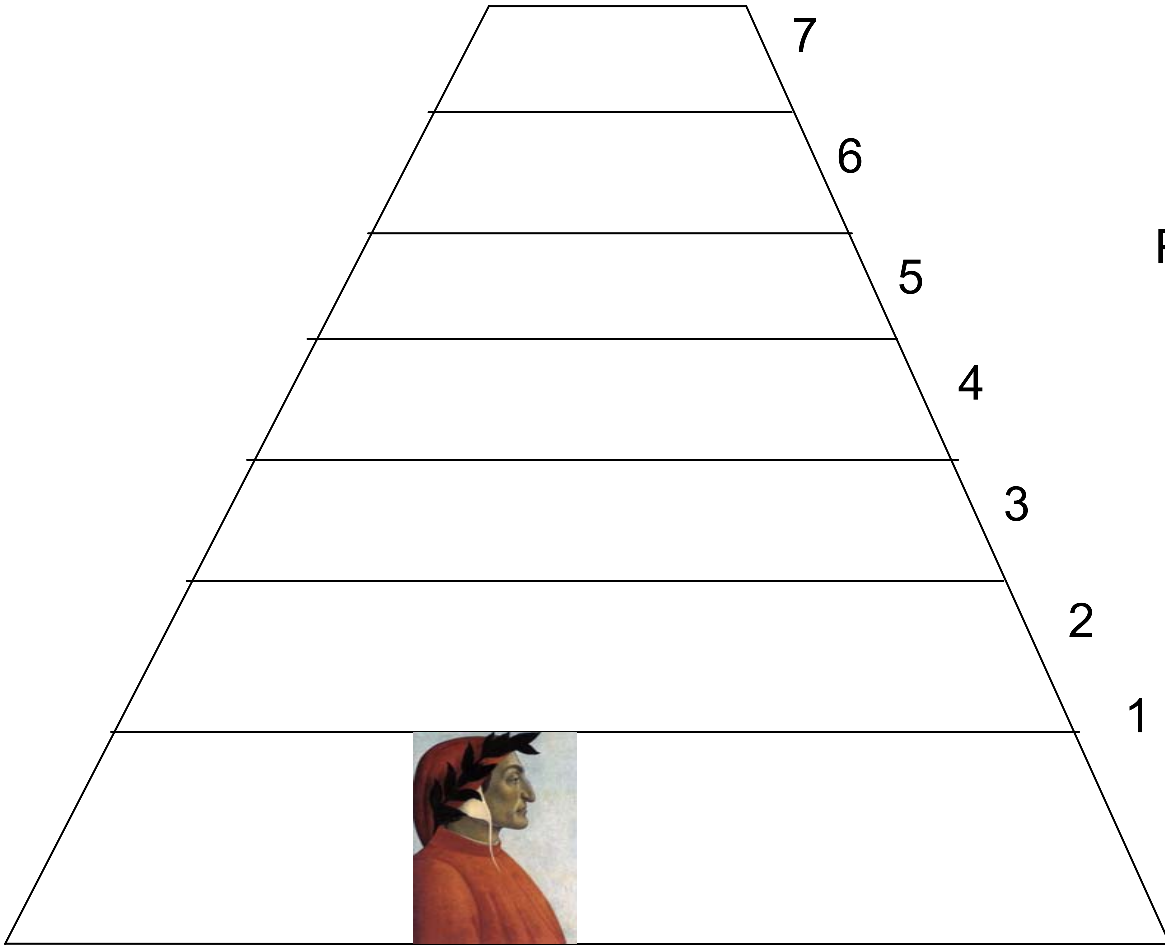
Some recent results and some open problems concerning solving infinite duration combinatorial games

Peter Bro Miltersen
Aarhus University

Purgatory

Mount Purgatory is on an island, the only land in the Southern Hemisphere, created with earth taken from the excavation of Hell (Dante, 1308).

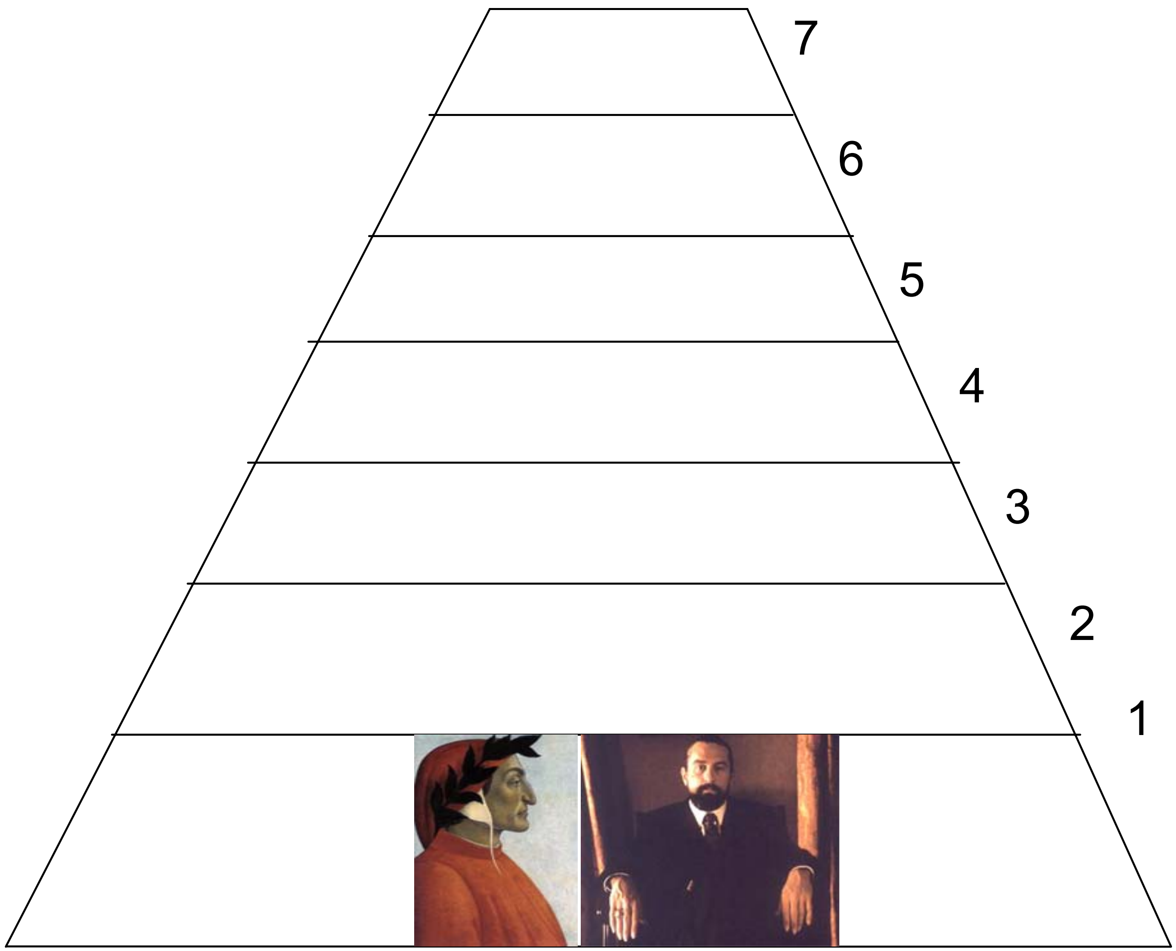
Dante in Purgatory



Purgatory has 7 terraces.

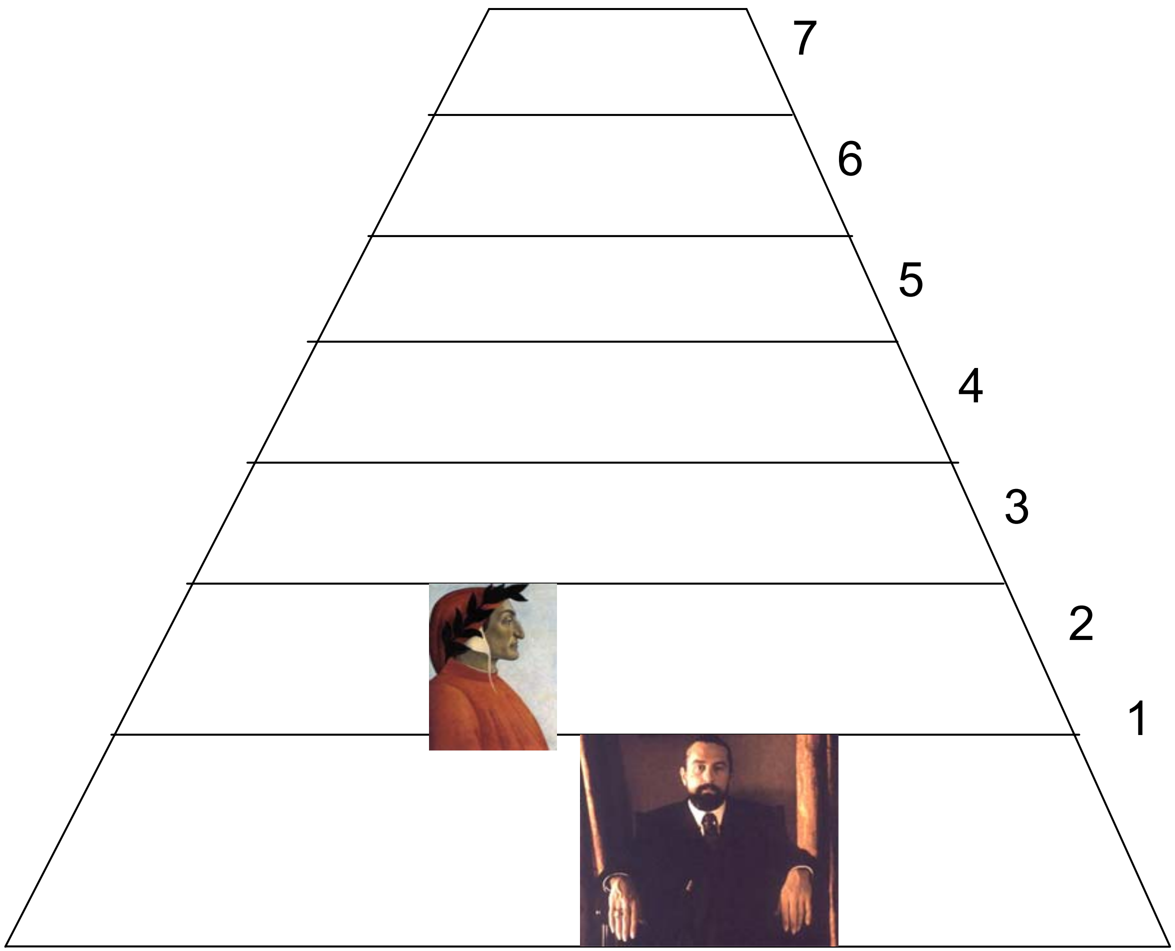
Dante enters Purgatory
at terrace 1.

Dante in Purgatory



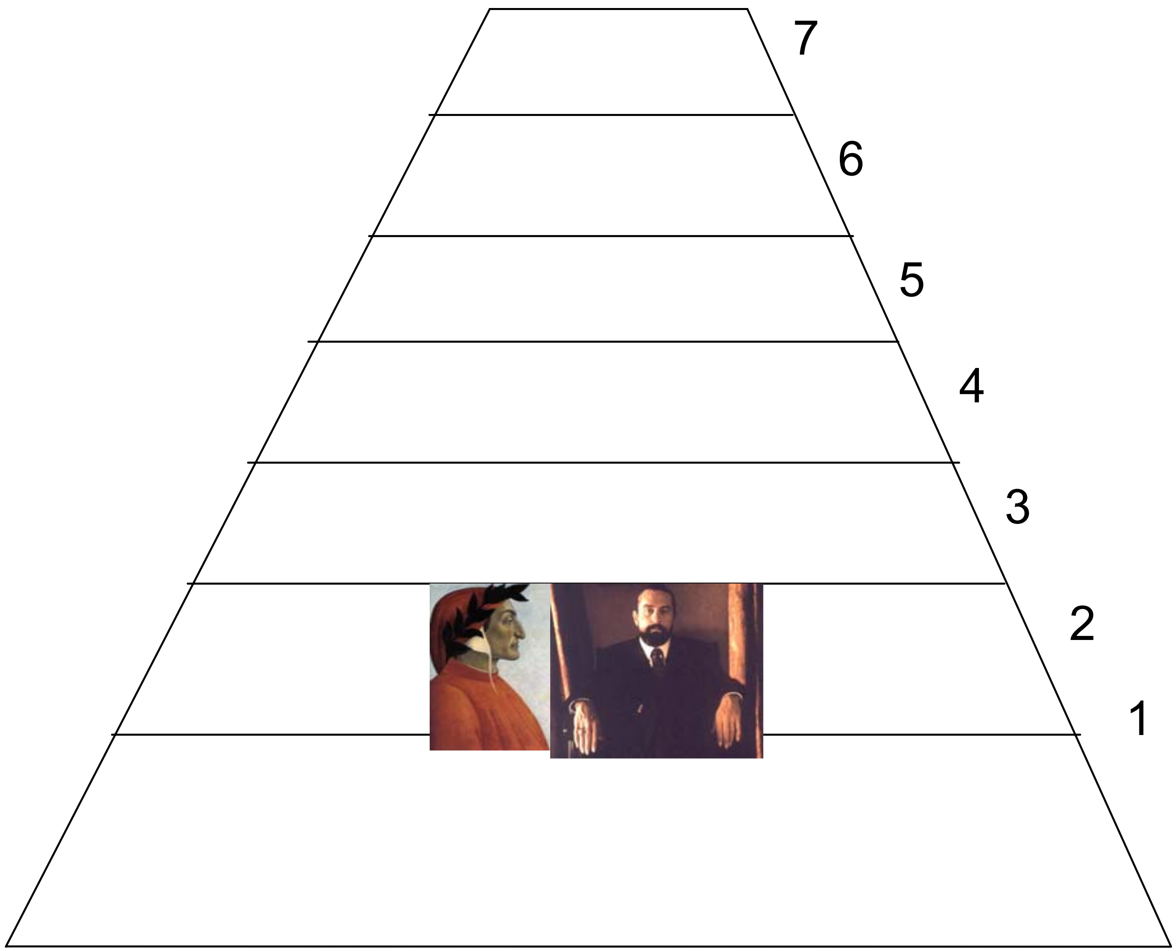
While in Purgatory, once a second, Dante must play "Guess-which-hand" with Lucifer

Dante in Purgatory



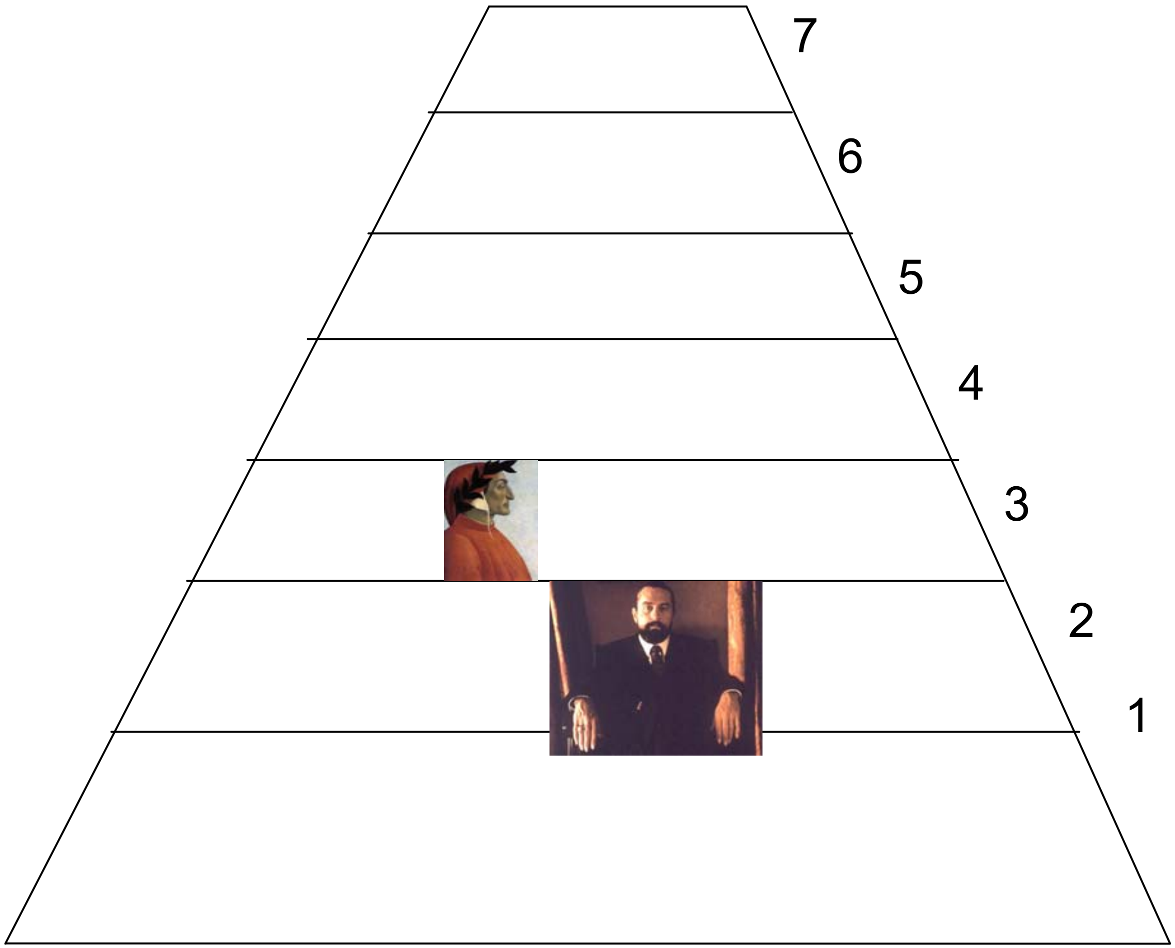
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



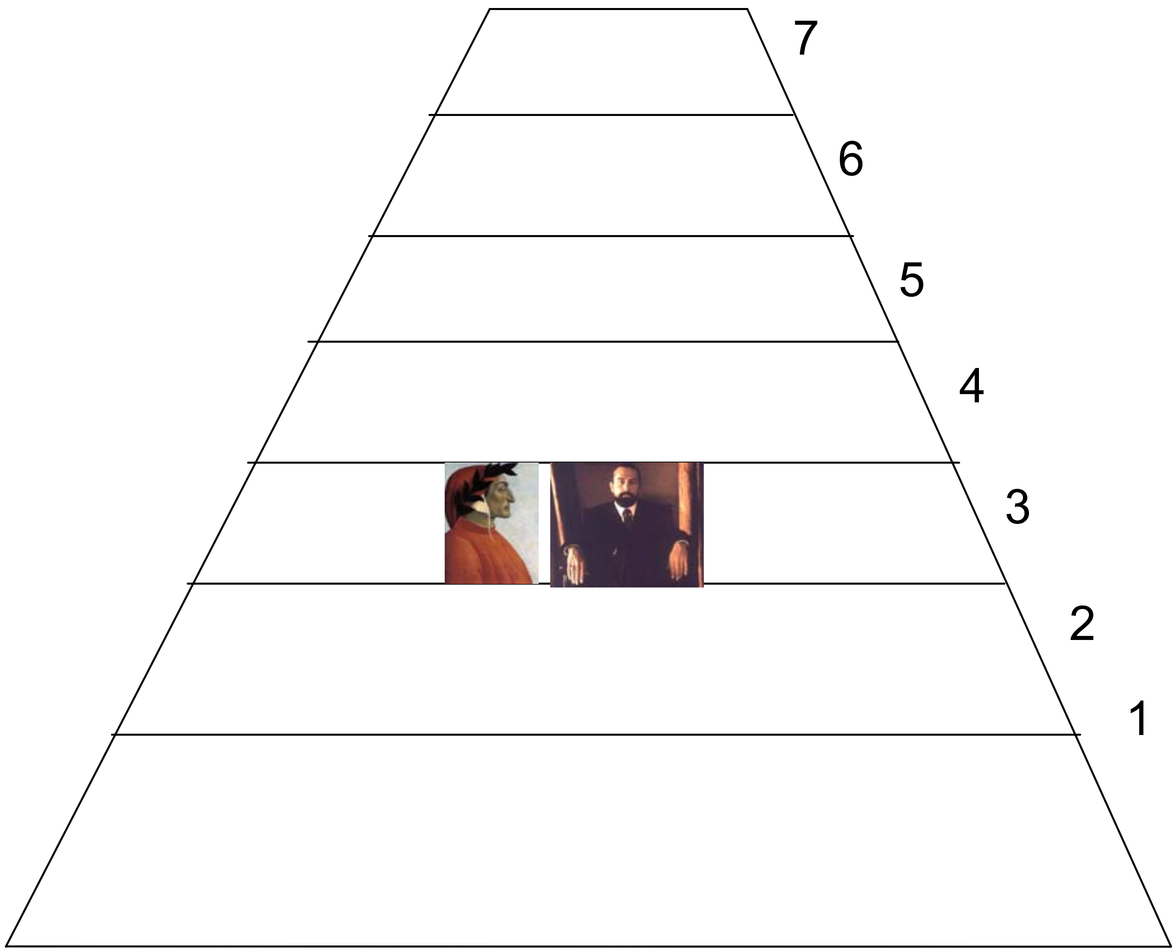
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



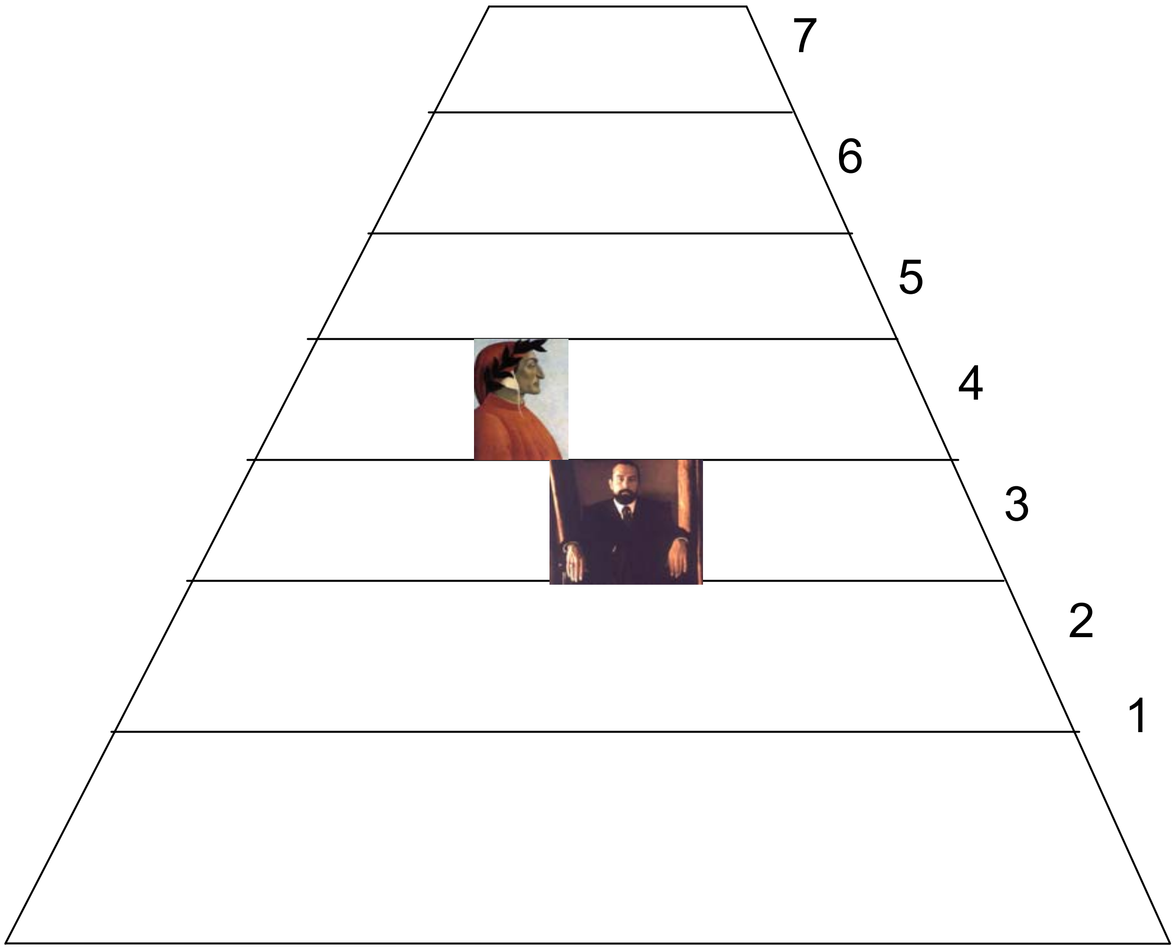
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



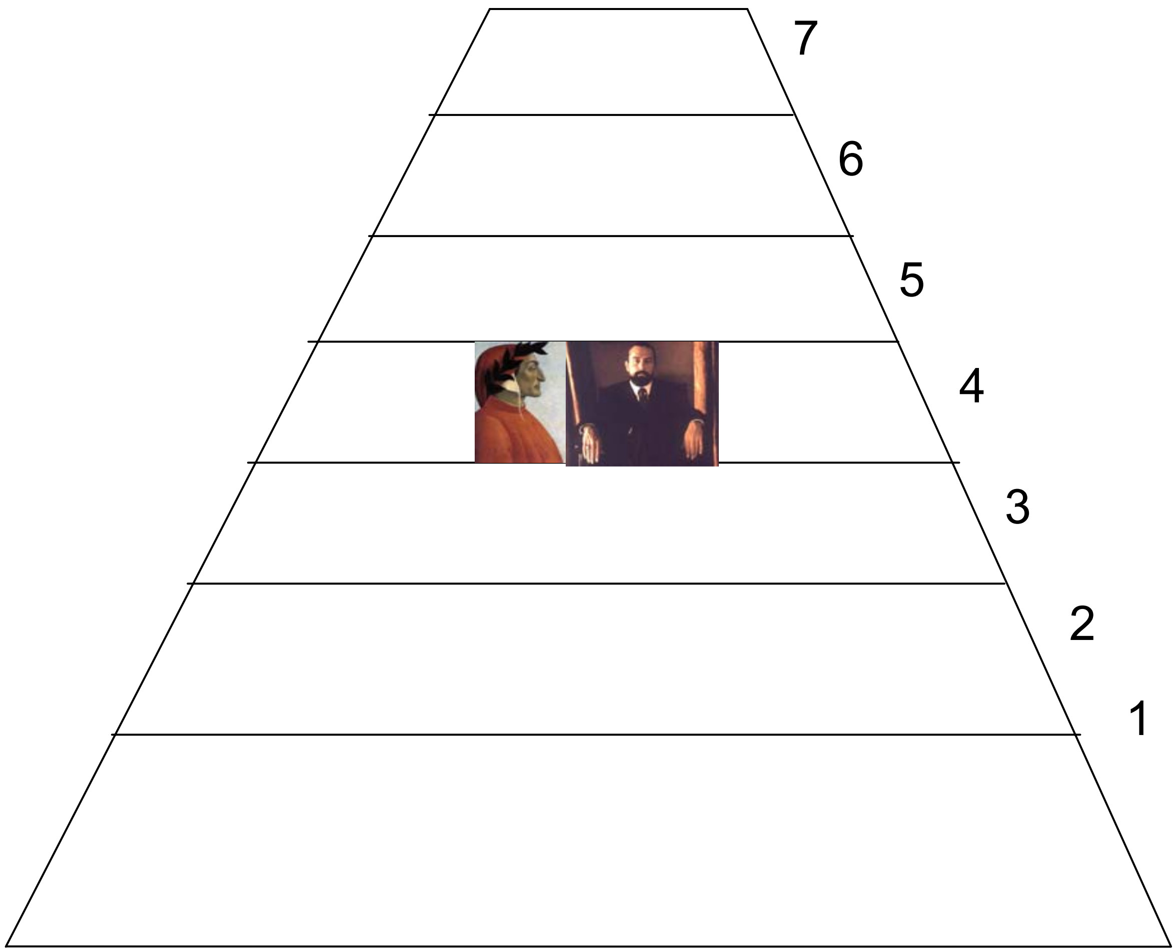
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



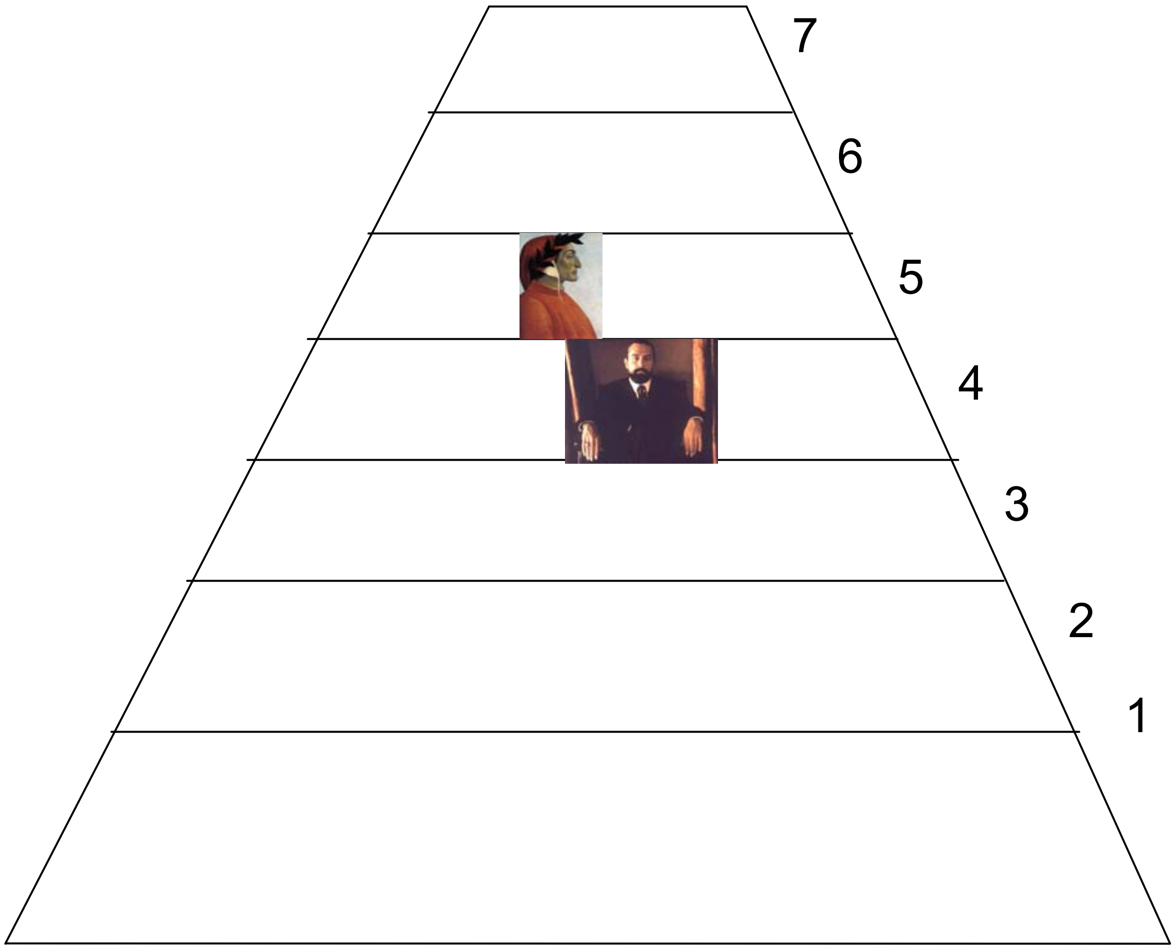
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



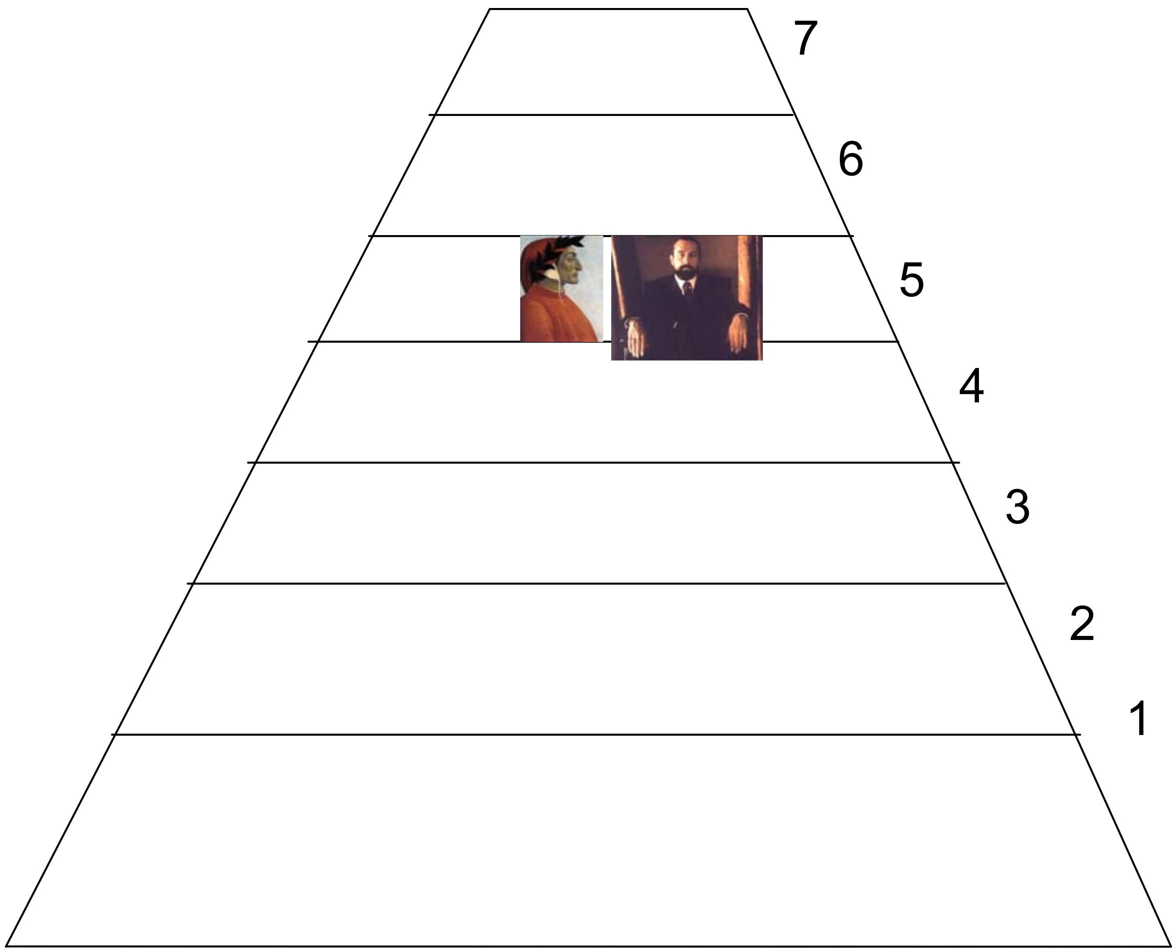
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



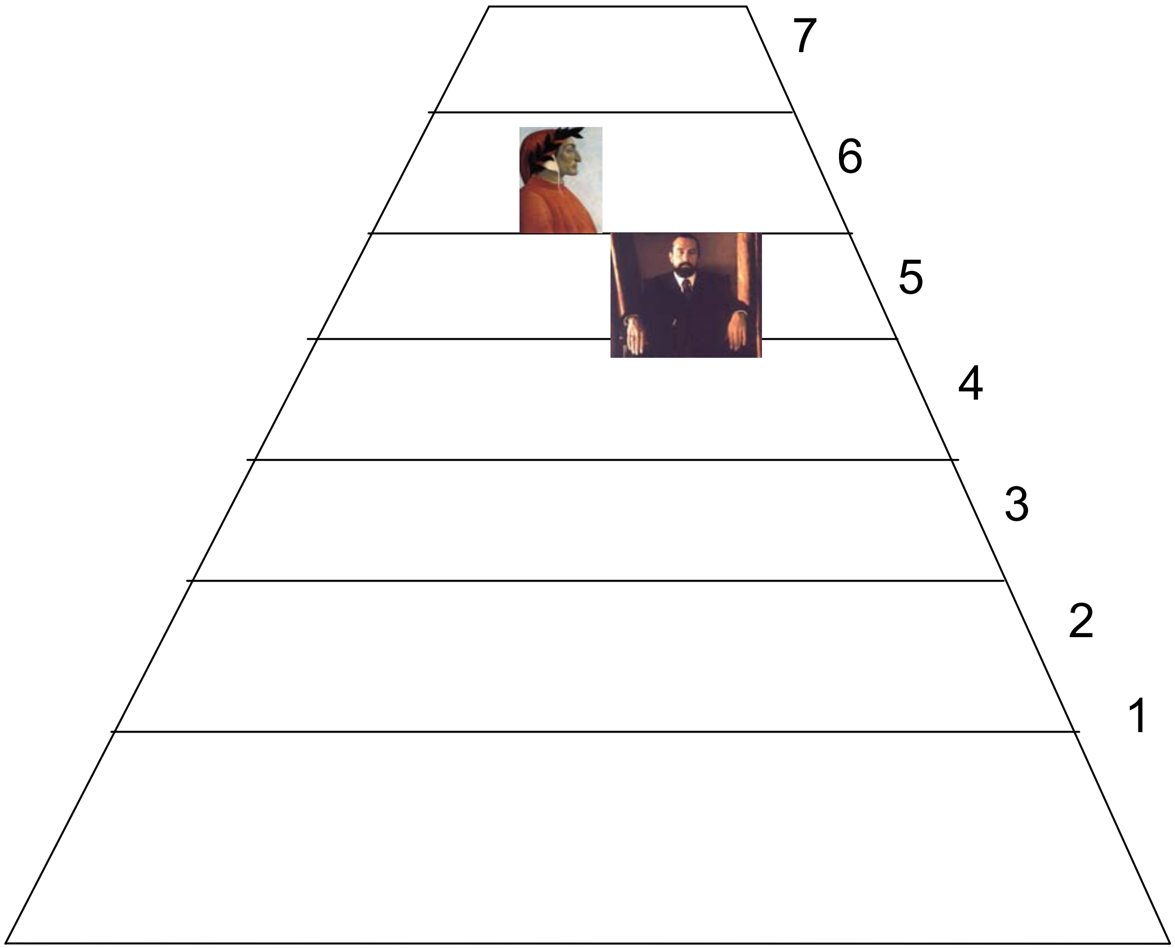
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



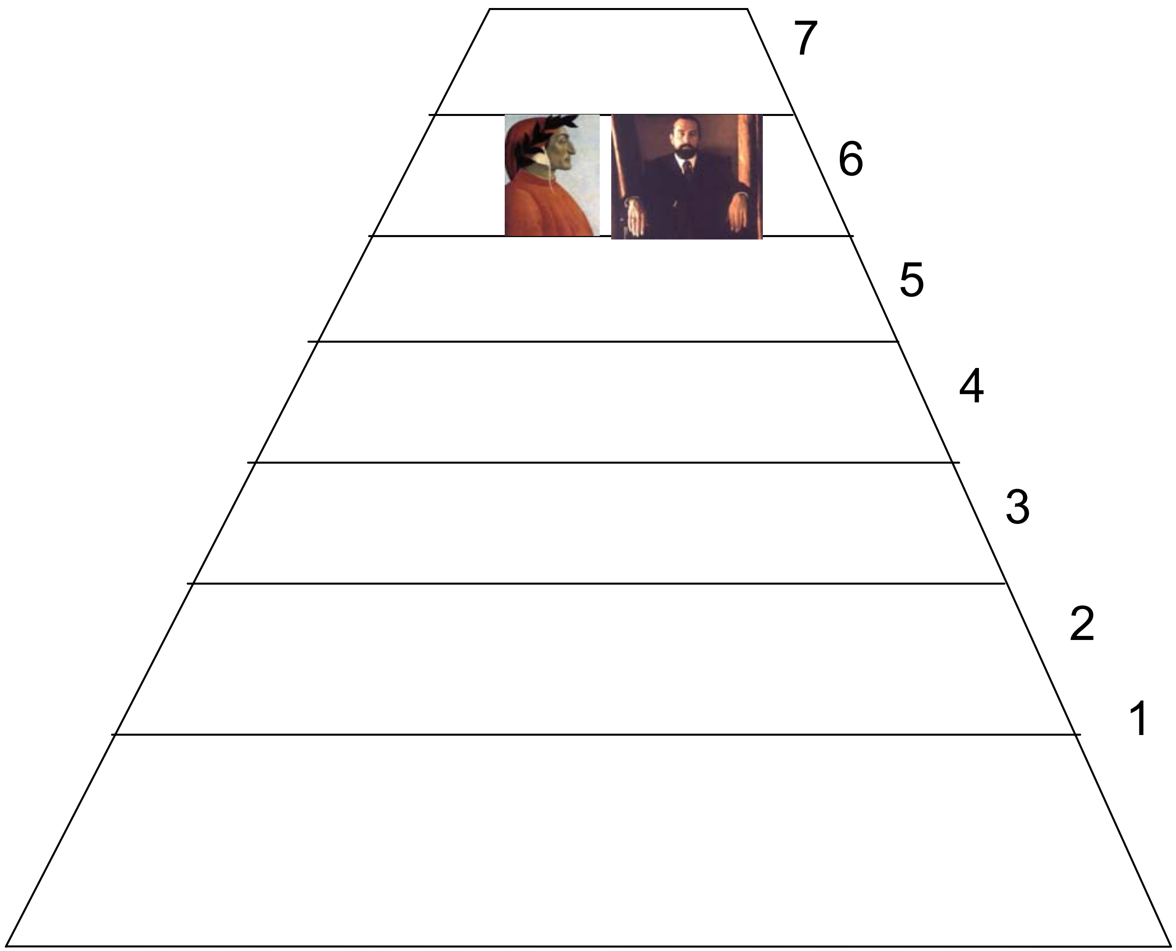
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



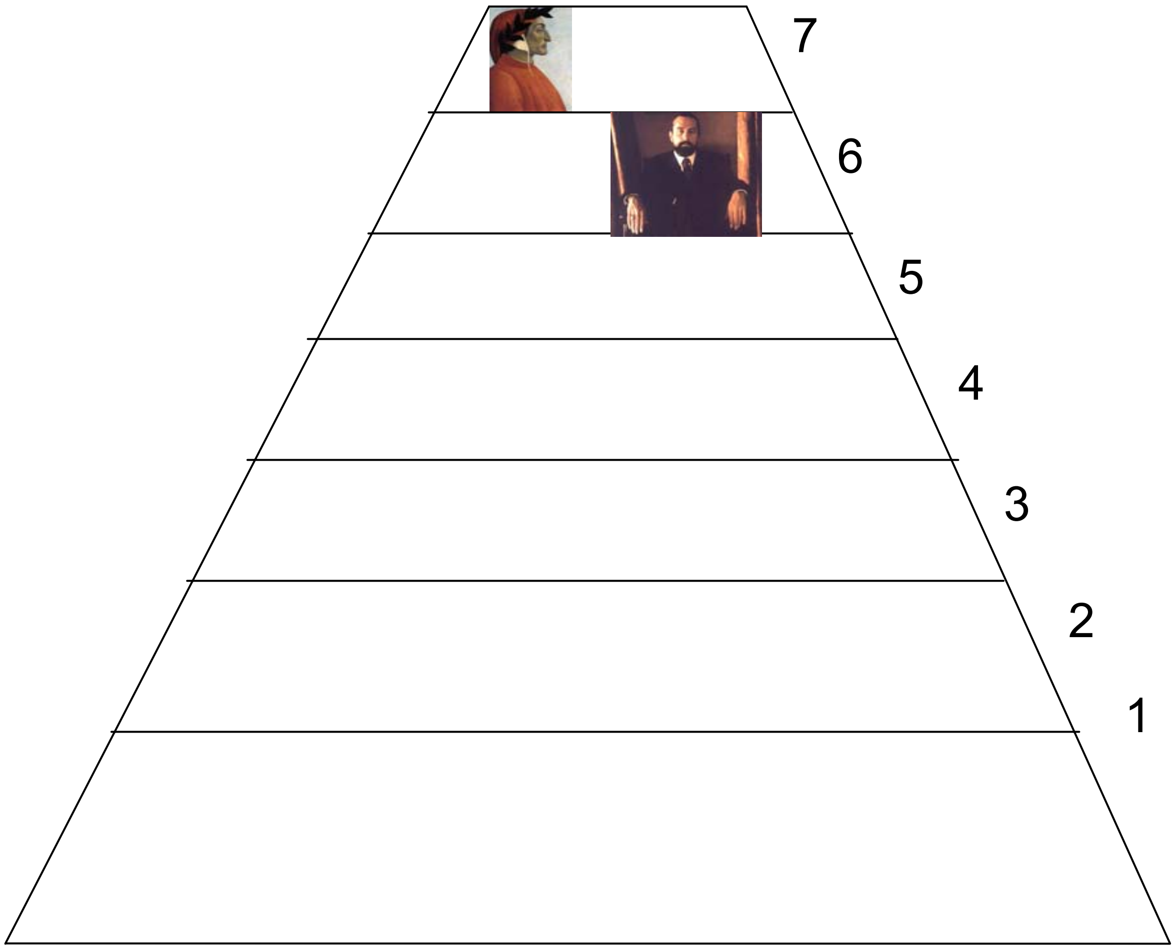
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



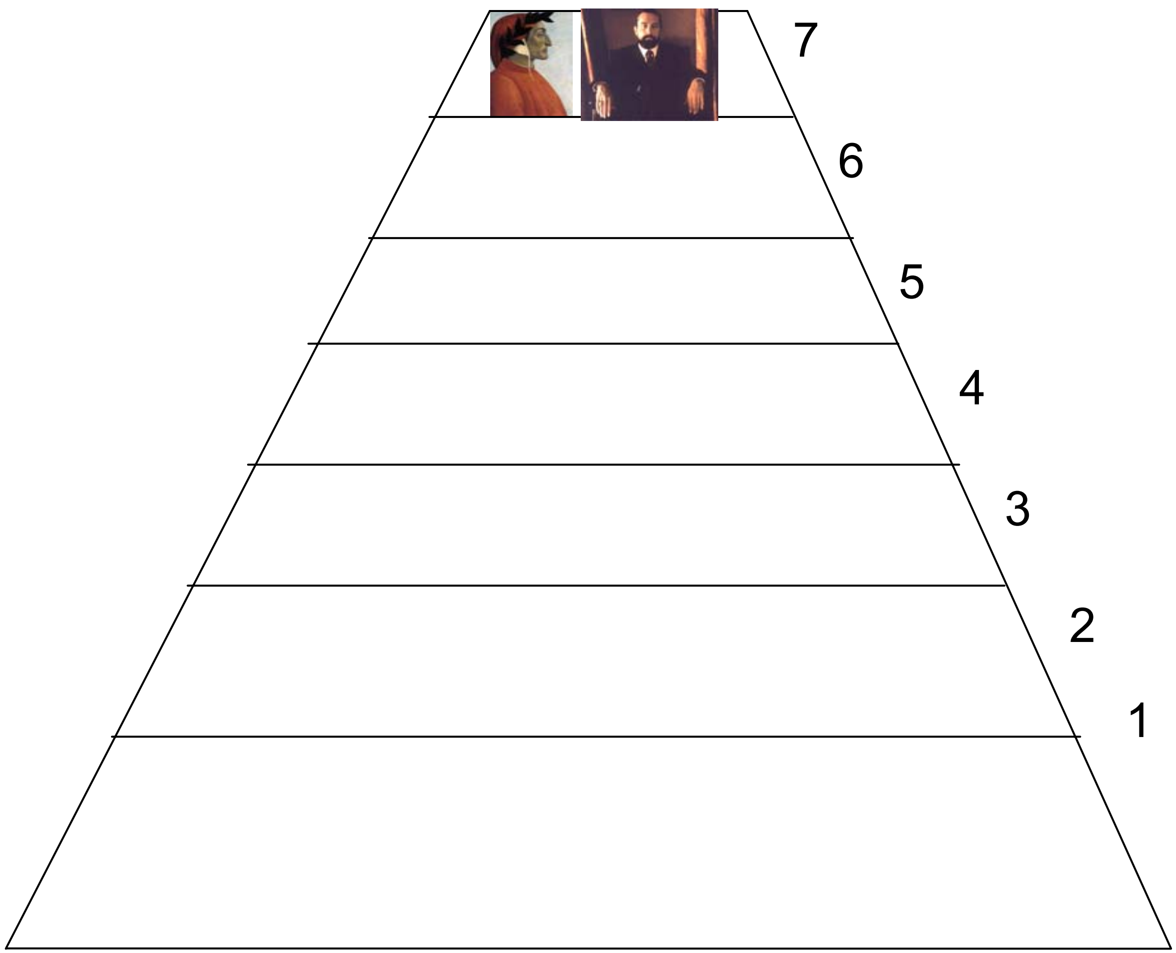
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



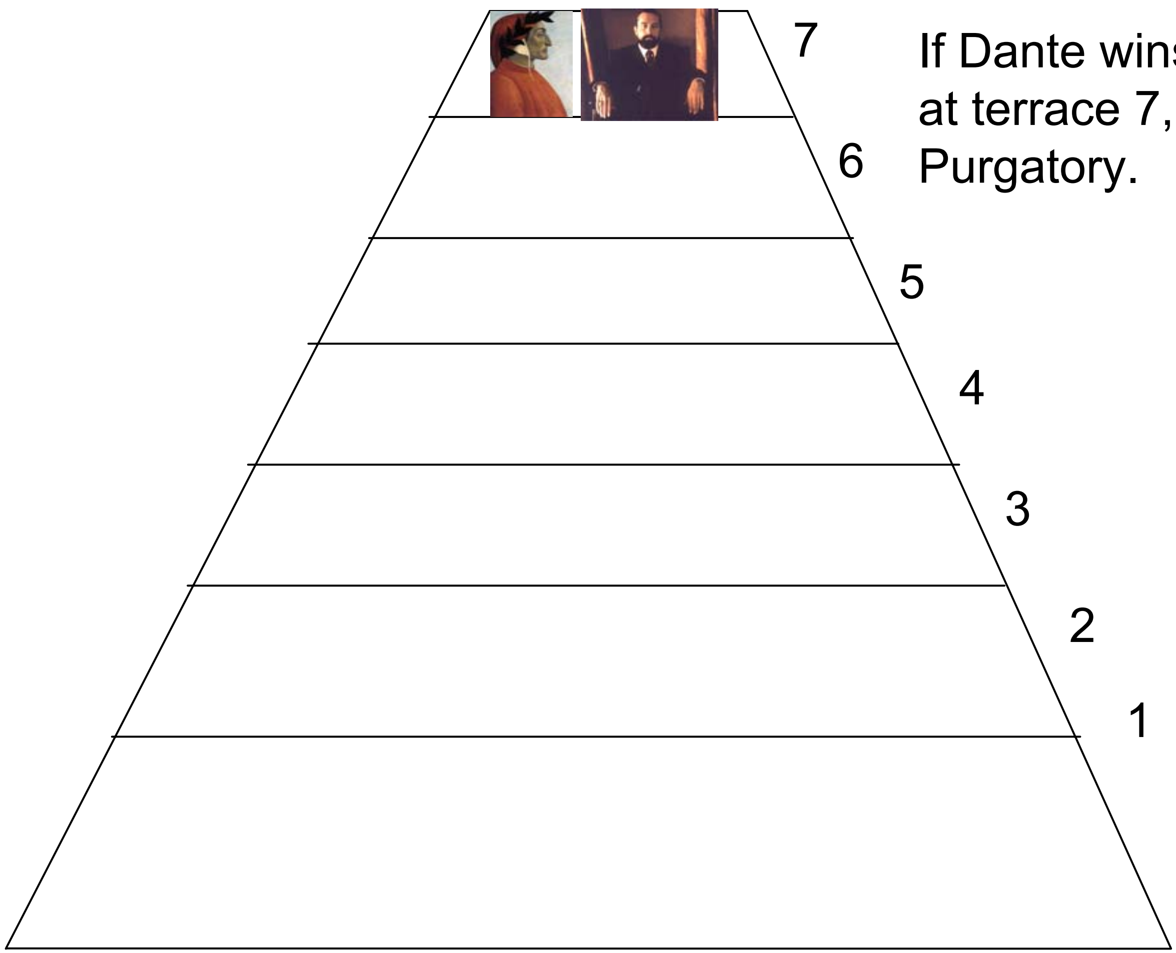
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



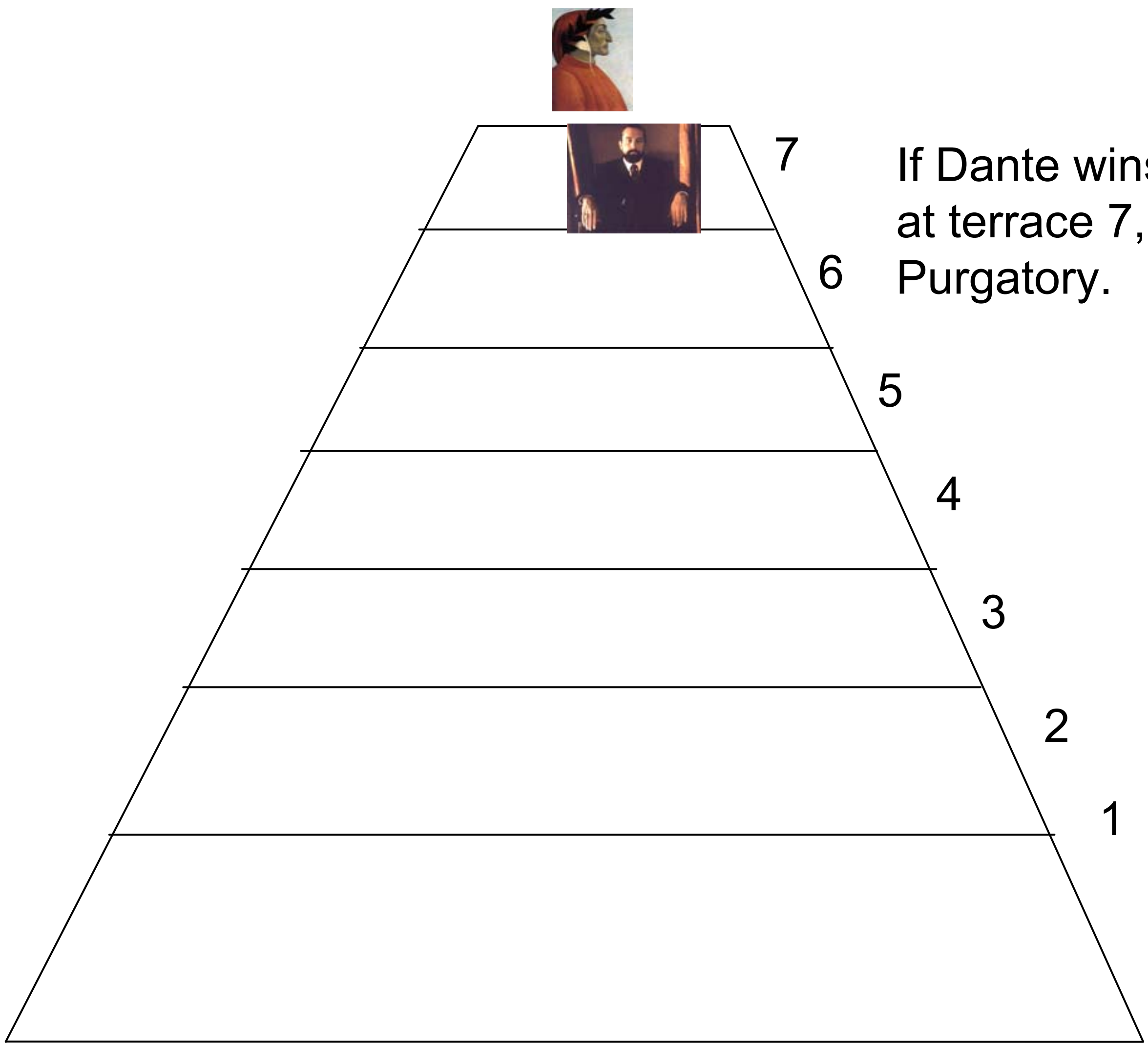
If Dante wins, he proceeds to the next terrace

Dante in Purgatory



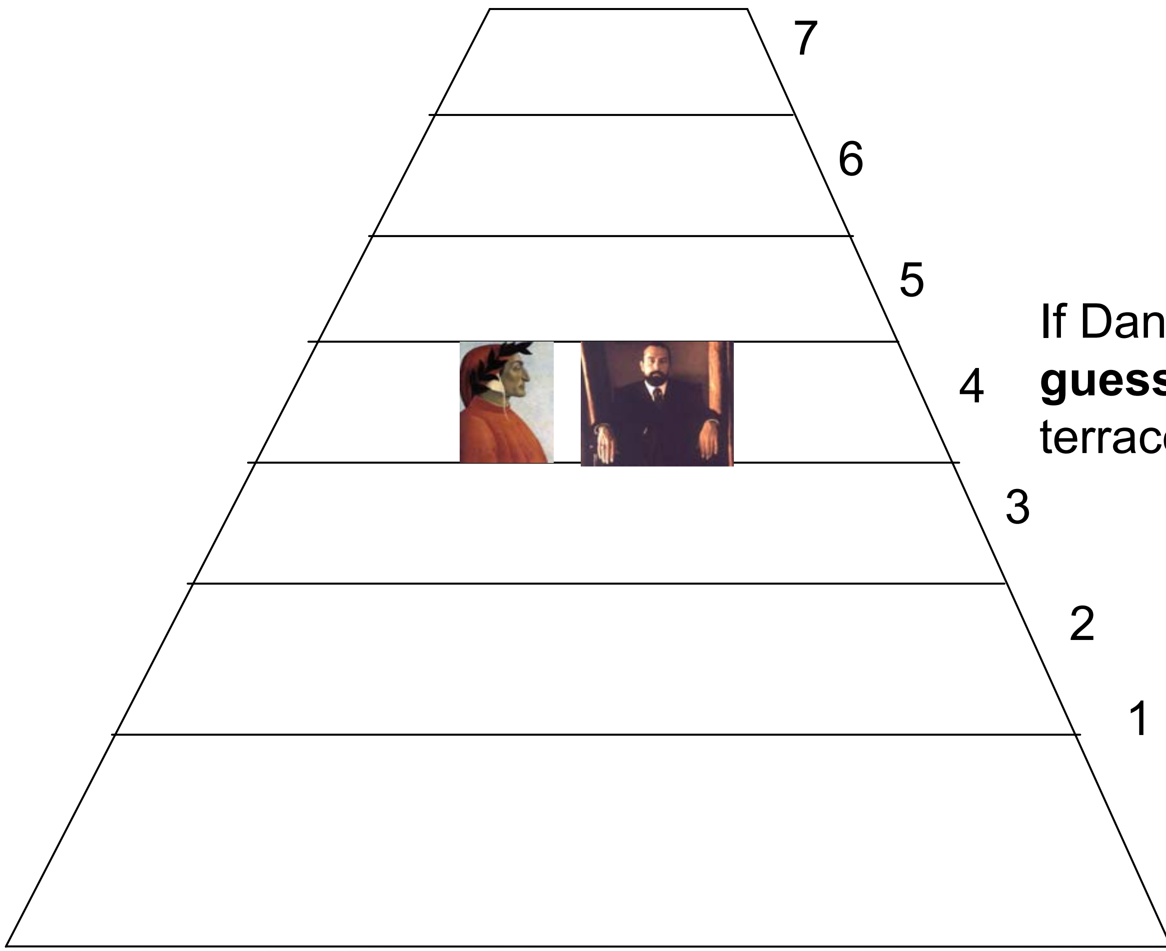
If Dante wins "Guess which hand" at terrace 7, he wins the game of Purgatory.

Dante in Purgatory



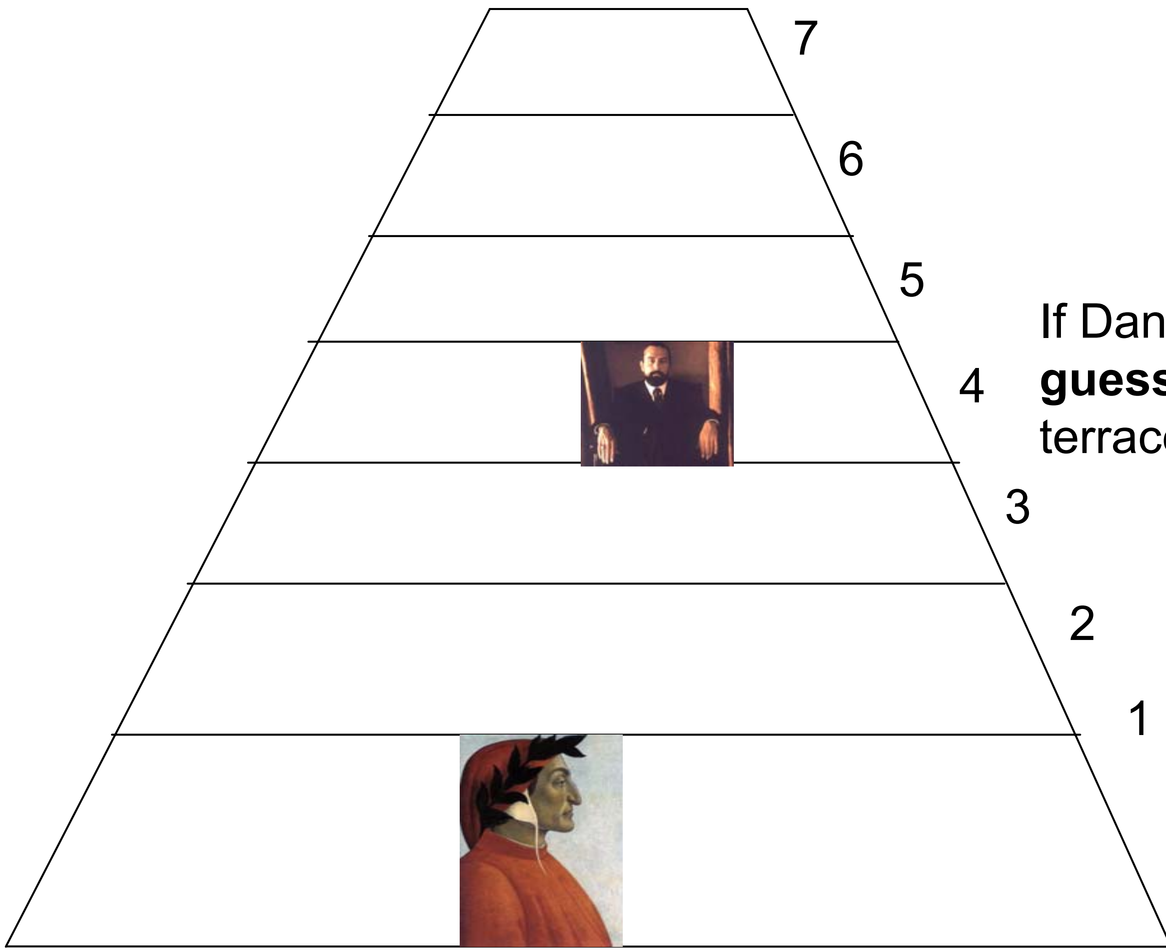
If Dante wins “Guess which hand” at terrace 7, he wins the game of Purgatory.

Dante in Purgatory



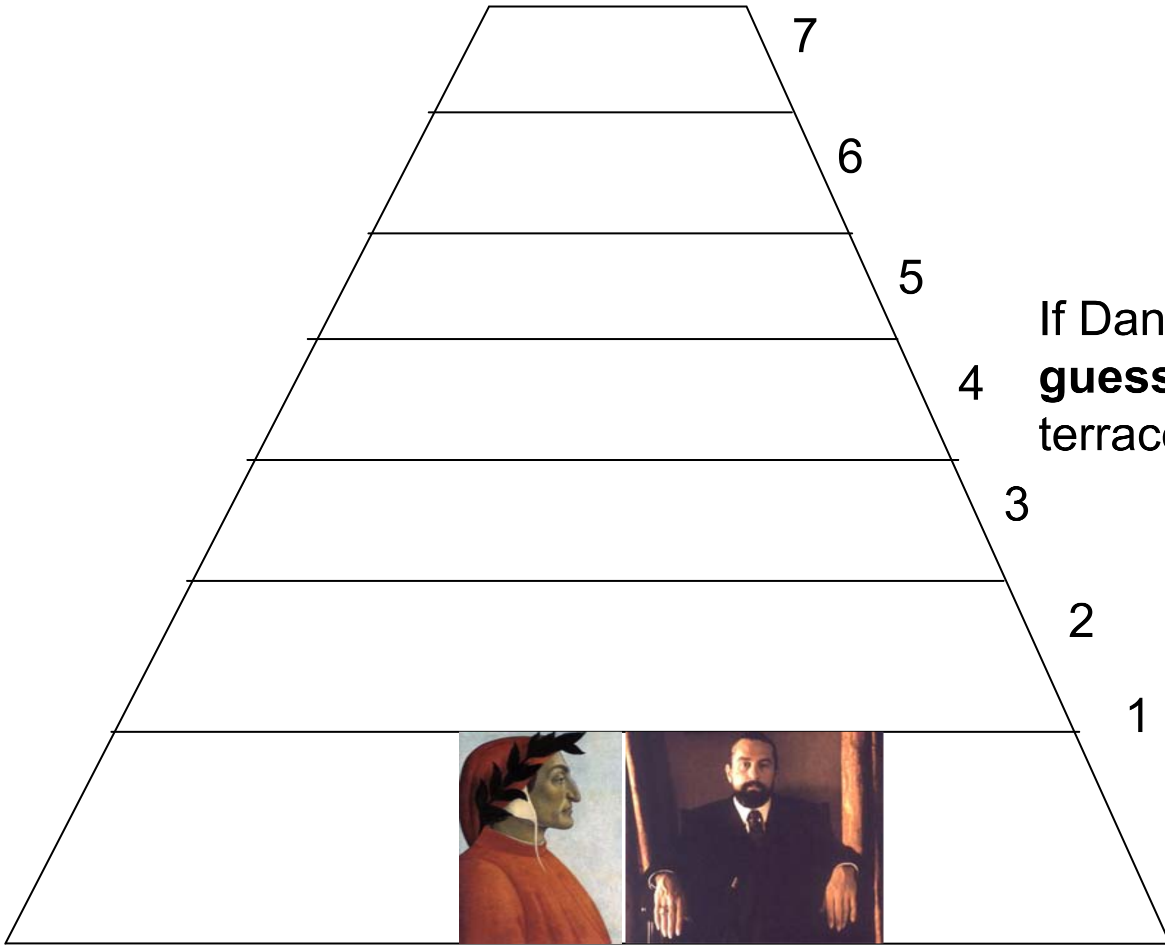
If Dante loses “Guess which hand” **guessing “Right”**, he goes back to terrace 1.

Dante in Purgatory



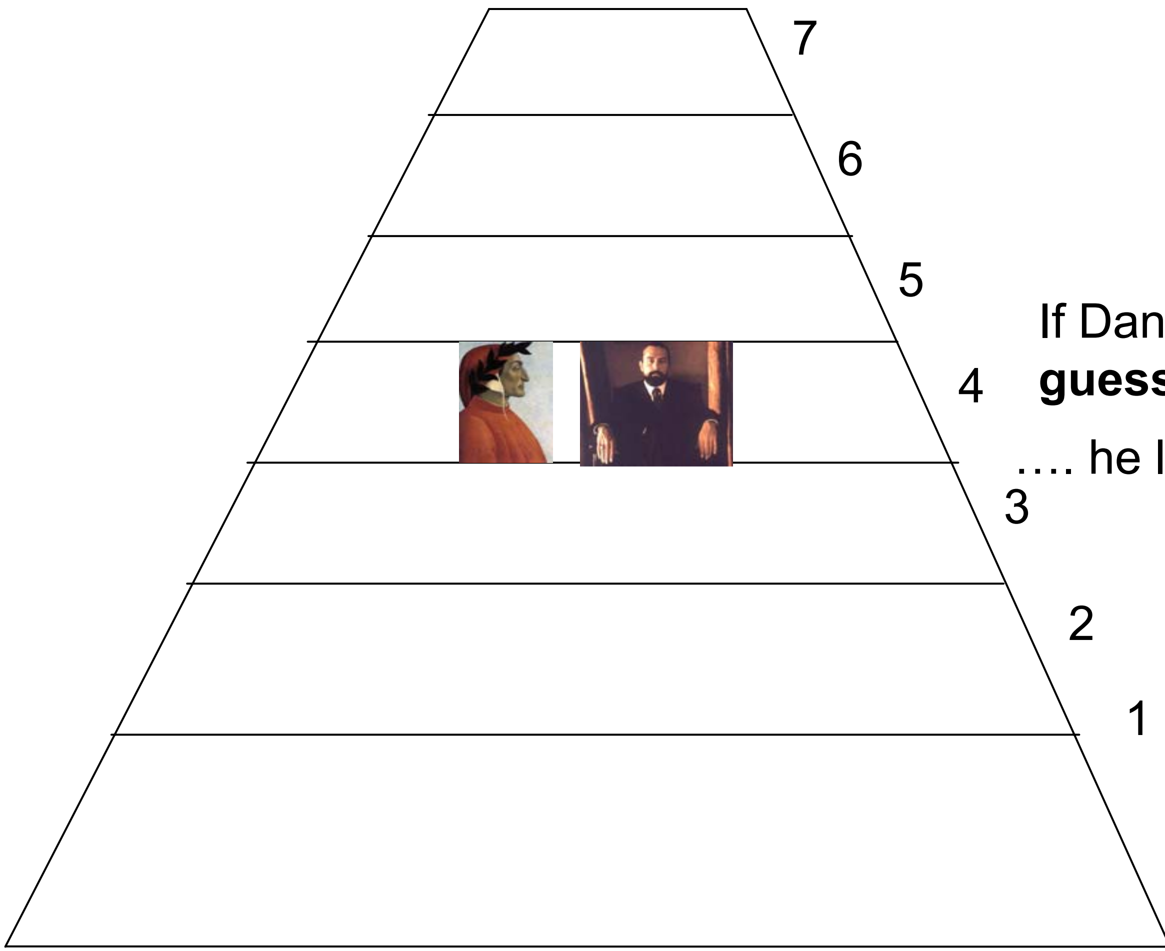
If Dante loses “Guess which hand” **guessing “Right”**, he goes back to terrace 1.

Dante in Purgatory



If Dante loses “Guess which hand” **guessing “Right”**, he goes back to terrace 1.

Dante in Purgatory



If Dante loses “Guess which hand” **guessing “Left”**….

…. he loses the game of Purgatory!!!!

Dante in Purgatory

- Is there is a strategy for Dante so that he is guaranteed to win the game of Purgatory with probability at least 90%?
  - Yes.

A bit surprising – when Dante wins, he has guessed correctly which hand seven times in a row!

Apply algorithm of de Alfaro, Henzinger and Kupferman

- How long can Lucifer confine Dante to Purgatory if Dante plays by such a strategy?
  - 10^{55} years.

Games considered

- Two-player, zero-sum, finite state, ***infinite duration*** games.

  ↑ **Sorry….**

  - **Deterministic graphical games**; DGGs (Awari-like games).

  - **Simple stochastic games**; SSGs (Backgammon-like games).

  - **Concurrent reachability games**; CRGs (Poker-tournament-like games) .

Zero-sum games vs. non-zero sum

- For two-player zero-sum games,
  Nash equilibria = (maximin, minimax)
  Stability in presence of rationality = **Guarantees**

- For non-zero sum games, not so… Solution concepts are concerned solely with stability when rational agents interact, ***not*** with guarantees.

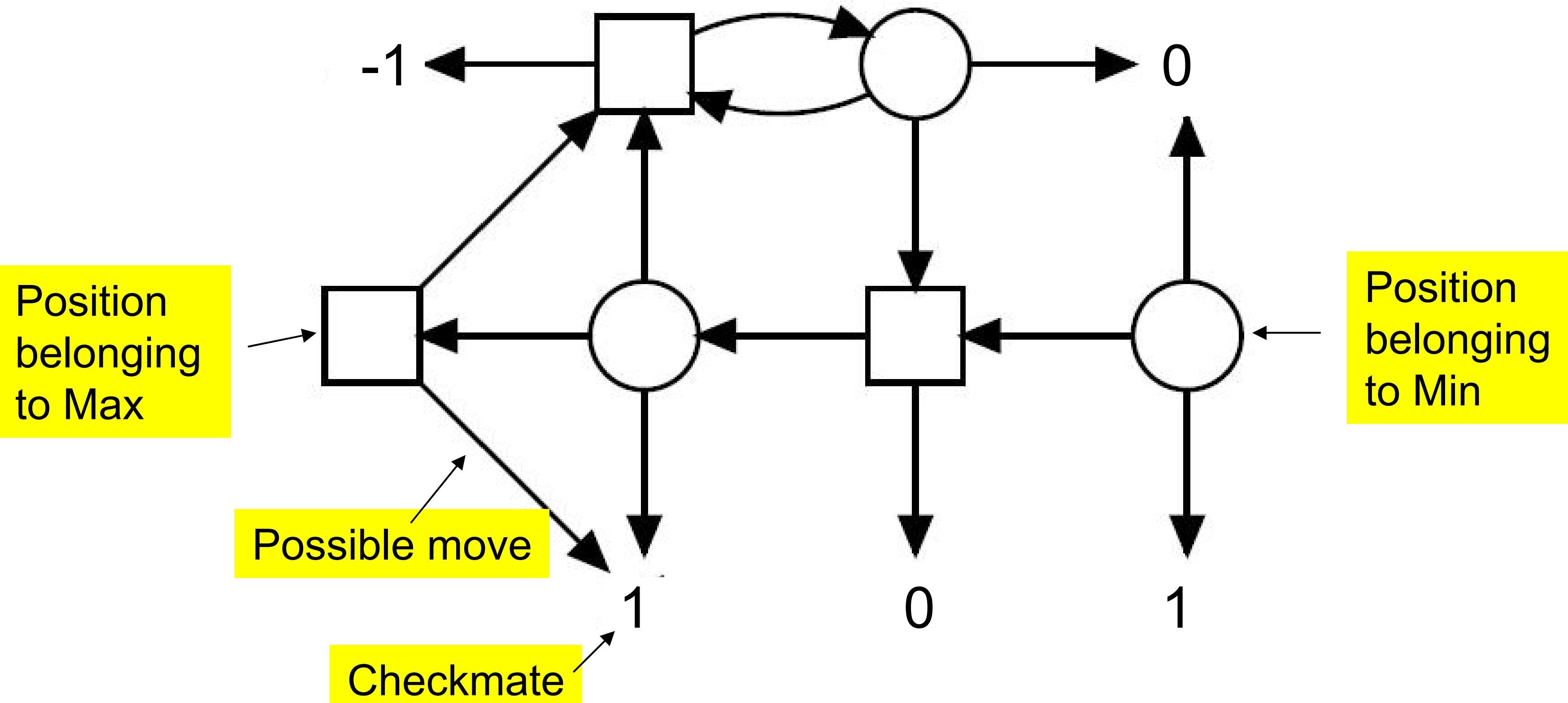
- Stability is not such a bad property to aim for… Example: Miltersen, Nielsen, Triandopoulos: *Privacy-enhancing auctions using rational cryptography*, CRYPTO'09.

Credits

- Daniel Andersson, Kristoffer Arnsfelt Hansen, Peter Bro Miltersen, Troels Bjerre Sørensen:
  Deterministic Graphical Games Revisted (CiE'08).
- Vladimir Gurvich, Peter Bro Miltersen:
  On the computational complexity of stochastic mean-payoff games (Arxiv).
- Daniel Andersson, Peter Bro Miltersen:
  The complexity of solving stochastic games on graphs (in review)
- Kristoffer Arnsfelt Hansen, Michal Koucky, Peter Bro Miltersen:
  Winning concurrent reachability games requires doubly-exponential patience (LICS'09).

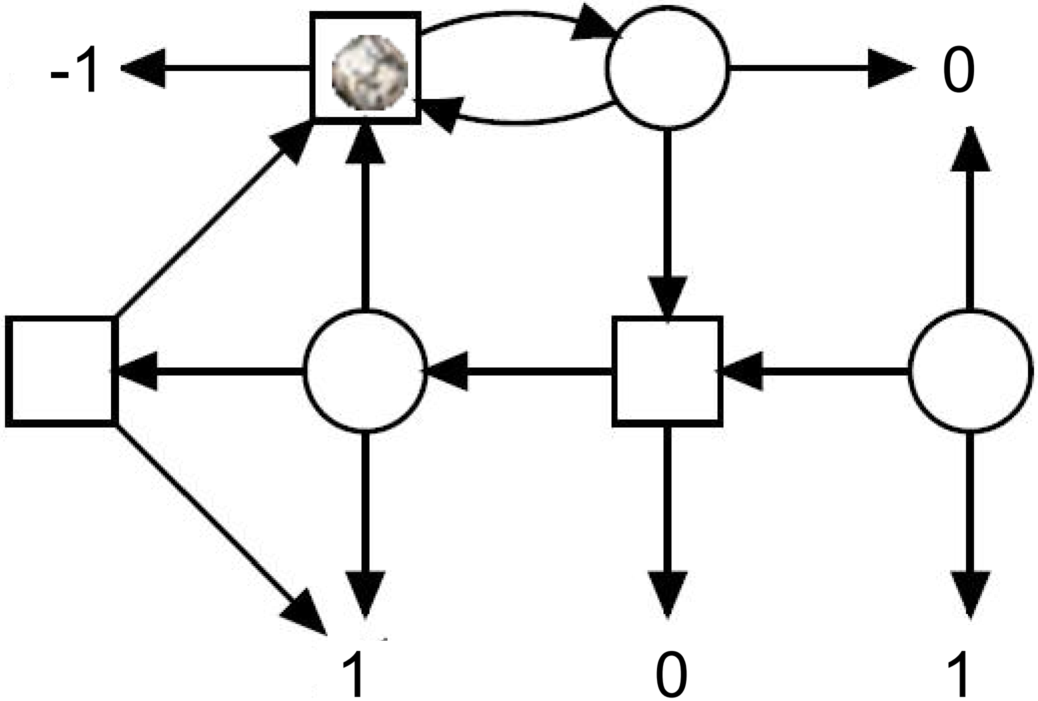
Deterministic Graphical Games

- “Chess-like games”



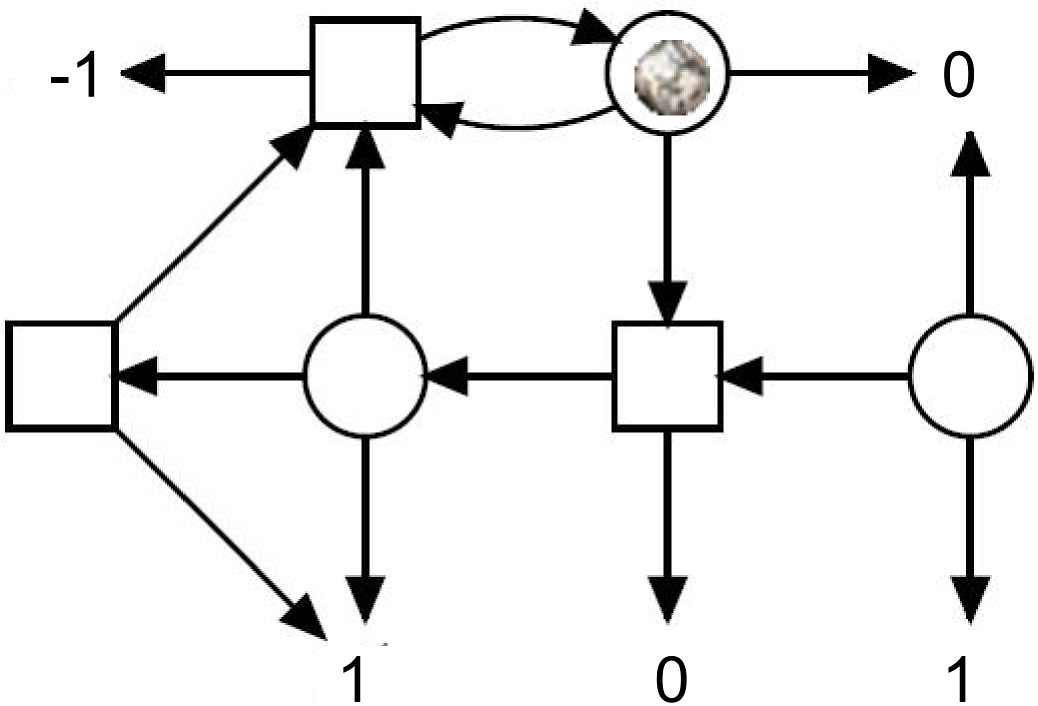
Deterministic Graphical Games

- “Chess-like games”



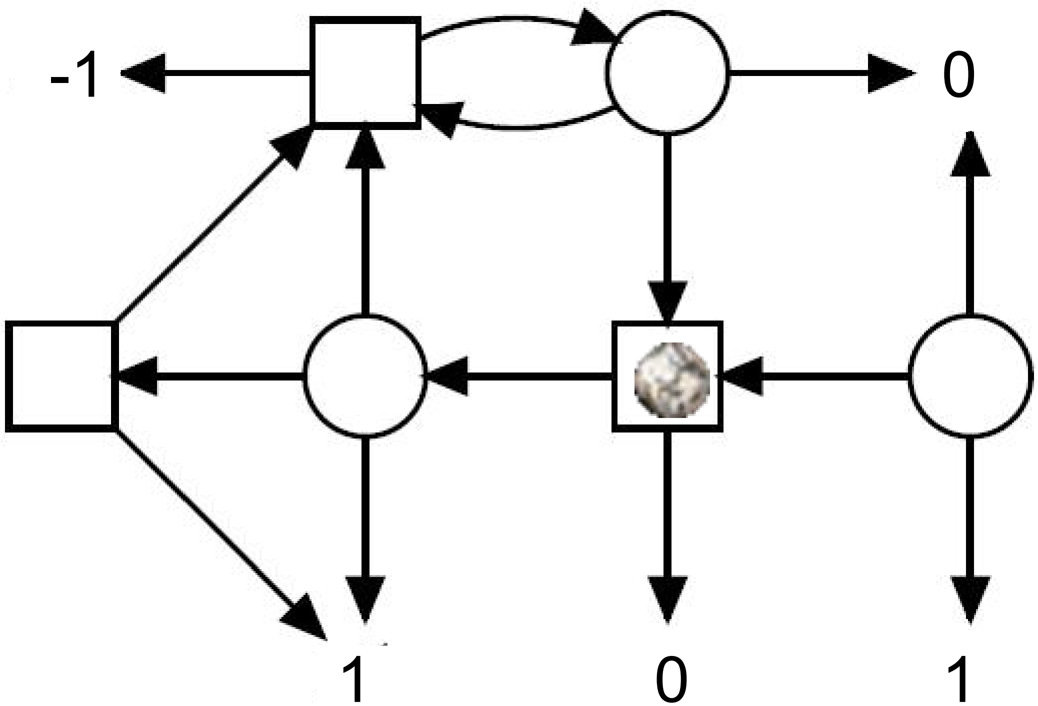
Deterministic Graphical Games

- "Chess-like games"



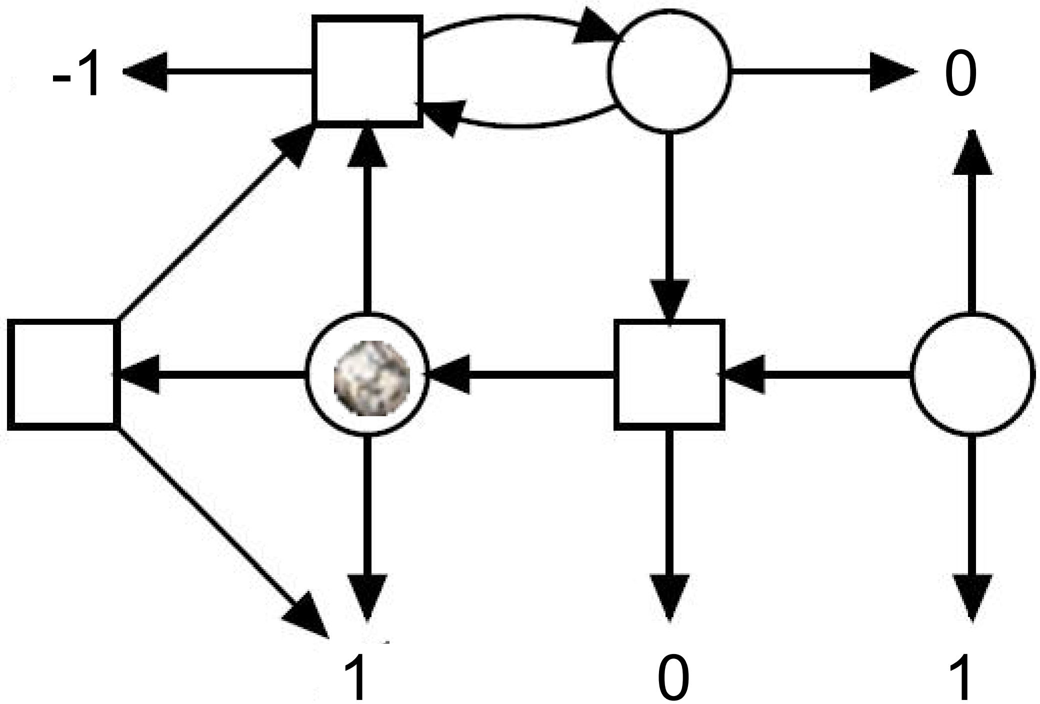
Deterministic Graphical Games

- "Chess-like games"



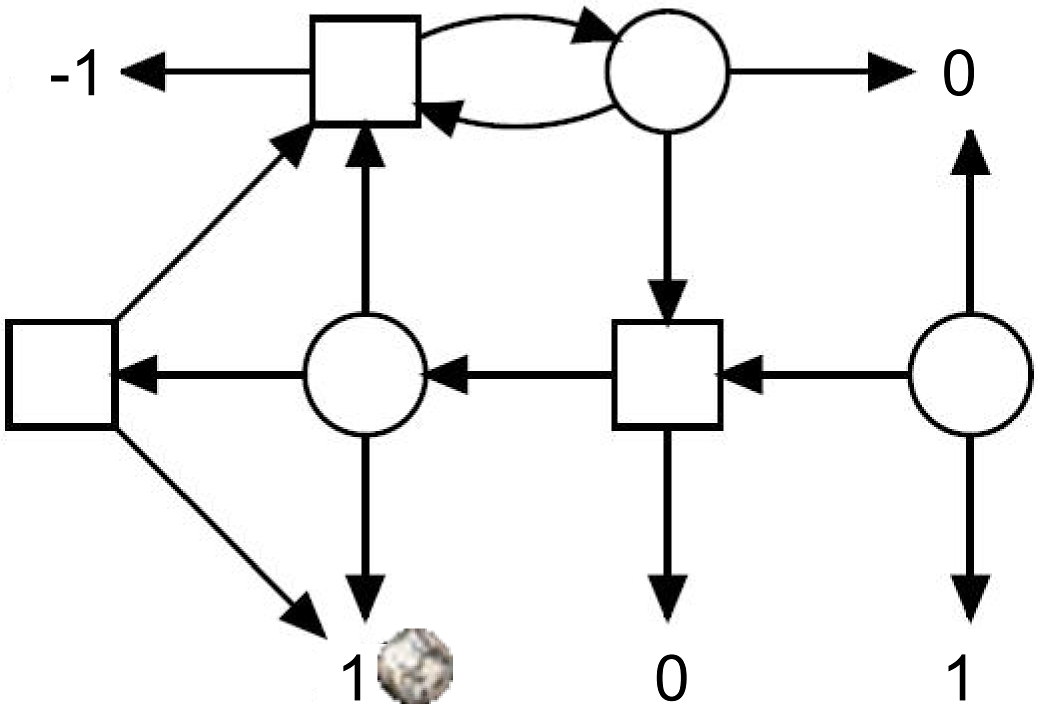
Deterministic Graphical Games

- "Chess-like games"



Deterministic Graphical Games

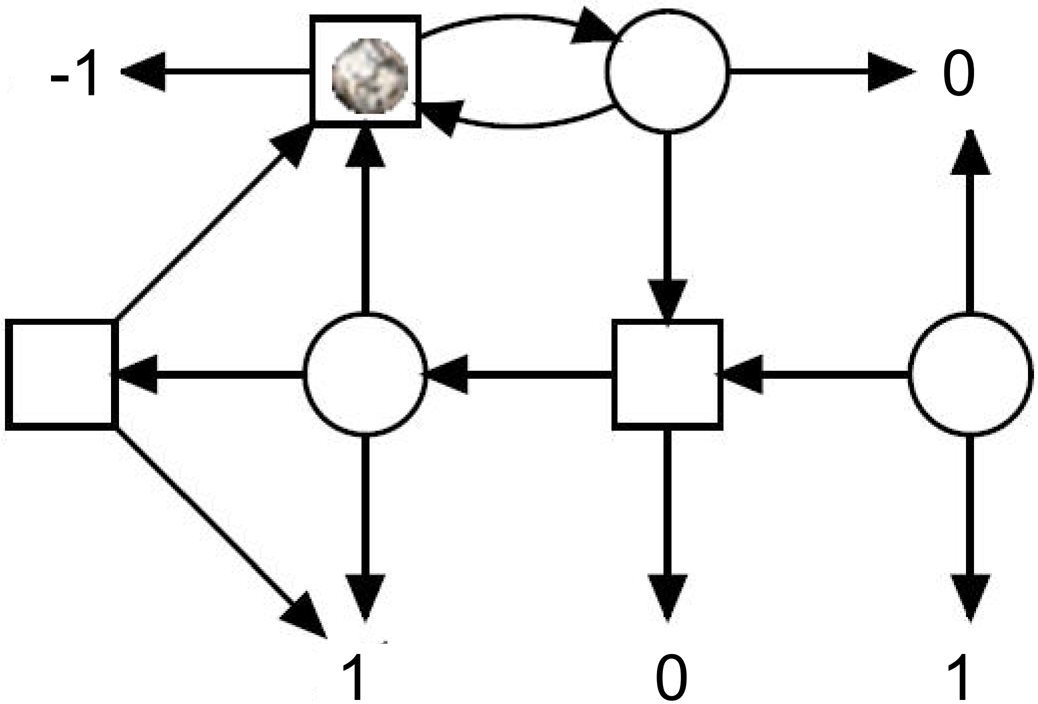
- “Chess-like games”



Player 1 (□) wins and Player 2 (○) loses.
Payoff is 1 to □ and -1 to ○.

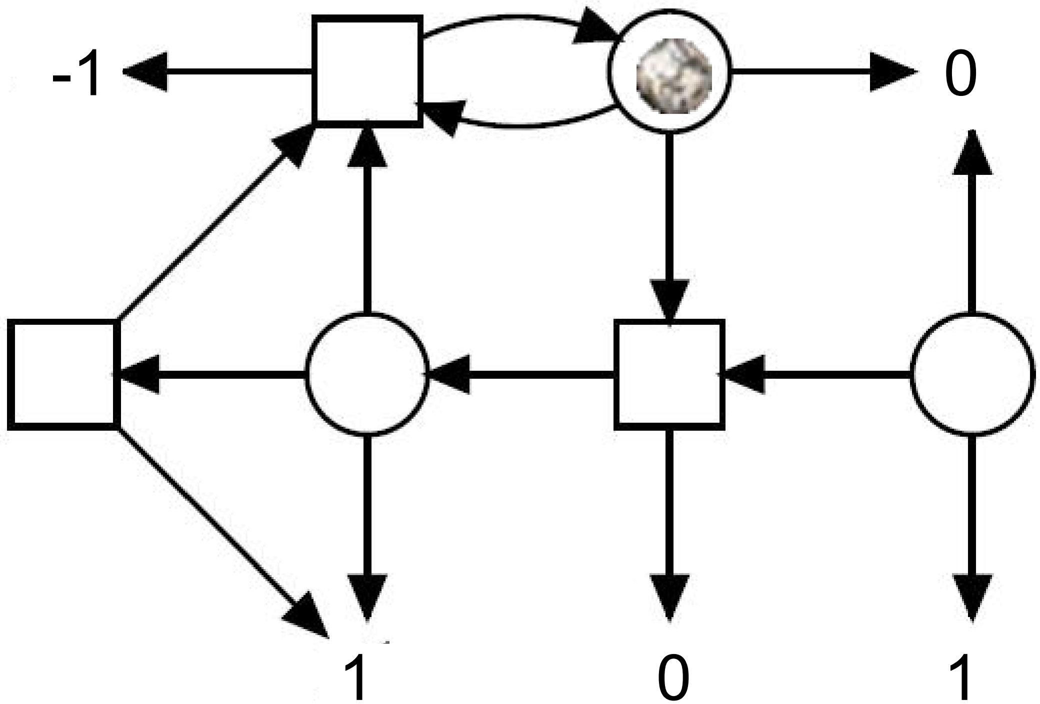
Deterministic Graphical Games

- “Chess-like games”



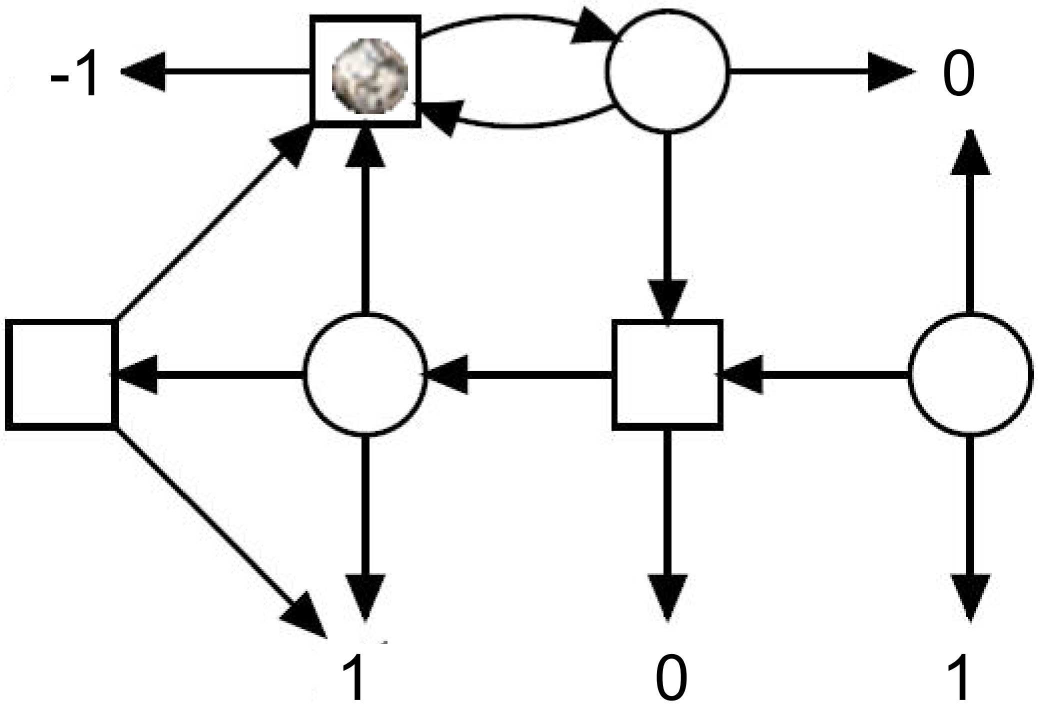
Deterministic Graphical Games

- "Chess-like games"



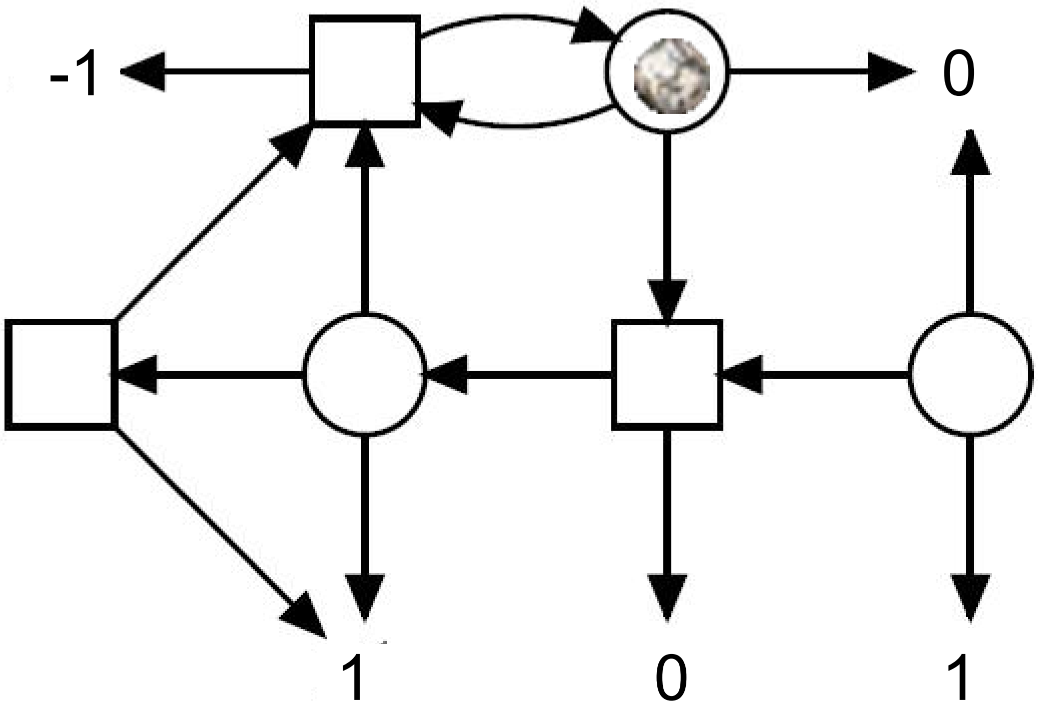
Deterministic Graphical Games

- “Chess-like games”



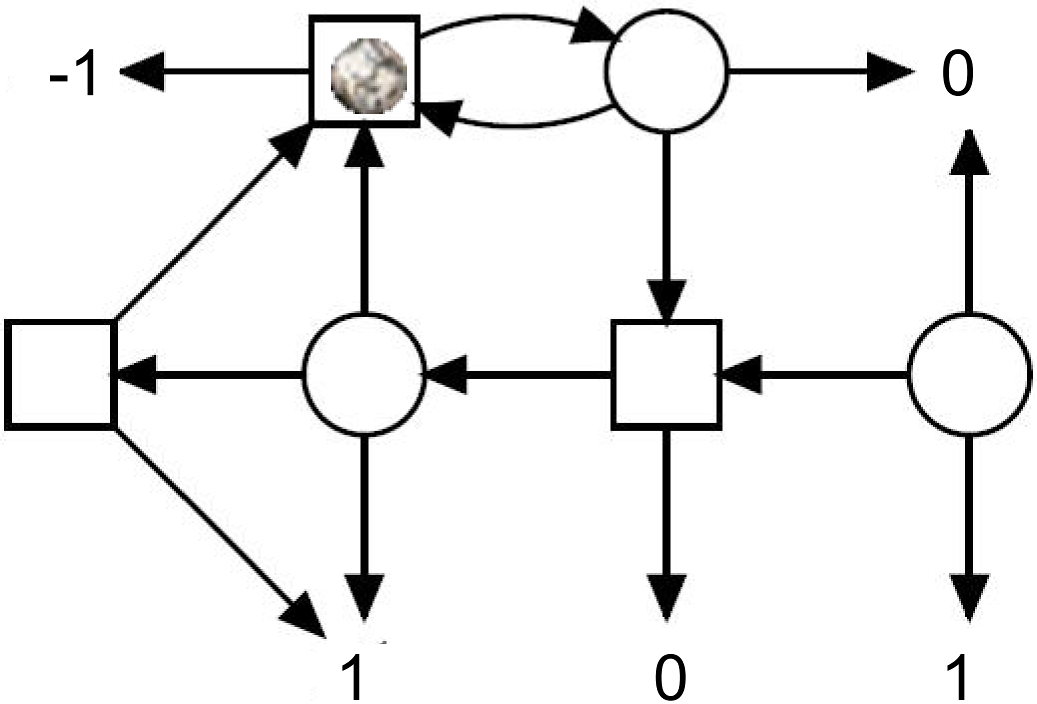
Deterministic Graphical Games

- "Chess-like games"



Deterministic Graphical Games

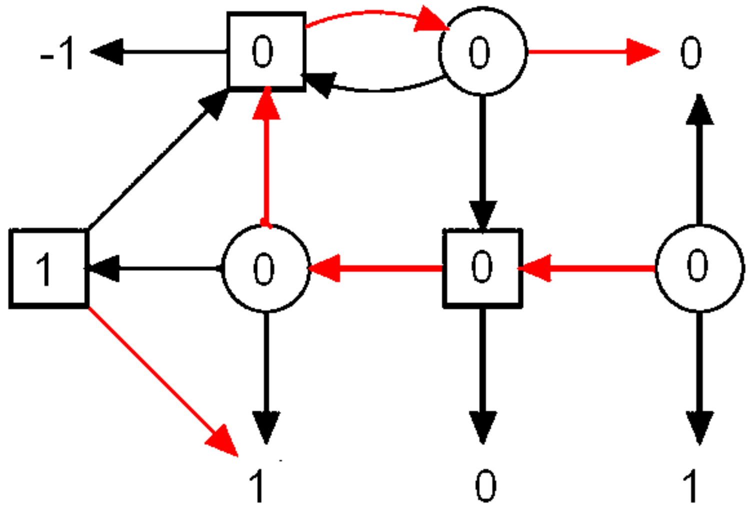
- “Chess-like games”



Values and optimal strategies

- Each position in a chess-like game has a ***value*** (Zermelo, 1911 & König, 1927).
- Each player has a ***pure positional strategy*** *guaranteeing* the value - an ***optimal*** strategy (Kalmár, 1928).

The algorithmic problem



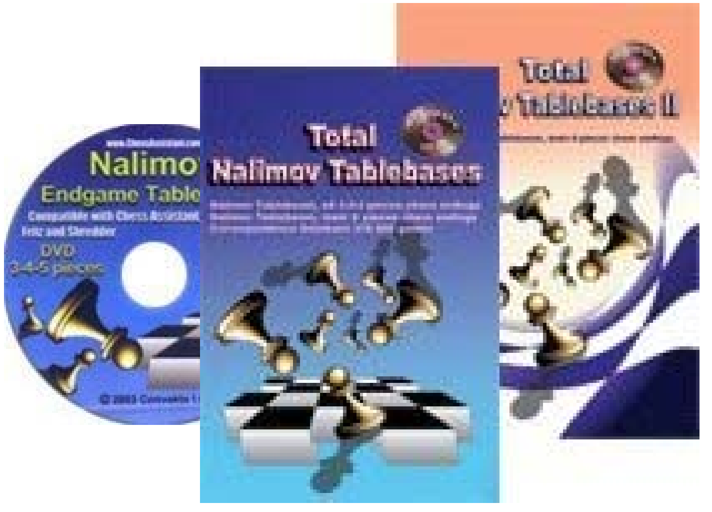
- **Solving a game**: Given an *explicit representaton* of a game, compute optimal strategies and values.

Variants

- ***Quantitatively*** solving a game – compute the value of each position.
- ***Strategically*** solving the game – compute optimal strategies.
- Strategically solving games are in general harder than solving them quantitatively….

Retrograde analysis

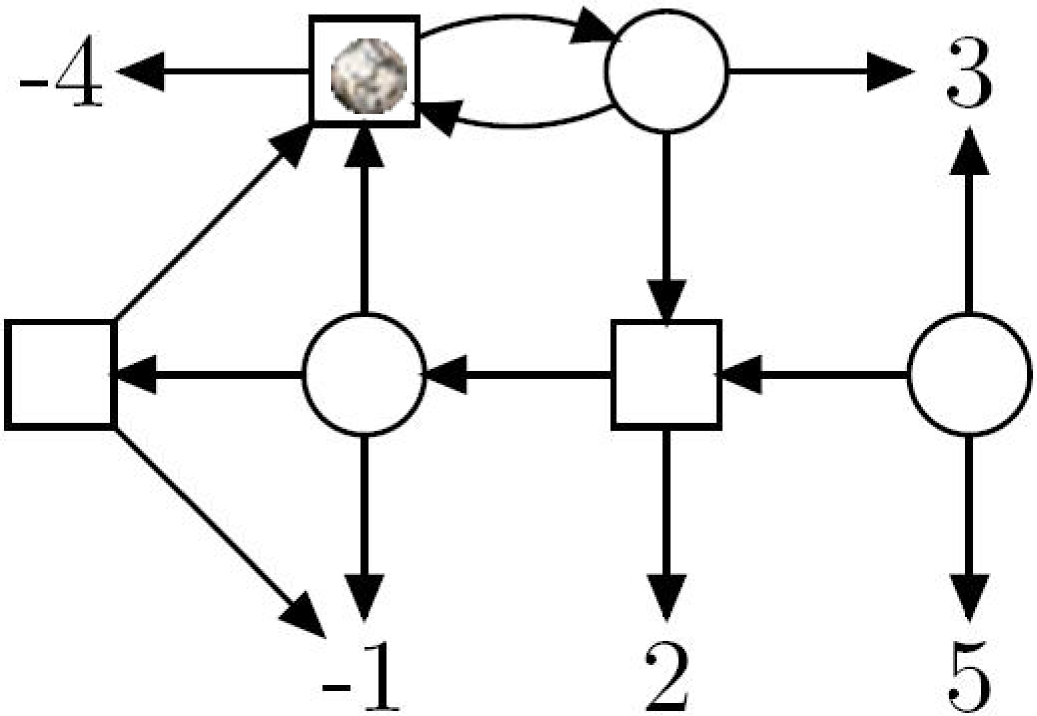
- Ströhlein 1970, crediting Knuth 1968 (the AI literature often credits Bellman 1965): Deterministic Graphical Games can be solved in linear time using *retrograde analysis.*



- *Only described (by Ströhlein as well as in subsequent literature) for games with payoffs 1,-1,0.*

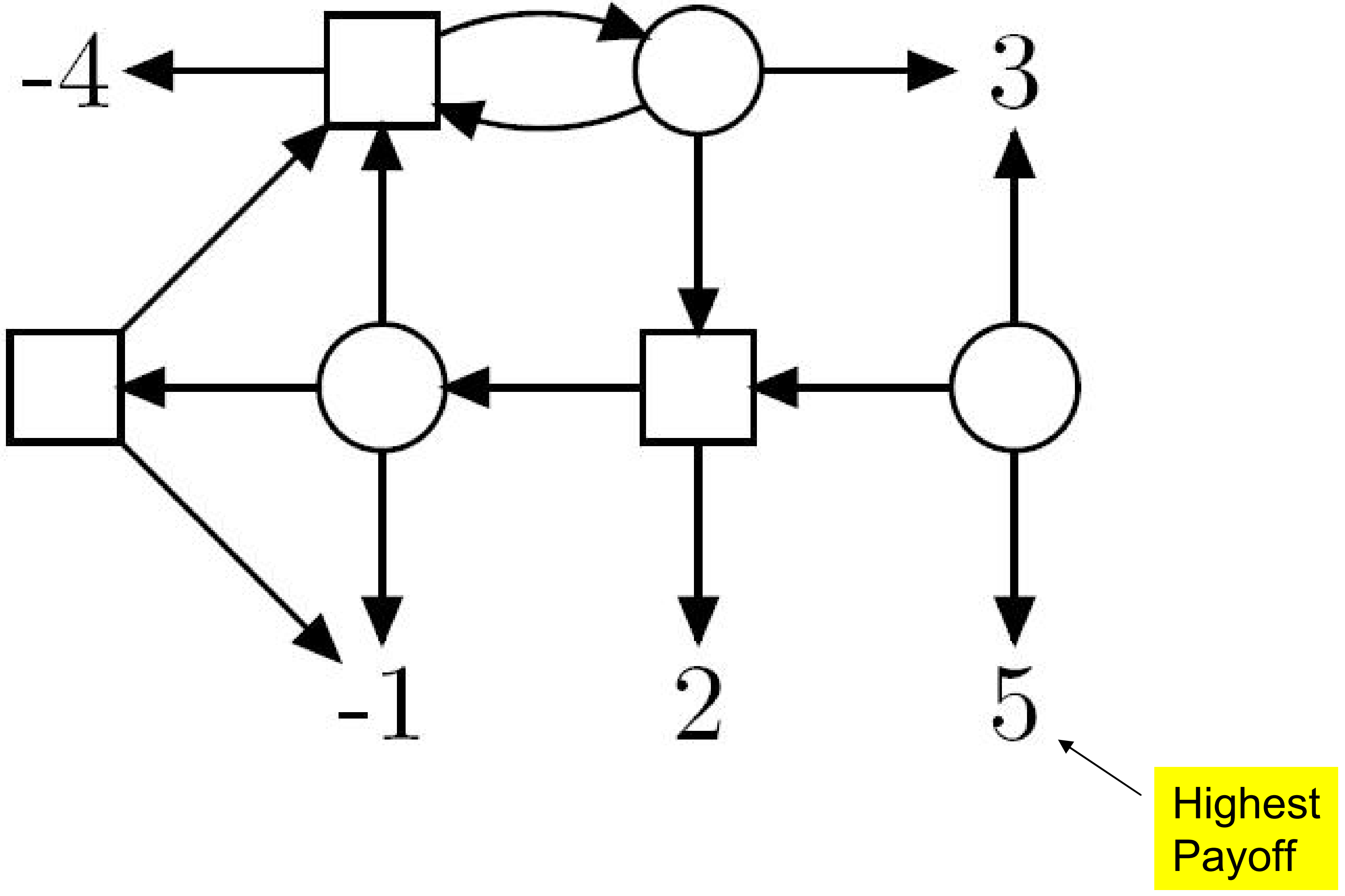
Deterministic Graphical Games

- "Awari-like games"

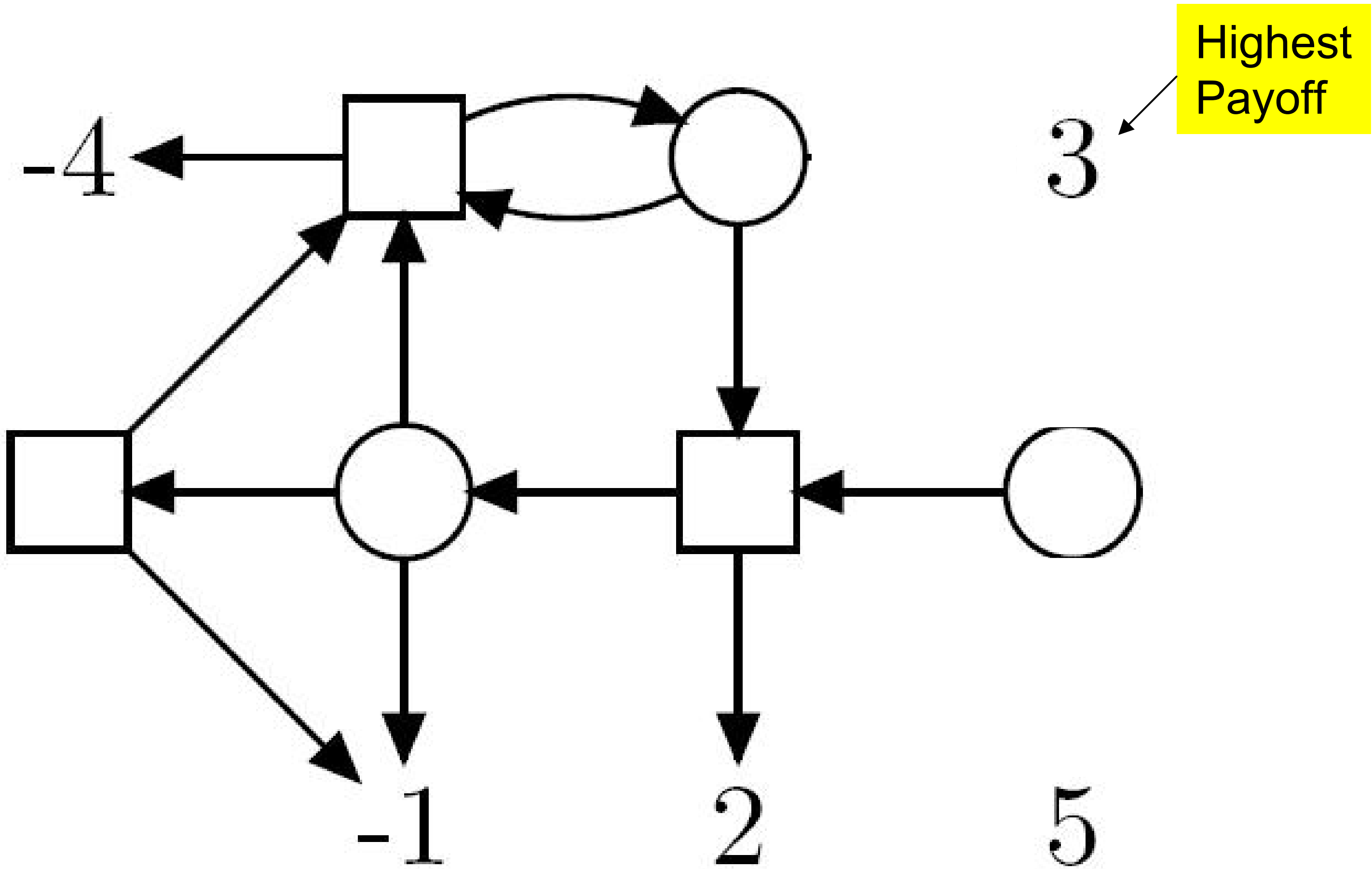


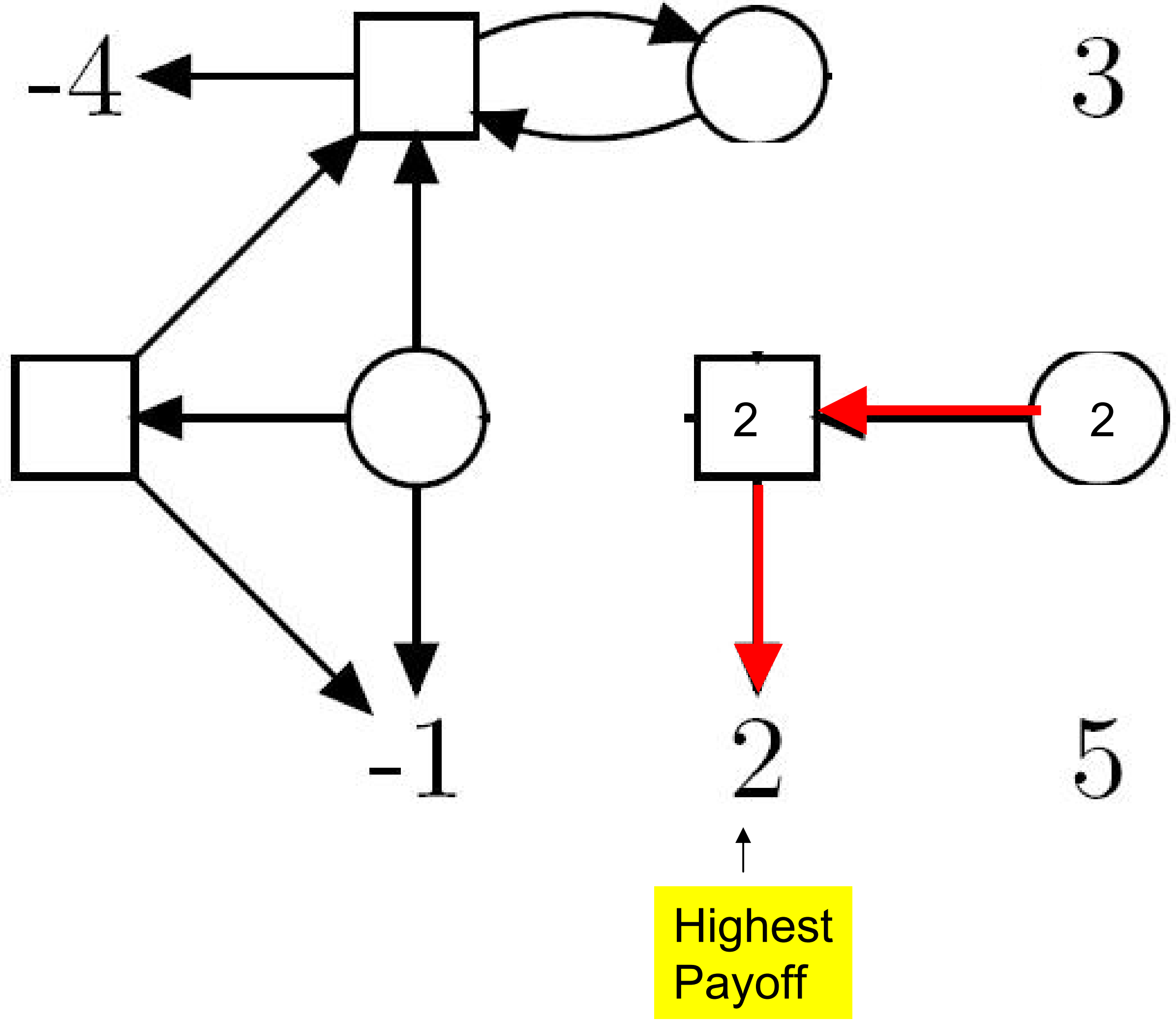
Andersson, Hansen, Miltersen, Sørensen, CiE’2008

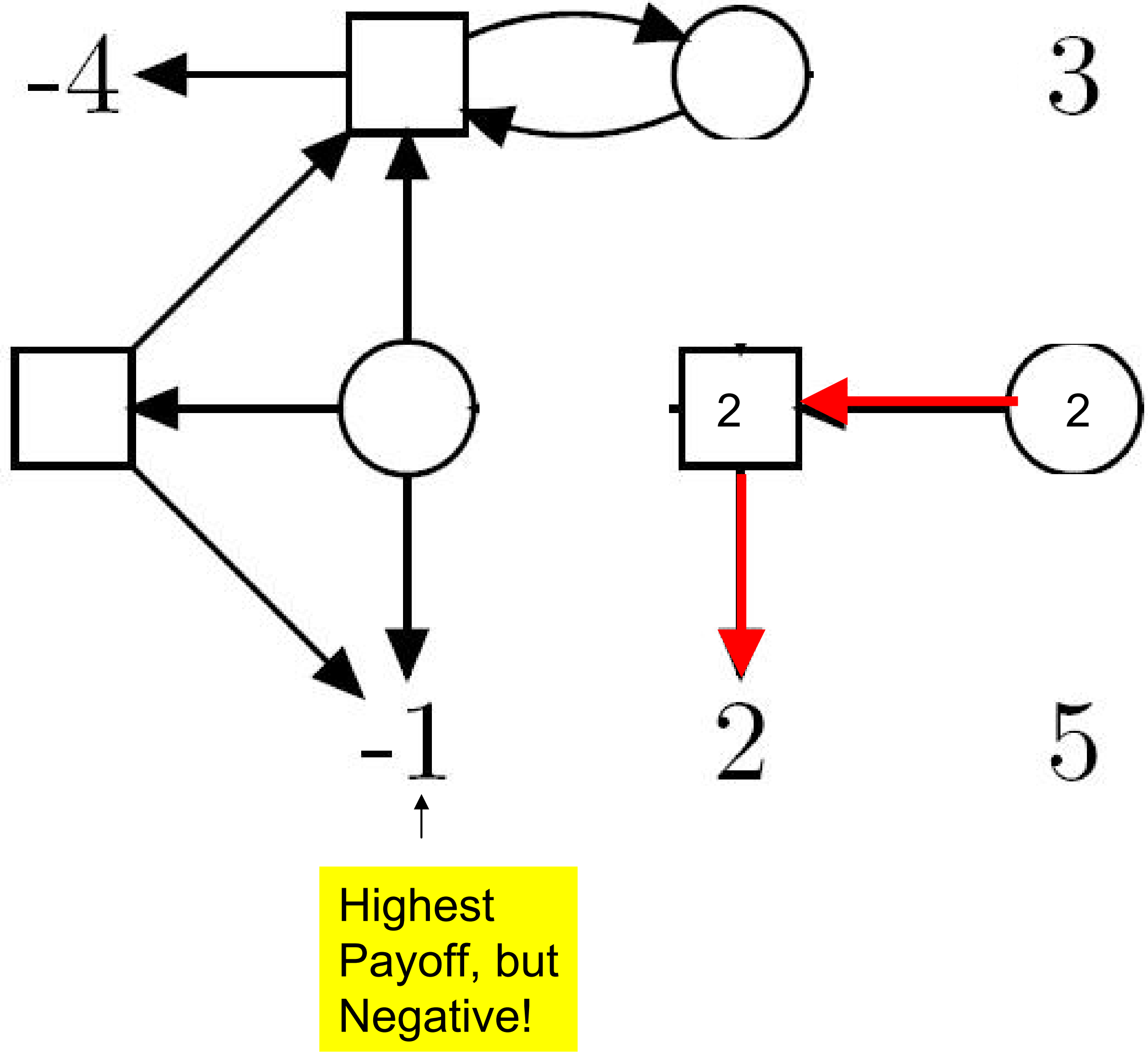
Retrograde analysis solves deterministic graphical games, *but not in linear time*. Bottleneck: ***Payoffs must be sorted***.

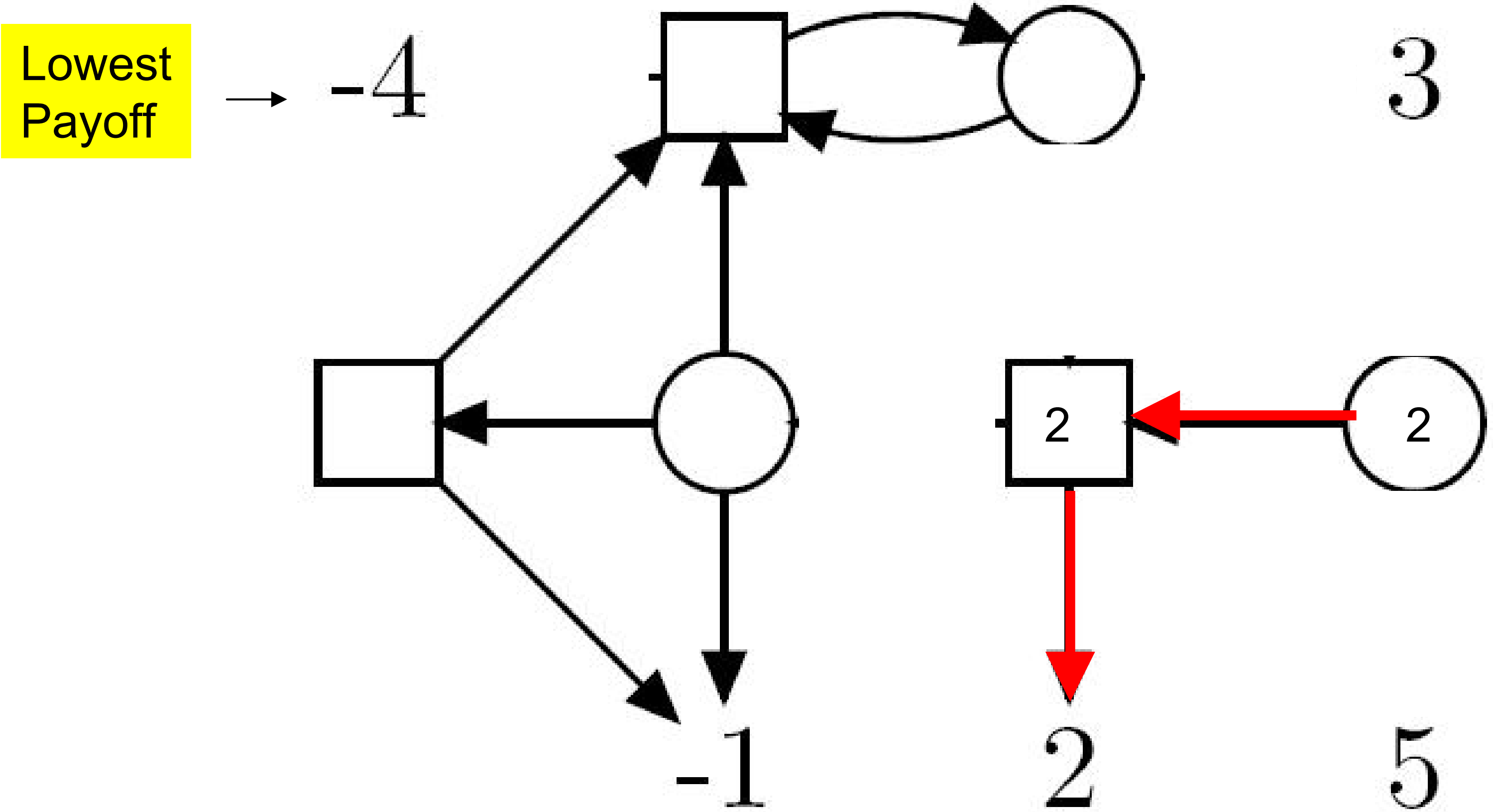


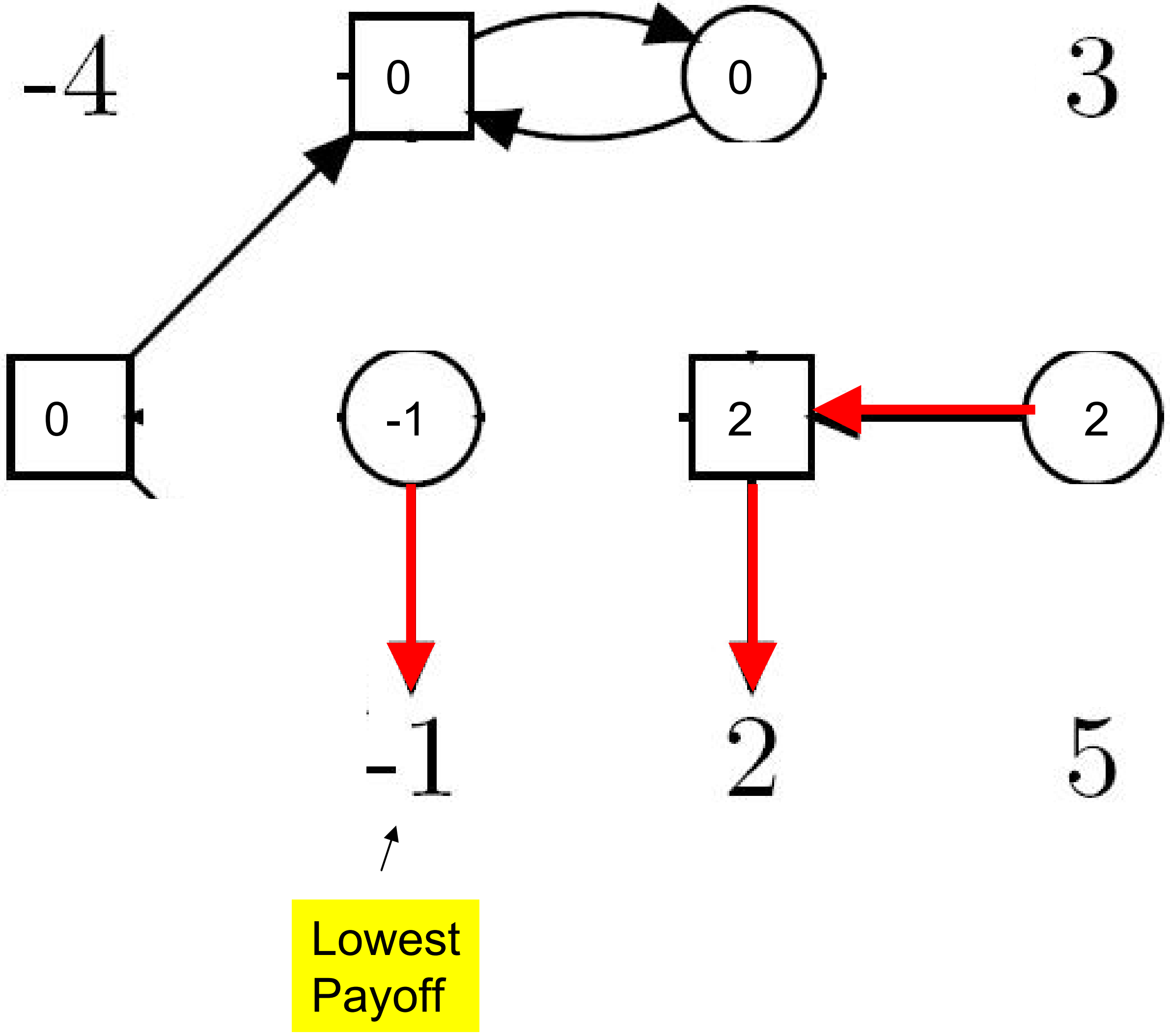
-4
3
-1
2
5
Highest Payoff


Highest
Payoff
-4
3
-1
2
5


-4
3
2
2
-1
2
5
Highest Payoff


-4
3
2
2
-1
2
5
Highest Payoff, but Negative!


Lowest Payoff
-4
3
2
2
-1
2
5


-4
0
0
3
0
-1
2
2
-1
2
5
Lowest Payoff

Andersson, Hansen, Miltersen, Sørensen, CiE’2008

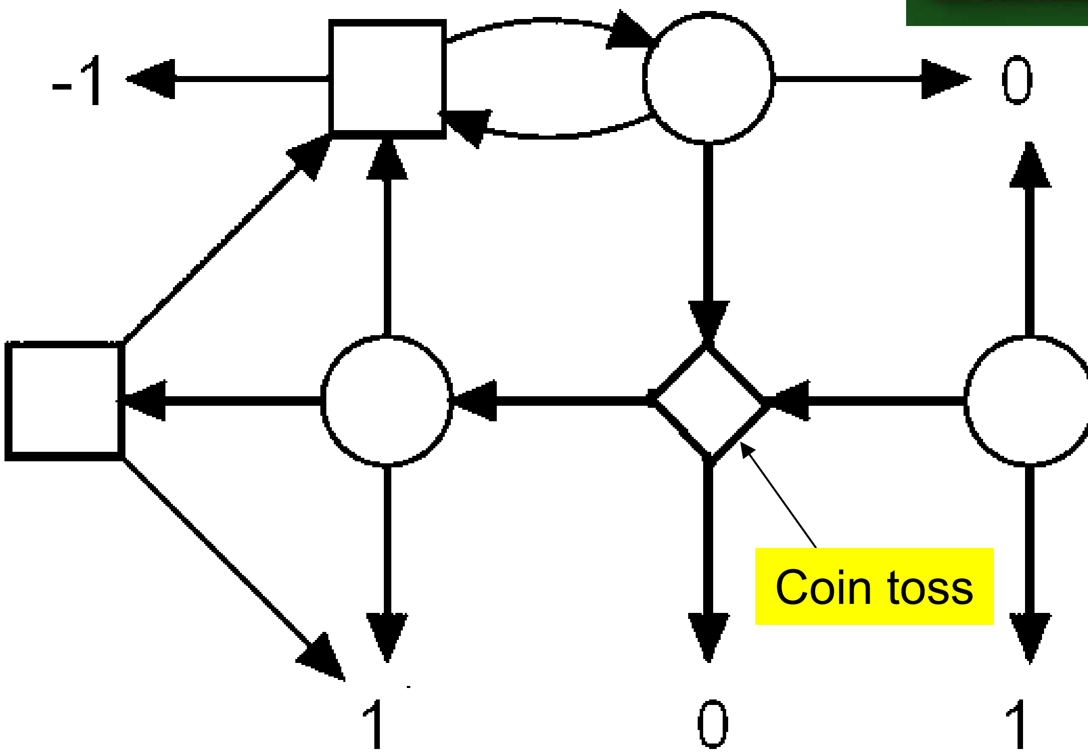
- Retrograde analysis solves deterministic graphical games, *but not in linear time*. Bottleneck: ***Payoffs must be sorted***.
- Alternative algorithm finds the value of a single position (“starting position”) in time $O(m \log^* m)$.

Open problem

Can a deterministic graphical game be solved in linear time by a comparison based algorithm?

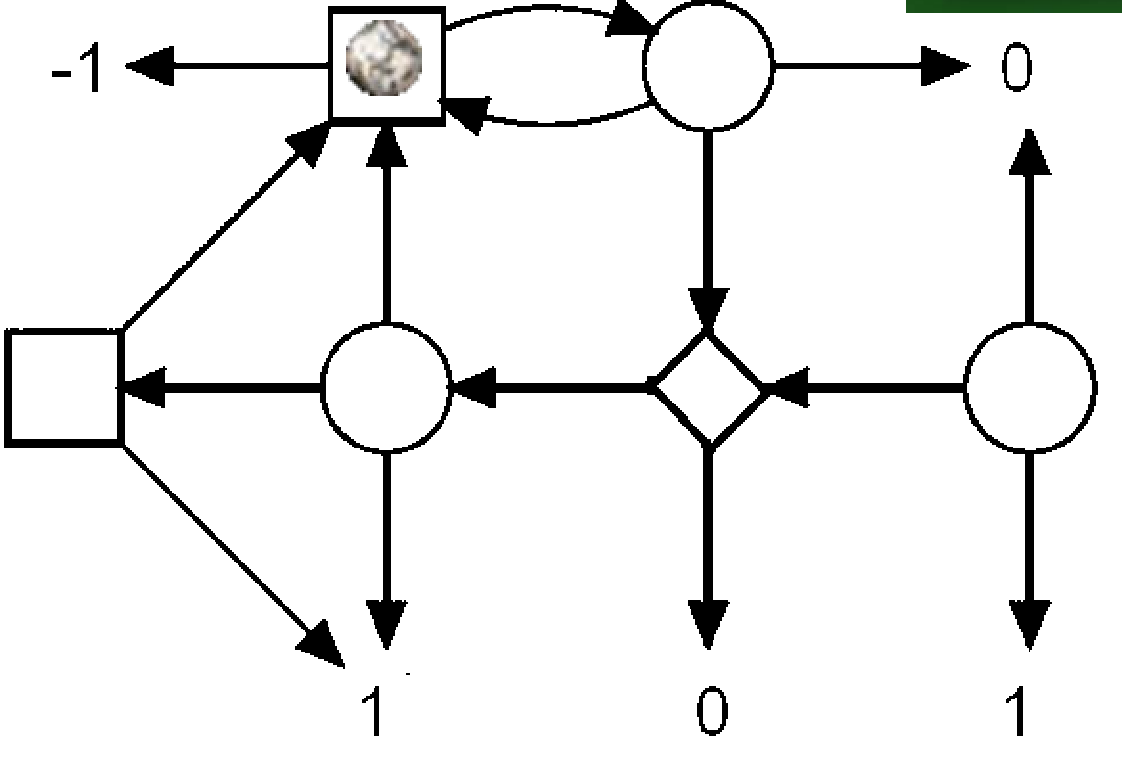
Simple stochastic games

- "Backgammon-like games"



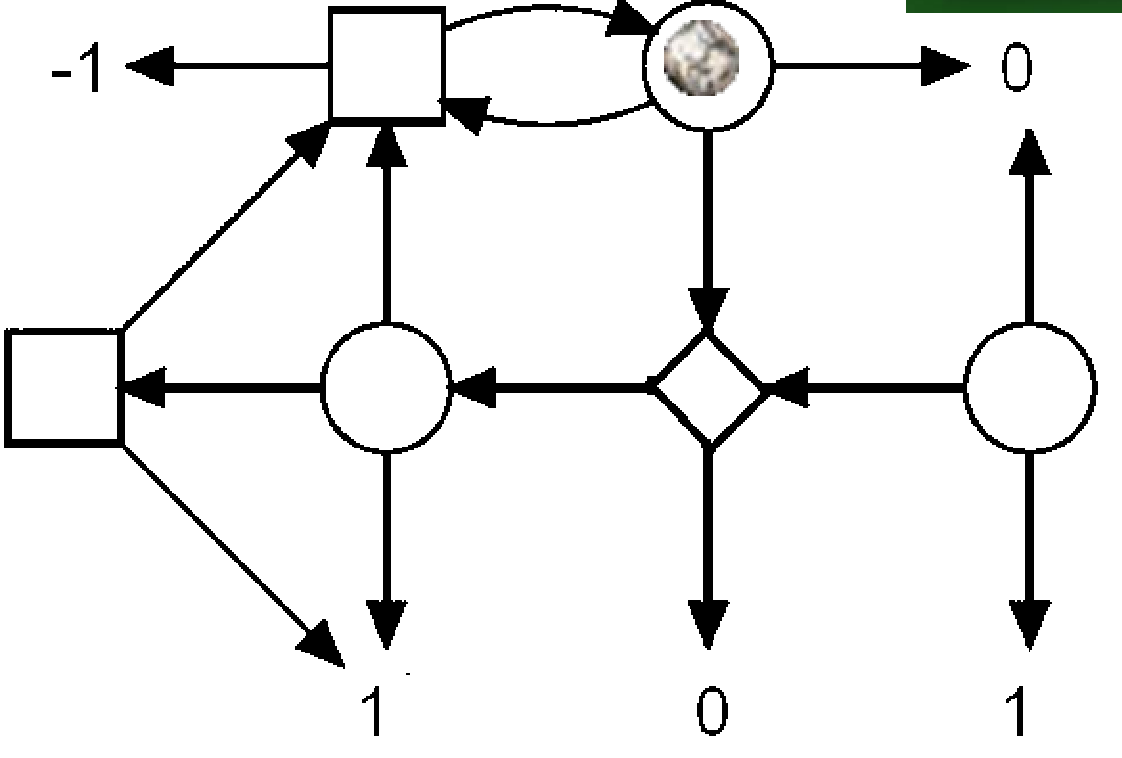
Simple stochastic games

- "Backgammon-like games"



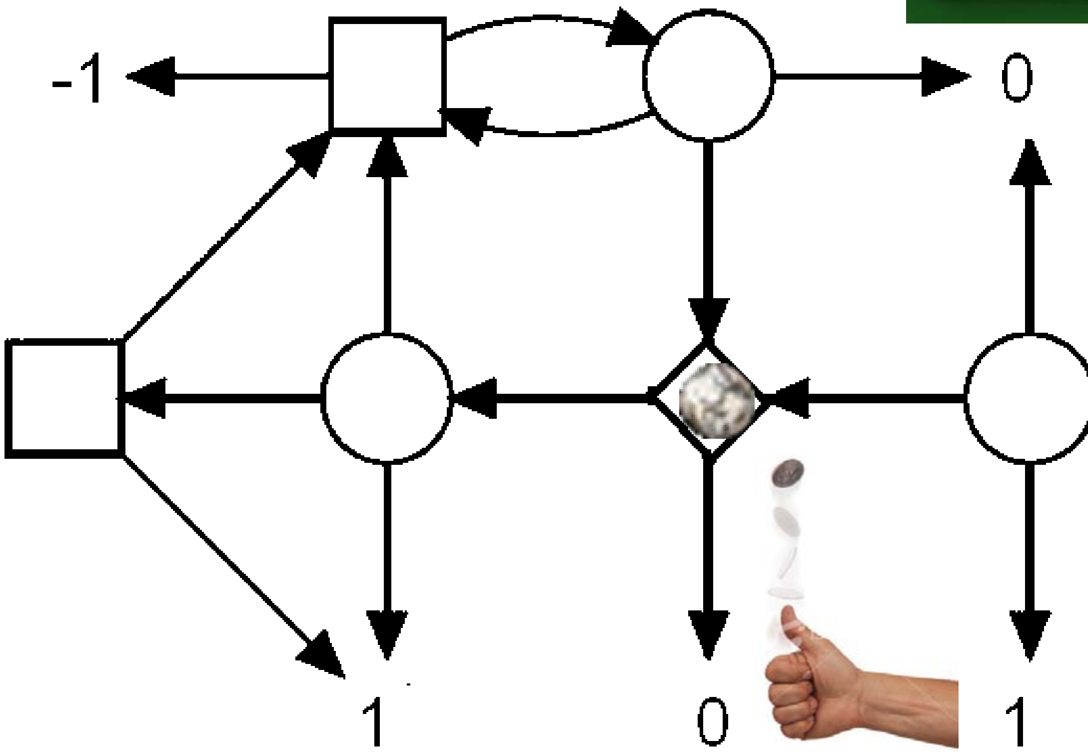
Simple stochastic games

- “Backgammon-like games”



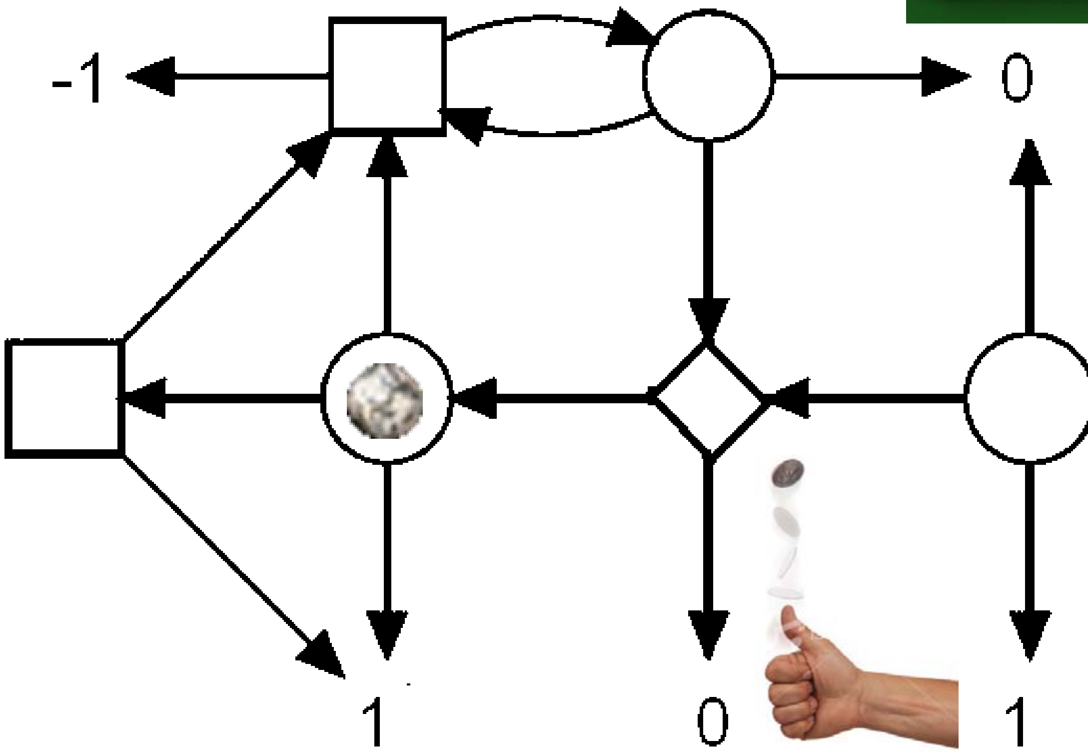
Simple stochastic games

- "Backgammon-like games"



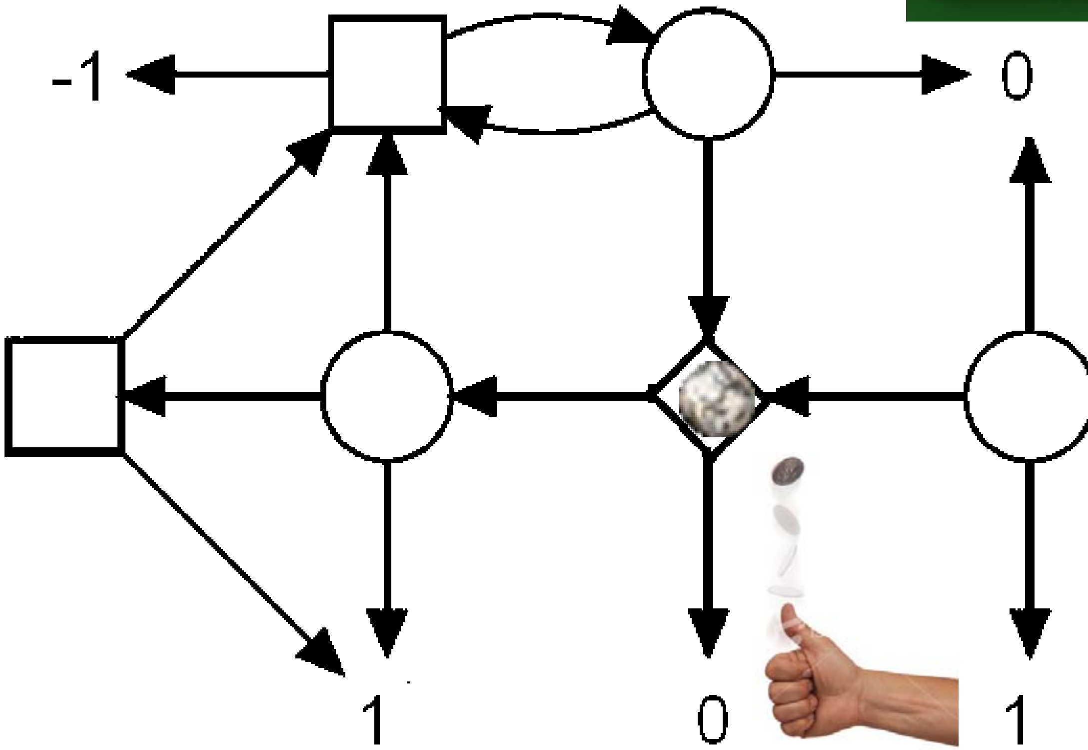
Simple stochastic games

- “Backgammon-like games”



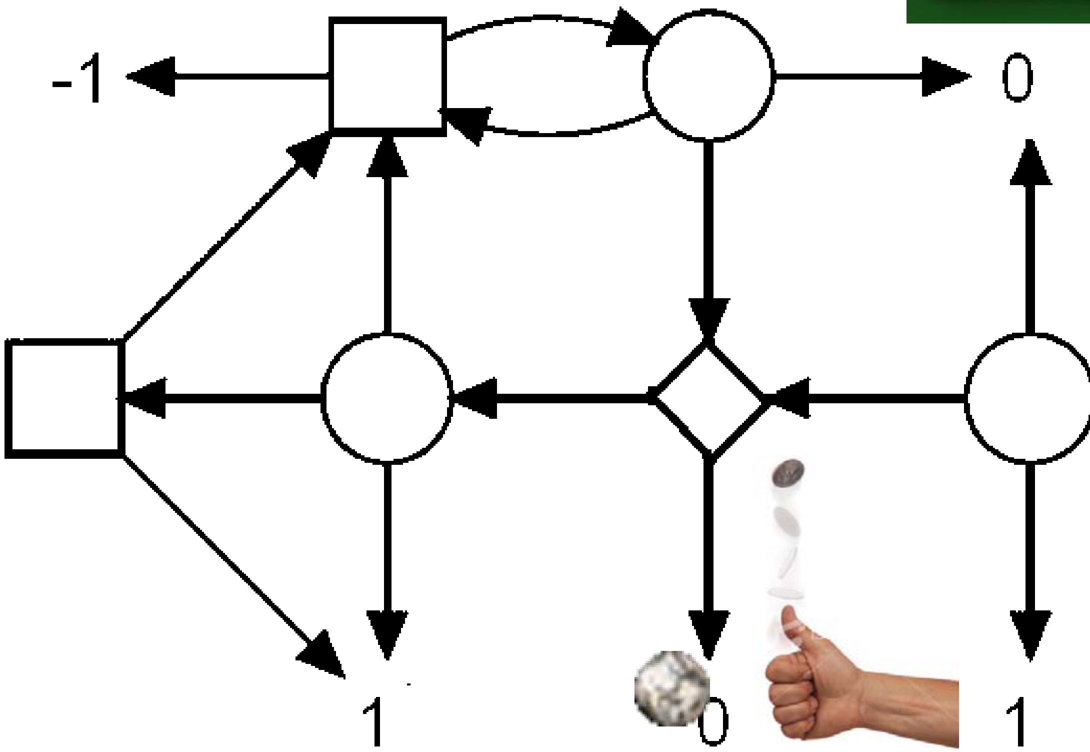
Simple stochastic games

- “Backgammon-like games”



Simple stochastic games

- “Backgammon-like games”



Values and optimal strategies

- Each position in a simple stochastic game has a ***value*** (Gillette,1957 & Liggett and Lippman,1969).
- Each player has a ***pure positional strategy*** guaranteeing the value in expectation - an ***optimal*** strategy (same refs).
- *It is not known how to compute in polynomial time the optimal strategies and the values given the SSG as input (Condon, 1988).*

Motivation: Games for verification

Verfification of reactive systems:

Will the hard disk recorder behave as Desired?

↓

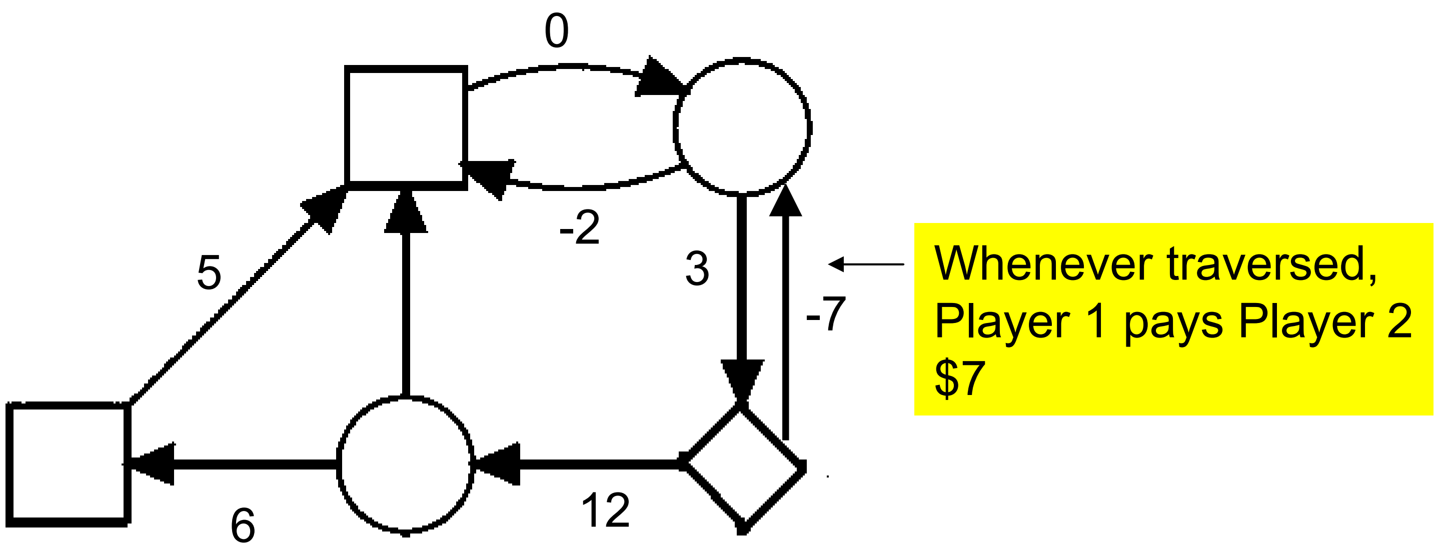
Model checking the μ-calculus

Polytime reduction

↓ E&J'88

Solving parity games → Solving deterministic mean payoff games → (Z&P'96) Solving simple Stochastic games

Mean-payoff and discounted payoff games



- Mean Payoff: asymptotic ***rate*** of rewards
- Discounted payoff: Total reward, when rewards are subject to ***inflation.***

Result

Will the hard disk
Recorder behave as
Desired?

↓

Model checking
the μ-calculus

↓

Solving parity games → Solving deterministic mean payoff games → Solving simple Stochastic games

Result

Will the hard disk Recorder behave as Desired?

Model checking the μ-calculus

Verification of stochastic reactive systems

C&J&H'04

Nir Halman'07: All are "LP-type problems"

Solving stochastic parity games

C&H'08

Solving stochastic mean-payoff games

Solving discounted payoff games

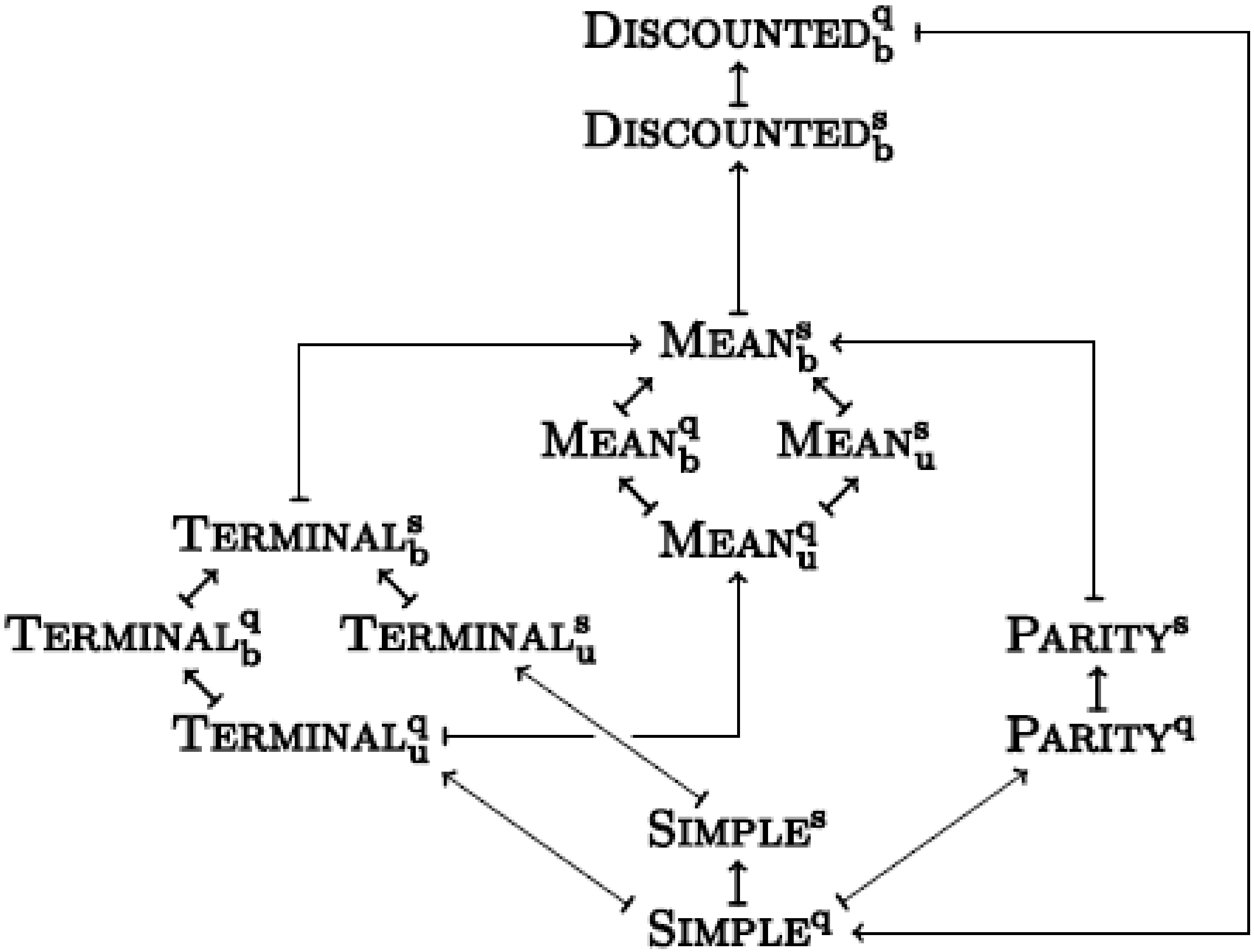
Andersson & M. '09

Solving parity games

Solving deterministic mean payoff games

Solving simple Stochastic games

The reductions



Stronger notion of equivalence: Strategy recovery

- For all the classes of games of this talk: If a birdy tells you optimal positional strategies, it is easy to compute values.

- Suppose a birdy tells you the values of all positions in a game. Can you efficiently find optimal strategies?

- ***Yes***, for all games on previous slide (Andersson and M, 2009), ***except***….

Open problems

- If a birdie tells you the values of all postions of a stochastic parity game, can you then efficiently find optimal pure positional strategies?

- If a birdie tells you the values of all positions of a stochastic mean payoff game, can you then efficiently find optimal pure positional strategies?

Hoffman-Karp algorithm for discounted payoff games

- X = a positional strategy for Player 1
- Repeat
  - Y = Optimal strategy for Player 2, *assuming that Player 1* ***must*** *play X.*
  - v = vector of expected payoffs under (X,Y)
  - Update X *locally* to go for best entries of v.
- Until stable

Does the Hoffman-Karp algorithm run in polynomial time???

A Super-Polynomial Lower Bound for the Parity Game Strategy Improvement Algorithm as We Know it

Oliver Friedmann
Institut für Informatik, LMU München

E-mail: `Oliver.Friedmann@googlemail.com`

Abstract

This paper presents a new lower bound for the discrete strategy improvement algorithm for solving parity games due to Vöge and Jurdziński. First, we informally show which structures are difficult to solve for the algorithm. Second, we outline a family of games of quadratic size on which the algorithm requires exponentially many strategy iterations, answering in the negative the long-standing question whether this algorithm runs in polynomial time. Additionally we note that the same family of games can be used to prove a similar result w.r.t. the strategy improvement variant by Schewe.

1. Introduction

Parity games are simple two-player games of perfect information played on directed graphs whose nodes are la-

All mentioned algorithms except for the two strategy improvement algorithms have been shown to have a super-polynomial worst-case runtime complexity at best or there is at least little doubt that their worst-case runtime complexity is super-polynomial or even exponential.

Solving parity games is one of the few problems that belongs to the complexity class NP ∩ coNP and that is not (yet) known to belong to P [2]. It has also been shown that solving parity games belongs to UP ∩ coUP [6]. The currently best known upper bound on the deterministic solution of parity games is $\mathcal{O}(|E| \cdot |V|^{\frac{1}{3}|\mathrm{ran}\,\Omega|})$ due to Schewe's big-step algorithm [11].

In this paper, we present a family of parity games comprising a linear number of nodes and a quadratic number of edges such that the strategy improvement algorithm by Vöge and Jurdziński requires an exponential number of iterations on them. Consequently, the algorithm requires at least super-polynomial time to solve parity games in the worst case. Due to page restrictions, we will only study

Seminal open problem (Condon 1988)

- ***Please*** solve simple stochastic games in worst case polynomial time!
- We now know that strategy improvement (“Hoffman-Karp”) runs in worst case exponential time.

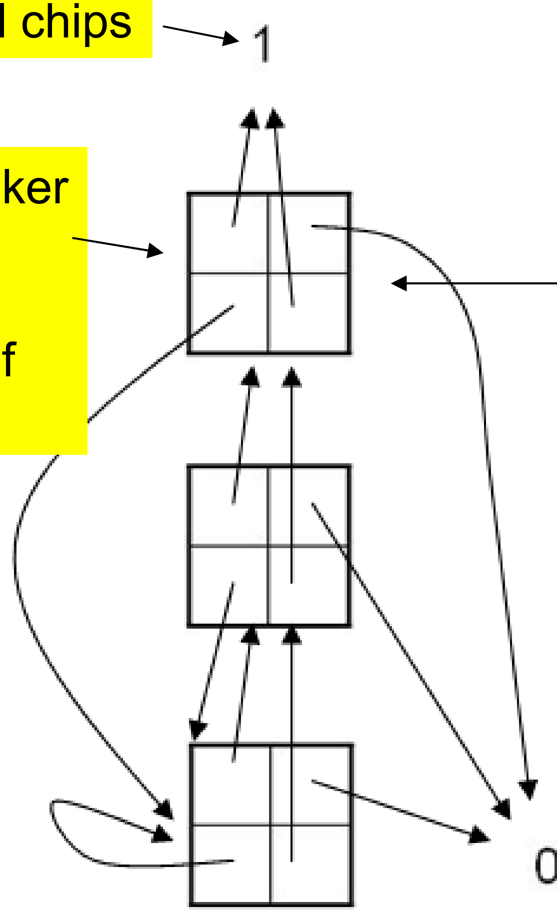
Concurrent reachability games

- “Poker tournament-like” games

Player 1 won all chips

A hand of poker played with a particular distribution of chips

In each position,
Player 1 chooses row
and Player 2 ***concurrently***
chooses column



http://www.daimi.au.dk/~bromille/Papers/aamaspoker.pdf

My most downloaded paper.
Download rate > 2*(combined rate of other papers)

A Near-Optimal Strategy for a Heads-Up No-Limit Texas Hold'em Poker Tournament

Peter Bro Miltersen
University of Aarhus
Åbogade 34, Århus, Denmark
bromille@daimi.au.dk

Troels Bjerre Sørensen
University of Aarhus
Åbogade 34, Århus, Denmark
trold@daimi.au.dk

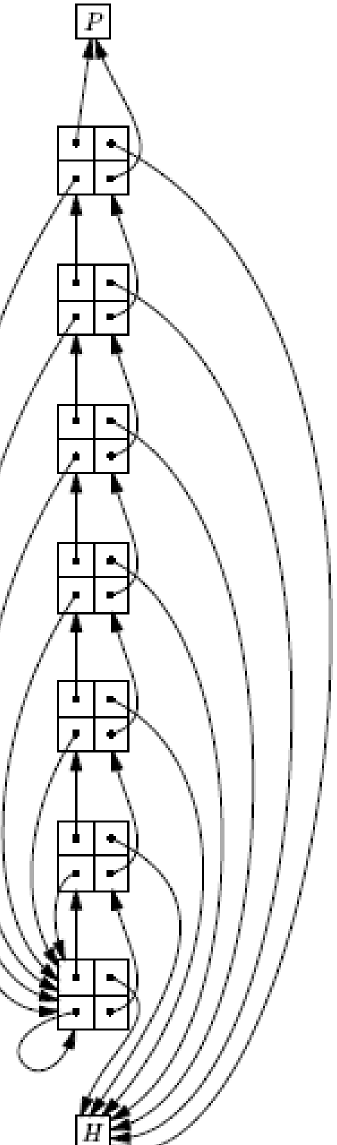
ABSTRACT

We analyze a heads-up no-limit Texas Hold'em poker tournament with a fixed small blind of 300 chips, a fixed big blind of 600 chips and a total amount of 8000 chips on the table (until recently, these parameters defined the heads-up endgame of sit-n-go tournaments on the popular PartyPoker.com online poker site). Due to the size of this game, a computation of an optimal (i.e. minimax) strategy for the game is completely infeasible. However, combining an algorithm due to Koller, Megiddo and von Stengel with concepts of Everett and suggestions of Sklansky, we compute an optimal *jam/fold* strategy, i.e. a strategy that would be optimal if any bet made by the player playing by the strategy (but

the computed strategy. These exact parameters were chosen as they until recently[1] defined the heads-up endgame of the \$5 through \$30 buy-in sit-n-go tournaments on the popular PartyPoker.com online poker site.

We briefly review the rules of such a tournament (and of no-limit Texas Hold'em in general). The tournament is played between two players, Player 1 and Player 2. When the tournament starts, Player 1 receives s_1 chips and Player 2 receives s_2 chips where $s_1 + s_2 = 8000$. We want to keep s_1 and s_2 as parameters as the heads-up tournament we consider may be the endgame of a tournament with more people, so the two players should be able to enter the game with different stack sizes. The total number of chips on the

Dante in Purgatory



Values and near-optimal strategies

- Each position in a concurrent reachability game has a ***value*** (Everett, 1957).
- For any ε>0, each player has a ***mixed positional strategy*** guaranteeing the value within ε (Everett, 1957).
- Player Min can guarantee the value ***exactly*** (de Alfaro & Majumdar, 2004).

Algorithmic problems

- Quantitatively solving CRG: Approximately compute the values.
- The values may be irrational, so they cannot be computed exactly…
- Strategically solving CRG: Given game and ε, compute ε-optimal strategies.

Algorithms strategically solving concurrent reachability games

Chatterjee, Majumdar, Jurdzinski, On Nash equilibria in stochastic games, *CSL'04.*

Chatterjee, de Alfaro, Henzinger. Strategy improvement for concurrent reachability games. *QEST'06.*

Chatterjee, de Alfaro, Henzinger. Termination criteria for solving concurrent safety and reachability games, *SODA'09.*

“Hardness” of solving CRGs

Theorem [Hansen, Koucky and M., LICS’09]:

– Any algorithm that manipulates ε-optimal strategies of concurrent reachability games must use exponential space.

…. solves open problem of Etessami and Yannakakis.

Dante in Purgatory

- Is there is a strategy for Dante so that he is guaranteed to win the game of Purgatory with probability at least 90%?
  - Yes.

A bit surprising – when Dante wins, he has guessed correctly which hand seven times in a row!

- How long can Lucifer confine Dante to Purgatory if Dante plays by such a strategy?
  - 10^{55} years.

Purgatory is a game of ***doubly exponential patience***.

- The ***patience*** of a mixed strategy is 1/p where p is the smallest non-zero probability used by the strategy (Everett, 1957).

- To win with probability 1-ε, Dante must choose “Right” at terrace i with probability greater than (approximately)

  $$1- \varepsilon^{2^{7-i}}$$

- On the other hand, choosing “Right” with probability 1 is no good!

- To win with probability 9/10, he must choose “Right” at terrace 1 with probability greater than $1-(1/10)^{64}$ = 0.999999999999999999999999999999999999999999999999999999999999999999. But then Lucifer can respond by always choosing “Left” at terrace 1.

“Hardness” of solving CRGs

Theorem [Hansen, Koucky and M.]:

– Any algorithm that manipulates ε-optimal strategies of concurrent reachability games must use exponential space.

- Proof: Storing 0.999999999999999999999999999999999999999999999 9999999999999999999999 takes up a lot of space!

Patience of Purgatory with n terraces and $\epsilon < ½$

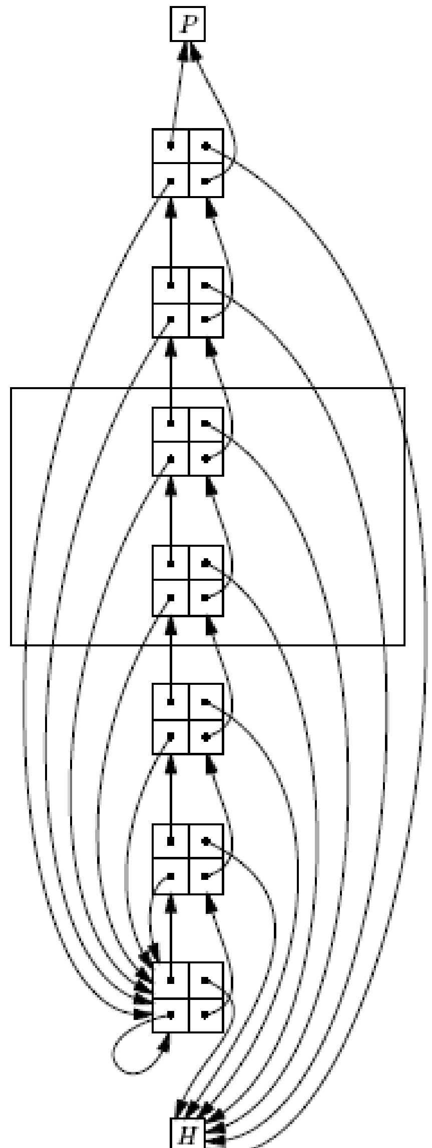
- Upper bound: $(1/\epsilon)^{2^{n-1}}$

- Lower bound: $((1-\epsilon)/\epsilon^2)^{2^{n-2}}$

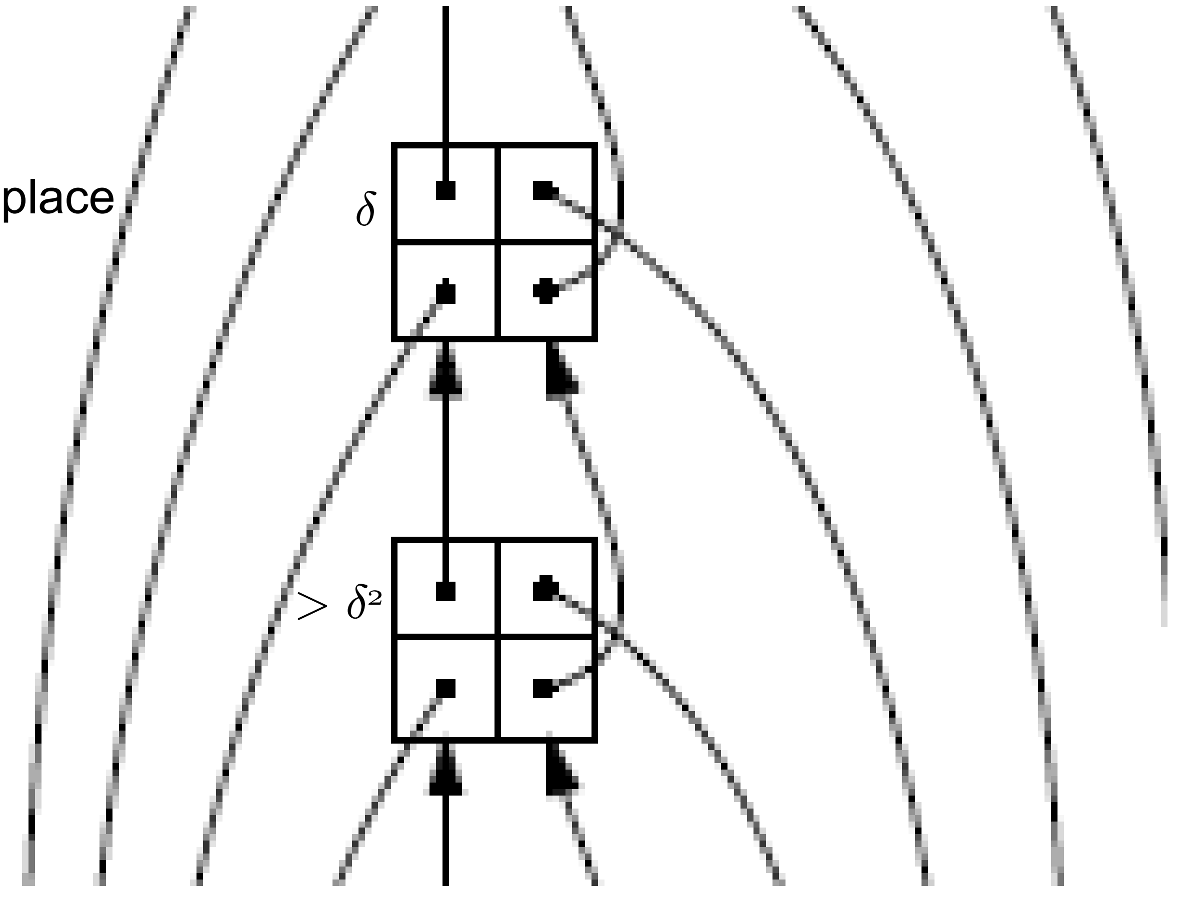
$$p_1 = \epsilon^{2^{n-1}}$$

$$p_i = \frac{\epsilon^{2^{n-i}} - \epsilon^{2^{n-i+1}}}{1 - \epsilon^{2^{n-i+1}}} \quad \text{for } i > 1$$

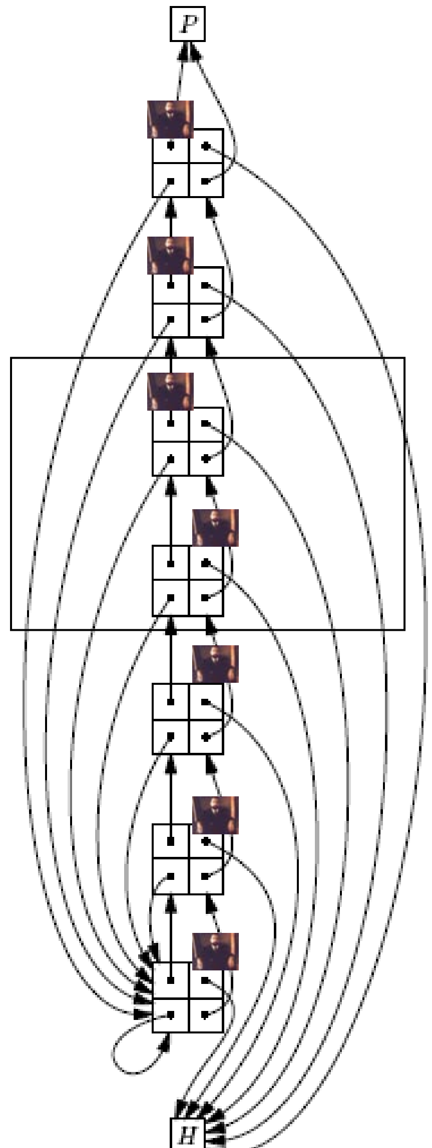
Proof of lower bound



WLOG first place
from above
where this
happens…



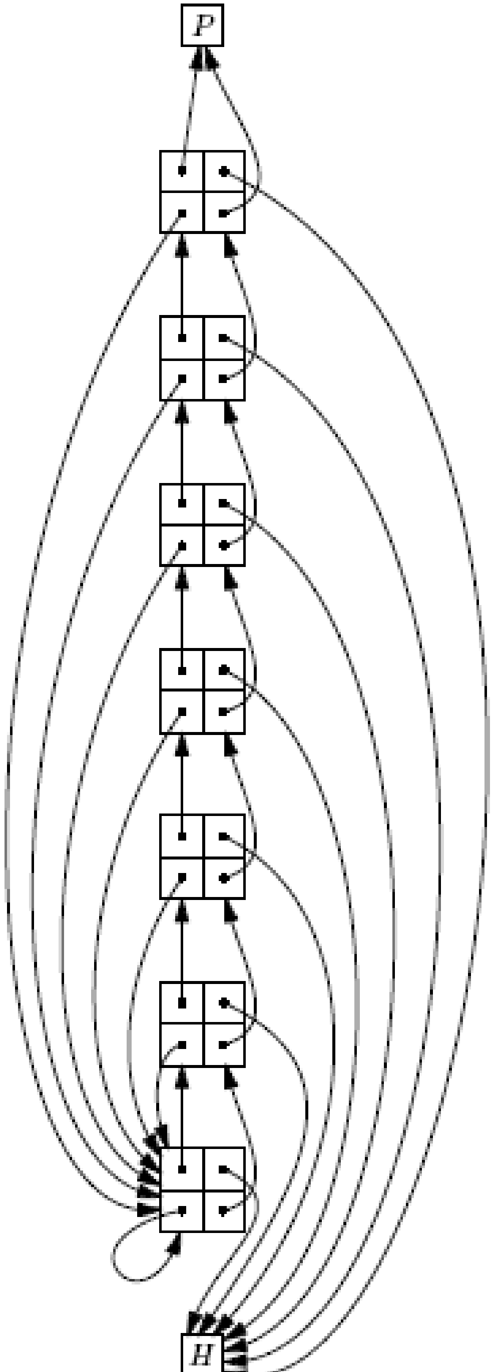
Proof of lower bound

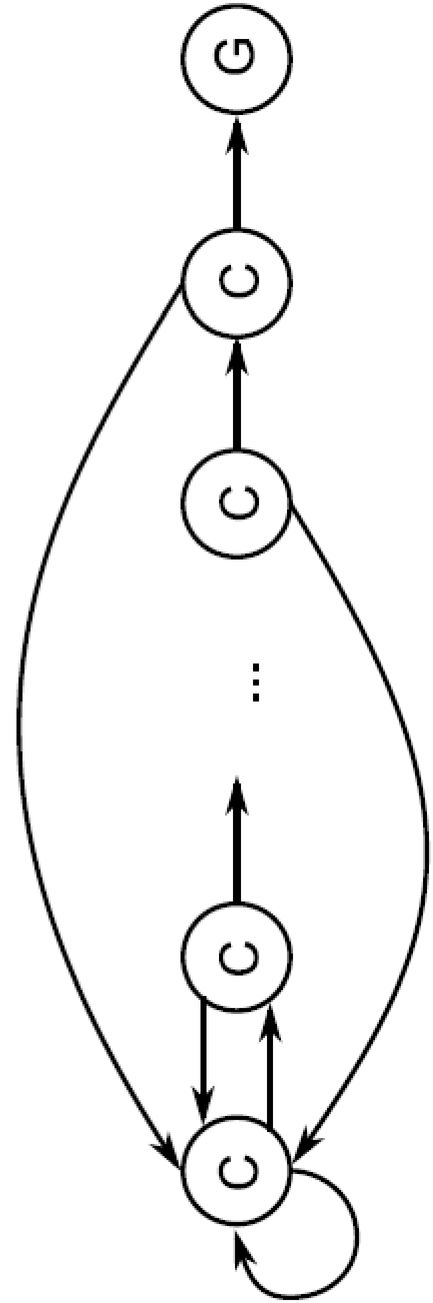


Open problems

- What is the exact patience of Purgatory? (upper bound tight for n=1,2)
- Is Purgatory *extremal* with respect to patience among *n*-state CRGs with binary choices?

Compare





Extremal with respect to, e.g., expected absorption time

Best upper bound I know

- **Theorem:** Patience $(1/\epsilon)^{2^{29\,m}}$ is sufficient to be ϵ-optimal in a concurrent reachability game with *m* actions.

- Shown by appealing to general theorems of semi-algebraic geometry (Basu et al.)

Time of play and value iteration

- To win Purgatory with probability $1-\epsilon$, almost all probability mass has to be assigned to strategies leading to plays of length at least $(1/\epsilon)^{2^{n-1}}$. Again, $(1/\epsilon)^{2^{29n}}$ is worst possible.

- To solve Purgatory quantitatively using value iteration, $2^{2^{n-1}}$ iterations are needed to get anywhere near the correct values. But $(1/\varepsilon)^{2^{29n}}$ iterations is enough to get ε-close for any n-position, binary-choice game.

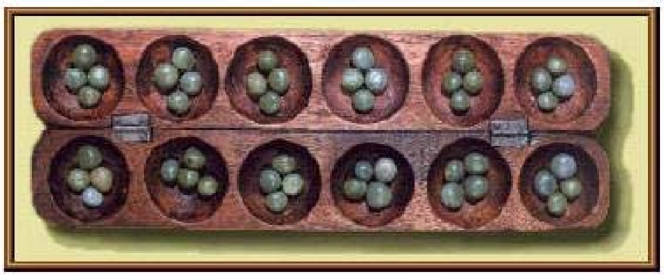
- If one shows Purgatory to be extremal, one gets a better bound on the complexity of value iteration (c becomes 1)!

Quantitatively solving CRGs

- Etessami and Yannakakis: CRGs can be quantitatively solved in polynomial space.
- Given rational α, we can even determine in polynomial space if the value is at least α.
- ….. So somehow polynomial space should be enough to "understand" CRGs fully.

Open Problem

- Is there a "natural" representation of probabilities so that
  - ε-optimal strategies of CRGs can be represented succinctly *and*
  - ε-optimal strategies of CRGs can be computed using polynomial space?
- De Alfaro, Henzinger, Kupferman '07: **Yes**, for the restricted case CRGs where the values of all postions are 0 or 1.



Thank you!