Rooks and ARC-KAYLES

Antoine Dailly[1], Valentin Gledel[1], Marc Heinrich[1]

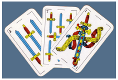
[1]Université Lyon 1, LIRIS, Lyon

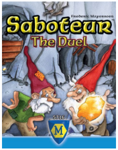
This work is part of the ANR GAG (Graphs and Games).
Thanks to Nicolas Bousquet for his help.

Seminario Preguntón, December 13, 2017

Combinatorial Games

Definition

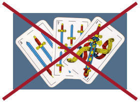


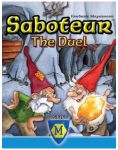


Combinatorial Games

Definition

1. Two-player games

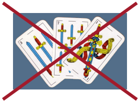


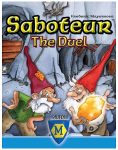


Combinatorial Games

Definition

1. Two-player games
2. No chance

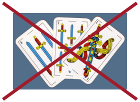



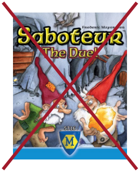



Combinatorial Games

Definition

1. Two-player games
2. No chance
3. Perfect information

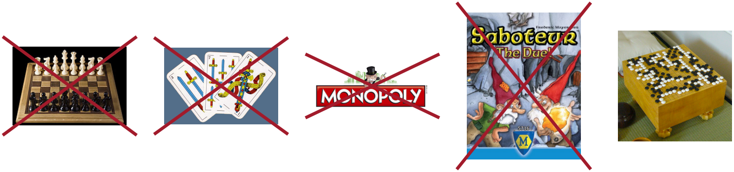



Combinatorial Games

Definition

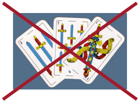
1. Two-player games
2. No chance
3. Perfect information
4. Finite games, no draw

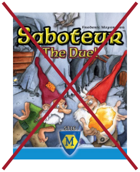


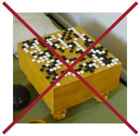
Combinatorial Games

Definition

1. Two-player games
2. No chance
3. Perfect information
4. Finite games, no draw
5. The last move alone determines the winner



Combinatorial Games

Relaxations

1. Two-player games $\rightarrow$ Multiplayer Theory
2. No chance $\rightarrow$ Economical Games
3. Perfect information $\rightarrow$ Economical Games
4. Finite games, no draw $\rightarrow$ Loopy Games
5. The last move alone determines the winner $\rightarrow$ Scoring Games, several graph games (coloring game, domination game, ...)

Combinatorial Games

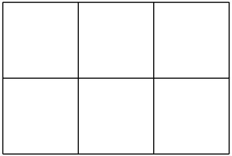
Relaxations

1. Two-player games $\rightarrow$ Multiplayer Theory
2. No chance $\rightarrow$ Economical Games
3. Perfect information $\rightarrow$ Economical Games
4. Finite games, no draw $\rightarrow$ Loopy Games
5. The last move alone determines the winner $\rightarrow$ Scoring Games, several graph games (coloring game, domination game, ...)

$\rightarrow$ We will talk about pure Combinatorial Games.
Note: Both players play perfectly!

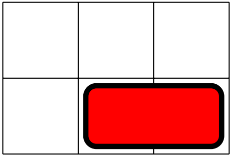
Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



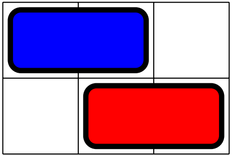
Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



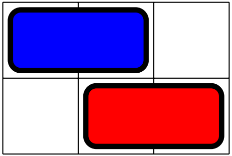
Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



Let's play!

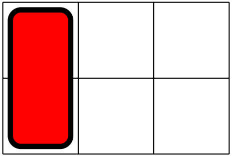
CRAM: The players place dominos on a grid. The player who plays the last domino wins.



$\Rightarrow$ Second player wins.

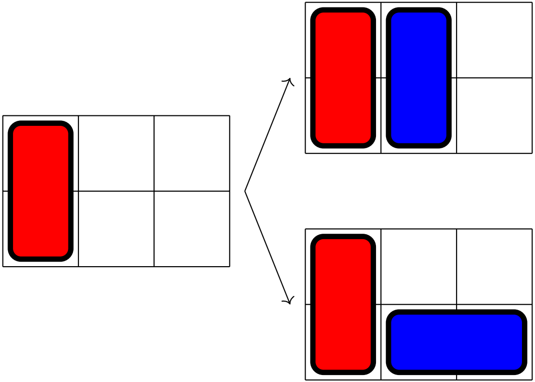
Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



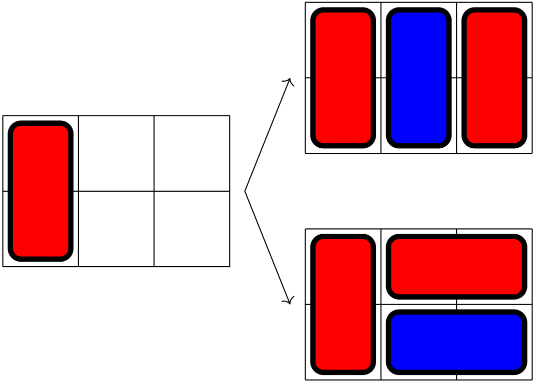
Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



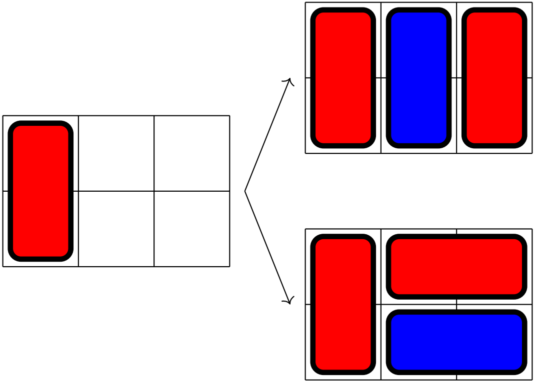
Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



Let's play!

CRAM: The players place dominos on a grid. The player who plays the last domino wins.



$\Rightarrow$ First player wins.

Definitions

Definitions

Impartial and partizan

A game is impartial $\Leftrightarrow$ Both players have the same moves.
Otherwise, it is partizan.

Definitions

Impartial and partizan

A game is impartial $\Leftrightarrow$ Both players have the same moves.
Otherwise, it is partizan.

$\rightarrow$ We will study impartial games

Definitions

Impartial and partizan

A game is impartial $\Leftrightarrow$ Both players have the same moves.
Otherwise, it is partizan.

$\rightarrow$ We will study impartial games

Outcome

A game is $\mathcal{N}$ $\Leftrightarrow$ The first player has a winning strategy.
Otherwise, it is $\mathcal{P}$.

Definitions

Impartial and partizan

A game is impartial $\Leftrightarrow$ Both players have the same moves.
Otherwise, it is partizan.

$\rightarrow$ We will study impartial games

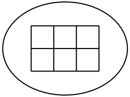
Outcome

A game is $\mathcal{N}$ $\Leftrightarrow$ The first player has a winning strategy.
Otherwise, it is $\mathcal{P}$.

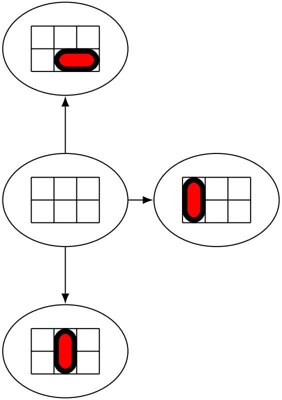
Problematics of Combinatorial Games

1. Is a given game $\mathcal{N}$ or $\mathcal{P}$?
2. What is the winning strategy?

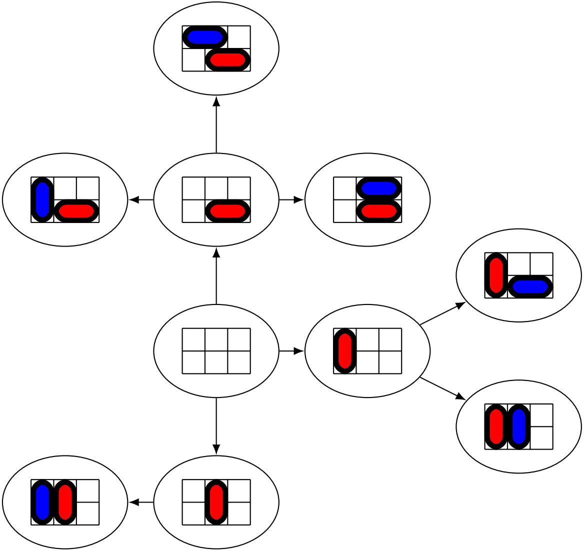
An inefficient algorithm: the game graph



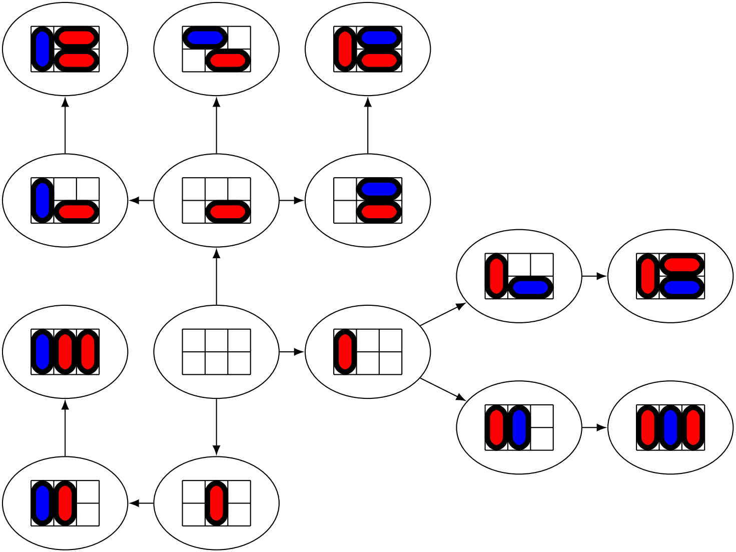
An inefficient algorithm: the game graph



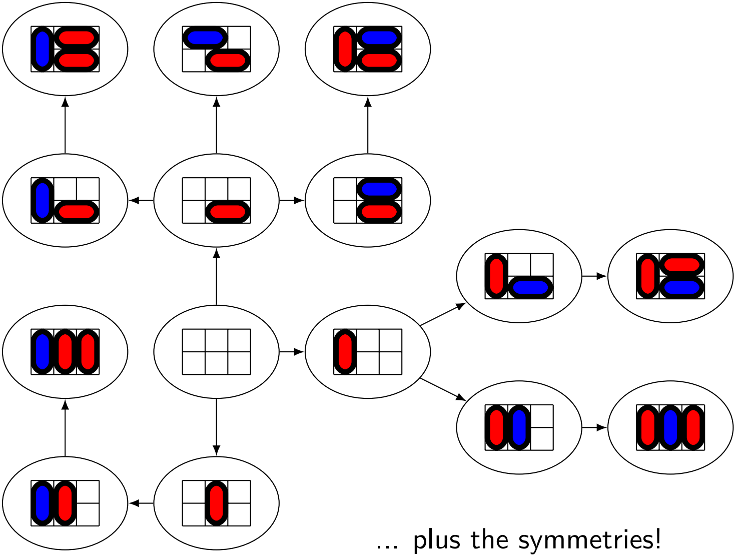
An inefficient algorithm: the game graph



An inefficient algorithm: the game graph

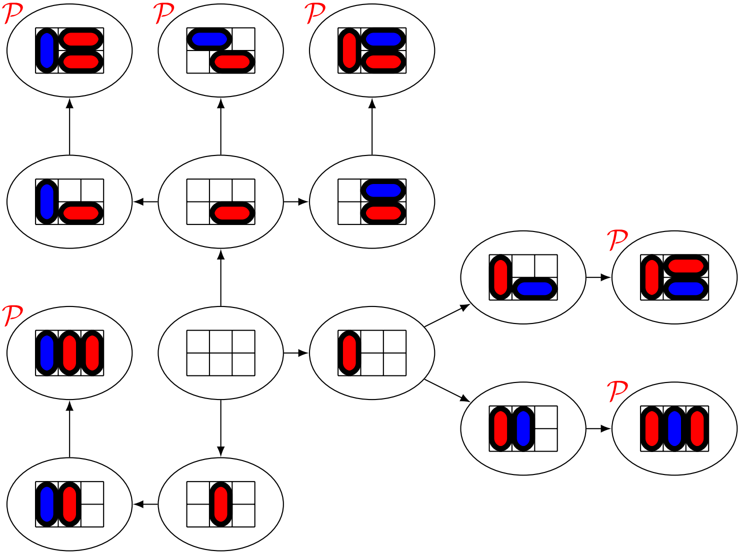


An inefficient algorithm: the game graph

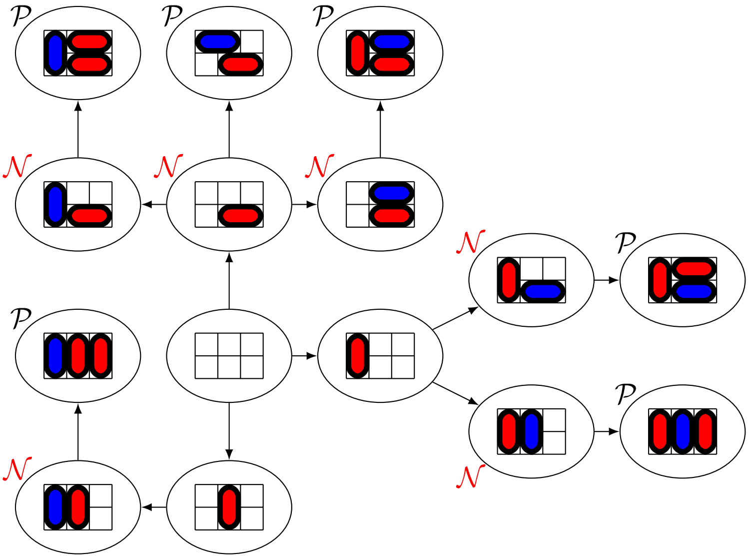


… plus the symmetries!

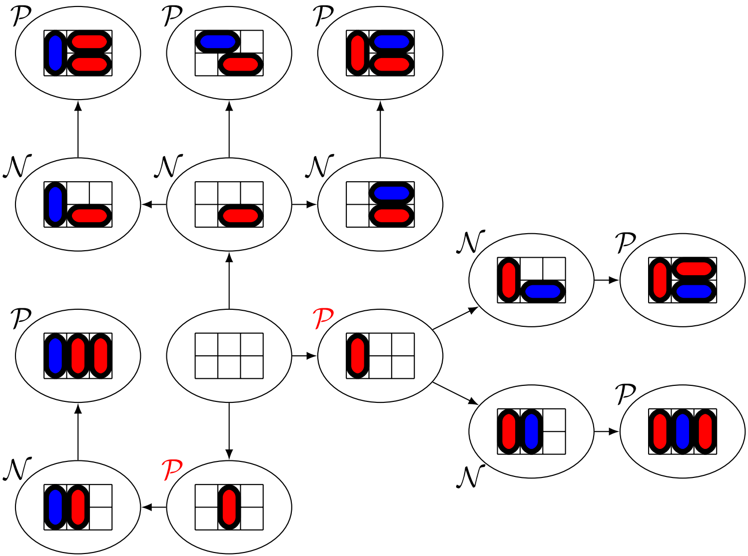
An inefficient algorithm: the game graph



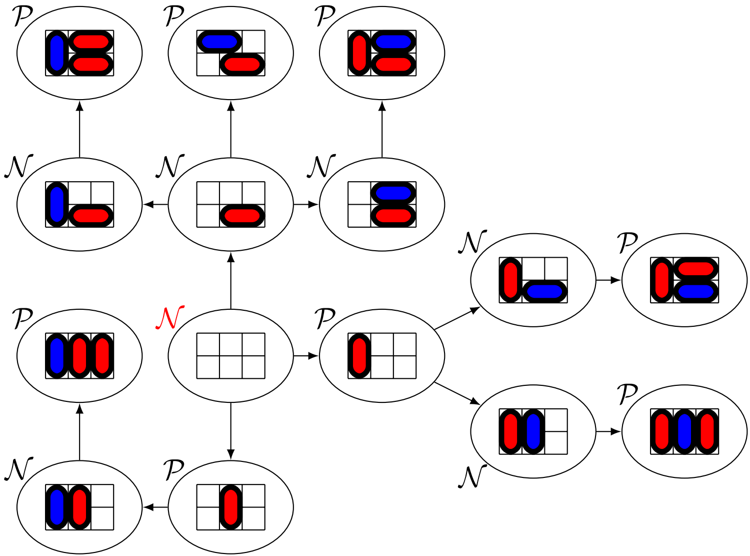
An inefficient algorithm: the game graph



An inefficient algorithm: the game graph



An inefficient algorithm: the game graph



An inefficient algorithm: the game graph

- Complete and finite

An inefficient algorithm: the game graph

- Complete and finite
- ... but exponential-time!

An inefficient algorithm: the game graph

- Complete and finite
- ... but exponential-time!

$\Rightarrow$ A more efficient method to study games

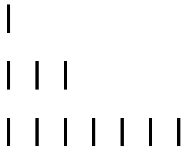
Nim

Nim

- Studied in 1901 by Charles Bouton

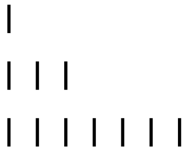
Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters



Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap



Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap

|

| | |

| |

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

Nim

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

|

|

$\Rightarrow$ Here, the first player won

NIM

- Studied in 1901 by Charles Bouton
- Played on heaps of counters
- Players remove counters from one heap
- Whoever takes the last counter(s) wins

$\Rightarrow$ Here, the first player won

$\Rightarrow$ Is there a strategy?

Solving Nim

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

|	1
| | |	3
| | | | | | |	7

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

I	$1 = 001$
I I I	$3 = 011$
I I I I I I I	$7 = 111$

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

|	$1 = 001$
	$\oplus$
| | |	$3 = 011$
	$\oplus$
| | | | | | |	$7 = 111$
	101

Solving Nim

Theorem

Let $(a_1, \ldots, a_n)$ be a Nim-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

|	$1 = 001$
	$\oplus$
| | |	$3 = 011$
	$\oplus$
| | | | | | |	$7 = 111$
	101

Proof (by induction)

- If $a_1 \oplus \ldots \oplus a_n \neq 0$, then move to such a position.

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

$$\begin{array}{rcl} 1 & = & 001 \\ & & \oplus \\ 3 & = & 011 \\ & & \oplus \\ 2 & = & 010 \\ \hline & & 000 \end{array}$$

Proof (by induction)

- If $a_1 \oplus \ldots \oplus a_n \neq 0$, then move to such a position.

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

$$\begin{array}{lcr} | & \qquad & 1 = 001 \\ & & \oplus \\ |\ |\ | & & 3 = 011 \\ & & \oplus \\ |\ | & & 2 = 010 \\ \hline & & 000 \end{array}$$

Proof (by induction)

- If $a_1 \oplus \ldots \oplus a_n \neq 0$, then move to such a position.
- Otherwise, then no move leaves such a position.

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

$$\begin{array}{rcl} 1 & = & 001 \\ & & \oplus \\ 0 & = & 000 \\ & & \oplus \\ 2 & = & 010 \\ \hline & & 011 \end{array}$$

Proof (by induction)

- If $a_1 \oplus \ldots \oplus a_n \neq 0$, then move to such a position.
- Otherwise, then no move leaves such a position.

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

|	$1 = 001$
	$\oplus$
| |	$2 = 010$
	011

Proof (by induction)

- If $a_1 \oplus \ldots \oplus a_n \neq 0$, then move to such a position.
- Otherwise, then no move leaves such a position.
- When two heaps are left, the result is trivial.

Solving NIM

Theorem

Let $(a_1, \ldots, a_n)$ be a NIM-position.
It is $\mathcal{P} \Leftrightarrow a_1 \oplus \ldots \oplus a_n = 0$.

| $1 = 001$

$$\oplus$$

| | $1 = \underline{001}$

$$000$$

Proof (by induction)

- If $a_1 \oplus \ldots \oplus a_n \neq 0$, then move to such a position.
- Otherwise, then no move leaves such a position.
- When two heaps are left, the result is trivial.

Back to CRAM: summing games

Back to Cram: summing games

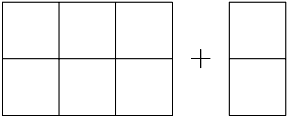
Sum of games

On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.

Back to CRAM: summing games

Sum of games

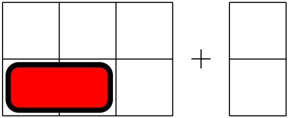
On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.



Back to CRAM: summing games

Sum of games

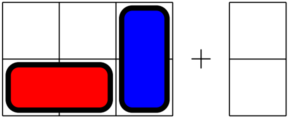
On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.



Back to CRAM: summing games

Sum of games

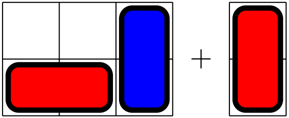
On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.



Back to CRAM: summing games

Sum of games

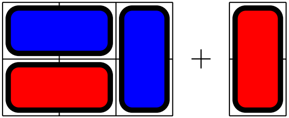
On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.



Back to CRAM: summing games

Sum of games

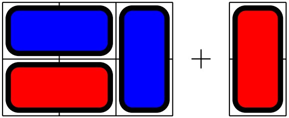
On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.



Back to CRAM: summing games

Sum of games

On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.

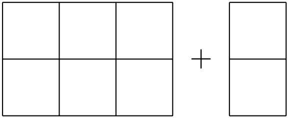


Why summing games?

Back to CRAM: summing games

Sum of games

On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.

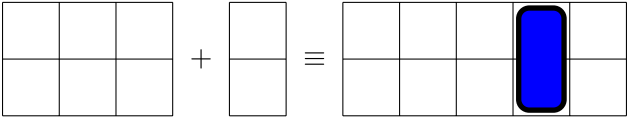


Why summing games?

Back to CRAM: summing games

Sum of games

On $G + H$, the players play either on G or on H. When G (resp. H) is over, they play on H (resp. G).
The player who plays the last move wins.



Why summing games?

Summing a $\mathcal{P}$ game

Theorem

Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

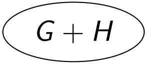
Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

Proof by induction

- If H is $\mathcal{P}$:



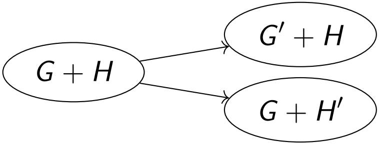
Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

Proof by induction

- If H is $\mathcal{P}$:



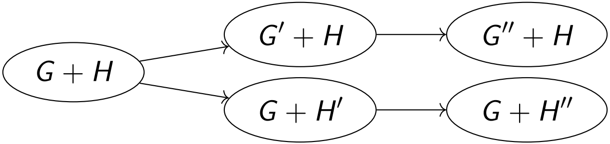
Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

Proof by induction

- If H is $\mathcal{P}$:



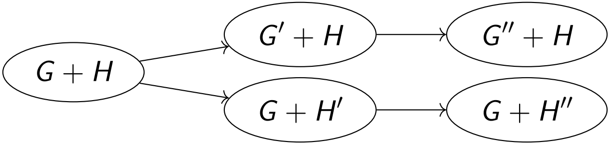
Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

Proof by induction

- If H is $\mathcal{P}$:



$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$
by induction hyp.
$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$

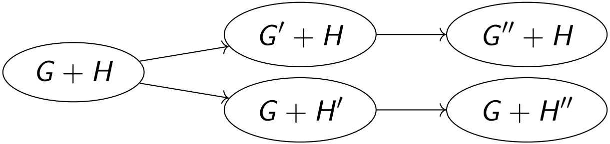
Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

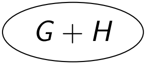
Proof by induction

- If H is $\mathcal{P}$:



$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$
by induction hyp.
$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$

- If H is $\mathcal{N}$:



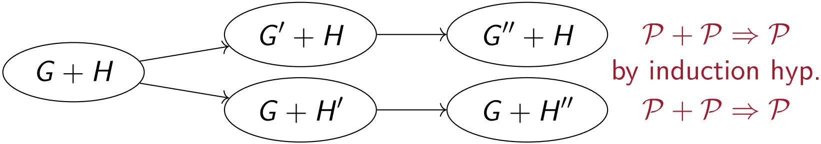
Summing a $\mathcal{P}$ game

Theorem

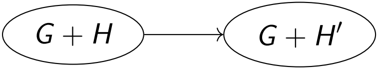
If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

Proof by induction

- If H is $\mathcal{P}$:



- If H is $\mathcal{N}$:



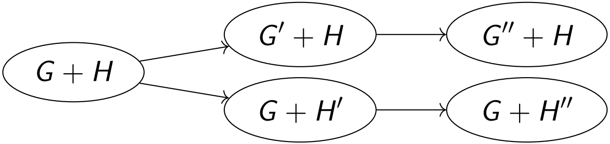
Summing a $\mathcal{P}$ game

Theorem

If G is $\mathcal{P}$, then $G + H$ has the same outcome than H.

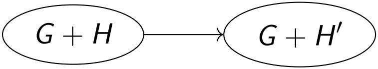
Proof by induction

- If H is $\mathcal{P}$:



$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$
by induction hyp.
$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$

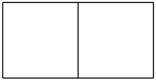
- If H is $\mathcal{N}$:



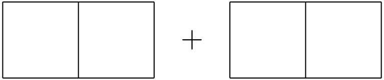
$\mathcal{P} + \mathcal{P} \Rightarrow \mathcal{P}$

Summing $\mathcal{N}$ games

Summing $\mathcal{N}$ games

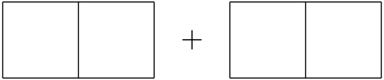


Summing $\mathcal{N}$ games

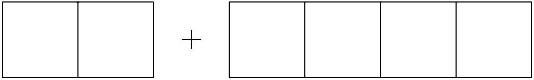


$\Rightarrow \mathcal{P}$

Summing $\mathcal{N}$ games

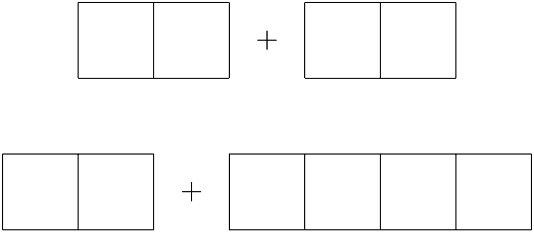


$\Rightarrow \mathcal{P}$



$\Rightarrow \mathcal{N}$

Summing $\mathcal{N}$ games



$\Rightarrow \mathcal{P}$

$\Rightarrow \mathcal{N}$

$\Rightarrow$ We need to define equivalence classes for games

Grundy values

Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy values

Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

Grundy values

Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.

Grundy values

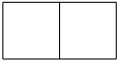
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.



Grundy values

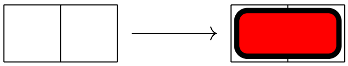
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.



Grundy values

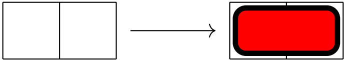
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.



$\mathcal{G} = 0$

Grundy values

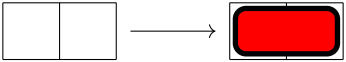
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.

$\mathcal{G} = 1$  $\mathcal{G} = 0$

Grundy values

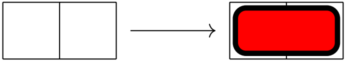
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.

$\mathcal{G} = 1$  $\mathcal{G} = 0$

Grundy values

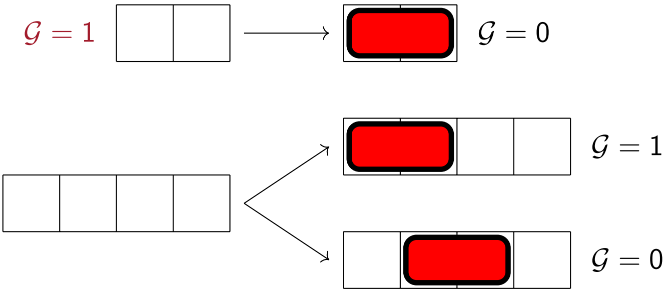
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.



Grundy values

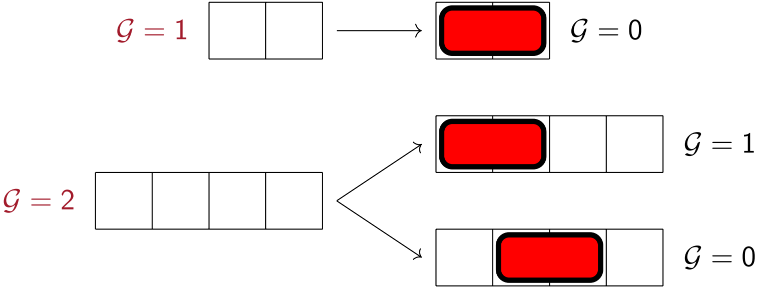
Equivalence of games

$G \equiv H \Leftrightarrow G + H$ is $\mathcal{P}$.

Grundy value

Mapping the equivalence classes to nonnegative integers.

- $\mathcal{G}(G) = 0 \Leftrightarrow G$ is $\mathcal{P}$.
- $\mathcal{G}(G)$ is the mex of the Grundy values of its options.



Sprague-Grundy Theorem

Sprague-Grundy Theorem

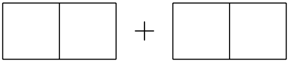
Theorem (Sprague 1935, Grundy 1939)

$\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$

Sprague-Grundy Theorem

Theorem (Sprague 1935, Grundy 1939)

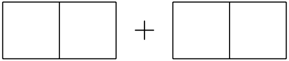
$\mathcal{G}(G+H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$



Sprague-Grundy Theorem

Theorem (Sprague 1935, Grundy 1939)

$\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$

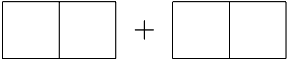


$$1 \oplus 1$$

Sprague-Grundy Theorem

Theorem (Sprague 1935, Grundy 1939)

$\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$

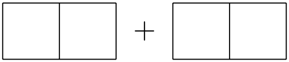


$$1 \oplus 1 = 0$$

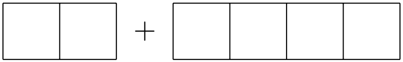
Sprague-Grundy Theorem

Theorem (Sprague 1935, Grundy 1939)

$\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$



$$1 \oplus 1 = 0$$

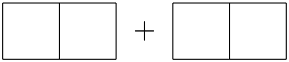


$$1 \oplus 2 = 3$$

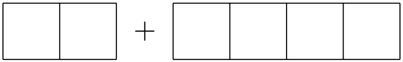
Sprague-Grundy Theorem

Theorem (Sprague 1935, Grundy 1939)

$\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H)$



$$1 \oplus 1 = 0$$



$$1 \oplus 2 = 3$$

Interpretation

Every impartial game is equivalent to a NIM heap.

The Rooks game: motivation

The Rooks game: motivation

The queens problem (Bezzel, 1848)

How many queens can one place on a chessboard without them attacking each other?

The Rooks game: motivation

The queens problem (Bezzel, 1848)

How many queens can one place on a chessboard without them attacking each other?

The queens game (Noon and Van Brummelen, 2006)

Two players alternate placing queens on a chessboard without attacking an already placed queen. The player who places the last queen wins.

The Rooks game: motivation

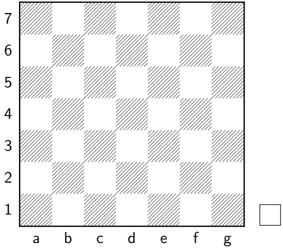
The queens problem (Bezzel, 1848)

How many queens can one place on a chessboard without them attacking each other?

The queens game (Noon and Van Brummelen, 2006)

Two players alternate placing queens on a chessboard without attacking an already placed queen. The player who places the last queen wins.

→ Solved on odd square chessboards.



The Rooks game: motivation

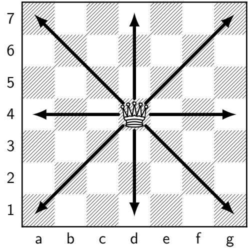
The queens problem (Bezzel, 1848)

How many queens can one place on a chessboard without them attacking each other?

The queens game (Noon and Van Brummelen, 2006)

Two players alternate placing queens on a chessboard without attacking an already placed queen. The player who places the last queen wins.

→ Solved on odd square chessboards.



The Rooks game: motivation

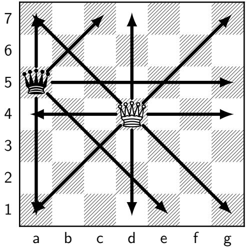
The queens problem (Bezzel, 1848)

How many queens can one place on a chessboard without them attacking each other?

The queens game (Noon and Van Brummelen, 2006)

Two players alternate placing queens on a chessboard without attacking an already placed queen. The player who places the last queen wins.

→ Solved on odd square chessboards.



□

The Rooks game: motivation

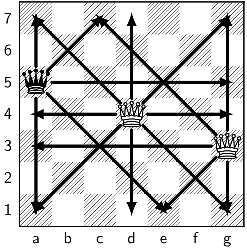
The queens problem (Bezzel, 1848)

How many queens can one place on a chessboard without them attacking each other?

The queens game (Noon and Van Brummelen, 2006)

Two players alternate placing queens on a chessboard without attacking an already placed queen. The player who places the last queen wins.

$\rightarrow$ Solved on odd square chessboards. $\Rightarrow \mathcal{N}$ by symmetry



□

The Rooks game

The rooks game

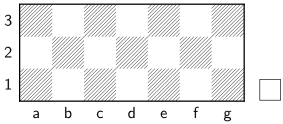
Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

The Rooks game

The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

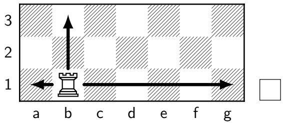


The Rooks game

The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

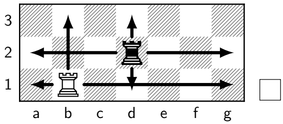


The Rooks game

The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

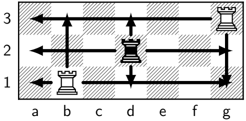


The Rooks game

The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.



□

The Rooks game

The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

The rooks game on a holed chessboard

Same than the rooks game, but the chessboard has holes. Rooks cannot attack through holes.

The Rooks game

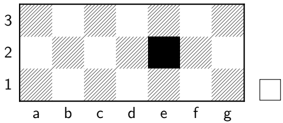
The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

The rooks game on a holed chessboard

Same than the rooks game, but the chessboard has holes. Rooks cannot attack through holes.



The Rooks game

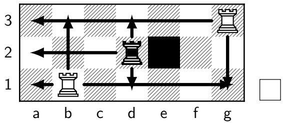
The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

The rooks game on a holed chessboard

Same than the rooks game, but the chessboard has holes. Rooks cannot attack through holes.



The Rooks game

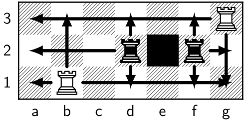
The rooks game

Two players alternate placing rooks on a chessboard without attacking an already placed rook. The player who places the last rook wins.

On rectangular chessboards: parity.

The rooks game on a holed chessboard

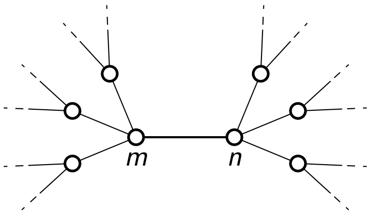
Same than the rooks game, but the chessboard has holes. Rooks cannot attack through holes.



A model for the rooks game: Weighted Arc-Kayles

Weighted Arc-Kayles (or WAK)

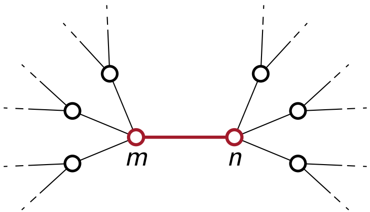
- Played on a weighted graph $G = (V, E, \omega)$ with $\omega : V \to \mathbb{N}$.
- The players alternate selecting edges.
- The weight of both endpoints is decreased by 1.
- Vertices with weight zero are removed.
- When there are no edges left, the game ends.



A model for the rooks game: Weighted Arc-Kayles

Weighted Arc-Kayles (or WAK)

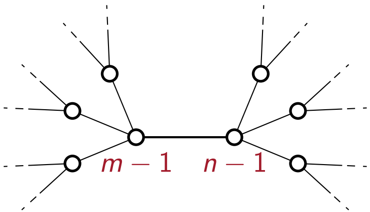
- Played on a weighted graph $G = (V, E, \omega)$ with $\omega : V \to \mathbb{N}$.
- The players alternate selecting edges.
- The weight of both endpoints is decreased by 1.
- Vertices with weight zero are removed.
- When there are no edges left, the game ends.



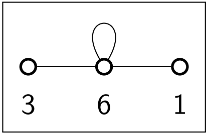
A model for the rooks game: Weighted Arc-Kayles

Weighted Arc-Kayles (or WAK)

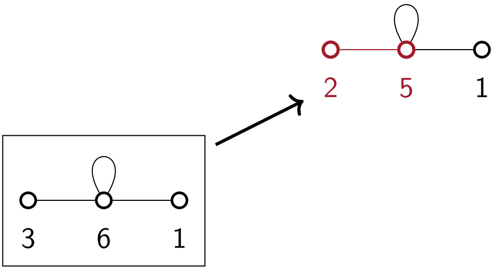
- Played on a weighted graph $G = (V, E, \omega)$ with $\omega : V \to \mathbb{N}$.
- The players alternate selecting edges.
- The weight of both endpoints is decreased by 1.
- Vertices with weight zero are removed.
- When there are no edges left, the game ends.



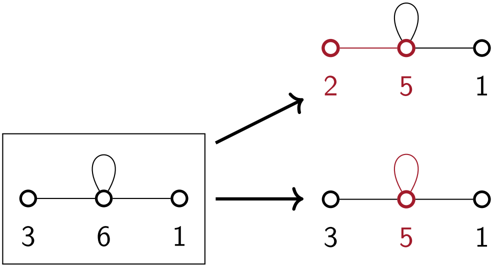
Example



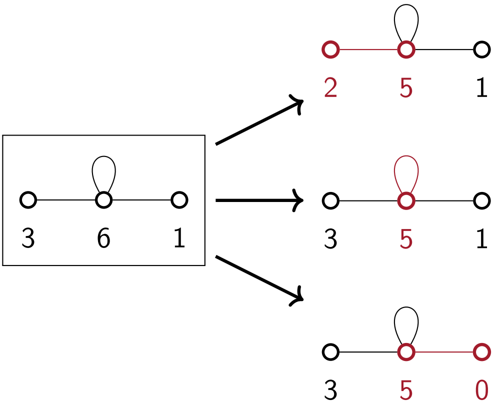
Example



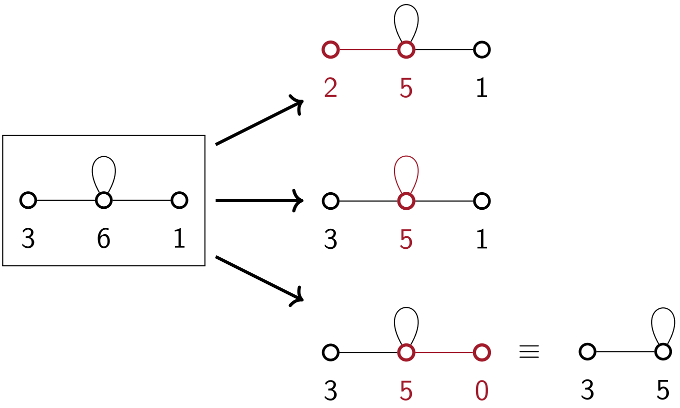
Example



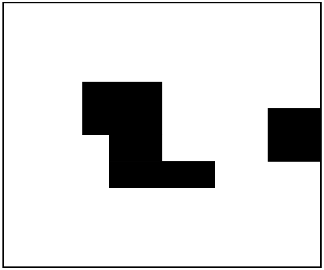
Example



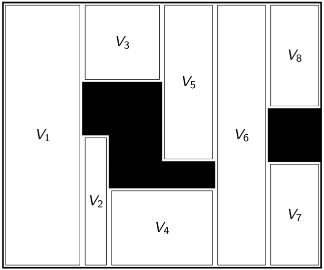
Example

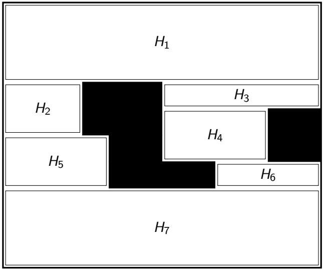


WAK is the rooks game

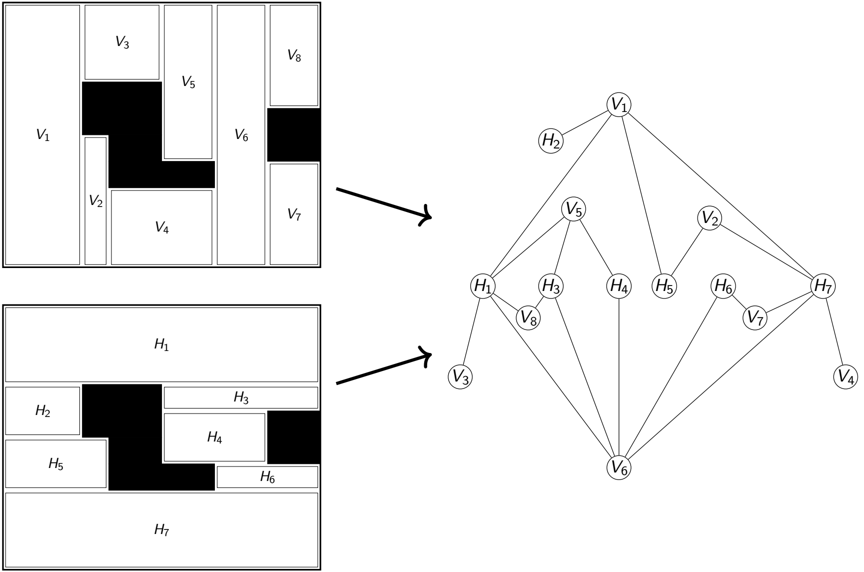


WAK is the rooks game

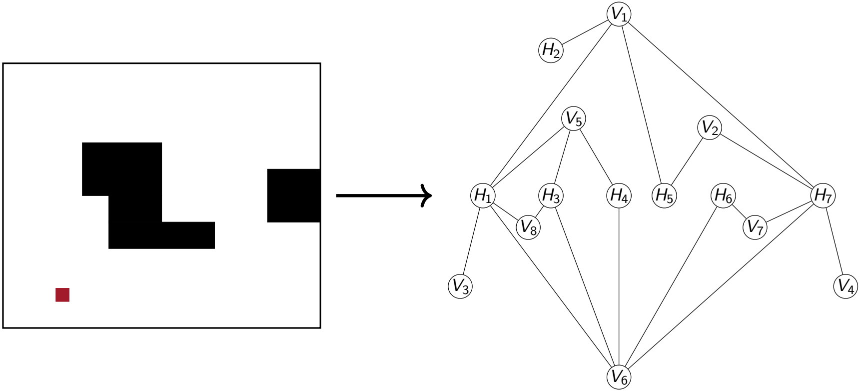




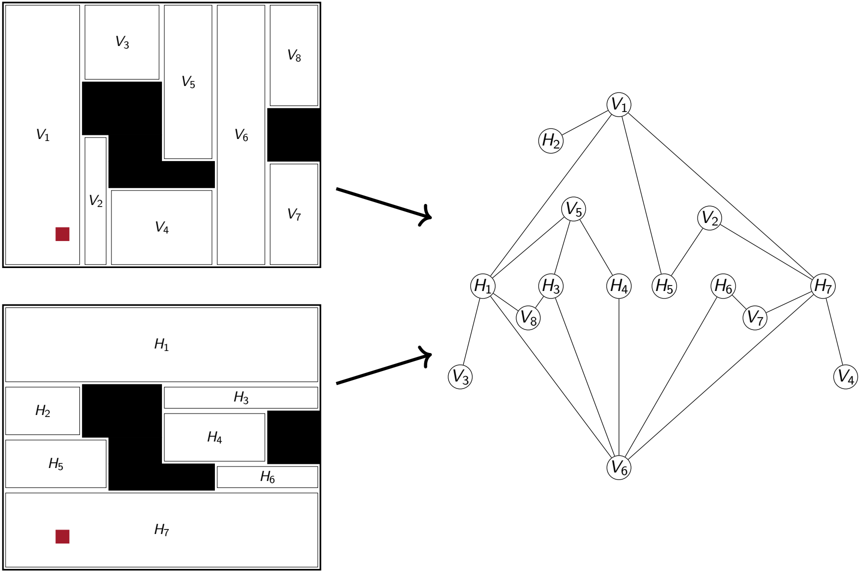
WAK is the rooks game



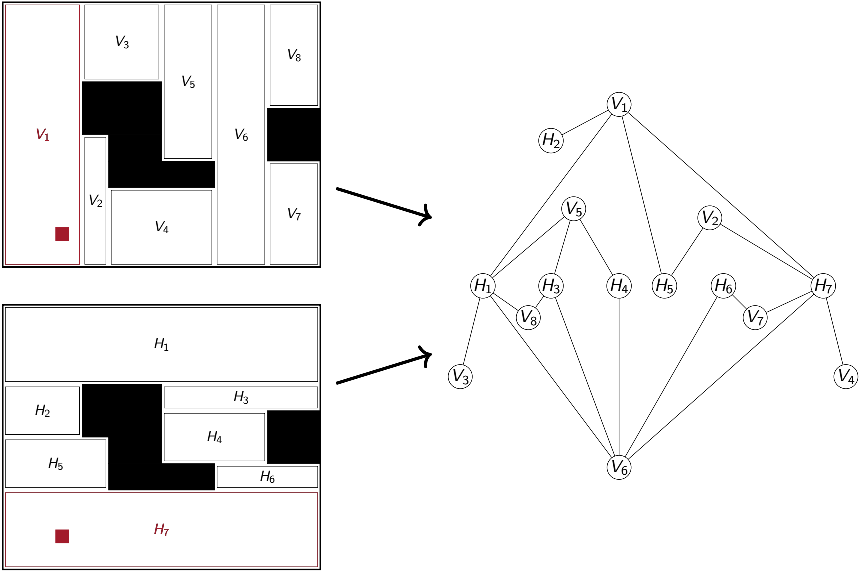
WAK is the rooks game



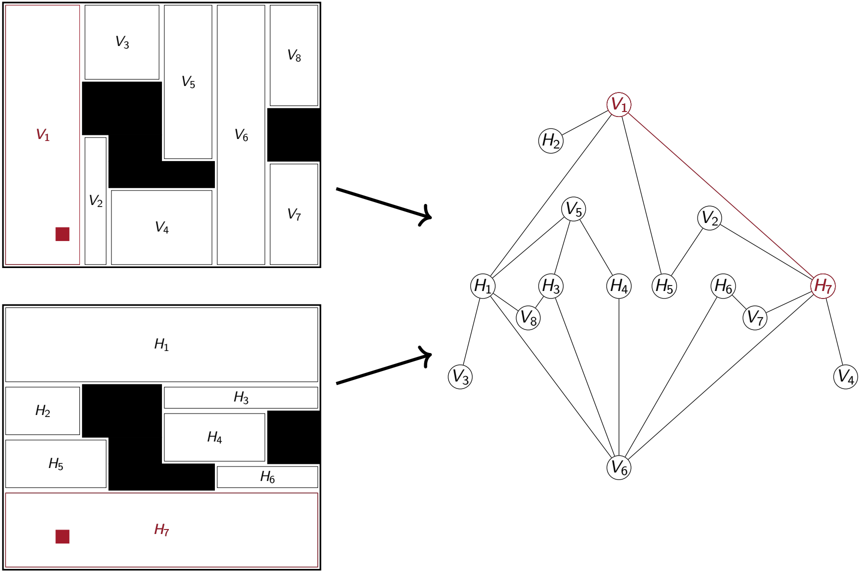
WAK is the rooks game



WAK is the rooks game

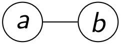


WAK is the rooks game

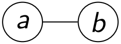


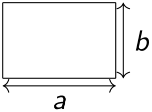
First results on WAK

First results on WAK

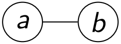


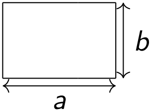
First results on WAK





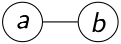
First results on WAK

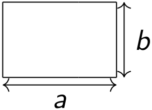




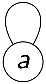
$\mathcal{G} = min(a, b) \bmod 2$

First results on WAK

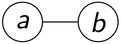


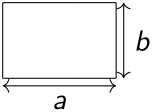


$\mathcal{G} = min(a, b) \text{ mod } 2$

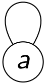


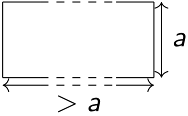
First results on WAK



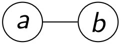


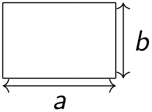
$\mathcal{G} = min(a, b) \bmod 2$



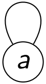


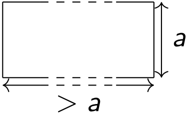
First results on WAK





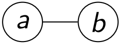
$\mathcal{G} = min(a, b) \bmod 2$

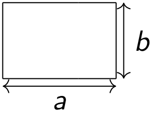




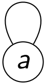
$\mathcal{G} = a \bmod 2$

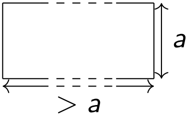
First results on WAK





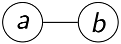
$\mathcal{G} = min(a, b) \bmod 2$



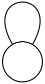


$\mathcal{G} = a \bmod 2$

$\Rightarrow$



$\equiv$



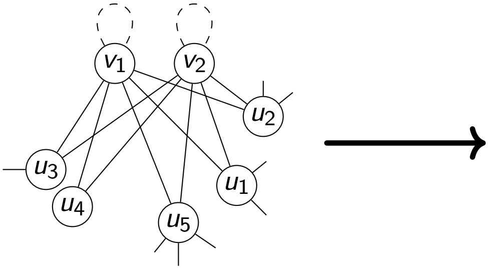
$min(a, b)$

Reduction Lemmas

Reduction Lemmas

Twin vertices lemma

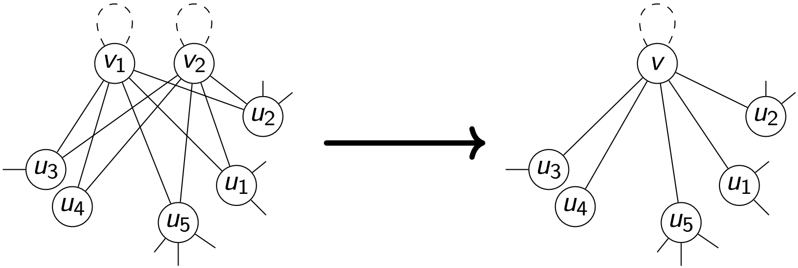
If two vertices are exact false twins (including loop edges),



Reduction Lemmas

Twin vertices lemma

If two vertices are exact false twins (including loop edges), then they can be fused together.

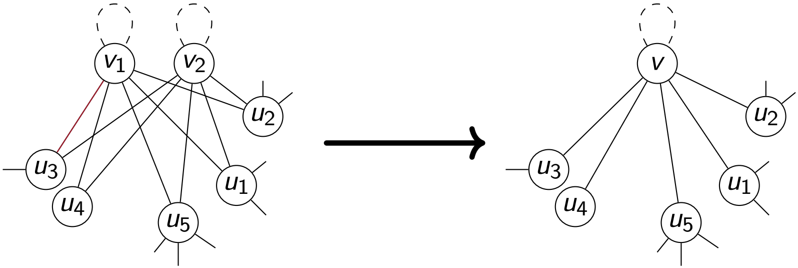


$$\omega(v) = \omega(v_1) + \omega(v_2)$$

Reduction Lemmas

Twin vertices lemma

If two vertices are exact false twins (including loop edges), then they can be fused together.

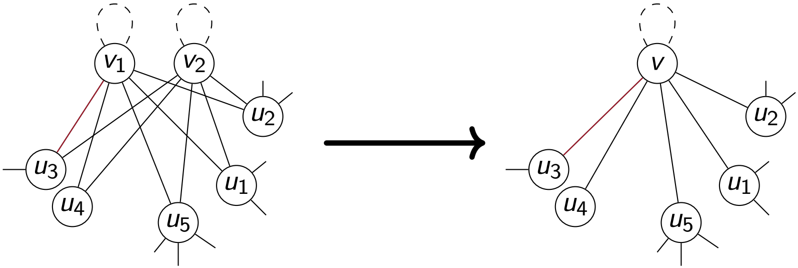


$$\omega(v) = \omega(v_1) + \omega(v_2)$$

Reduction Lemmas

Twin vertices lemma

If two vertices are exact false twins (including loop edges), then they can be fused together.

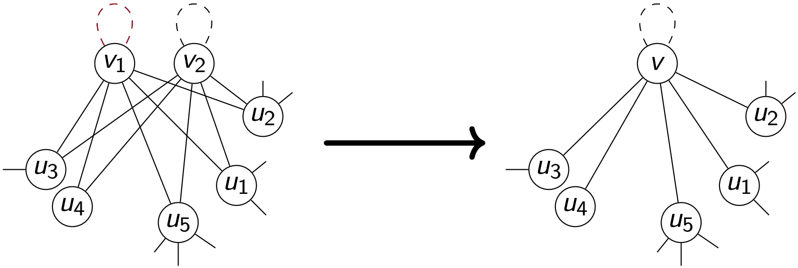


$$\omega(v) = \omega(v_1) + \omega(v_2)$$

Reduction Lemmas

Twin vertices lemma

If two vertices are exact false twins (including loop edges), then they can be fused together.

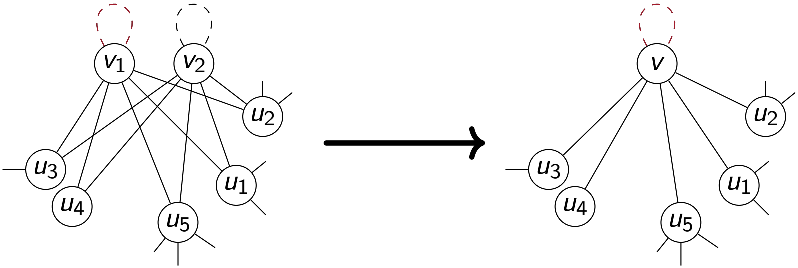


$$\omega(v) = \omega(v_1) + \omega(v_2)$$

Reduction Lemmas

Twin vertices lemma

If two vertices are exact false twins (including loop edges), then they can be fused together.



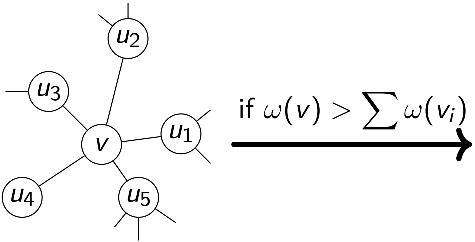
$$\omega(v) = \omega(v_1) + \omega(v_2)$$

Reduction Lemmas

Reduction Lemmas

Heavy vertex lemma

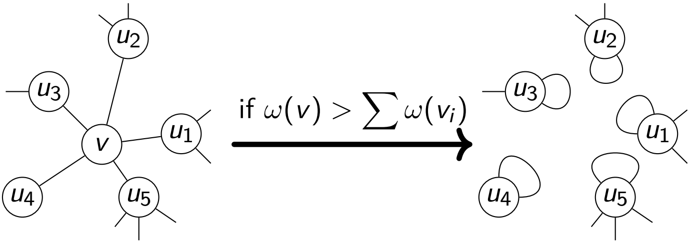
If a vertex without loop has a weight greater than the sum of its neighbour's weights,



Reduction Lemmas

Heavy vertex lemma

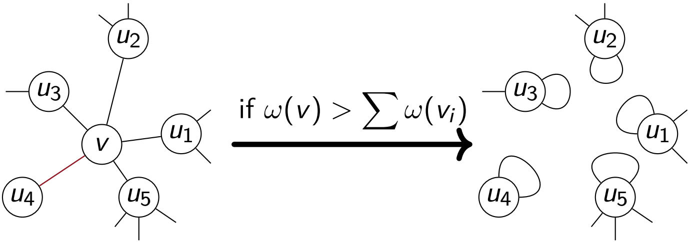
If a vertex without loop has a weight greater than the sum of its neighbour's weights, then it can be removed and loops added to its neighbours.



Reduction Lemmas

Heavy vertex lemma

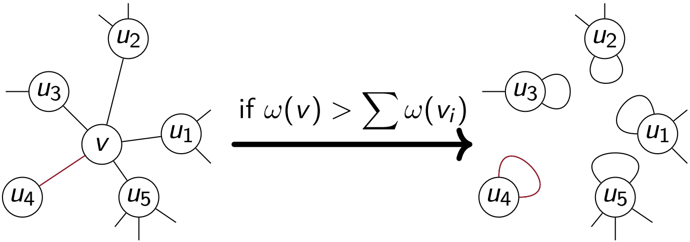
If a vertex without loop has a weight greater than the sum of its neighbour's weights, then it can be removed and loops added to its neighbours.



Reduction Lemmas

Heavy vertex lemma

If a vertex without loop has a weight greater than the sum of its neighbour's weights, then it can be removed and loops added to its neighbours.

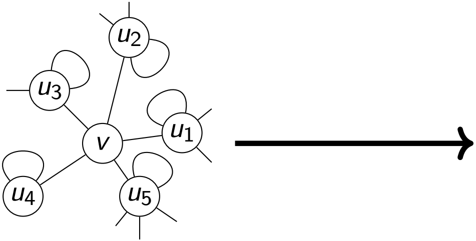


Reduction Lemmas

Reduction Lemmas

Useless vertex lemma

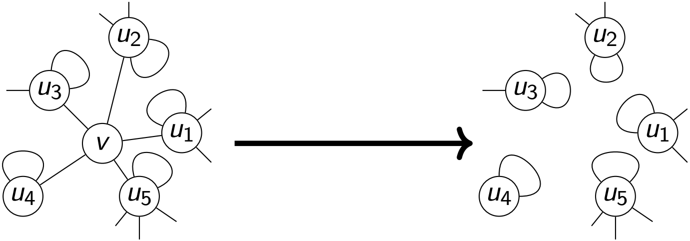
If all the neighbours of a vertex without loop have a loop,



Reduction Lemmas

Useless vertex lemma

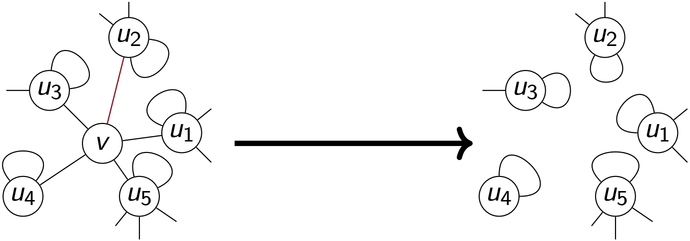
If all the neighbours of a vertex without loop have a loop, then it can be removed.



Reduction Lemmas

Useless vertex lemma

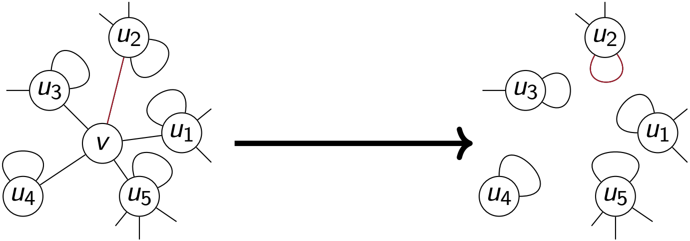
If all the neighbours of a vertex without loop have a loop, then it can be removed.



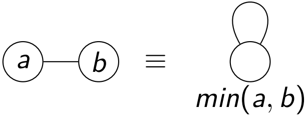
Reduction Lemmas

Useless vertex lemma

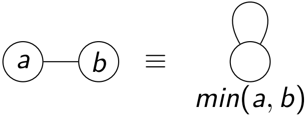
If all the neighbours of a vertex without loop have a loop, then it can be removed.



Canonical form

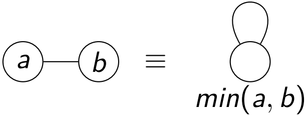


Canonical form



⇒ Application of the Heavy vertex lemma

Canonical form

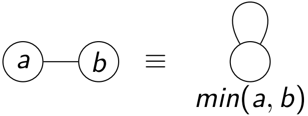


$\Rightarrow$ Application of the Heavy vertex lemma

Canonical form

A graph is canonical if it has no false twin, no heavy vertex and no useless vertex.

Canonical form



$\Rightarrow$ Application of the Heavy vertex lemma

Canonical form

A graph is canonical if it has no false twin, no heavy vertex and no useless vertex.

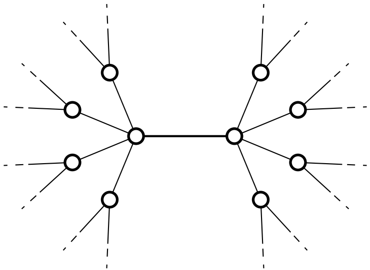
Proposition

If G is a graph and H its canonical form after application of the reduction lemmas, then $\mathcal{G}(G) = \mathcal{G}(H)$.

What about unweighted Arc-Kayles?

Arc-Kayles (Schaefer, 1978)

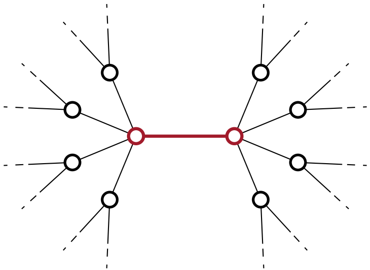
- This game is played on a graph $G = (V, E)$.
- The players alternate selecting edges.
- Its endpoints are deleted.
- When there are no edges left, the game ends.



What about unweighted Arc-Kayles?

Arc-Kayles (Schaefer, 1978)

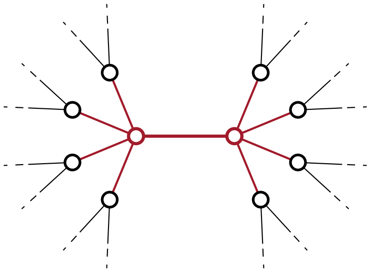
- This game is played on a graph $G = (V, E)$.
- The players alternate selecting edges.
- Its endpoints are deleted.
- When there are no edges left, the game ends.



What about unweighted Arc-Kayles?

Arc-Kayles (Schaefer, 1978)

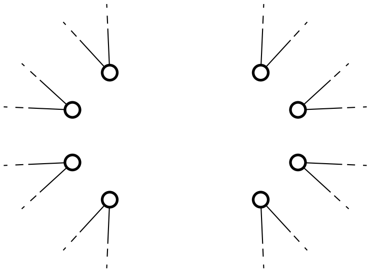
- This game is played on a graph $G = (V, E)$.
- The players alternate selecting edges.
- Its endpoints are deleted.
- When there are no edges left, the game ends.



What about unweighted ARC-KAYLES?

ARC-KAYLES (Schaefer, 1978)

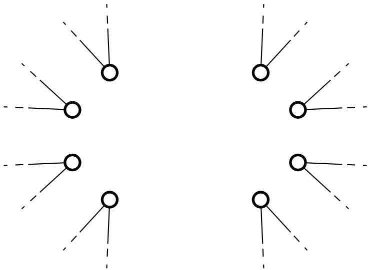
- This game is played on a graph $G = (V, E)$.
- The players alternate selecting edges.
- Its endpoints are deleted.
- When there are no edges left, the game ends.



What about unweighted Arc-Kayles?

Arc-Kayles (Schaefer, 1978)

- This game is played on a graph $G = (V, E)$.
- The players alternate selecting edges.
- Its endpoints are deleted.
- When there are no edges left, the game ends.



$\Rightarrow$ Arc-Kayles is WAK with $\omega(u) = 1$ for all vertex u

ARC-KAYLES: a history

Arc-Kayles: a history

- 1976: introduction (Schaeffer)

Arc-Kayles: a history

- 1976: introduction (Schaeffer)
- 1956: solved on paths (Guy and Smith)

Arc-Kayles: a history

- 1976: introduction (Schaeffer)
- 1956: solved on paths (Guy and Smith)
- 2014: FPT complexity (Lampis and Mitsou)

Arc-Kayles: a history

- 1976: introduction (Schaeffer)
- 1956: solved on paths (Guy and Smith)
- 2014: FPT complexity (Lampis and Mitsou)
- 2016: solved on cycles, wheels and stars with 3 paths (Huggan and Stevens)

Arc-Kayles: a history

- 1976: introduction (Schaeffer)
- 1956: solved on paths (Guy and Smith)
- 2014: FPT complexity (Lampis and Mitsou)
- 2016: solved on cycles, wheels and stars with 3 paths (Huggan and Stevens)

→ Links with many other games (Cram, octal games ...)

From WAK to ARC-KAYLES

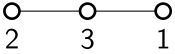
WEIGHTED ARC-KAYLES

ARC-KAYLES

From WAK to Arc-Kayles

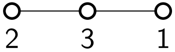
Weighted Arc-Kayles

Arc-Kayles



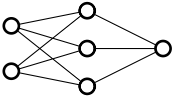
From WAK to Arc-Kayles

Weighted Arc-Kayles



$\Longleftrightarrow$

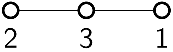
Arc-Kayles



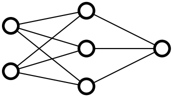
From WAK to Arc-Kayles

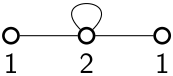
Weighted Arc-Kayles

Arc-Kayles



$\Longleftrightarrow$

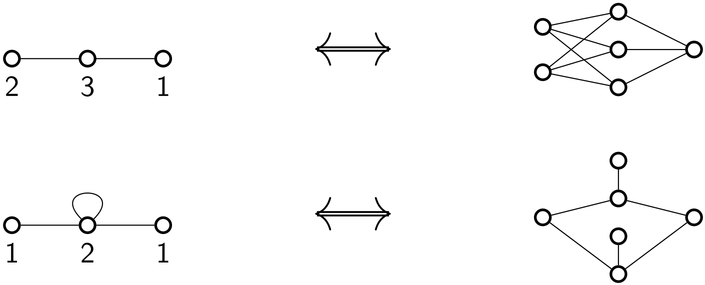




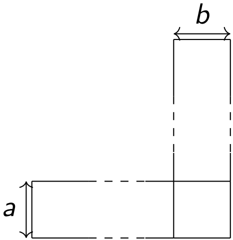
From WAK to Arc-Kayles

Weighted Arc-Kayles

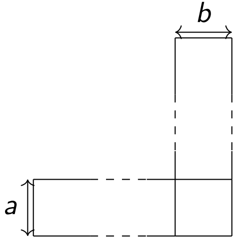
Arc-Kayles

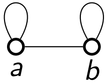


A chessboard with a hole in the corner

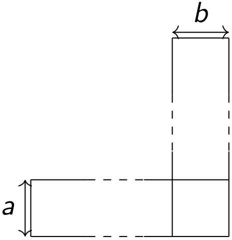


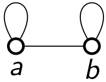
A chessboard with a hole in the corner

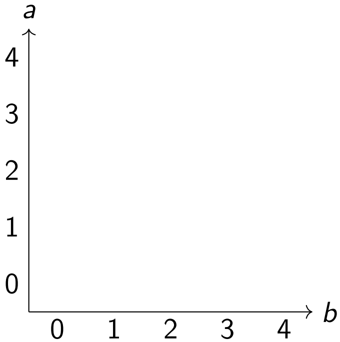




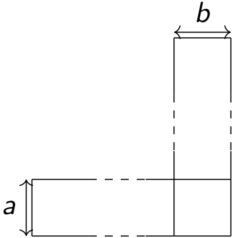
A chessboard with a hole in the corner

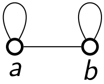


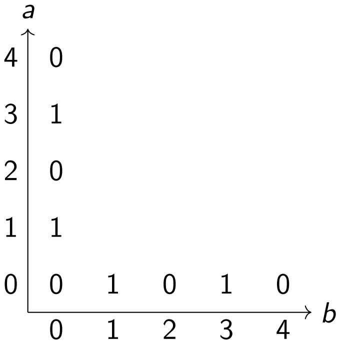




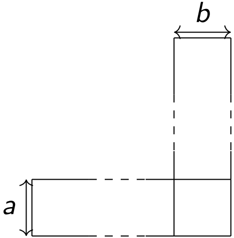
A chessboard with a hole in the corner

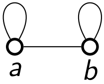


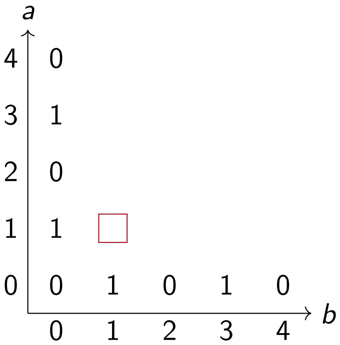




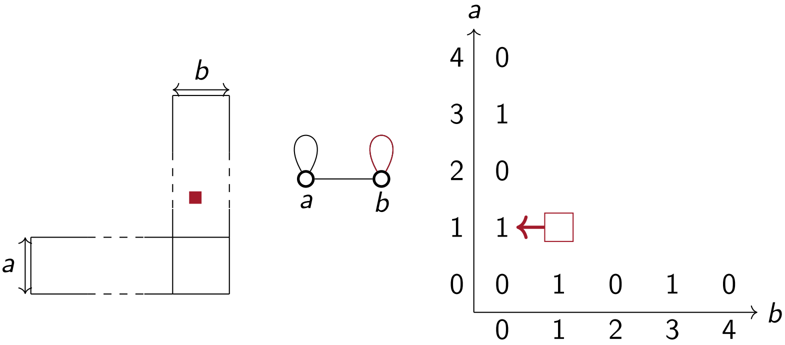
A chessboard with a hole in the corner



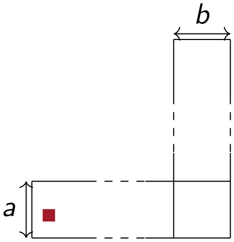


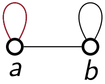


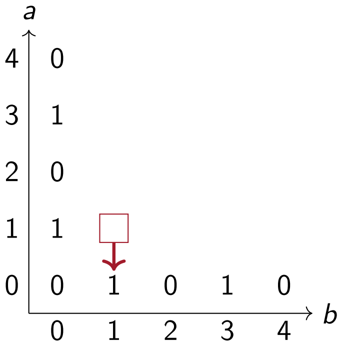
A chessboard with a hole in the corner



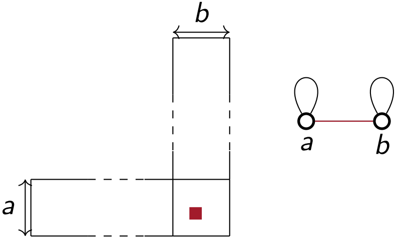
A chessboard with a hole in the corner

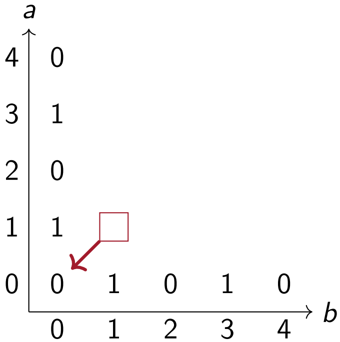




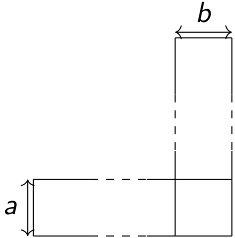


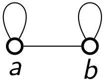
A chessboard with a hole in the corner





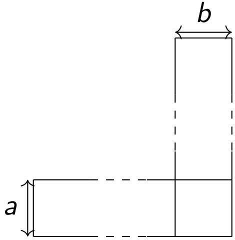
A chessboard with a hole in the corner

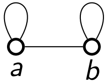


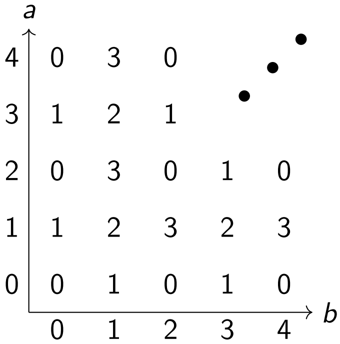


a \ b	0	1	2	3	4
4	0	3			
3	1	2			
2	0	3			
1	1	2	3	2	3
0	0	1	0	1	0

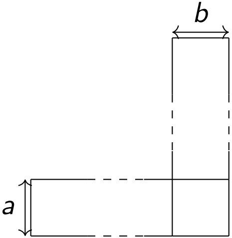
A chessboard with a hole in the corner

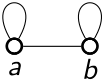


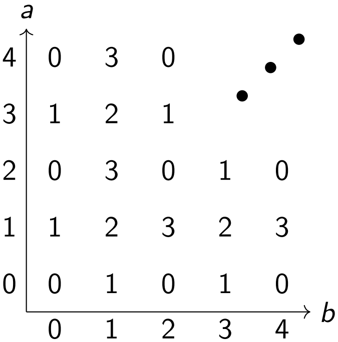




A chessboard with a hole in the corner





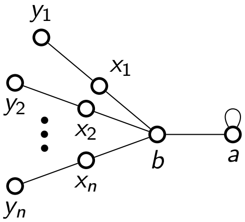


$\mathcal{P} \Leftrightarrow a$ and b even

Trees of depth at most 2

Theorem

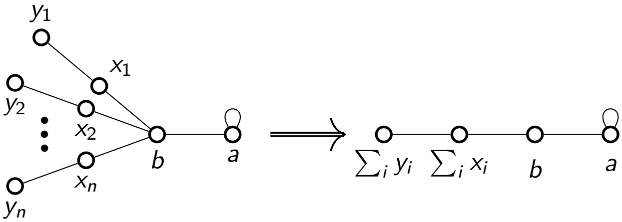
Let $x_i > y_i$



Trees of depth at most 2

Theorem

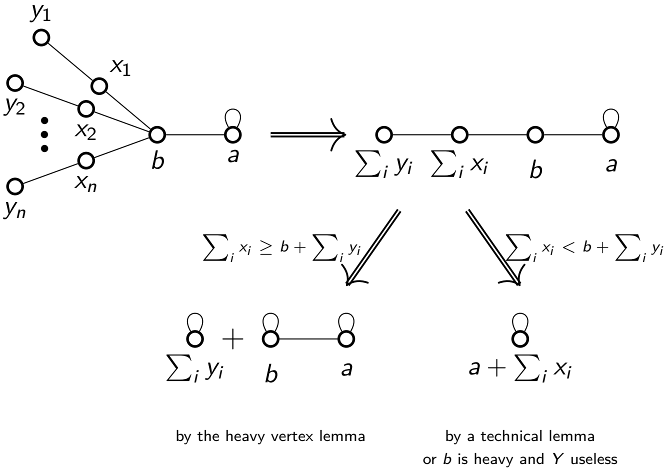
Let $x_i > y_i$



Trees of depth at most 2

Theorem

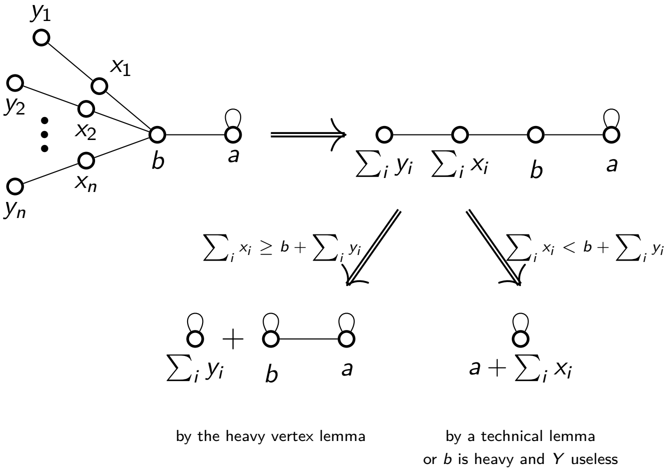
Let $x_i > y_i$



Trees of depth at most 2

Theorem

Let $x_i > y_i$

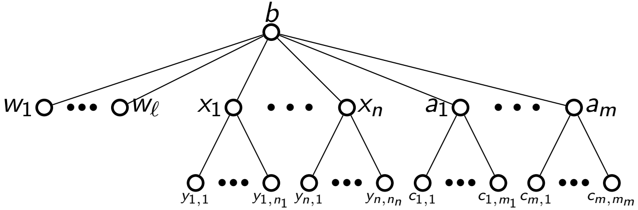


Same outcome, not equivalence!

Trees of depth at most 2

Theorem

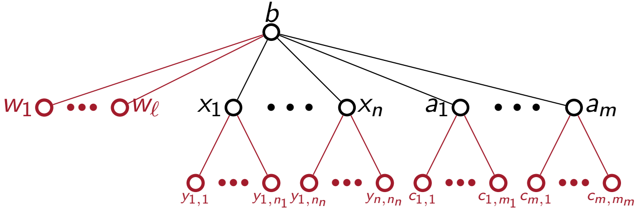
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

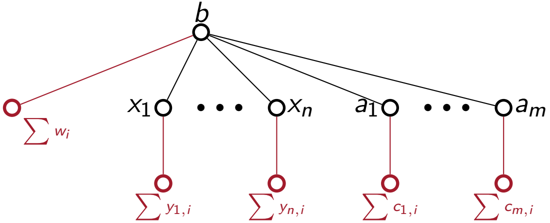
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

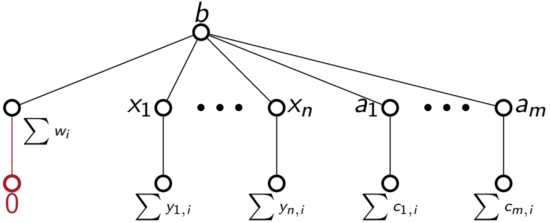
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

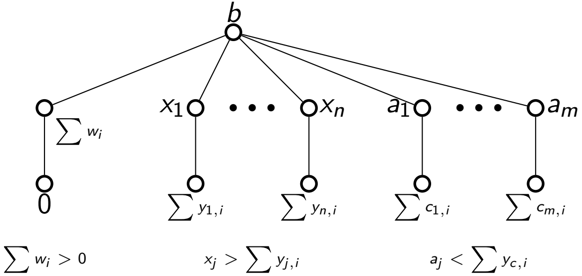
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

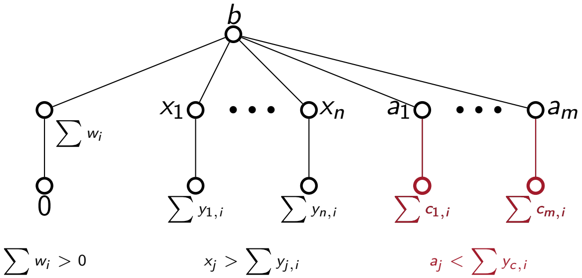
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

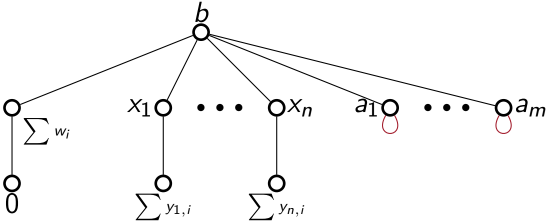
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.

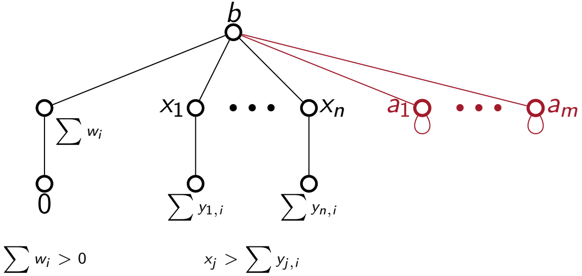


$\sum w_i > 0$ $\qquad$ $x_j > \sum y_{j,i}$

Trees of depth at most 2

Theorem

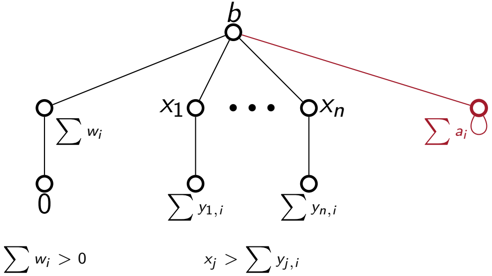
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

Theorem

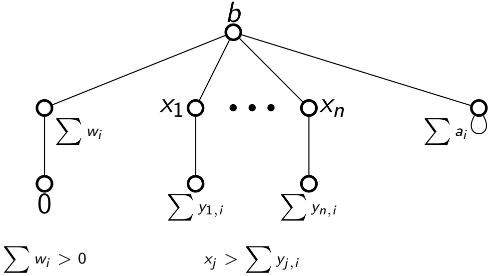
There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.



Trees of depth at most 2

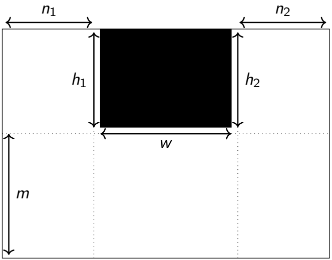
Theorem

There is a polynomial-time algorithm computing the outcome of a tree of depth at most 2.

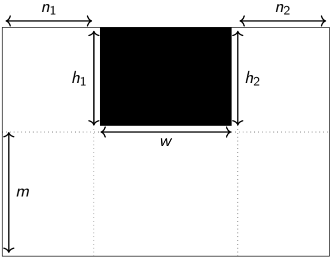


$\rightarrow$ Now we can apply the previous theorem and find the outcome

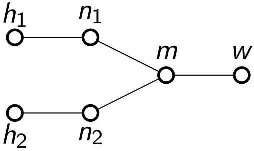
Trees of depth at most 2: implication



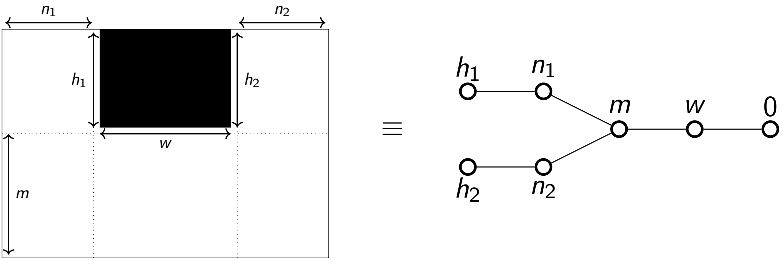
Trees of depth at most 2: implication



$\equiv$

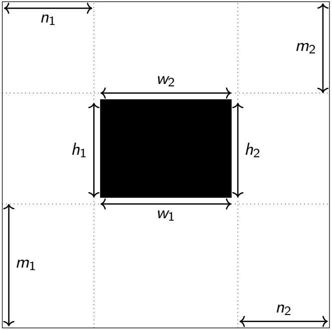


Trees of depth at most 2: implication

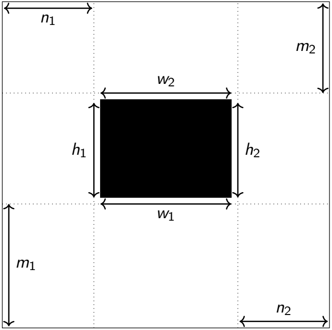


→ Tree of depth 2

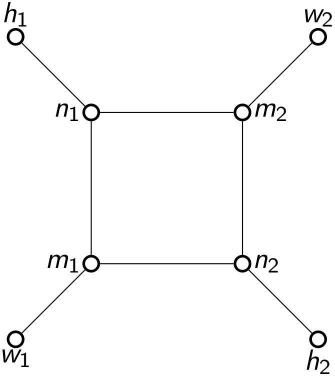
Other chessboards?



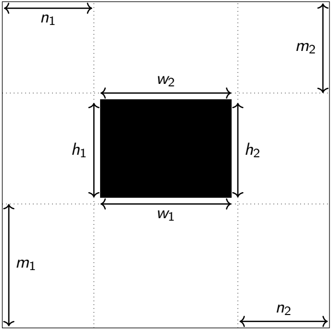
Other chessboards?



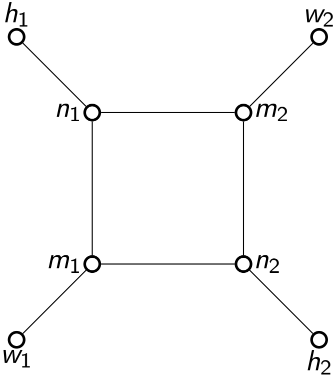
$\equiv$



Other chessboards?



$\equiv$



$\rightarrow$ Hard...

Are the Grundy values bounded?

Are the Grundy values bounded?

Theorem

The Grundy values for WAK are unbounded.

Are the Grundy values bounded?

Theorem

The Grundy values for WAK are unbounded.

Proof (by induction)

Construct a sequence $G_1, G_2, \ldots$ such that:

- $\mathcal{G}(G_i) \neq \mathcal{G}(G_j)$ for $j < i$
- A winning move is by removing a certain vertex u_i
- Every vertex has weight 1

Are the Grundy values bounded?

Theorem

The Grundy values for WAK are unbounded.

Proof (by induction)

Construct a sequence $G_1, G_2, \ldots$ such that:

- $\mathcal{G}(G_i) \neq \mathcal{G}(G_j)$ for $j < i$
- A winning move is by removing a certain vertex u_i
- Every vertex has weight 1

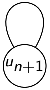
$G_1 =$ 

Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n + 1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1

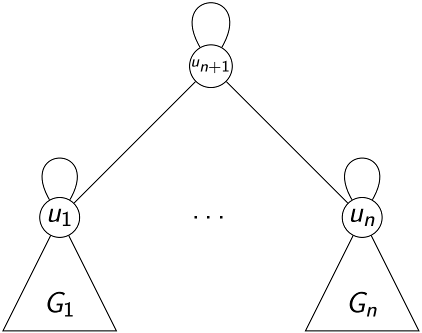
Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1



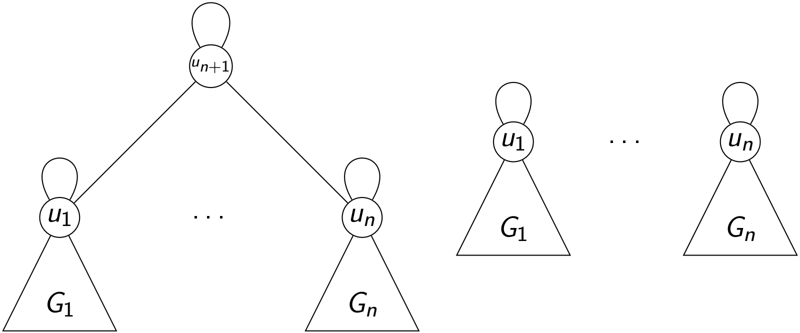
Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1



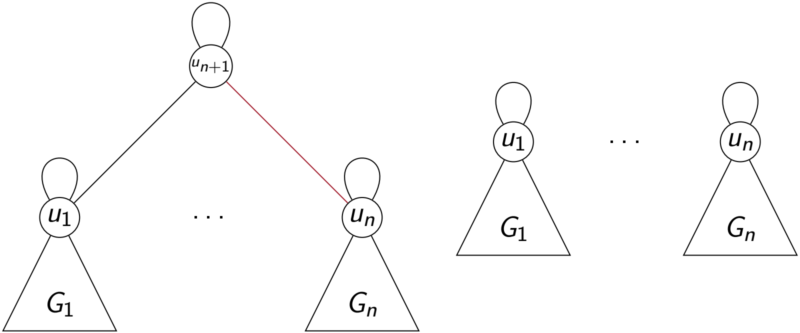
Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



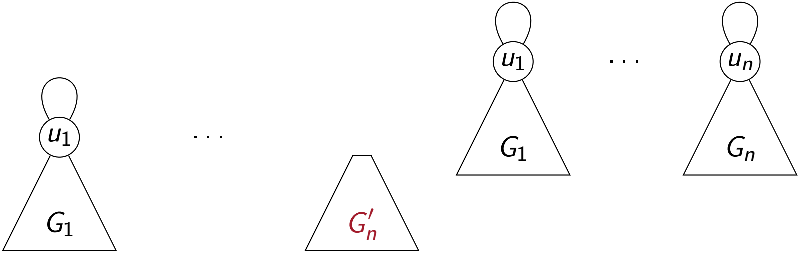
Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



Unboundedness of Grundy values: building G_{n+1}

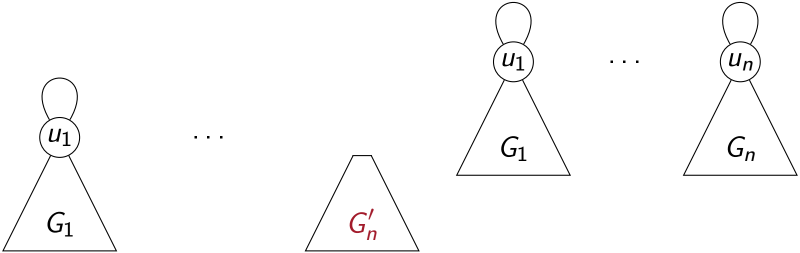
- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



$\forall i, \mathcal{G}(G'_i) = 0$ by induction hypothesis,

Unboundedness of Grundy values: building G_{n+1}

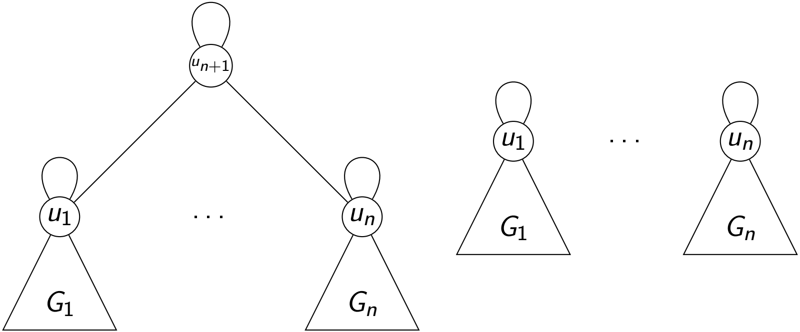
- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



$\forall i, \mathcal{G}(G'_i) = 0$ by induction hypothesis, this graph has $\mathcal{G} = \mathcal{G}(G_i)$.

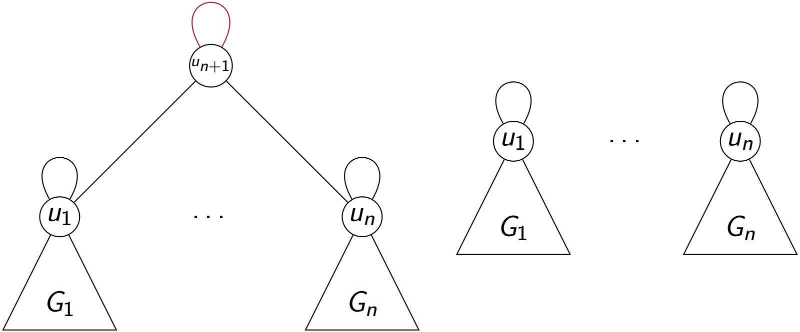
Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$ ✓
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



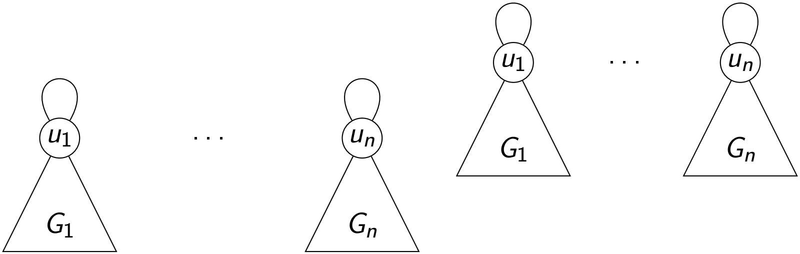
Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$ ✓
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



Unboundedness of Grundy values: building G_{n+1}

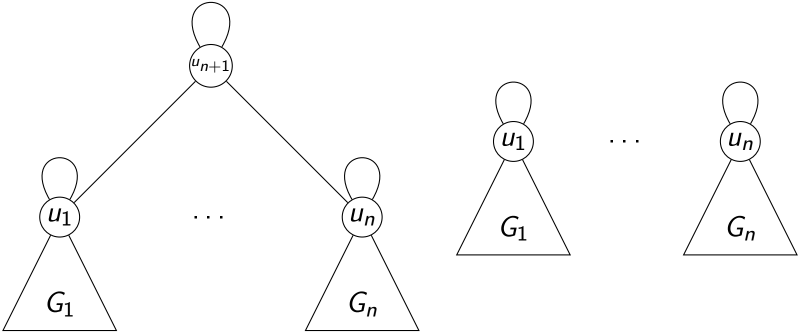
- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n + 1$ ✓
- A winning move is by removing a certain vertex u_{n+1}
- Every vertex has weight 1 ✓



This graph has $\mathcal{G} = 0$.

Unboundedness of Grundy values: building G_{n+1}

- $\mathcal{G}(G_n) \neq \mathcal{G}(G_j)$ for $j < n+1$ ✓
- A winning move is by removing a certain vertex u_{n+1} ✓
- Every vertex has weight 1 ✓



$\Rightarrow$ Infinite sequence of graphs with distinct Grundy values.

Unboundedness of Grundy values

Theorem

The Grundy values for WAK are unbounded.

Unboundedness of Grundy values

Theorem

The Grundy values for WAK are unbounded.

Corollary

The Grundy values for Arc-Kayles are unbounded.
The Grundy values for Node-Kayles are unbounded.

Conclusion

Results

- Introduction of WAK, links with the rooks game
- Outcome of WAK on trees of depth at most 2
- Unboundedness of Grundy values

Conclusion

Results

- Introduction of WAK, links with the rooks game
- Outcome of WAK on trees of depth at most 2
- Unboundedness of Grundy values

Perspectives

- Solving more complex graphs and graphs for the rooks game
- Studying the complexity of WAK

Conclusion

Results

- Introduction of WAK, links with the rooks game
- Outcome of WAK on trees of depth at most 2
- Unboundedness of Grundy values

Perspectives

- Solving more complex graphs and graphs for the rooks game
- Studying the complexity of WAK