Giochi su grafi

Marco Faella

Università di Napoli “Federico II”

Teoria dei giochi

- Teoria matematica del comportamento di agenti “razionali” in competizione tra loro
- Nata per applicazioni economiche
- von Neumann, Morgenstern, anni '40
- Shapley, Nash, anni '50+

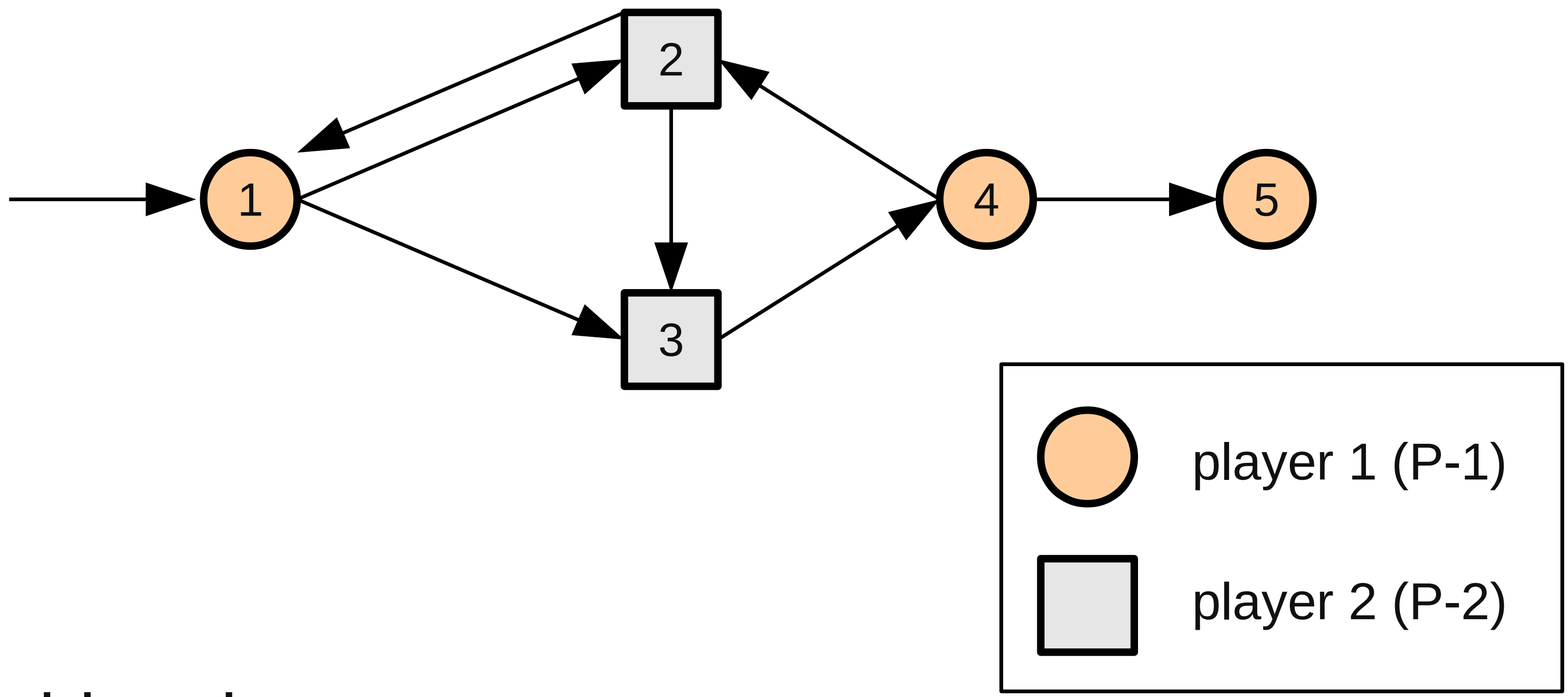
Il dilemma del prigioniero

- Due complici sono tenuti separati e devono decidere se confessare o meno
- In base alla loro scelta, verranno condannati secondo la seguente tabella:

	confessa	non confessa
confessa	(6, 6)	(0, 7)
non confessa	(7, 0)	(1,1)

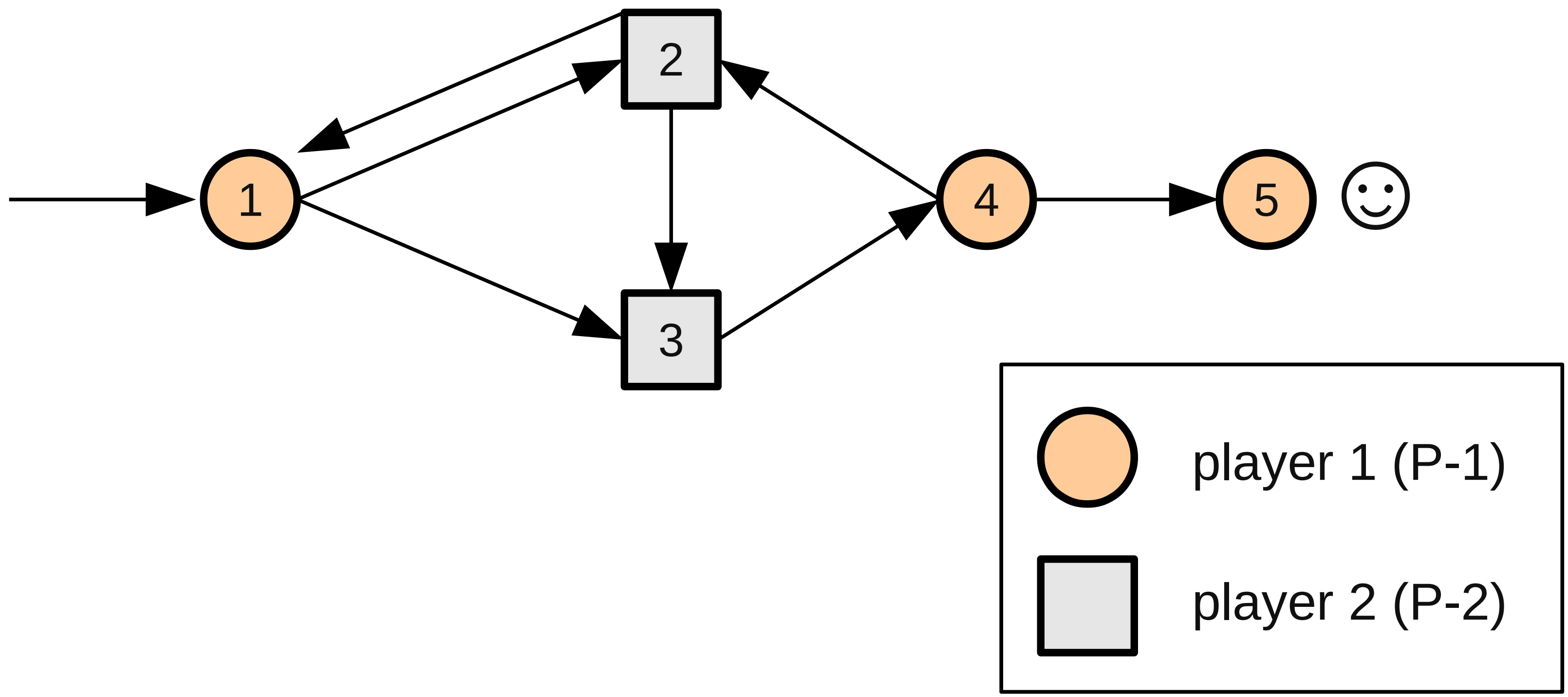
- Qual è il comportamento “razionale” da adottare?

Giochi su grafi



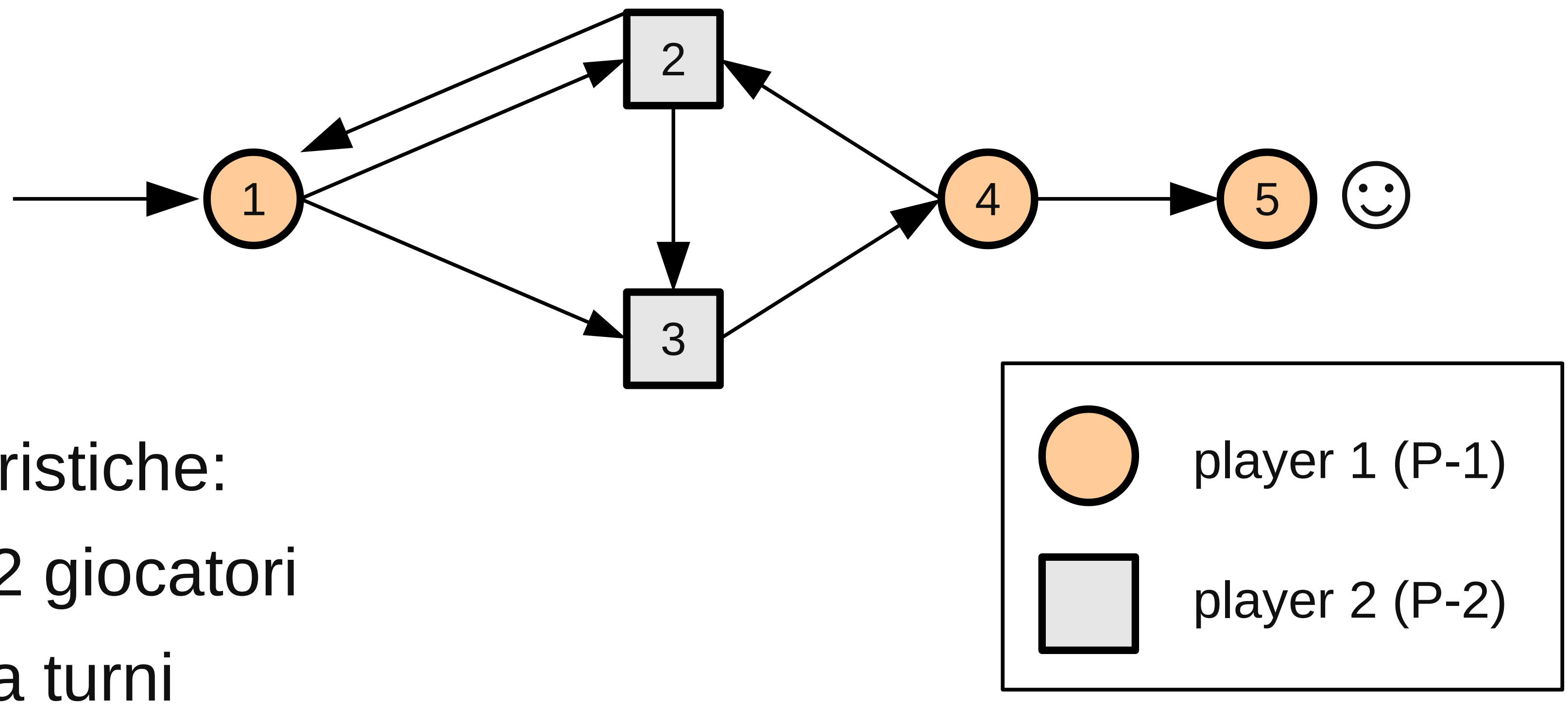
- Grafo bipartito
- I giocatori si alternano nello spostare una pedina immaginaria

Giochi su grafi: reachability



- Può il player 1 *raggiungere* 5 con certezza?
- Ha bisogno di ricordare il cammino fatto?

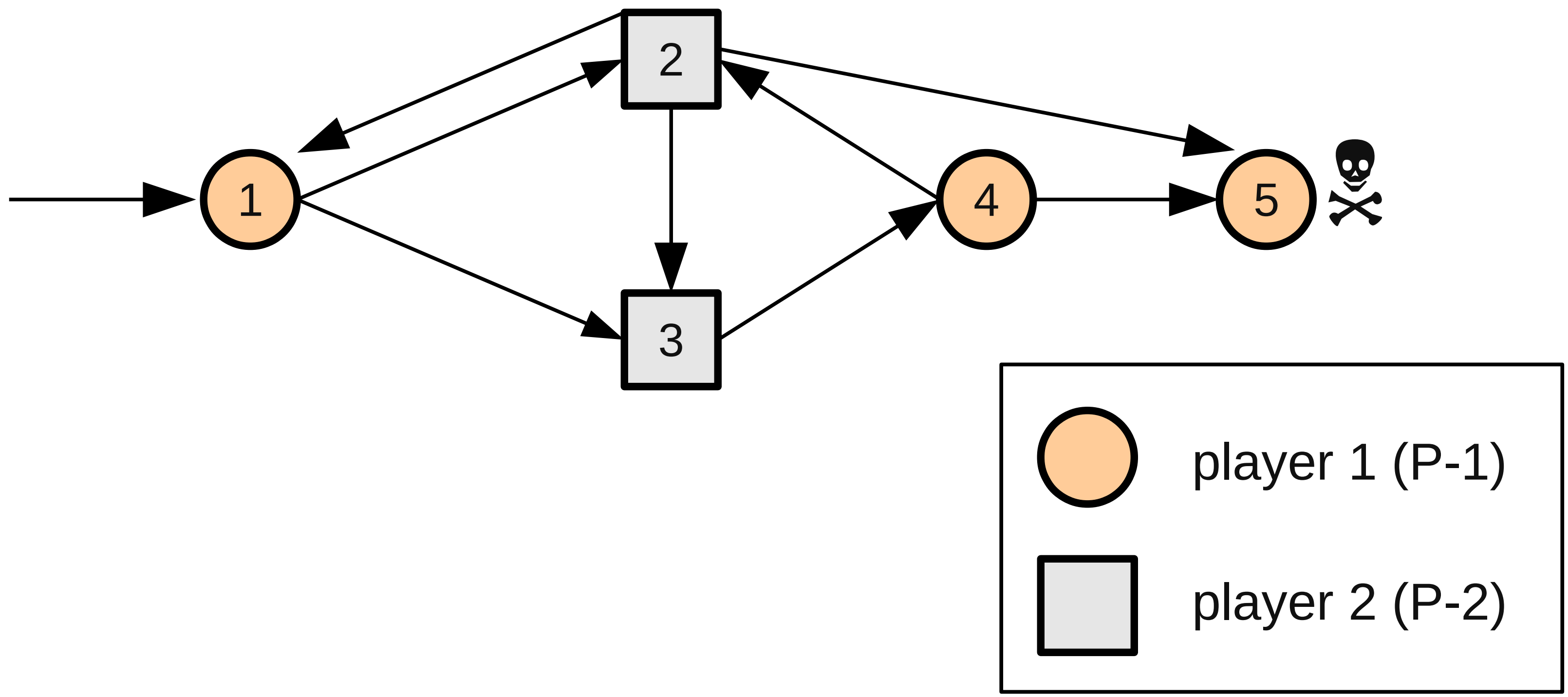
Giochi su grafi: reachability



Caratteristiche:

- 2 giocatori
- a turni
- obiettivo qualitativo: raggiungibilità
- completa informazione
- *strategie deterministiche*

Giochi su grafi: safety



- Può il player 1 *evitare* 5 con certezza?
- Ha bisogno di ricordare il cammino fatto?

Giochi su grafi: notazione

- insieme di stati: $S = S_1 \cup S_2$
- transizioni: $E \subseteq S \times S$
- obiettivo (per P-1): $goal \subseteq S^\omega$
  - è un insieme di sequenze di stati

Strategie deterministiche

- Sia Paths l'insieme dei percorsi finiti nel gioco
- Una strategia per P-1 è: $f : Paths \rightarrow S$

- Dato un percorso nel gioco, sceglie lo stato successivo
  - solo se il percorso finisce in S1
  - può solo scegliere uno stato adiacente

- Similmente per P-2

Strategie e stati vincenti

- f: strategia di P-1
- g: strategia di P-2

- Una strategia **f** è detta **vincente** (per P-1) dallo stato **s** se, per tutte le g, il percorso risultante soddisfa l'obiettivo

- Uno stato **s** è detto **vincente** (per P-1) se esiste una strategia vincente da s

Proprietà di interesse

- complessità
  - determinare l'insieme di stati vincenti
- memoria
  - è necessario ricordare il percorso?

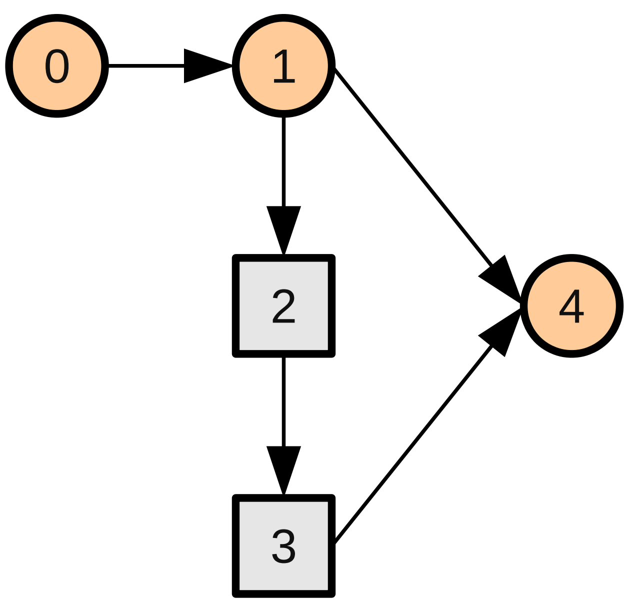
- determinatezza (*determinacy*)
  - se non vince P-1, vince P-2?

Reachability games

- Si usa l'operatore CPre (Controllable Predecessors)

$$CPre : 2^S \rightarrow 2^S$$

- CPre(X) = stati da cui P-1 può andare in X in un solo passo, con certezza



Ad esempio:

- 1 appartiene a CPre({4}) perché P-1 ha *una* mossa che va in 4
- 3 appartiene a CPre({4}) perché *tutte* le mosse di P-2 portano a 4
- 0 e 2 non appartengono a CPre({4})

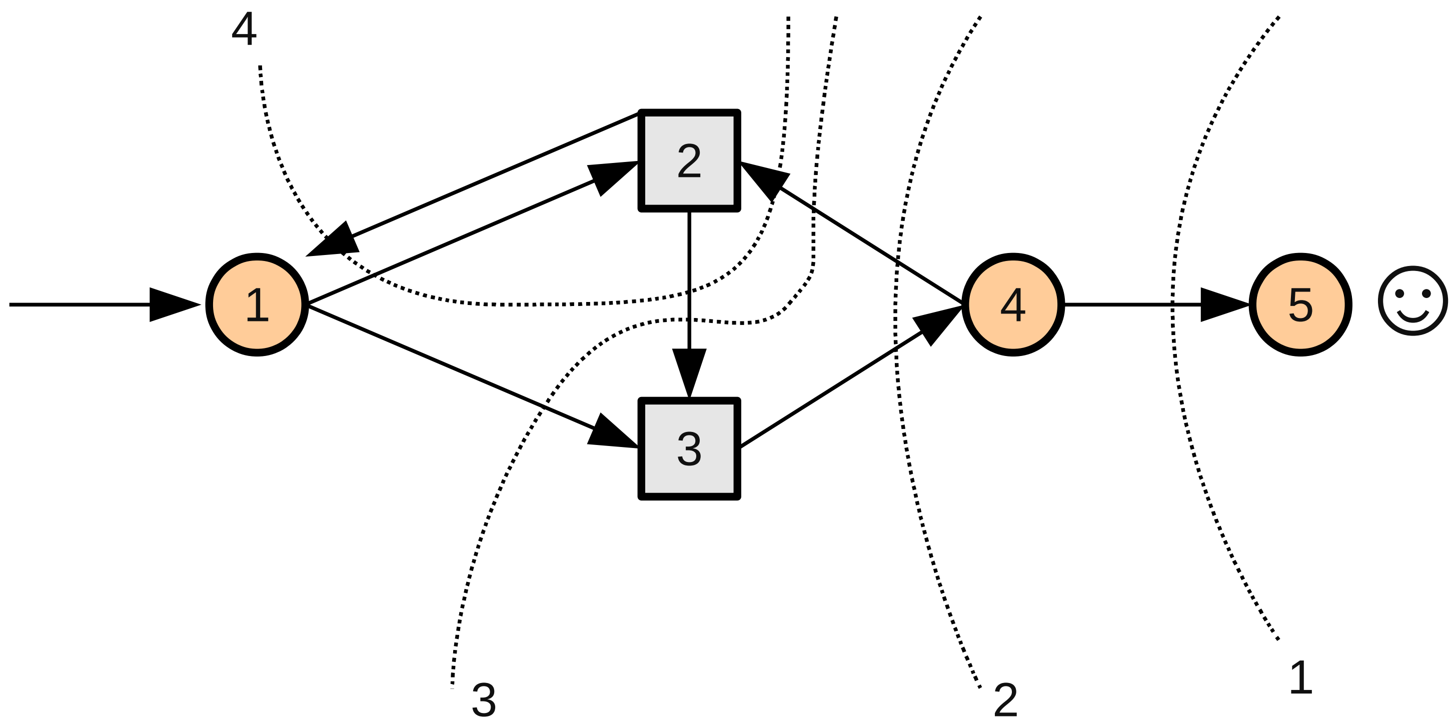
Reachability games

- Algoritmo: ricerca all'indietro (*backward search*) usando l'operatore “Controllable Predecessors”
- obiettivo: raggiungere l'insieme di stati R

$$X_0 = R$$
$$X_1 = X_0 \cup CPre(X_0)$$
$$\ldots$$
$$X_{n+1} = X_n$$

- sequenza crescente di sottoinsiemi di un insieme finito
- ovviamente, converge in numero finito di passi

Reachability games



- Può il player 1 raggiungere 5 con certezza?

Implementazione enumerativa

- Codifica del gioco: un grafo rappresentato con liste di adiacenza
- Complessità di CPre(X): tempo lineare
- Complessità dell'intero algoritmo: tempo quadratico

$$X_0 = R$$
$$X_1 = X_0 \cup CPre(X_0)$$
$$\dots$$
$$X_{n+1} = X_n \longleftarrow \mu X . R \cup CPre(X)$$

Implementazione simbolica

- Un insieme di stati è rappresentato da un BDD sull'insieme di variabili $V \cup \{turn\}$
- Le transizioni E sono rappresentate da un BDD sull'insieme di variabili $V \cup V' \cup \{turn, turn'\}$
- L'operatore CPre è facilmente ottenibile con operazioni di base su BDD
- Se X e' un BDD su $V' \cup \{turn'\}$

$$CPre(X) = (turn \wedge \exists V', turn'. E \wedge X) \vee (\neg turn \wedge \forall V', turn'. E \Rightarrow X)$$

Safety games

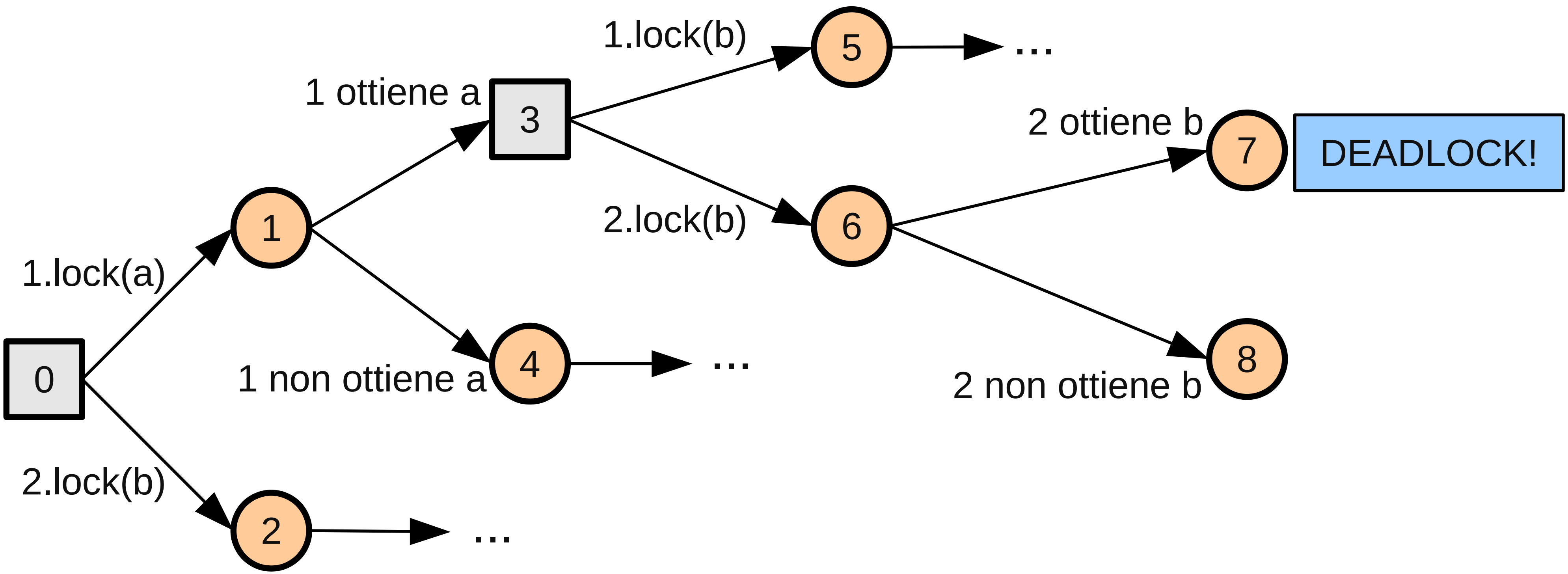
- Obiettivo: safety, cioè restare in un certo insieme di stati R
- Usiamo di nuovo CPre!

$$
\begin{aligned}
&X_0 = R\\
&X_1 = X_0 \cap CPre(X_0)\\
&\dots\\
&X_{n+1} = X_n \longleftarrow \nu X . R \cap CPre(X)
\end{aligned}
$$

- Complessità dell'intero algoritmo: tempo quadratico

Applicazioni

- Evitare i deadlock
  - processi 1 e 2
  - mutex a e b
  - P-1: implementazione di lock
  - P-2: il caso, lo scheduler, etc.



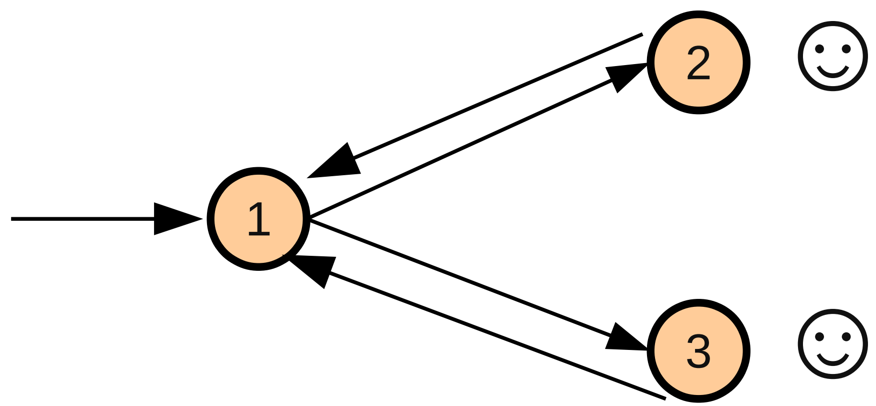
Proprietà: memoria

- E' necessario ricordare il percorso seguito?
  - No: strategie senza memoria (*memoryless*)
  - Si, ma solo la lunghezza: *counting strategies*
  - Si, ma solo quanto un automa finito può ricordare: strategie a memoria finita
  - Si: strategie a memoria illimitata (*calcolabile?*)

- Reachability e safety: nessuna memoria

Proprietà: memoria

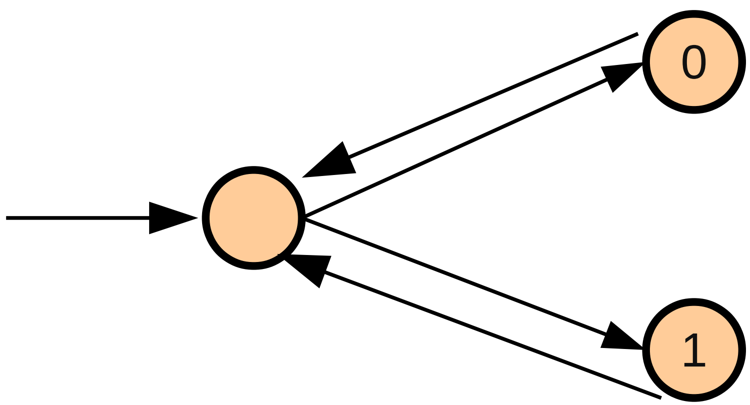
- Un gioco che richiede memoria
- obiettivo: visitare sia 2 che 3 infinite volte



- Basta 1 bit di memoria, ovvero un automa con due stati

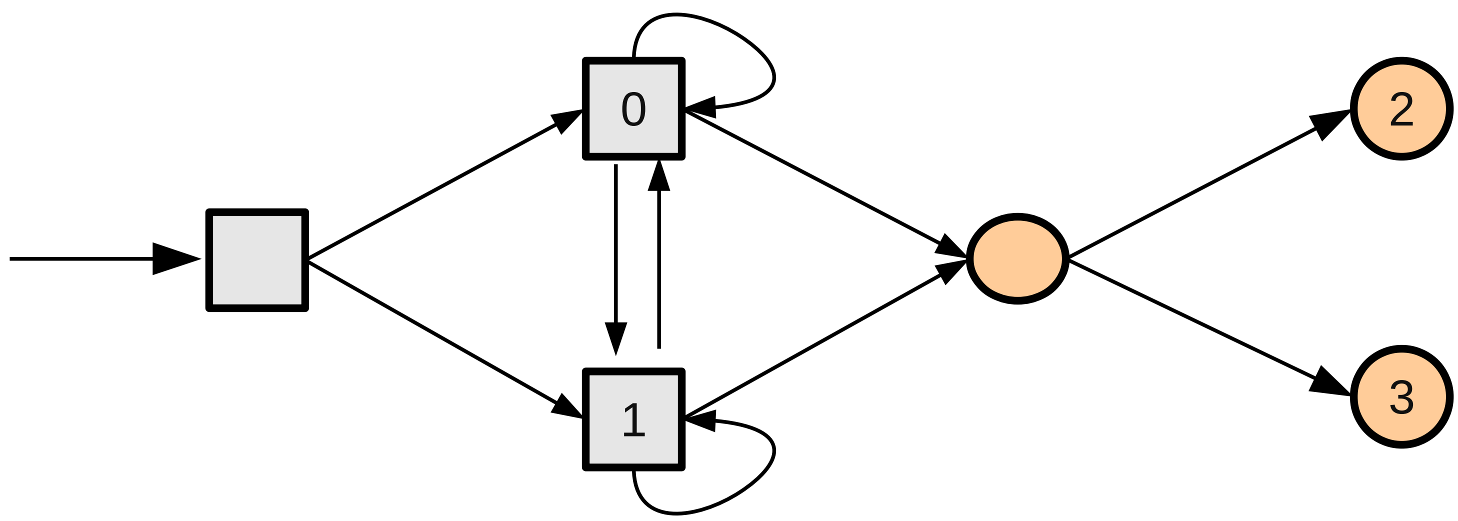
Proprietà: memoria

- Un gioco che richiede memoria illimitata ma calcolabile
- obiettivo: ottenere la codifica binaria di pi greco



Proprietà: memoria

- Un gioco che richiede memoria illimitata e **non calcolabile**
- obiettivo: se l'avversario codifica una macchina di Turing che si arresta sulla parola vuota, raggiungere 2, altrimenti raggiungere 3



Proprietà: determinatezza

- Se uno stato *non* è vincente per P-1, è vincente per P-2?

- Reachability e safety: si
- Thm: tutti i giochi con obiettivo di tipo “Borel” sono determinati [Martin75]

Perché queste proprietà?

- complessità e memoria per:
  - verificare modelli di sistemi
  - sintetizzare controllori
  - teoria del controllo
- determinatezza per:
  - automi e logica
  - complementazione di automi
  - completezza di logiche

Distribuzioni

- Dato un insieme finito A, una *distribuzione* (di probabilità) è una funzione

$$f : A \to [0,1]$$

- tale che

$$\sum_{a \in A} f(a) = 1$$

Strategie

- deterministiche $f : Paths \rightarrow S$
- non-deterministiche $f : Paths \rightarrow 2^S$
- randomizzate $f : Paths \rightarrow Distr(S)$

- cambia la definizione di strategia vincente

Strategie

- due strategie f e g di categoria X determinano:
  - X=det: un percorso
  - X=non-det: un insieme di percorsi
  - X=rand: una distribuzione sui percorsi

- nuova def. di vittoria
  - esiste una strategia f tale che per tutte le g:
    - X=det: il percorso soddisfa l'obiettivo
    - X=non-det: tutti i percorsi soddisfano l'obiettivo
    - X=rand: ??

Strategie randomizzate

$$f : Paths \rightarrow Distr(S)$$

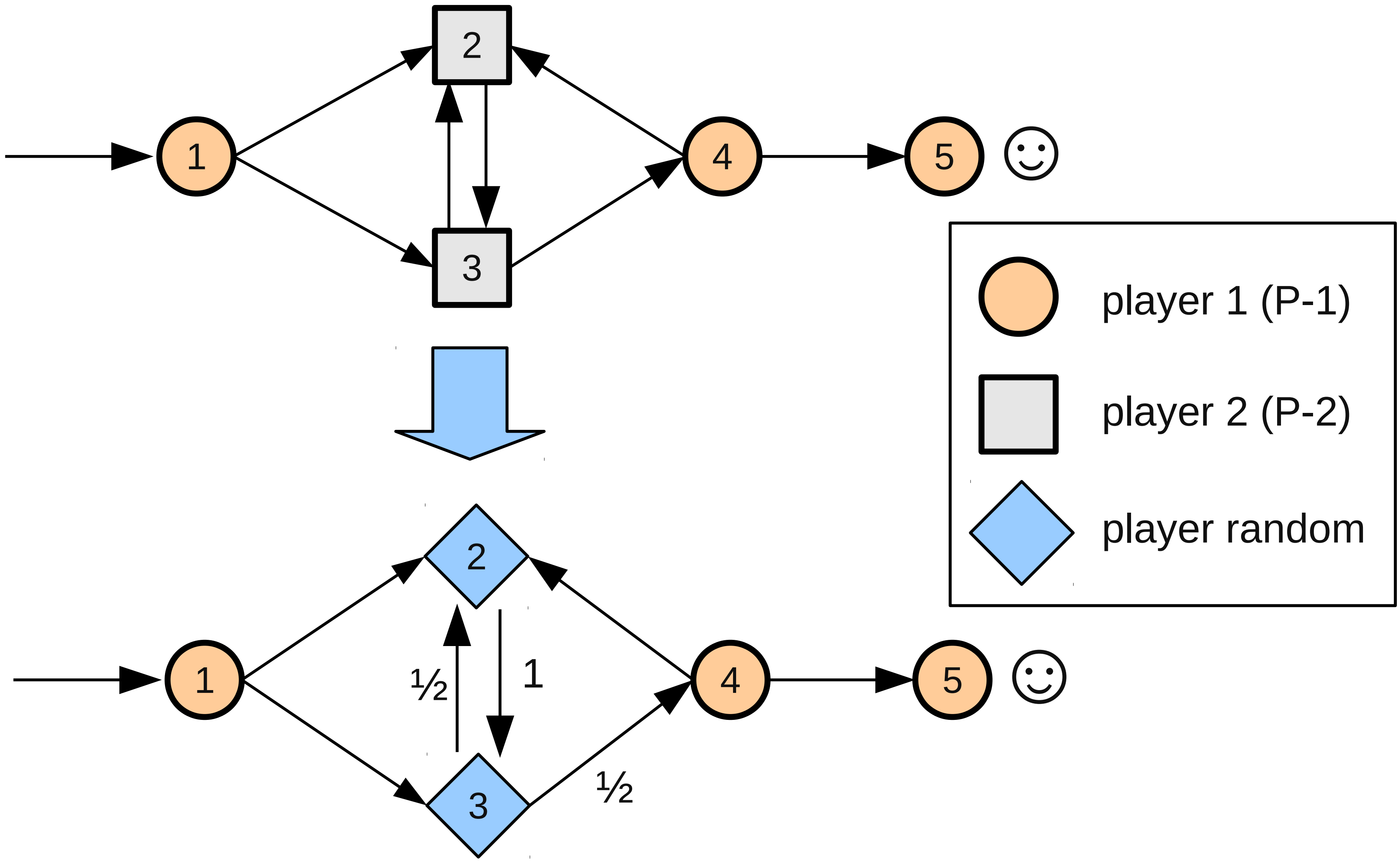
- una strategia f è vincente se, per tutte le g:
  - tutti i percorsi soddisfano l'obiettivo (*sure win*)
  - la probabilità di soddisfare l'obiettivo è 1 (*almost sure win*)
- oppure, uno stato è vincente se esiste una sequenza di strategie f1, f2, ... che si avvicinano sempre più alla probabilità di vittoria 1 (*limit win*)

Strategie randomizzate

$$f : Paths \rightarrow Distr(S)$$

- Thm: per tutti i giochi a turni, le strategie randomizzate non servono

Da 2 a 1.5 giocatori



Altri tipi di giochi: Numero di giocatori

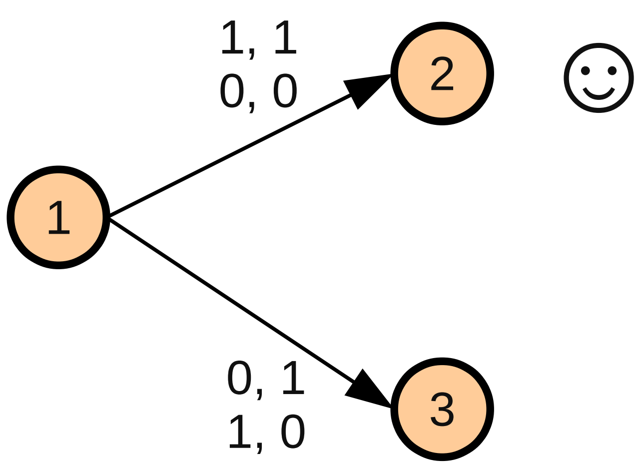
0.5	catena di Markov
1	grafo
1.5	Markov Decision Process
2	**gioco**
2.5	gioco stocastico
3	
...	

Altri tipi di giochi:
A turni VS concorrenti

- Nei giochi concorrenti, i giocatori muovono **contemporaneamente**
- esempio: morra cinese
- Strategie randomizzate possono essere utili, anche per la semplice reachability

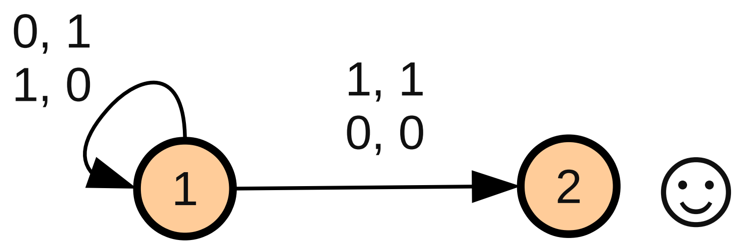
Un gioco concorrente

- ciascun giocatore sceglie 0 o 1
- la destinazione dipende dalla coppia di mosse scelte
- P-1 vince se sceglie lo stesso numero di P-2



Un altro gioco concorrente

- ciascun giocatore sceglie 0 o 1
- la destinazione dipende dalla coppia di mosse scelte
- P-1 vince se sceglie lo stesso numero di P-2



Dominazione

- Date due strategie deterministiche f e g, sia
  - val(f, g) = 1 se f vince contro g
  - val(f, g) = 0 altrimenti

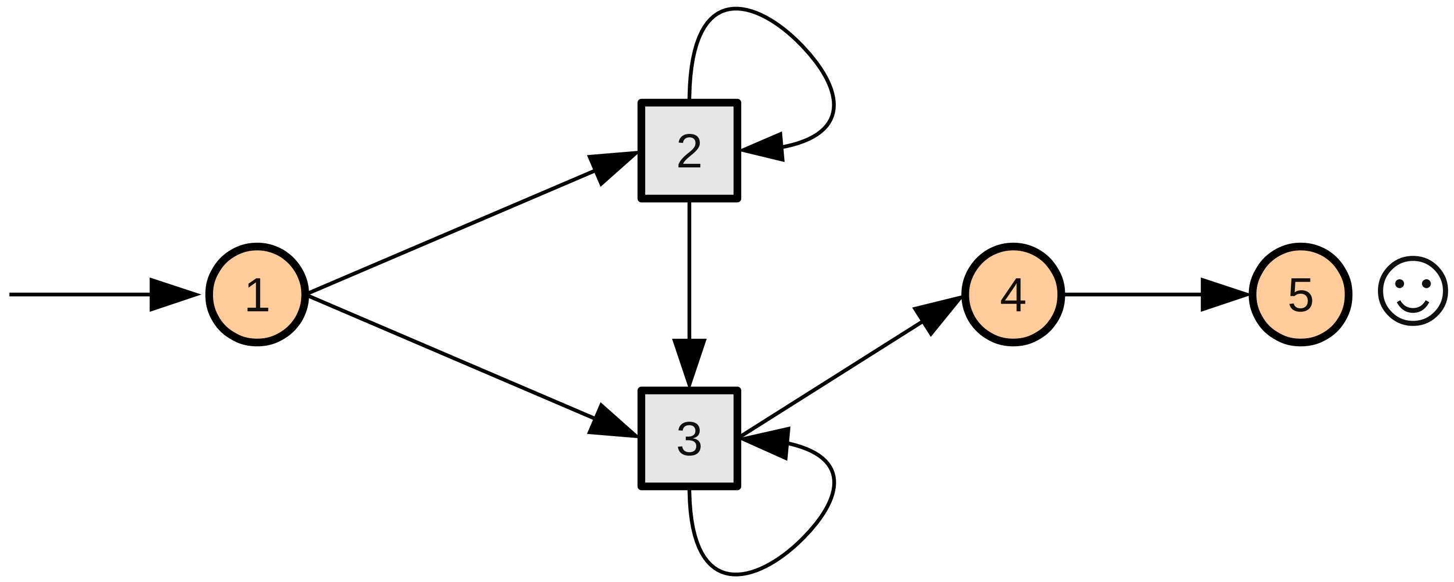
Dominazione

- Date due strategie di P-1: f1 ed f2,
- si dice che f1 *domina* f2 se
  - per tutte le strategie g di P-2:

    val(f1, g) >= val(f2, g)
  - ed esiste una strategia g di P-2 tale che:

    val(f1, g) > val(f2, g)

Dominazione: esempio



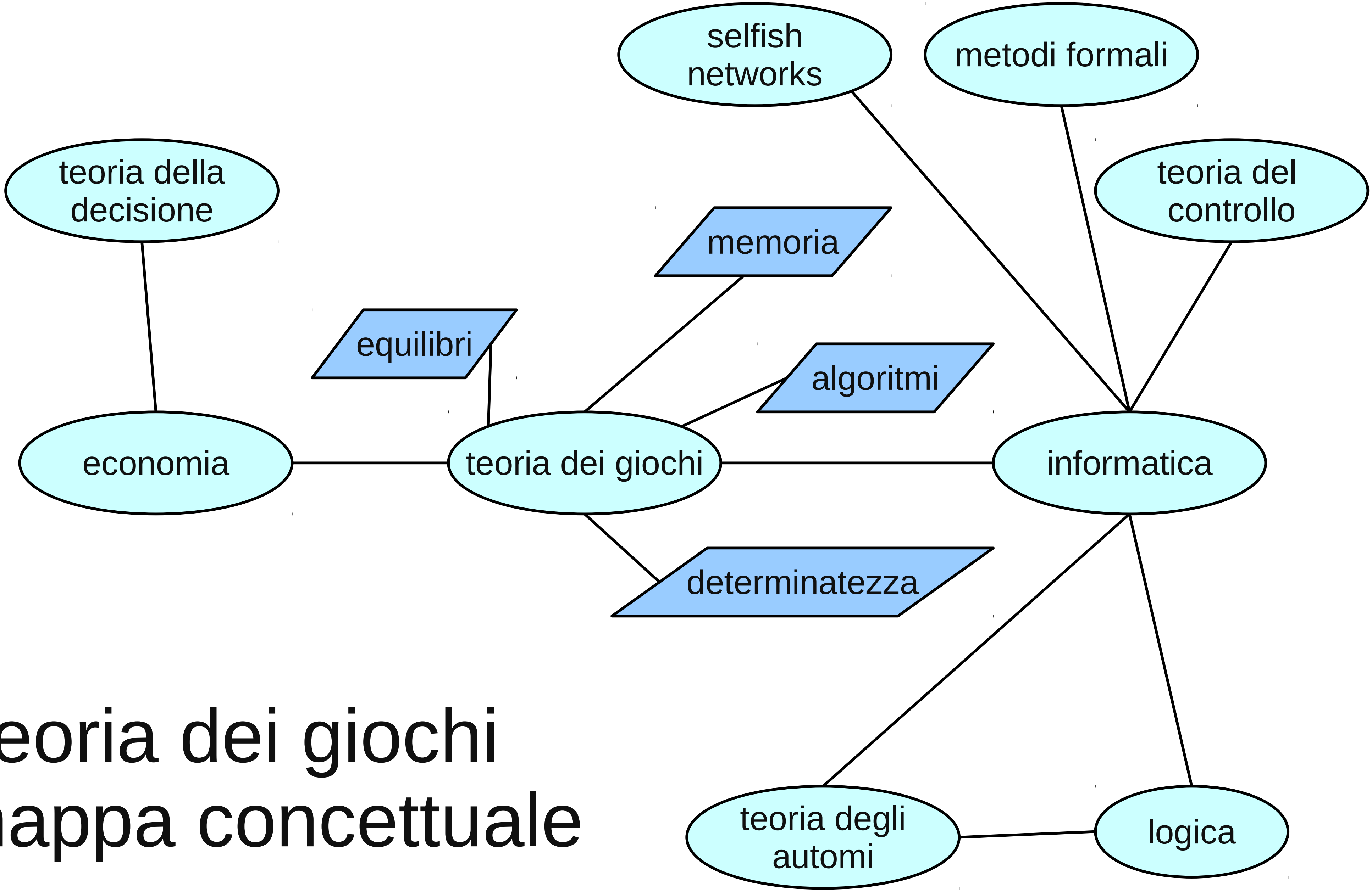
- P-1 non può vincere dallo stato 1
- Però, la mossa verso 3 domina quella verso 2
  - (almeno, se P-2 gioca senza memoria)

Dominazione

- In un gioco che non si può vincere, una strategia **non dominata** rappresenta la miglior scelta per un giocatore
- Qual è la complessità di calcolare una strategia non dominata?
- Come si caratterizza una strategia non dominata?

Bibliografia

- D. Gale and F. Stewart, Infinite games with perfect information. *Annals of mathematical studies 28 (1953)*
- W. Thomas, **On the synthesis of strategies in infinite games.** *STACS1995*
- W. Zielonka, Infinite games on finitely coloured graphs with applications to automata on infinite trees. *TCS (1998)*
- L. de Alfaro and T.A. Henzinger. Concurrent Omega-Regular Games. *LICS2000*



Teoria dei giochi
mappa concettuale