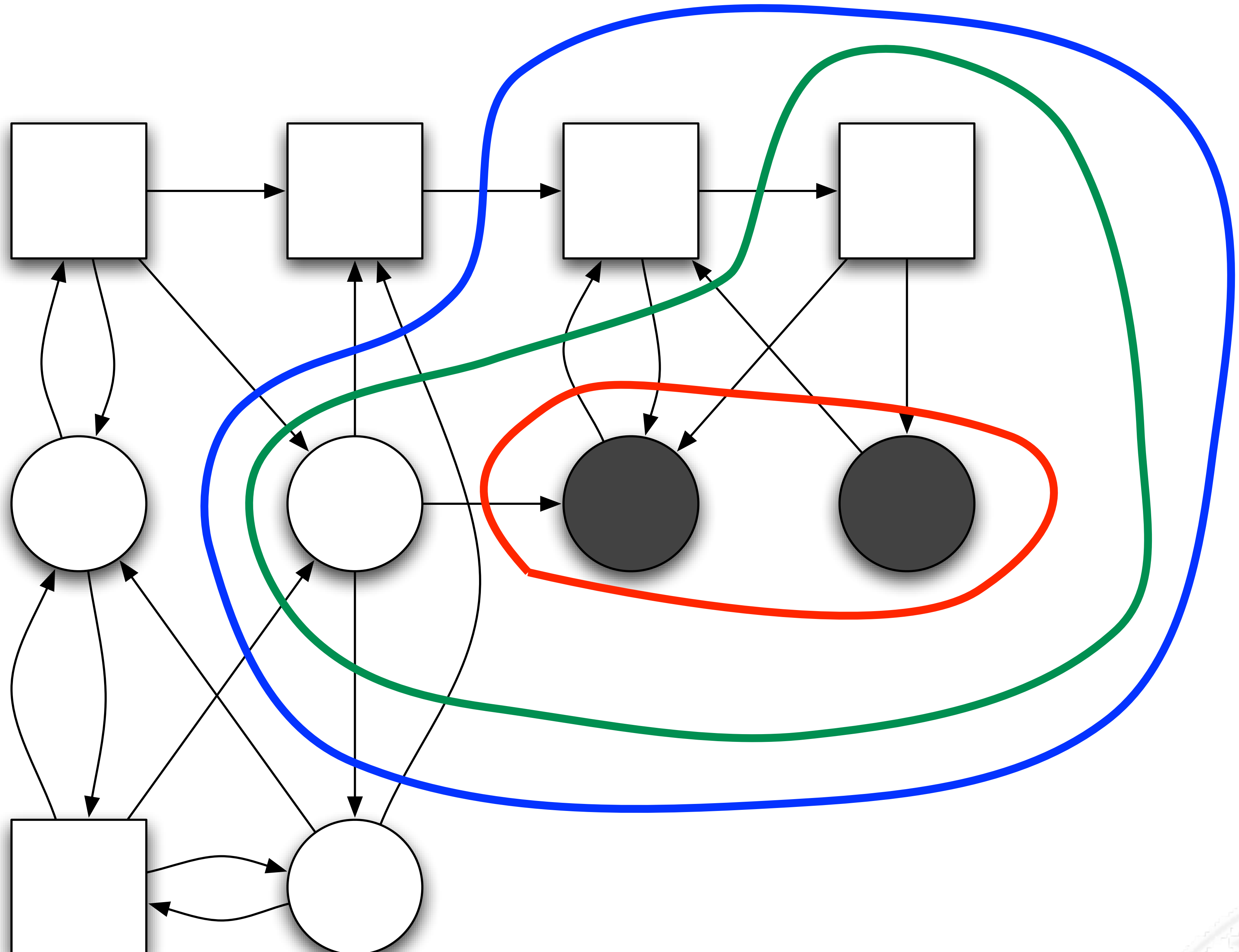




Grey nodes are the objective for B

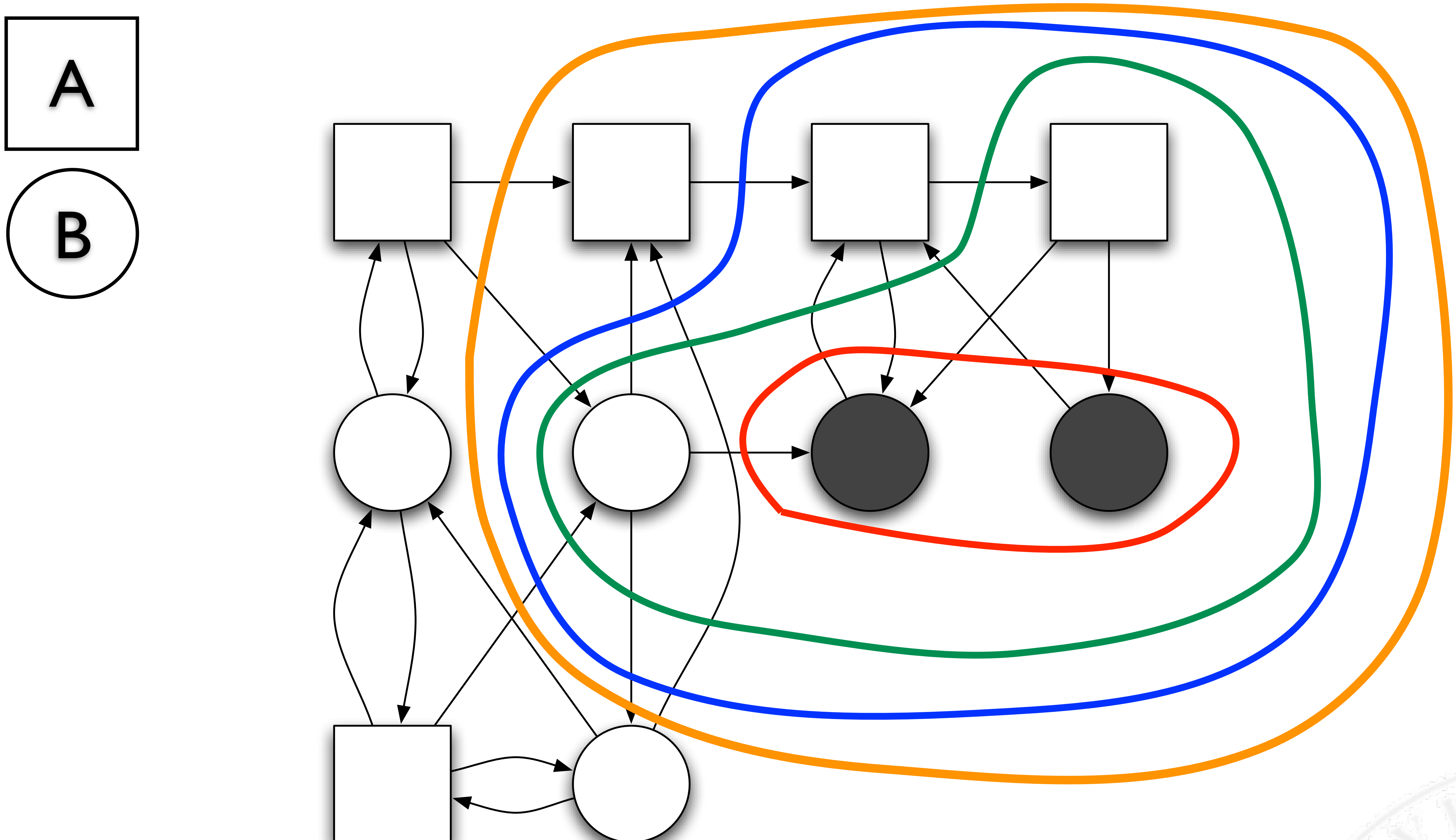
Example





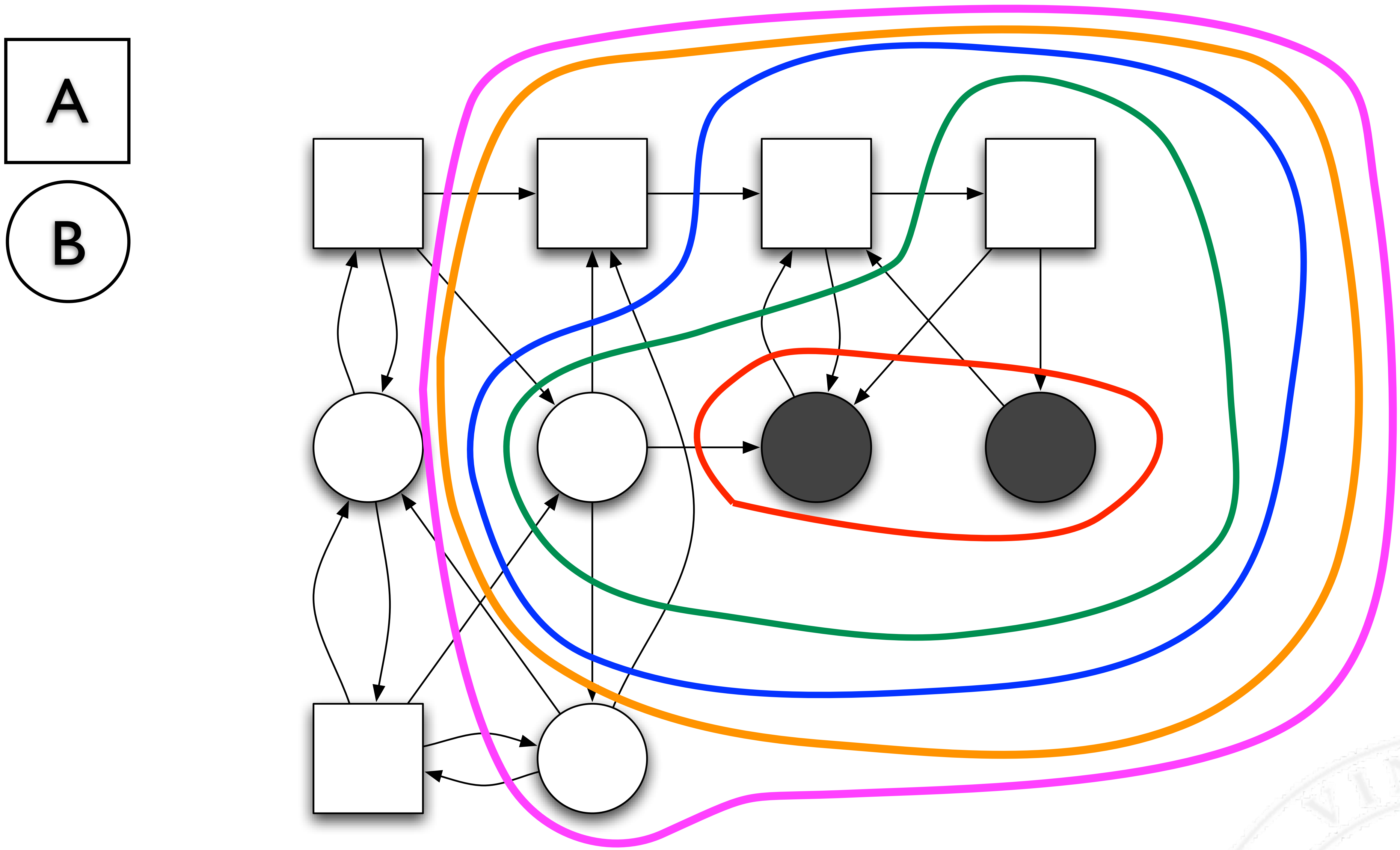
Grey nodes are the objective for B

Example



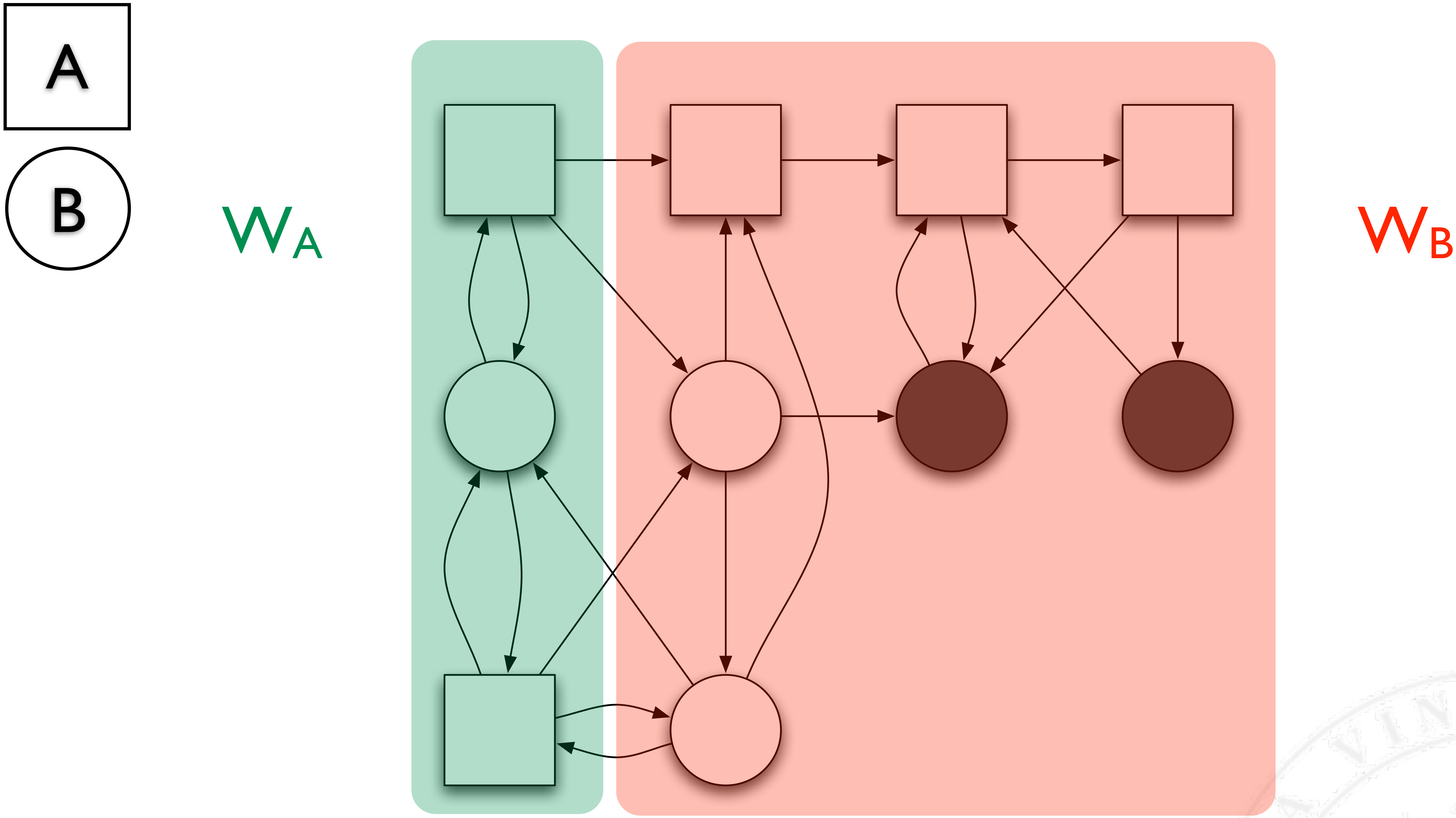
Grey nodes are the objective for B

Example



Grey nodes are the objective for B

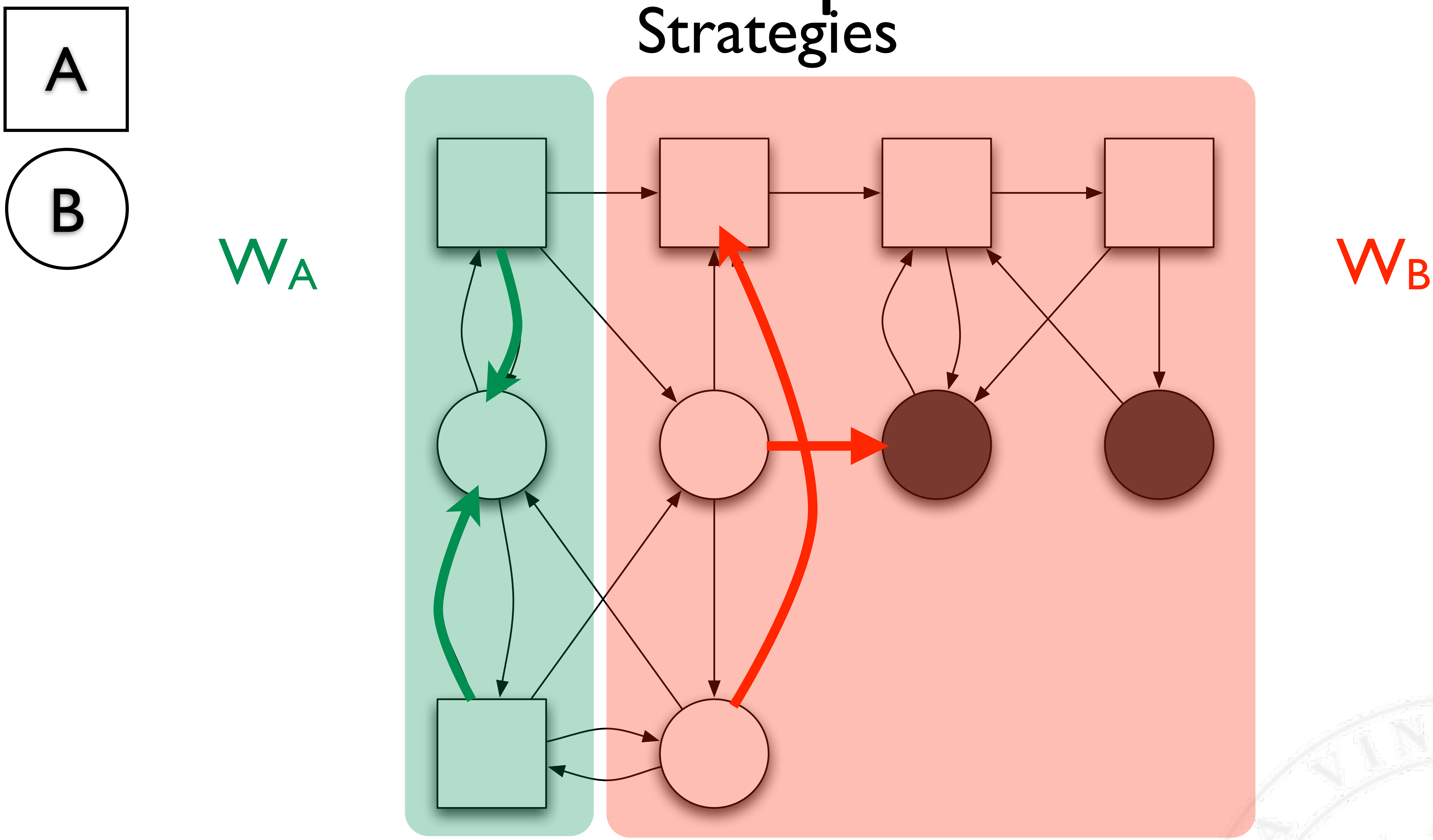
Example



Grey nodes are the objective for B

Example

Strategies



Grey nodes are the objective for B

Safety

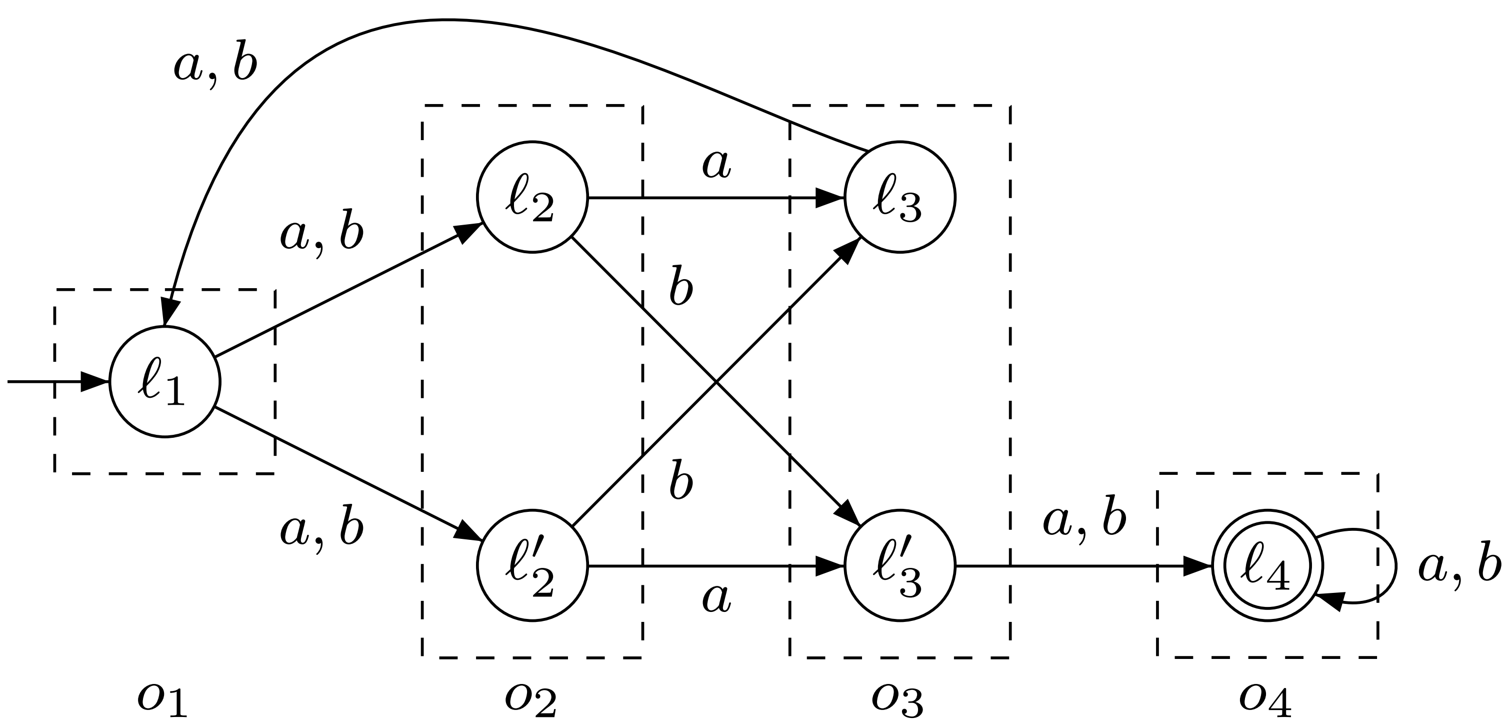
- A safety game is the dual of a **reachability game**.
  - If player A wants to reach T, B wants to avoid it
  - T is thus a reachability objective for A and a safety objective for B.
- We can thus re-use the attractor technique to solve safety games.
  - The attractor is then a set of **unsafe states**
- **Theorem**: Safety games are positionally determined

Extensions

- Too many...
  - Weighted graphs: each edge has a weight which is a price to pay when taking it.
    - Player A wants to reach a target with optimal cost
    - Player A wants to repeatedly reach the target with minimal mean-payoff
  - Probabilities
    - The second player is probabilistic (1,5 players)

Extensions

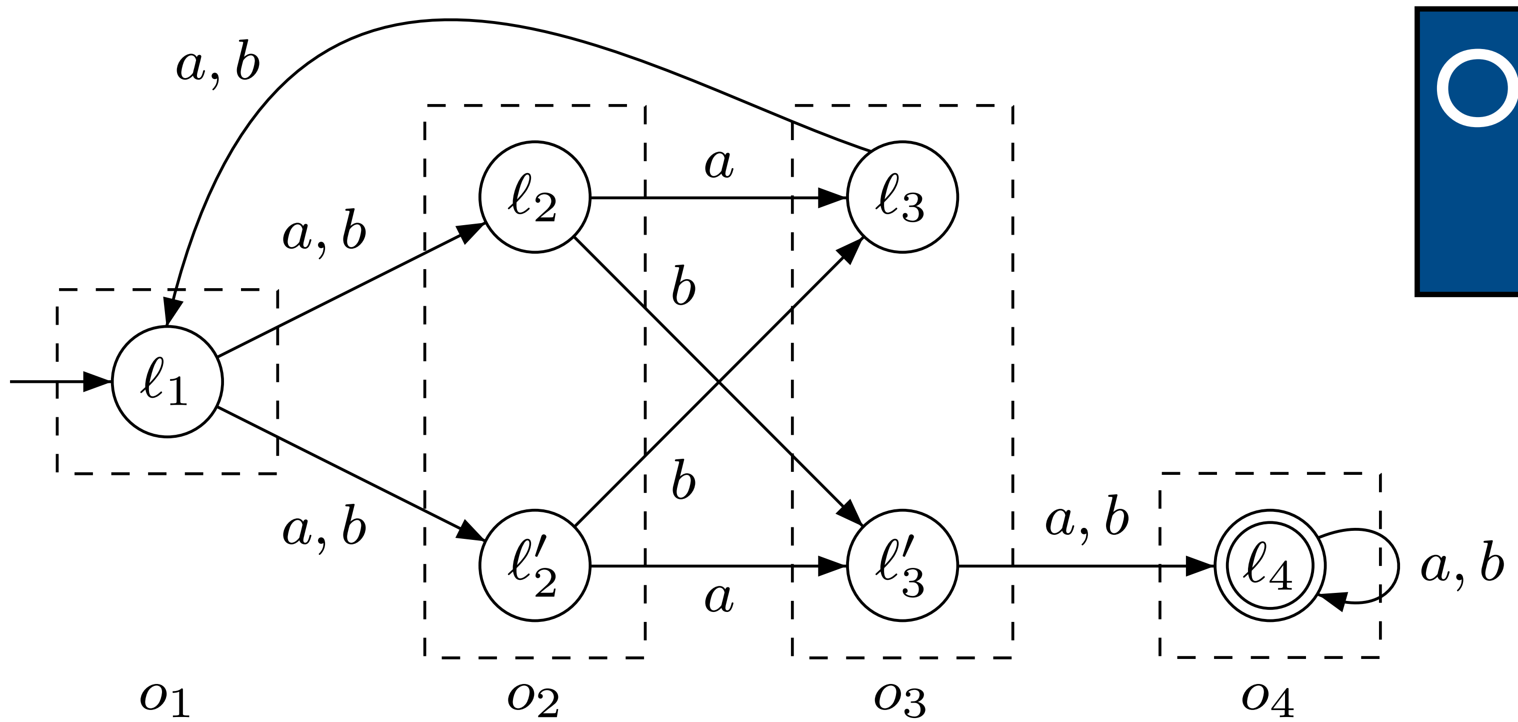
- Too many...
  - –Imperfect information
    - Player A cannot always observe in which node the game is



Player 1 chooses a letter, Player 2 chooses the successor

Extensions

- Too many...
  - Imperfect information
    - Player A cannot always observe in which node the game is



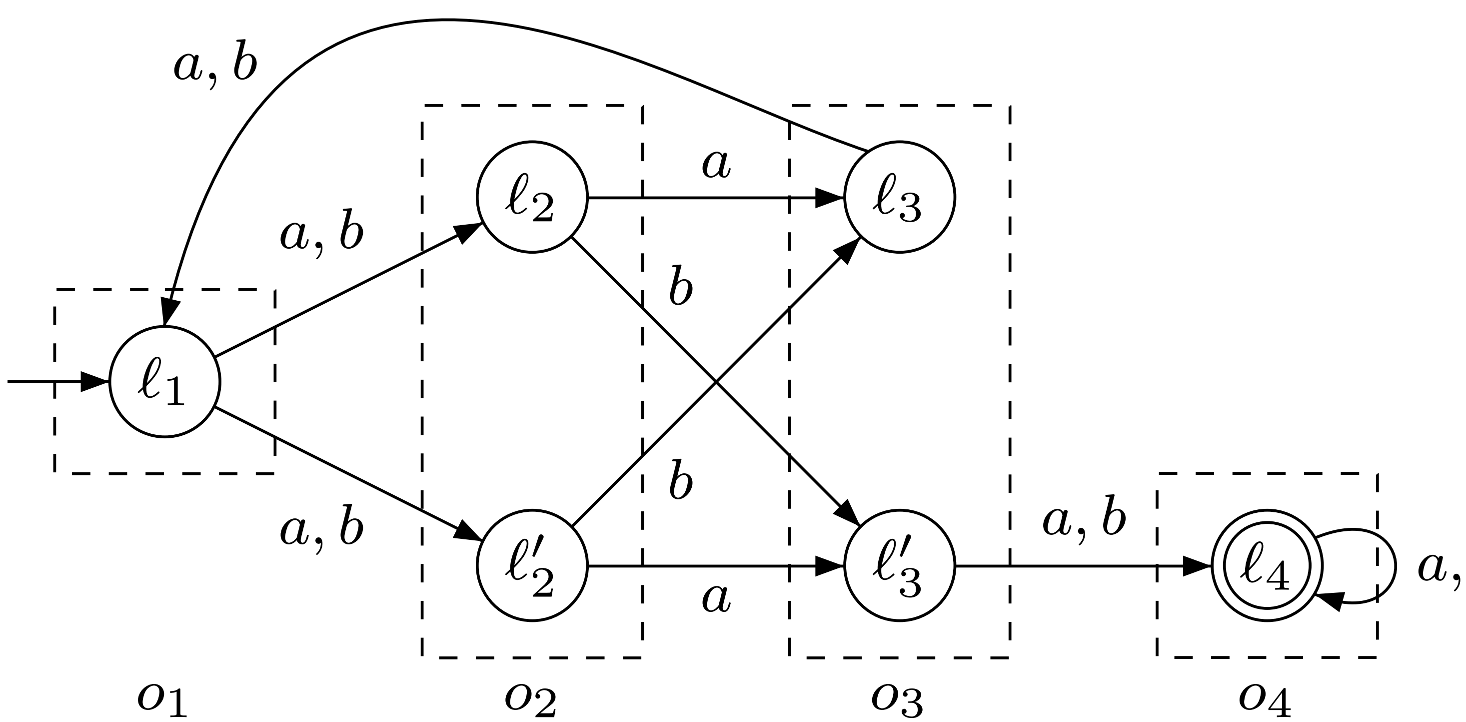
Objective: reach ℓ_4

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Extensions

- Too many...
  - Imperfect information
    - Player A cannot always observe in which node the game is



Player I cannot guarantee to reach ℓ_4, but can reach it with **high probability**

Extensions

- Too many...
  - Evolving arenas
    - The opponent might delete edges, or change the weights.
  - ...