Combinatorial Games on Graphs

Éric SOPENA

LaBRI, University of Bordeaux

France

CALDAM Indo-French Pre-Conference School on Algorithms and Combinatorics

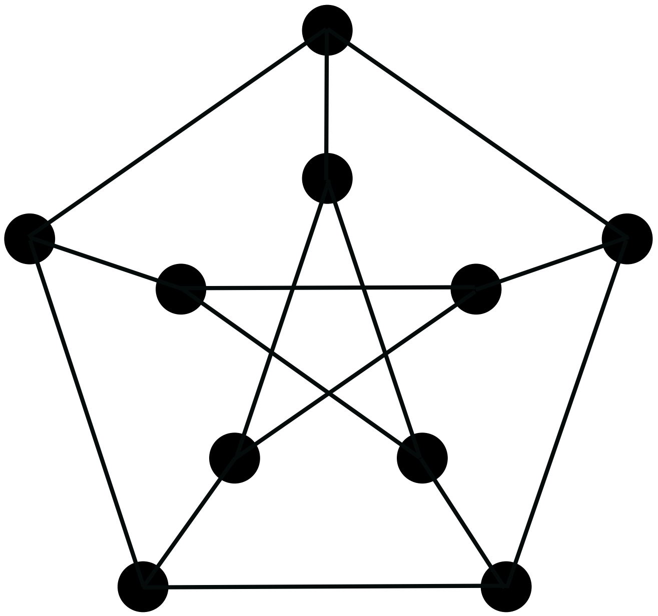
February 10-11, 2020

IIT Hyderabad

Let’s first play…

Let’s first play…

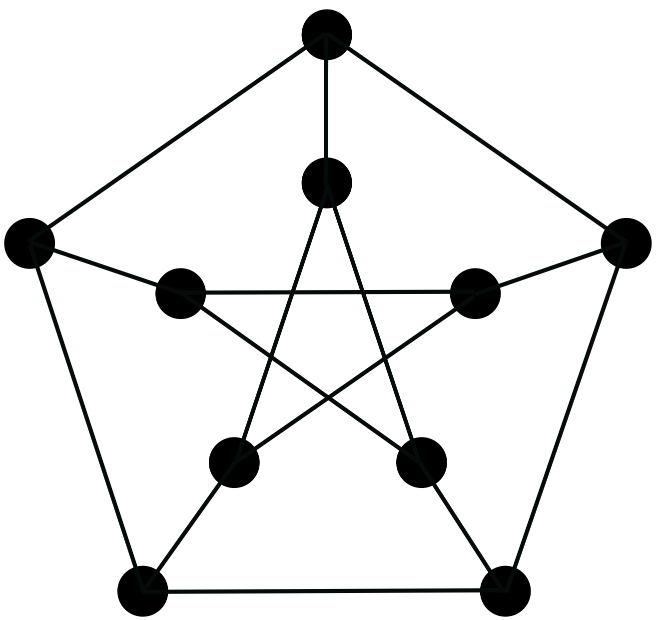
Take your favorite graph, e.g. Petersen graph.



Let's first play...

Take your favorite graph, e.g. Petersen graph.

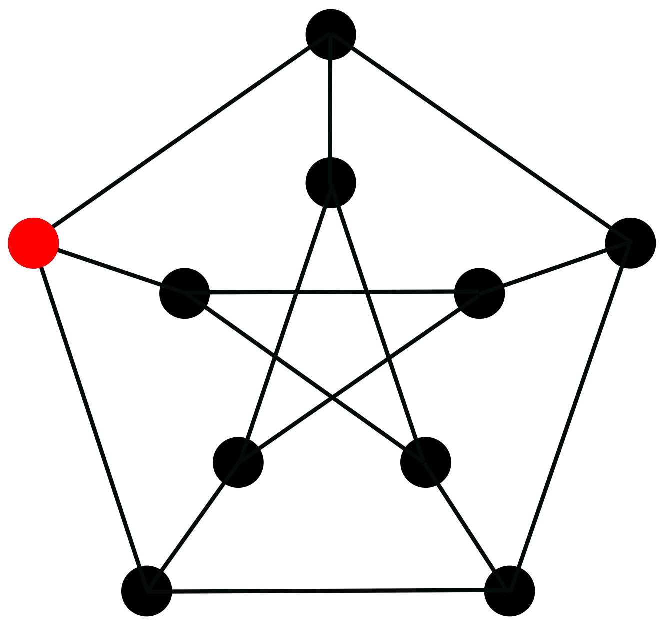
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Let’s first play...

Take your favorite graph, e.g. Petersen graph.

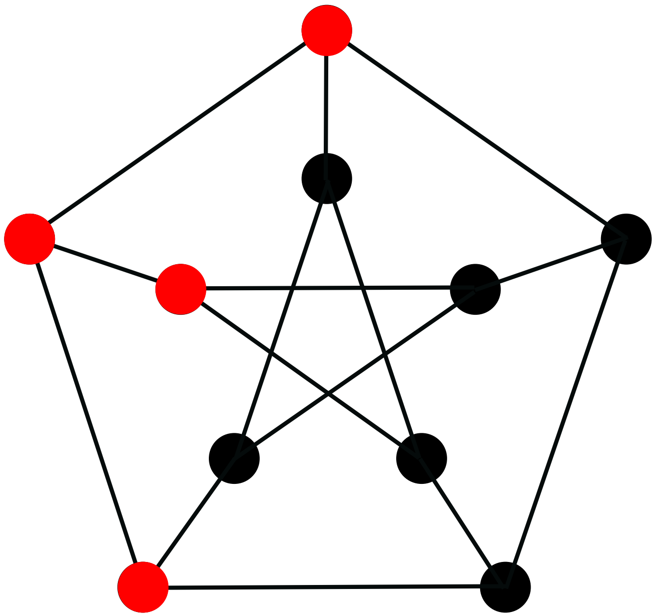
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Let's first play...

Take your favorite graph, e.g. Petersen graph.

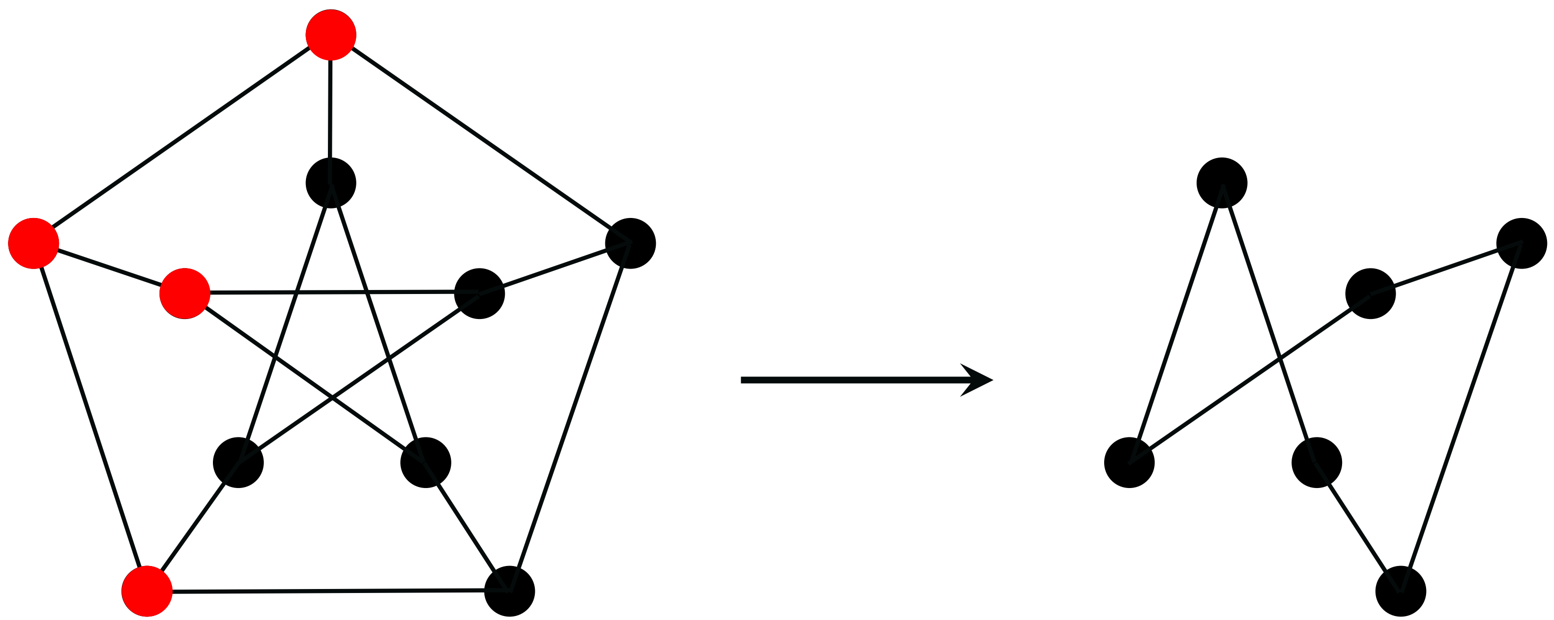
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Let's first play...

Take your favorite graph, e.g. Petersen graph.

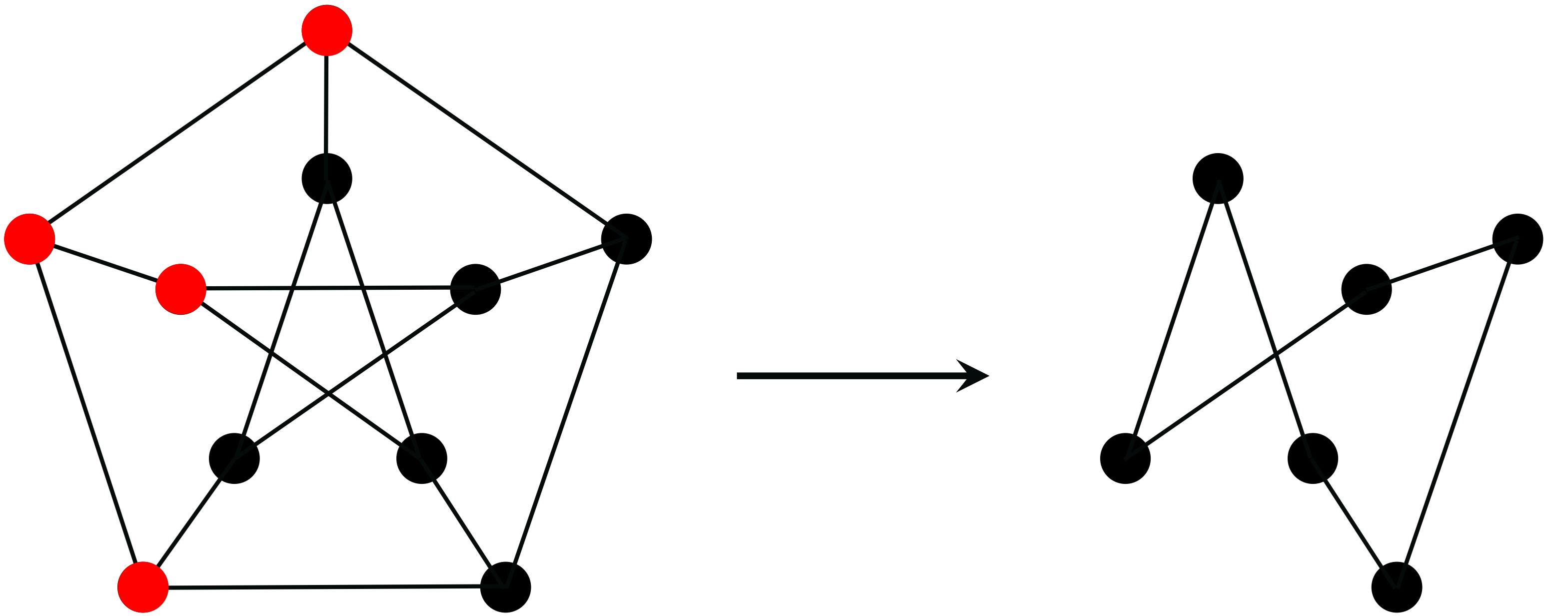
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Let's first play...

Take your favorite graph, e.g. Petersen graph.

On her turn, each player chooses a vertex and deletes its closed neighbourhood...

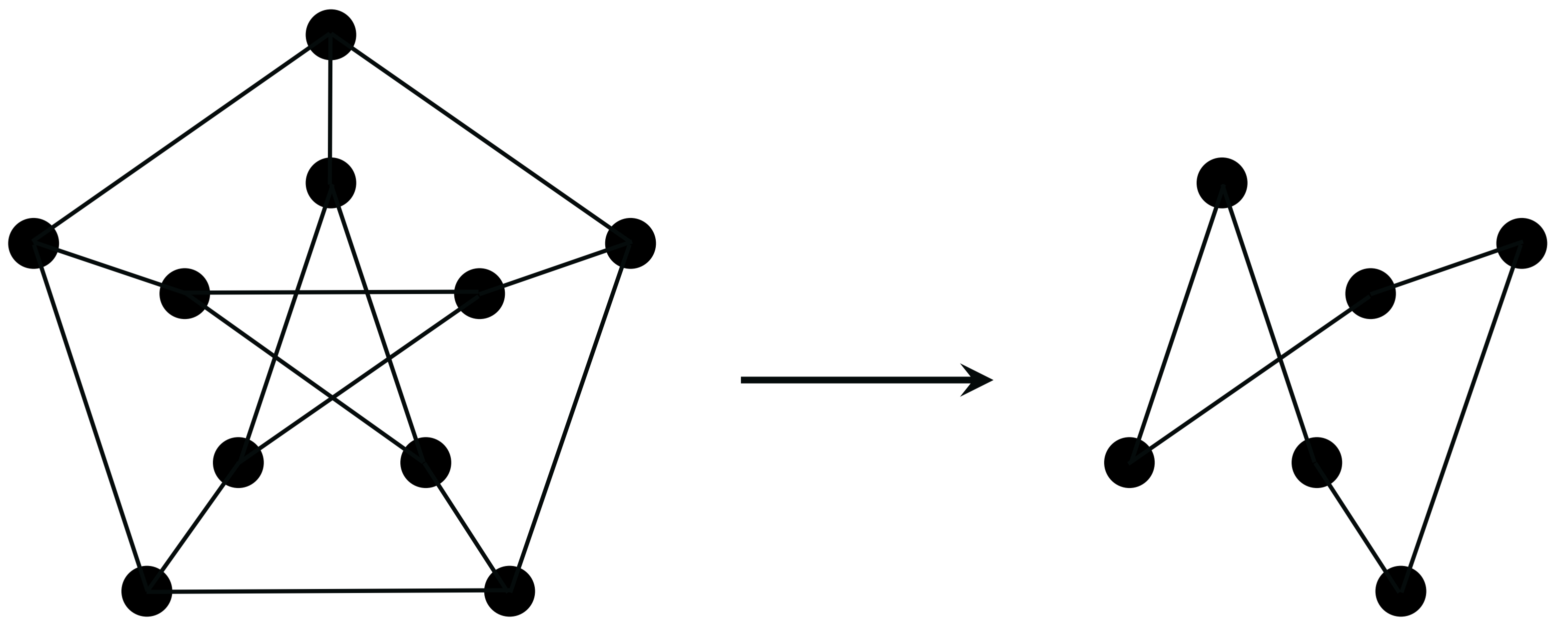


The first player unable to move (empty graph) looses the game...

Let’s first play...

Take your favorite graph, e.g. Petersen graph.

On her turn, each player chooses a vertex and deletes its closed neighbourhood...

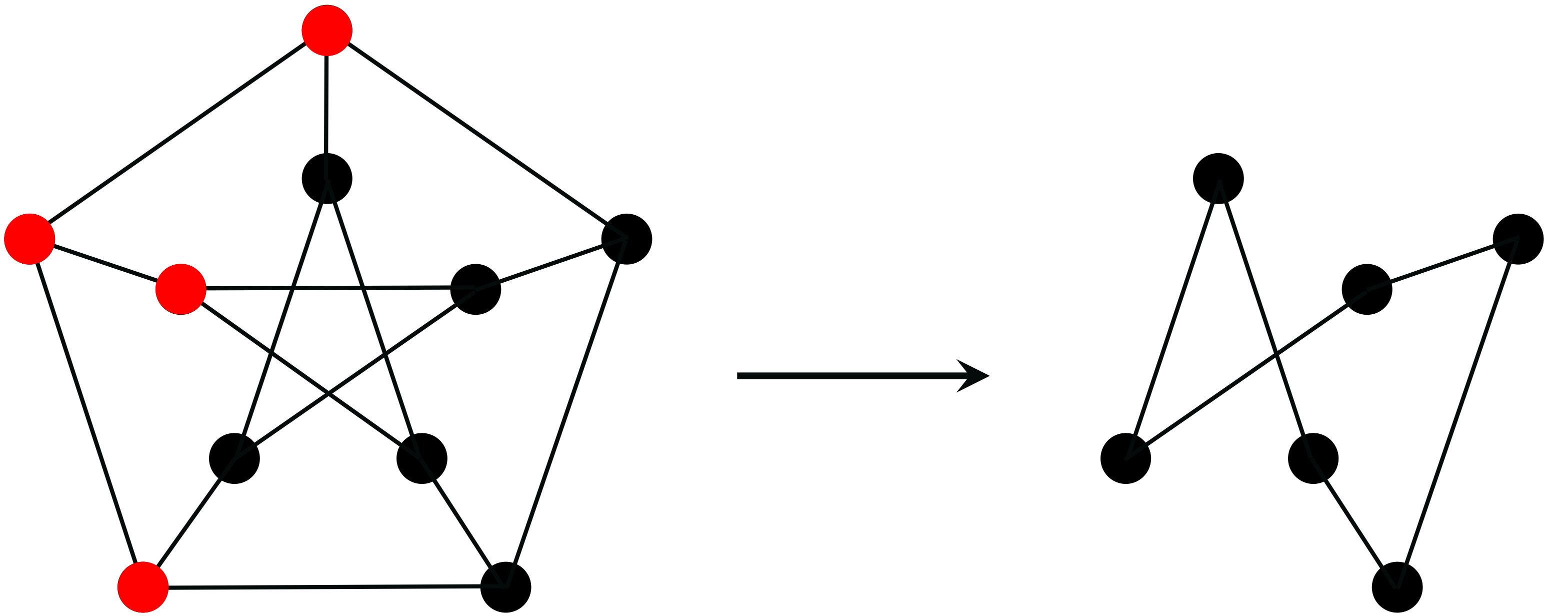


Would you prefer to be the first player? the second player?

Let’s first play...

Take your favorite graph, e.g. Petersen graph.

On her turn, each player chooses a vertex and deletes its closed neighbourhood...

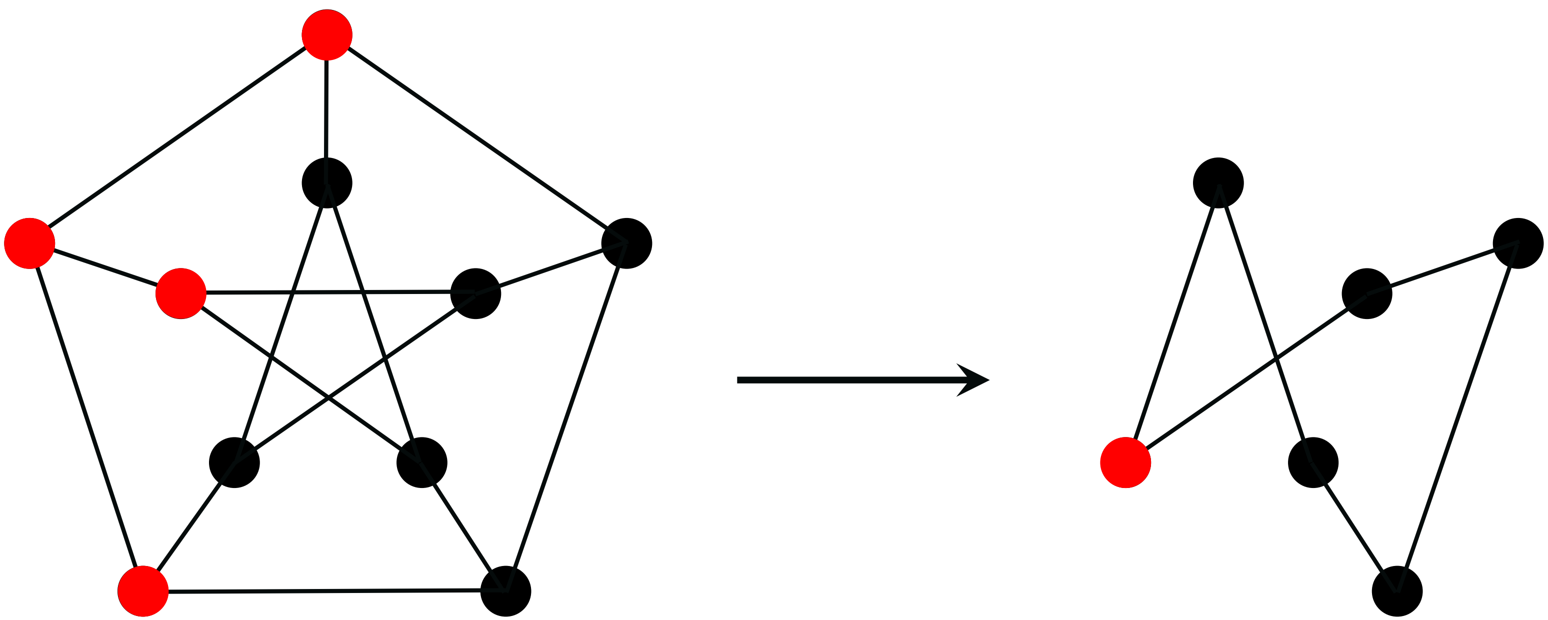


Would you prefer to be the first player? the second player?

Let's first play...

Take your favorite graph, e.g. Petersen graph.

On her turn, each player chooses a vertex and deletes its closed neighbourhood...

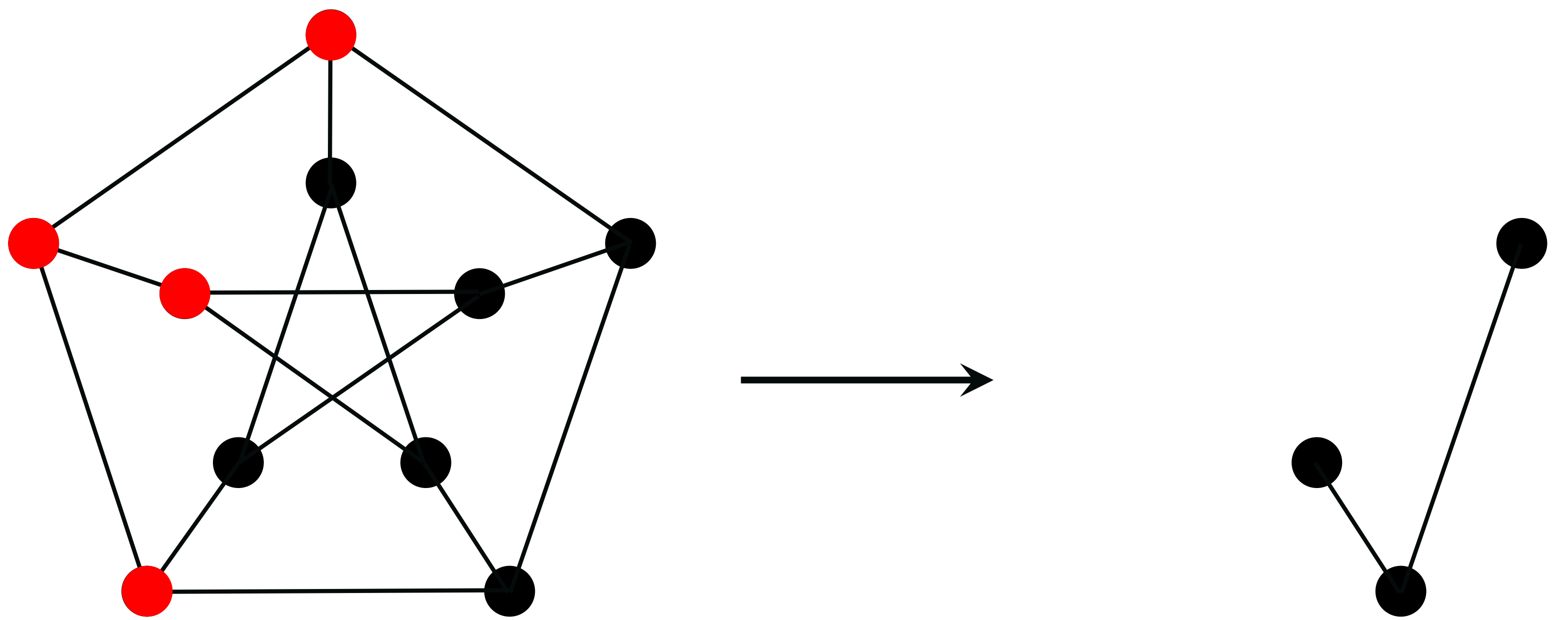


Would you prefer to be the first player? the second player?

Let's first play...

Take your favorite graph, e.g. Petersen graph.

On her turn, each player chooses a vertex and deletes its closed neighbourhood...

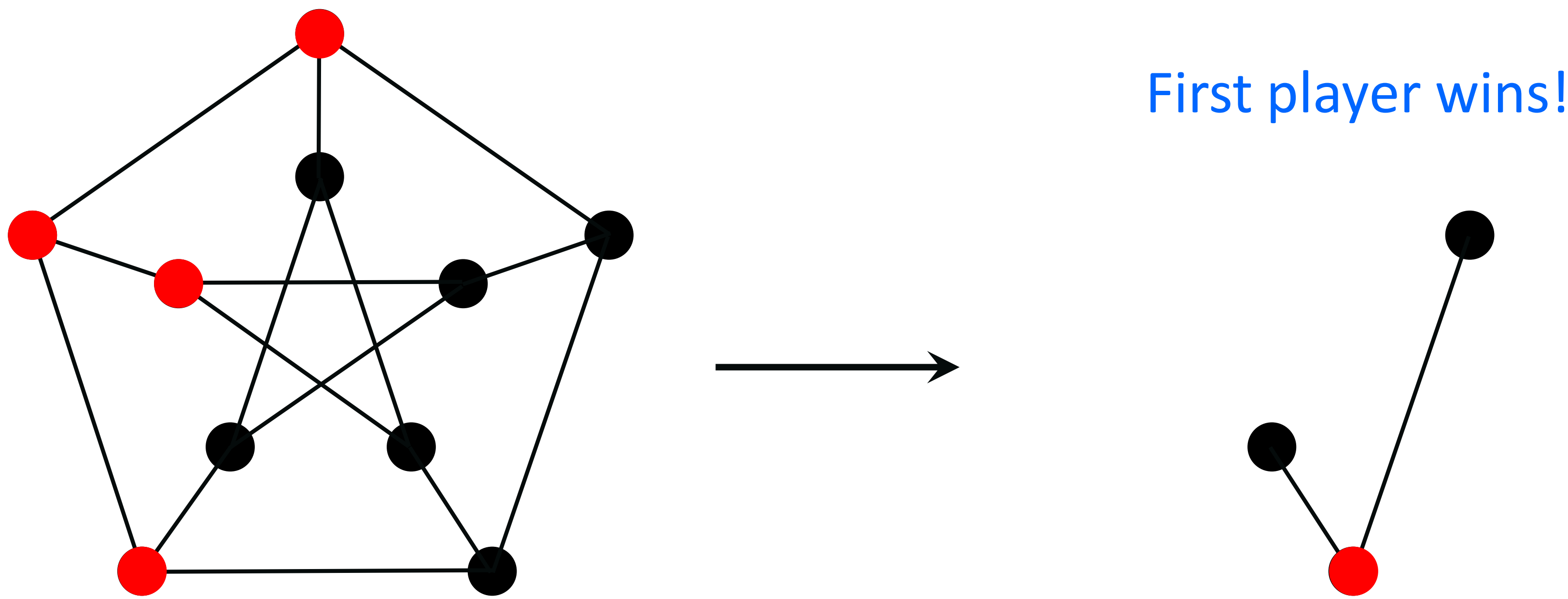


Would you prefer to be the first player? the second player?

Let’s first play...

Take your favorite graph, e.g. Petersen graph.

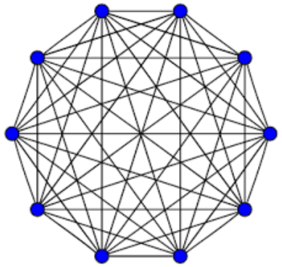
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Would you prefer to be the first player? the second player?

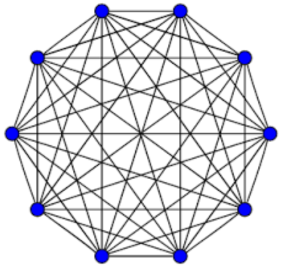
Let’s first play...

- Suppose now that the initial graph is the complete graph K_n on n vertices...



Would you prefer to be the first player? the second player?

Let's first play...

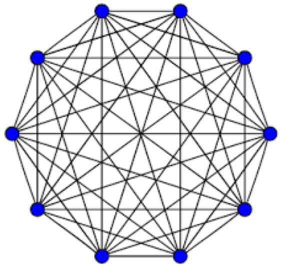


- Suppose now that the initial graph is the complete graph K_n on n vertices...

  Would you prefer to be the first player? the second player?

  Of course, the first player always wins...

Let's first play...



- Suppose now that the initial graph is the complete graph K_n on n vertices...

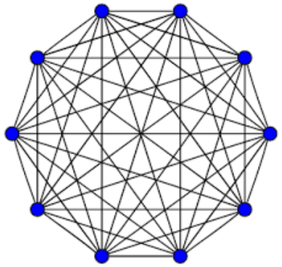
  Would you prefer to be the first player? the second player?

  Of course, the first player always wins...

- And if the initial graph is the path P_n on n vertices?

Let's first play...

- Suppose now that the initial graph is the complete graph K_n on n vertices...



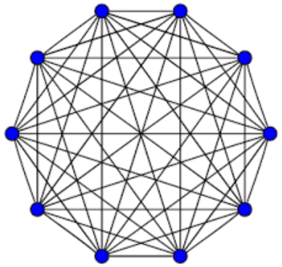
Would you prefer to be the first player? the second player?

Of course, the first player always wins...

- And if the initial graph is the path P_n on n vertices?

Would you prefer to be the first player? the second player?

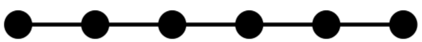
Let's first play...



- Suppose now that the initial graph is the complete graph K_n on n vertices...

  Would you prefer to be the first player? the second player?

  Of course, the first player always wins...

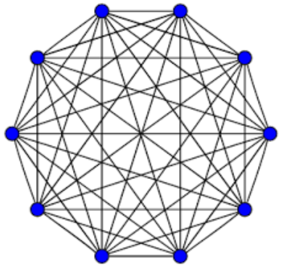


- And if the initial graph is the path P_n on n vertices?

  Would you prefer to be the first player? the second player?

  Hum hum... seems not so easy...

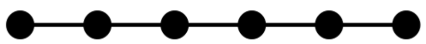
Let's first play...



- Suppose now that the initial graph is the complete graph K_n on n vertices...

  Would you prefer to be the first player? the second player?

  Of course, the first player always wins...



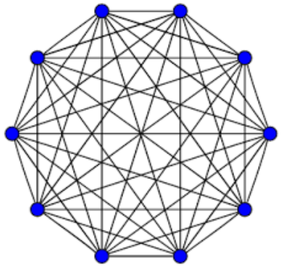
- And if the initial graph is the path P_n on n vertices?

  Would you prefer to be the first player? the second player?

  Hum hum... seems not so easy...

  Easy if n is odd:

Let's first play...



- Suppose now that the initial graph is the complete graph K_n on n vertices...

  Would you prefer to be the first player? the second player?

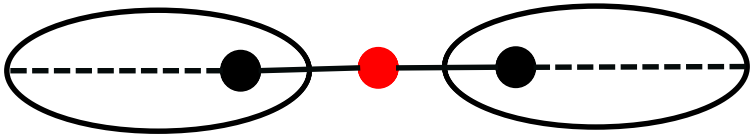
  Of course, the first player always wins...

- And if the initial graph is the path P_n on n vertices?

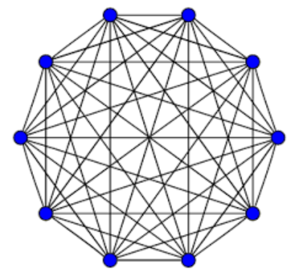
  Would you prefer to be the first player? the second player?

  Hum hum... seems not so easy...

  Easy if n is odd:



Let’s first play...



- Suppose now that the initial graph is the complete graph K_n on n vertices...

  Would you prefer to be the first player? the second player?

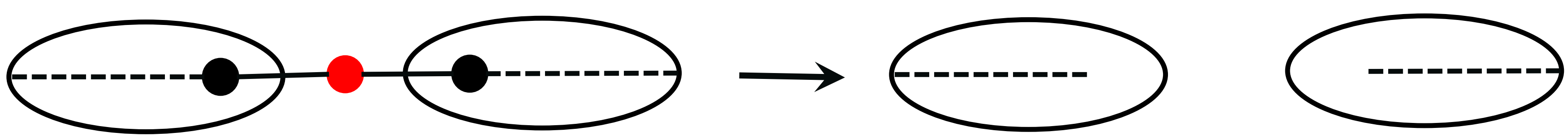
  Of course, the first player always wins...

- And if the initial graph is the path P_n on n vertices?

  Would you prefer to be the first player? the second player?

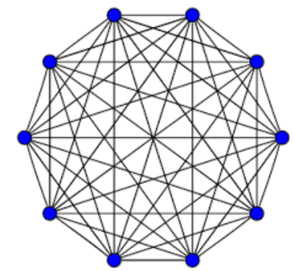
  Hum hum... seems not so easy...

  Easy if n is odd:



First player wins: mimicking strategy...

Let's first play...



- Suppose now that the initial graph is the complete graph K_n on n vertices...

  Would you prefer to be the first player? the second player?

  Of course, the first player always wins...

- And if the initial graph is the path P_n on n vertices?

  Would you prefer to be the first player? the second player?

  Hum hum... seems not so easy...

  In that case, the first player looses if and only if either

  - *$n = 4, 8, 14, 20, 24, 28, 34, 38, 42$, or*
  - *$n > 51$ and $n \equiv 4, 8, 20, 24, 28 \pmod{34}$.* [GUY, SMITH, 1956]

Let’s first play...

Let us now change the “winning rule” as follows: the first player unable to move wins the game...

Let’s first play...

Let us now change the “winning rule” as follows: the first player unable to move wins the game...

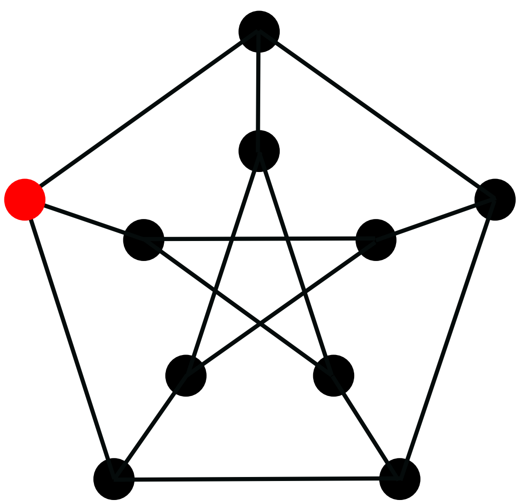
The case of K_n is again easy: *the first player always looses*...

Let’s first play...

Let us now change the “winning rule” as follows: the first player unable to move wins the game...

The case of K_n is again easy: *the first player always looses...*

The first player still wins the game on Petersen graph:

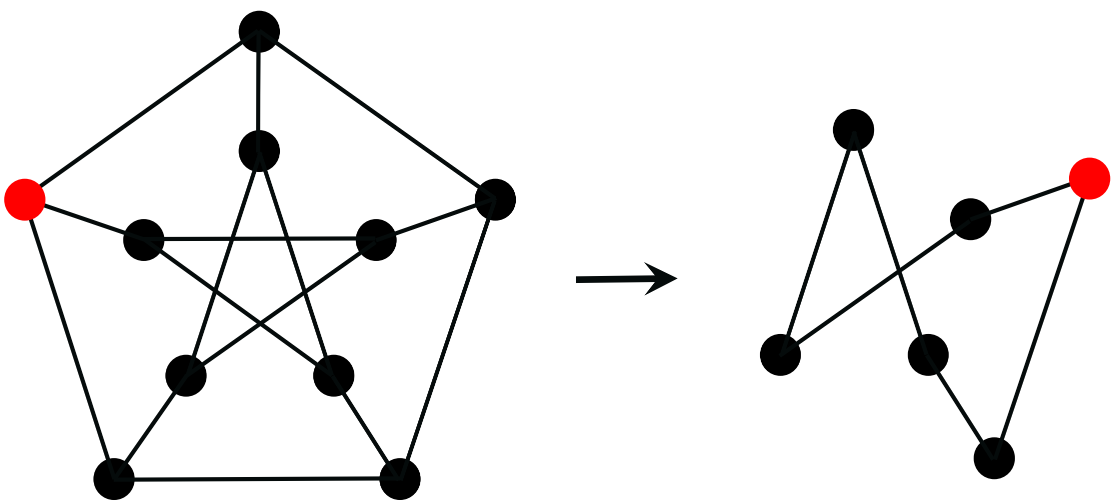


Let's first play...

Let us now change the "winning rule" as follows: the first player unable to move wins the game...

The case of K_n is again easy: *the first player always looses...*

The first player still wins the game on Petersen graph:

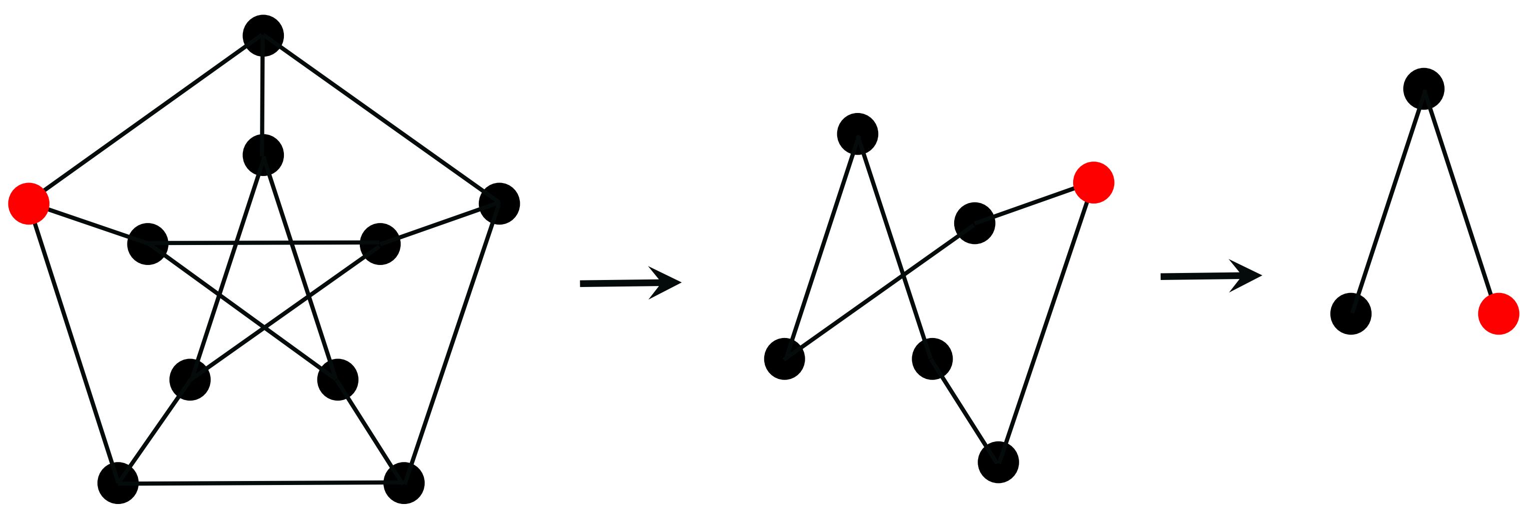


Let's first play...

Let us now change the "winning rule" as follows: the first player unable to move wins the game...

The case of K_n is again easy: *the first player always looses*...

The first player still wins the game on Petersen graph:

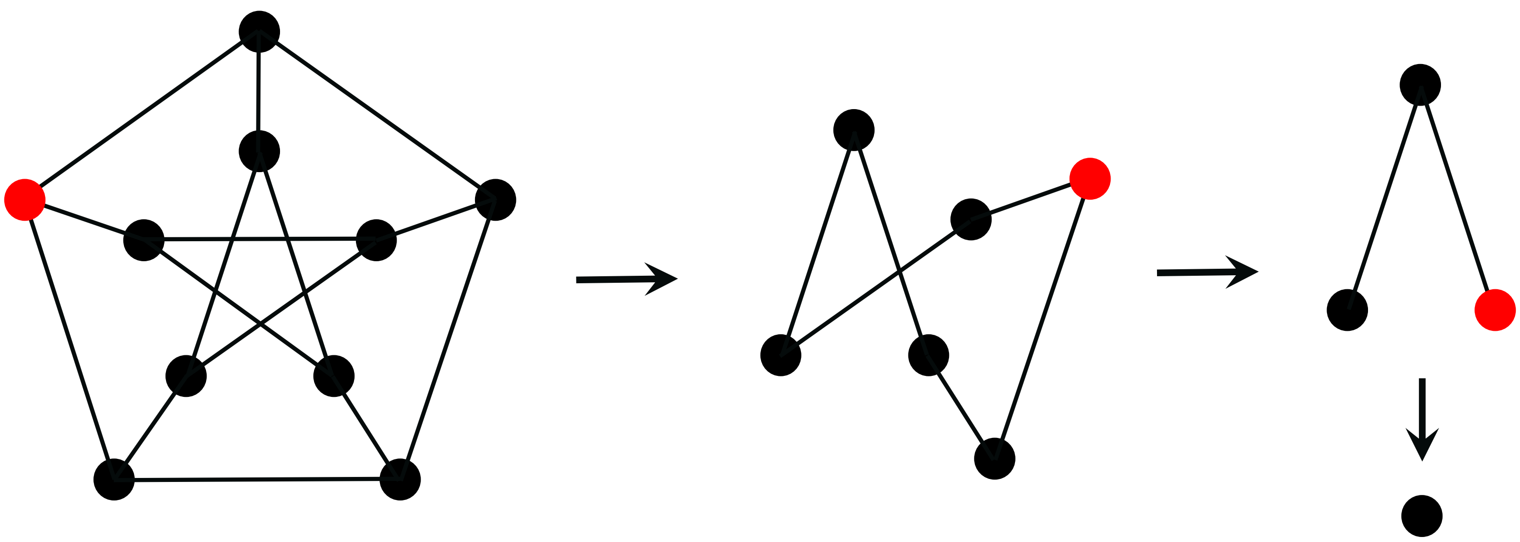


Let's first play...

Let us now change the "winning rule" as follows: the first player unable to move wins the game...

The case of K_n is again easy: *the first player always looses...*

The first player still wins the game on Petersen graph:



Let's first play...

Let us now change the "winning rule" as follows: the first player unable to move wins the game...

What about the game on the path P_n?

Let’s first play...

Let us now change the “winning rule” as follows: the first player unable to move wins the game...

What about the game on the path P_n?

Again not easy...

Let's first play...

Let us now change the "winning rule" as follows: the first player unable to move wins the game...

What about the game on the path P_n?

Again not easy...

Really not easy: a well-known open problem since 1935!...

Let's first play...

Let us now change the "winning rule" as follows: the first player unable to move wins the game...

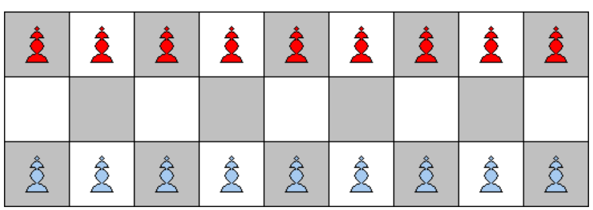
What about the game on the path P_n?

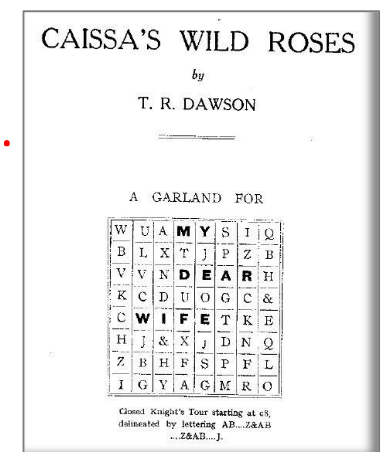
Again not easy...

Really not easy: a well-known open problem since 1935!...

This game is known as DAWSON'S CHESS game.

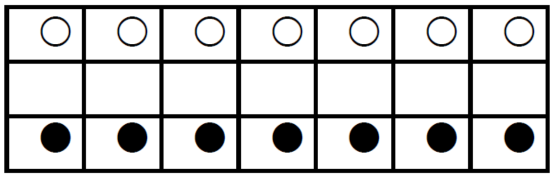
T. R. DAWSON. *Caissa's Wild Roses*. Problem #80 (1935).





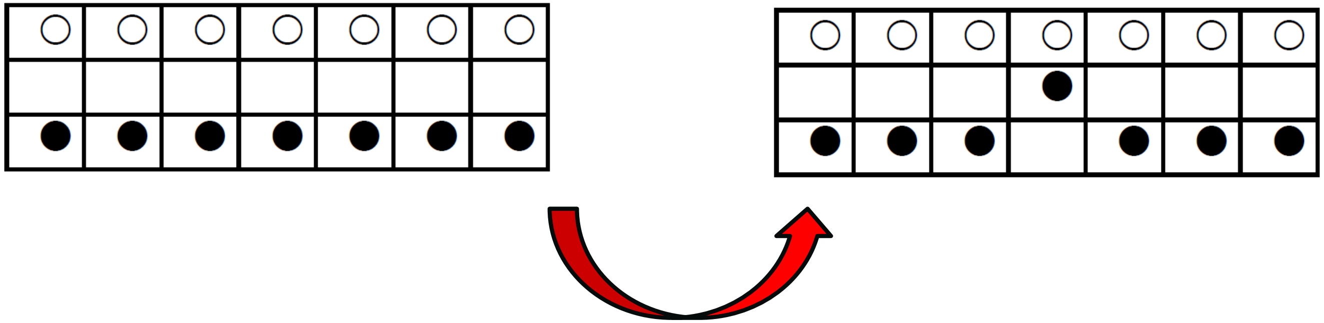
Let’s first play...

DAWSON’S CHESS (*Two rows of pawns, capturing is mandatory...*)



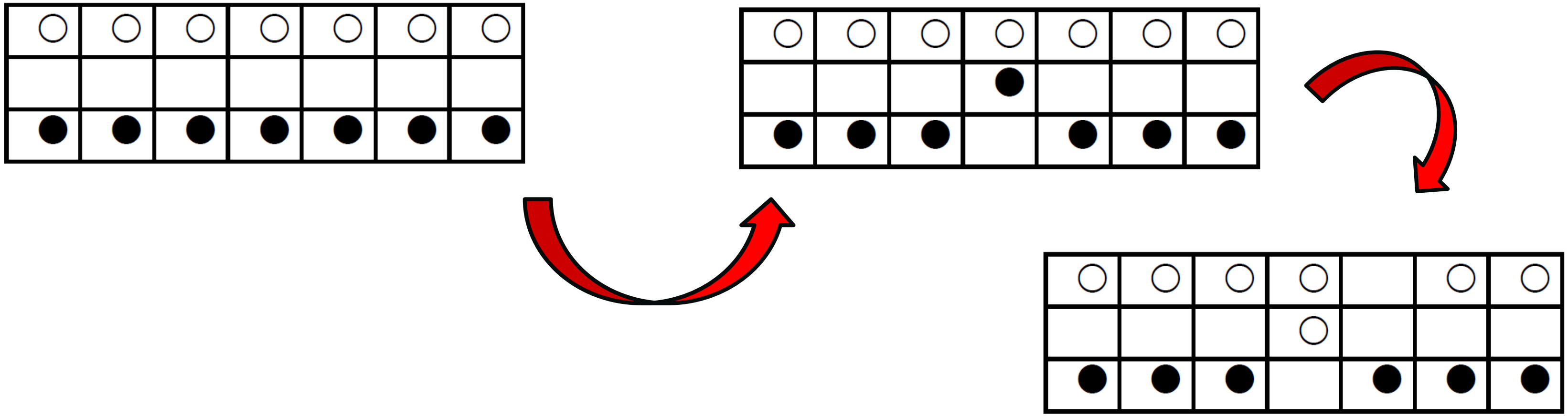
Let's first play...

Dawson's Chess (*Two rows of pawns, capturing is mandatory...*)



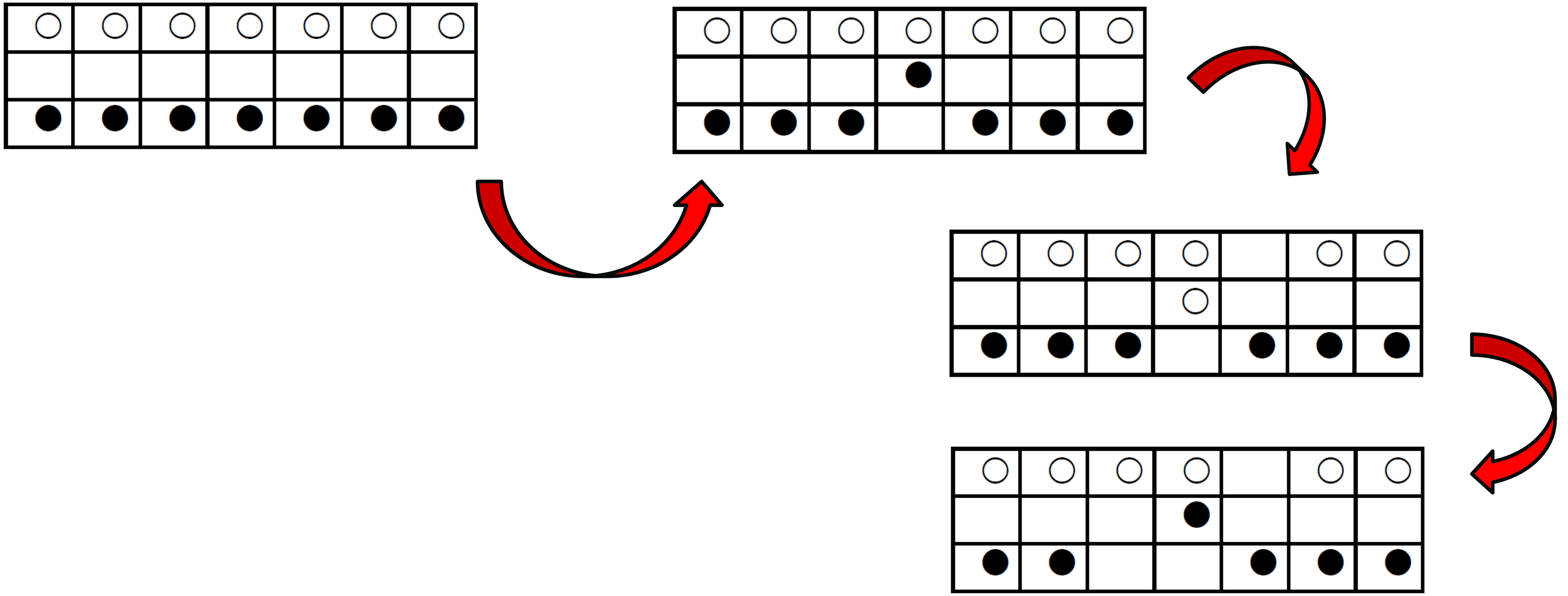
Let's first play...

DAWSON'S CHESS (*Two rows of pawns, capturing is mandatory...*)



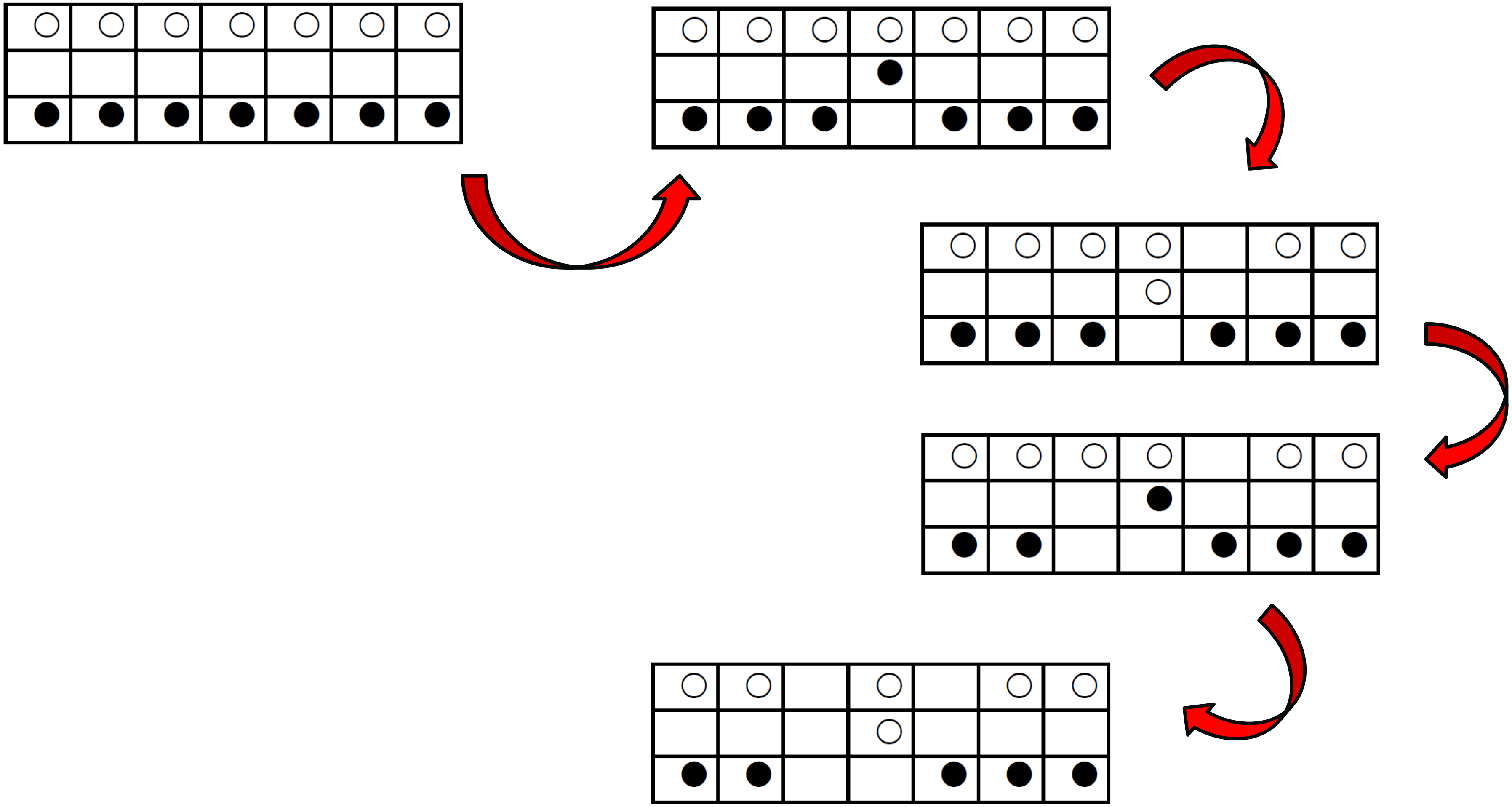
Let's first play...

DAWSON'S CHESS (*Two rows of pawns, capturing is mandatory...*)



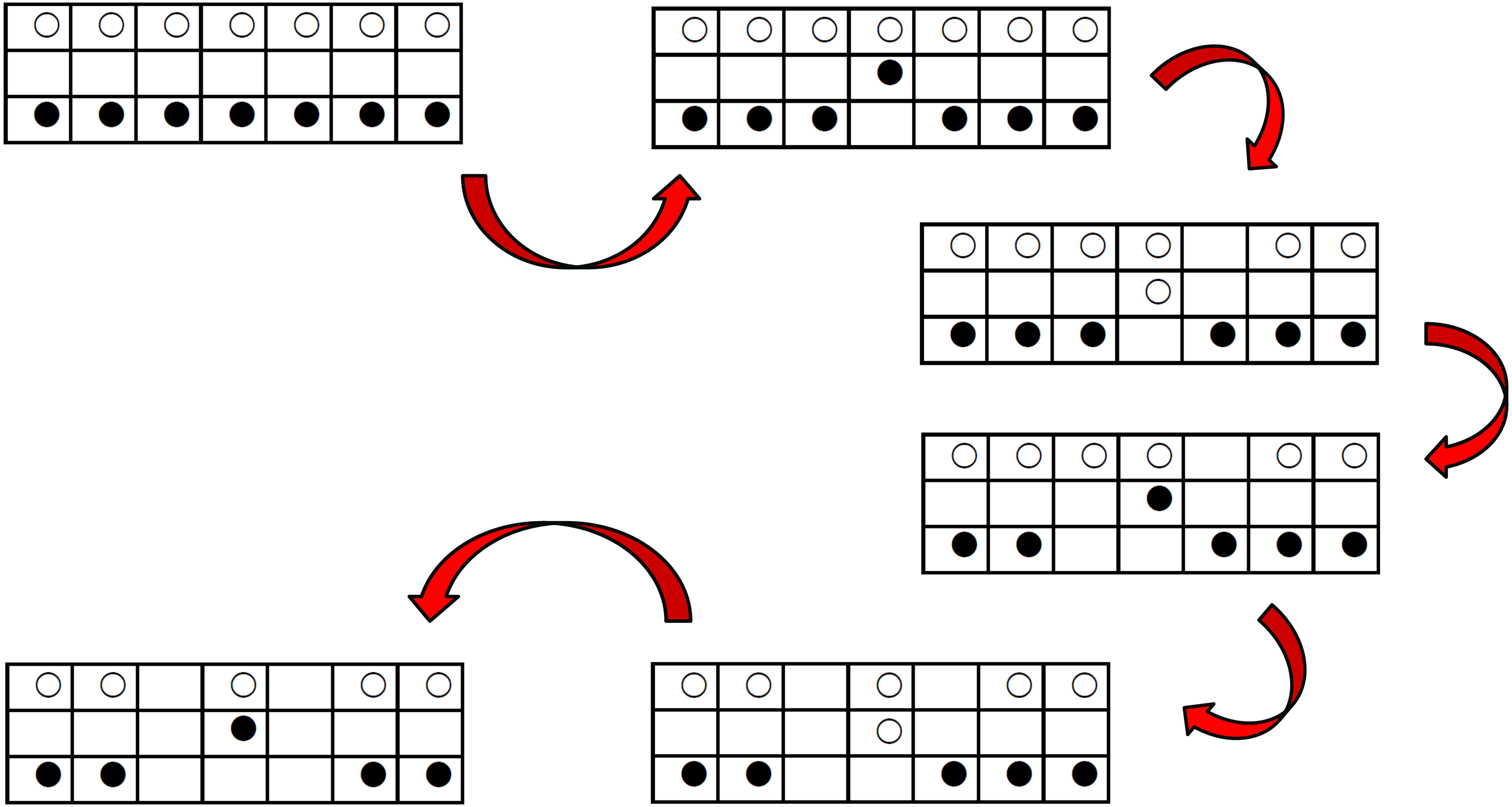
Let's first play...

DAWSON'S CHESS (*Two rows of pawns, capturing is mandatory...*)



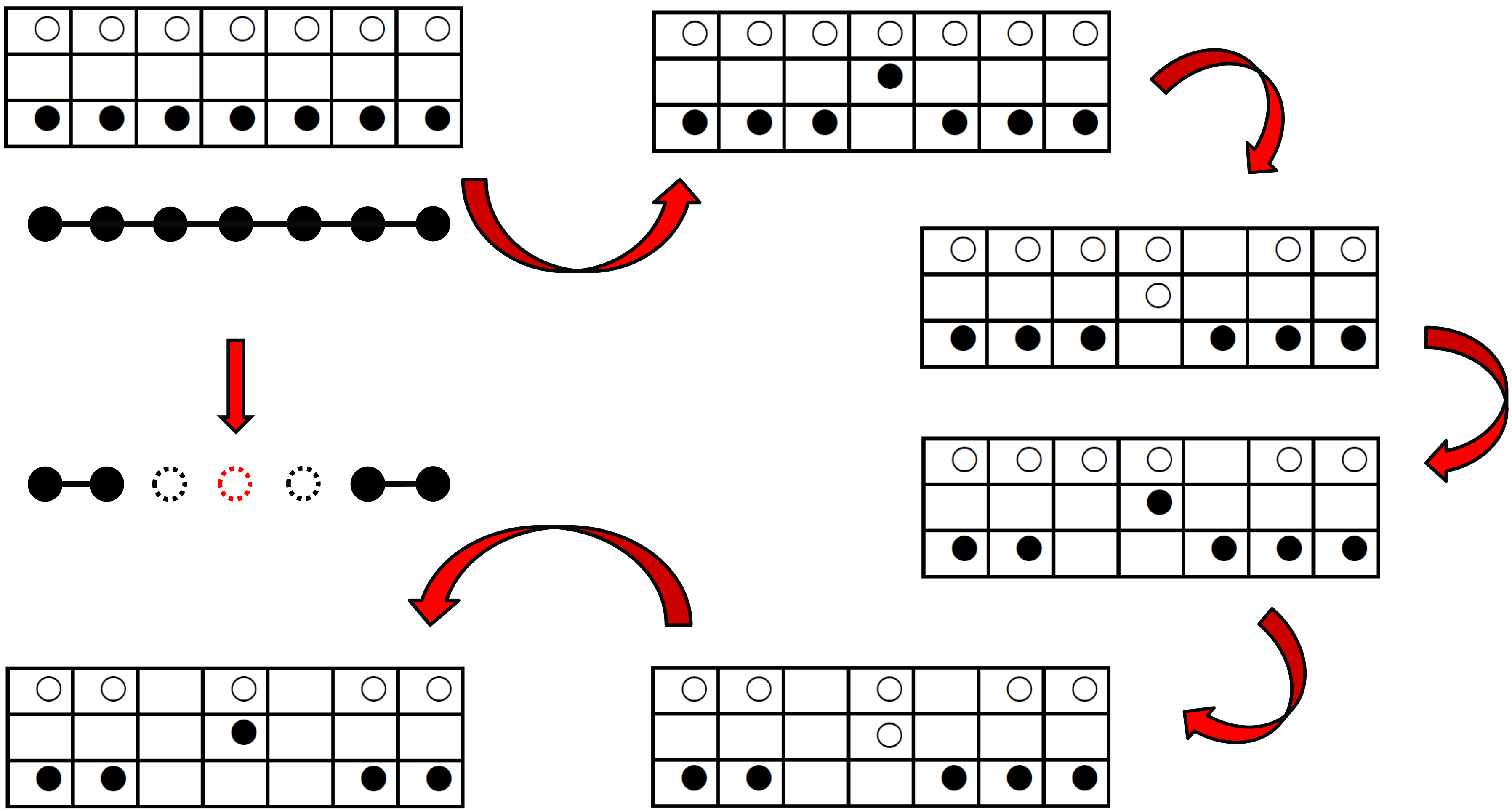
Let's first play...

DAWSON'S CHESS (*Two rows of pawns, capturing is mandatory...*)



Let's first play...

DAWSON'S CHESS (*Two rows of pawns, capturing is mandatory...*)



Outline

Outline

Starters

A flavour of
Combinatorial
Game Theory

Impartial games – Sums of games – Sprague-Grundy value – Game-graph...

Outline

Starters

A flavour of Combinatorial Game Theory

Impartial games – Sums of games – Sprague-Grundy value – Game-graph...

Discovery menu

Geography – Nim on graphs

Node-kayles

Proper k-colouring

0.33 – Timber!

Outline

Starters

A flavour of
Combinatorial
Game Theory

Impartial games – Sums of games – Sprague-Grundy value – Game-graph...

Tea Break

Discovery menu

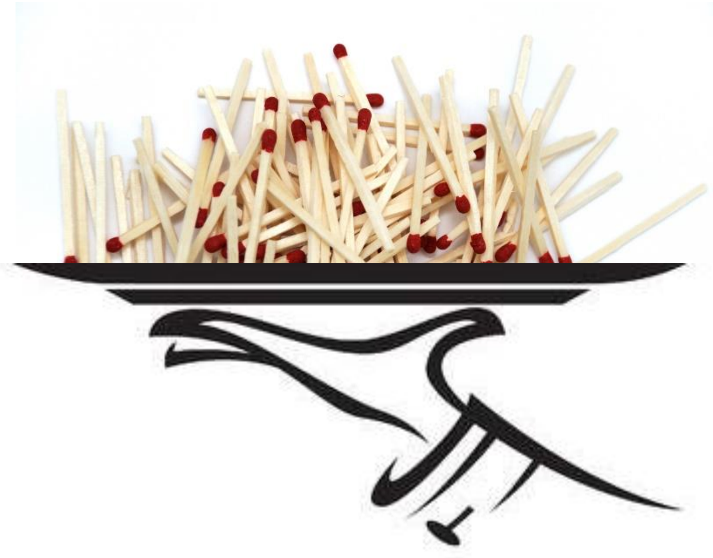
Geography – Nim on graphs

Node-kayles

Proper k-colouring

0.33 – Timber!

A flavour of Combinatorial Game Theory



Combinatorial games (1)

Combinatorial game

A combinatorial game is a 2-player game such that:

Combinatorial games (1)

Combinatorial game

A combinatorial game is a 2-player game such that:

- players alternate in turn,
- there is no hidden information and no chance elements,
- the number of positions (configurations) is finite,
- no position can be encountered twice during a game (the game is thus finite).

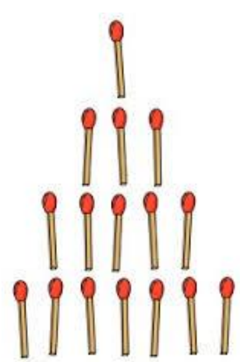
Combinatorial games (1)

Combinatorial game

A combinatorial game is a 2-player game such that:

- players alternate in turn,
- there is no hidden information and no chance elements,
- the number of positions (configurations) is finite,
- no position can be encountered twice during a game (the game is thus finite).

The game of **Nim**

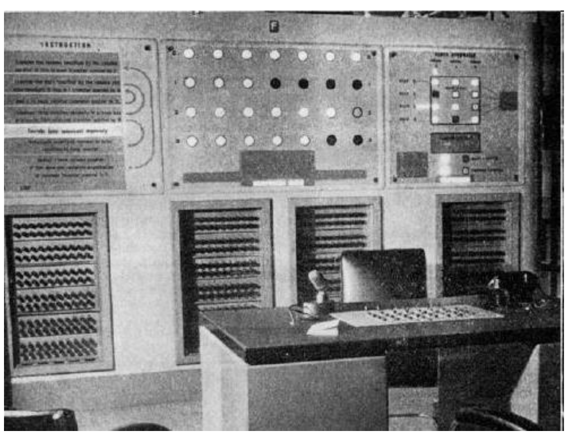


Combinatorial games (1)

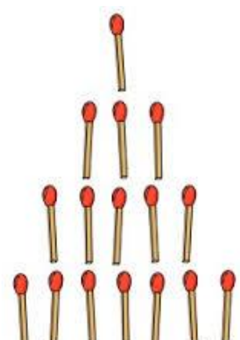
Combinatorial game

A combinatorial game is a 2-player game such that:

- players alternate in turn,
- there is no hidden information and no chance elements,
- the number of positions (configurations) is finite,
- no position can be encountered twice during a game (the game is thus finite).



The game of **NIM**



Nimrod
1951

Combinatorial games (2)

Winning rule

- **Normal play**
  The first player unable to move looses the game.

- **Misère play**
  The first player unable to move wins the game.

Combinatorial games (2)

Winning rule

- **Normal play**
  The first player unable to move looses the game.

- **Misère play**
  The first player unable to move wins the game.

The normal version is usually “easier” to deal with...

Combinatorial games (3)

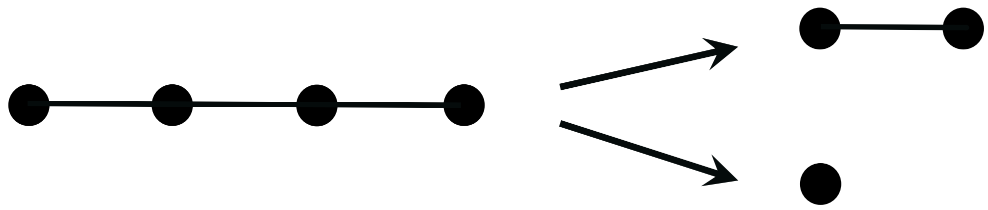
Rules and options

The set of rules of the game gives, for each position and each player, the options of this position.

Combinatorial games (3)

Rules and options

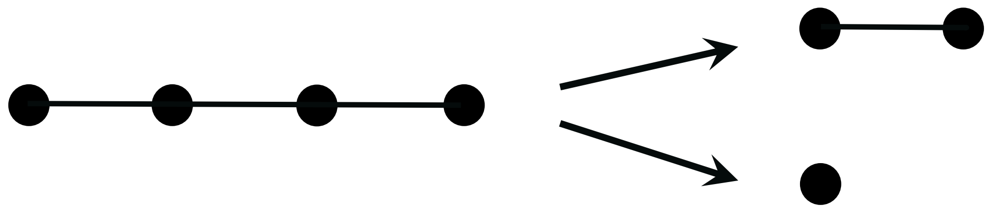
The set of rules of the game gives, for each position and each player, the options of this position.



Combinatorial games (3)

Rules and options

The set of rules of the game gives, for each position and each player, the options of this position.



Impartial vs partisan combinatorial games

The game is impartial if both players have the same options for every position, it is partisan otherwise.

Combinatorial game theory

Since the mathematical solution of the game of **NIM** by C.L. BOUTON (1901), the theory of combinatorial games has been increasingly developed.

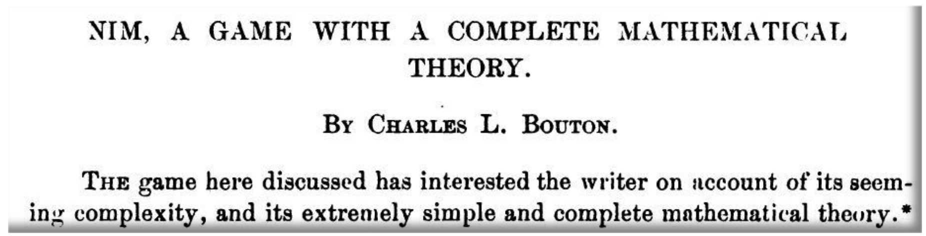
NIM, A GAME WITH A COMPLETE MATHEMATICAL THEORY.

BY CHARLES L. BOUTON.

THE game here discussed has interested the writer on account of its seeming complexity, and its extremely simple and complete mathematical theory.*

Combinatorial game theory

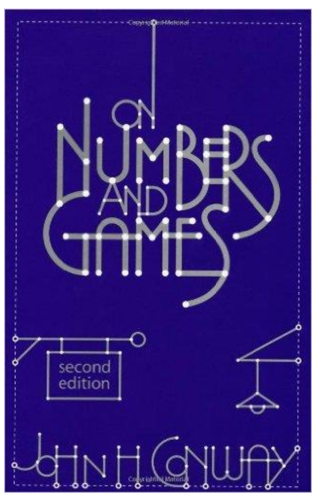
Since the mathematical solution of the game of **NIM** by C.L. BOUTON (1901), the theory of combinatorial games has been increasingly developed.



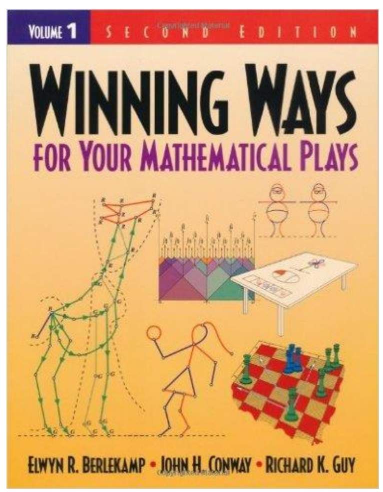
NIM, A GAME WITH A COMPLETE MATHEMATICAL THEORY.

By CHARLES L. BOUTON.

THE game here discussed has interested the writer on account of its seeming complexity, and its extremely simple and complete mathematical theory.*



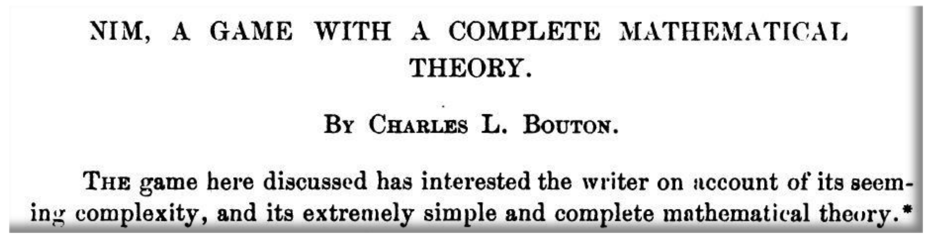
JOHN H. CONWAY (1976)



ELVIN R. BERLEKAMP
JOHN H. CONWAY
RICHARD K. GUY (1982)

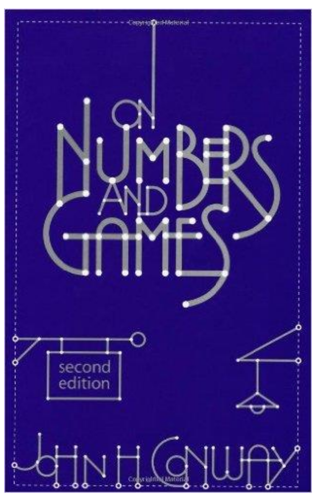
Combinatorial game theory

Since the mathematical solution of the game of **NIM** by C.L. BOUTON (1901), the theory of combinatorial games has been increasingly developed.

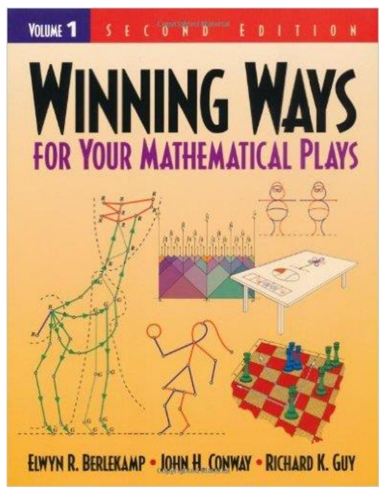

NIM, A GAME WITH A COMPLETE MATHEMATICAL THEORY.

BY CHARLES L. BOUTON.

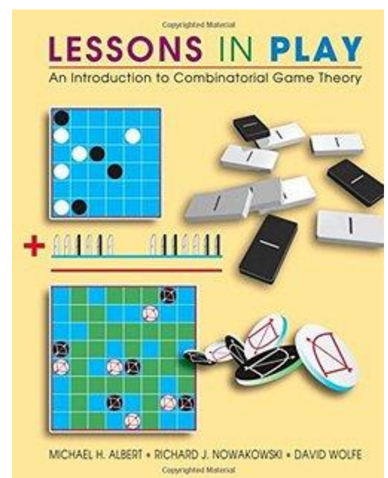
THE game here discussed has interested the writer on account of its seeming complexity, and its extremely simple and complete mathematical theory.*



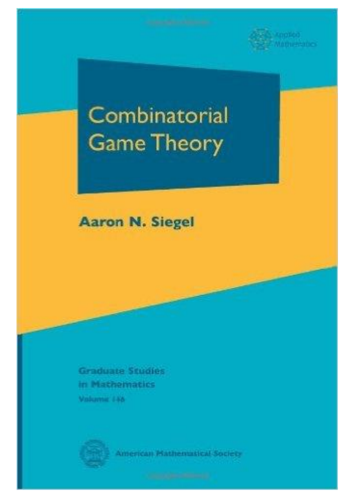
JOHN H. CONWAY
(1976)



ELVIN R. BERLEKAMP
JOHN H. CONWAY
RICHARD K. GUY (1982)



MICHAEL H. ALBERT
RICHARD J. NOWAKOWSKI
DAVID WOLFE (2007)



AARON N. SIEGEL
(2013)

Outcomes

Outcomes

The Fundamental Theorem

If G is an impartial game then either the first or the second player can force a win.

Outcomes

Outcomes

The Fundamental Theorem

If G is an impartial game then either the first or the second player can force a win.

Therefore, every position of an impartial combinatorial game is either a winning position (1^{st}-player wins), or a losing position (2^{nd}-player wins).

Outcomes

Outcomes

The Fundamental Theorem

If G is an impartial game then either the first or the second player can force a win.

Therefore, every position of an impartial combinatorial game is either a winning position (1^{st}-player wins), or a losing position (2^{nd}-player wins).

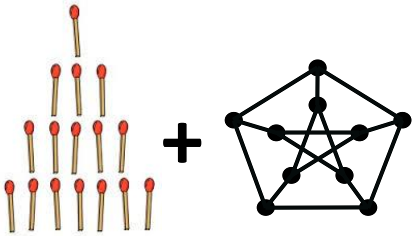
Observe that

- G is a winning position iff G has *at least one losing option*,
- G is a losing position iff either G is *empty*, or G has *only winning options*.

Sum of games (1)

Sum of games

Let G1 and G2 be two games. The (disjunctive) sum of G1 and G2 is the game **G1 + G2**, played as follows:

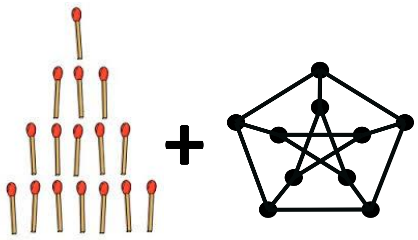


Sum of games (1)

Sum of games

Let G1 and G2 be two games. The (disjunctive) sum of G1 and G2 is the game **G1 + G2**, played as follows:

- on her turn, each player chooses the current position in G1 or in G2, and then moves according to the rules of G1 or G2, respectively,

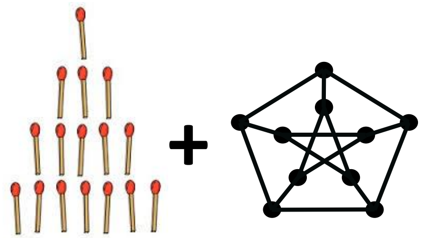


Sum of games (1)

Sum of games

Let G1 and G2 be two games. The (disjunctive) sum of G1 and G2 is the game **G1 + G2**, played as follows:

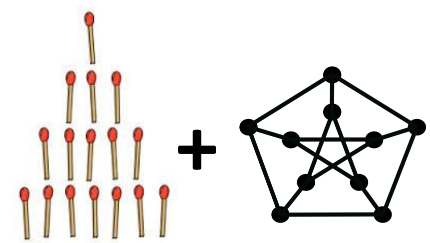
- on her turn, each player chooses the current position in G1 or in G2, and then moves according to the rules of G1 or G2, respectively,
- the game ends as soon as a player has no move in any of the two games.



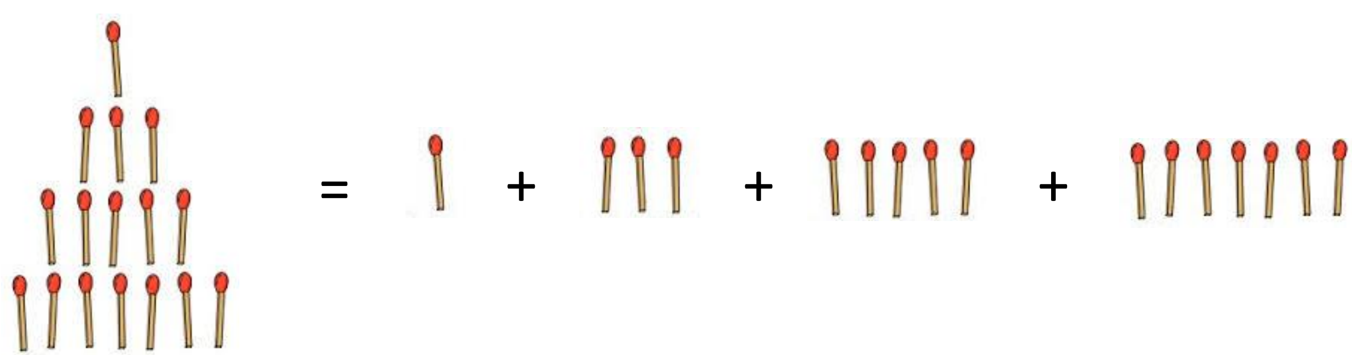
Sum of games

Let G1 and G2 be two games. The (disjunctive) sum of G1 and G2 is the game **G1 + G2**, played as follows:

- on her turn, each player chooses the current position in G1 or in G2, and then moves according to the rules of G1 or G2, respectively,



- the game ends as soon as a player has no move in any of the two games.

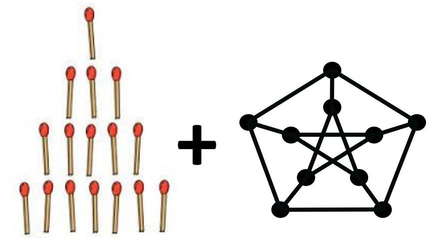


Sum of games (1)

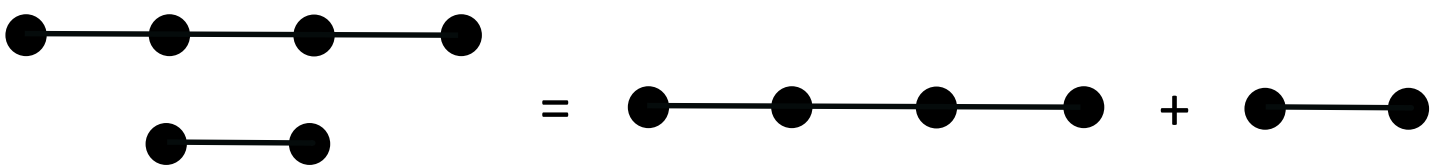
Sum of games

Let G1 and G2 be two games. The (disjunctive) sum of G1 and G2 is the game **G1 + G2**, played as follows:

- on her turn, each player chooses the current position in G1 or in G2, and then moves according to the rules of G1 or G2, respectively,



- the game ends as soon as a player has no move in any of the two games.



Sum of games (2)

Outcome of the sum of two games (normal play)

Knowing the outcome of both games G1 and G2 does not suffice for determining the outcome of G1 + G2...

Sum of games (2)

Outcome of the sum of two games (normal play)

Knowing the outcome of both games G1 and G2 does not suffice for determining the outcome of G1 + G2...

Outcome of G1 + G2		
G1 \ G2	**winning**	**losing**
winning	**????**	**winning**
losing	**winning**	**losing**

Sprague-Grundy function (1)

The Sprague-Grundy function (impartial games, normal)

Sprague-Grundy function (1)

The Sprague-Grundy function (impartial games, normal)

Theorem [R.P. SPRAGUE, 1935 – P.M. GRUNDY, 1939]

Every game G is "equivalent" to the game of NIM on a heap of n tokens (or a row of n matches) for some positive integer n.

Sprague-Grundy function (1)

The Sprague-Grundy function (impartial games, normal)

Theorem [R.P. SPRAGUE, 1935 – P.M. GRUNDY, 1939]

Every game G is "equivalent" to the game of NIM on a heap of n tokens (or a row of n matches) for some positive integer n.

Two games G and H are equivalent whenever we can replace any occurrence of G by H in any sum of games, without changing the outcome of the sum (in particular, G and H have the same outcome)...

Sprague-Grundy function (1)

The Sprague-Grundy function (impartial games, normal)

Theorem [R.P. Sprague, 1935 – P.M. Grundy, 1939]
Every game G is "equivalent" to the game of Nim on a heap of n tokens (or a row of n matches) for some positive integer n.

Two games G and H are equivalent whenever we can replace any occurrence of G by H in any sum of games, without changing the outcome of the sum (in particular, G and H have the same outcome)...

We then set $\sigma(G) = n$ (n is the Sprague-Grundy value of G).

Sprague-Grundy function (1)

The Sprague-Grundy function (impartial games, normal)

Theorem [R.P. SPRAGUE, 1935 – P.M. GRUNDY, 1939]

Every game G is "equivalent" to the game of NIM on a heap of n tokens (or a row of n matches) for some positive integer n.

Two games G and H are equivalent whenever we can replace any occurrence of G by H in any sum of games, without changing the outcome of the sum (in particular, G and H have the same outcome)...

We then set $\sigma(G) = n$ (n is the Sprague-Grundy value of G).

Therefore, a game G is a 2nd-player win if and only if $\sigma(G) = 0$.

(*Every heap with n > 0 tokens is a winning position*.)

Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, \ldots, G_k\}$, then

$$\sigma(G) = \text{mex}\ (\sigma(G_1), \ldots, \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\emptyset$) = **0**).

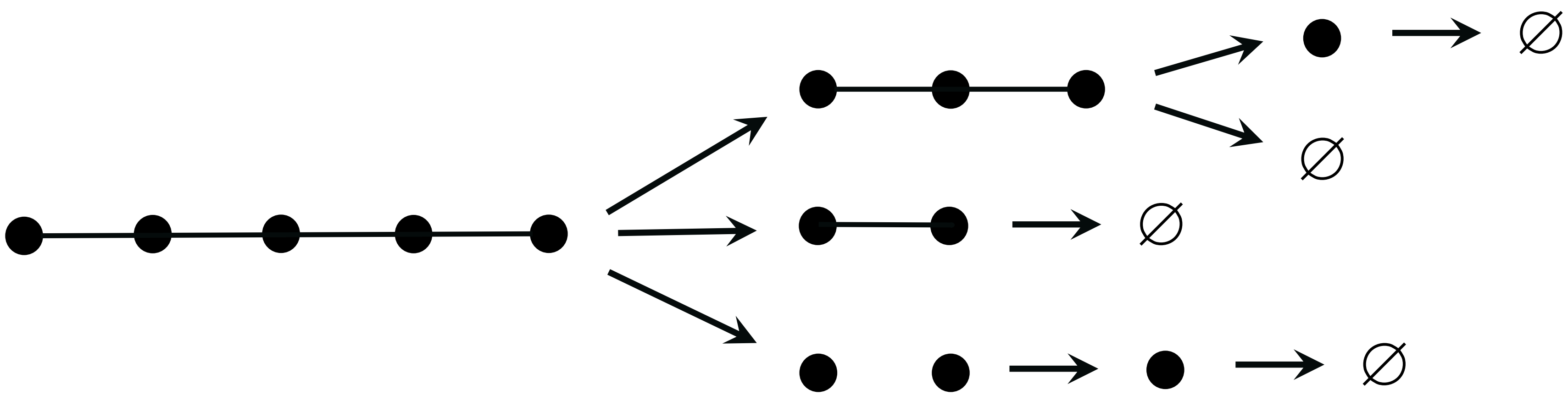
Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, \ldots, G_k\}$, then

$$\sigma(G) = \text{mex}\,(\sigma(G_1), \ldots, \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).



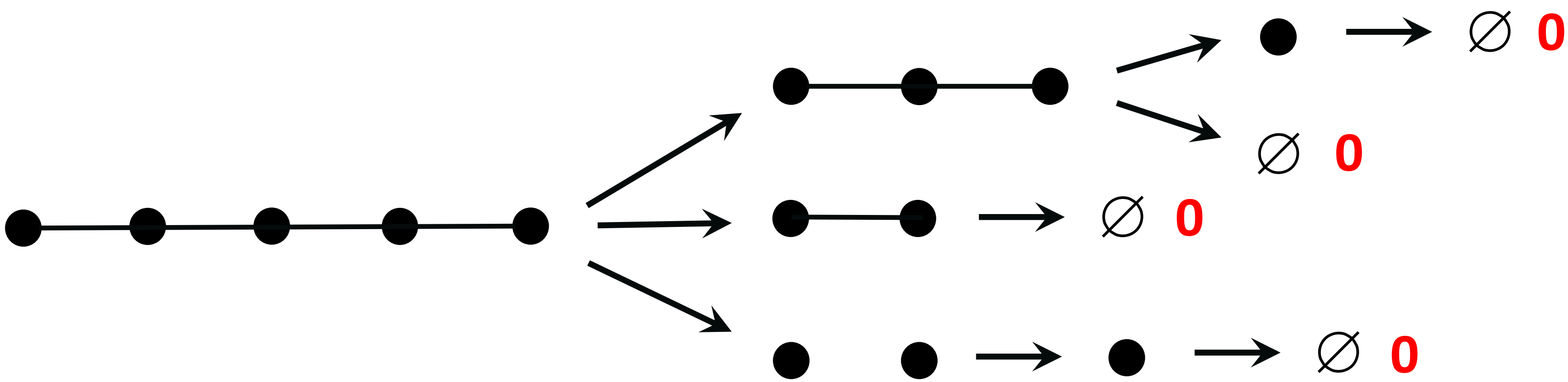
Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, \ldots, G_k\}$, then

$$\sigma(G) = \text{mex}\,(\sigma(G_1), \ldots, \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).



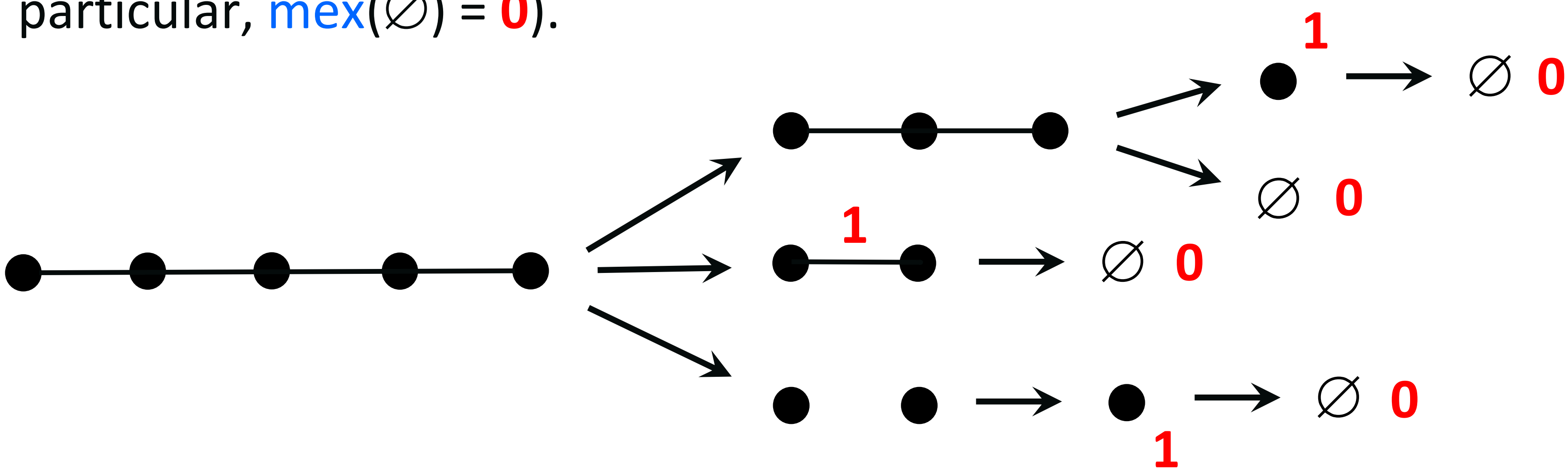
Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, \ldots, G_k\}$, then

$$\sigma(G) = \text{mex}\ (\sigma(G_1), \ldots, \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).



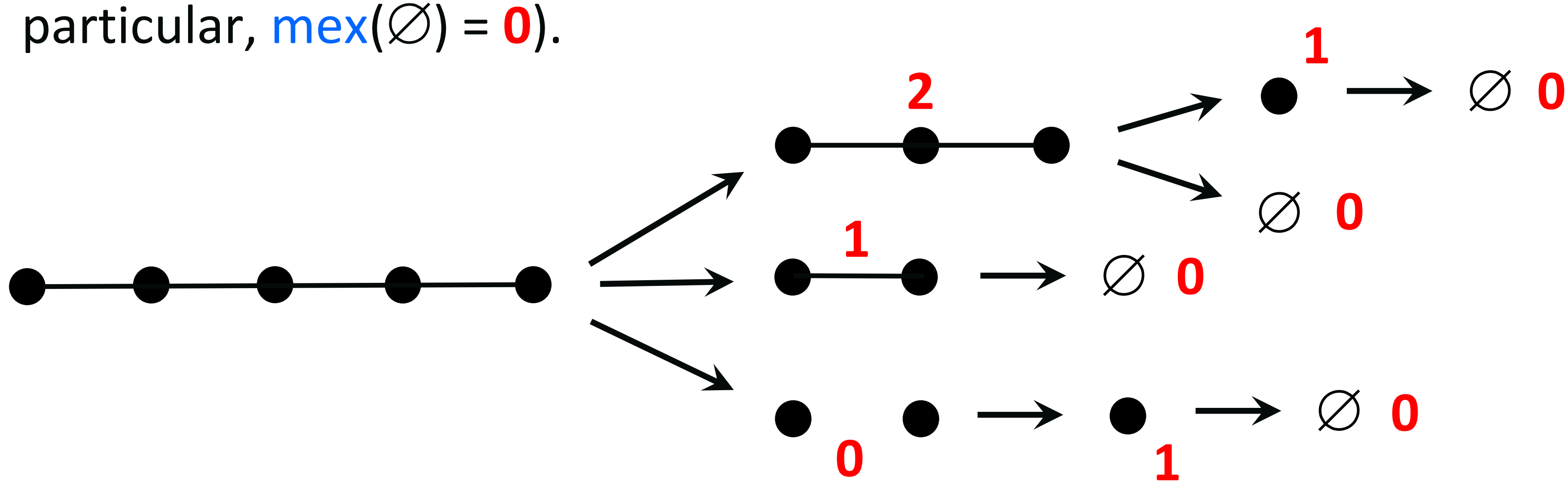
Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, ..., G_k\}$, then

$$\sigma(G) = \text{mex}\,(\sigma(G_1), ..., \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).



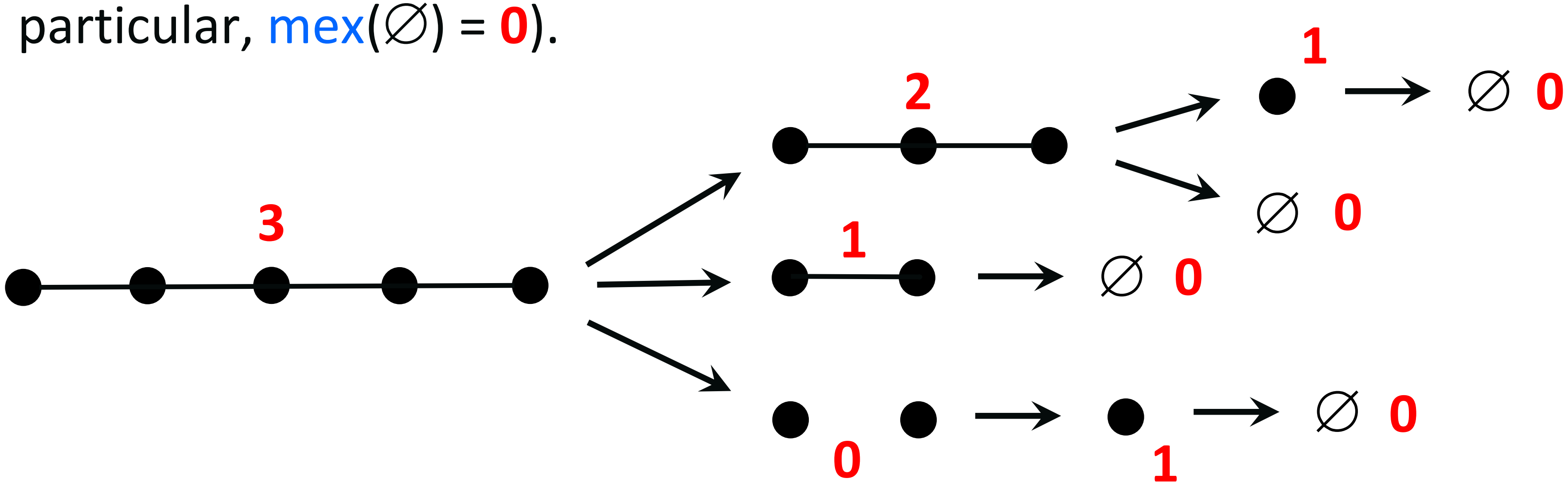
Sprague-Grundy function (2)

Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, ..., G_k\}$, then

$$\sigma(G) = \text{mex}\,(\sigma(G_1), ..., \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).

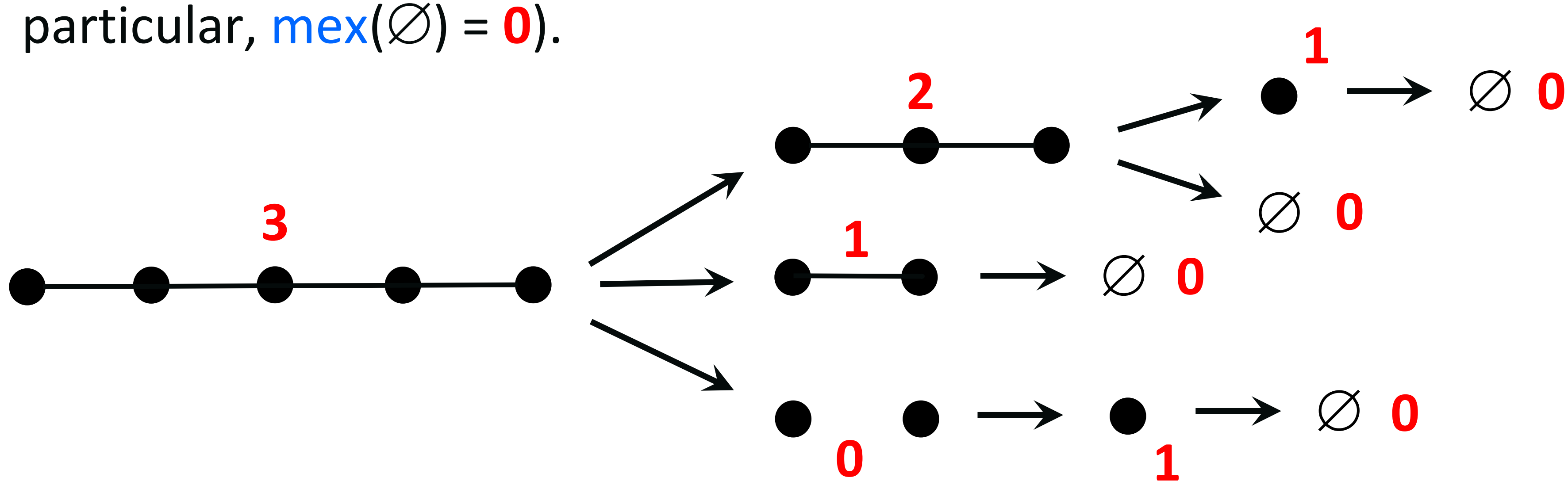


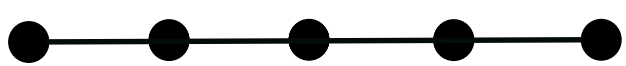
Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, \ldots, G_k\}$, then

$$\sigma(G) = \text{mex}\,(\sigma(G_1), \ldots, \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).



 is thus a winning position...

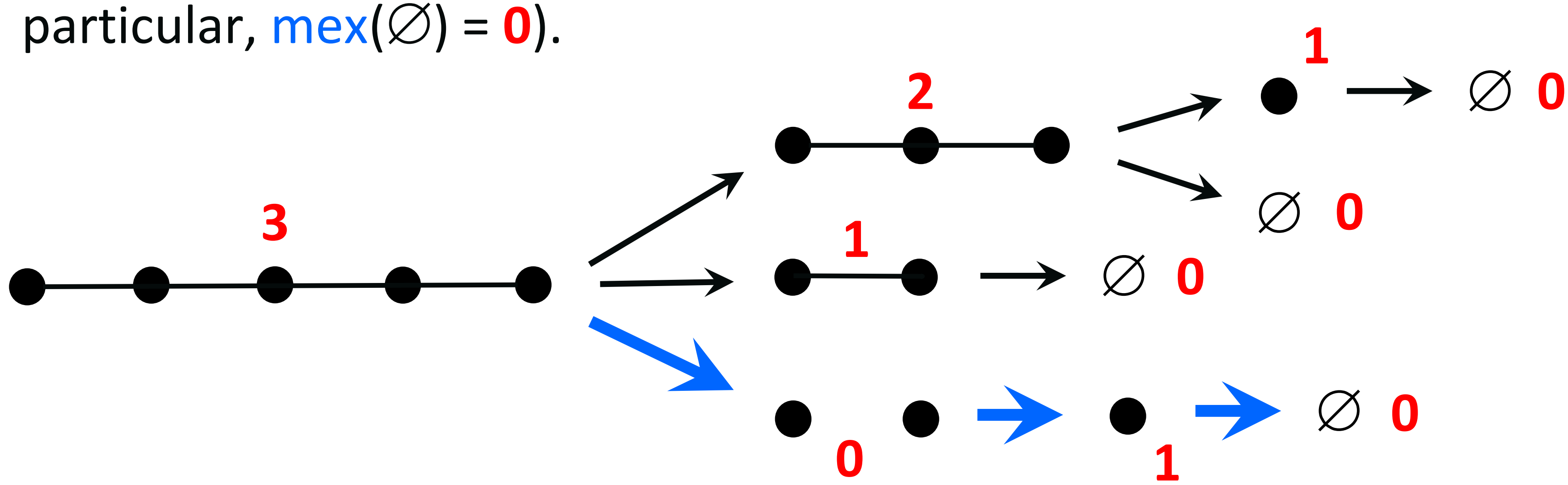
Sprague-Grundy function (2)

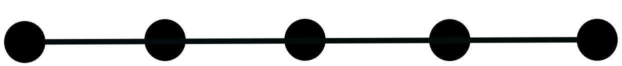
Computing the SG-value of an impartial game (1)

If the set of options of G is $\{G_1, \ldots, G_k\}$, then

$$\sigma(G) = \text{mex}\,(\sigma(G_1), \ldots, \sigma(G_k))$$

where mex(S) is the smallest positive integer value not in S (in particular, mex($\varnothing$) = **0**).



 is thus a winning position...

Sprague-Grundy function (3)

Computing the SG-value of an impartial game (2)

Sprague-Grundy function (3)

Computing the SG-value of an impartial game (2)

If G is a sum of games, say $G = G_1 + \ldots + G_k$, then

$$\sigma(G) = \sigma(G_1) \oplus \ldots \oplus \sigma(G_k)$$

where $\oplus$ denotes the *xor* operation on binary numbers (nim-sum).

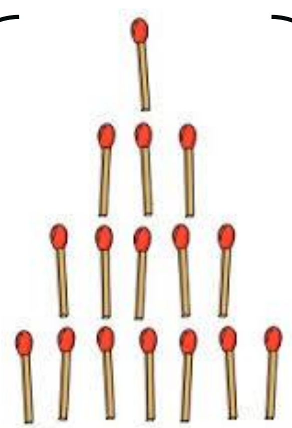
Sprague-Grundy function (3)

Computing the SG-value of an impartial game (2)

If G is a sum of games, say $G = G_1 + \ldots + G_k$, then

$$\sigma(G) = \sigma(G_1) \oplus \ldots \oplus \sigma(G_k)$$

where $\oplus$ denotes the *xor* operation on binary numbers (nim-sum).

$\sigma \Bigg($  $\Bigg)$ $= 1 \oplus 3 \oplus 5 \oplus 7 = 001 \oplus 011 \oplus 101 \oplus 111 = \mathbf{0}$

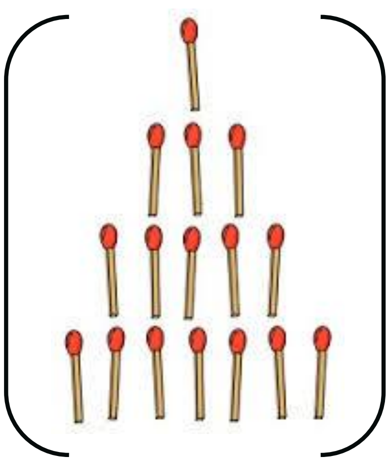
Sprague-Grundy function (3)

Computing the SG-value of an impartial game (2)

If G is a sum of games, say $G = G_1 + \ldots + G_k$, then

$$\sigma(G) = \sigma(G_1) \oplus \ldots \oplus \sigma(G_k)$$

where $\oplus$ denotes the *xor* operation on binary numbers (nim-sum).

σ  $= 1 \oplus 3 \oplus 5 \oplus 7 = 001 \oplus 011 \oplus 101 \oplus 111 = \mathbf{0}$

This position of NIM is thus a losing position...

The graph of a combinatorial game

Game-graph

With every impartial combinatorial game G, one can associate a graph (the game-graph of G), denoted G_g and defined as follows:

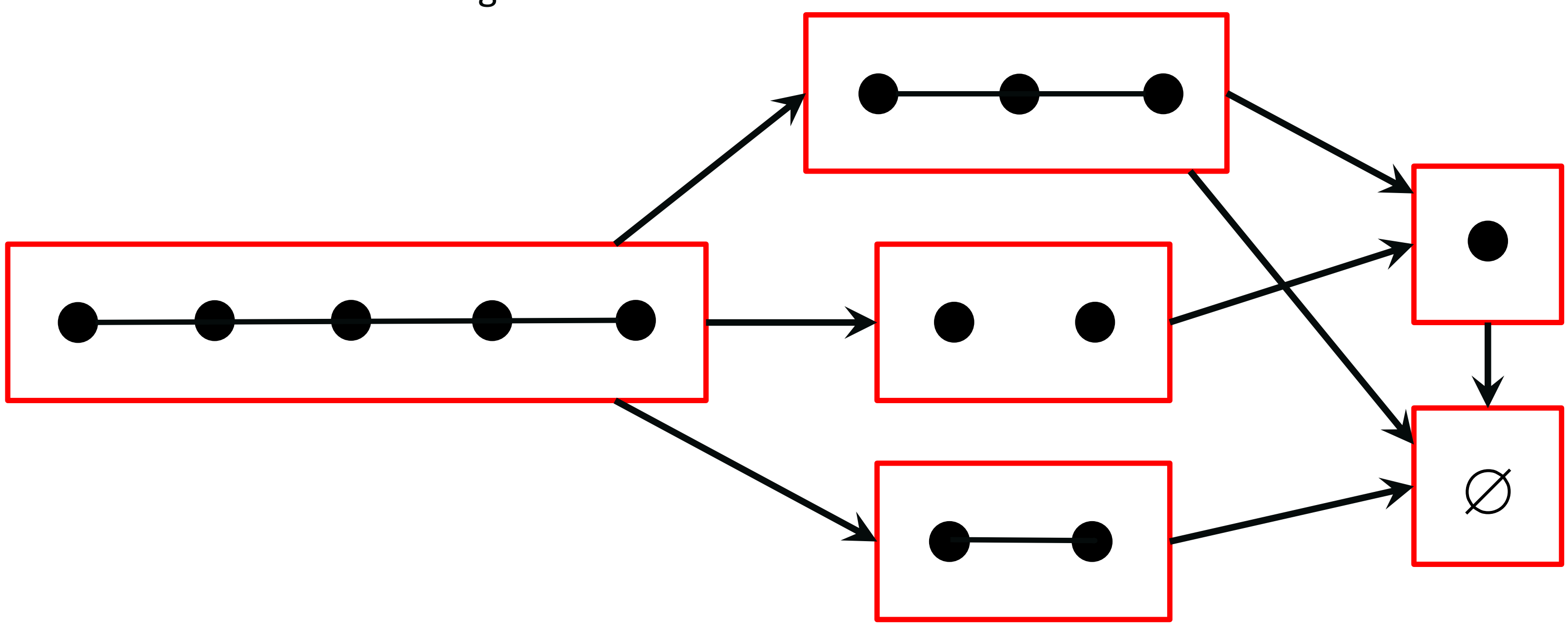
- vertices of G_g are positions of G,
- P_1P_2 is an arc in G_g, whenever P_2 is an option of P_1.

The graph of a combinatorial game

Game-graph

With every impartial combinatorial game G, one can associate a graph (the game-graph of G), denoted G_g and defined as follows:

- vertices of G_g are positions of G,
- P_1P_2 is an arc in G_g, whenever P_2 is an option of P_1.



Playing on the game-graph

Playing on G_g

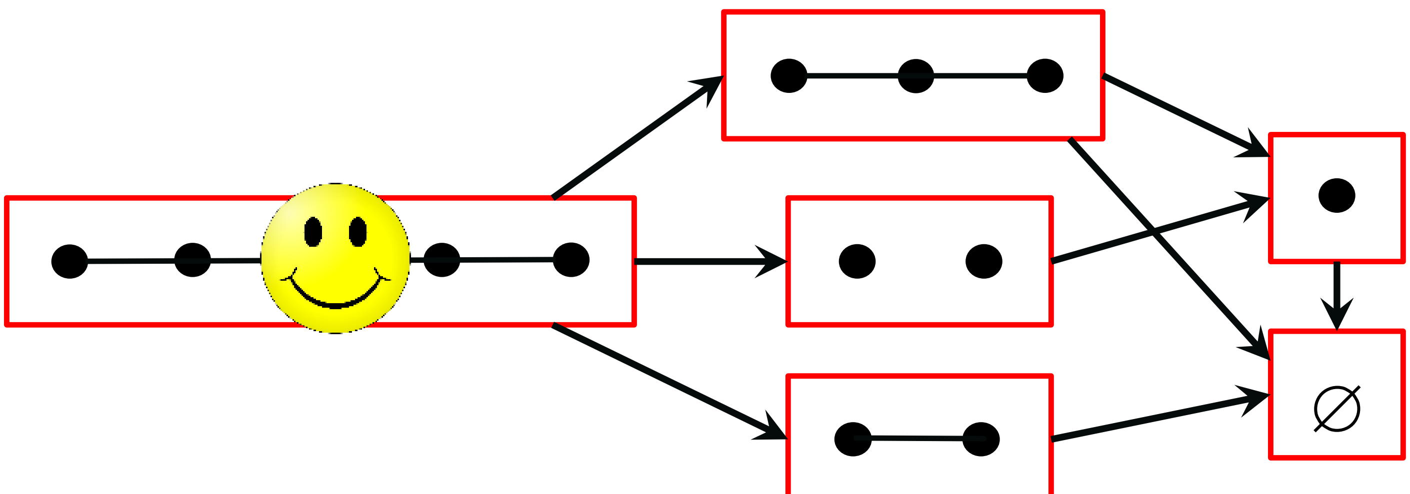
Every impartial combinatorial game G can be viewed as a game on the oriented graph G_g defined as follows:

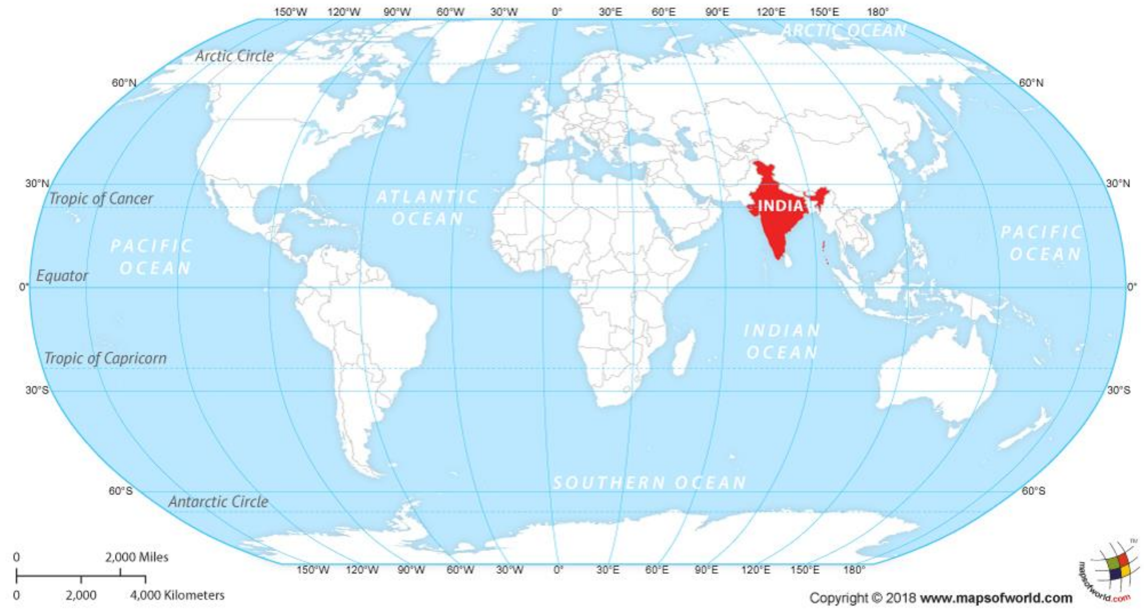
Playing on the game-graph

Playing on G_g

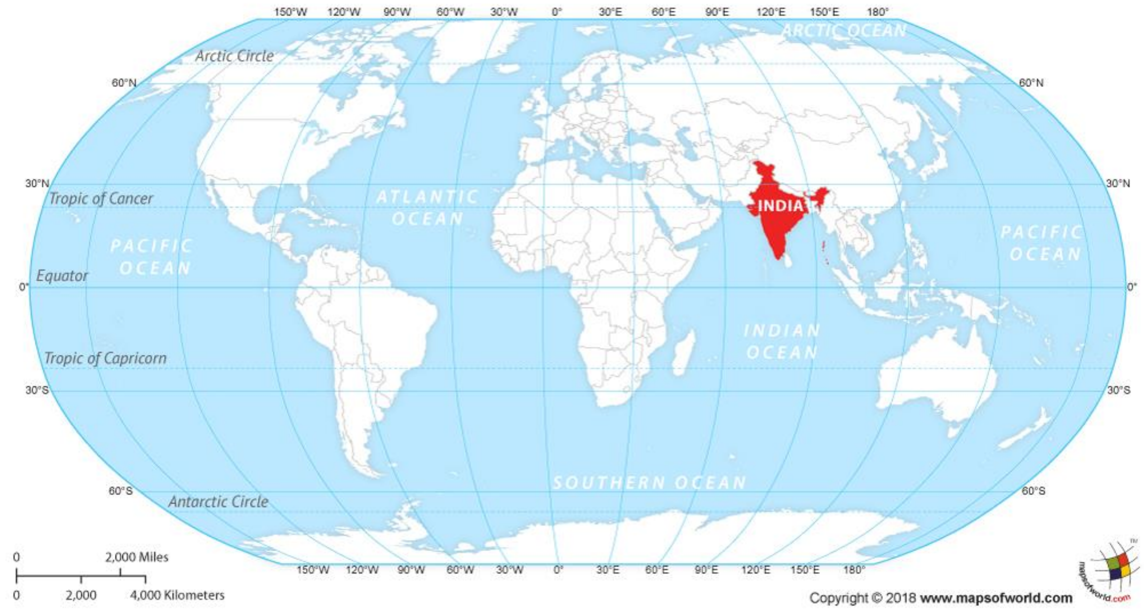
Every impartial combinatorial game G can be viewed as a game on the oriented graph G_g defined as follows:

- a token is put on the initial vertex (initial position),
- on her turn, each player moves the token along one arc,
- the first player unable to move looses (or wins...).





The game GEOGRAPHY



The game GEOGRAPHY

Fiji → Iceland → Denmark → Kiribati → ??

GEOGRAPHY (1)

VERTEX GEOGRAPHY [suggested by R.M. KARP]

The game is played on an undirected graph G. Initially, a token is placed on some “current vertex” v (starting position (G,v)).

GEOGRAPHY (1)

VERTEX GEOGRAPHY [suggested by R.M. KARP]

The game is played on an undirected graph G. Initially, a token is placed on some "current vertex" v (starting position (G,v)).

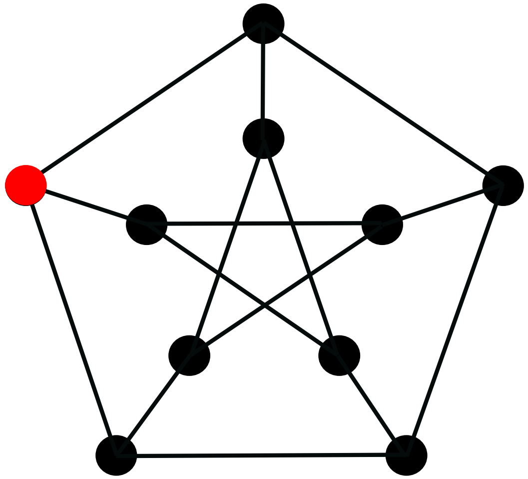
- On her turn, each player moves the token to a neighbour of the current vertex and deletes the current vertex.
  The vertex having the token becomes the current vertex.

GEOGRAPHY (1)

VERTEX GEOGRAPHY [suggested by R.M. KARP]

The game is played on an undirected graph G. Initially, a token is placed on some "current vertex" v (starting position (G,v)).

- On her turn, each player moves the token to a neighbour of the current vertex and deletes the current vertex.
  The vertex having the token becomes the current vertex.

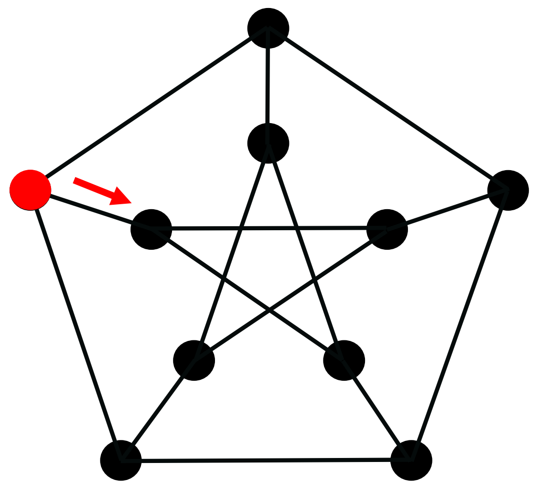


(G,v)

VERTEX GEOGRAPHY [suggested by R.M. KARP]

The game is played on an undirected graph G. Initially, a token is placed on some “current vertex” v (starting position (G,v)).

- On her turn, each player moves the token to a neighbour of the current vertex and deletes the current vertex.
  The vertex having the token becomes the current vertex.

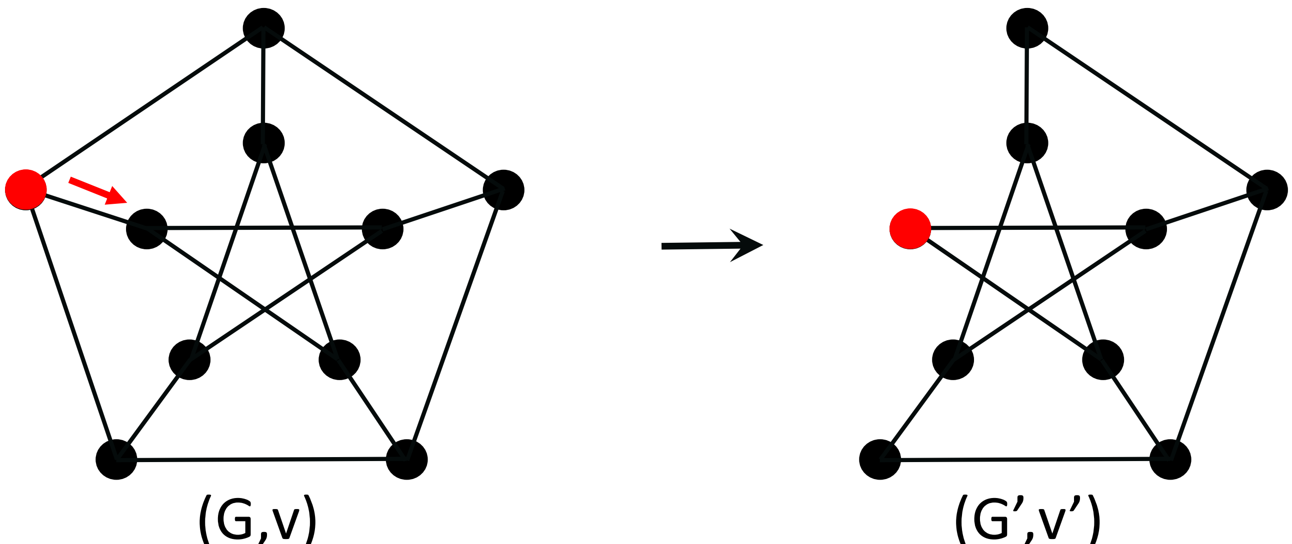


(G,v)

Vertex Geography [suggested by R.M. Karp]

The game is played on an undirected graph G. Initially, a token is placed on some "current vertex" v (starting position (G,v)).

- On her turn, each player moves the token to a neighbour of the current vertex and deletes the current vertex.
  The vertex having the token becomes the current vertex.



GEOGRAPHY (2)

EDGE GEOGRAPHY

The game is played on an undirected graph G. Initially, a token is placed on some “current vertex” v (starting position (G,v)).

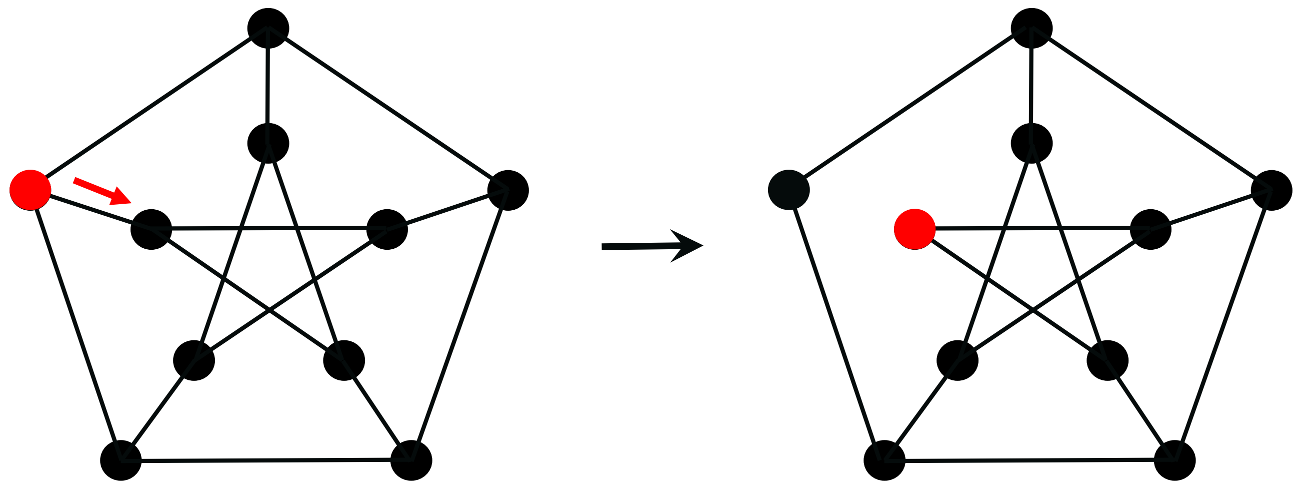
- On her turn, each player moves the token to a neighbour of the current vertex and deletes the traversed edge.
  The vertex having the token becomes the current vertex.

GEOGRAPHY (2)

EDGE GEOGRAPHY

The game is played on an undirected graph G. Initially, a token is placed on some "current vertex" v (starting position (G,v)).

- On her turn, each player moves the token to a neighbour of the current vertex and deletes the traversed edge.
  The vertex having the token becomes the current vertex.



GEOGRAPHY (3)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

GEOGRAPHY (3)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

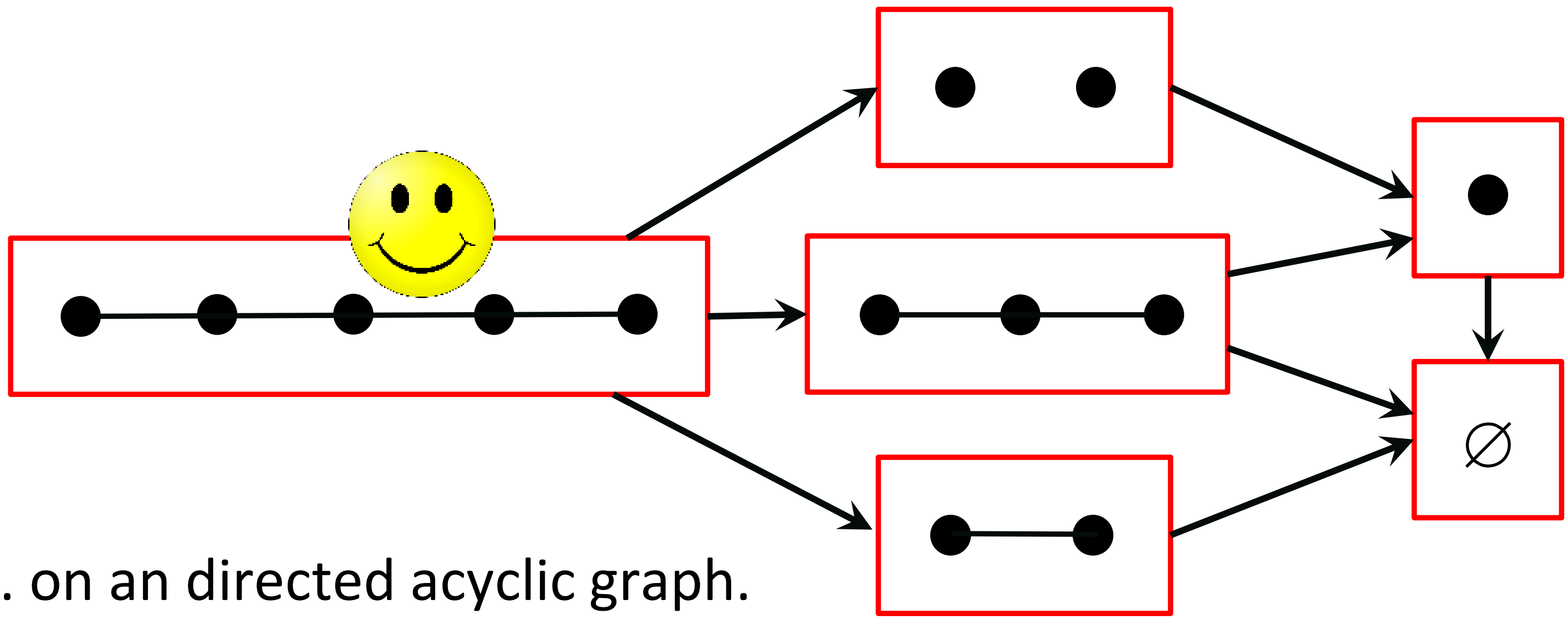
Playing on a game-graph = DIRECTED VERTEX GEOGRAPHY...

GEOGRAPHY (3)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

Playing on a game-graph = DIRECTED VERTEX GEOGRAPHY...



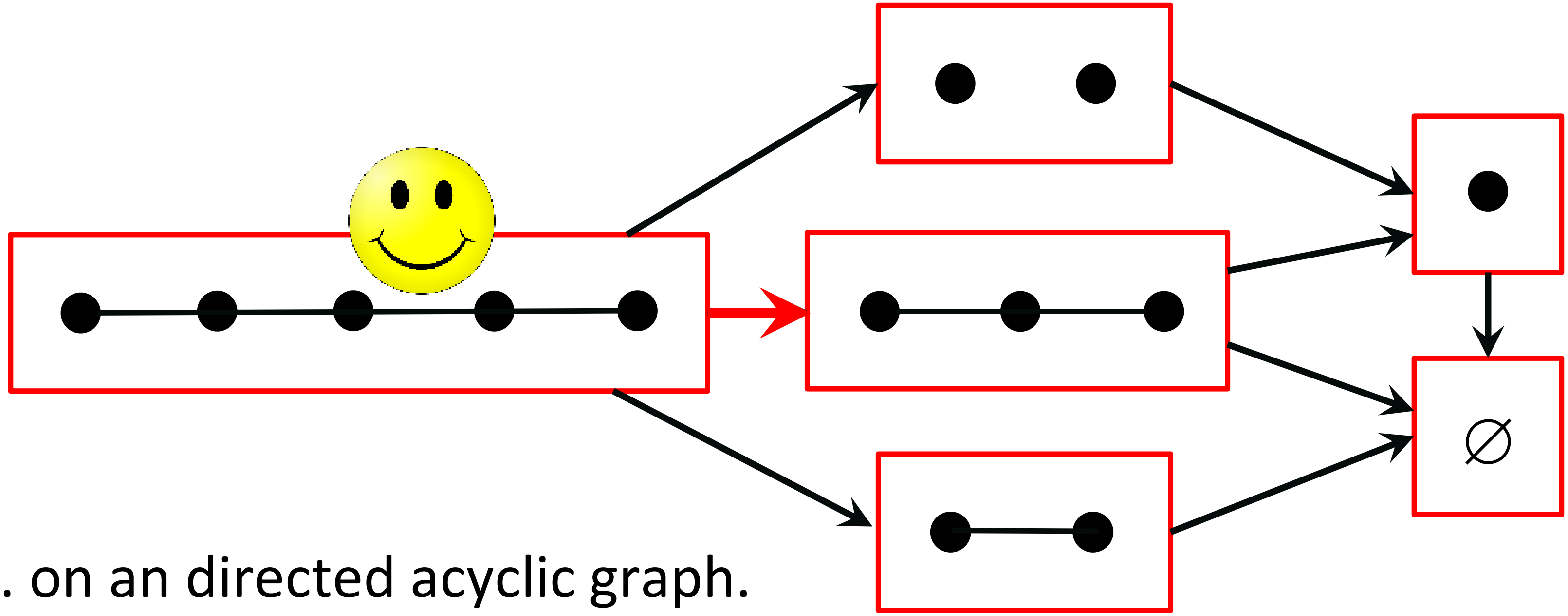
... on an directed acyclic graph.

GEOGRAPHY (3)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

Playing on a game-graph = DIRECTED VERTEX GEOGRAPHY...



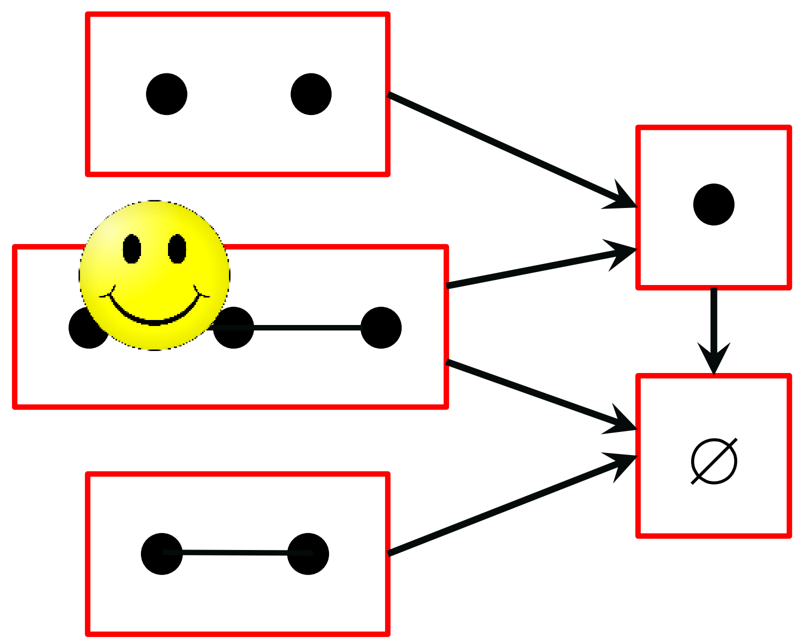
... on an directed acyclic graph.

GEOGRAPHY (3)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

Playing on a game-graph = DIRECTED VERTEX GEOGRAPHY...



... on an directed acyclic graph.

GEOGRAPHY (4)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

Complexity of GEOGRAPHY games (normal play)

(*deciding the outcome of a given position*)

UNDIRECTED VERTEX: polynomial
[A.S. FRAENKEL, E.R. SCHEINERMAN, D. ULLMAN, 1993]

UNDIRECTED EDGE: PSPACE-complete
[A.S. FRAENKEL, E.R. SCHEINERMAN, D. ULLMAN, 1993]

DIRECTED VERTEX: PSPACE-complete
[D. LICHTENSTEIN, M. SIPSER, 1980]

DIRECTED EDGE: PSPACE-complete
[T.J. SCHAEFER, 1978]

GEOGRAPHY (5)

DIRECTED (VERTEX OR EDGE) GEOGRAPHY

The game is played on a directed graph....

Complexity of GEOGRAPHY games

But for misère play, all these four games are PSPACE-complete...

[G. RENAULT, S. SCHMIDT, 2015]

UNDIRECTED VERTEX GEOGRAPHY

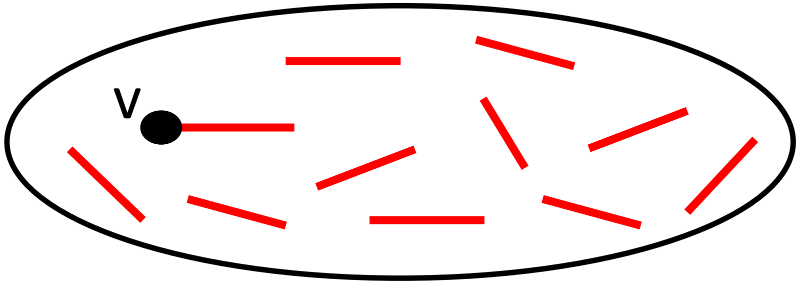
Theorem [A.S. FRAENKEL, E.R. SCHEINERMAN, D. ULLMAN, 1993]

The position (G, v) is a winning position for the game UNDIRECTED VERTEX GEOGRAPHY (normal play) iff every maximum matching (that is, of maximum cardinality) of G saturates v.

UNDIRECTED VERTEX GEOGRAPHY

Theorem [A.S. FRAENKEL, E.R. SCHEINERMAN, D. ULLMAN, 1993]

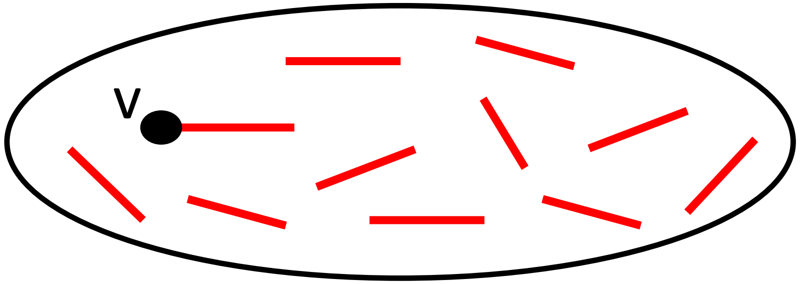
The position (G, v) is a winning position for the game UNDIRECTED VERTEX GEOGRAPHY (normal play) iff every maximum matching (that is, of maximum cardinality) of G saturates v.



UNDIRECTED VERTEX GEOGRAPHY

Theorem [A.S. FRAENKEL, E.R. SCHEINERMAN, D. ULLMAN, 1993]

The position (G, v) is a winning position for the game UNDIRECTED VERTEX GEOGRAPHY (normal play) iff every maximum matching (that is, of maximum cardinality) of G saturates v.



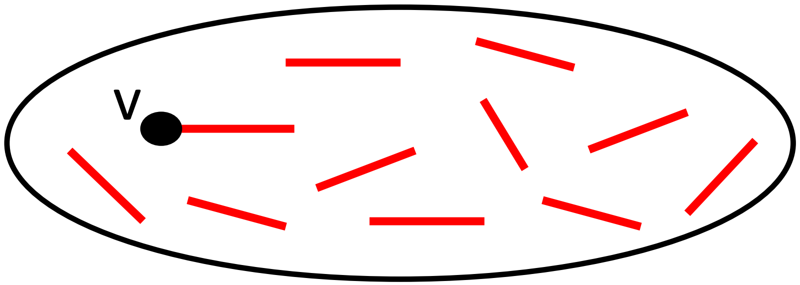
***Proof*.**

- ($\Rightarrow$) **2nd-player winning strategy:** choose a maximum matching M that does not saturate v, and always move along an edge in M.

UNDIRECTED VERTEX GEOGRAPHY

Theorem [A.S. FRAENKEL, E.R. SCHEINERMAN, D. ULLMAN, 1993]

The position (G, v) is a winning position for the game UNDIRECTED VERTEX GEOGRAPHY (normal play) iff every maximum matching (that is, of maximum cardinality) of G saturates v.



***Proof*.**

- ($\Rightarrow$) **2nd-player winning strategy:** choose a maximum matching M that does not saturate v, and always move along an edge in M.
- ($\Leftarrow$) **1st-player winning strategy:** choose a maximum matching M (which thus saturates v) and always move along an edge in M.
  (*if no such move is possible, there exists M' which does not saturate v...*)

DIRECTED VERTEX GEOGRAPHY

Theorem [R.J. NOWAKOWSKI, D.G. POOLE, 1996]

The position $(C_m \square C_n\,, v)$ is a winning position for the game DIRECTED VERTEX GEOGRAPHY whenever:

- *$m = 2$, or*
- *n and m are both even.*

Directed Vertex Geography

Theorem [R.J. Nowakowski, D.G. Poole, 1996]

The position ($C_m \square C_n$, v) is a winning position for the game Directed Vertex Geography whenever:

- *$m = 2$, or*
- *n and m are both even.*

Theorem [R.J. Nowakowski, D.G. Poole, 1996]

The position ($C_3 \square C_n$, v) is a winning position for the game Directed Vertex Geography iff $n > 0$ and $n \equiv 0, 2, 4, 6, 10, 11, 13, 15, 16, 17, 19, 21, 22, 23, 25, 27, 28, 32, 34, 36, 38, 40 \pmod{42}$.

DIRECTED VERTEX GEOGRAPHY

Theorem [R.J. NOWAKOWSKI, D.G. POOLE, 1996]

The position $(C_m \square C_n, v)$ is a winning position for the game DIRECTED VERTEX GEOGRAPHY whenever:

- *$m = 2$, or*
- *n and m are both even.*

Theorem [R.J. NOWAKOWSKI, D.G. POOLE, 1996]

The position $(C_3 \square C_n, v)$ is a winning position for the game DIRECTED VERTEX GEOGRAPHY iff $n > 0$ and $n \equiv 0, 2, 4, 6, 10, 11, 13, 15, 16, 17, 19, 21, 22, 23, 25, 27, 28, 32, 34, 36, 38, 40 \pmod{42}$.

Theorem [M.S. HOGAN, D.G. HORROCKS, 2003]

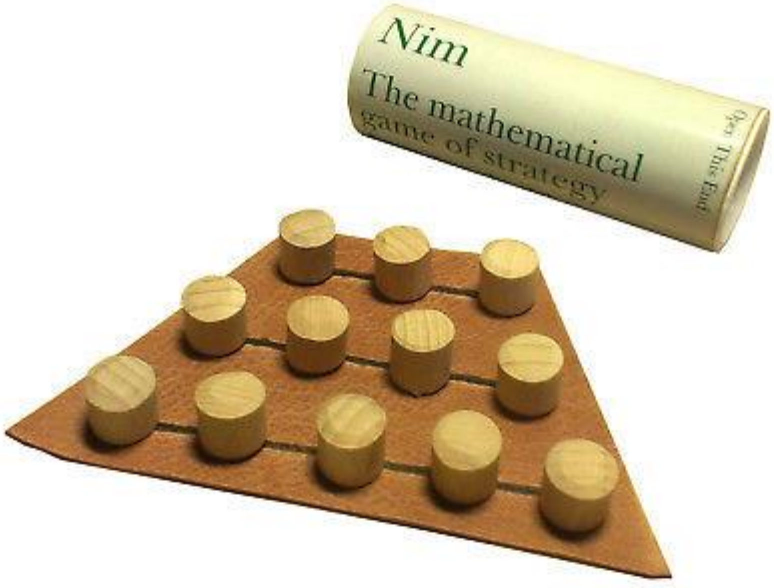
The position $(C_4 \square C_n, v)$ is a losing position for the game DIRECTED VERTEX GEOGRAPHY iff $n \equiv 11 \pmod{12}$.

Geography – Open problems

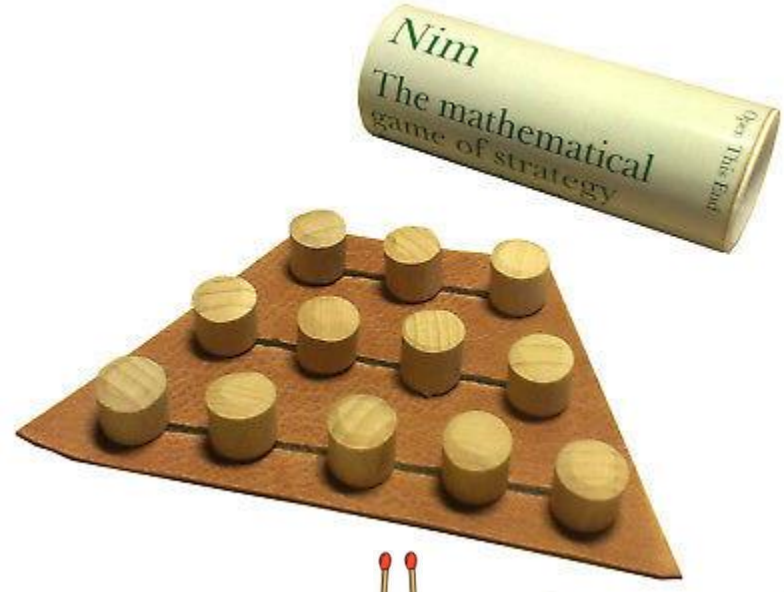
Open Problems.

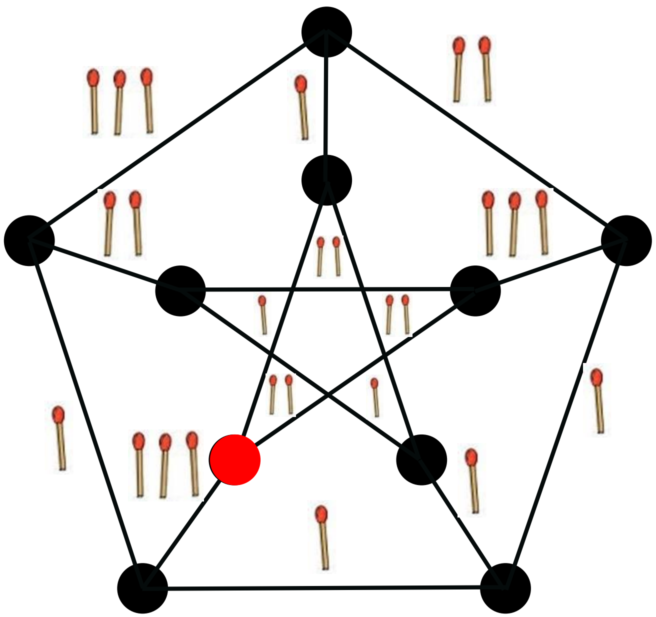
- For which classes of graphs the outcome of GEOGRAPHY (any variant) is "easy" to determine?
- Can you characterize the winning positions of DIRECTED VERTEX GEOGRAPHY on the Cartesian product $C_m \square C_n$ of two directed cycles when m > 4 ?

Playing NIM on graphs

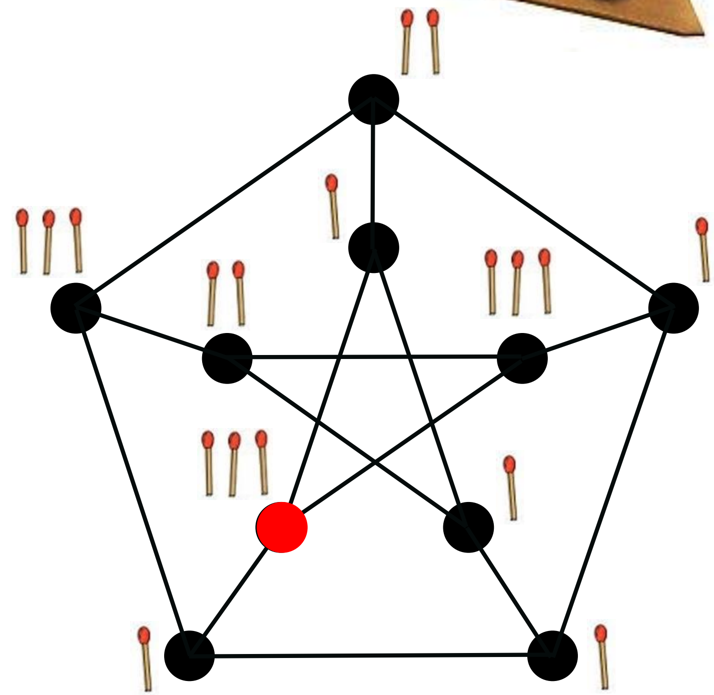


Playing NIM on graphs





or



Edge NimG (1)

Edge NimG [M. Fukuyama, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,

EDGE NIMG (1)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

EDGE NIMG (1)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

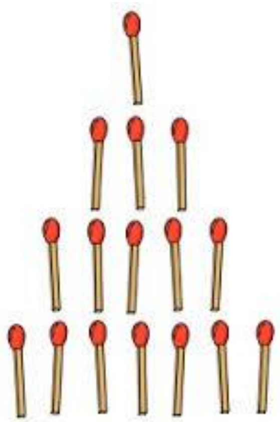
EDGE NIMG (on multigraphs) extends ordinary NIM:

EDGE NIMG (1)

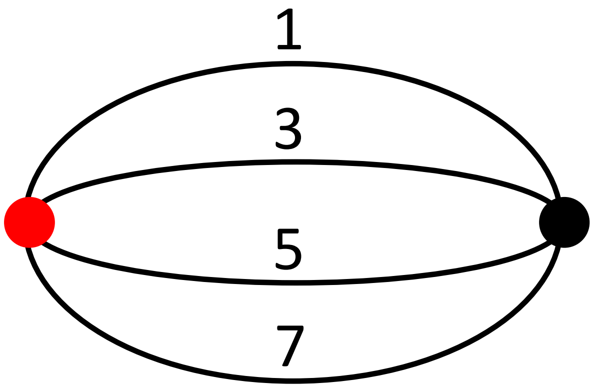
EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

EDGE NIMG (on multigraphs) extends ordinary NIM:



$\Leftrightarrow$



EDGE NIMG (2)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

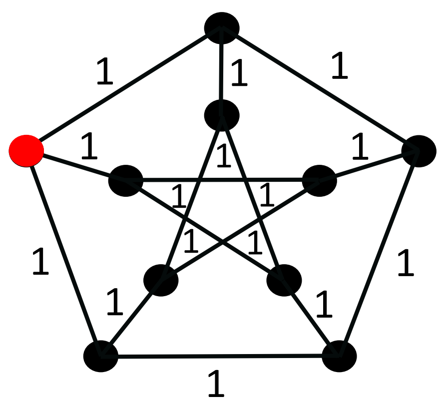
EDGE NIMG extends UNDIRECTED EDGE GEOGRAPHY (PSPACE-complete):

EDGE NIMG (2)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

EDGE NIMG extends UNDIRECTED EDGE GEOGRAPHY (PSPACE-complete):

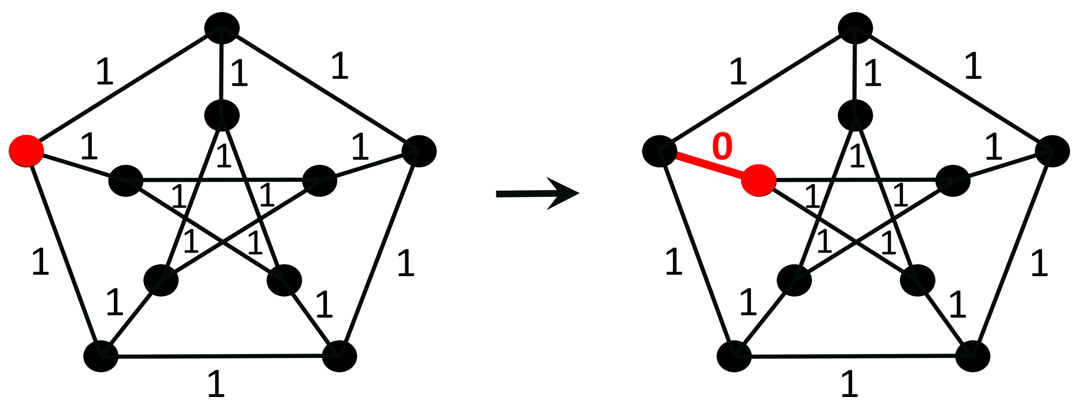


EDGE NIMG (2)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

EDGE NIMG extends UNDIRECTED EDGE GEOGRAPHY (PSPACE-complete):



EDGE NIMG (3)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

FUKUYAMA determined the Sprague-Grundy values of EDGE NIMG positions whenever G is either a cycle or a tree.

EDGE NIMG (3)

EDGE NIMG [M. FUKUYAMA, 2003]

- each edge contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in a unique action:
  - move to a neighbour of the current vertex and delete any non-negative number of tokens on the traversed edge.

FUKUYAMA determined the Sprague-Grundy values of EDGE NIMG positions whenever G is either a cycle or a tree.

He also determined whether a position is a winning or a losing position whenever G is bipartite...

EDGE NIMG (4)

L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.

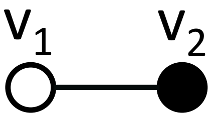
EDGE NIMG (4)

L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.

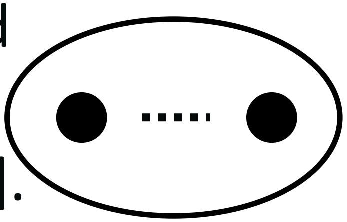
- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].

EDGE NIMG (4)

L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.

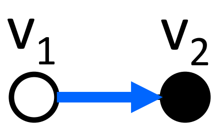


- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].

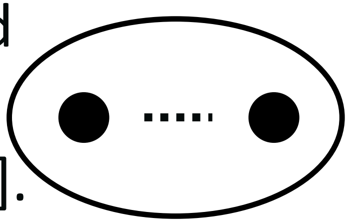


EDGE NIMG (4)

L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.

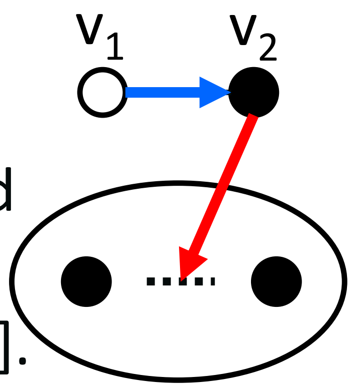


- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].



EDGE NIMG (4)

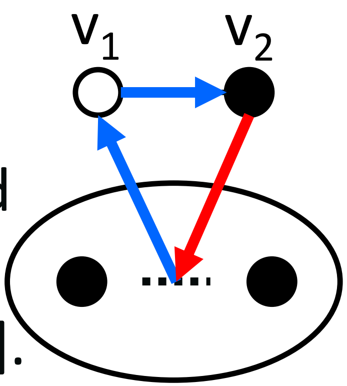
L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].

EDGE NIMG (4)

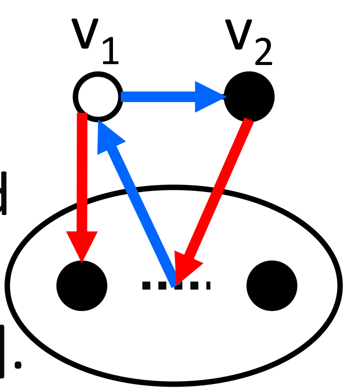
L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].

EDGE NIMG (4)

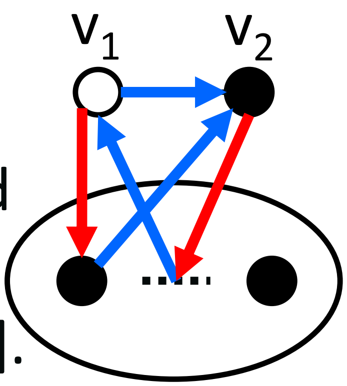
L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].

EDGE NIMG (4)

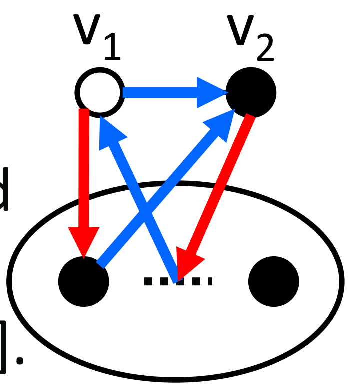
L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].

EDGE NIMG (4)

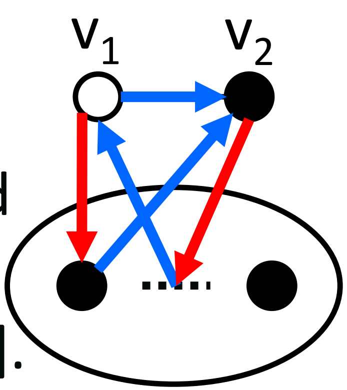
L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].
- Therefore, every position (K_n, v), n ≥ 2, is a winning position.

EDGE NIMG (4)

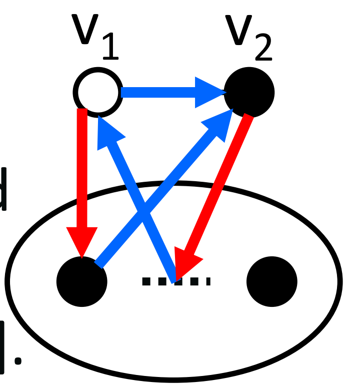
L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].
- Therefore, every position (K_n, v), n ≥ 2, is a winning position.
- Let Q_n denote the n-dimensional hypercube. A position (Q_n, v) is a winning position iff n is odd [L. ERICKSON, W. SHREVE, 2012].

EDGE NIMG (4)

L. ERICKSON (2010), studied the case where each edge has exactly one token (UNDIRECTED EDGE GEOGRAPHY), and gave several sufficient conditions for a position to be a winning position.



- If G contains two twin vertices v_1 and v_2 (that is, v_1 and v_2 have the same closed neighbourhood) then the position (G, v_1) is a winning position [L. ERICKSON, 2010].
- Therefore, every position (K_n, v), $n \geq 2$, is a winning position.
- Let Q_n denote the n-dimensional hypercube. A position (Q_n, v) is a winning position iff n is odd [L. ERICKSON, W. SHREVE, 2012].

Open Problem.

- What about such graphs with an arbitrary number of tokens at each vertex? with at most two tokens?

VERTEX NIMG (1)

VERTEX NIMG [G. STOCKMAN, A. FRIEZE, J. VERA, 2004]

- each vertex contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,

VERTEX NIMG (1)

VERTEX NIMG [G. STOCKMAN, A. FRIEZE, J. VERA, 2004]

- each vertex contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in two actions:
  - delete any non-negative number of tokens on the current vertex, and then
  - move to a neighbour of the current vertex.

VERTEX NIMG (1)

VERTEX NIMG [G. STOCKMAN, A. FRIEZE, J. VERA, 2004]

- each vertex contains a given (non-negative) number of tokens,
- one vertex of the graph is the starting vertex,
- each move consists in two actions:
  - delete any non-negative number of tokens on the current vertex, and then
  - move to a neighbour of the current vertex.

Several variants can thus be considered:

delete-then-move or move-then-delete

loops on vertices are allowed or not (move-then-delete)

move to an "empty vertex" is allowed or not (delete-then-move)

VERTEX NIMG (2)

VERTEX NIMG, delete-then-move, no loop

VERTEX NIMG (2)

VERTEX NIMG, delete-then-move, no loop

- If the number of tokens is bounded by some constant, then deciding whether a position is winning or losing can be done in polynomial time [G. STOCKMAN, A. FRIEZE, J. VERA, 2004].

VERTEX NIMG (2)

VERTEX NIMG, delete-then-move, no loop

- If the number of tokens is bounded by some constant, then deciding whether a position is winning or losing can be done in polynomial time [G. STOCKMAN, A. FRIEZE, J. VERA, 2004].

VERTEX NIMG, move-then-delete, loop on every vertex

VERTEX NIMG (2)

VERTEX NIMG, delete-then-move, no loop

- If the number of tokens is bounded by some constant, then deciding whether a position is winning or losing can be done in polynomial time [G. STOCKMAN, A. FRIEZE, J. VERA, 2004].

VERTEX NIMG, move-then-delete, loop on every vertex

- If the number of tokens is bounded by some constant $k \geq 2$, then deciding whether a position is winning or losing is PSPACE-complete [K.G. BURKE, O.C. GEORGE, 2014].

VERTEX NIMG (2)

VERTEX NIMG, delete-then-move, no loop

- If the number of tokens is bounded by some constant, then deciding whether a position is winning or losing can be done in polynomial time [G. STOCKMAN, A. FRIEZE, J. VERA, 2004].

VERTEX NIMG, move-then-delete, loop on every vertex

- If the number of tokens is bounded by some constant $k \geq 2$, then deciding whether a position is winning or losing is PSPACE-complete [K.G. BURKE, O.C. GEORGE, 2014].

Open Problem.

- What is the computational complexity of VERTEX NIMG on graphs with optional loops?

VERTEXNIM (1)

In all versions of NIMG, the game may end with remaining tokens on the graph, contrary to ordinary NIM...

VERTEXNIM (1)

In all versions of NIMG, the game may end with remaining tokens on the graph, contrary to ordinary NIM...

UNDIRECTED VERTEXNIM [E. DUCHÊNE, G. RENAULT, 2014]

- Variant of delete-then-move VERTEX NIMG:
  - delete any non-negative number of tokens on the current vertex, and then
  - move to the next current vertex (having a non-negative number of tokens), *along a path whose internal vertices do not have any token*.

VERTEXNIM (1)

In all versions of NIMG, the game may end with remaining tokens on the graph, contrary to ordinary NIM...

UNDIRECTED VERTEXNIM [E. DUCHÊNE, G. RENAULT, 2014]

➢ Variant of delete-then-move VERTEX NIMG:
- delete any non-negative number of tokens on the current vertex, and then
- move to the next current vertex (having a non-negative number of tokens), *along a path whose internal vertices do not have any token*.

➢ The outcome of any UNDIRECTED VERTEXNIM position (loops are allowed) can be computed in polynomial time.

VERTEXNIM (2)

DIRECTED VERTEXNIM [E. DUCHÊNE, G. RENAULT, 2014]

VertexNim (2)

Directed VertexNim [E. Duchêne, G. Renault, 2014]

- The outcome of any Directed VertexNim position (a loop at each vertex, the graph is strongly connected) can be computed in polynomial time.

VertexNim (2)

Directed VertexNim [E. Duchêne, G. Renault, 2014]

- The outcome of any Directed VertexNim position (a loop at each vertex, the graph is strongly connected) can be computed in polynomial time.
- Let C_n be a directed cycle of order n, $n \geq 3$, with at least two tokens at each vertex. For every vertex v, the outcome of the position (C_n, v) can be computed in polynomial time.

VertexNim (2)

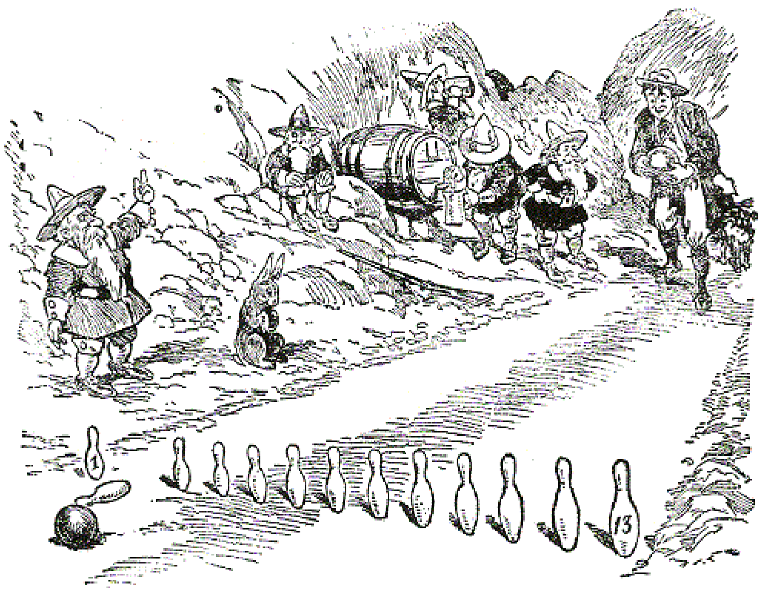
Directed VertexNim [E. Duchêne, G. Renault, 2014]

- The outcome of any Directed VertexNim position (a loop at each vertex, the graph is strongly connected) can be computed in polynomial time.
- Let C_n be a directed cycle of order n, $n \geq 3$, with at least two tokens at each vertex. For every vertex v, the outcome of the position (C_n, v) can be computed in polynomial time.

Open Problems.

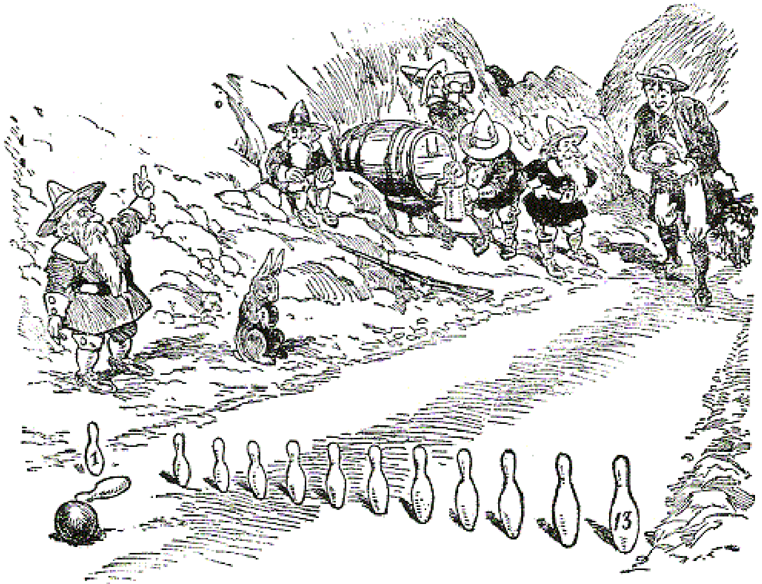
- What about strongly connected graphs with optional loops?
- What about C_n if some vertices have only one token?
- What about the move-then-delete version?

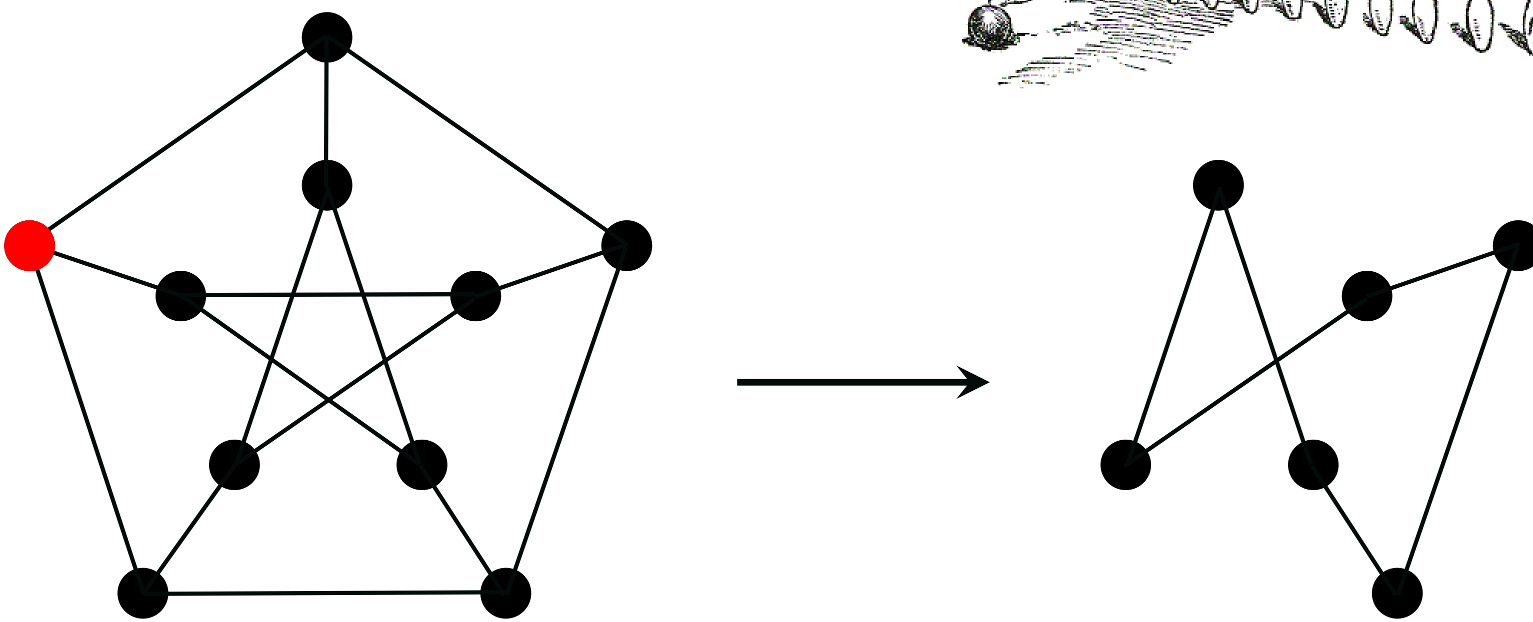
NODE-KAYLES



Geography | Nim on graphs | Node-Kayles | k-Colouring | 0.33 game | Timber! | Conclusion

NODE-KAYLES

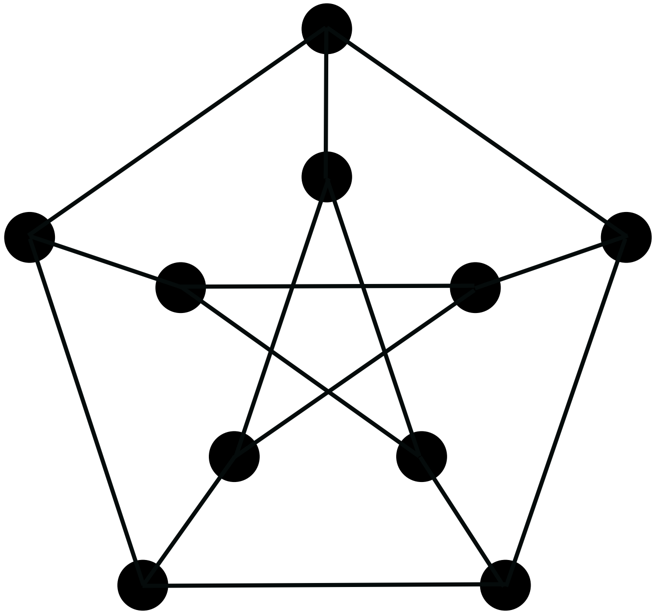




Recall our first game...

Recall our first game...

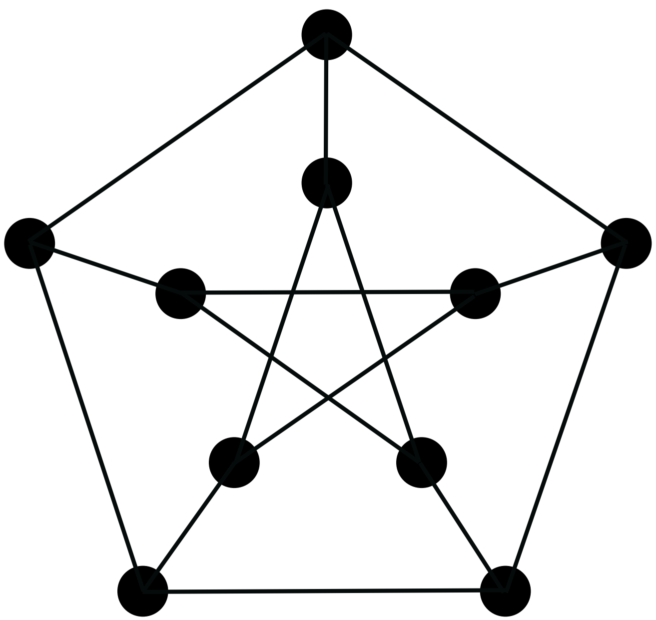
Take your favorite graph, e.g. Petersen graph.



Recall our first game...

Take your favorite graph, e.g. Petersen graph.

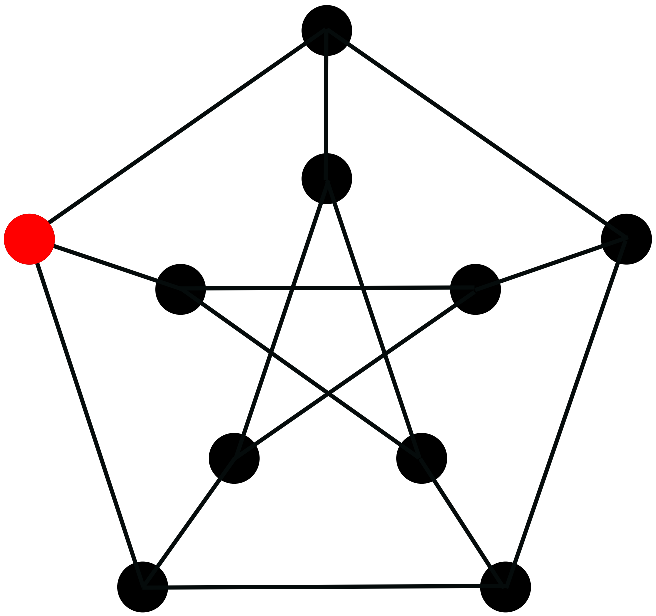
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Recall our first game...

Take your favorite graph, e.g. Petersen graph.

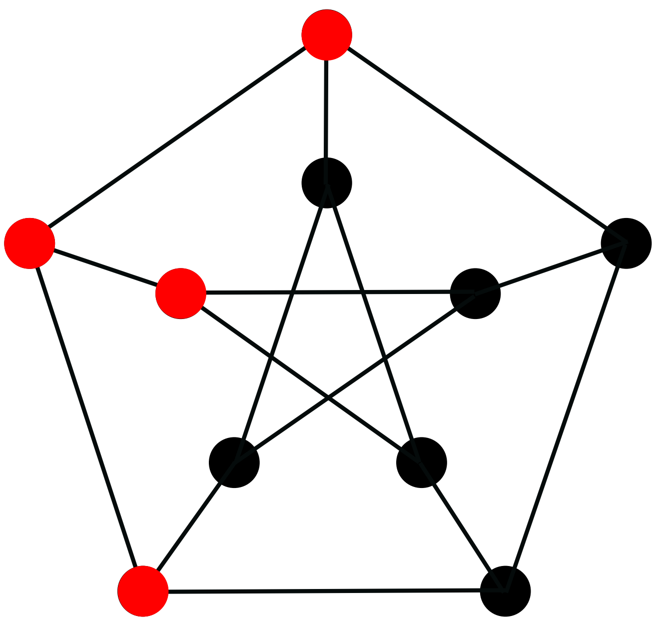
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Recall our first game...

Take your favorite graph, e.g. Petersen graph.

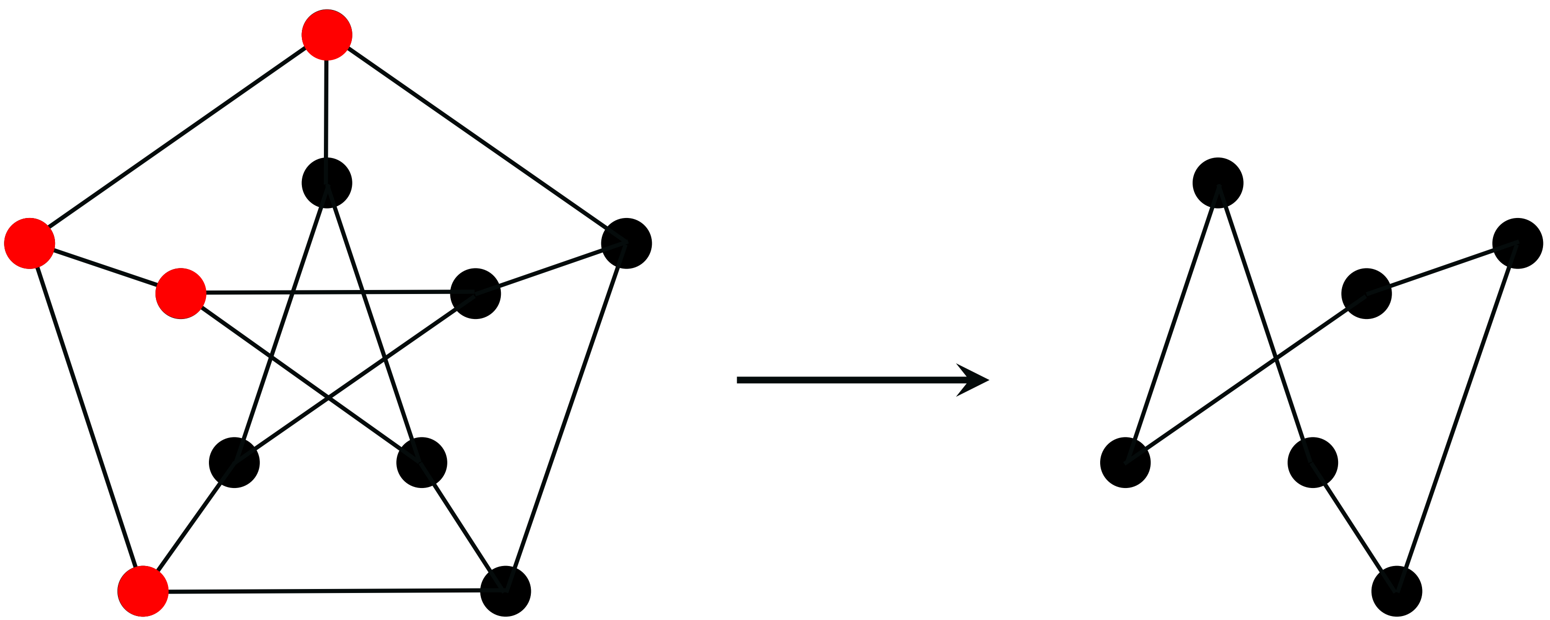
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Recall our first game...

Take your favorite graph, e.g. Petersen graph.

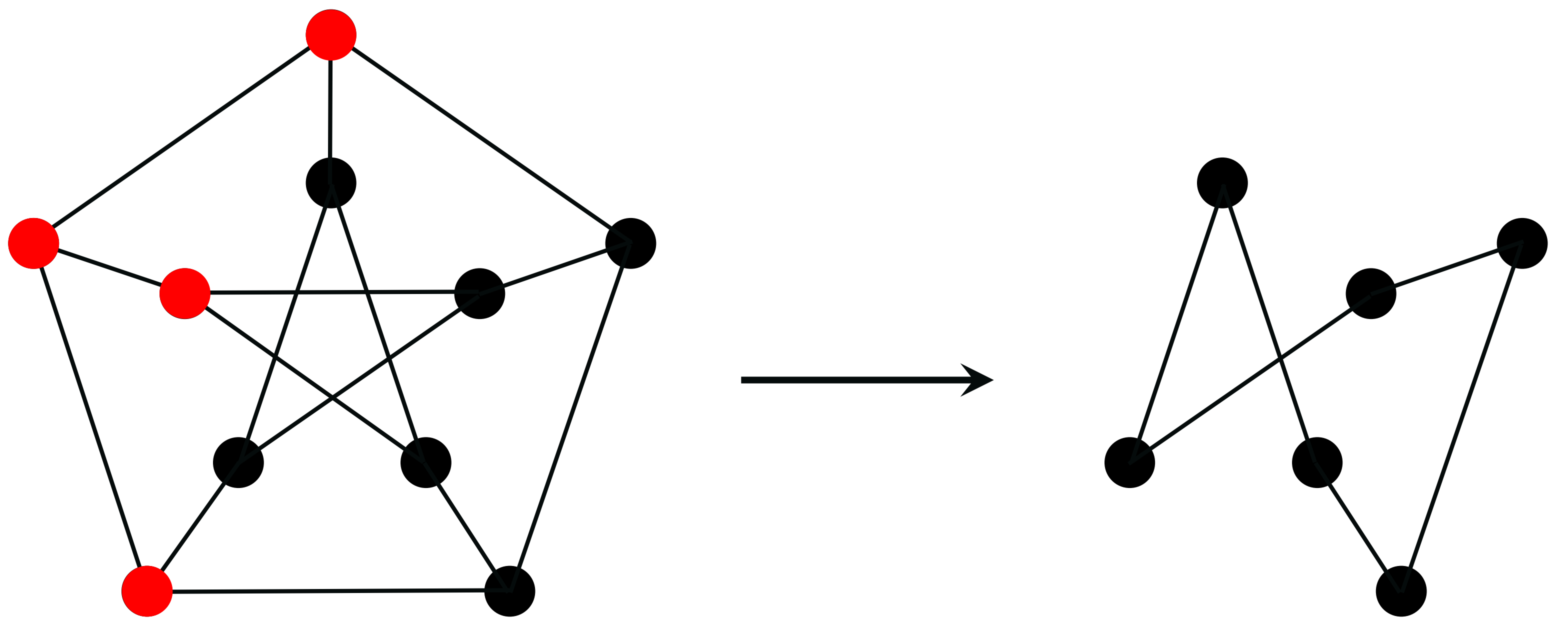
On her turn, each player chooses a vertex and deletes its closed neighbourhood...



Recall our first game...

Take your favorite graph, e.g. Petersen graph.

On her turn, each player chooses a vertex and deletes its closed neighbourhood...



The first player unable to move looses the game...

NODE-KAYLES – COMPLEXITY (1)

Theorem [T.J. SCHAEFER, 1978]

Determining whether a given position (graph) is a winning position or a losing position for NODE-KAYLES is PSPACE-complete.

NODE-KAYLES – COMPLEXITY (1)

Theorem [T.J. SCHAEFER, 1978]

Determining whether a given position (graph) is a winning position or a losing position for NODE-KAYLES is PSPACE-complete.

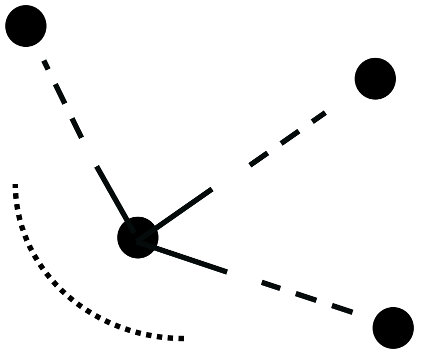
Theorem [H. BODLAENDER, D. KRATSCH, 2002]

Determining whether a given position G is a winning position or a losing position for NODE-KAYLES is polynomial whenever G is a cocomparability graph, a circular arc graph, a cograph, or has bounded asteroidal number.

NODE-KAYLES – COMPLEXITY (2)

Theorem [R. FLEISCHER, G. TRIPPEN, 2004]

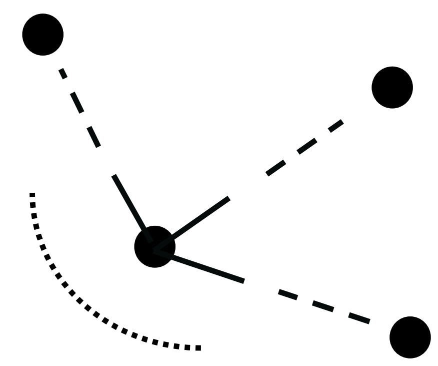
Determining whether a subdivided star with bounded degree is a winning position or a losing position for NODE-KAYLES is polynomial.



Node-kayles – Complexity (2)

Theorem [R. Fleischer, G. Trippen, 2004]

Determining whether a subdivided star with bounded degree is a winning position or a losing position for Node-kayles is polynomial.



Theorem [H. Bodlaender, D. Kratsch, 2011]

Determining whether a given position G with n vertices is a winning position or a losing position for Node-kayles can be done in time $O(1.6052^n)$, or in time $O(1.4423^n)$ if G is a tree.

NODE-KAYLES on paths (DAWSON'S CHESS)

Sprague-Grundy sequence

The Sprague-Grundy sequence of NODE-KAYLES on paths is the (infinite) sequence of Sprague-Grundy values:

$$\sigma(P_0)\ \sigma(P_1)\ \sigma(P_2)\ \sigma(P_3)\ \ldots$$

NODE-KAYLES on paths (DAWSON'S CHESS)

Sprague-Grundy sequence

The Sprague-Grundy sequence of NODE-KAYLES on paths is the (infinite) sequence of Sprague-Grundy values:

$$\sigma(P_0)\ \sigma(P_1)\ \sigma(P_2)\ \sigma(P_3)\ \dots$$

The Sprague-Grundy sequence of NODE-KAYLES on paths is ultimately periodic, with a period of length 34 and a preperiod of length 51:

0	1	1	2	0	3	1	1	0	3	3	2	2	4	0	5	2	2
18	3	3	0	1	1	3	0	2	1	1	0	4	5	2	7	4	0
35	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5	2
52	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4	8
69	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5	9
86	3	3	0	1	1	3	0	2	1	1	0	4	5	3	7	4	8
103	1	1	2	0	3	1	1	0	3	3	2	2	4	4	5	5	9
120	3	3	0	1	1	3	0	2	...								

Compound games (1)

Sum of games (reminder)

The (disjunctive) sum of G1 and G2 is the game G1 + G2, played as follows:

- on her turn, each player chooses the current position in G1 or in G2, and then moves according to the rules of G1 or G2, respectively,
- the game ends as soon as a player has no move in any of the two games.

Compound games (1)

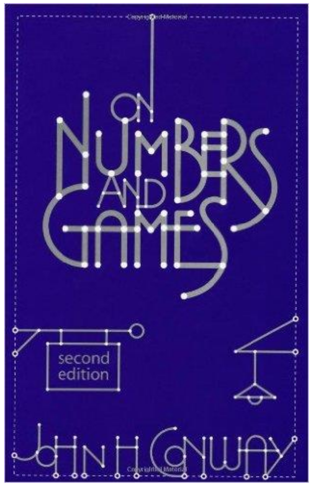
Sum of games (reminder)

The (disjunctive) sum of G1 and G2 is the game G1 + G2, played as follows:

- on her turn, each player chooses the current position in G1 or in G2, and then moves according to the rules of G1 or G2, respectively,
- the game ends as soon as a player has no move in any of the two games.

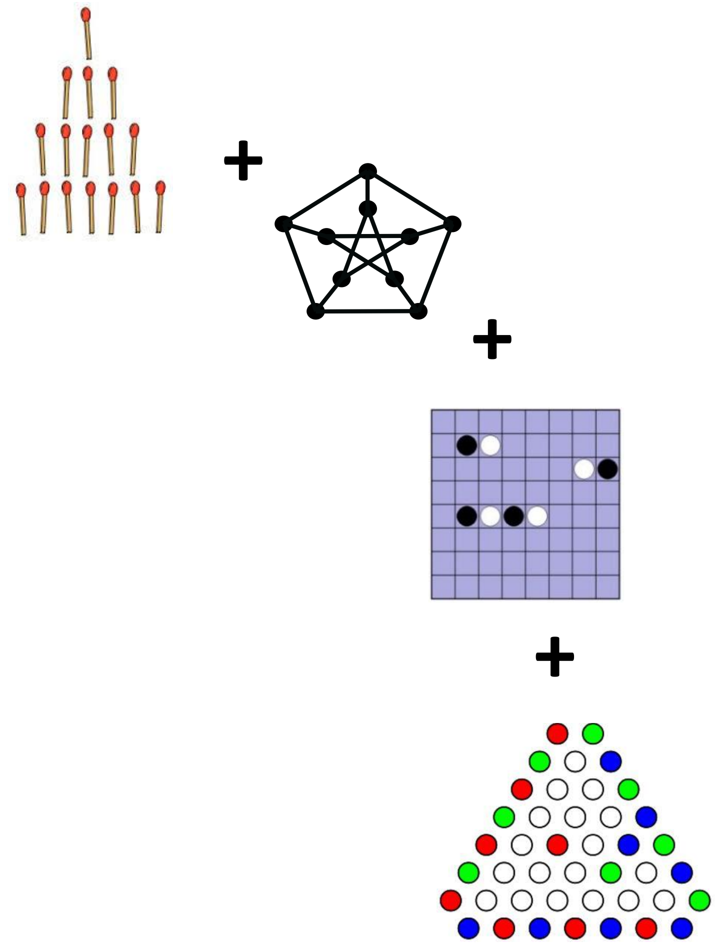
Compound games

In his book (1976), JOHN H. CONWAY introduced 12 distinct notions of compound games, following an inspiring paper of C.A.B. SMITH (1966).



Compound games (2)

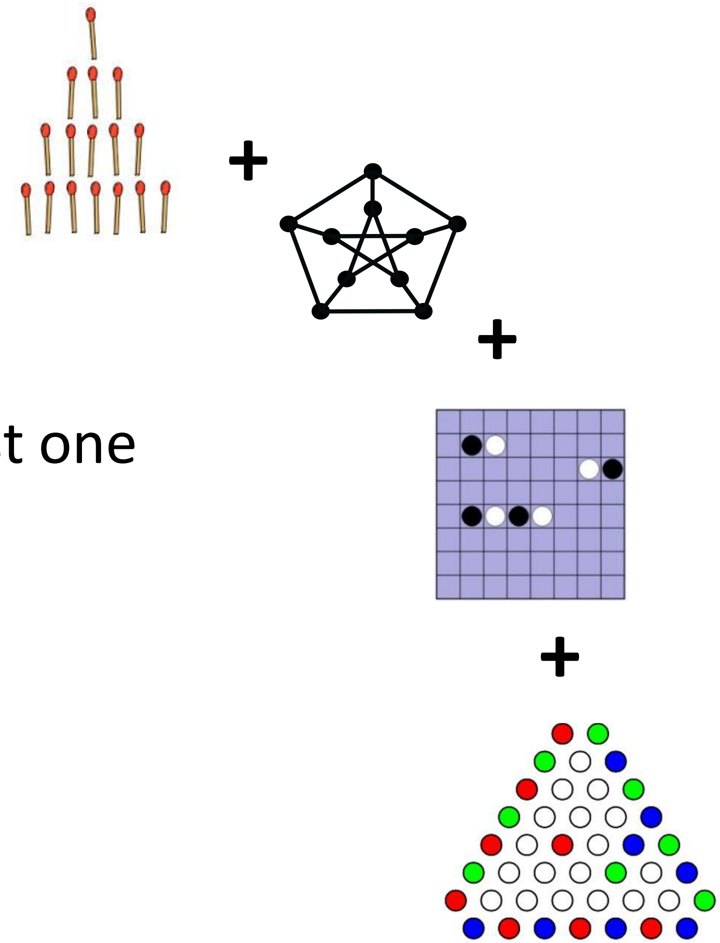
How to play in G_1 + ... + G_k?



Compound games (2)

How to play in $G_1 + \ldots + G_k$?

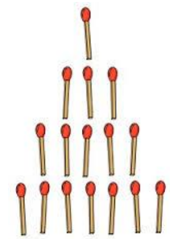
- **Component selection**
  - one component (disjunctive sum),
  - all components (conjonctive sum),
  - any number of components, at least one (selective sum).



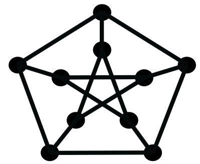
Compound games (2)

How to play in $G_1 + \ldots + G_k$?

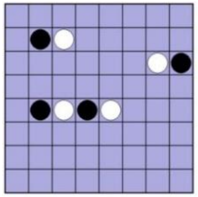
- **Component selection**
  - one component (disjunctive sum),
  - all components (conjonctive sum),
  - any number of components, at least one (selective sum).
- **Ending rule**
  - all components have ended (long rule),
  - one component has ended (short rule).



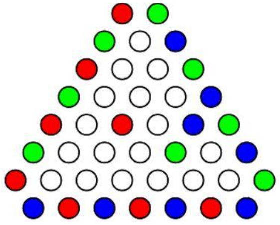
+



+

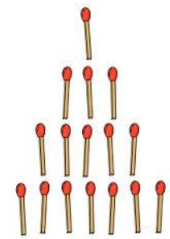


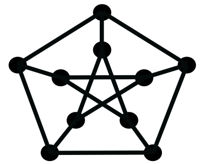
+

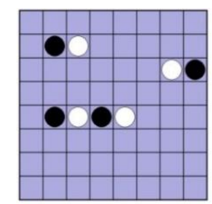


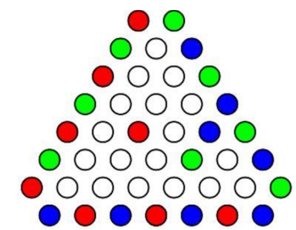
Compound games (2)

How to play in $G_1 + \ldots + G_k$?





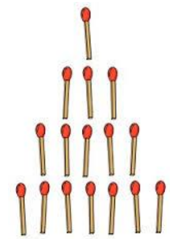




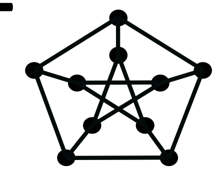
- **Component selection**
  - one component (disjunctive sum),
  - all components (conjonctive sum),
  - any number of components, at least one (selective sum).
- **Ending rule**
  - all components have ended (long rule),
  - one component has ended (short rule).
- **Winning rule**
  - normal play,
  - misère play.

Compound games (2)

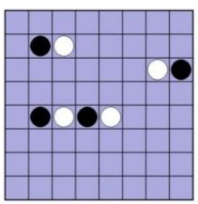
How to play in $G_1 + \ldots + G_k$?



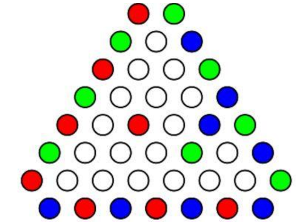
+



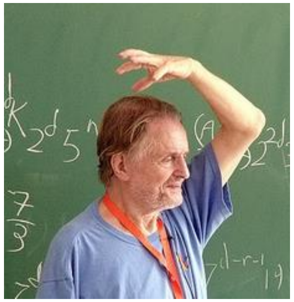
+

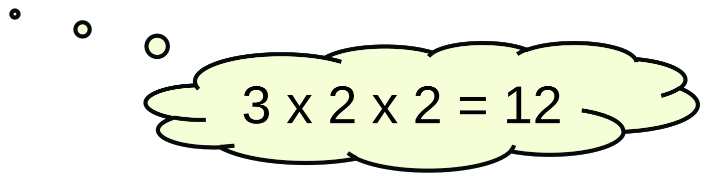


+



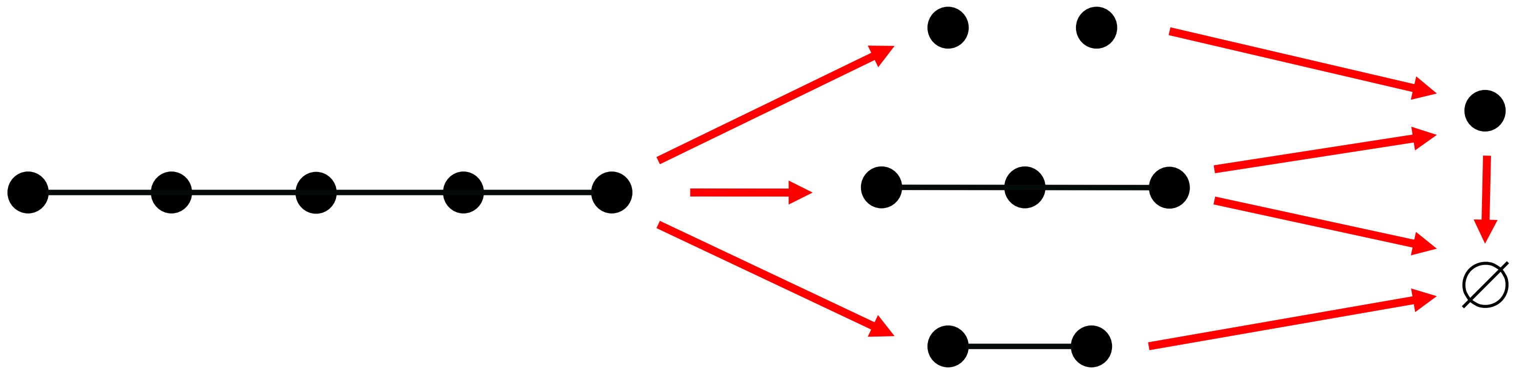
- **Component selection**
  - one component (disjunctive sum),
  - all components (conjonctive sum),
  - any number of components, at least one (selective sum).
- **Ending rule**
  - all components have ended (long rule),
  - one component has ended (short rule).
- **Winning rule**
  - normal play,
  - misère play.





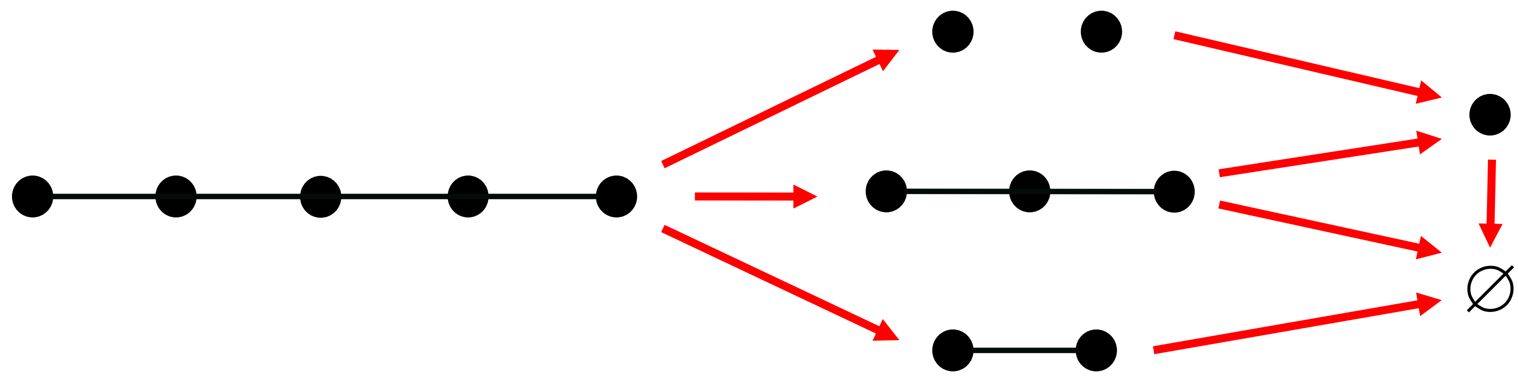
Let's play again... (1)

Let us consider the path P_5 of order 5:



Let's play again... (1)

Let us consider the path P_5 of order 5:

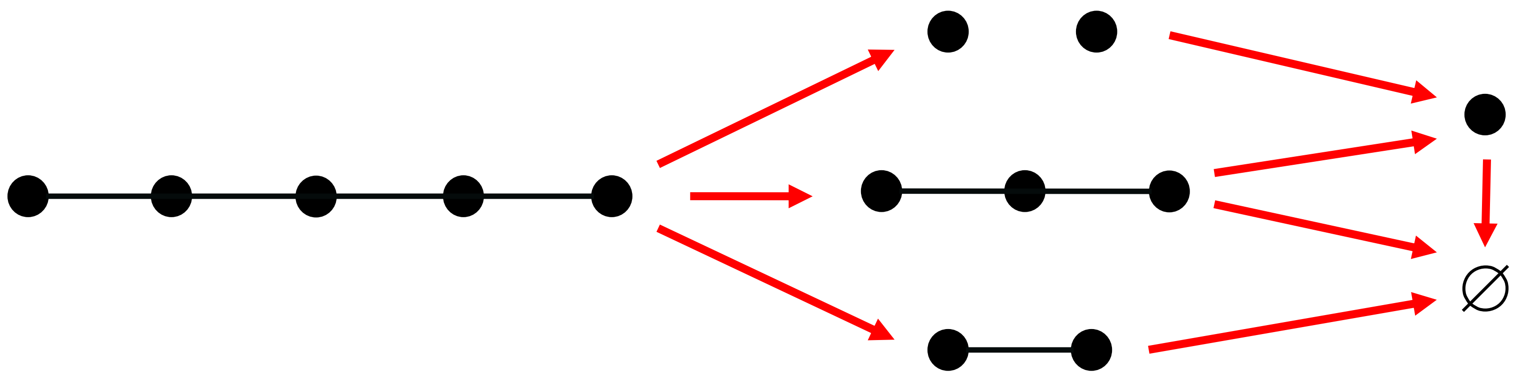


Disjunctive sum, long rule, normal play

- Component selection: one component
- Ending rule: all components must have ended
- Winning rule: the first player unable to move looses

Let's play again... (1)

Let us consider the path P_5 of order 5:



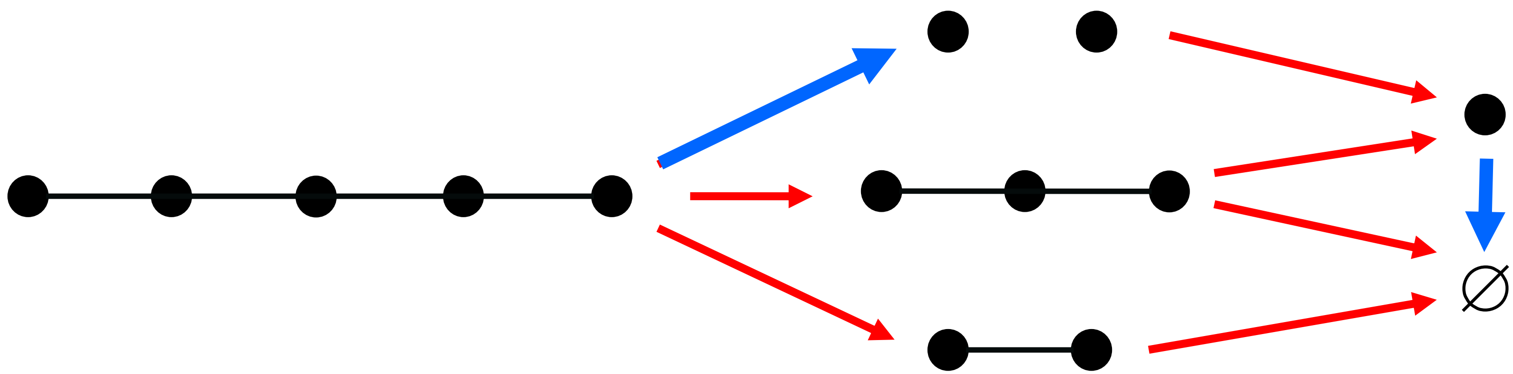
Disjunctive sum, long rule, normal play

- Component selection: one component
- Ending rule: all components must have ended
- Winning rule: the first player unable to move looses

Is P_5 a winning or a losing position?

Let's play again... (1)

Let us consider the path P_5 of order 5:



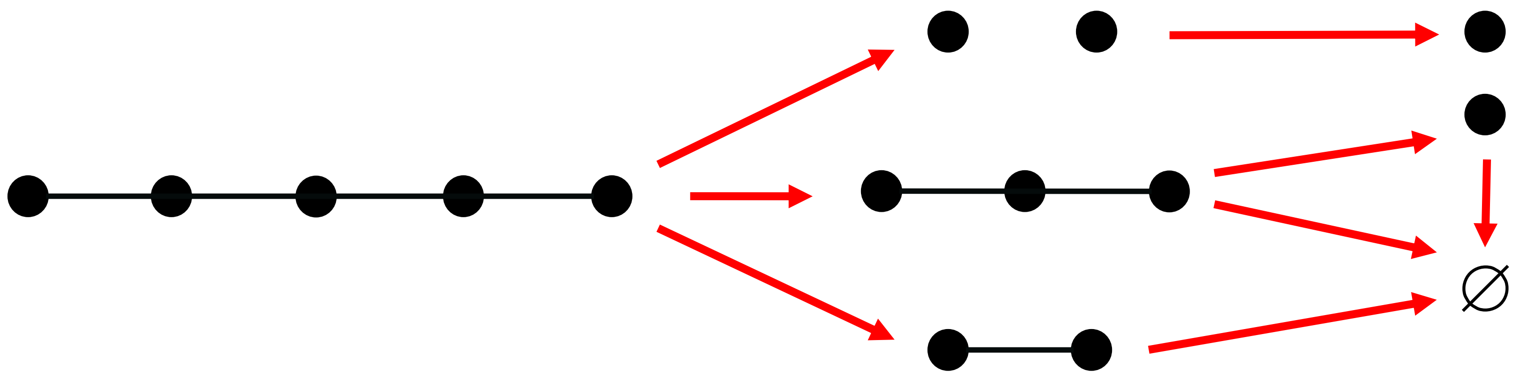
Disjunctive sum, long rule, normal play

- Component selection: one component
- Ending rule: all components must have ended
- Winning rule: the first player unable to move looses

Is P_5 a winning or a losing position? **winning**

Let's play again... (2)

Let us consider the path P_5 of order 5:



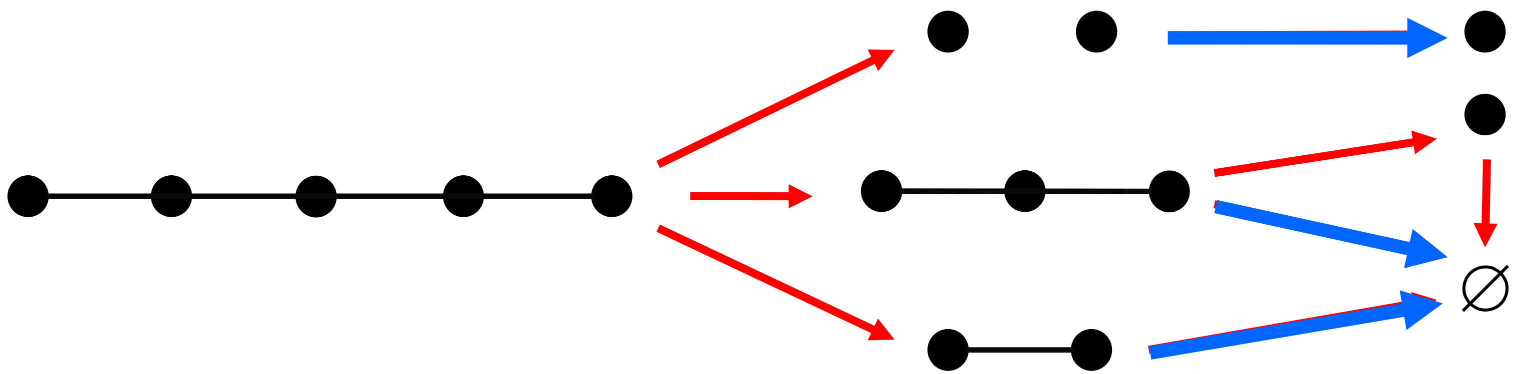
Disjunctive sum, <u>short rule</u>, normal play

- Component selection: one component
- Ending rule: <u>one component has ended</u>
- Winning rule: the first player unable to move looses

Is P_5 a winning or a losing position?

Let’s play again... (2)

Let us consider the path P_5 of order 5:



Disjunctive sum, <u>short rule</u>, normal play

- Component selection: one component
- Ending rule: <u>one component has ended</u>
- Winning rule: the first player unable to move looses

Is P_5 a winning or a losing position?

losing

Foreclosed Sprague-Grundy number of paths

- The foreclosed Sprague-Grundy sequence of paths (under normal play) is ultimately periodic:
  - preperiod of length **245**,
  - period of length **84**.

n	$F^+(P_n)$				
0–49	****001120	0112031122	3112334105	3415534255	3225532255
50–99	0225042253	4423344253	4455341553	4285322853	4285442804
100–149	4283442234	4253345533	1253322533	2253422534	2253422334
150–199	2233425334	4533425532	2553425544	2554425344	2234425334
200–249	5533125342	2533225342	2534225342	2334223342	5334453342
250–299	5532255342	5344255442	5344253442	5334553342	5342253322
300–349	5342253422	5342233422	3342533425	3342553225	...

Disjunctive sum, short rule (2)

Foreclosed Sprague-Grundy number of paths

- The foreclosed Sprague-Grundy sequence of paths (under normal play) is ultimately periodic:
  - preperiod of length **245**,
  - period of length **84**.
- The number of losing positions is finite:

L = { 0, 4, 5, 9, 10, 14, 28, 50, 54, 98 }

n	$F^+(P_n)$				
0–49	****001120	0112031122	3112334105	3415534255	3225532255
50–99	0225042253	4423344253	4455341553	4285322853	4285442804
100–149	4283442234	4253345533	1253322533	2253422534	2253422334
150–199	2233425334	4533425532	2553425544	2554425344	2234425334
200–249	5533125342	2533225342	2534225342	2334223342	5334453342
250–299	5532255342	5344255442	5344253442	5334553342	5342253322
300–349	5342253422	5342233422	3342533425	3342553225	...

Disjunctive sum, short rule (2)

Foreclosed Sprague-Grundy number of paths

- The foreclosed Sprague-Grundy sequence of paths (under normal play) is ultimately periodic:
  - preperiod of length **245**,
  - period of length **84**.

still open for misère play...

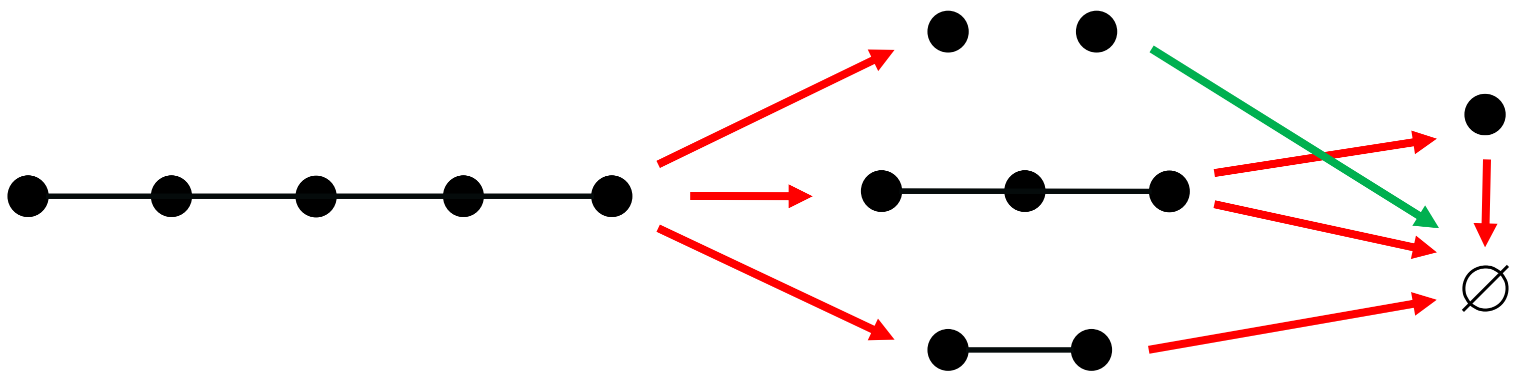
- The number of losing positions is finite:

L = { 0, 4, 5, 9, 10, 14, 28, 50, 54, 98 }

n	$F^+(P_n)$				
0–49	****001120	0112031122	3112334105	3415534255	3225532255
50–99	0225042253	4423344253	4455341553	4285322853	4285442804
100–149	4283442234	4253345533	1253322533	2253422534	2253422334
150–199	2233425334	4533425532	2553425544	2554425344	2234425334
200–249	5533125342	2533225342	2534225342	2334223342	5334453342
250–299	5532255342	5344255442	5344253442	5334553342	5342253322
300–349	5342253422	5342233422	3342533425	3342553225	...

Let's play again... (3)

Let us consider the path P_5 of order 5:



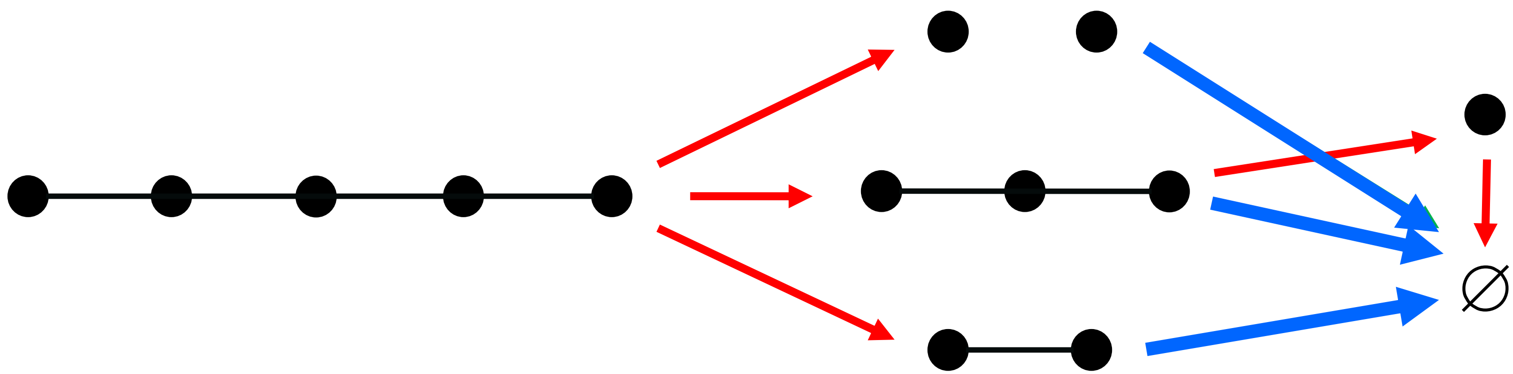
<u>Conjunctive sum</u>, long rule, normal play

- Component selection: <u>all components</u>
- Ending rule: all components have ended
- Winning rule: the first player unable to move looses

Is P_5 a winning or a losing position?

Let's play again... (3)

Let us consider the path P_5 of order 5:



<u>Conjunctive sum</u>, long rule, normal play

- Component selection: <u>all components</u>
- Ending rule: all components have ended
- Winning rule: the first player unable to move looses

Is P_5 a winning or a losing position? **losing**

Conjunctive sum, long rule (1)

Suspense number

- Strategy: losing quickly on losing components and postponing win as long as possible on winning ones...

Conjunctive sum, long rule (1)

Suspense number

- Strategy: losing quickly on losing components and postponing win as long as possible on winning ones...
- The suspense number **$S^+(G)$** (normal play) of a position G is the number of coming turns, using this strategy:

Conjunctive sum, long rule (1)

Suspense number

- Strategy: losing quickly on losing components and postponing win as long as possible on winning ones...
- The suspense number **$S^+(G)$** (normal play) of a position G is the number of coming turns, using this strategy:
  - **$S^+(G) = 0$** if G is an ended position,

Conjunctive sum, long rule (1)

Suspense number

- Strategy: losing quickly on losing components and postponing win as long as possible on winning ones...
- The suspense number $S^+(G)$ (normal play) of a position G is the number of coming turns, using this strategy:
  - $S^+(G) = 0$ if G is an ended position,
  - if G' is an option of G with maximal even suspense, then $S^+(G) = S^+(G') + 1$,
  - if no such option exists and G'' is an option of G with minimal odd suspense, then $S^+(G) = S^+(G'') + 1$.

Conjunctive sum, long rule (1)

Suspense number

- Strategy: losing quickly on losing components and postponing win as long as possible on winning ones...
- The suspense number $S^+(G)$ (normal play) of a position G is the number of coming turns, using this strategy:
  - $S^+(G) = 0$ if G is an ended position,
  - if G' is an option of G with maximal even suspense, then $S^+(G) = S^+(G') + 1$,
  - if no such option exists and G" is an option of G with minimal odd suspense, then $S^+(G) = S^+(G'') + 1$.

A position G is a winning position iff **$S^+(G)$ is odd**...

Conjunctive sum, long rule (2)

Suspense number of paths

- The suspense sequence of paths (normal play) has a geometric period with geometric ratio 2.

Conjunctive sum, long rule (2)

Suspense number of paths

- The suspense sequence of paths (normal play) has a geometric period with geometric ratio 2.

  For every $n \geq 0$, we have:

  - **$S^+(P_k) = 2n$**, if $k = 5(2^n - 1)$,
  - **$S^+(P_k) = 2n + 1$**, if $5(2^n - 1) < k < 5(2^{n+1} - 1) - 1$,
  - **$S^+(P_k) = 2n + 2$**, if $k = 5(2^{n+1} - 1) - 1$.

Conjunctive sum, long rule (2)

Suspense number of paths

- The suspense sequence of paths (normal play) has a geometric period with geometric ratio 2.

  For every $n \geq 0$, we have:

  - $\mathbf{S^+(P_k) = 2n}$, if $k = 5(2^n - 1)$,
  - $\mathbf{S^+(P_k) = 2n + 1}$, if $5(2^n - 1) < k < 5(2^{n+1} - 1) - 1$,
  - $\mathbf{S^+(P_k) = 2n + 2}$, if $k = 5(2^{n+1} - 1) - 1$.

- The set of losing positions is:

$$\{\, 5(2^n - 1),\ n \geq 0 \,\} \cup \{\, 5(2^{n+1} - 1) - 1,\ n \geq 0 \,\}$$

Compound NODE-KAYLES on paths

Theorem [A. GUIGNARD, E.S., 2009]

*For **ten** over twelve versions of compound NODE-KAYLES on paths, the set of losing positions can be characterized.*

The two remaining unsolved versions are the following:

- *disjunctive sum, misère play, long rule (DAWSON's problem, 1935),*
- *disjunctive sum, misère play, short rule.*

Compound NODE-KAYLES on paths

Theorem [A. GUIGNARD, E.S., 2009]

*For **ten** over twelve versions of compound NODE-KAYLES on paths, the set of losing positions can be characterized.*

The two remaining unsolved versions are the following:

- *disjunctive sum, misère play, long rule (DAWSON's problem, 1935),*
- *disjunctive sum, misère play, short rule.*

Compound version	Losing set $\mathcal{L}$
disj. comp., normal play	$\{0, 4, 8, 14, 19, 24, 28, 34, 38, 42\} \cup \{54 + 34i, 58 + 34i, 62 + 34i, 72 + 34i, 76 + 34i, i \geq 0\}$
disj. comp., misère play	*unsolved*
dim. disj. comp., normal play	$\{0, 4, 5, 9, 10, 14, 28, 50, 54, 98\}$
dim. disj. comp., misère play	*unsolved*
conj. comp., normal play	$\{0, 4, 5, 9, 10\}$
conj. comp., misère play	$\{1, 2\}$
cont. conj. comp., normal play	$\{5(2^n - 1), n \geq 0\} \cup \{5(2^{n+1} - 1) - 1, n \geq 0\}$
cont. conj. comp., misère play	$\{7.2^n - 6, n \geq 0\} \cup \{7.2^n - 5, n \geq 0\}$
sel. comp., normal play	$\{5n, n \geq 0\} \cup \{5n + 4, n \geq 0\}$
sel. comp., misère play	$\{7n + 1, n \geq 0\} \cup \{7n + 2, n \geq 0\}$
short. sel. comp., normal play	$\{5n, n \geq 0\} \cup \{5n + 4, n \geq 0\}$
short. sel. comp., misère play	$\{1, 2, 8, 9\} \cup \{5n, n \geq 3\} \cup \{5n + 4, n \geq 3\}$

NODE-KAYLES – Open problems

NODE-KAYLES – Open problems

Open Problems.

What about NODE-KAYLES on

- caterpillars?
- subdivided caterpillars?
- other subclasses of trees?
- …

NODE-KAYLES – Open problems

Open Problems.

What about NODE-KAYLES on

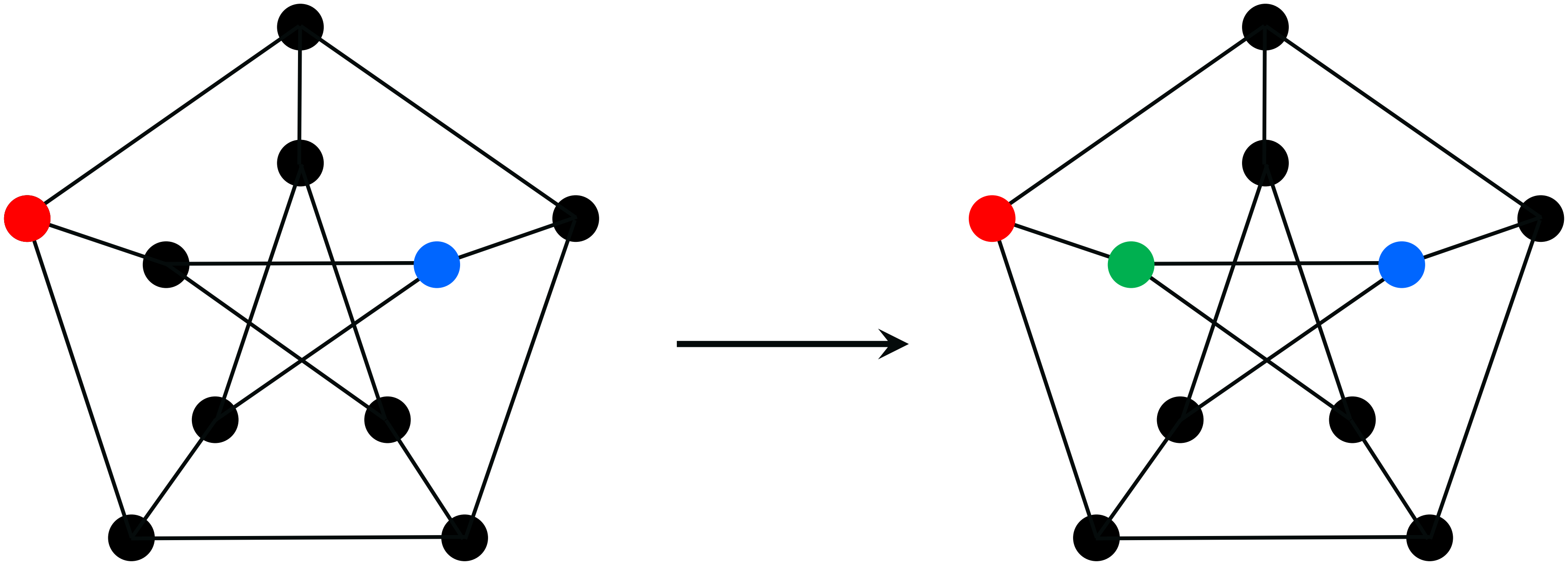
- caterpillars?
- subdivided caterpillars?
- other subclasses of trees?
- …

Suggestion.

Consider compound versions of other combinatorial games on graphs?...

PROPER K-COLOURING

PROPER K-COLOURING



Geography | Nim on graphs | Node-Kayles | k-Colouring | 0.33 game | Timber! | Conclusion

A Maker / Breaker version

Non-combinatorial Graph Colouring Game

A Maker / Breaker version

Non-combinatorial Graph Colouring Game

- Using a set of k colours, on her turn, each player properly colours an uncoloured vertex of a graph G.

A Maker / Breaker version

Non-combinatorial Graph Colouring Game

- Using a set of k colours, on her turn, each player properly colours an uncoloured vertex of a graph G.
- If the whole graph is properly coloured the 1st player wins the game, otherwise the 2nd player wins the game.

A Maker / Breaker version

Non-combinatorial Graph Colouring Game

- Using a set of k colours, on her turn, each player properly colours an uncoloured vertex of a graph G.
- If the whole graph is properly coloured the 1st player wins the game, otherwise the 2nd player wins the game.
- The game chromatic number of G is the least integer k for which the 1st player has a winning strategy.

A Maker / Breaker version

Non-combinatorial Graph Colouring Game

- Using a set of k colours, on her turn, each player properly colours an uncoloured vertex of a graph G.
- If the whole graph is properly coloured the 1st player wins the game, otherwise the 2nd player wins the game.
- The game chromatic number of G is the least integer k for which the 1st player has a winning strategy.

Most intriguing question

- *If the first player wins the game on some graph G using a set of k colours, is it true that she can also win the game on G using a set of k + 1 colours?*

PROPER K-COLOURING (1)

PROPER K-COLOURING

- An undirected graph G and a set of k colours.

PROPER K-COLOURING (1)

PROPER K-COLOURING

- An undirected graph G and a set of k colours.
- On her turn, each player picks an uncoloured vertex and colours it in a proper way (using a colour that does not appear on any of its neighbours).

PROPER K-COLOURING (1)

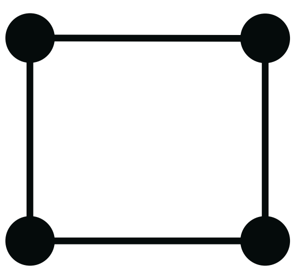
PROPER K-COLOURING

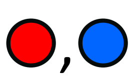
- An undirected graph G and a set of k colours.
- On her turn, each player picks an uncoloured vertex and colours it in a proper way (using a colour that does not appear on any of its neighbours).
- Under normal (resp. misère) convention, the first player unable to play loses (resp. wins) the game.

PROPER K-COLOURING

- An undirected graph G and a set of k colours.
- On her turn, each player picks an uncoloured vertex and colours it in a proper way (using a colour that does not appear on any of its neighbours).
- Under normal (resp. misère) convention, the first player unable to play loses (resp. wins) the game.

G :

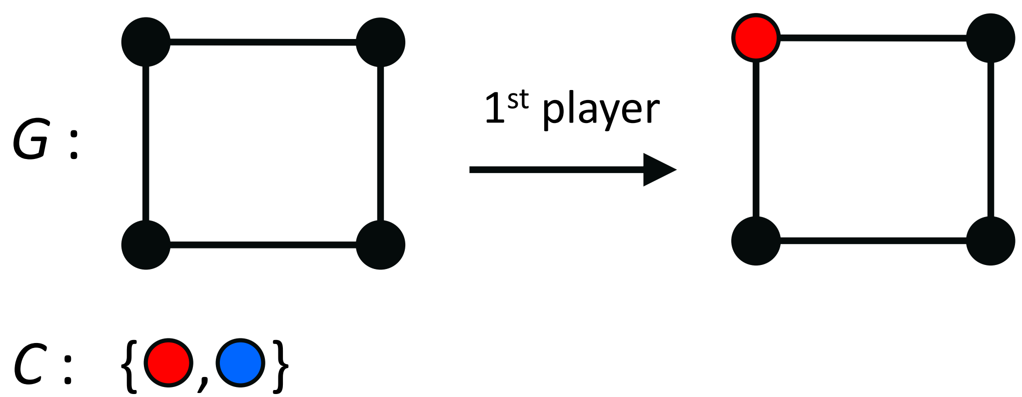


C : {}

PROPER K-COLOURING (1)

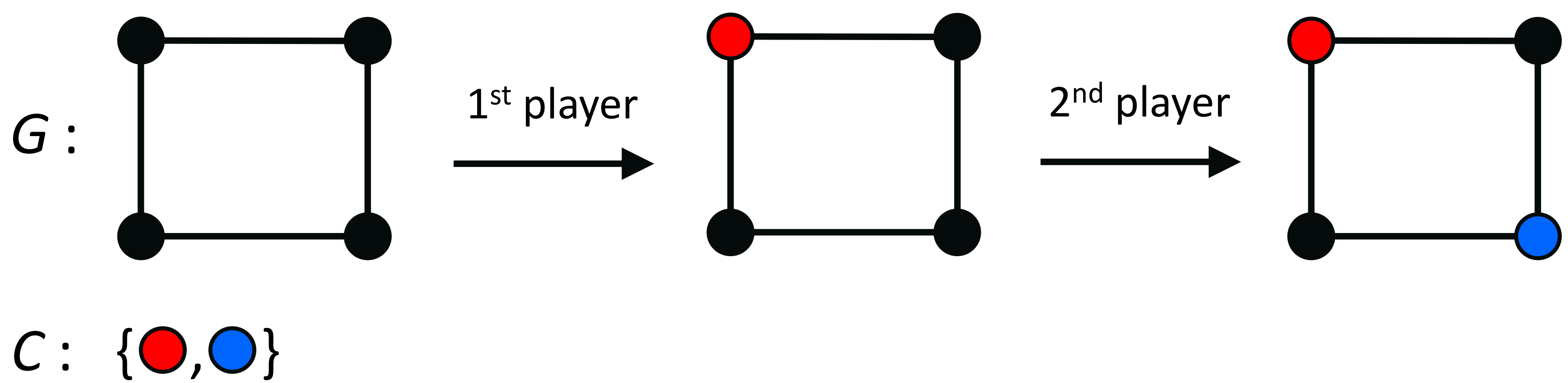
PROPER K-COLOURING

- An undirected graph G and a set of k colours.
- On her turn, each player picks an uncoloured vertex and colours it in a proper way (using a colour that does not appear on any of its neighbours).
- Under normal (resp. misère) convention, the first player unable to play loses (resp. wins) the game.



PROPER K-COLOURING

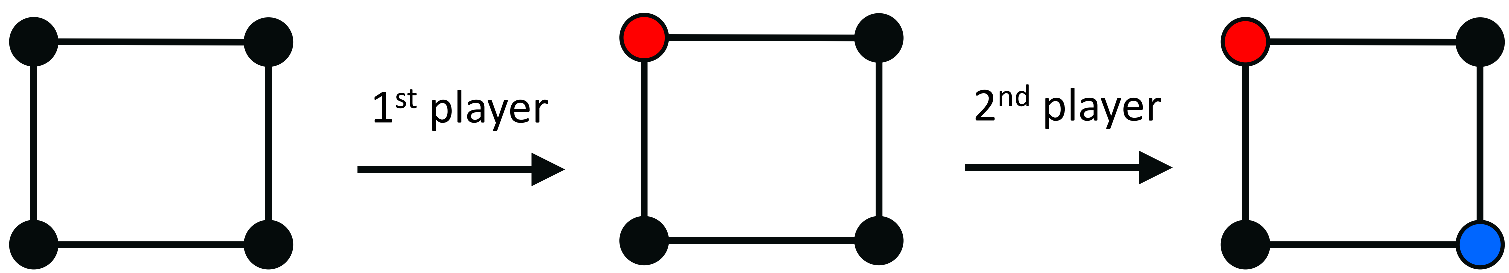
- An undirected graph G and a set of k colours.
- On her turn, each player picks an uncoloured vertex and colours it in a proper way (using a colour that does not appear on any of its neighbours).
- Under normal (resp. misère) convention, the first player unable to play loses (resp. wins) the game.

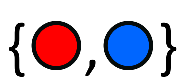


PROPER K-COLOURING

- An undirected graph G and a set of k colours.
- On her turn, each player picks an uncoloured vertex and colours it in a proper way (using a colour that does not appear on any of its neighbours).
- Under normal (resp. misère) convention, the first player unable to play loses (resp. wins) the game.

G :



C : 

End of the game: 2nd player wins!...

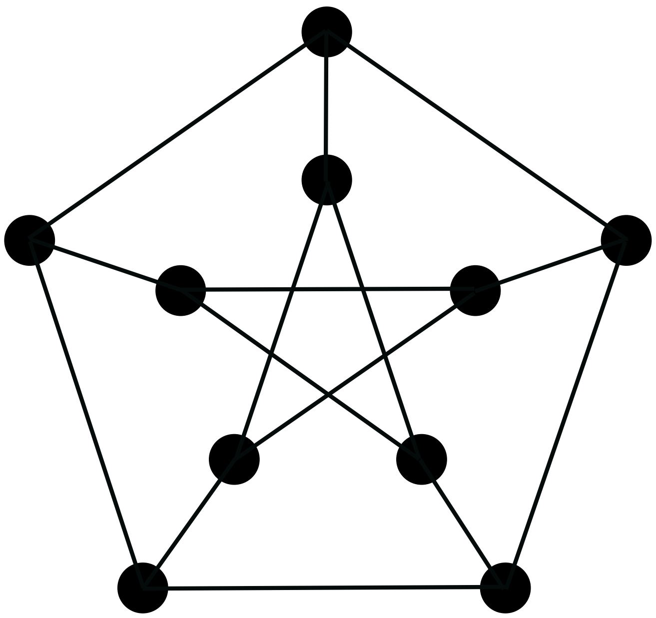
PROPER K-COLOURING (2)

PROPER K-COLOURING

- Playing this game with a unique colour ($k = 1$) is equivalent to playing NODE-KAYLES...

PROPER K-COLOURING

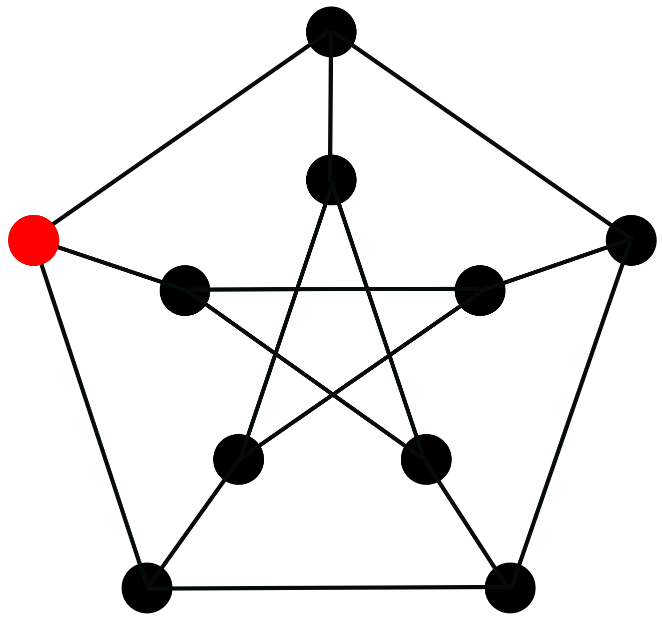
- Playing this game with a unique colour ($k = 1$) is equivalent to playing NODE-KAYLES...



PROPER K-COLOURING (2)

PROPER K-COLOURING

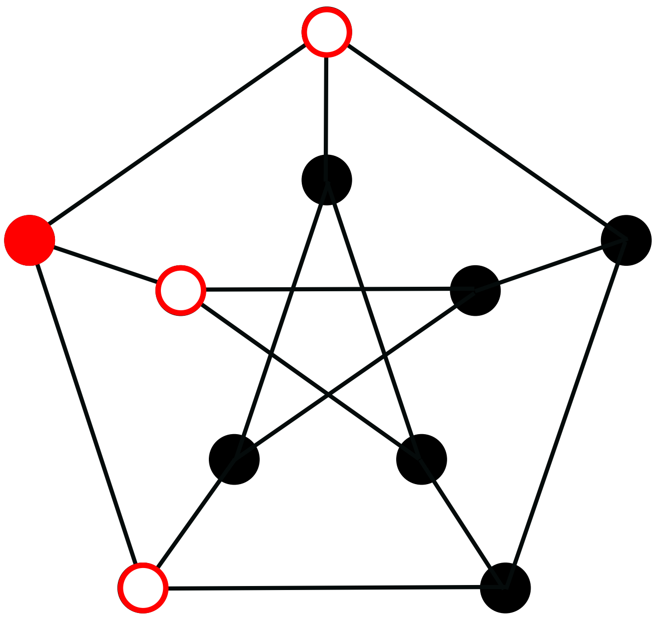
- Playing this game with a unique colour (k = 1) is equivalent to playing NODE-KAYLES...



PROPER K-COLOURING (2)

PROPER K-COLOURING

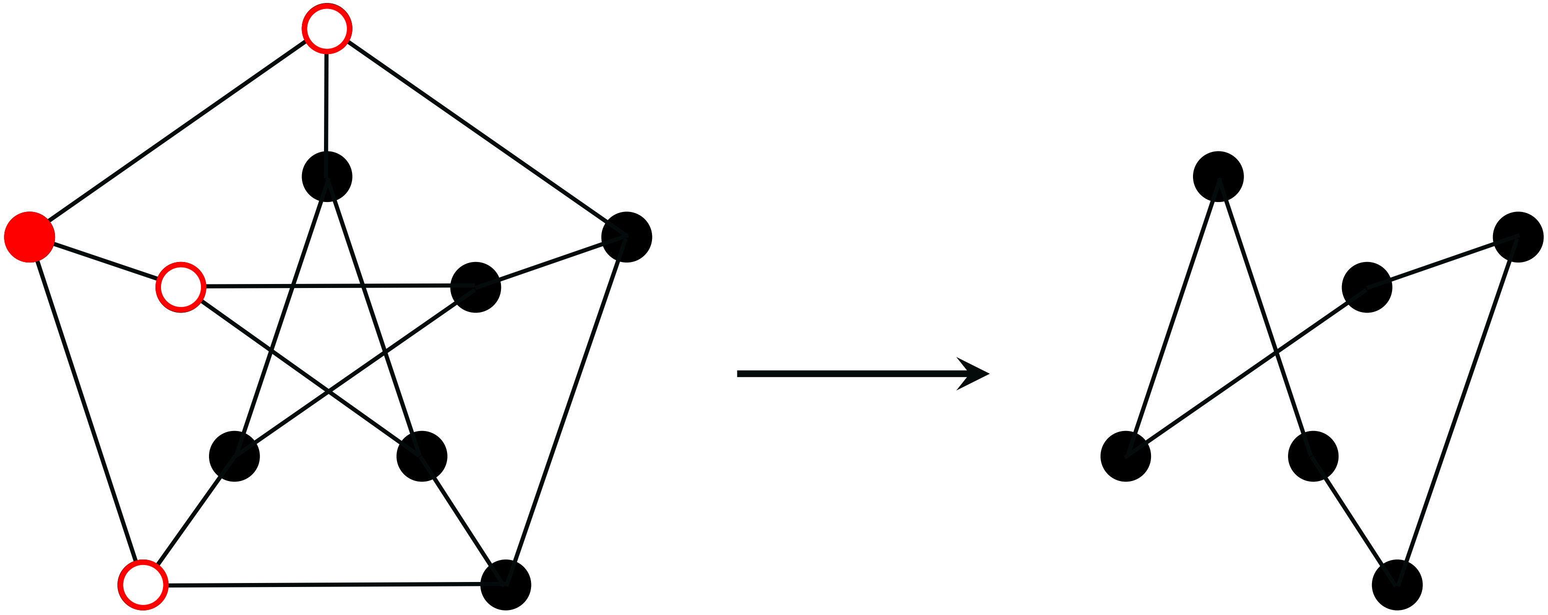
- Playing this game with a unique colour ($k = 1$) is equivalent to playing NODE-KAYLES...



PROPER K-COLOURING (2)

PROPER K-COLOURING

- Playing this game with a unique colour ($k = 1$) is equivalent to playing NODE-KAYLES...



NODE-KAYLES VS. PROPER K-COLOURING

Observation.

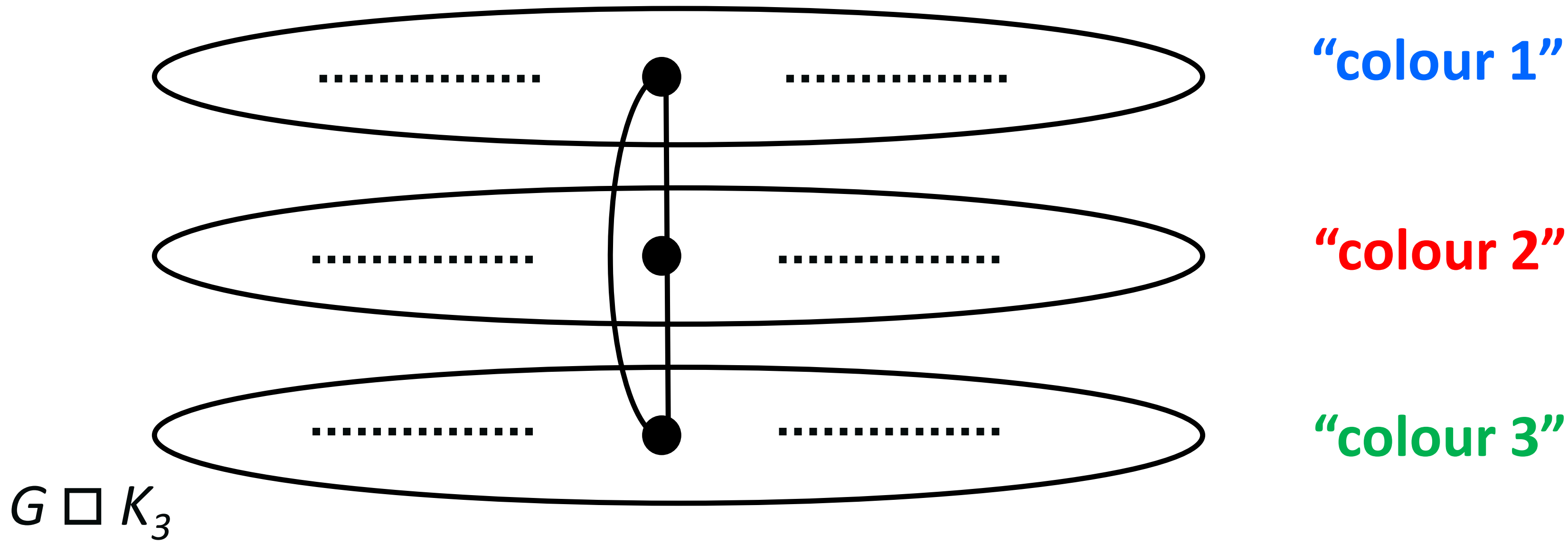
Playing PROPER K-COLOURING on G is equivalent to playing NODE-KAYLES on $G \,\square\, K_k$.

NODE-KAYLES VS. PROPER K-COLOURING

Observation.

Playing PROPER K-COLOURING on G is equivalent to playing NODE-KAYLES on $G \square K_k$.

Example with k = 3:

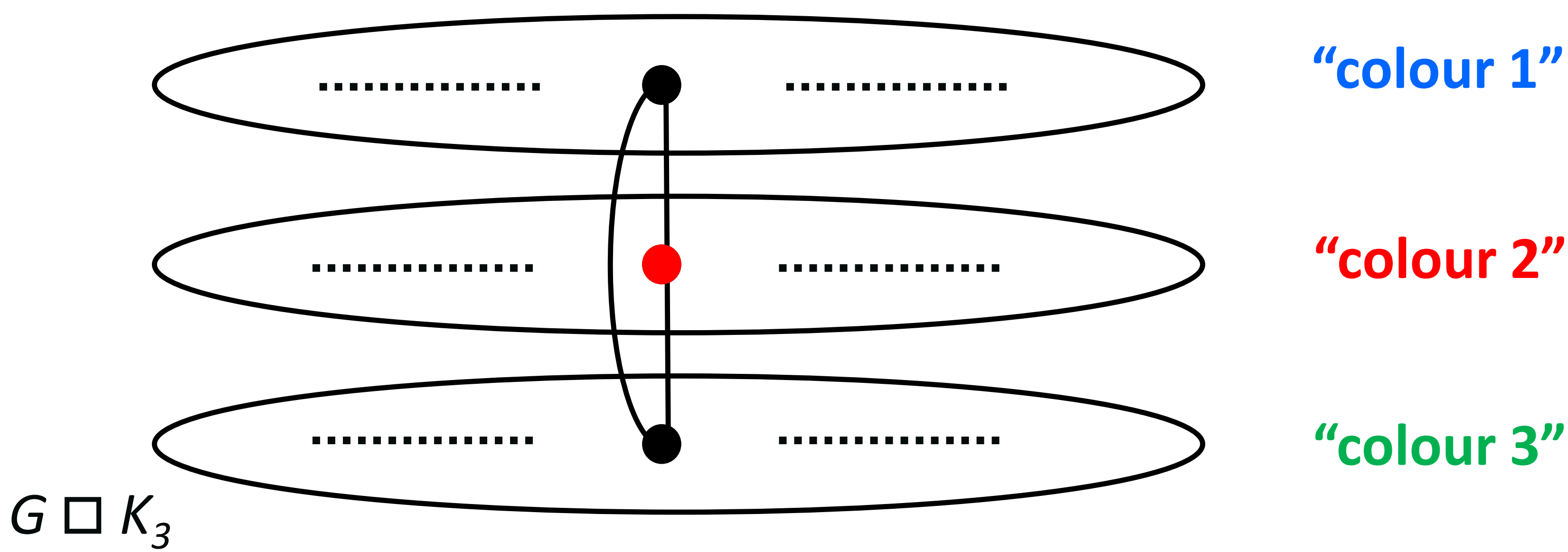


NODE-KAYLES VS. PROPER K-COLOURING

Observation.

Playing PROPER K-COLOURING *on G is equivalent to playing* NODE-KAYLES *on* $G \square K_k$.

Example with k = 3:

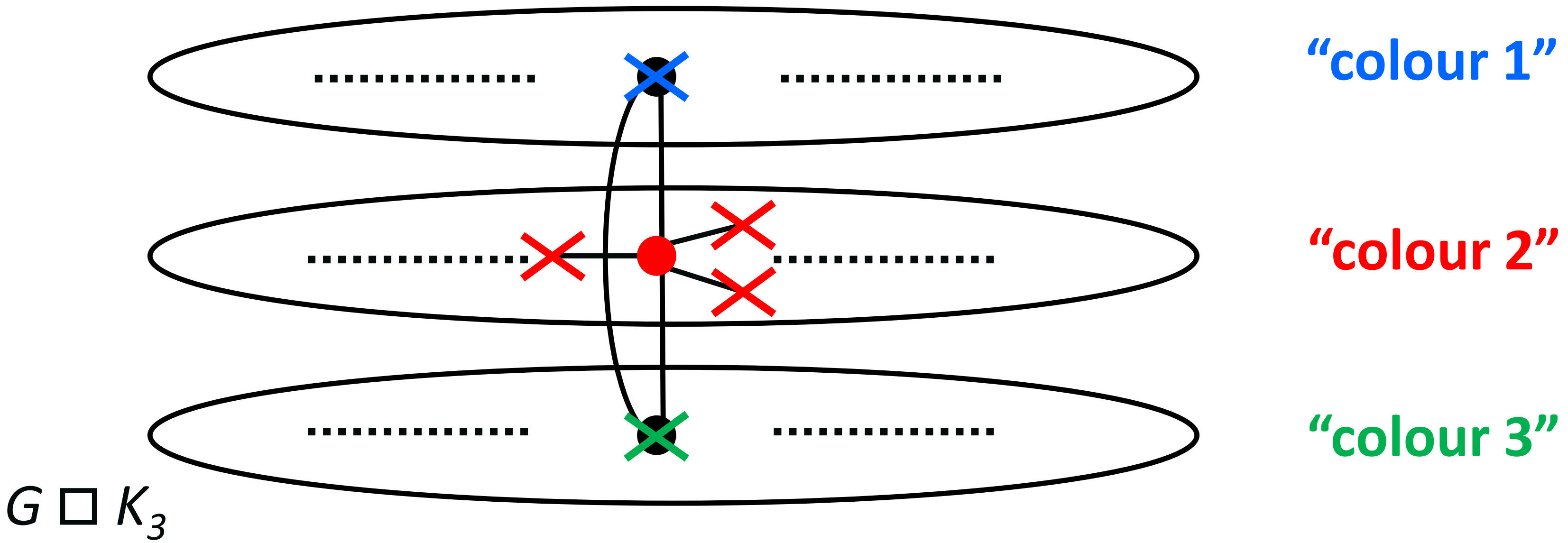


NODE-KAYLES VS. PROPER K-COLOURING

Observation.

Playing PROPER K-COLOURING *on G is equivalent to playing* NODE-KAYLES *on* $G \square K_k$.

Example with k = 3:



PROPER K-COLOURING

Complexity

Theorem [BEAULIEU, BURKE, DUCHÊNE, 2013].

For every integer k ≥ 1, determining whether a position of PROPER K-COLOURING is a winning position or not is PSPACE-complete.

PROPER K-COLOURING

Complexity

Theorem [BEAULIEU, BURKE, DUCHÊNE, 2013].
For every integer k ≥ 1, determining whether a position of PROPER K-COLOURING is a winning position or not is PSPACE-complete.

Sprague-Grundy values [BEAULIEU, BURKE, DUCHÊNE, 2013]

- Sufficient conditions for a position to be a winning or loosing position are known for d-dimensional grids when all dimensions are odd, complete d-ary trees when d is odd...
- PROPER K-COLOURING is solved for paths and cycles

PROPER K-COLOURING

Open Problems.

PROPER K-COLOURING

Open Problems.

- What about PROPER K-COLOURING on caterpillars? on complete k-ary trees with k even? on trees?...

PROPER K-COLOURING

Open Problems.

- What about PROPER K-COLOURING on caterpillars? on complete k-ary trees with k even? on trees?...
- Other combinatorial games, based on other types of colourings? (e.g. acyclic, distance-two, or edge-colourings...)

The (partisan) games of COL and SNORT

The (partisan) games of COL and SNORT

The game of COL (attributed to COLIN VOUT)

- A partisan version of the k-COLOURING GAME.
- The first player uses only colour RED, while the second player uses only colour BLUE.

The (partisan) games of COL and SNORT

The game of COL (attributed to COLIN VOUT)

- A partisan version of the k-COLOURING GAME.
- The first player uses only colour RED, while the second player uses only colour BLUE.
- *The computational complexity of COL seems to be unknown...*

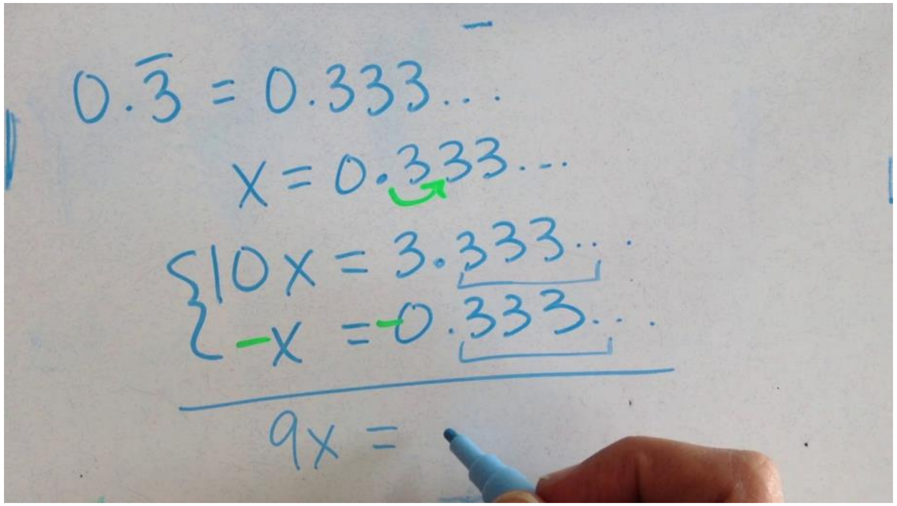
The (partisan) games of COL and SNORT

The game of COL (attributed to COLIN VOUT)

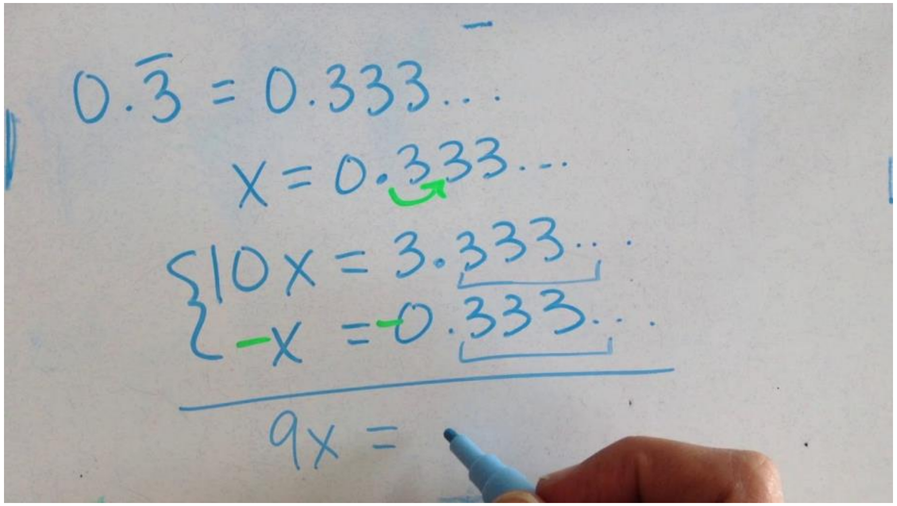
- A partisan version of the K-COLOURING GAME.
- The first player uses only colour RED, while the second player uses only colour BLUE.
- *The computational complexity of COL seems to be unknown...*

The game of SNORT (proposed by SIMON P. NORTON)

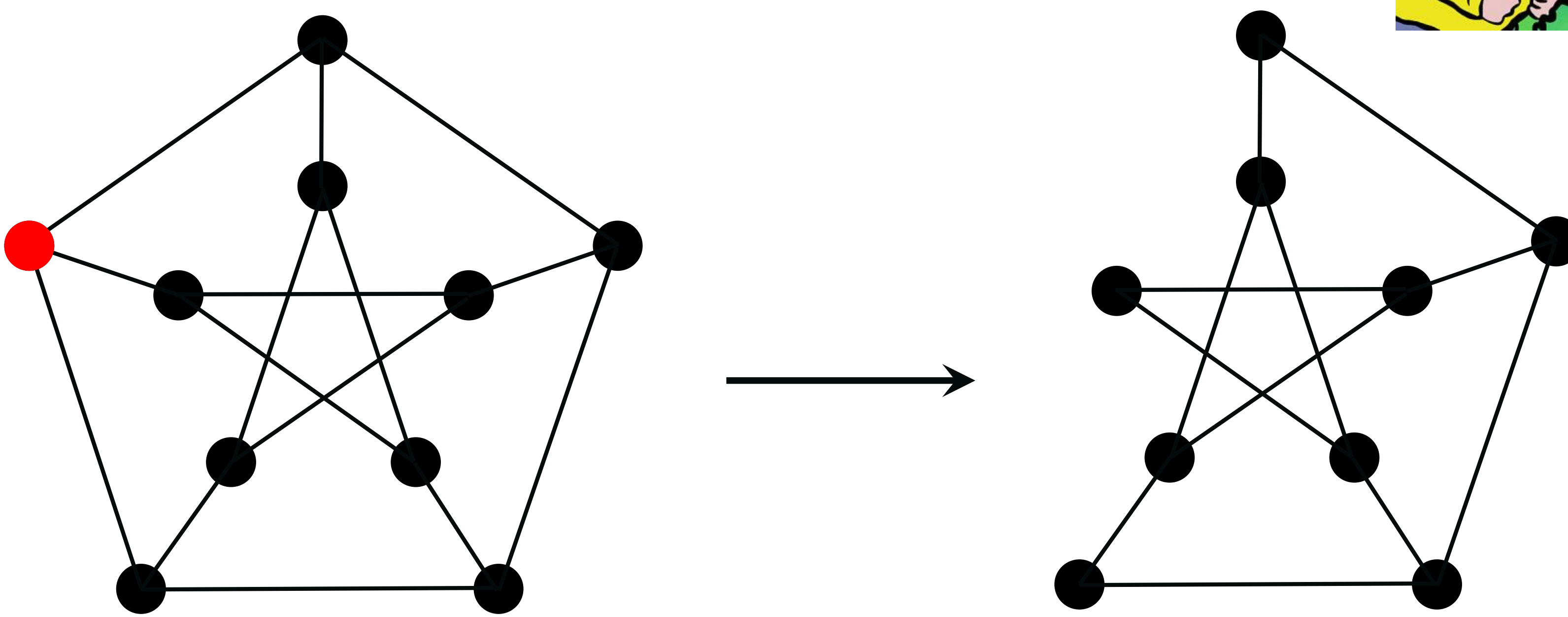
- Same as COL, except that adjacent vertices cannot get distinct colours (a.k.a. CATS & DOGS)...
- Determining the outcome of a SNORT position is PSPACE-complete.


$0.\overline{3} = 0.333\ldots$
$x = 0.333\ldots$
$10x = 3.333\ldots$
$-x = -0.333\ldots$
$9x =$

The 0.33 Game



The 0.33 Game



Octal games (Take-and-Break games)

Octal games

- These games are played on heaps of tokens
- On her turn, each player chooses one heap, and remove $k > 0$ tokens from this heap, according to the rules of the game

Octal games (Take-and-Break games)

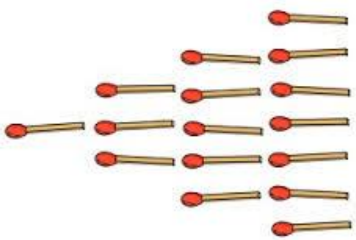
Octal games

- These games are played on heaps of tokens
- On her turn, each player chooses one heap, and remove $k > 0$ tokens from this heap, according to the rules of the game
- These rules are encoded by a sequence $\mathbf{0.d_1d_2d_3...}$ of octal digits, describing the moves that are allowed on a heap:
  - if you can take j tokens and leave no heap, set $\mathbf{J_0 = 1}$
  - if you can take j tokens and leave one heap, set $\mathbf{J_1 = 2}$
  - if you can take j tokens and leave two heaps, set $\mathbf{J_2 = 4}$
  - then let $\mathbf{d_j = J_0 + J_1 + J_2}$

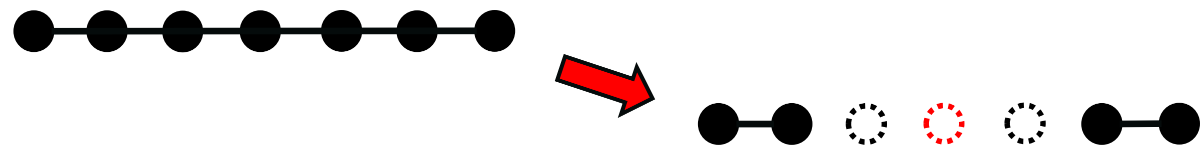
Octal games (Take-and-Break games)

Octal games

- These games are played on heaps of tokens
- On her turn, each player chooses one heap, and remove $k > 0$ tokens from this heap, according to the rules of the game
- These rules are encoded by a sequence $\mathbf{0.d_1d_2d_3...}$ of octal digits, describing the moves that are allowed on a heap:
  - if you can take j tokens and leave no heap, set $\mathbf{J_0 = 1}$
  - if you can take j tokens and leave one heap, set $\mathbf{J_1 = 2}$
  - if you can take j tokens and leave two heaps, set $\mathbf{J_2 = 4}$
  - then let $\mathbf{d_j = J_0 + J_1 + J_2}$
- The ordinary game of NIM is 0.33333...



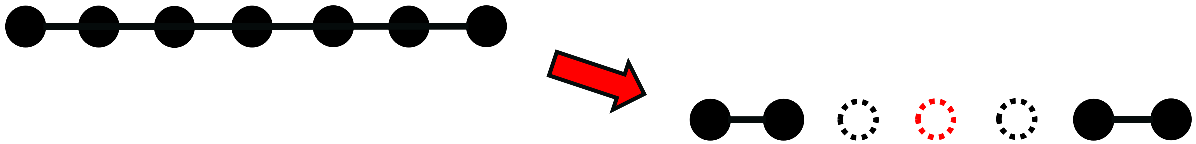
Octal games: DAWSON’S CHESS



DAWSON’S CHESS

- Played on a path of order n (a heap of n tokens)
- On her turn, each player picks one vertex and deletes its closed neighbourhood

Octal games: DAWSON'S CHESS



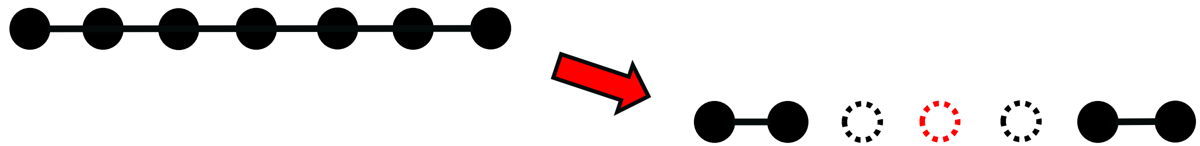
DAWSON'S CHESS

- Played on a path of order n (a heap of n tokens)
- On her turn, each player picks one vertex and deletes its closed neighbourhood

Octal encoding of DAWSON'S CHESS

- You can delete one vertex iff the graph is P_1, and thus $d_1 = 1$

Octal games: DAWSON'S CHESS



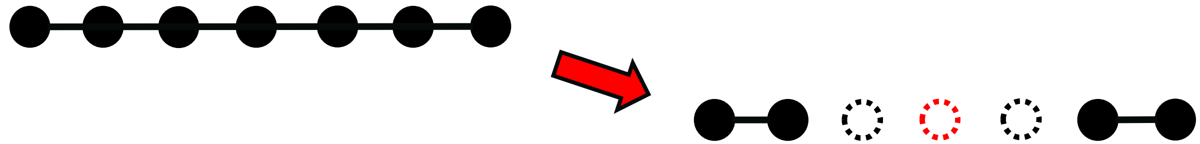
DAWSON'S CHESS

- Played on a path of order n (a heap of n tokens)
- On her turn, each player picks one vertex and deletes its closed neighbourhood

Octal encoding of DAWSON'S CHESS

- You can delete one vertex iff the graph is P_1, and thus $d_1 = 1$
- You can delete two adjacent vertices iff at least one of them is an endpoint, and thus $d_2 = 1 + 2 = 3$

Octal games: DAWSON'S CHESS



DAWSON'S CHESS

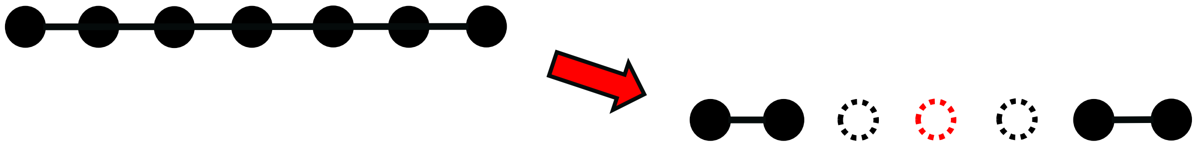
- Played on a path of order n (a heap of n tokens)
- On her turn, each player picks one vertex and deletes its closed neighbourhood

Octal encoding of DAWSON'S CHESS

- You can delete one vertex iff the graph is P_1, and thus $d_1 = 1$
- You can delete two adjacent vertices iff at least one of them is an endpoint, and thus $d_2 = 1 + 2 = 3$
- You can always delete three adjacent vertices, and thus $d_3 = 1 + 2 + 4 = 7$

Octal games: DAWSON'S CHESS

DAWSON'S CHESS



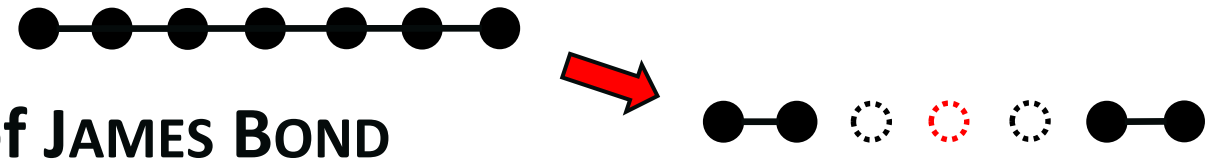
- Played on a path of order n (a heap of n tokens)
- On her turn, each player picks one vertex and deletes its closed neighbourhood

Octal encoding of DAWSON'S CHESS

- You can delete one vertex iff the graph is P_1, and thus $d_1 = 1$
- You can delete two adjacent vertices iff at least one of them is an endpoint, and thus $d_2 = 1 + 2 = 3$
- You can always delete three adjacent vertices, and thus $d_3 = 1 + 2 + 4 = 7$
- Therefore, DAWSON'S CHESS is the octal game **0.137**

Octal games: JAMES BOND ☺

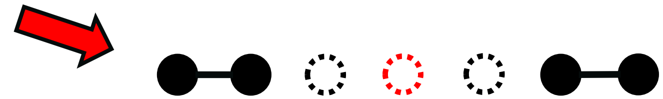
Octal games: JAMES BOND ☺



The game of JAMES BOND

- Played on a path of order n (a heap of n tokens)
- On her turn, each player deletes three adjacent vertices

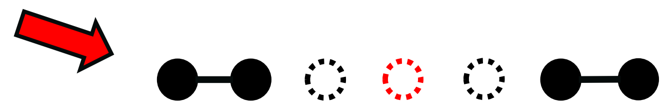
Octal games: JAMES BOND ☺



The game of JAMES BOND

- Played on a path of order n (a heap of n tokens)
- On her turn, each player deletes three adjacent vertices
- The octal encoding of this game is... **0.007**

Octal games: JAMES BOND ☺



The game of JAMES BOND

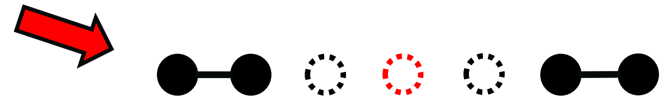
- Played on a path of order n (a heap of n tokens)
- On her turn, each player deletes three adjacent vertices
- The octal encoding of this game is... **0.007**

Sprague-Grundy sequence of JAMES BOND

- About **2^{28}** values have been computed :

0 0 0 1 1 1 2 2 0 3 3 1 1 1 0 4 ...

Octal games: JAMES BOND ☺



The game of JAMES BOND

- Played on a path of order n (a heap of n tokens)
- On her turn, each player deletes three adjacent vertices
- The octal encoding of this game is... **0.007**

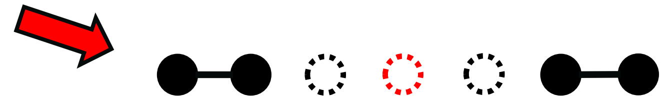
Sprague-Grundy sequence of JAMES BOND

- About **2^{28}** values have been computed :

  0 0 0 1 1 1 2 2 0 3 3 1 1 1 0 4 ...

- The ultimate periodicity of this sequence is conjectured

Octal games: JAMES BOND ☺



The game of JAMES BOND

- Played on a path of order n (a heap of n tokens)
- On her turn, each player deletes three adjacent vertices
- The octal encoding of this game is... **0.007**

Sprague-Grundy sequence of JAMES BOND

- About **2^{28}** values have been computed :

  0 0 0 1 1 1 2 2 0 3 3 1 1 1 0 4 ...
- The ultimate periodicity of this sequence is conjectured

> **Conjecture [GUY, 1996].** *The Sprague-Grundy sequence of every finite octal game is ultimately periodic.*

Octal games on graphs: 0.33

Octal games on graphs: 0.33

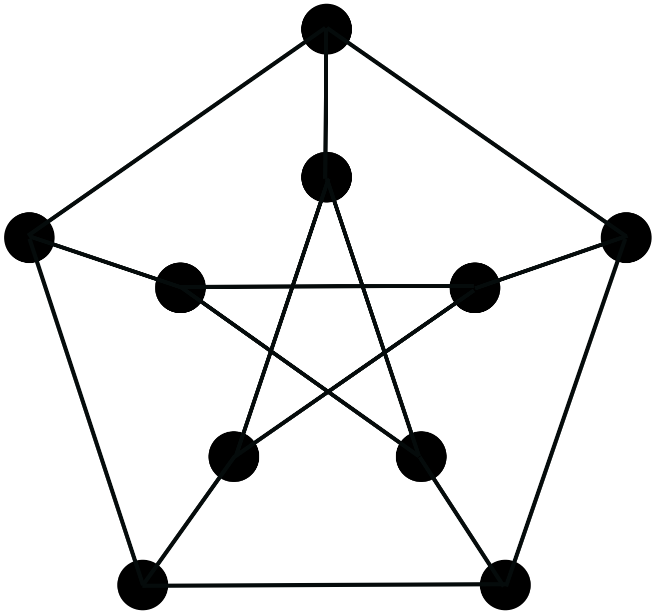
The game 0.33 on graphs

- Played on an undirected connected graph
- On her turn, each player deletes one vertex, or two adjacent vertices, provided that the remaining graph is still connected

Octal games on graphs: 0.33

The game 0.33 on graphs

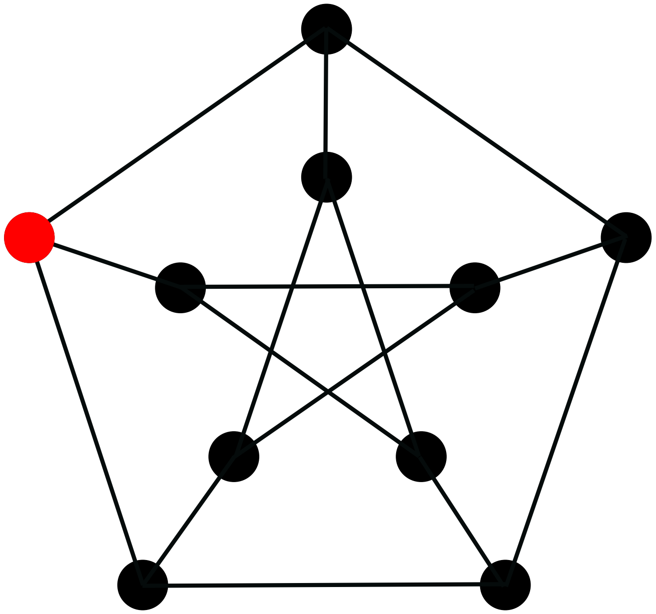
- Played on an undirected connected graph
- On her turn, each player deletes one vertex, or two adjacent vertices, provided that the remaining graph is still connected



Octal games on graphs: 0.33

The game 0.33 on graphs

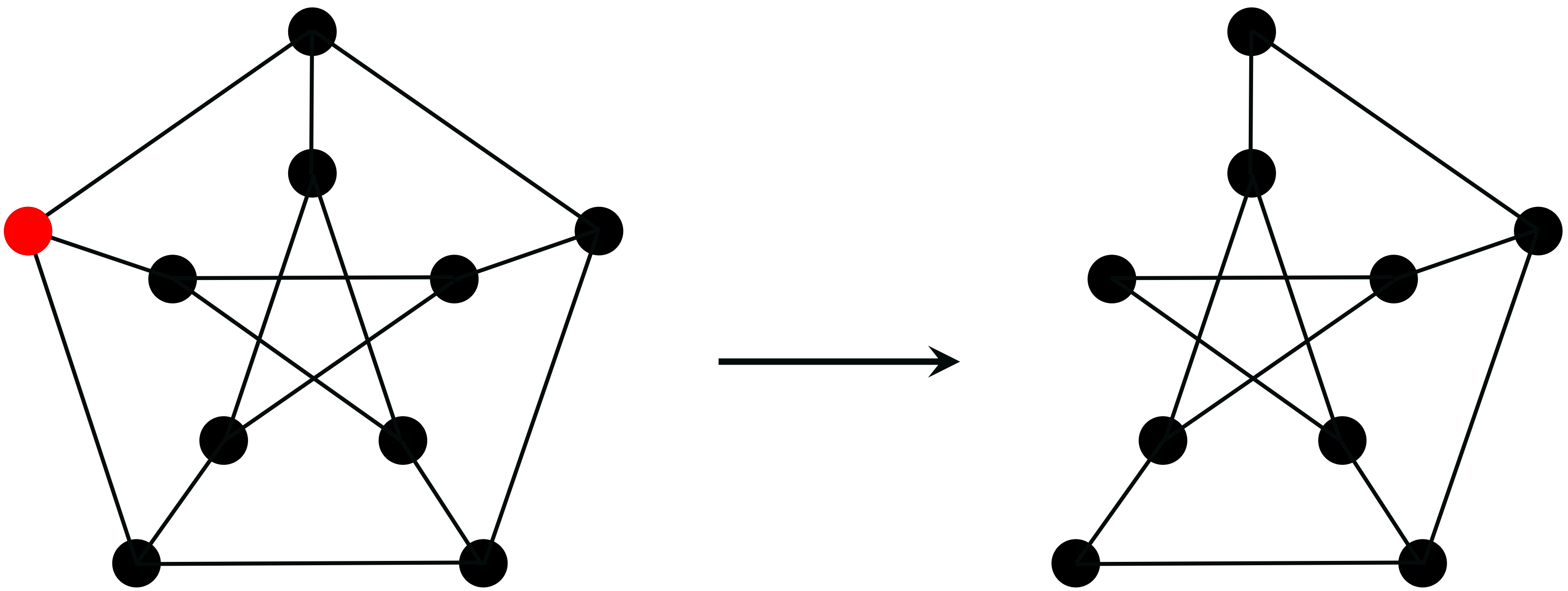
- Played on an undirected connected graph
- On her turn, each player deletes one vertex, or two adjacent vertices, provided that the remaining graph is still connected



Octal games on graphs: 0.33

The game 0.33 on graphs

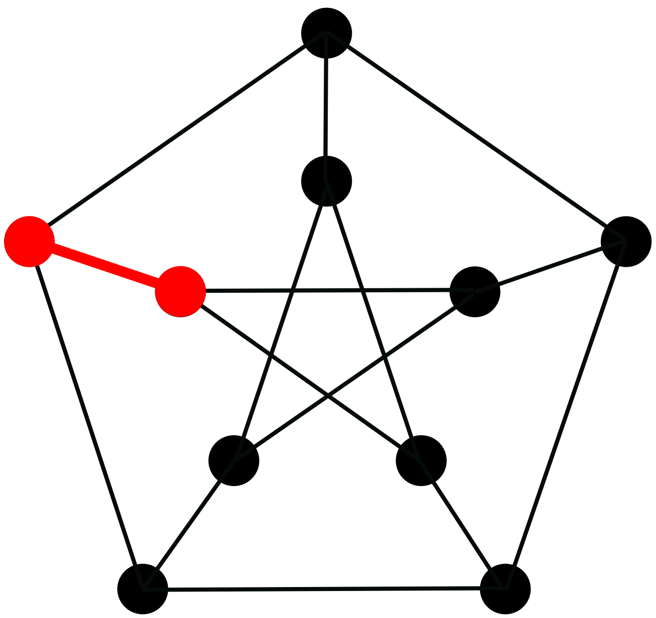
- Played on an undirected connected graph
- On her turn, each player deletes one vertex, or two adjacent vertices, provided that the remaining graph is still connected



Octal games on graphs: 0.33

The game 0.33 on graphs

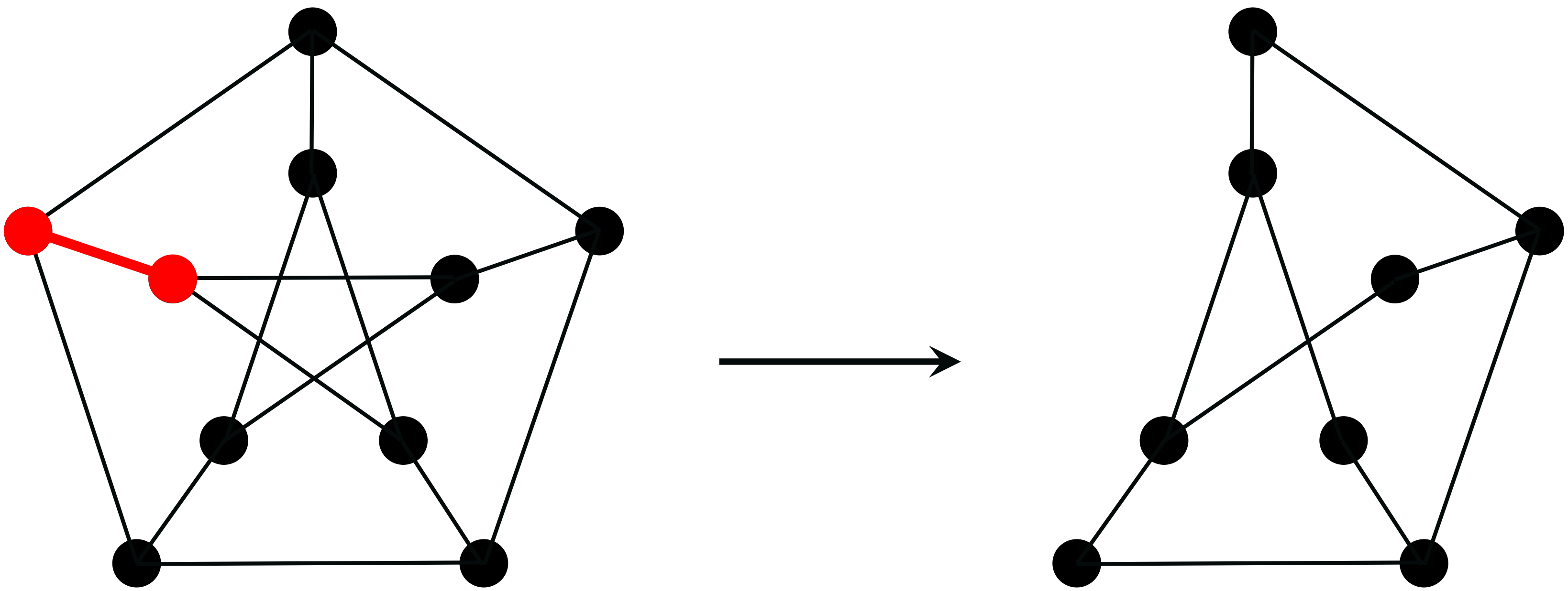
- Played on an undirected connected graph
- On her turn, each player deletes one vertex, or two adjacent vertices, provided that the remaining graph is still connected



Octal games on graphs: 0.33

The game 0.33 on graphs

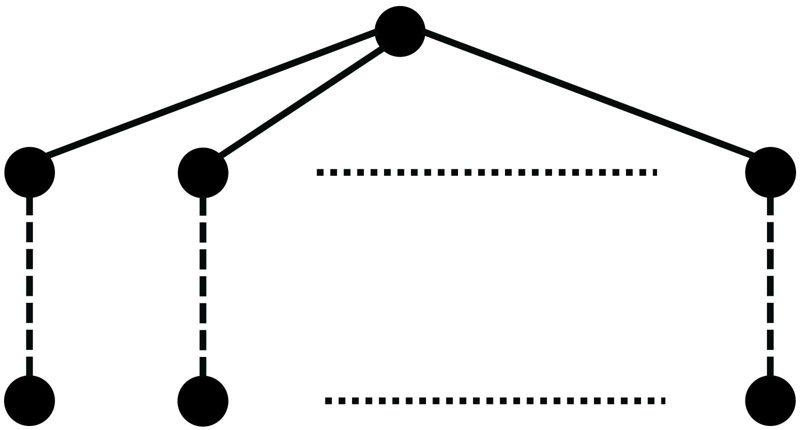
- Played on an undirected connected graph
- On her turn, each player deletes one vertex, or two adjacent vertices, provided that the remaining graph is still connected



0.33 on subdivided stars (1)

Subdivided stars

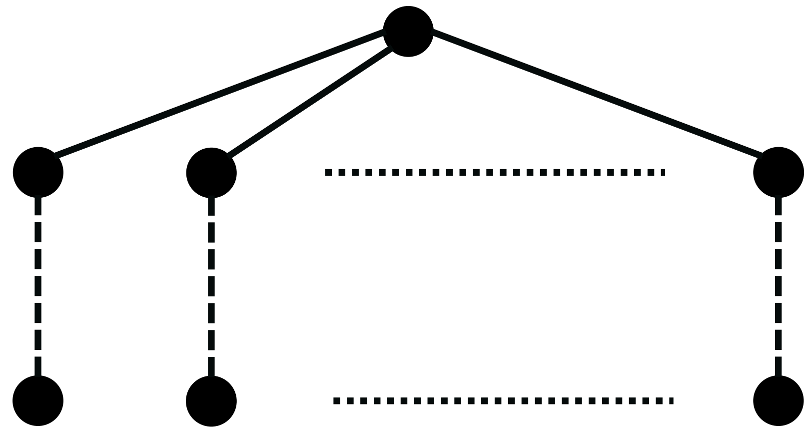
$S(p_1, p_2, \ldots, p_k)$:



0.33 on subdivided stars (1)

Subdivided stars

$S(p_1, p_2, \ldots, p_k)$:



Sprague-Grundy values: reduction

Theorem [Beaudou *et al.*, 2018].

For every subdivided star $S(p_1, p_2, \ldots, p_k)$, we have

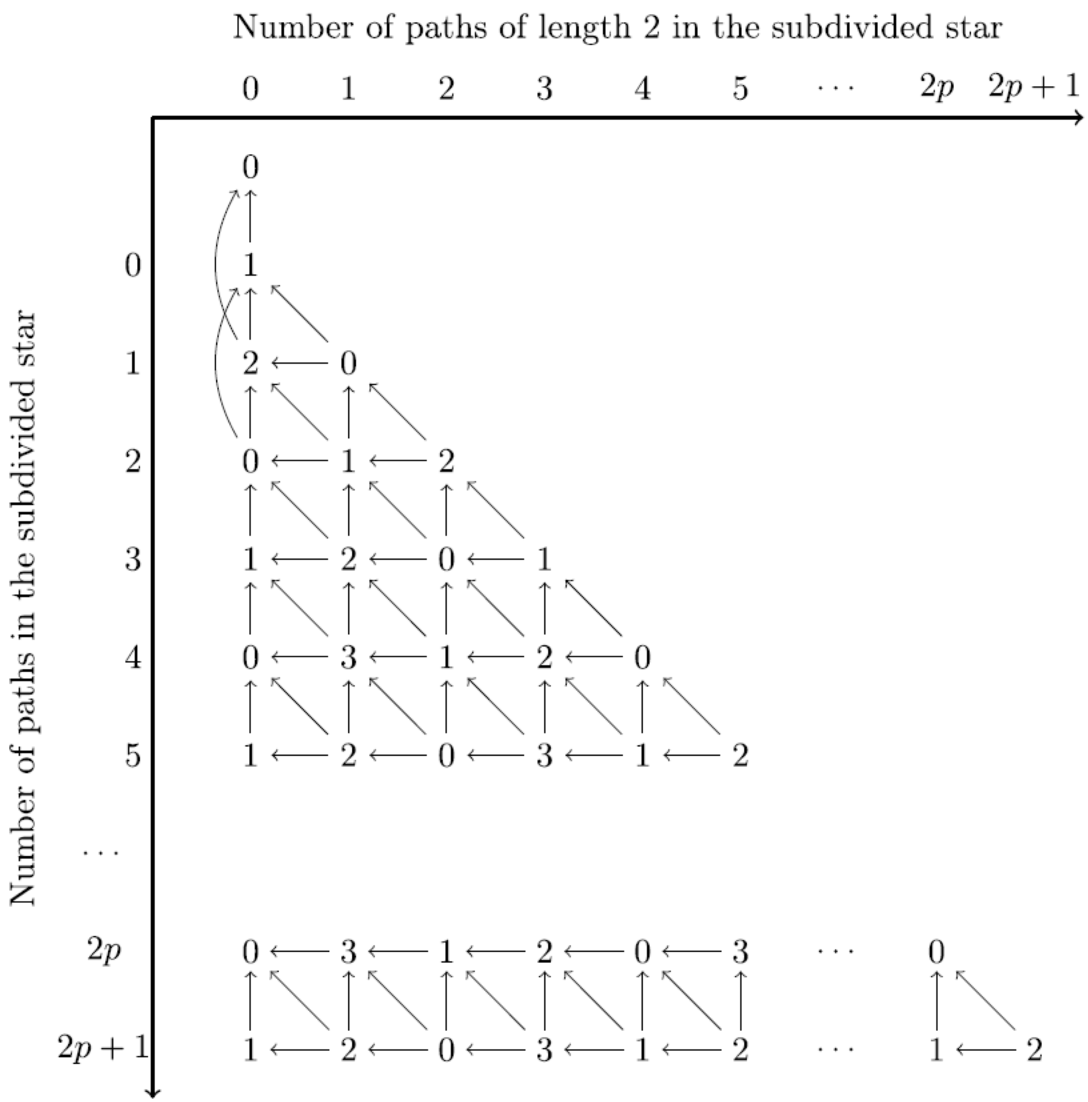
$$\sigma(\, S(p_1, p_2, \ldots, p_k)\,) = \sigma(\, S(p_1 \bmod 3, p_2 \bmod 3, \ldots, p_k \bmod 3\,).$$

0.33 on subdivided stars (2)

Sprague-Grundy values

All the Sprague-Grundy values are in {0,...,3}.

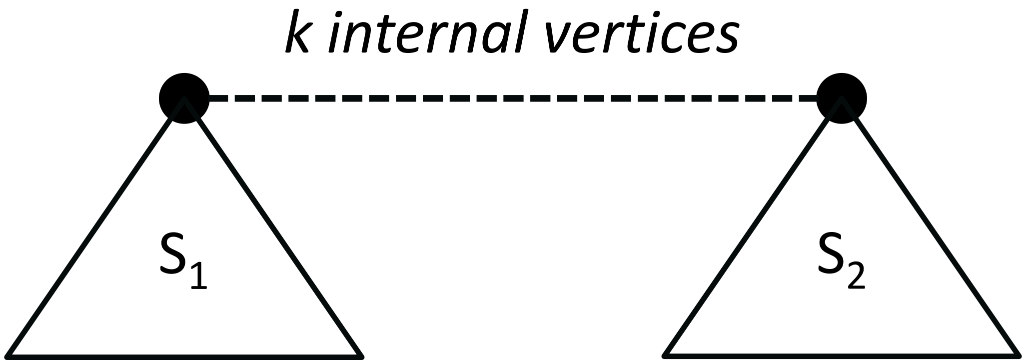
These values can be computed, according to the number of paths and the number of paths of length 2. [BEAUDOU *et al.*, 2018]



0.33 on subdivided bistars

Subdivided bistars

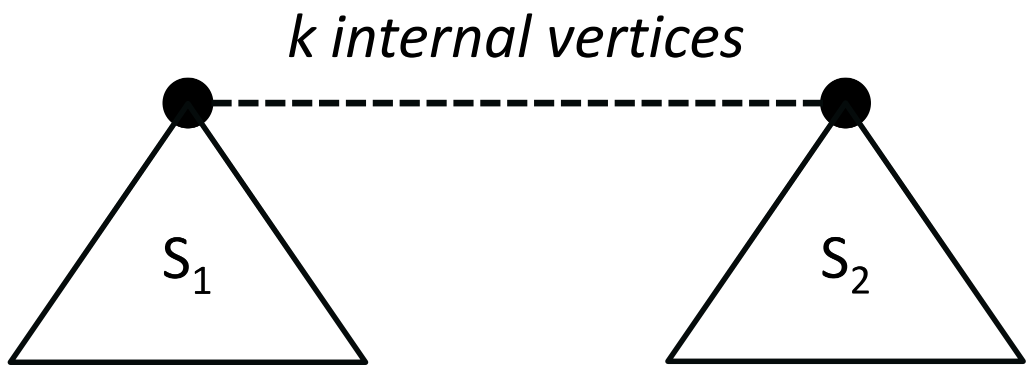
S_1-k-S_2



0.33 on subdivided bistars

Subdivided bistars

S_1-k-S_2



Sprague-Grundy values

Theorem [BEAUDOU *et al.*, 2018].

For every subdivided bistar S_1-k-S_2, we have

$$\sigma(\,S_1\text{-}k\text{-}S_2\,) = f(\sigma(S_1),\, \sigma(S_2)).$$

0.33: Open problems

0.33: Open problems

Open Problem.

- What about 0.33 on trees?
- Is the Sprague-Grundy value of trees bounded?
- What about the misère version?

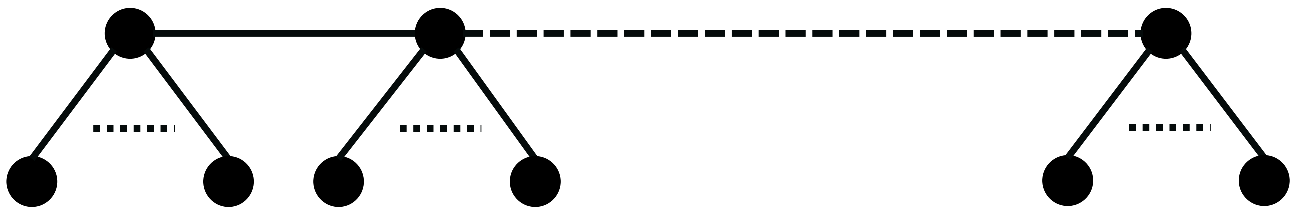
0.33: Open problems

Open Problem.

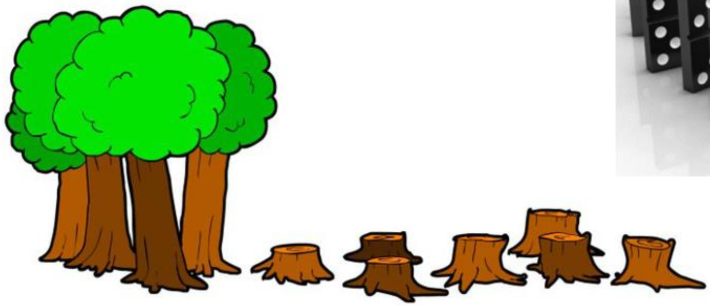
- What about 0.33 on trees?
- Is the Sprague-Grundy value of trees bounded?
- What about the misère version?

Conjecture [Beaudou *et al.*, 2018].

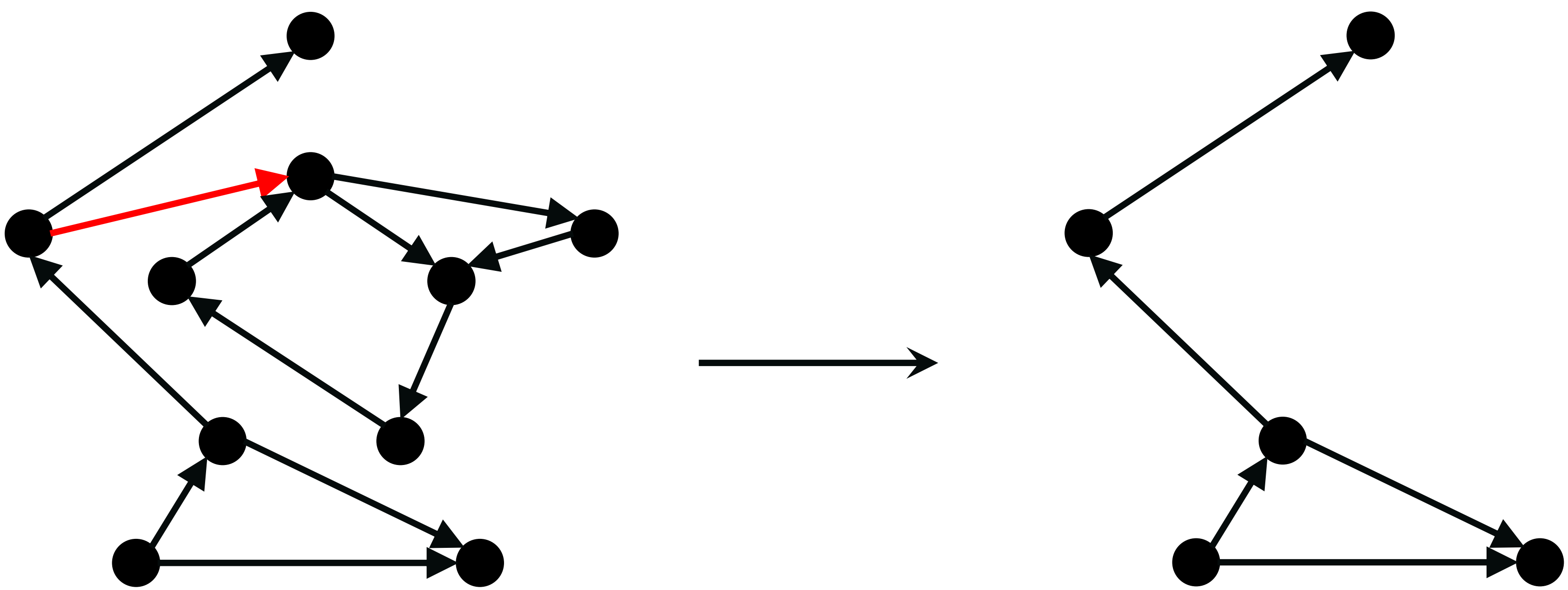
For every integer n, there exists a caterpillar CT with $\sigma(CT) = n$.



TIMBER!



TIMBER!



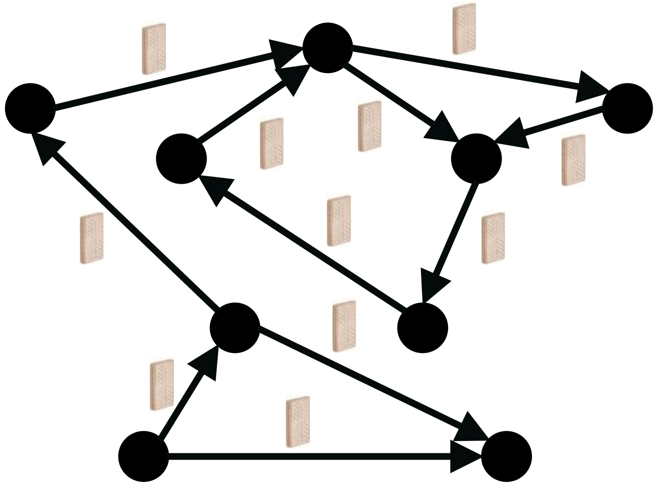
TIMBER! (1)

TIMBER! [A graph version of the game TOPPLING PEAKS]

TIMBER! (1)

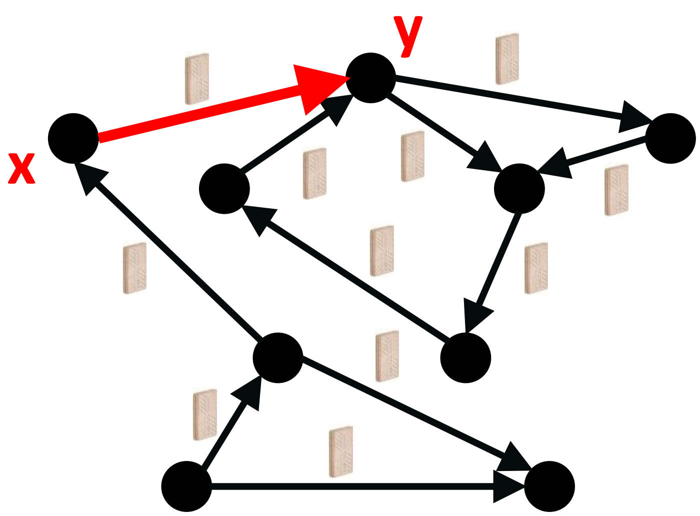
TIMBER! [A graph version of the game TOPPLING PEAKS]

- the game is played on a digraph, all of whose arcs are equipped with a domino,



TIMBER! [A graph version of the game TOPPLING PEAKS]

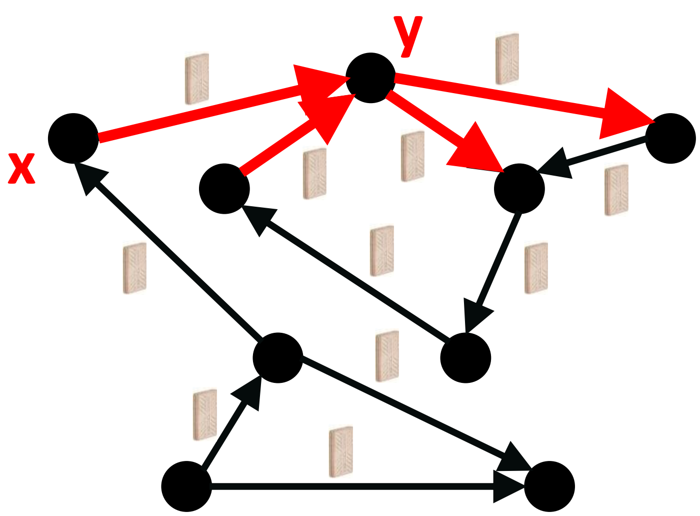
- the game is played on a digraph, all of whose arcs are equipped with a domino,
- on her turn, each player chooses one arc, say **xy**, and topples its domino in the direction of y...



TIMBER! (1)

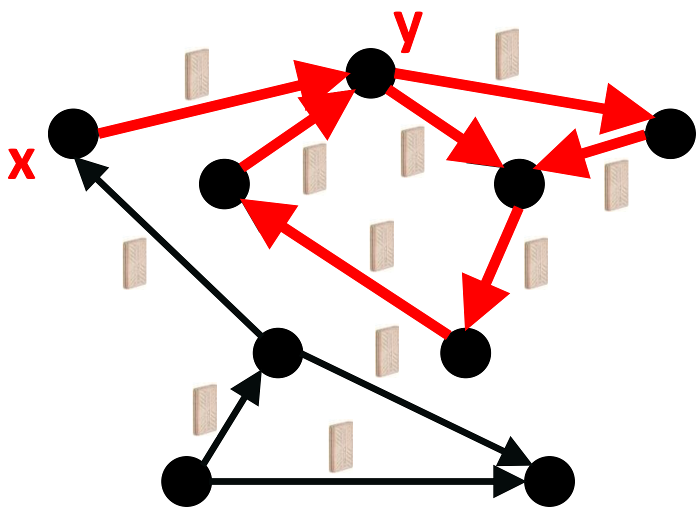
TIMBER! [A graph version of the game TOPPLING PEAKS]

- the game is played on a digraph, all of whose arcs are equipped with a domino,
- on her turn, each player chooses one arc, say **xy**, and topples its domino in the direction of y...
- ...and the toppling process propagates (the orientation does not matter).



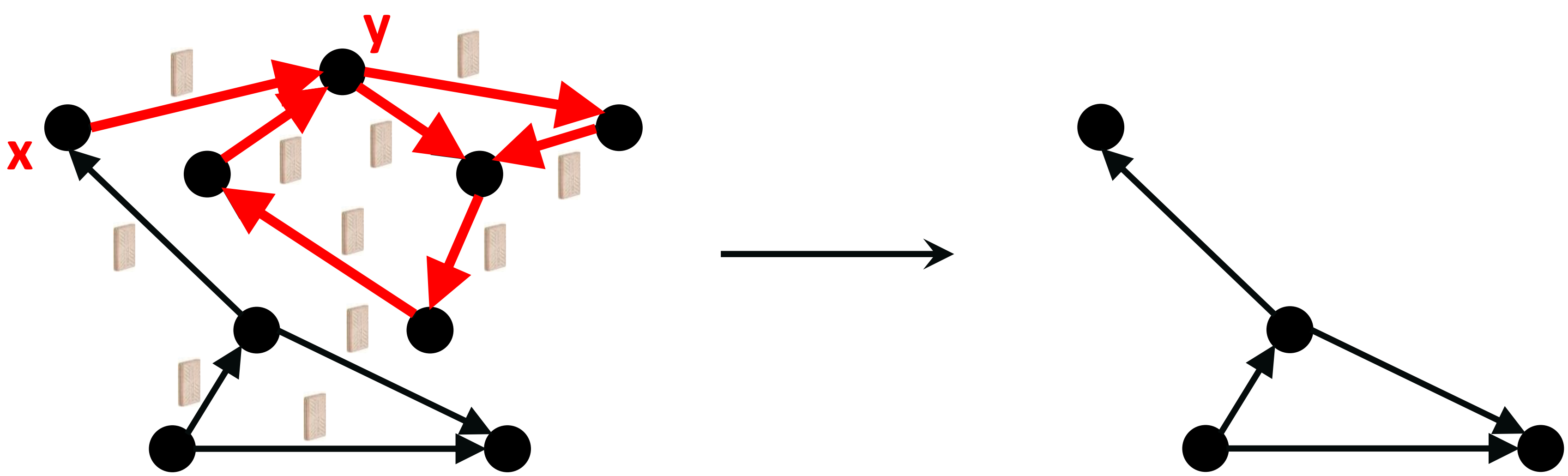
TIMBER! [A graph version of the game TOPPLING PEAKS]

- the game is played on a digraph, all of whose arcs are equipped with a domino,
- on her turn, each player chooses one arc, say **xy**, and topples its domino in the direction of y...
- ...and the toppling process propagates (the orientation does not matter).



TIMBER! [A graph version of the game TOPPLING PEAKS]

- the game is played on a digraph, all of whose arcs are equipped with a domino,
- on her turn, each player chooses one arc, say **xy**, and topples its domino in the direction of y...
- ...and the toppling process propagates (the orientation does not matter).



TIMBER! (2)

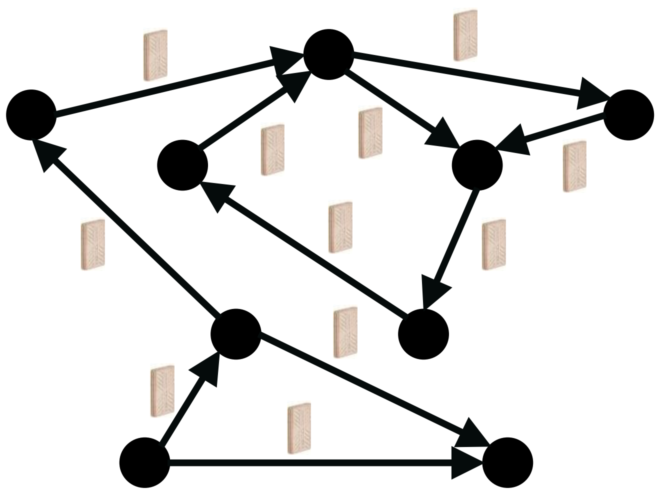
Observation.

If the underlying (undirected) graph contains a 2-connected subgraph of order at least 2, then the first player wins the game.

TIMBER! (2)

Observation.

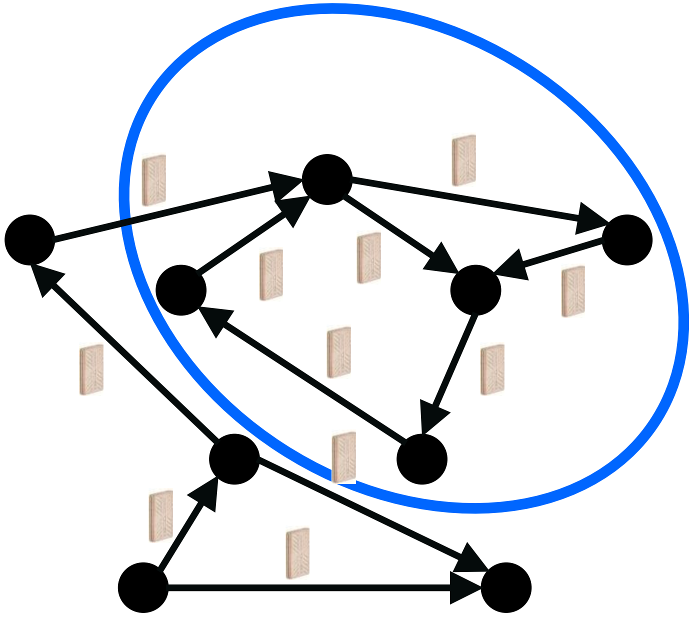
If the underlying (undirected) graph contains a 2-connected subgraph of order at least 2, then the first player wins the game.



TIMBER! (2)

Observation.

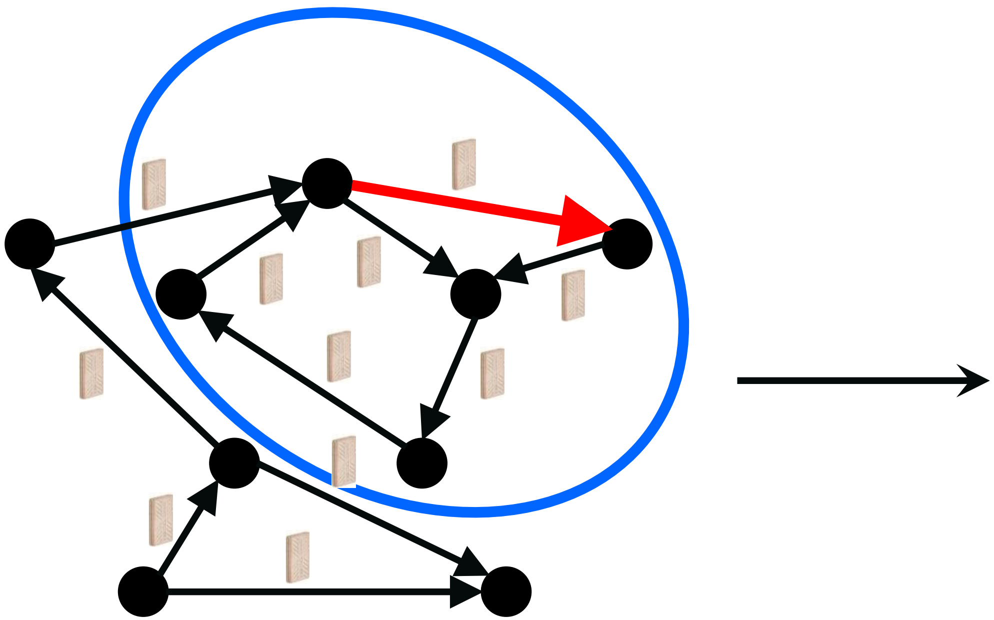
If the underlying (undirected) graph contains a 2-connected subgraph of order at least 2, then the first player wins the game.



TIMBER! (2)

Observation.

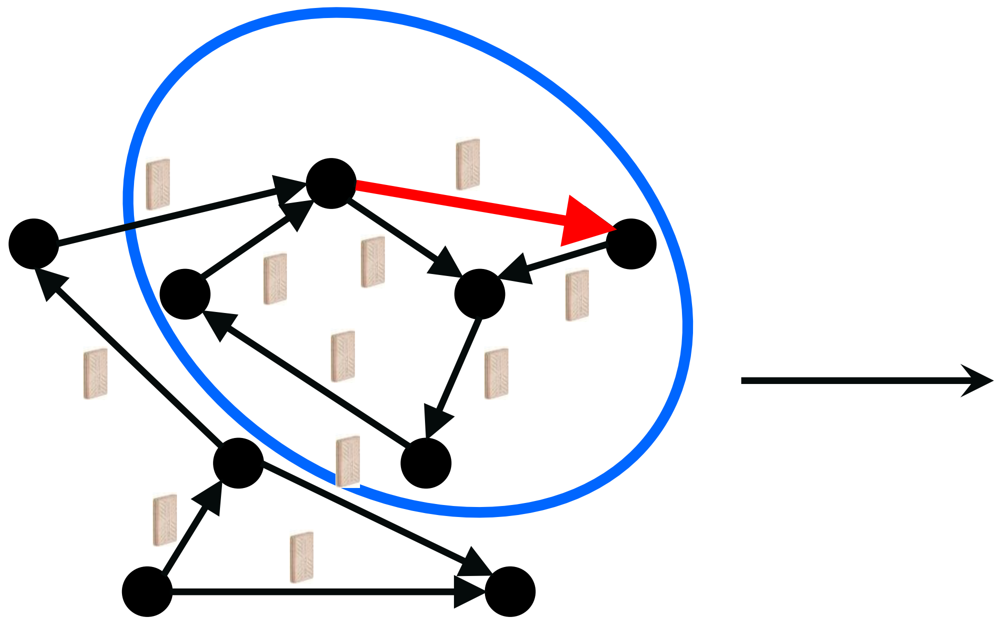
If the underlying (undirected) graph contains a 2-connected subgraph of order at least 2, then the first player wins the game.



TIMBER! (2)

Observation.

If the underlying (undirected) graph contains a 2-connected subgraph of order at least 2, then the first player wins the game.



➢ Therefore, this game is only interesting for trees!

TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, ... is ***0, 1, 0, 2, 0, 5, 0, 14, 0, 42, ...***

TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, ... is ***0, 1, 0, 2, 0, 5, 0, 14, 0, 42, ...***

When $k = 2n$ is even, this number is the n^{th} Catalan number:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

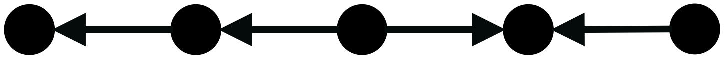
TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, ... is ***0, 1, 0, 2, 0, 5, 0, 14, 0, 42, ...***

When $k = 2n$ is even, this number is the n^{th} Catalan number:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$



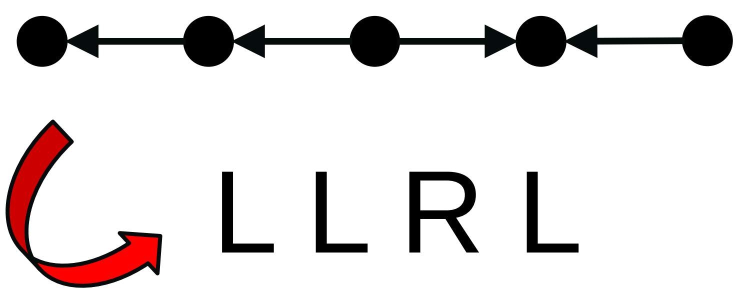
TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, ... is **0, 1, 0, 2, 0, 5, 0, 14, 0, 42, ...**

When $k = 2n$ is even, this number is the n^{th} Catalan number:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$



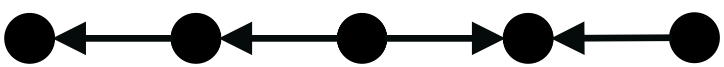
TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, … is ***0, 1, 0, 2, 0, 5, 0, 14, 0, 42, …***

When $k = 2n$ is even, this number is the n^{th} Catalan number:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$



L L R L

Up Up Down Up

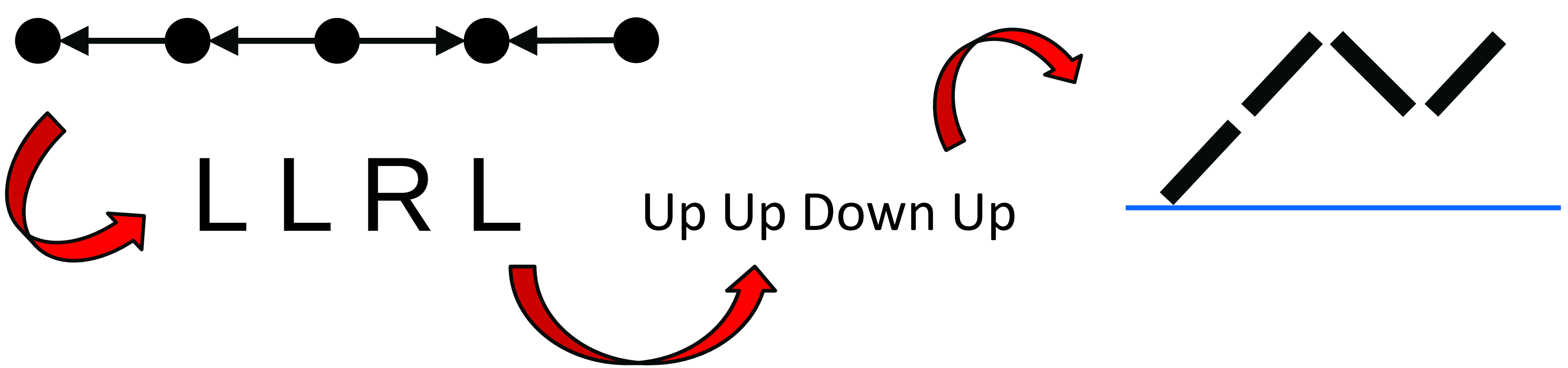
TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, … is ***0, 1, 0, 2, 0, 5, 0, 14, 0, 42, …***

When k = 2n is even, this number is the n^{th} Catalan number:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$



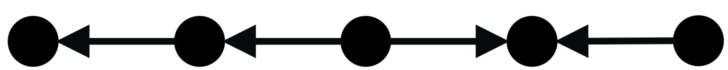
TIMBER! on paths (1)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

The number of loosing positions (orientations) in normal play on a path of length k = 1, 2, ... is ***0, 1, 0, 2, 0, 5, 0, 14, 0, 42, ...***

When k = 2n is even, this number is the n^{th} Catalan number:

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$



L L R L

Up Up Down Up

C_n = number of Dyck paths...

TIMBER! on paths (2)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

TIMBER! on paths (2)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).

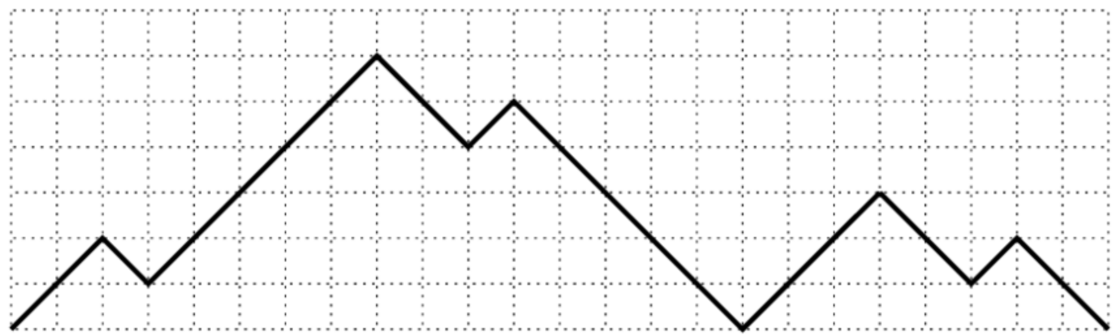
TIMBER! on paths (2)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



Dyck path

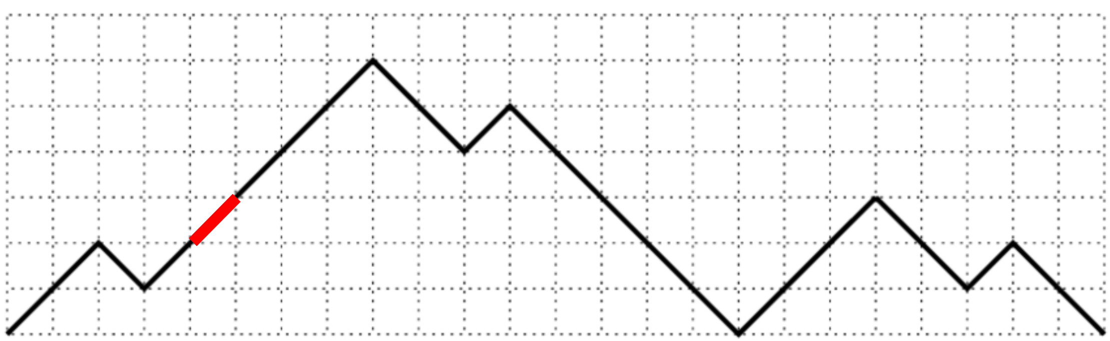
TIMBER! on paths (2)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



Dyck path ⟶ 1st player

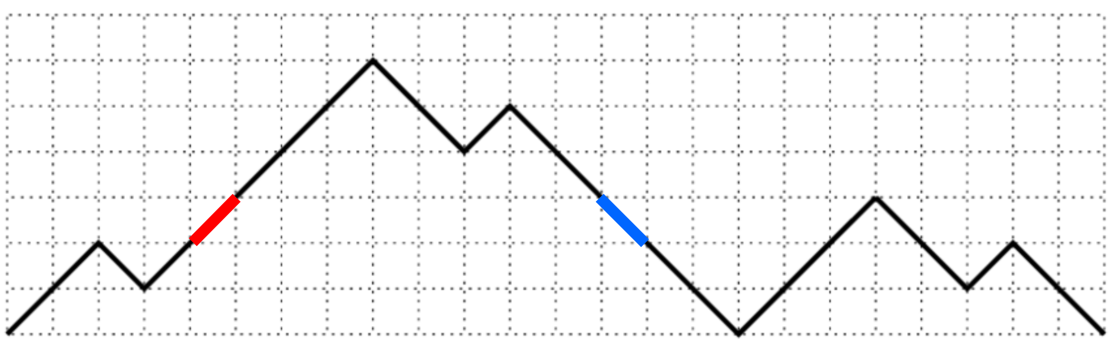
TIMBER! on paths (2)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



Dyck path ⟶ 1st player ⟶ 2nd player

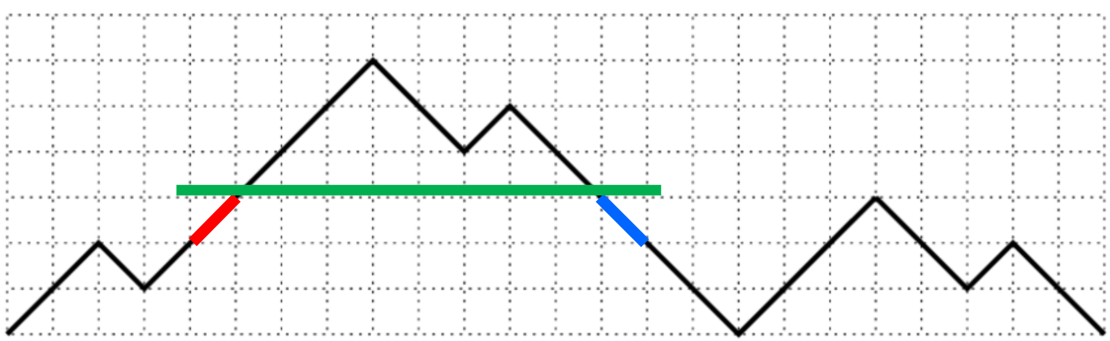
TIMBER! on paths (2)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



Dyck path ⟶ 1st player ⟶ 2nd player ⟶ Dyck path!

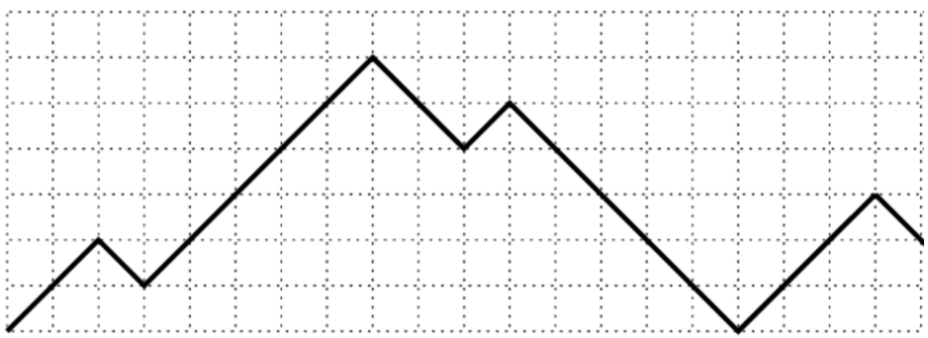
TIMBER! on paths (3)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



non-Dyck path

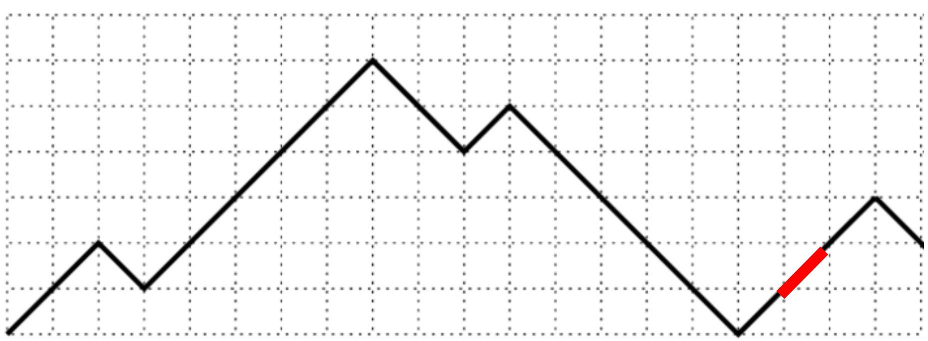
TIMBER! on paths (3)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



non-Dyck path ⟶ 1st player

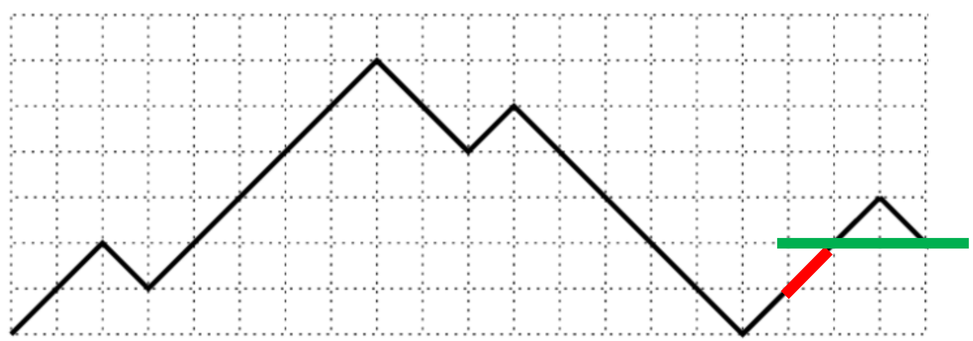
TIMBER! on paths (3)

Theorem [R. NOWAKOWSKI *et al.*, 2014]

In normal play, loosing positions are exactly those positions whose path representation is a Dyck path.

Proof.

- The empty position is a Dyck path (empty).
- Induction step :



non-Dyck path ⟶ 1st player ⟶ Dyck path!

TIMBER! on trees

Theorem [R. NOWAKOWSKI *et al.*, 2014]

- *The outcome of a (directed) tree of order n can by computed in time $O(n^2)$.*
- *A tree is a loosing position if and only if it can be reduced to an empty tree, using two reduction operations.*

TIMBER! on trees

Theorem [R. NOWAKOWSKI *et al.*, 2014]

- *The outcome of a (directed) tree of order n can by computed in time $O(n^2)$.*
- *A tree is a loosing position if and only if it can be reduced to an empty tree, using two reduction operations.*

Open problems

- Is there an efficient algorithm to find the Sprague-Grundy value of a TIMBER! position on a path?

TIMBER! on trees

Theorem [R. NOWAKOWSKI *et al.*, 2014]

- *The outcome of a (directed) tree of order n can by computed in time $O(n^2)$.*
- *A tree is a loosing position if and only if it can be reduced to an empty tree, using two reduction operations.*

Open problems

- Is there an efficient algorithm to find the Sprague-Grundy value of a TIMBER! position on a path?
- Which reductions on trees (or on paths) preserve the Sprague-Grundy value? (*One such reduction is known*.)

TIMBER! on trees

Theorem [R. NOWAKOWSKI *et al.*, 2014]

- *The outcome of a (directed) tree of order n can by computed in time $O(n^2)$.*
- *A tree is a loosing position if and only if it can be reduced to an empty tree, using two reduction operations.*

Open problems

- Is there an efficient algorithm to find the Sprague-Grundy value of a TIMBER! position on a path?
- Which reductions on trees (or on paths) preserve the Sprague-Grundy value? (*One such reduction is known*.)
- Propagation according to the orientation?...

It’s now time to conclude...

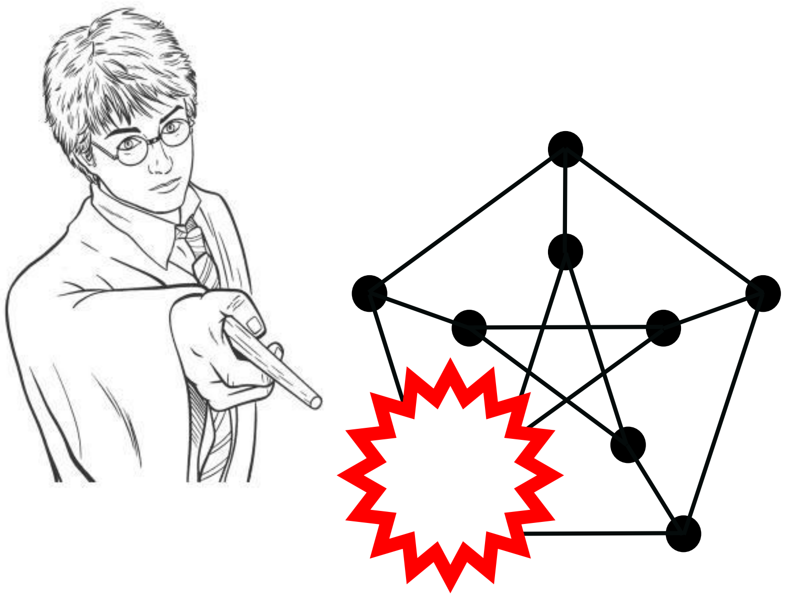
To conclude...

Many other types of combinatorial games on graphs:

To conclude...

Many other types of combinatorial games on graphs:

- Graph deletion games



To conclude...

Many other types of combinatorial games on graphs:

- Graph deletion games
- Avoidance / Achievement games (adding edges until some structure appears, or while some structure does not appear...)

To conclude...

Many other types of combinatorial games on graphs:

- Graph deletion games
- Avoidance / Achievement games (adding edges until some structure appears, or while some structure does not appear...)
- PEG DUOTAIRE (2-player version of PEG SOLITAIRE)

To conclude...

Many other types of combinatorial games on graphs:

- Graph deletion games
- Avoidance / Achievement games (adding edges until some structure appears, or while some structure does not appear...)
- PEG DUOTAIRE (2-player version of PEG SOLITAIRE)
- Take your favourite "graph colouring problem" and consider its combinatorial game version...

Acyclic, 2-distance and a few others in [G. BEAULIEU, K. BURKE, E. DUCHÊNE, 2013]

To conclude...

Many other types of combinatorial games on graphs:

- Graph deletion games
- Avoidance / Achievement games (adding edges until some structure appears, or while some structure does not appear...)
- PEG DUOTAIRE (2-player version of PEG SOLITAIRE)
- Take your favourite "graph colouring problem" and consider its combinatorial game version...

  Acyclic, 2-distance and a few others
  in [G. BEAULIEU, K. BURKE, E. DUCHÊNE, 2013]

- Partisan games (different options for players, e.g. playing with black or white tokens)

To conclude...

Many other types of combinatorial games on graphs:

- Graph deletion games
- Avoidance / Achievement games (adding edges until some structure appears, or while some structure does not appear...)
- PEG DUOTAIRE (2-player version of PEG SOLITAIRE)
- Take your favourite "graph colouring problem" and consider its combinatorial game version...

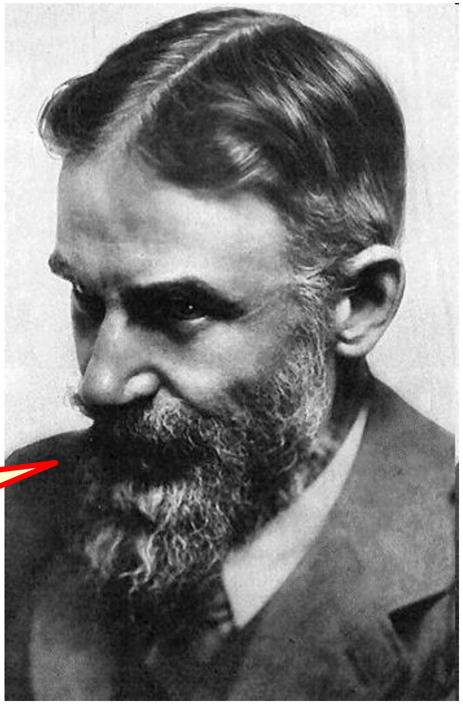
  Acyclic, 2-distance and a few others in [G. BEAULIEU, K. BURKE, E. DUCHÊNE, 2013]

- Partisan games (different options for players, e.g. playing with black or white tokens)
- ...

To conclude…

Human beings are never more ingenious than in the invention of games.

GOTTFRIED WILHELM LEIBNIZ



We don't stop playing because we grow old; we grow old because we stop playing.

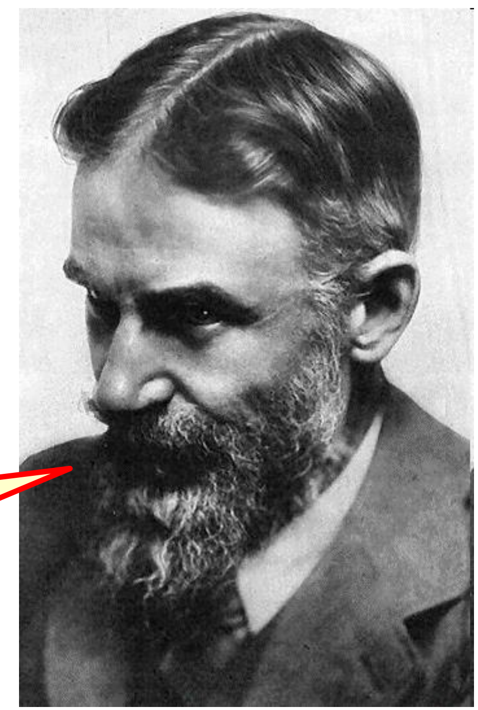
GEORGE BERNARD SHAW

To conclude...

Human beings are never more ingenious than in the invention of games.

Thank you for your attention...

Gottfried Wilhelm Leibniz



We don't stop playing because we grow old; we grow old because we stop playing.

George Bernard Shaw