Bisimulations in Game Analysis

Roland Glück[1]

[1]Universität Augsburg

2.6.2011
Rotterdam

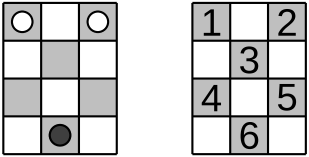
About

- about: simple two-player games
- solution in terms of relations/graphs
- simplification using bisimulations

Outline

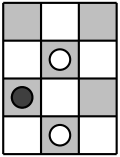
- introductory example
- algebraic theory
- bisimulations
- solution using bisimulations

Example Game

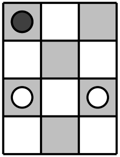


- white party begins
- no taking of pieces
- white moves one field diagonally downwards
- black moves one field diagonally in each direction
- white wins if no black move is possible
- black wins if no white move is possible or
- black reaches the back rank

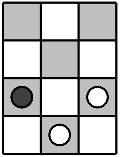
Example Positions



black to move, white wins



black wins

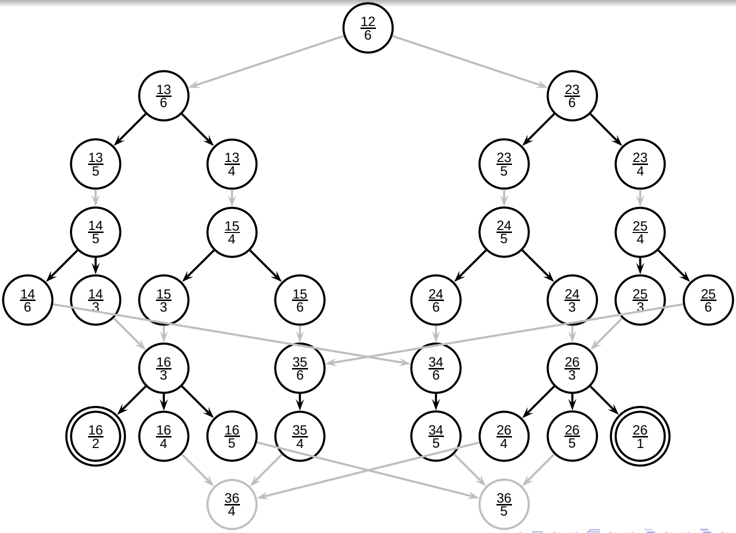


white to move, black wins

Solution Strategy

- which player can enforce the win?
- what is the winning strategy?
- solution via backward analysis
- using game graph

Game Graph

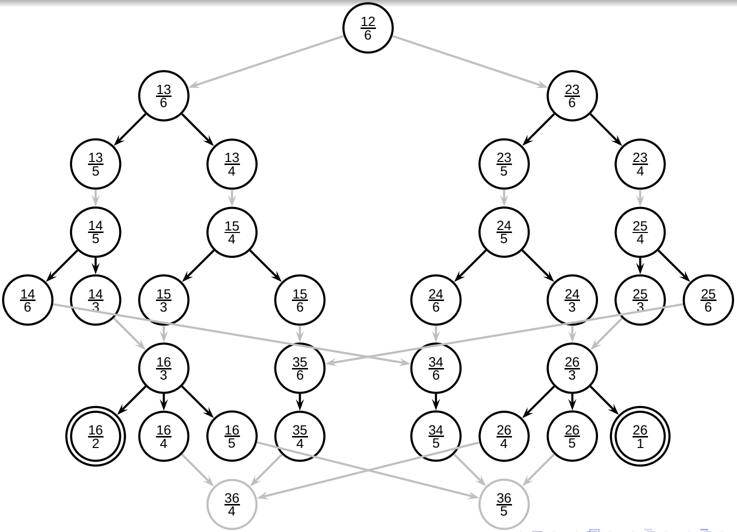

12
6
13
6
23
6
13
5
13
4
23
5
23
4
14
5
15
4
24
5
25
4
14
6
14
3
15
3
15
6
24
6
24
3
25
3
25
6
16
3
35
6
34
6
26
3
16
2
16
4
16
5
35
4
34
5
26
4
26
5
26
1
36
4
36
5

Initial Considerations

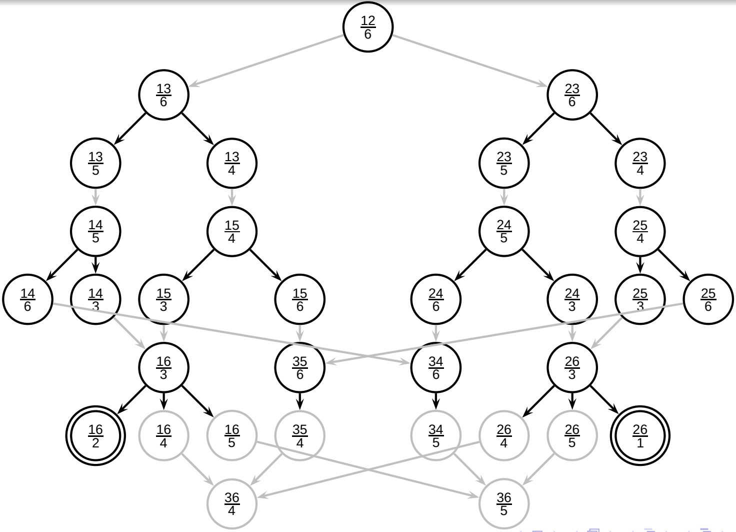
Observations:

- if from position P black has a move leading into a winning position for black then P is also a winning position for black (and hence a losing position for white)
- if from position P all black's moves are leading into a winning position for white then P is a losing position for black (hence a winning position for white)
- obviously the other way round (change the roles of black and white)

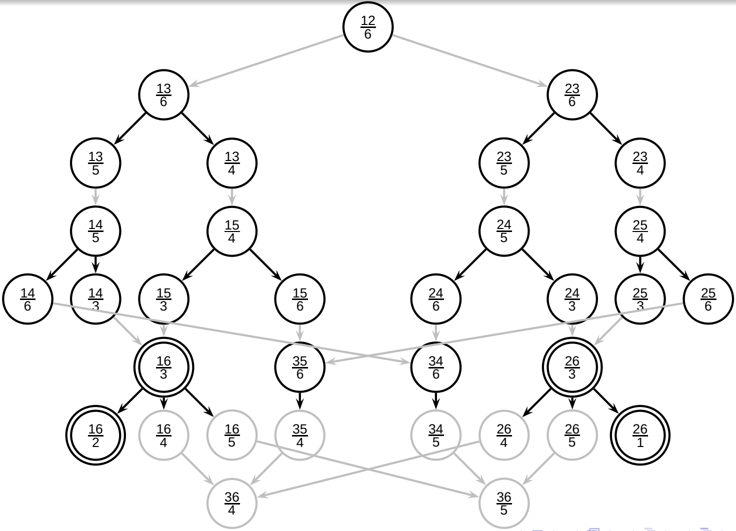
Solving the Game



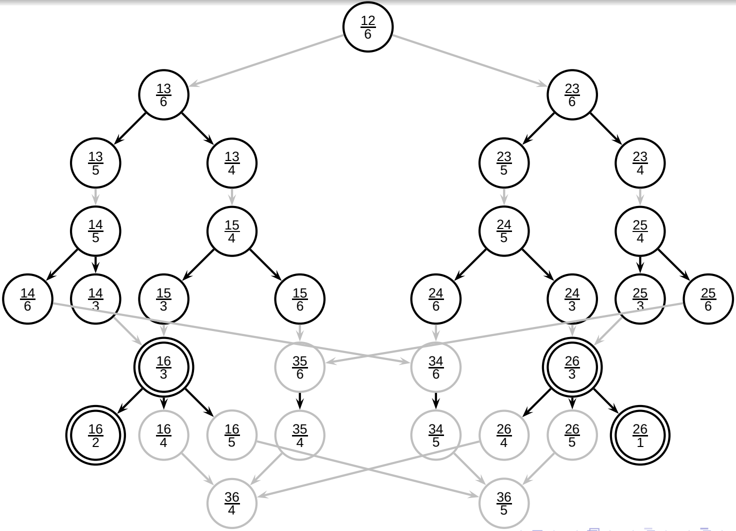
Solving the Game



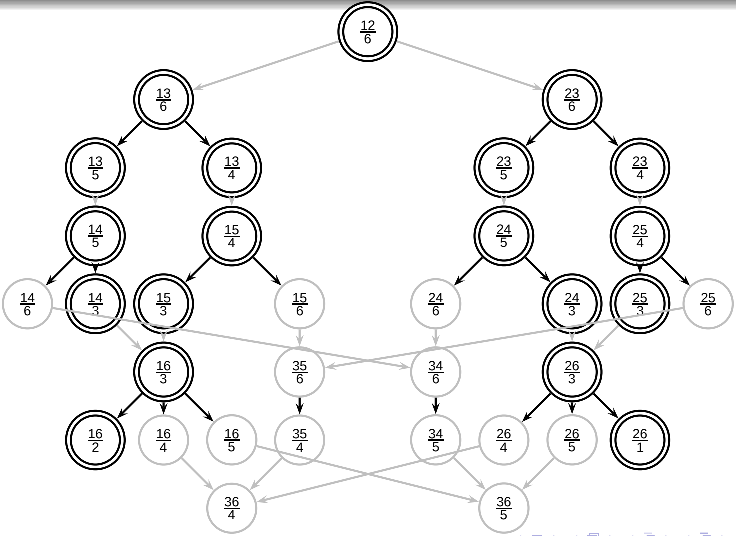
Solving the Game



Solving the Game



Solving the Game



Simplification

- again two-player games
- no move labels
- terminal positions are losing positions
- winning and losing positions are characterised similarly as above
- previous work by Backhouse and Michaelis, RelMiCS 7, 2003 (impartial two-player games)

Diamond and Box

- game corresponds to a finite graph $G = (V, E)$
- for $W \subseteq V$ define the modal operators *diamond*and *box* by
  - $|E\rangle W = \{v \in V \mid \exists w \in W : (v, w) \in E\}$
  - $|E]W = V - |E\rangle(V - W)$
- $|E\rangle W$ is the preimage of W under E
- $|E]W$ is the set of all nodes from which *every* E-transition leads into a node in W

Properties of Modal Operators

- diamond and box are isotone in their second argument wrt. the subset order $\subseteq$
- hence $|E\rangle|E] : \mathcal{P}(V) \to \mathcal{P}(V)$ is isotone wrt. to $\subseteq$
- every node without outgoing edges lies in $|E]W$ for arbitrary $W \subseteq V$
- $|E]\emptyset$ corresponds to nodes without outgoing edges in G, so-called *terminal* nodes or positions

Game Analysis using Modal Operators

- compute $L_0 = |E]\emptyset$
- compute $W_0 = |E\rangle L_0$
- compute $L_1 = |E]W_0$
- compute $W_1 = |E\rangle L_1$
- ....
- compute $W_i = |E\rangle|E]W_{i-1}$ till $W_i = W_{i-1}$

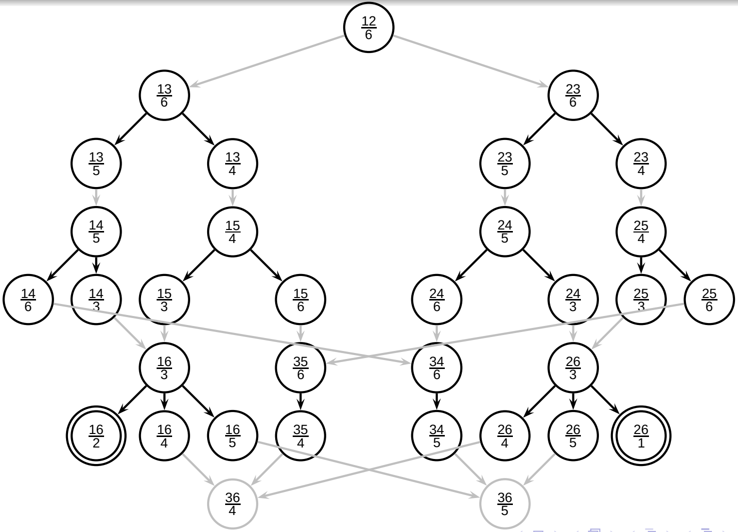
Fixpoint Characterisation

- fixpoint W (winning positions) reached due to isotony of $|E\rangle|E]$ and finiteness of V
- analogously for L (losing positions)
- winning and losing positions can be characterised as smallest fixpoints
- remaining positions are stalemate positions
- no stalemate positions in acyclic games

Variations

- slight adaptations for example game
- $E = E_w \dot\cup E_b$
- $W_{w,i} = L_{b,i}$ and $W_{b,i} = L_{w,i}$
- $W_{w,i} = |E_w\rangle W_{w,i-1} \cup (|E_b]W_{w,i-1} \cap |E_b\rangle W_{w,i-1})$
- symmetrically for $W_{b,i}$
- iteration starts at predefined winning and losing positions

Take a Look back



Observations on the Game Graph

- game graph is symmetric
- pairs of equivalent positions
- w.l.o.g.
- how to exploit this?
- use of bisimulations

Definition

$B \subseteq V_1 \times V_2$ is a *bisimulation* between two graphs (V_1, E_1) and (V_2, E_2) iff

- $Dom(B) = X_1$ and $Cod(B) = X_2$
- $v_1 B v_2 \wedge v_1 E_1 w_1 \Rightarrow \exists w_2 : w_1 B w_2 \wedge v_2 E_2 w_2$
- $v_2 B^{\smile} v_1 \wedge v_2 E_2 w_2 \Rightarrow \exists w_1 : w_2 B^{\smile} w_1 \wedge v_1 E_1 w_1$

relational definition:

- $B^{\smile}; E_1 \subseteq E_2; B^{\smile} \wedge R; E_2 \subseteq E_1; B$

Coarsest Bisimulation

- bisimulations between G and itself are closed under
  - union,
  - composition, and
  - taking the converse
- identity is a bisimulation between G and itself
- existence of a *coarsest bisimulation equivalence on G*

Quotient Graph

for a bisimulation equivalence B and a graph $G = (V, E)$ the *quotient* $G/B = (V/B, E/B)$ is defined by

- V/B is the set of equivalence classes of B
- $(v/B, w/B) \in E/B \Leftrightarrow (v, w) \in E$

G/B has in general a smaller node set then G

Hooray!

!!! good news !!!

Theorem

Under certain conditions the sets $|E\rangle W$ and $|E]W$ can be written as the union of the sets from $|E/B\rangle W/B$ and $|E/B]W/B$.

Hooray!

!!! good news !!!

Theorem

Under certain conditions the sets $|E\rangle W$ and $|E]W$ can be written as the union of the sets from $|E/B\rangle W/B$ and $|E/B]W/B$.

!!! very good news !!!

The sets occurring in the computation of W and L fulfil these conditions!

Final Algorithm

Algorithm:

- compute the coarsest bisimulation B on $G = (V, E)$
- compute the winning positions W_B and the losing positions L_B in G/B using the known algorithm
- the winning positions W in G is the set of nodes $\bigcup_{w_B \in W_B} w_B$
- analogously for the losing and stalemate positions

Efficiency

- makes sense if computing G/B can be done faster than running the algorithm immediately on G
- computation of G/B in $\mathcal{O}(|E| \cdot log(|V|))$ possible (Paige and Tarjan)
- original algorithm can take up to $\Theta(|V|^{1.5})$ on a certain class of games
- for this class of games the detour over bisimulations has worst case runtime of $\Theta(|V| \cdot log(|V|))$

future work:

- application to stochastic games
- application to multi-player games
- formalisation in semiring framework

Bisimulations in Game Analysis

Roland Glück[1]

[1]Universität Augsburg

2.6.2011
Rotterdam

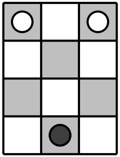
About

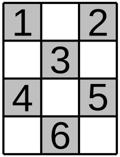
- about: simple two-player games
- solution in terms of relations/graphs
- simplification using bisimulations

Outline

- introductory example
- algebraic theory
- bisimulations
- solution using bisimulations

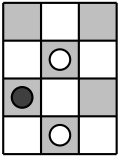
Example Game



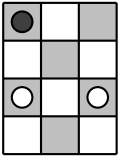


- white party begins
- no taking of pieces
- white moves one field diagonally downwards
- black moves one field diagonally in each direction
- white wins if no black move is possible
- black wins if no white move is possible or
- black reaches the back rank

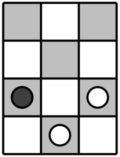
Example Positions



black to move,
white wins



black wins

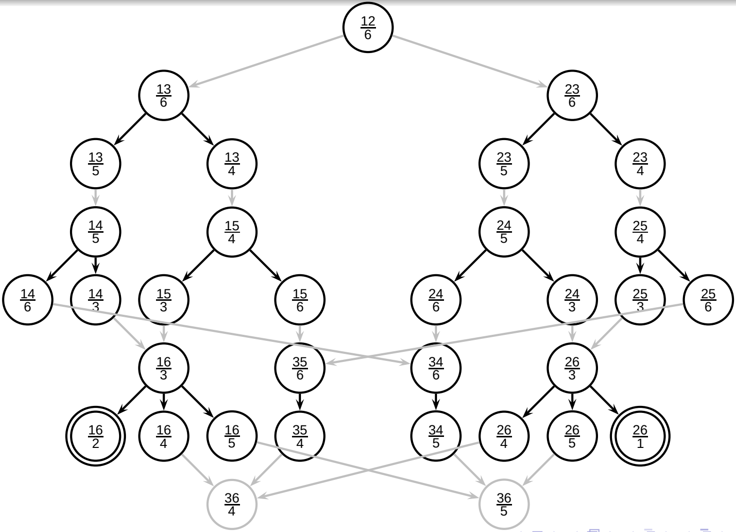


white to move,
black wins

Solution Strategy

- which player can enforce the win?
- what is the winning strategy?
- solution via backward analysis
- using game graph

Game Graph


12 6
13 6
23 6
13 5
13 4
23 5
23 4
14 5
15 4
24 5
25 4
14 6
14 3
15 3
15 6
24 6
24 3
25 3
25 6
16 3
35 6
34 6
26 3
16 2
16 4
16 5
35 4
34 5
26 4
26 5
26 1
36 4
36 5