Taking and merging games as rewrite games

Eric Duchêne and Aline Parreau

LIRIS, CNRS, Université Lyon 1

Joint work with Victor Marsault and Michel Rigo

One World Combinatorics on Words Seminar, February 27th 2023

First part

Combinatorial games

Combinatorial games: definition

Berlekamp, Conway and Guy, *Winning Ways*, 1981

Combinatorial games: definition

Berlekamp, Conway and Guy, *Winning Ways*, 1981

- 2 players

Chess

Tarot

Othello

Checkers

TicTacToe

Ludo

Go

Combinatorial games: definition

Berlekamp, Conway and Guy, *Winning Ways*, 1981

- 2 players
- Total information, no chance

Chess

Tarot

Othello

Checkers

TicTacToe

Ludo

Go

Combinatorial games: definition

Berlekamp, Conway and Guy, *Winning Ways*, 1981

- 2 players
- Total information, no chance
- Finite number of turns, no draw


Chess


Tarot


Othello


Checkers


TicTacToe


Ludo


Go

Combinatorial games: definition

Berlekamp, Conway and Guy, *Winning Ways*, 1981

- 2 players
- Total information, no chance
- Finite number of turns, no draw
- Winner given by the last move.
  Normal Convention: the player who cannot play loses.

~~Chess~~

~~Tarot~~

~~Othello~~

~~Checkers~~

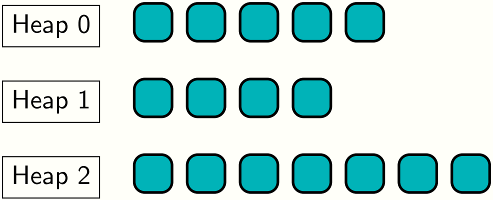
~~TicTacToe~~

~~Ludo~~

~~Go~~

“Taking and breaking” games

Board: Heaps of tokens
Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.
The player who cannot play loses.

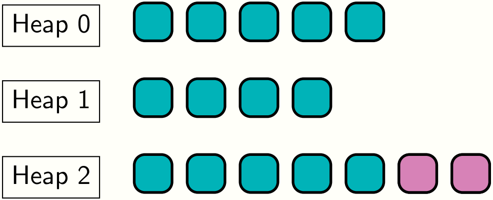


"Taking and breaking" games

Board: Heaps of tokens

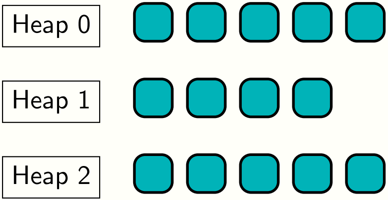
Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.

The player who cannot play loses.



“Taking and breaking” games

Board: Heaps of tokens
Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.
The player who cannot play loses.

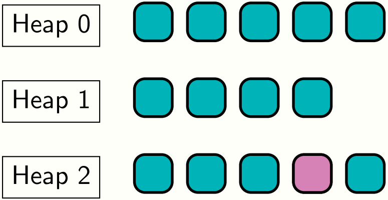


“Taking and breaking” games

Board: Heaps of tokens

Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.

The player who cannot play loses.

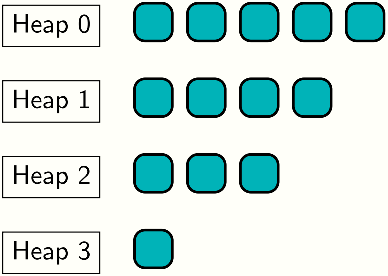


“Taking and breaking” games

Board: Heaps of tokens

Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.

The player who cannot play loses.

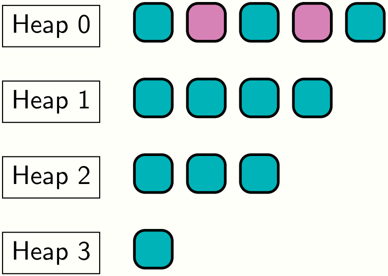


“Taking and breaking” games

Board: Heaps of tokens

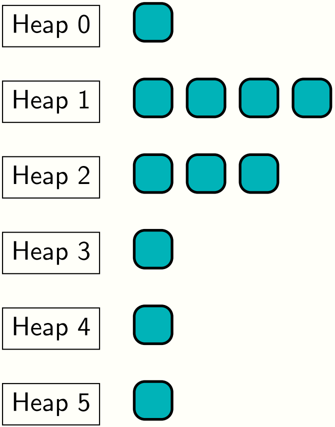
Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.

The player who cannot play loses.



“Taking and breaking” games

Board: Heaps of tokens
Rules: A player takes tokens in a single heap, with some constraints on the number, and possibly splits the heap.
The player who cannot play loses.



Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap → Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

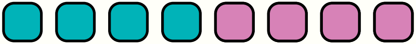
Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

Example : $S = \{1, 2, 4\}$



Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

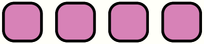
Example : $S = \{1, 2, 4\}$

Subtraction games

One cannot split a heap $\rightarrow$ Subtraction game
Defined by a set $S \subseteq \mathbb{N}$:

- At his turn, a player removes $k \in S$ tokens from a heap, without breaking it.
- The player who cannot play loses.

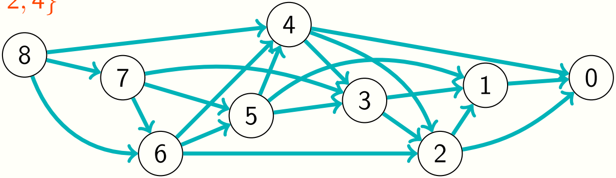
Example : $S = \{1, 2, 4\}$



Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

$S = \{1, 2, 4\}$

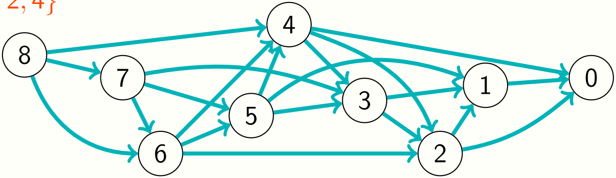


- Playing in the game $\Leftrightarrow$ Moving a token along the arcs

Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

$S = \{1, 2, 4\}$

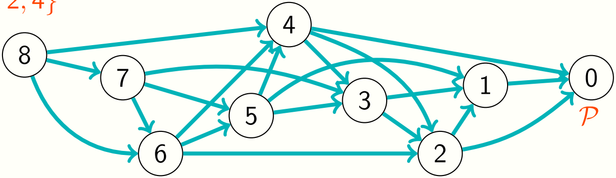


- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.

Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

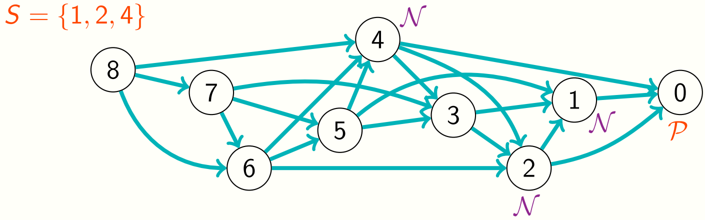
$S = \{1, 2, 4\}$



- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Existence of a winning strategy

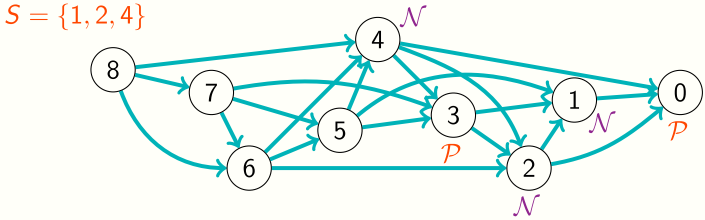
Any combinatorial game can be represented by a finite DAG.



- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Existence of a winning strategy

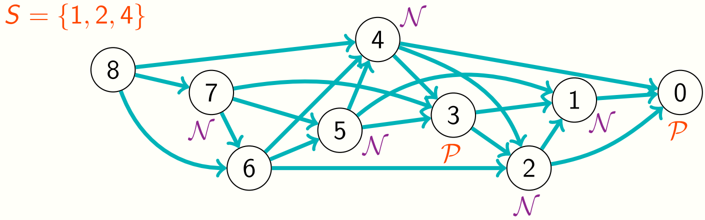
Any combinatorial game can be represented by a finite DAG.



- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

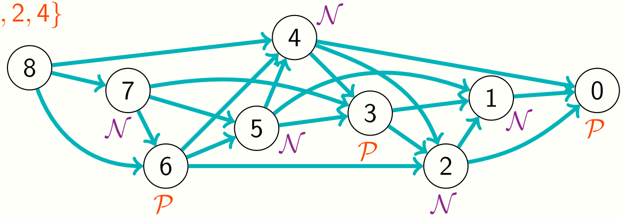


- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

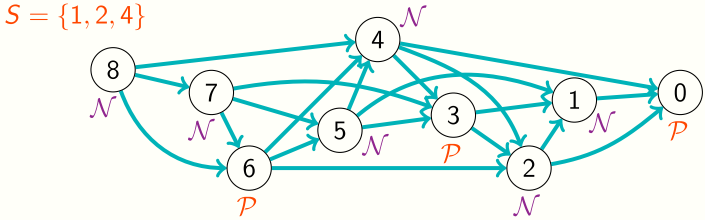
$S = \{1, 2, 4\}$



- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

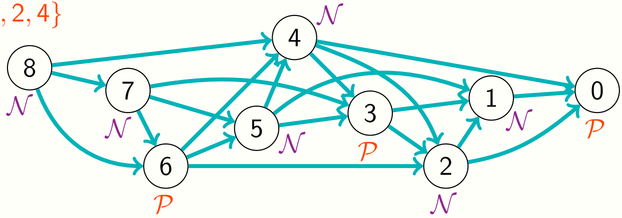


- Playing in the game $\Leftrightarrow$ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Existence of a winning strategy

Any combinatorial game can be represented by a finite DAG.

$S = \{1, 2, 4\}$



- Playing in the game ⇔ Moving a token along the arcs
- Starting from the sinks, one can determine the winner:
  - $\mathcal{N}$ if the Next player can force the win,
  - $\mathcal{P}$ if the Previous player can force the win.
- From a $\mathcal{N}$-position, there is always a move to a $\mathcal{P}$-position.
- From a $\mathcal{P}$-position, all moves are to $\mathcal{N}$-positions

Theorem

One of the players has a winning strategy.

Main issue

Outcome of the game

Input : Game position
Output : First ($\mathcal{N}$) or second ($\mathcal{P}$) player wins?

Winning strategy

Input : Game position
Output : If the game is $\mathcal{N}$, a winning move.

Main issue

Outcome of the game

Input : Game position
Output : First ($\mathcal{N}$) or second ($\mathcal{P}$) player wins?

Winning strategy

Input : Game position
Output : If the game is $\mathcal{N}$, a winning move.

These two problems can be solved using the DAG...

Main issue

Outcome of the game

Input : Game position
Output : First ($\mathcal{N}$) or second ($\mathcal{P}$) player wins?

Winning strategy

Input : Game position
Output : If the game is $\mathcal{N}$, a winning move.

These two problems can be solved using the DAG...but its size is often exponential !

They are generally in PSPACE

A standard PSPACE problem

Quantified Boolean Formula (QBF)

Input : $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \Phi(x_1, \ldots, x_n)$: $Q_i \in \{\forall, \exists\}$
Output : Is the formula true?

A standard PSPACE problem

Quantified Boolean Formula (QBF)

Input : $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \Phi(x_1, \dots, x_n)$: $Q_i \in \{\forall, \exists\}$
Output : Is the formula true?

Description of a winning strategy?
"There exists a move for J1, such that, for all moves of J2, there exists a move for J1..."

A standard PSPACE problem

Quantified Boolean Formula (QBF)

Input : $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \Phi(x_1, \dots, x_n)$: $Q_i \in \{\forall, \exists\}$
Output : Is the formula true?

Description of a winning strategy?
"There exists a move for J1, such that, for all moves of J2, there exists a move for J1..."

QBF-game :

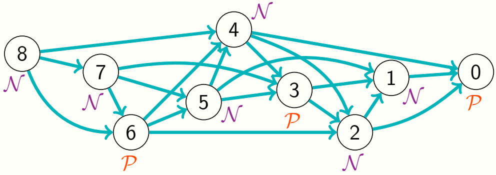
- Board: logic formula $\Phi(x_1, \dots, x_n)$
- Players assign boolean values to x_1,...,x_n, following this order.
- First player wins if at the end the formula is true.

Theorem Schaeffer, 1989 and Arora,Barak, 2009

Deciding if there is a winning strategy for the first player at QBF-game is PSPACE-complete.

Polynomiality of subtraction games

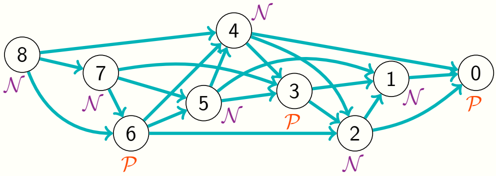
Consider the subtraction game $S = \{1, 2, 4\}$ on n token.



n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$

Polynomiality of subtraction games

Consider the subtraction game $S = \{1, 2, 4\}$ on n token.

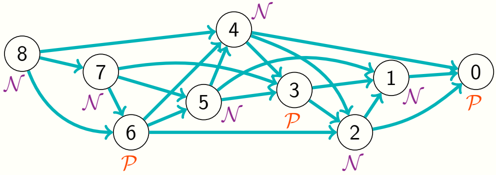


n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$

A position n is $\mathcal{P}$ if and only if $n \equiv 0 \bmod 3$.

Polynomiality of subtraction games

Consider the subtraction game $S = \{1, 2, 4\}$ on n token.



n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$

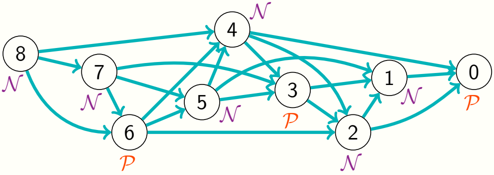
A position n is $\mathcal{P}$ if and only if $n \equiv 0 \bmod 3$.

Proposition

Any finite subtraction game has its outcome sequence that is ultimately periodic.

Polynomiality of subtraction games

Consider the subtraction game $S = \{1, 2, 4\}$ on n token.



n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$

A position n is $\mathcal{P}$ if and only if $n \equiv 0 \bmod 3$.

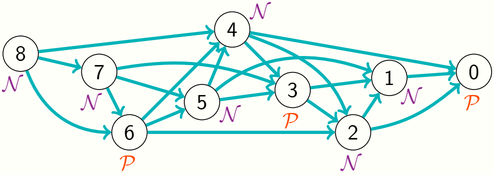
Proposition

Any finite subtraction game has its outcome sequence that is ultimately periodic.

$\rightarrow$ computing the outcome of the game is polynomial.

Polynomiality of subtraction games

Consider the subtraction game $S = \{1, 2, 4\}$ on n token.



n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$

A position n is $\mathcal{P}$ if and only if $n \equiv 0 \bmod 3$.

Proposition

Any finite subtraction game has its outcome sequence that is ultimately periodic.

$\rightarrow$ computing the outcome of the game is polynomial.

Open

Size of the preperiod and the period in function of S?

Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.

Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.

Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.

Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.

Breaking the heaps?

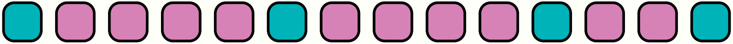
CRAM: Players take two tokens in a heap and can split it into two heaps.

Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.

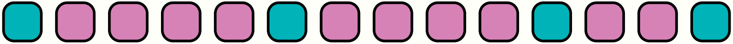
Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.



Breaking the heaps?

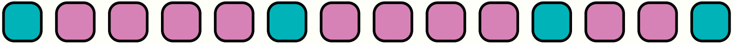
CRAM: Players take two tokens in a heap and can split it into two heaps.



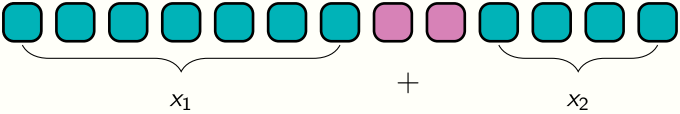
The number of positions is exponential, how to simplify?

Breaking the heaps?

CRAM: Players take two tokens in a heap and can split it into two heaps.



The number of positions is exponential, how to simplify?
$\rightarrow$ with sum of games



Computing the outcome of a sum

With a $\mathcal{P}$-position:

$$\mathcal{P} + x \equiv x$$

Computing the outcome of a sum

With a $\mathcal{P}$-position:

$$\mathcal{P} + \mathrm{x} \equiv \mathrm{x}$$

With only $\mathcal{N}$-positions:

$$\mathcal{N} + \mathcal{N} \equiv \ ?$$

Computing the outcome of a sum

With a $\mathcal{P}$-position:

$$\mathcal{P} + x \equiv x$$

With only $\mathcal{N}$-positions:

$$\mathcal{N} + \mathcal{N} \equiv \ ?$$

 $+$  $\equiv \mathcal{P}$

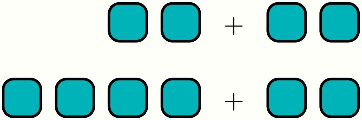
Computing the outcome of a sum

With a $\mathcal{P}$-position:

$$\mathcal{P} + x \equiv x$$

With only $\mathcal{N}$-positions:

$$\mathcal{N} + \mathcal{N} \equiv \; ?$$



$\equiv \mathcal{P}$

$\equiv \mathcal{N}$

Computing the outcome of a sum

With a $\mathcal{P}$-position:

$$\mathcal{P} + x \equiv x$$

With only $\mathcal{N}$-positions:

$$\mathcal{N} + \mathcal{N} \equiv \ ?$$

$\equiv \mathcal{P}$

$\equiv \mathcal{N}$

$+$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$ or $\mathcal{N}$

Grundy values

Let $I \subset \mathbb{N}$. MeX (minimum excluded value) of $I = \min \mathbb{N} \setminus I$.

$$\text{MeX}(\{0, 1, 3, 5\}) = 2, \quad \text{MeX}(\{2, 3, 6\}) = 0, \quad \text{MeX}(\emptyset) = 0.$$

Grundy values

Let $I \subset \mathbb{N}$. MeX (minimum excluded value) of $I = \min \mathbb{N} \setminus I$.

$$\mathrm{MeX}(\{0,1,3,5\}) = 2, \quad \mathrm{MeX}(\{2,3,6\}) = 0, \quad \mathrm{MeX}(\emptyset) = 0.$$

The Grundy value of a position x is given by

$$\mathcal{G}(x) = \mathrm{MeX}(\mathcal{G}(N^+(x)))$$

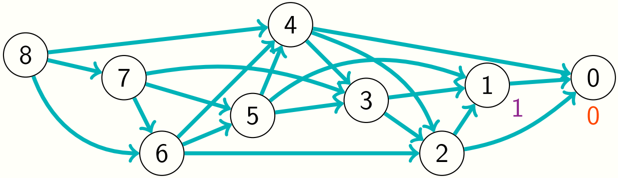
Grundy values

Let $I \subset \mathbb{N}$. MeX (minimum excluded value) of $I = \min \mathbb{N} \setminus I$.

$$\mathrm{MeX}(\{0,1,3,5\}) = 2, \quad \mathrm{MeX}(\{2,3,6\}) = 0, \quad \mathrm{MeX}(\emptyset) = 0.$$

The Grundy value of a position x is given by

$$\mathcal{G}(x) = \mathrm{MeX}(\mathcal{G}(N^+(x)))$$



Proposition

$\mathcal{G}(x) = 0$ if and only if x is $\mathcal{P}$.

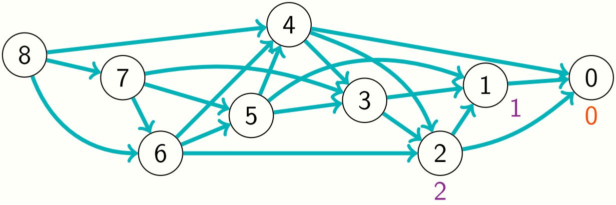
Grundy values

Let $I \subset \mathbb{N}$. MeX (minimum excluded value) of $I = \min \mathbb{N} \setminus I$.

$$\mathrm{MeX}(\{0,1,3,5\}) = 2, \quad \mathrm{MeX}(\{2,3,6\}) = 0, \quad \mathrm{MeX}(\emptyset) = 0.$$

The Grundy value of a position x is given by

$$\mathcal{G}(x) = \mathrm{MeX}(\mathcal{G}(N^+(x)))$$



Proposition

$\mathcal{G}(x) = 0$ if and only if x is $\mathcal{P}$.

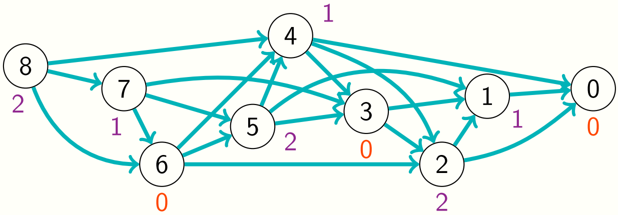
Grundy values

Let $I \subset \mathbb{N}$. MeX (minimum excluded value) of $I = \min \mathbb{N} \setminus I$.

$$\mathrm{MeX}(\{0,1,3,5\}) = 2, \quad \mathrm{MeX}(\{2,3,6\}) = 0, \quad \mathrm{MeX}(\emptyset) = 0.$$

The Grundy value of a position x is given by

$$\mathcal{G}(x) = \mathrm{MeX}(\mathcal{G}(N^+(x)))$$



Proposition

$\mathcal{G}(x) = 0$ if and only if x is $\mathcal{P}$.

Grundy value of the sum of games

$+$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$ or $\mathcal{N}$

Grundy value of the sum of games

$+$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$ or $\mathcal{N}$

Theorem Sprague–Grundy

Let x_1, x_2 be two game positions. Then:

$$\mathcal{G}(x_1 + x_2) = \mathcal{G}(x_1) \oplus \mathcal{G}(x_2)$$

where $\oplus$ is the XOR operator.

Grundy value of the sum of games

$+$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}$
$\mathcal{N}$	$\mathcal{N}$	$\mathcal{P}$ or $\mathcal{N}$

Theorem Sprague–Grundy

Let x_1, x_2 be two game positions. Then:

$$\mathcal{G}(x_1 + x_2) = \mathcal{G}(x_1) \oplus \mathcal{G}(x_2)$$

where $\oplus$ is the XOR operator.

Corollary

The sum $x_1 + x_2$ is $\mathcal{P}$ if and only if $\mathcal{G}(x_1) = \mathcal{G}(x_2)$.

Grundy sequence

Grundy sequence: sequence of grundy values for 1,2,3,..., n tokens.
For the subtraction game $\{1, 2, 4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{G}(n)$	0	1	2	0	1	2	0	1	2	0	1	2	0

Grundy sequence

Grundy sequence: sequence of grundy values for 1,2,3,..., n tokens.
For the subtraction game $\{1,2,4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{G}(n)$	0	1	2	0	1	2	0	1	2	0	1	2	0

Theorem Berlekamp, Conway, Guy, 1981

Finite subtraction games have ultimaltely periodic sequences.

Grundy sequence

Grundy sequence: sequence of grundy values for 1,2,3,..., n tokens.
For the subtraction game $\{1,2,4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{G}(n)$	0	1	2	0	1	2	0	1	2	0	1	2	0

Theorem Berlekamp, Conway, Guy, 1981

Finite subtraction games have ultimaltely periodic sequences.

For CRAM:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{G}(n)$	0	0	1	1	2	0	3	1	1	0	3	3	2

Grundy sequence

Grundy sequence: sequence of grundy values for 1,2,3,..., n tokens.
For the subtraction game $\{1, 2, 4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{G}(n)$	0	1	2	0	1	2	0	1	2	0	1	2	0

Theorem Berlekamp, Conway, Guy, 1981

Finite subtraction games have ultimaltely periodic sequences.

For CRAM:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$\mathcal{G}(n)$	0	0	1	1	2	0	3	1	1	0	3	3	2

Theorem Guy, Smith, 1956

The Grundy sequence of CRAM is periodic with period 34 and preperiod 53.

Octal games

Played on several heaps of tokens. A move consists in choosing a heap and, according to the rules:

- remove all the tokens from the heap, and delete this heap,
- remove some tokens from the heap, leaving 1 non-empty heap,
- remove some tokens from the heap, separating the remaining tokens into 2 non-empty heaps.

The number of tokens that can be removed is given by the game rules via an octal code

$$d_0 \bullet d_1 d_2 d_3 \cdots \qquad d_i \in 0, \ldots, 7$$

Octal games

Played on several heaps of tokens. A move consists in choosing a heap and, according to the rules:

- remove all the tokens from the heap, and delete this heap,
- remove some tokens from the heap, leaving 1 non-empty heap,
- remove some tokens from the heap, separating the remaining tokens into 2 non-empty heaps.

The number of tokens that can be removed is given by the game rules via an octal code

$$d_0 \bullet d_1 d_2 d_3 \cdots \quad d_i \in 0, \ldots, 7$$

Examples:

- The subtraction game $\{1, 2, 4\}$ is the octal game $0 \bullet 3303$.
- The game CRAM corresponds to the octal game $0 \bullet 07$.
- The game $0 \bullet 304$ allows you to remove 1 token without splitting the heap, or 3 tokens by necessarily dividing the heap.

Guy's Conjecture

Conjecture Guy, 1956

The Grundy sequence of a finite octal game is ultimately periodic.

Guy's Conjecture

Conjecture Guy, 1956

The Grundy sequence of a finite octal game is ultimately periodic.

- For the game $0 \bullet 106$, the Grundy sequence is ultimately periodic, with a period of 328226140474, and a pre-period of 465384263797.

Guy's Conjecture

Conjecture Guy, 1956

The Grundy sequence of a finite octal game is ultimately periodic.

- For the game $0 \bullet 106$, the Grundy sequence is ultimately periodic, with a period of 328226140474, and a pre-period of 465384263797.
- The Grundy sequence of the game $0 \bullet 007$ (James Bond Game) is conjectured to be periodic (tested up to 2^{28})

Guy's Conjecture

Conjecture Guy, 1956

The Grundy sequence of a finite octal game is ultimately periodic.

- For the game $0 \bullet 106$, the Grundy sequence is ultimately periodic, with a period of 328226140474, and a pre-period of 465384263797.
- The Grundy sequence of the game $0 \bullet 007$ (James Bond Game) is conjectured to be periodic (tested up to 2^{28})
- It is open for very simple games like $0 \bullet 6$!

Which octal games are in P ?

Nontrivial Octal-Games with at most 3 places

Game	sgv-sequence	type	bitstring	rare	last	max n	max G	index	lost	depth	period	preperiod	except
.6	0012012312340342...	0	0111011111111...	1584	20627	2^{33}	363	7775706554	14	1008823			
.04	0000111220331110...	0	0001110111111...	22476	5029984	2^{28}	1689	248902928	38	5218954			
.06	0001122031122334...	0	?			2^{24}	22097	16360327	37				
.14	0100102122104144...	0	0011111111111...	1896	178727	2^{32}	85	1839780623	172	576735			
.16	0100122140142140...	0	0111111111111...	53	13935	-	23	229790	7	21577	149459	105351	16
.36	0102102132132430...	0	0111111111111...	516	11798	2^{34}	208	1762187846	14	17168			
.37	0120123123403421...	0	0111011111111...	1583	20626	2^{33}	363	7775706553	13	1008822			
.45	0011223114432211...	0	1111111111111...	11	198	-	8	37	2	37	20	498	8
.56	0102241132446621...	0	1101101111111...	46	1795	-	64	22778	2	7405	144	326640	26
.64	0012341532154268...	0	0111110111111...	488	156751	2^{33}	262	1911635806	2	470403814			
.74	0101232414623215...	0	1101101111111...	1386	15929	2^{31}	512	76103606	2	137102			
.76	0102341623416732...	0	0000000110011...	219248	5208068	2^{24}	16814	4995486	2				
.004	0000011112220333...	0	0001111111111...	184854	15869181	2^{25}	6063	22057995	32				
.005	0001011222033411...	1	1110101001011...	95660	67070800	2^{26}	1059	3022366	-				
.006	0000111222033111...	0	0000000101111...	470413	16772624	2^{24}	6532	4798522	40				
.007	0001112203311104...	0	0001110111111...	22476	5029983	2^{28}	1689	248902927	37	5218954			
.014	0010010122123401...	0	0111111111111...	2037	64126	2^{31}	365	169860345	13	126438			
.015	0011010212230142...	0	0111111111111...	237	11973	2^{35}	101	2350397235	7	27036			
.016	0010122201014422...	0	0010111111111...	21439	102335997	2^{27}	1093	102705419	18	41416941			
.024	0001122304112532...	0	?			2^{25}	12371	30810166	26				
.026	0001122304112533...	0	?			2^{25}	37903	33220674	27				
.034	0011022314014312...	0	1111111111111...	1079	374473	2^{34}	256	26376	10	596840			
.054	0010122234411163...	0	1011111111111...	38	796	-	41	33671802	3	16284	10015179	193235616	18
.055	0011122231114443...	0	1111111111111...	6	43	-	8	51	2	20	148	259	2
.064	0001122334115533...	0	0111111111111...	6795	528569986	2^{29}	523	275511554	3	28677643			
.104	0100010221224104...	1	1110111111111...	20	284	-	29	186892397	-	4178	11770282	197769598	9
.106	0100012221440106...	1	1011011111111...	15	1103	-	31	1937780317	-	15343	328226140474	465384263797	25
.114	0110011202120411...	0	1111111101111...	100891	33547932	2^{25}	1610	20501458	11				
.125	0102110213011302...	0	?			2^{24}	44496	16775217	145				
.126	0100213321042503...	1	0111001110001...	20444	102973539	2^{28}	2222	265978	-	40637003			
.127	0102210441220144...	1	1000001111111...	693	27106	-	56	24734	1190	13551	4	46578	11
.135	0112011203110312...	0	?			2^{24}	27960	16768149	91				
.136	0110021302110223...	0	?			2^{24}	25272	15750407	40				
.142	0100222110332410...	1	1100011101111...	1357	117323	2^{34}	441	17142768844	-	411815			
.143	0101222010422150...	0	1011111111111...	9417	2561883	2^{27}	148	26789789	13	3047015			

http://wwwhomes.uni-bielefeld.de/achim/octal.html

Second Part

Rewriting games

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word t, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : R_1 : $\texttt{ab} \rightarrow \varepsilon$, R_2 : $\texttt{aaa} \rightarrow \texttt{b}$ and $t = \texttt{aabbbaabaaa}$

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word `t`, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : R_1 : $\mathtt{ab} \to \varepsilon$, R_2 : $\mathtt{aaa} \to \mathtt{b}$ and $t = \mathtt{aabbbaabaaa}$

`aabbbaabaaa`

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word t, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : $R_1 : \texttt{ab} \to \varepsilon$, $R_2 : \texttt{aaa} \to \texttt{b}$ and $t = \texttt{aabbbaabaaa}$

$\texttt{aabbbaabaaa} \to \texttt{aabbbaaaa}$

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word `t`, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : $R_1 : \mathtt{ab} \to \varepsilon$, $R_2 : \mathtt{aaa} \to \mathtt{b}$ and $t = \mathtt{aabbbaabaaa}$

$\mathtt{aabbbaabaaa} \to \mathtt{aabbbaaaa} \to \mathtt{aabbbba}$

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word t, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : R_1 : ab $\to \varepsilon$, R_2 : aaa $\to$ b and $t =$ aabbbaabaaa

aabbbaabaaa $\to$ aabbbaaaa $\to$ aabbbba $\to$ abbba

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word t, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : $R_1 : \texttt{ab} \to \varepsilon$, $R_2 : \texttt{aaa} \to \texttt{b}$ and $t = \texttt{aabbbaabaaa}$

$\texttt{aabbbaabaaa} \to \texttt{aabbbaaaa} \to \texttt{aabbbba} \to \texttt{abbba} \to \texttt{bba}$

The first player can no longer play and loses.

Rewriting games

Rewriting games (Waldmann, 2002):

- Rewriting system (terminal)
- Starting from a word t, players alternate applying rules to the word.
- The player who can no longer apply a rule loses.

Example : R_1 : ab $\rightarrow \varepsilon$, R_2 : aaa $\rightarrow$ b and $t =$ aabbbaabaaa

aabbbaabaaa $\rightarrow$ aabbbaaaa $\rightarrow$ aabbbba $\rightarrow$ abbba $\rightarrow$ bba

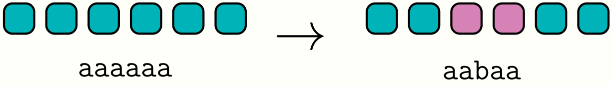
The first player can no longer play and loses.

Allows to model many games, including octal games.

Octal games as rewrite games

Alphabet on two letters: a (for tokens), b (to separate the heaps)

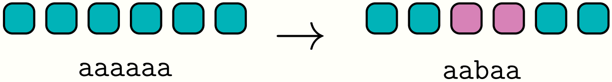
Example: Game CRAM can be modeled with rules $\mathtt{aa} \to \varepsilon$ and $\mathtt{aa} \to \mathtt{b}$.



Octal games as rewrite games

Alphabet on two letters: a (for tokens), b (to separate the heaps)

Example: Game CRAM can be modeled with rules $\mathtt{aa} \to \varepsilon$ and $\mathtt{aa} \to \mathtt{b}$.



The three rules of octal games can be translated with rewrite rules:

- Emptying a heap of k tokens: $\mathtt{ba}^k\mathtt{b} \to \mathtt{b}$
- Removing k without emptying: $\mathtt{a}^{k+1} \to \mathtt{a}$
- Removing k and splitting in 2: $\mathtt{a}^{k+2} \to \mathtt{aba}$

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $\mathcal{G}(t)$.
Grundy class $\mathcal{L}_k$: words with value k.

Theorem Waldmann, 2002

The Grundy sequence of an octal game is ultimately periodic iff, in the associated rewriting game, there is a finite number of non-empty Grundy classes, and each class is rational.

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $\mathcal{G}(t)$.
Grundy class $\mathcal{L}_k$: words with value k.

Theorem Waldmann, 2002

The Grundy sequence of an octal game is ultimately periodic iff, in the associated rewriting game, there is a finite number of non-empty Grundy classes, and each class is rational.

$\Rightarrow$ Find a DFA that determines if a given word $\mathtt{ba}^{x_1}\mathtt{ba}^{x_2}\cdots\mathtt{ba}^{x_n}\mathtt{b}$ satisfies $\mathcal{G}(x_1) \oplus \ldots \oplus \mathcal{G}(x_n) = k$.

- There exists a DFA that computes $\mathcal{G}(x_i)$ $\forall i$
- Before each new x_i, we keep in memory the previous sum $\mathcal{G}(x_1) \oplus \ldots \oplus \mathcal{G}(x_{i-1})$: possible because the number of Grundy classes is bounded (by M).
- The new sum can be computed by a DFA.

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $\mathcal{G}(t)$.
Grundy class $\mathcal{L}_k$: words with value k.

Theorem Waldmann, 2002

The Grundy sequence of an octal game is ultimately periodic iff, in the associated rewriting game, there is a finite number of non-empty Grundy classes, and each class is rational.

$\Leftarrow$ The $\mathcal{L}_k$ are rational.

- $\mathcal{L}_k \cap \mathtt{ba}^*\mathtt{b}$ is rational.
- Rational language with one letter $\Leftrightarrow \bigcup \mathtt{ba}^{kp+\ell}\mathtt{b} : k \in \mathbb{N}$
- Partition of $\mathbb{N} \Rightarrow$ periods are multiple.

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $\mathcal{G}(t)$.
Grundy class $\mathcal{L}_k$: words with value k.

Theorem Waldmann, 2002

The Grundy sequence of an octal game is ultimately periodic iff, in the associated rewriting game, there is a finite number of non-empty Grundy classes, and each class is rational.

Conjecture Guy, 1956

For any octal rewriting game, there is a finite number of non-empty Grundy classes, and each class is a rational language.

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $\mathcal{G}(t)$.
Grundy class $\mathcal{L}_k$: words with value k.

Theorem Waldmann, 2002

The Grundy sequence of an octal game is ultimately periodic iff, in the associated rewriting game, there is a finite number of non-empty Grundy classes, and each class is rational.

Conjecture Guy, 1956

For any octal rewriting game, there is a finite number of non-empty Grundy classes, and each class is a rational language.

With different rules? Rational classes?

What about other types of rules?

- What happens if we can delete `b`'s? $\rightarrow$ Taking-and-merging games

What about other types of rules?

- What happens if we can delete $\mathtt{b}$'s? $\rightarrow$ Taking-and-merging games

Définition

A rewriting game is said to be "taking-and-merging" if all the rules are of the form $\mathtt{a}^k \rightarrow \varepsilon$ or $\mathtt{b}^\ell \rightarrow \varepsilon$

Notation: $\mathtt{a}^{k_1}, \mathtt{a}^{k_2}, \ldots, \mathtt{a}^{k_n}, \mathtt{b}^{\ell_1}, \mathtt{b}^{\ell_2}, \ldots \mathtt{b}^{\ell_m}$

What about other types of rules?

- What happens if we can delete $\mathtt{b}$'s? $\rightarrow$ Taking-and-merging games

Définition

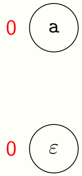
A rewriting game is said to be "taking-and-merging" if all the rules are of the form $\mathtt{a}^k \rightarrow \varepsilon$ or $\mathtt{b}^\ell \rightarrow \varepsilon$

Notation: $\mathtt{a}^{k_1}, \mathtt{a}^{k_2}, \ldots, \mathtt{a}^{k_n}, \mathtt{b}^{\ell_1}, \mathtt{b}^{\ell_2}, \ldots \mathtt{b}^{\ell_m}$

Question: Are the Grundy classes rational?

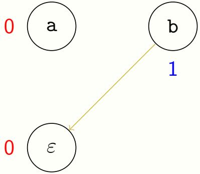
A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



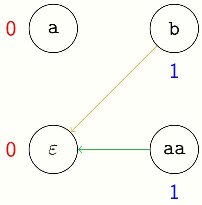
A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



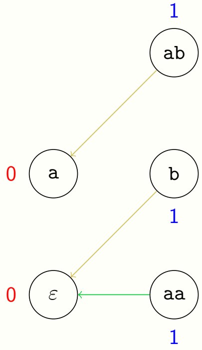
A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



A first example: the game $\mathtt{a}^2, \mathtt{b}$

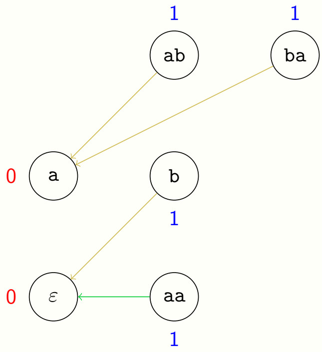
Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



A first example: the game $\mathtt{a}^2, \mathtt{b}$

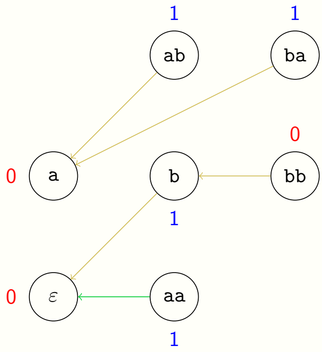
Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$

We construct the DAG starting with small words:



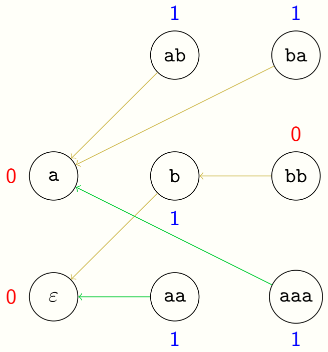
A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



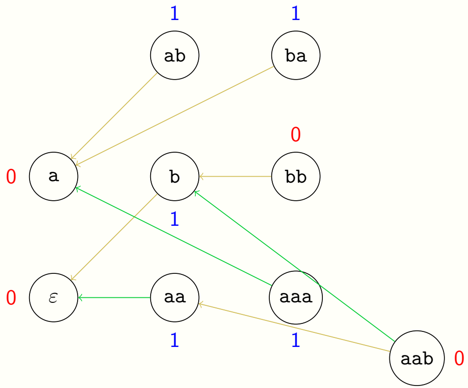
A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



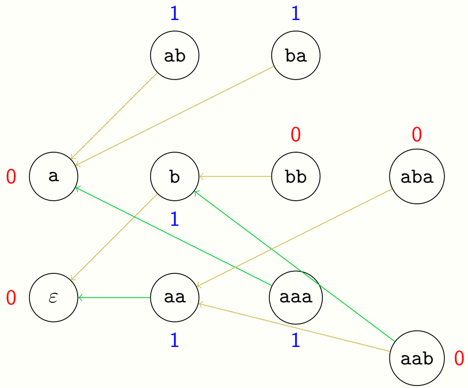
A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



A first example: the game $\mathtt{a}^2, \mathtt{b}$

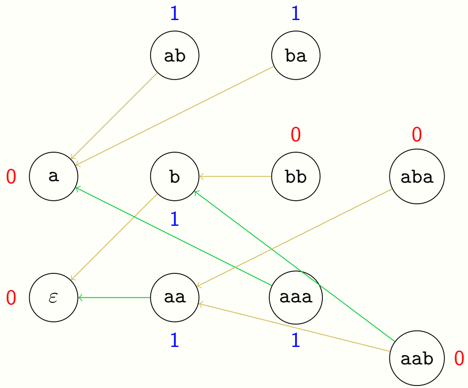
Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



The quantity $|u|_a + 2|u|_b$ decreases by 2 after each move.

A first example: the game $\mathtt{a}^2, \mathtt{b}$

Rules: $\mathtt{aa} \to \varepsilon$ and $\mathtt{b} \to \varepsilon$
We construct the DAG starting with small words:



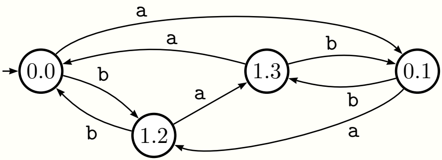
The quantity $|u|_a + 2|u|_b$ decreases by 2 after each move.
$\to \mathcal{G}(u) = 0$ iff $|u|_a + 2|u|_b \bmod 4 \in \{0, 1\}$

The game $\mathtt{a}^2, \mathtt{b}$ is rational

Theorem D., Marsault, P., Rigo, 2020

The game $\{a^2, b\}$ has two classes of Grundy values $\mathcal{L}_0$ and $\mathcal{L}_1$, each forming a rational language.

DFA computing $S(u) = (|u|_a + 2|u|_b) \bmod 4$:



State $g.s$ stands for Grundy value g and $S(u) = s$

Rationality of rewriting games?

Theorem D., Marsault, P., Rigo, 2020

Let G be the rewriting game $\{\mathtt{a}^{k_1}, \mathtt{a}^{k_2}, \cdots, \mathtt{b}^{\ell_1}, \mathtt{b}^{\ell_2}, \cdots\}$, where $1 < k_1 \leq k_2 \leq \ldots$ and $1 < \ell_1 \leq \ell_2 \leq \ldots$.
The language $\mathcal{L}_0$ formed by the $\mathcal{P}$-positions of G is <u>not</u> rational.

Rationality of rewriting games?

Theorem D., Marsault, P., Rigo, 2020

Let G be the rewriting game $\{\mathtt{a}^{k_1}, \mathtt{a}^{k_2}, \cdots, \mathtt{b}^{\ell_1}, \mathtt{b}^{\ell_2}, \cdots\}$, where $1 < k_1 \leq k_2 \leq \ldots$ and $1 < \ell_1 \leq \ell_2 \leq \ldots$.
The language $\mathcal{L}_0$ formed by the $\mathcal{P}$-positions of G is <u>not</u> rational.

Proof idea:

- Intersection of $\mathcal{L}_0$ with

$$L = \mathtt{b}^{\ell_1 - 1}(\mathtt{a}\mathtt{b}^{\ell_1 - 1})^*(\mathtt{b}\mathtt{a}^{k_1 - 1})^*.$$

Rationality of rewriting games?

Theorem D., Marsault, P., Rigo, 2020

Let G be the rewriting game $\{\mathtt{a}^{k_1}, \mathtt{a}^{k_2}, \cdots, \mathtt{b}^{\ell_1}, \mathtt{b}^{\ell_2}, \cdots\}$, where $1 < k_1 \leq k_2 \leq \ldots$ and $1 < \ell_1 \leq \ell_2 \leq \ldots$.
The language $\mathcal{L}_0$ formed by the $\mathcal{P}$-positions of G is not rational.

Proof idea:

- Intersection of $\mathcal{L}_0$ with

$$L = \mathtt{b}^{\ell_1 - 1}(\mathtt{ab}^{\ell_1 - 1})^*(\mathtt{ba}^{k_1 - 1})^*.$$

- By induction: $\mathtt{b}^{\ell_1 - 1}(\mathtt{ab}^{\ell_1 - 1})^{\mathtt{i}}(\mathtt{ba}^{k_1 - 1})^{\mathtt{j}}$ from L is $\mathcal{P}$ iff $i \geq j$.
- The intersection of L and $\mathcal{L}_0$ is not rational, and thus $\mathcal{L}_0$ is not rational.

Games with two rules

Theorem D.,Marsault, P.,Rigo, 2020

The game $\{\mathtt{a}^k, \mathtt{b}^\ell\}$ has only two Grundy classes and the associated languages $\mathcal{L}_0$ and $\mathcal{L}_1$ are context-free.

Games with two rules

Theorem D.,Marsault, P.,Rigo, 2020

The game $\{\mathtt{a}^k, \mathtt{b}^\ell\}$ has only two Grundy classes and the associated languages $\mathcal{L}_0$ and $\mathcal{L}_1$ are context-free.

Proof:

- From a word u, there is a unique terminal word $f(u)$.
- All reductions $u \longrightarrow^* f(u)$ have the same length.

Games with two rules

Theorem D.,Marsault, P.,Rigo, 2020

The game $\{\mathtt{a}^k, \mathtt{b}^\ell\}$ has only two Grundy classes and the associated languages $\mathcal{L}_0$ and $\mathcal{L}_1$ are context-free.

Proof:

- From a word u, there is a unique terminal word $f(u)$.
- All reductions $u \longrightarrow^* f(u)$ have the same length.
- $\mathcal{G}(u) = 0$ if and only if the previous sequence has an even length.
- Construction of a pushdown automaton that computes the parity of the number of reductions.

Games with two rules

Theorem D.,Marsault, P.,Rigo, 2020

The game $\{\mathtt{a}^k, \mathtt{b}^\ell\}$ has only two Grundy classes and the associated languages $\mathcal{L}_0$ and $\mathcal{L}_1$ are context-free.

Proof:

- From a word u, there is a unique terminal word $f(u)$.
- All reductions $u \longrightarrow^* f(u)$ have the same length.
- $\mathcal{G}(u) = 0$ if and only if the previous sequence has an even length.
- Construction of a pushdown automaton that computes the parity of the number of reductions.

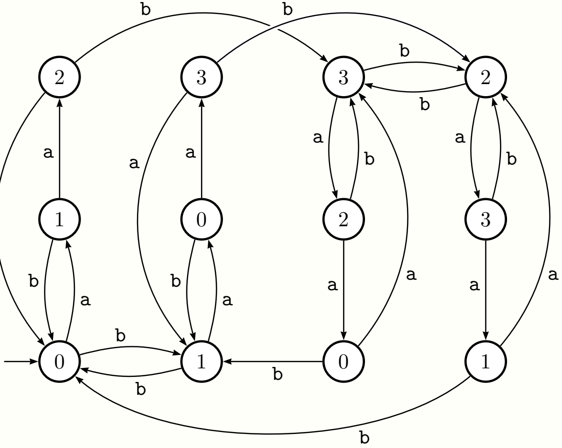
Remark: If $\ell = 1$ (or $k = 1$), the stack is not longer needed and $\mathcal{L}_0$, $\mathcal{L}_1$ are rational. The number of moves is: $|u|_{\mathtt{b}} + \lfloor \frac{|u|_{\mathtt{a}}}{k} \rfloor$.

Games with $a \to \varepsilon$ and $b \to \varepsilon$

- $\{\mathtt{a}, \mathtt{a}^{2k+1}, \mathtt{b}\}$: $\mathcal{G}$ has 2 values, $\mathcal{L}_0$ and $\mathcal{L}_1$ are rational.

Games with $a \to \varepsilon$ and $b \to \varepsilon$

- $\{\mathtt{a}, \mathtt{a}^{2k+1}, \mathtt{b}\}$: $\mathcal{G}$ has 2 values, $\mathcal{L}_0$ and $\mathcal{L}_1$ are rational.
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 12 states.



Games with $a \to \varepsilon$ and $b \to \varepsilon$

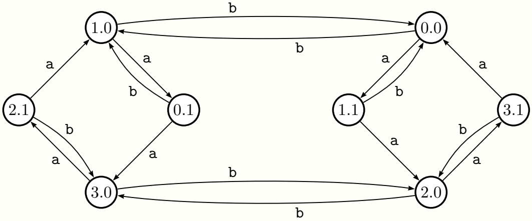
- $\{\mathtt{a}, \mathtt{a}^{2k+1}, \mathtt{b}\}$: $\mathcal{G}$ has 2 values, $\mathcal{L}_0$ and $\mathcal{L}_1$ are rational.
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 12 states.
- $\{\mathtt{a}, \mathtt{a}^4, \mathtt{b}\}$:
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{a}^3, \mathtt{b}\}$:

Games with $a \to \varepsilon$ and $b \to \varepsilon$

- $\{\mathtt{a}, \mathtt{a}^{2k+1}, \mathtt{b}\}$: $\mathcal{G}$ has 2 values, $\mathcal{L}_0$ and $\mathcal{L}_1$ are rational.
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 12 states.
- $\{\mathtt{a}, \mathtt{a}^4, \mathtt{b}\}$: open (no DFA found, $\mathcal{G} \leq 3$?)
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{a}^3, \mathtt{b}\}$:

Games with $a \to \varepsilon$ and $b \to \varepsilon$

- $\{\mathtt{a}, \mathtt{a}^{2k+1}, \mathtt{b}\}$: $\mathcal{G}$ has 2 values, $\mathcal{L}_0$ and $\mathcal{L}_1$ are rational.
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 12 states.
- $\{\mathtt{a}, \mathtt{a}^4, \mathtt{b}\}$: open (no DFA found, $\mathcal{G} \leq 3$?)
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{a}^3, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 8 states.



Games with $a \to \varepsilon$ and $b \to \varepsilon$

- $\{\mathtt{a}, \mathtt{a}^{2k+1}, \mathtt{b}\}$: $\mathcal{G}$ has 2 values, $\mathcal{L}_0$ and $\mathcal{L}_1$ are rational.
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 12 states.
- $\{\mathtt{a}, \mathtt{a}^4, \mathtt{b}\}$: open (no DFA found, $\mathcal{G} \leq 3$?)
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{a}^3, \mathtt{b}\}$: $\mathcal{G}$ has 4 values, computed by a DFA with 8 states.
- $\{\mathtt{a}, \mathtt{a}^2, \mathtt{a}^3, \mathtt{a}^4, \mathtt{b}\}$: open

$$(\max \mathcal{G}(u))_{|u|=0,1,2,\dots} =$$

$$0, 1, 2, 3, 4, 5, 5, 6, 7, 7, 7, 7, 7, 8, 9, 9, 10, 11, 11, 12, 13, 13, 13, 14$$

Question: Are the values for this game bounded?

Computing the rationality more easily

Need to examine all the $\mathcal{L}_i$'s ?

Theorem Waldmann, 2002

For an octal rewriting game:
$\mathcal{L}_0$ rational $\Rightarrow$ Grundy is bounded $+$ all $\mathcal{L}_i$ are rational.

Computing the rationality more easily

Need to examine all the $\mathcal{L}_i$'s ?

Theorem Waldmann, 2002

For an octal rewriting game:
$\mathcal{L}_0$ rational $\Rightarrow$ Grundy is bounded $+$ all $\mathcal{L}_i$ are rational.

- Remark: does not mean that
  Periodicity of $\mathcal{P}$-positions in Grundy sequence of octal game $\Rightarrow$ periodicity of the whole sequence

Computing the rationality more easily

Need to examine all the $\mathcal{L}_i$'s ?

Theorem Waldmann, 2002

For an octal rewriting game:
$\mathcal{L}_0$ rational $\Rightarrow$ Grundy is bounded + all $\mathcal{L}_i$ are rational.

- Remark: does not mean that
  Periodicity of $\mathcal{P}$-positions in Grundy sequence of octal game $\Rightarrow$ periodicity of the whole sequence
- Does not seem to hold for taking-and-merging games:

Computing the rationality more easily

Need to examine all the $\mathcal{L}_i$'s ?

Theorem Waldmann, 2002

For an octal rewriting game:
$\mathcal{L}_0$ rational $\Rightarrow$ Grundy is bounded + all $\mathcal{L}_i$ are rational.

- Remark: does not mean that
  Periodicity of $\mathcal{P}$-positions in Grundy sequence of octal game $\Rightarrow$ periodicity of the whole sequence
- Does not seem to hold for taking-and-merging games:
  - The game $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}, \mathtt{b}^2\}$ has its language $\mathcal{L}_0$ rational.

Computing the rationality more easily

Need to examine all the $\mathcal{L}_i$'s ?

Theorem Waldmann, 2002

For an octal rewriting game:
$\mathcal{L}_0$ rational $\Rightarrow$ Grundy is bounded + all $\mathcal{L}_i$ are rational.

- Remark: does not mean that
  Periodicity of $\mathcal{P}$-positions in Grundy sequence of octal game $\Rightarrow$ periodicity of the whole sequence
- Does not seem to hold for taking-and-merging games:
  - The game $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}, \mathtt{b}^2\}$ has its language $\mathcal{L}_0$ rational.
  - But

$$(\max \mathcal{G}(u))|u| = 0, 1, 2, \ldots =$$

$$0, 1, 2, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 6, 6, 6, 7, 7, 8, 8$$

Computing the rationality more easily

Need to examine all the $\mathcal{L}_i$'s ?

Theorem Waldmann, 2002

For an octal rewriting game:
$\mathcal{L}_0$ rational $\Rightarrow$ Grundy is bounded $+$ all $\mathcal{L}_i$ are rational.

- Remark: does not mean that
  Periodicity of $\mathcal{P}$-positions in Grundy sequence of octal game $\Rightarrow$ periodicity of the whole sequence
- Does not seem to hold for taking-and-merging games:
  - The game $\{\mathtt{a}, \mathtt{a}^2, \mathtt{b}, \mathtt{b}^2\}$ has its language $\mathcal{L}_0$ rational.
  - But
    $$(\max \mathcal{G}(u))|u| = 0, 1, 2, \ldots =$$
    $$0, 1, 2, 3, 3, 4, 4, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 6, 6, 6, 7, 7, 8, 8$$
  - Question : For the game $\mathtt{a}, \mathtt{a}^2, \mathtt{b}, \mathtt{b}^2$, is $\mathcal{G}$ bounded? Is there an i such that $\mathcal{L}_i$ is not rational?

Deciding $\mathcal{P}$-position of a rewriting game is not decidable

Theorem DMPR,2020

Given a rewriting game and a rational language $\mathcal{L}_R$ of starting positions, it is undecidable to determine if there exists a $\mathcal{P}$-position in $\mathcal{L}_R$.

Deciding $\mathcal{P}$-position of a rewriting game is not decidable

Theorem DMPR,2020

Given a rewriting game and a rational language $\mathcal{L}_R$ of starting positions, it is undecidable to determine if there exists a $\mathcal{P}$-position in $\mathcal{L}_R$.

Conjecture

Given a rewriting game, it is undecidable to determine if the $\mathcal{P}$-positions are rational.

Conclusion

- Rewriting games generalize a large set of combinatorial games with a nice equivalence between periodicity and regularity.
- Some very simple games are not solved yet like $\{\mathtt{a}, \mathtt{a}^4, \mathtt{b}\}$.
- What happens if we add the rule $\mathtt{a} \rightarrow \mathtt{b}$?
- Is there some relation between the DAG, the rules and the DFA ?
- Which games are (not) rewriting games?

Conclusion

- Rewriting games generalize a large set of combinatorial games with a nice equivalence between periodicity and regularity.
- Some very simple games are not solved yet like $\{\mathtt{a}, \mathtt{a}^4, \mathtt{b}\}$.
- What happens if we add the rule $\mathtt{a} \to \mathtt{b}$?
- Is there some relation between the DAG, the rules and the DFA ?
- Which games are (not) rewriting games?

Thank you !