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

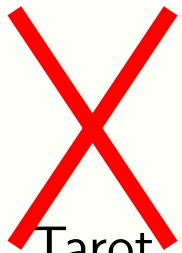
Berlekamp, Conway and Guy, *Winning Ways*, 1981

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

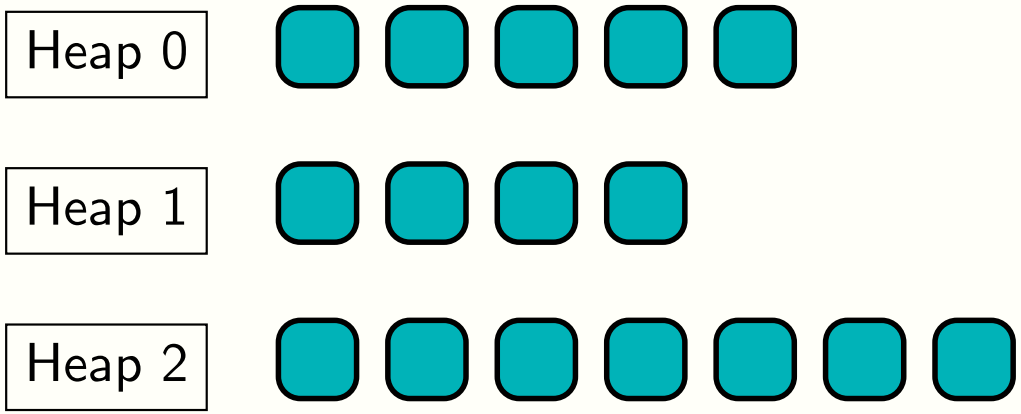
Go

“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.

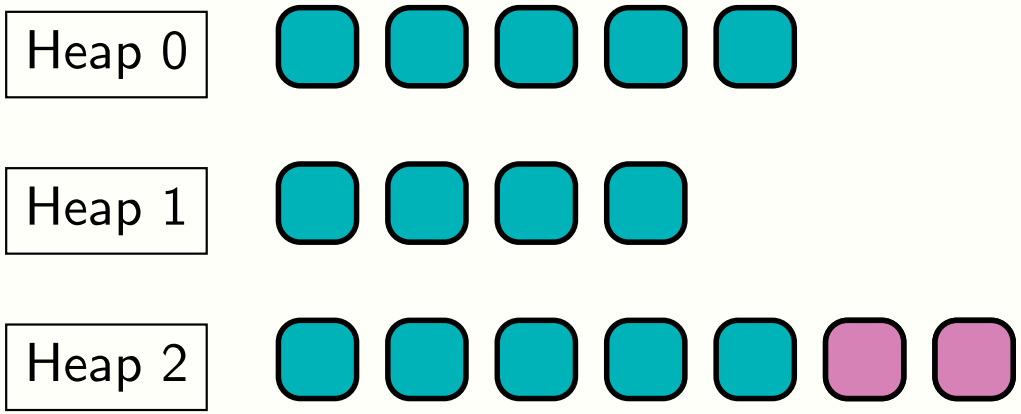


“Taking and breaking” games

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.

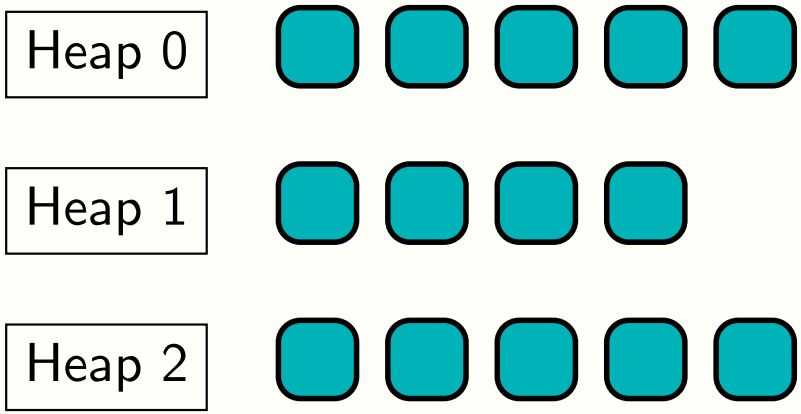


“Taking and breaking” games

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.

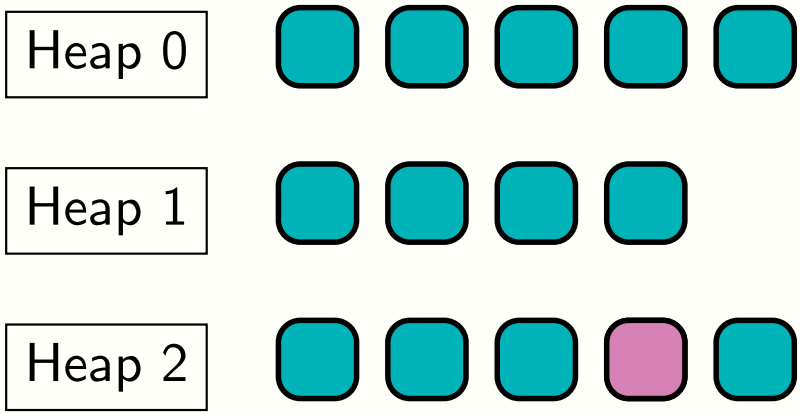


“Taking and breaking” games

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.

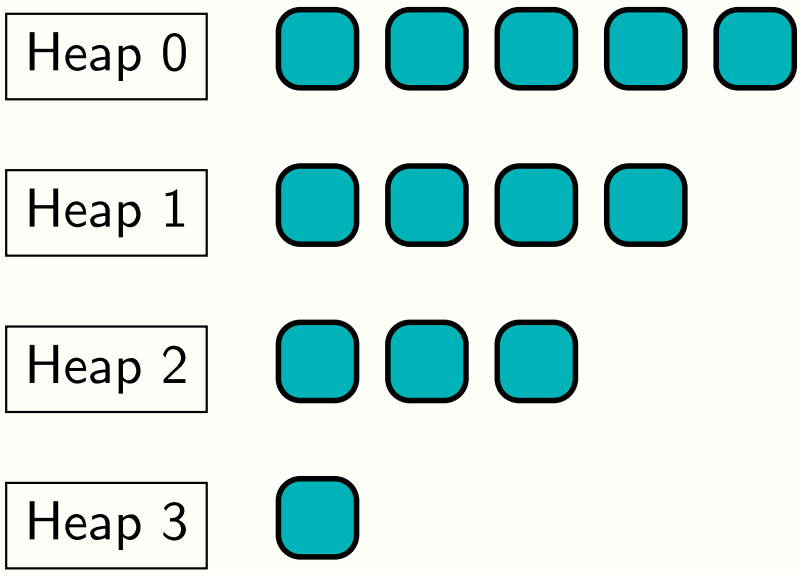


“Taking and breaking” games

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.

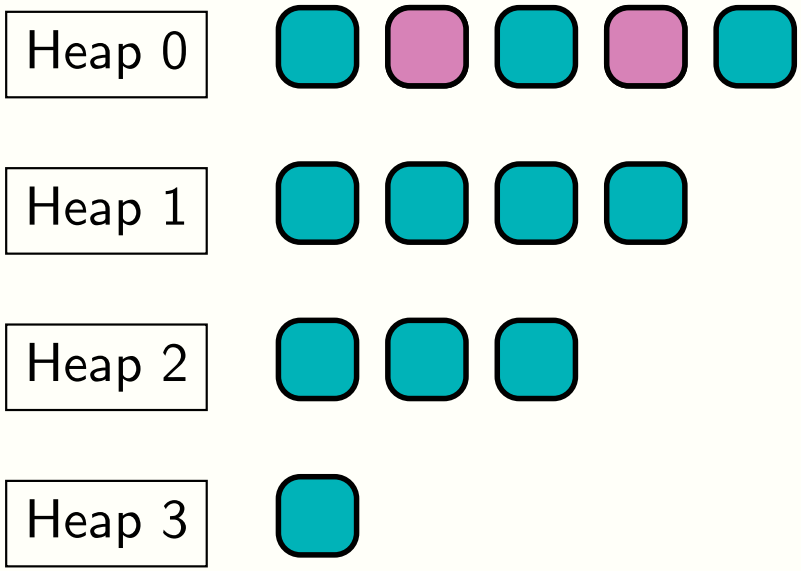


“Taking and breaking” games

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.

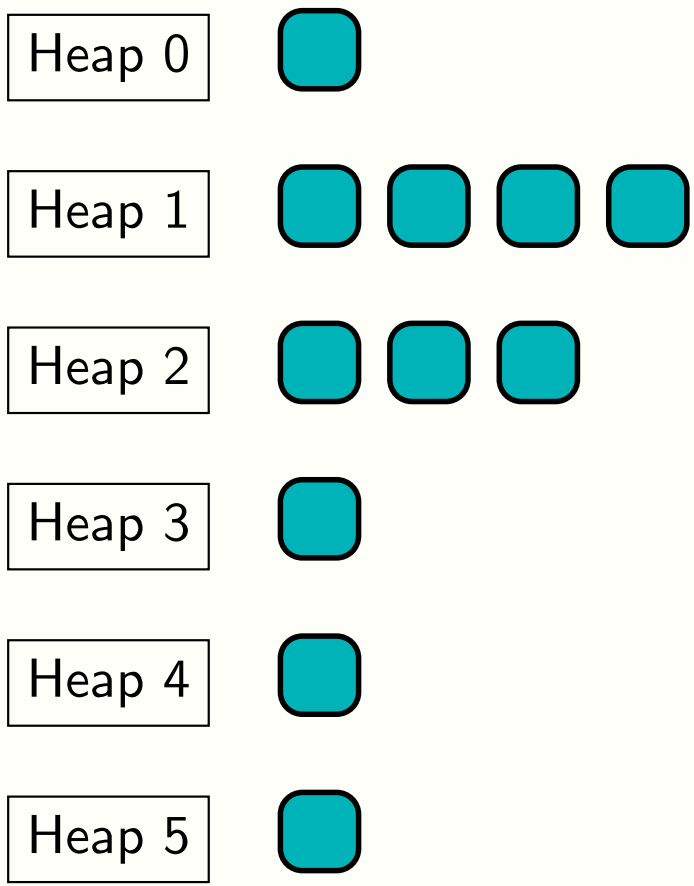


“Taking and breaking” games

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 ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

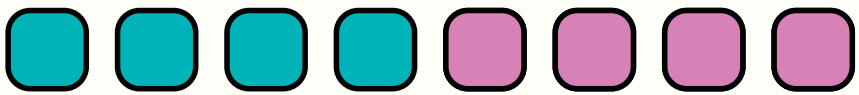
Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

Example : $S = f1; 2; 4g$



Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

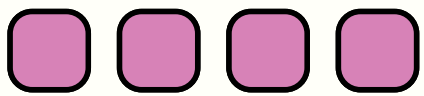
Example : $S = f1; 2; 4g$

Subtraction games

One cannot split a heap ! Subtraction game
Defined by a set S N:

> At his turn, a player removes k 2 S tokens from a heap, without breaking it.
>
> The player who cannot play loses.

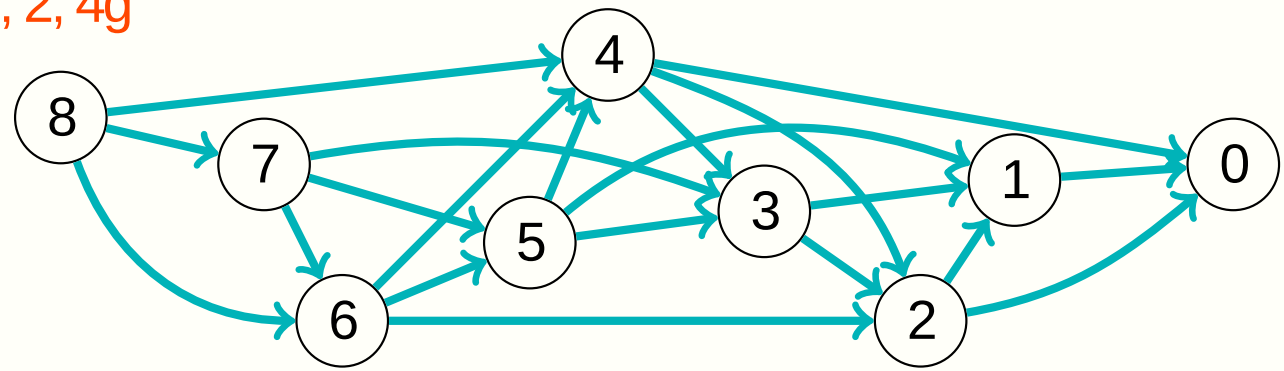
Example : $S = f1; 2; 4g$



Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.

$S = f 1; 2; 4g$

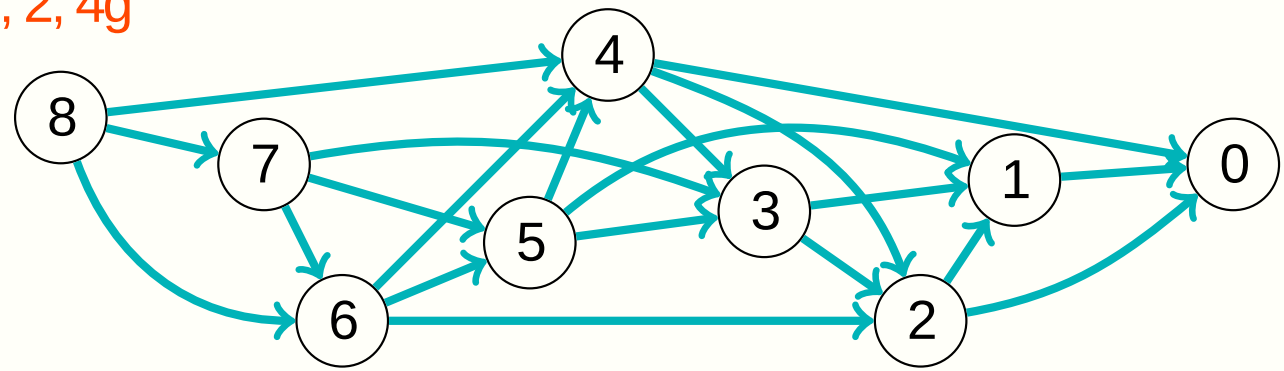


Playing in the game Moving a token along the arcs

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.

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



Playing in the game: Moving a token along the arcs

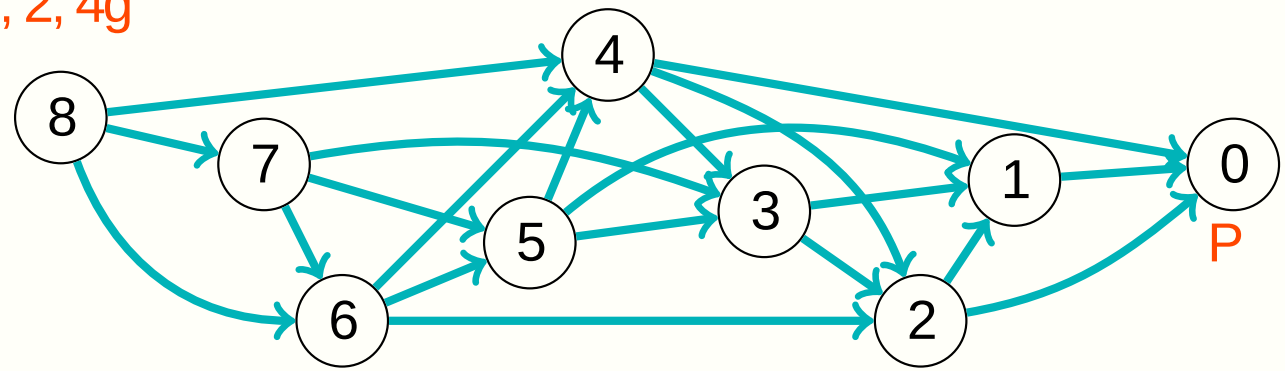
Starting from the sinks, one can determine the winner:

- N if the Next player can force the win,
- P if the Previous player can force the win.

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.

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



Playing in the game: Moving a token along the arcs

Starting from the sinks, one can determine the winner:

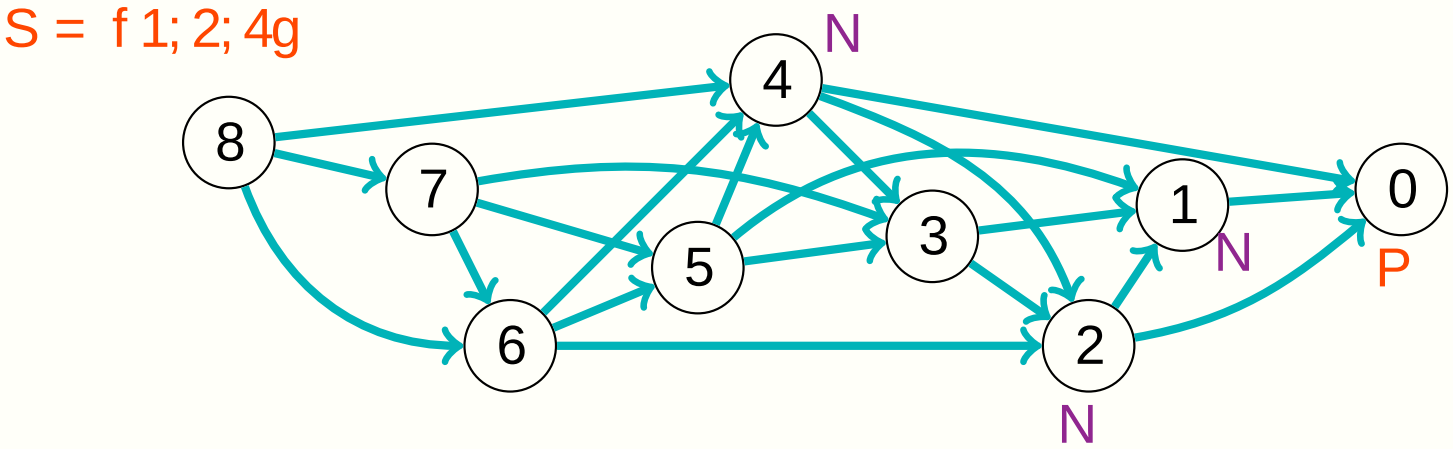
- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to N -positions

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.



Playing in the game; Moving a token along the arcs

Starting from the sinks, one can determine the winner:

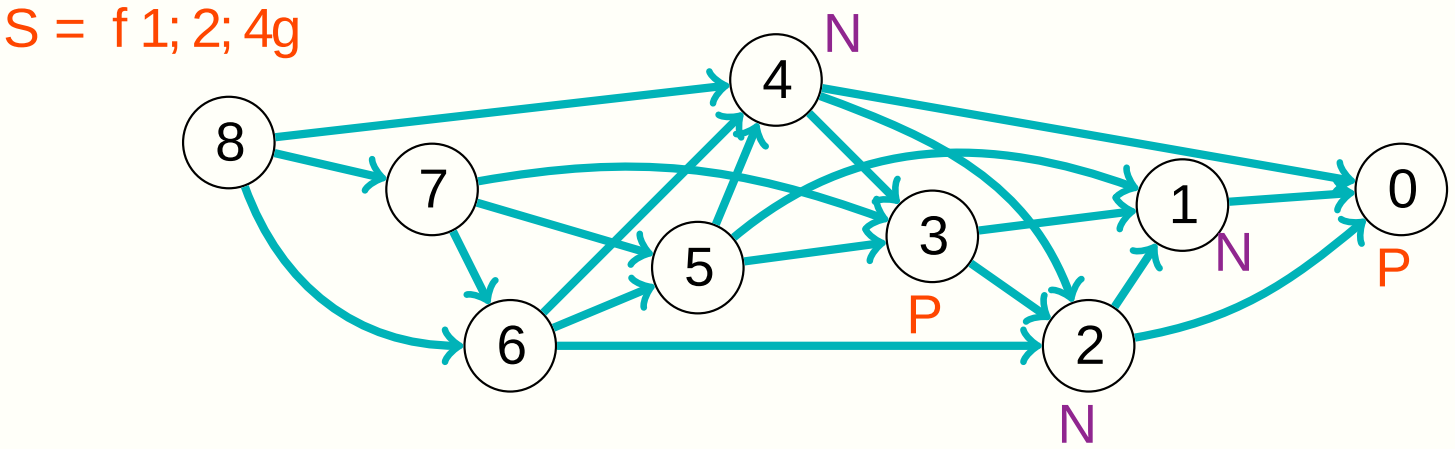
- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to N -positions

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.



Playing in the game: Moving a token along the arcs

Starting from the sinks, one can determine the winner:

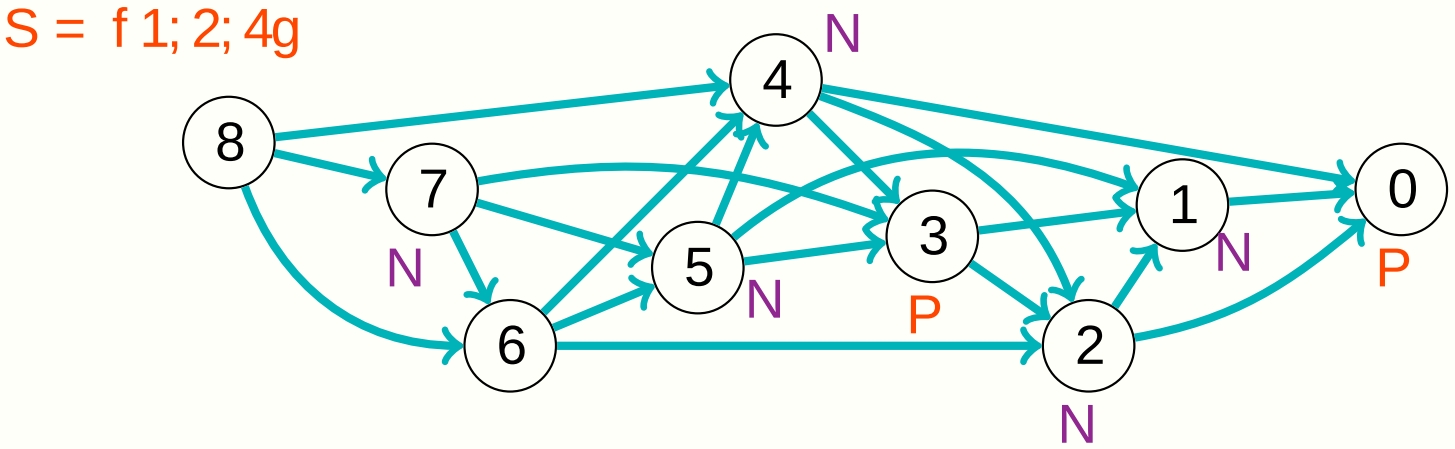
- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to N -positions

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.



Playing in the game: Moving a token along the arcs

Starting from the sinks, one can determine the winner:

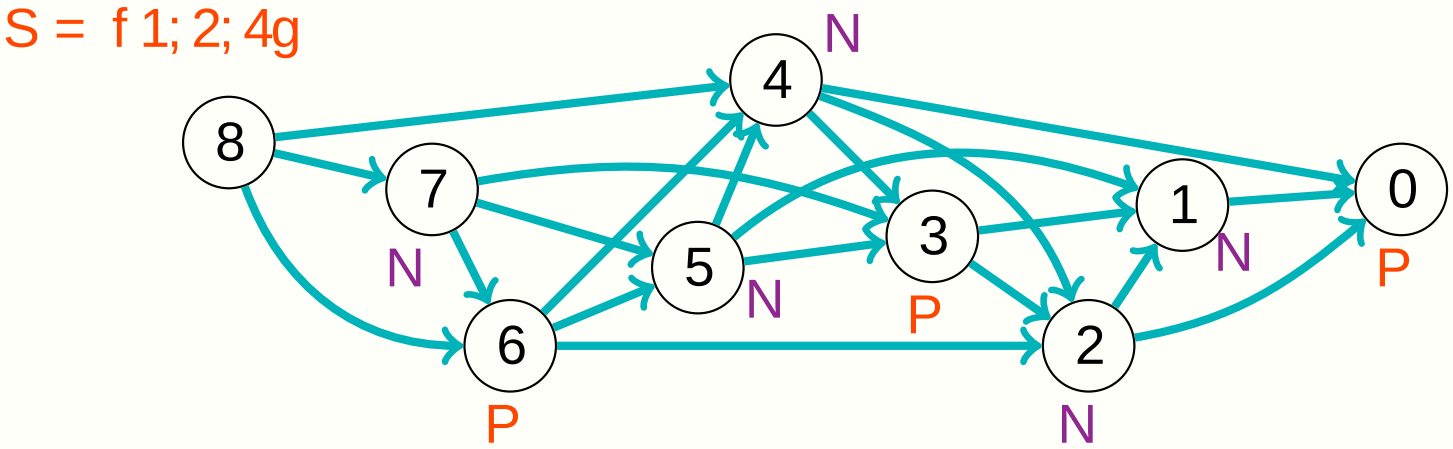
- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to N -positions

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.



Playing in the game: Moving a token along the arcs

Starting from the sinks, one can determine the winner:

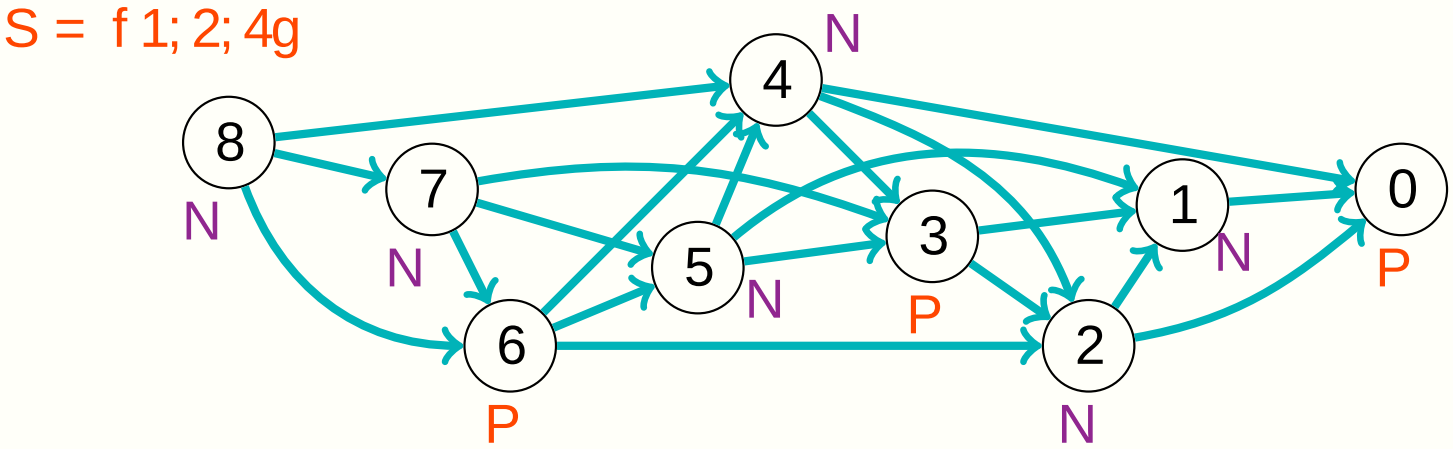
- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to N -positions

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.



Playing in the game: Moving a token along the arcs

Starting from the sinks, one can determine the winner:

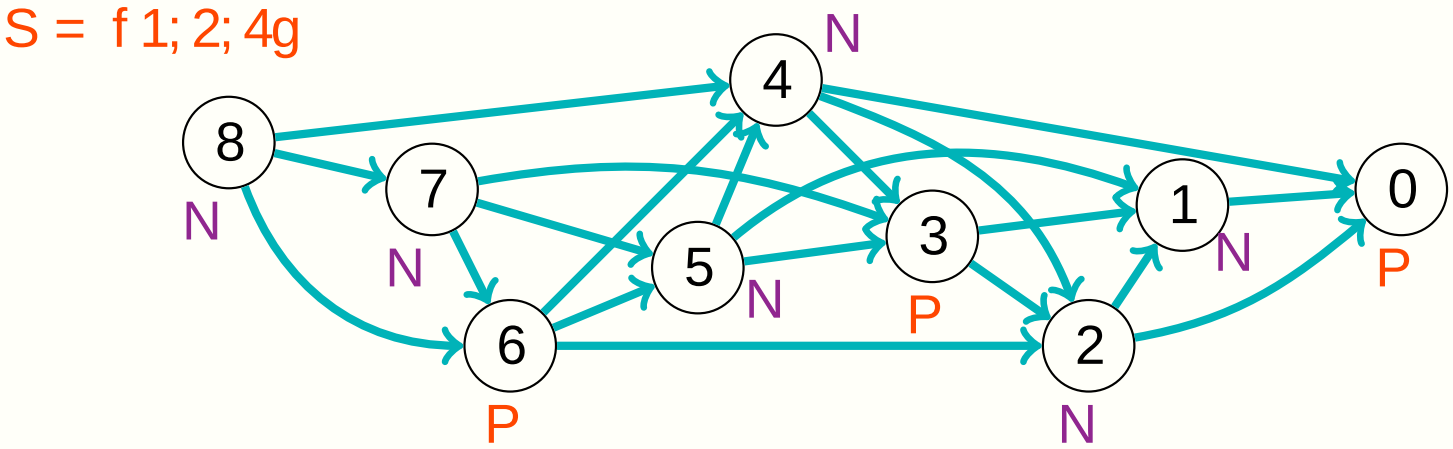
- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to N -positions

Existence of a winning strategy

Any combinatorial game can be represented by a nite DAG.



Playing in the game, Moving a token along the arcs

Starting from the sinks, one can determine the winner:

- N if the Next player can force the win,
- P if the Previous player can force the win.

From a N -position, there is always a move to a P -position.

From a P -position, all moves are to 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

Quanti ed Boolean Formula (QBF)

Input : $Q_1 x_1 Q_2 x_2 : : : Q_n x_n \; (\; x_1 ; : : : ; x_n) \; : \; Q_i \; 2 \; f 8 \; ; 9 g$

Output : Is the formula true?

A standardPspace problem

Quanti ed Boolean Formula (QBF)

Input : $Q_1 x_1 Q_2 x_2 : : : Q_n x_n \ (\ x_1 ; : : : ; x_n) \ : \ Q_i \ 2 \ f 8 \ ; 9 g$
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

Quanti ed Boolean Formula (QBF)

Input : $Q_1x_1Q_2x_2:::Q_nx_n\ (\ x_1;::::;x_n)\ :\ Q_i\ 2\ f8\ ;9g$
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 $(x_1;::::;x_n)$

Players assign boolean values x_1,...,x_n, following this order.

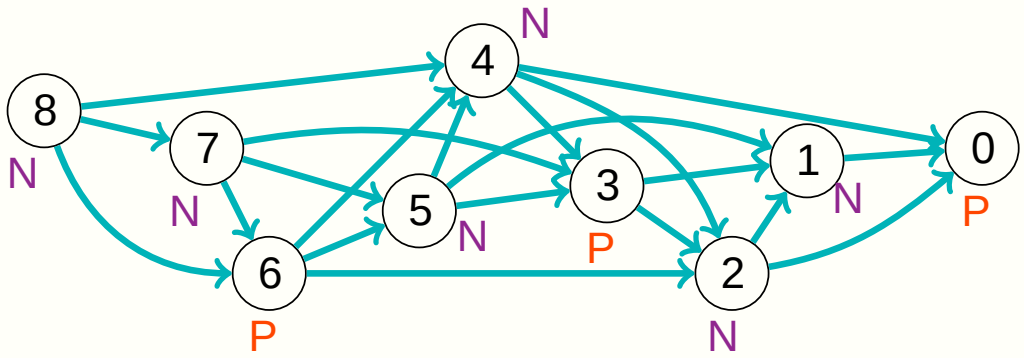
First player wins if at the end the formula is true.

Theorem Schae er, 1989 and Arora,Barak, 2009

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

Polynomiality of subtraction games

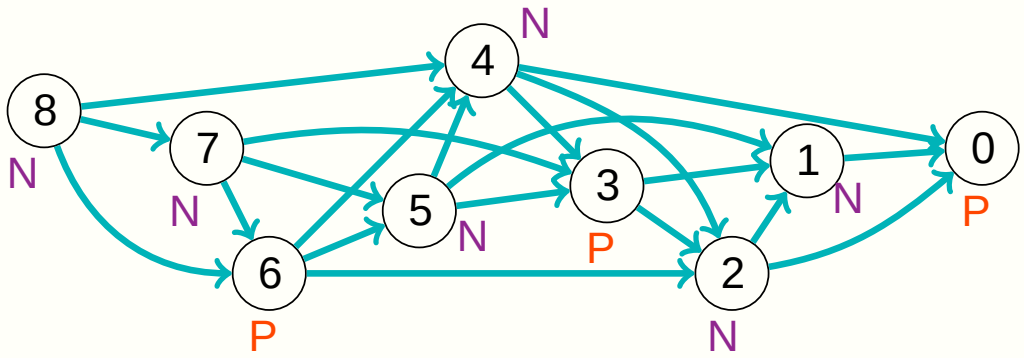
Consider the subtraction game $S = \{1, 2, 4\}$ on n token.



n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	P	N	N	P	N	N	P	N	N	P	N	N	P

Polynomiality of subtraction games

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

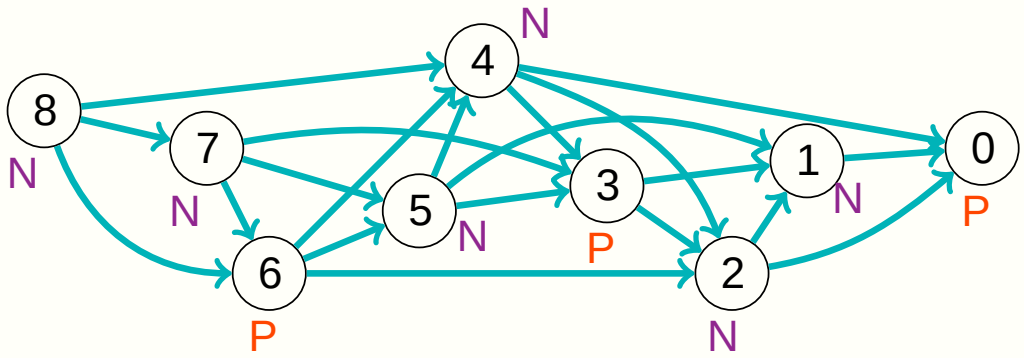


n	0	1	2	3	4	5	6	7	8	9	10	11	12
outcome	P	N	N	P	N	N	P	N	N	P	N	N	P

A position n is 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	P	N	N	P	N	N	P	N	N	P	N	N	P

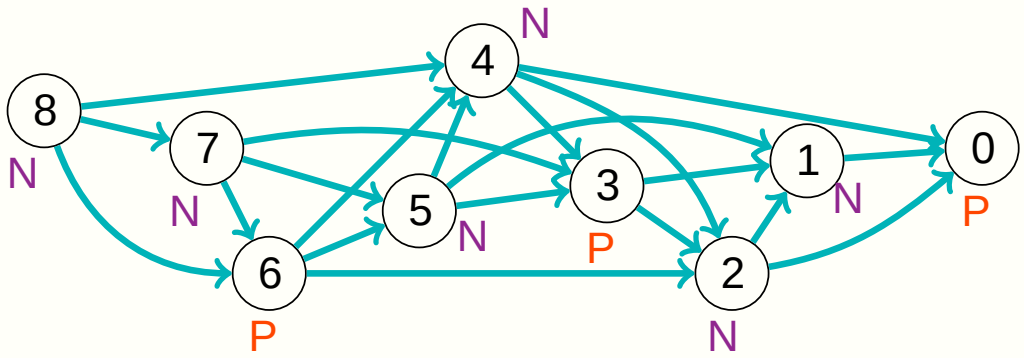
A position n is P if and only if $n \equiv 0 \bmod 3$.

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	P	N	N	P	N	N	P	N	N	P	N	N	P

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

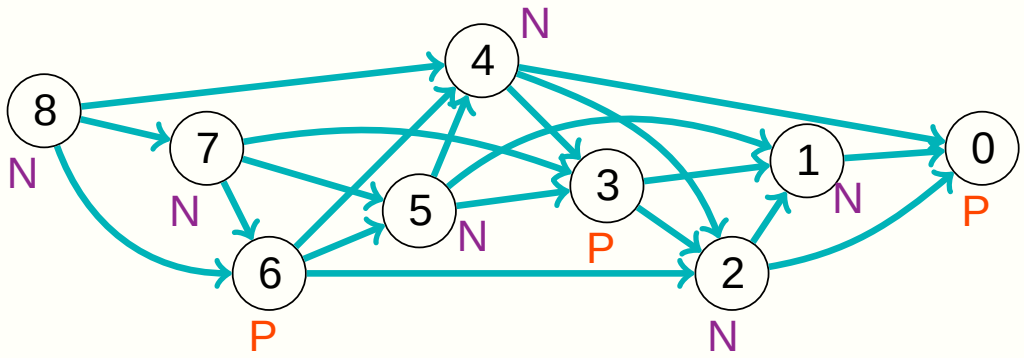
Proposition

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

! 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	P	N	N	P	N	N	P	N	N	P	N	N	P

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

Proposition

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

! 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?

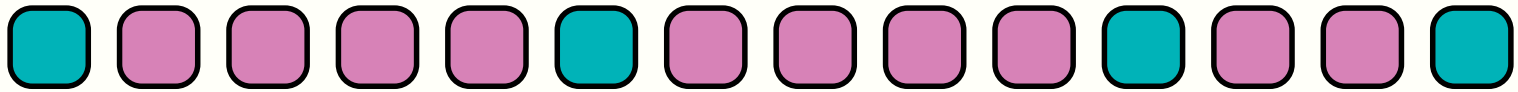
cram : Players take two tokens in a heap and can split it into two heaps.

Breaking the heaps?

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

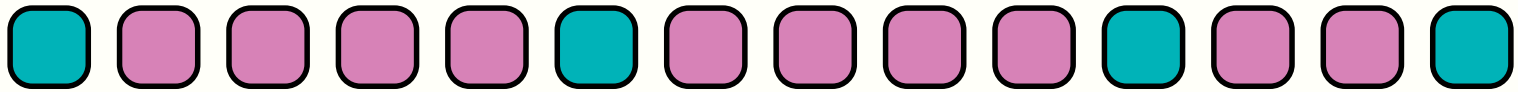
Breaking the heaps?

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



Breaking the heaps?

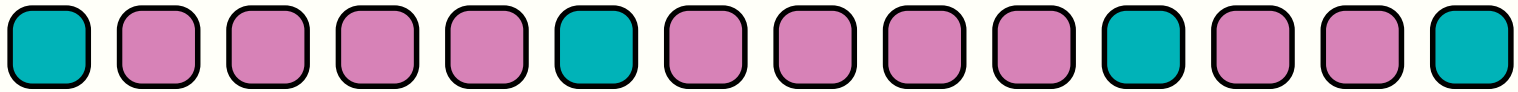
cram : Players take two tokens in a heap and can split it into two heaps.



The number of positions is exponential, how to simplify?

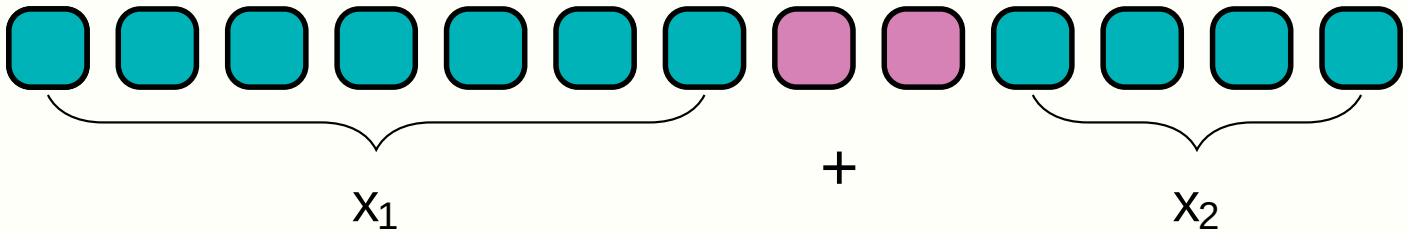
Breaking the heaps?

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



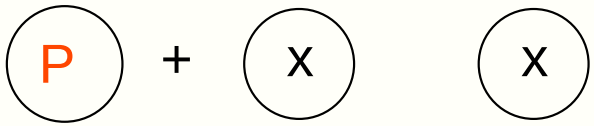
The number of positions is exponential, how to simplify?

! with sum of games



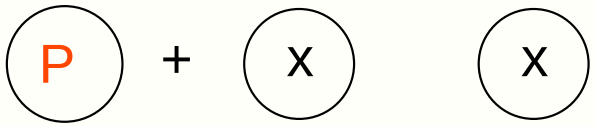
Computing the outcome of a sum

With a P -position:

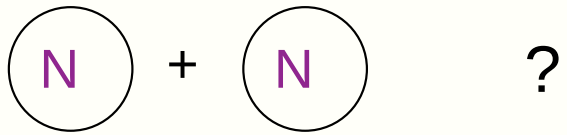


Computing the outcome of a sum

With a P -position:

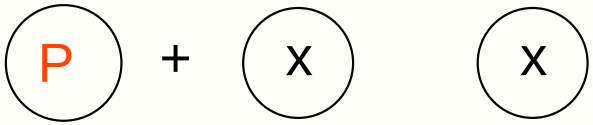


With only N -positions:

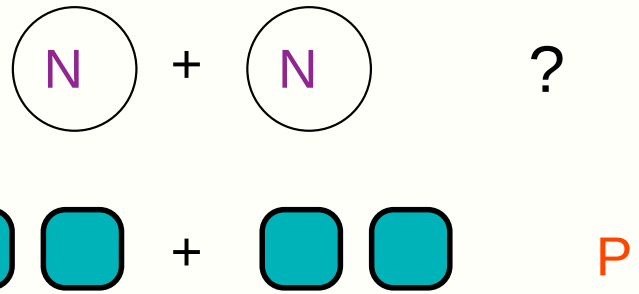


Computing the outcome of a sum

With a P -position:

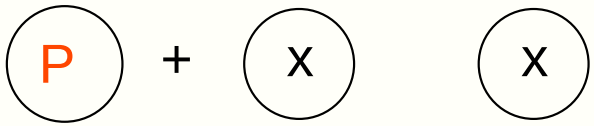


With only N -positions:

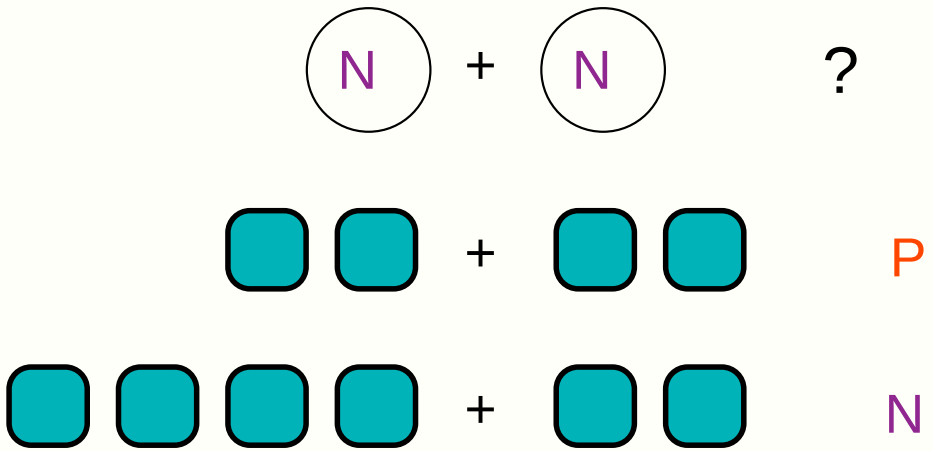


Computing the outcome of a sum

With a P -position:

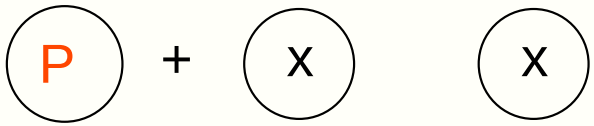


With only N -positions:

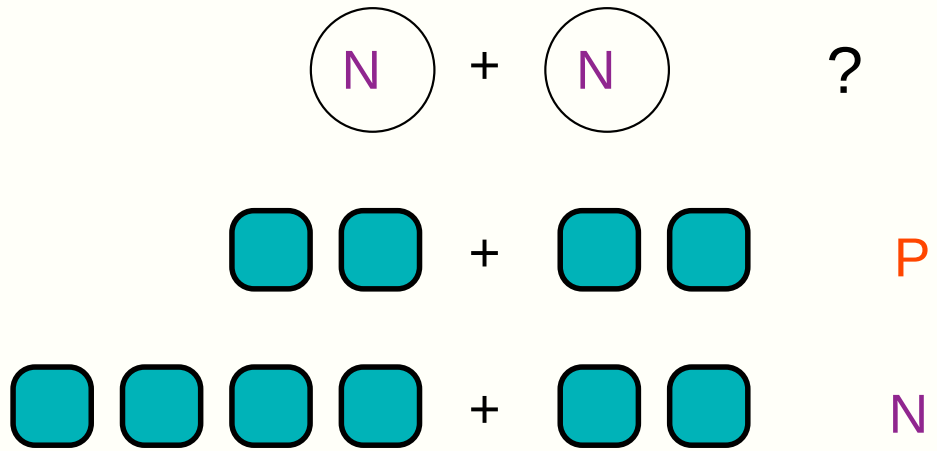


Computing the outcome of a sum

With a P -position:



With only N -positions:



+	P	N
P	P	N
N	N	P or N

Grundy values

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

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

Grundy values

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

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

The Grundy value of a position x is given by

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

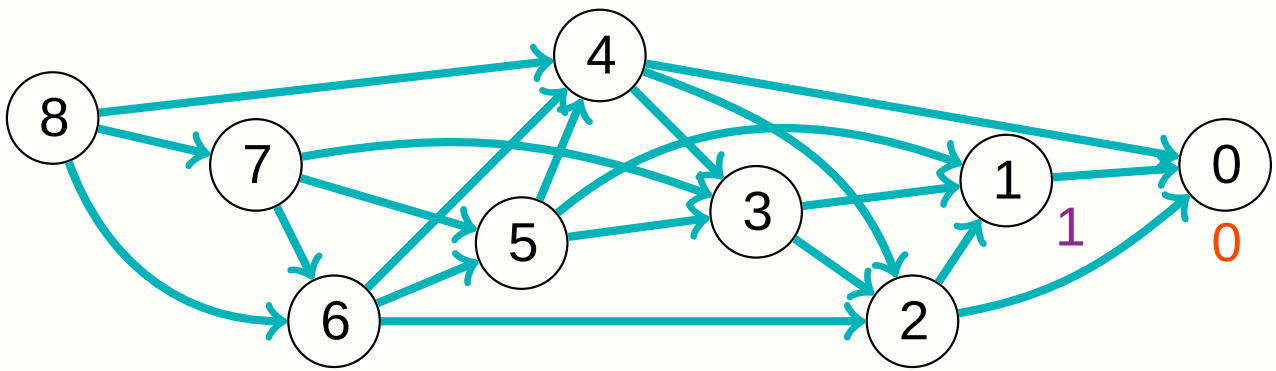
Grundy values

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

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

The Grundy value of a position x is given by

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



Proposition

$G(x) = 0$ if and only if x is P.

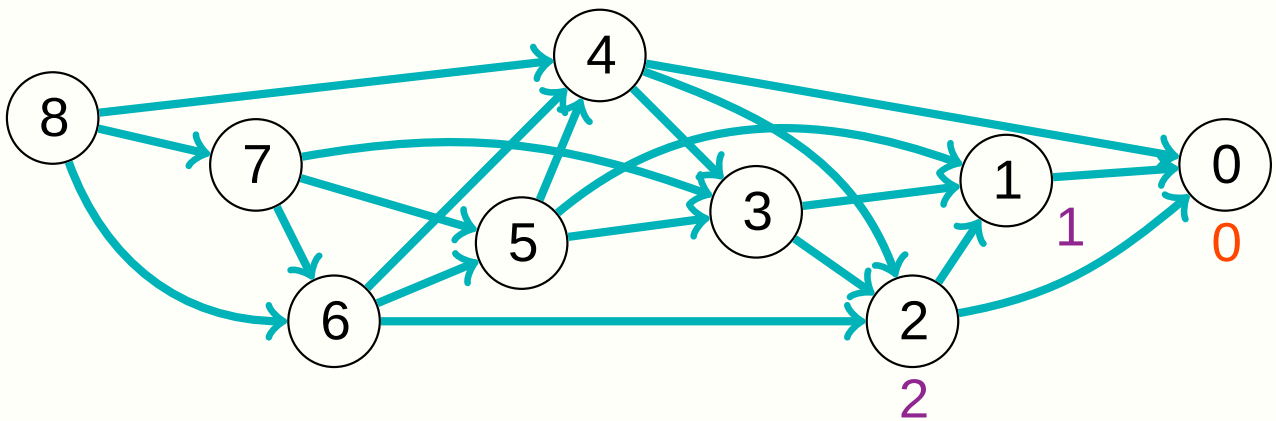
Grundy values

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

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

The Grundy value of a position x is given by

$$G(x) = \text{MeX}(\, G(N^+(x)))$$



Proposition

$G(x) = 0$ if and only if x is P.

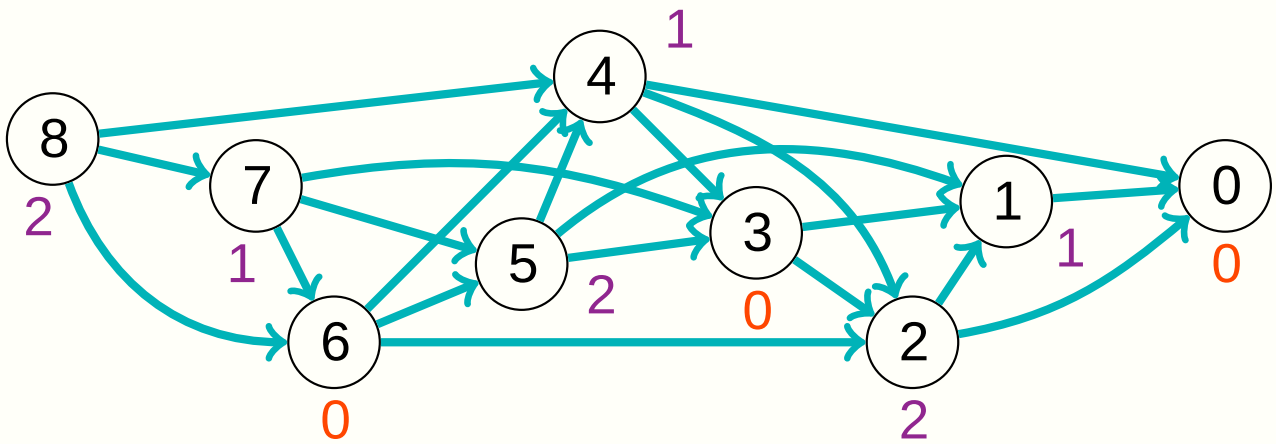
Grundy values

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

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

The Grundy value of a position x is given by

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



Proposition

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

Grundy value of the sum of games

+	P	N
P	P	N
N	N	P or N

Grundy value of the sum of games

+	P	N
P	P	N
N	N	P or N

Theorem Sprague{Grundy

Let $x_1; x_2$ be two game positions. Then:

$$G(x_1 + x_2) = G(x_1) \quad G(x_2)$$

where is the XOR operator.

Grundy value of the sum of games

+	P	N
P	P	N
N	N	P or N

Theorem Sprague{Grundy

Let $x_1; x_2$ be two game positions. Then:

$$G(x_1 + x_2) = G(x_1) \quad G(x_2)$$

where is the XOR operator.

Corollary

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

Grundy sequence

Grundy sequence: sequence of grundy values for 1,2,3,... tokens.

For the subtraction game $\{1, 2, 4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
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,... tokens.
For the subtraction game $\{1, 2, 4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
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,... tokens.
For the subtraction game $\{1; 2; 4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
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
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,... tokens.
For the subtraction game $\{1; 2; 4\}$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
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
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

$$d_0 \quad d_1 d_2 d_3 \qquad d_i \ 2\ 0;:::;7$$

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

$$d_0 \quad d_1 d_2 d_3 \qquad d_i \ 2 \ 0;:::;7$$

Examples:

- The subtraction game$f1; 2; 4g$ is the octal game $0\ 3303$.
- The gamecram corresponds to the octal game 0007.
- The game $0\ 304$ 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 nite octal game is ultimately periodic.

For the game 0 106, 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 0.106, the Grundy sequence is ultimately periodic, with a period of 328226140474, and a pre-period of 465384263797.

The Grundy sequence of the game 0.007 (James Bond Game) is conjectured to be periodic (tested up to 2^{28})

Guy's Conjecture

Conjecture Guy, 1956

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

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

The Grundy sequence of the game 0007 (James Bond Game) is conjectured to be periodic (tested up to 2^{28})

It is open for very simple games like 06!

Which octal games are in P ?

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

Second Part
Rewriting games

Rewriting games

Rewriting games (Waldmann, 2002):

Rewriting system (terminal)

Starting from a wordt , 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 $t = aabbbaabaaa$

Rewriting games

Rewriting games (Waldmann, 2002):

Rewriting system (terminal)

Starting from a wordt , 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

Rewriting games

Rewriting games (Waldmann, 2002):

Rewriting system (terminal)

Starting from a wordt , 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 $t = aabbbaabaaa$

$aabbbaabaaa \to aabbbaaaa$

Rewriting games

Rewriting games (Waldmann, 2002):

Rewriting system (terminal)

Starting from a wordt , 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 $t = aabbbaabaaa$

$aabbbaabaaa \to aabbbaaaa \to aabbbba$

Rewriting games

Rewriting games (Waldmann, 2002):

Rewriting system (terminal)

Starting from a wordt , 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 wordt , players alternate applying rules to the word.

The player who can no longer apply a rule loses.

Example : R_1 : ab ! " , R_2 : aaa ! b and t = aabbbaabaaa

aabbbaabaaa! aabbbaaaa! aabbbba! abbba! bba

The rst player can no longer play and loses.

Rewriting games

Rewriting games (Waldmann, 2002):

Rewriting system (terminal)

Starting from a wordt , players alternate applying rules to the word.

The player who can no longer apply a rule loses.

Example : R_1 : ab ! " , R_2 : aaa ! b and t = aabbbaabaaa

aabbbaabaaa! aabbbaaaa! aabbbba! abbba ! bba

The rst 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 $a \to \varepsilon$ and $aa \to b$.

aaaaaa ! aabaa

Octal games as rewrite games

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

Example: Game of CRAM can be modeled with rules aa ! " and aa ! b.

aaaaaa ! aabaa

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

- Emptying a heap of k tokens: ba^kb ! b
- Removing k without emptying: a^{k+1} ! a
- Removing k and splitting in 2: a^{k+2} ! aba

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $G(t)$.
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 $G(t)$.
Grundy class L_k: words with value k.

Theorem Waldmann, 2002

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

) Find a DFA that determines if a given word $ba^{x_1}ba^{x_2}\cdots ba^{x_n}b$ satis es $G(x_1)\ :::\ G(x_n) = k$.

There exists a DFA that computes $G(x_i)$ $8i$

Before each new x_i, we keep in memory the previous sum $G(x_1)\ :::\ G(x_{i-1})$: 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 w, there is a corresponding Grundy value $G(w)$.
Grundy class L_k: words with value k.

Theorem Waldmann, 2002

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

($\Leftarrow$) The L_k are rational.

$L_k \setminus ba^*b$ is rational.

Rational language with one letter: $\bigcup ba^{kp+\ell}b : k \in \mathbb{N}$

Partition of $\mathbb{N}$ $\Rightarrow$ periods are multiple.

Interpretation of Periodicity

For each word t, there is a corresponding Grundy value $G(t)$.
Grundy class 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 $G(t)$.
Grundy class L_k: words with value k.

Theorem Waldmann, 2002

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

Conjecture Guy, 1956

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

With di erent rules? Rational classes?

What about other types of rules?

What happens if we can delete edges? ! Taking-and-merging games

What about other types of rules?

What happens if we can delete letters? ! Taking-and-merging games

Definition

A rewriting game is said to be \taking-and-merging" if all the rules are of the form $a^k ! "$ or $b^` ! "$

Notation: $a^{k_1}; a^{k_2}; : : : ; a^{k_n}; b^{`_1}; b^{`_2}; : : : b^{`_m}$

What about other types of rules?

What happens if we can delete 0's? ! Taking-and-merging games

Definition

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

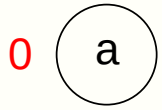
Notation: $a^{k_1}; a^{k_2}; \ldots; a^{k_n}; b^{\ell_1}; b^{\ell_2}; \ldots b^{\ell_m}$

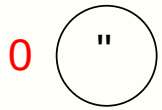
Question: Are the Grundy classes rational?

A first example: the game $a^2; b$

Rules: $aa \to \varepsilon$ and $b \to \varepsilon$

We construct the DAG starting with small words:

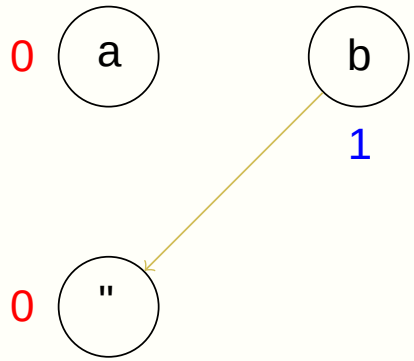




A first example: the game $a^2; b$

Rules: $aa \to \varepsilon$ and $b \to \varepsilon$

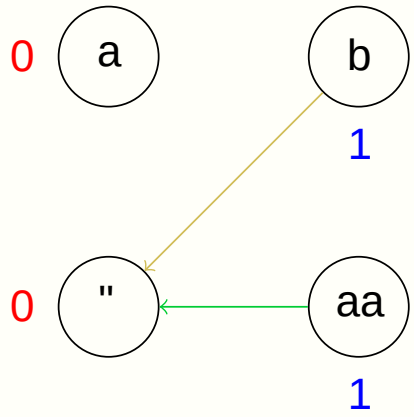
We construct the DAG starting with small words:



A first example: the game a^2, b

Rules: $aa \to \varepsilon$ and $b \to \varepsilon$

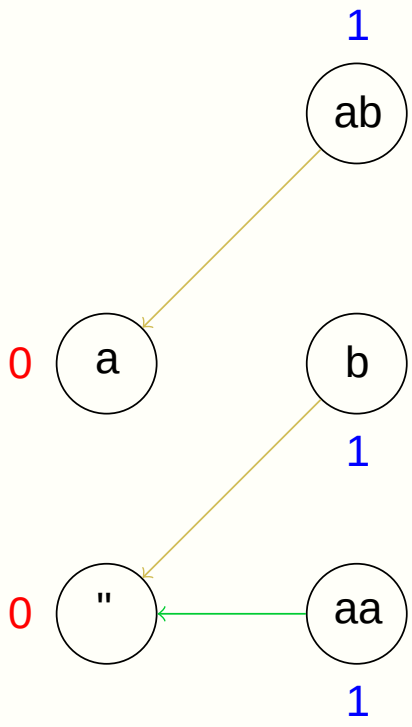
We construct the DAG starting with small words:



A first example: the game $a^2; b$

Rules: aa ! " and b ! "

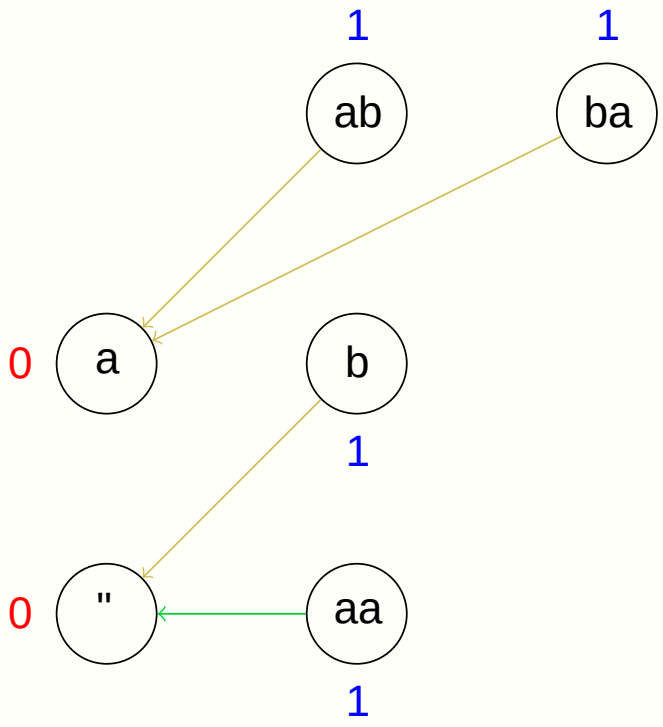
We construct the DAG starting with small words:



A first example: the game $a^2; b$

Rules: $aa \to \varepsilon$ and $b \to \varepsilon$

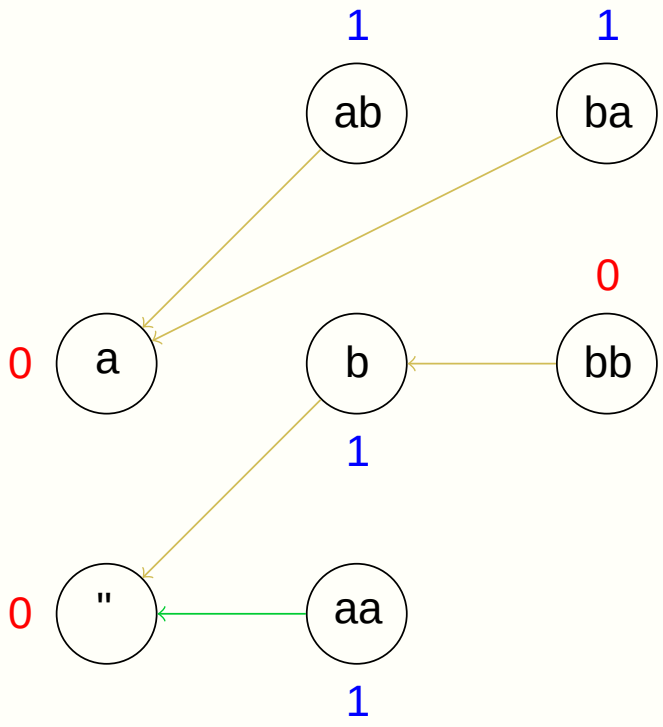
We construct the DAG starting with small words:



A first example: the game $a^2; b$

Rules: aa ! " and b ! "

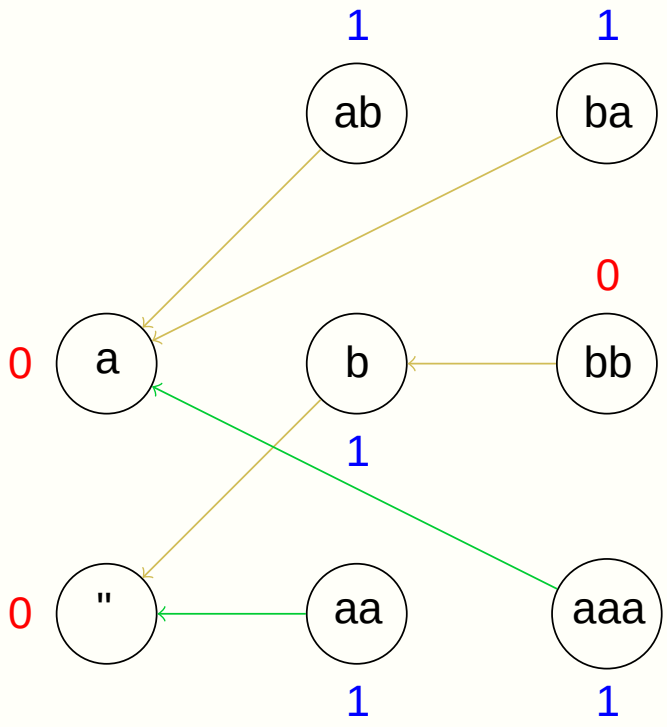
We construct the DAG starting with small words:



A first example: the game $a^2; b$

Rules: aa ! " and b ! "

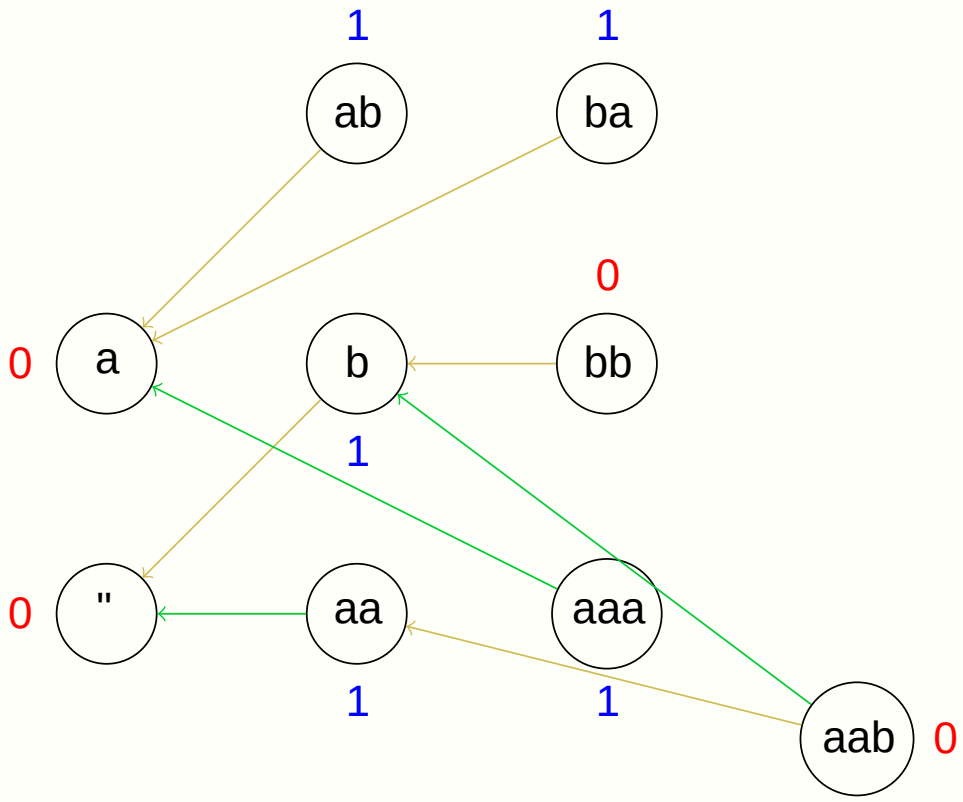
We construct the DAG starting with small words:



A first example: the game $a^2; b$

Rules: aa ! " and b ! "

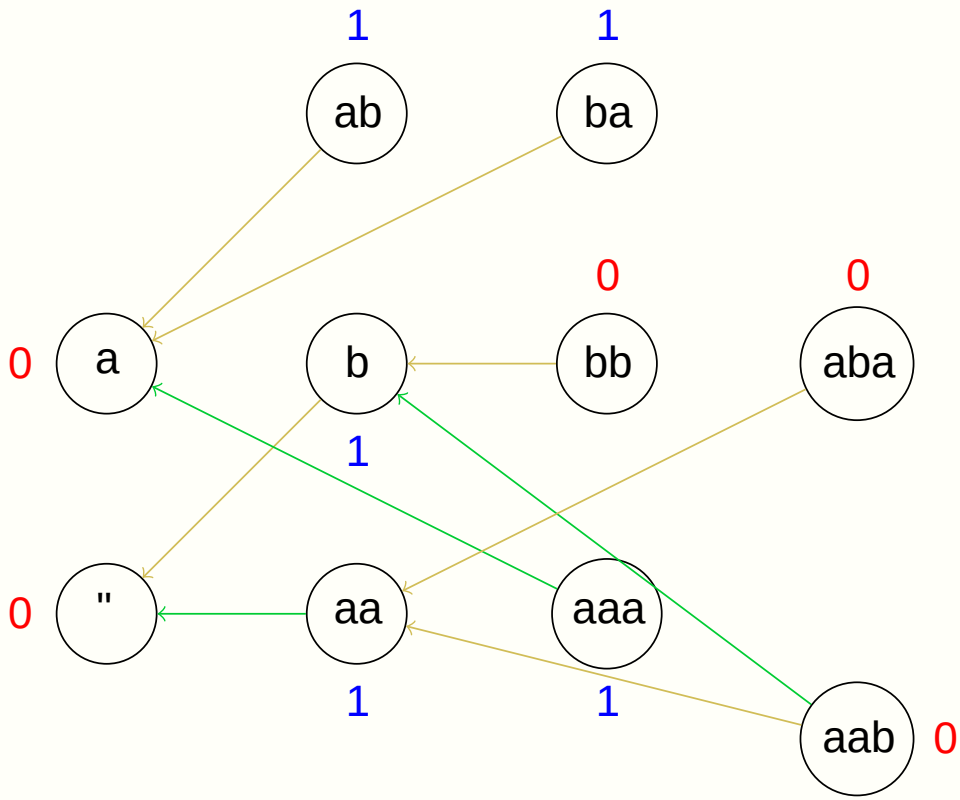
We construct the DAG starting with small words:



A first example: the game $a^2; b$

Rules: $aa \to \varepsilon$ and $b \to \varepsilon$

We construct the DAG starting with small words:

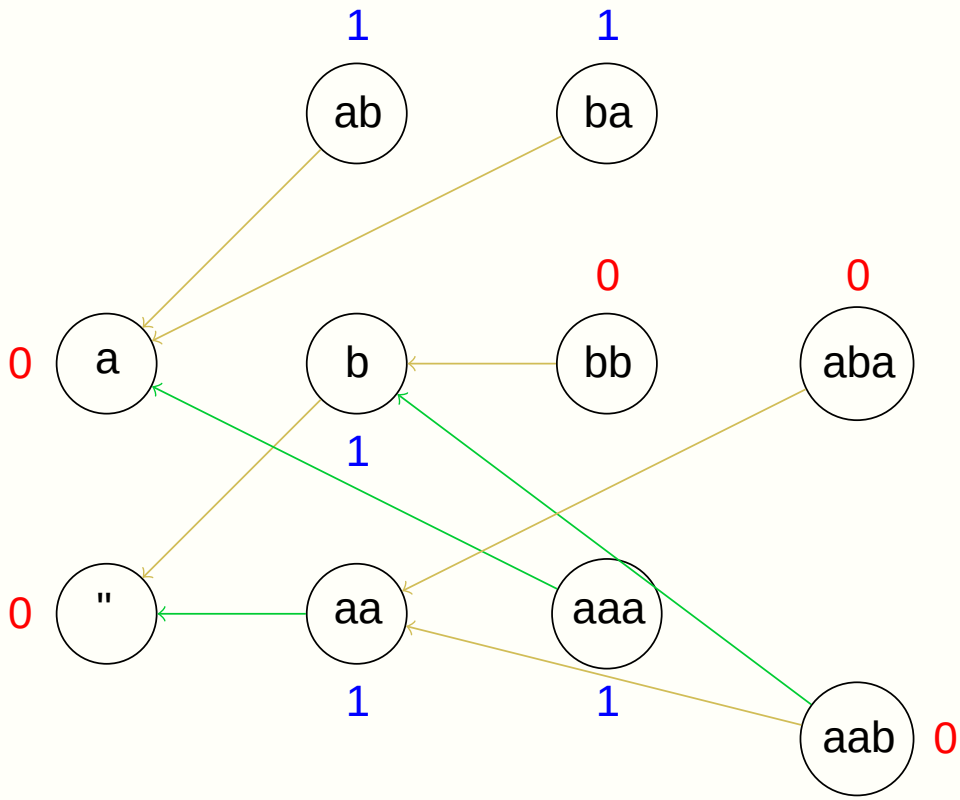


The quantity $|u|_a + 2|u|_b$ decreases by 2 after each move.

A rst example: the game $a^2; b$

Rules: $aa \to \varepsilon$ and $b \to \varepsilon$

We construct the DAG starting with small words:



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

$! G\ (u) = 0 \iff |u|_a + 2|u|_b \bmod 4 \in \{0; 1\}$

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

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

The game $\{a^2, b\}$ has two classes of Grundy values L_0 and L_1, each forming a rational language.

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

State $\overset{g:s}{\bigcirc}$ 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 $\{a^{k_1}, a^{k_2}, \ldots, b^{\ell_1}, b^{\ell_2}, \ldots\}$, where $1 < k_1 < k_2 < \cdots$ and $1 < \ell_1 < \ell_2 < \cdots$.
The language L_0 formed by the P-positions of G is not rational.

Rationality of rewriting games?

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

Let G be the rewriting game $\{a^{k_1}; a^{k_2}; \ldots; b^{\ell_1}; b^{\ell_2}; \ldots\}$, where $1 < k_1 < k_2 < \cdots$ and $1 < \ell_1 < \ell_2 < \cdots$.
The language L_0 formed by the P-positions of G is not rational.

Proof idea:

Intersection of L_0 with

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

Rationality of rewriting games?

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

Let G be the rewriting game $\{a^{k_1}; a^{k_2}; \ldots; b^{\ell_1}; b^{\ell_2}; \ldots\}$, where $1 < k_1 < k_2 < \cdots$ and $1 < \ell_1 < \ell_2 < \cdots$.
The language L_0 formed by the $\mathcal{P}$-positions of G is not rational.

Proof idea:

Intersection of L_0 with

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

By induction: $b^{\ell_1 - 1}(ab^{\ell_1 - 1})^i (ba^{k_1 - 1})^j$ from L is $\mathcal{P}$ i i j.

The intersection of L and L_0 is not rational, and thus L_0 is not rational.

Games with two rules

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

The game $\{a^k; b^\ell\}$ has only two Grundy classes and the associated languages L_0 and L_1 are context-free.

Games with two rules

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

The game $\{a^k; b^\ell\}$ has only two Grundy classes and the associated languages L_0 and L_1 are context-free.

Proof:

From a word u, there is a unique terminal word $f(u)$.

All reductions $u \to f(u)$ have the same length.

Games with two rules

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

The game $\{a^k; b^\ell\}$ has only two Grundy classes and the associated languages L_0 and L_1 are context-free.

Proof:

From a word u, there is a unique terminal word $f(u)$.

All reductions $u \to f(u)$ have the same length.

$G(u) = 0$ 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 $\{a^k; b^\ell\}$ has only two Grundy classes and the associated languages L_0 and L_1 are context-free.

Proof:

From a word u, there is a unique terminal word $f(u)$.

All reductions $u \to \cdots \to f(u)$ have the same length.

$G(u) = 0$ 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 L_0, L_1 are rational. The number of moves is $|u|_b + b\frac{|u|_a}{k}c$.

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

$\{a, a^{2k+1}, b\}$: G has 2 values, L_0 and L_1 are rational.

Games with $a!$ " and $b!$ "

$\{a, a^{2k+1}, b\}$: G has 2 values, L_0 and L_1 are rational.

$\{a, a^2, b\}$: G has 4 values, computed by a DFA with 12 states.

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

$\{a, a^{2k+1}, b\}$: G has 2 values, L_0 and L_1 are rational.

$\{a, a^2, b\}$: G has 4 values, computed by a DFA with 12 states.

$\{a, a^4, b\}$:

$\{a, a^2, a^3, b\}$:

Games with $a^! \ "$ and $b^! \ "$

$\{a, a^{2k+1}, b\}$: G has 2 values, L_0 and L_1 are rational.

$\{a, a^2, b\}$: G has 4 values, computed by a DFA with 12 states.

$\{a, a^4, b\}$: open (no DFA found, $G \quad 3$?)

$\{a, a^2, a^3, b\}$:

Games with $a \to a\,!\,"$ and $b \to b\,!\,"$

$\{a, a^{2k+1}, b\}$: G has 2 values, L_0 and L_1 are rational.

$\{a, a^2, b\}$: G has 4 values, computed by a DFA with 12 states.

$\{a, a^4, b\}$: open (no DFA found, $G \neq 3$?)

$\{a, a^2, a^3, b\}$: G has 4 values, computed by a DFA with 8 states.

Games with $a\ !\ ''$ and $b\ !\ ''$

$\{a; a^{2k+1}; b\}$: G has 2 values, L_0 and L_1 are rational.

$\{a; a^2; b\}$: G has 4 values, computed by a DFA with 12 states.

$\{a; a^4; b\}$: open (no DFA found, G 3?)

$\{a; a^2; a^3; b\}$: G has 4 values, computed by a DFA with 8 states.

$\{a; a^2; a^3; a^4; b\}$: open

$$(\max G(u))_{|u|=0;1;2;\ldots} =$$

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

Question: Are the values for this game bounded?

Computing the rationality more easily

Need to examine all the L_i's ?

Theorem Waldmann, 2002

For an octal rewriting game:
L_0 rational $\Rightarrow$ Grundy is bounded + all L_i are rational.

Computing the rationality more easily

Need to examine all the L_i's ?

Theorem Waldmann, 2002

For an octal rewriting game:
L_0 rational $\Rightarrow$ Grundy is bounded + all L_i are rational.

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

Computing the rationality more easily

Need to examine all the L_i's ?

Theorem Waldmann, 2002

For an octal rewriting game:
L_0 rational $\Rightarrow$ Grundy is bounded + all L_i are rational.

Remark: does not mean that
Periodicity of 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 L_i's ?

Theorem Waldmann, 2002

For an octal rewriting game:
L_0 rational $\Rightarrow$ Grundy is bounded + all L_i are rational.

Remark: does not mean that
Periodicity of 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 $\{a; a^2; b; b^2\}$ has its language L_0 rational.

Computing the rationality more easily

Need to examine all the L_i's ?

Theorem Waldmann, 2002

For an octal rewriting game:
L_0 rational $\Rightarrow$ Grundy is bounded + all L_i are rational.

Remark: does not mean that
Periodicity of 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 $\{a; a^2; b; b^2\}$ has its language L_0 rational.
- But

$$(\max G(u))_{|u|=n} = 0; 1; 2; \ldots =$$

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

Computing the rationality more easily

Need to examine all the L_i's ?

Theorem Waldmann, 2002

For an octal rewriting game:
L_0 rational $\Rightarrow$ Grundy is bounded + all L_i are rational.

Remark: does not mean that
Periodicity of 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 $\{a; a^2; b; b^2\}$ has its language L_0 rational.
- But

$$(\max G(u))juj = 0; 1; 2; \ldots =$$

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

- Question : For the game $a; a^2; b; b^2$, is G bounded? Is there an i such that L_i is not rational?

Deciding P -position of a rewriting game is not decidable

Theorem DMPR,2020

Given a rewriting game and a rational language L_R of starting positions, it is undecidable to determine if there exists a P -position in L_R.

Deciding P-position of a rewriting game is not decidable

Theorem DMPR,2020

Given a rewriting game and a rational language L_R of starting positions, it is undecidable to determine if there exists a P-position in L_R.

Conjecture

Given a rewriting game, it is undecidable to determine if the 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 $f\mathsf{a}; \mathsf{a}^4; \mathsf{b}g$.

What happens if we add the rule $\mathsf{a} \; ! \; \mathsf{b}$?

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 $f\mathsf{a}; \mathsf{a}^4; \mathsf{b}g$.

What happens if we add the rule $\mathsf{a} \; ! \; \mathsf{b}$?

Is there some relation between the DAG, the rules and the DFA ?

Which games are (not) rewriting games?

Thank you !