Combinatorial Games and Computability

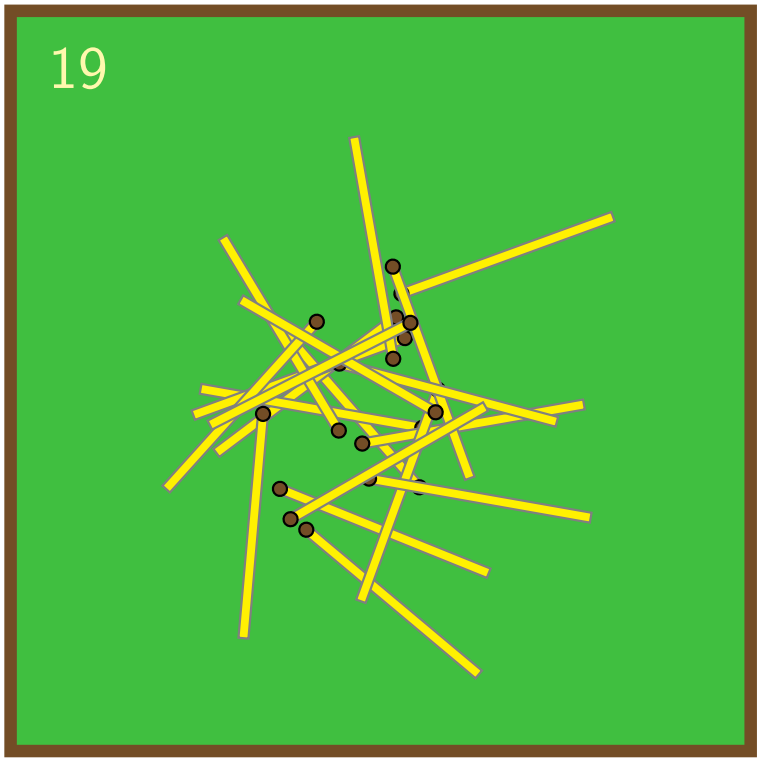
Urban Larsson,
Killam, Dalhousie University, Halifax, Canada

June 13, 2014

Thanks for the invitation!

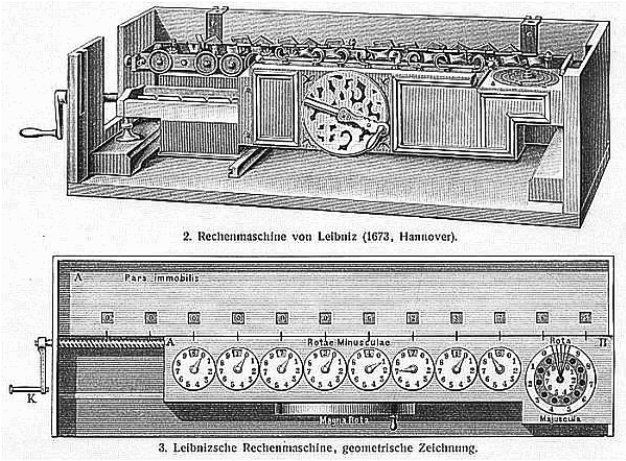
A heap game

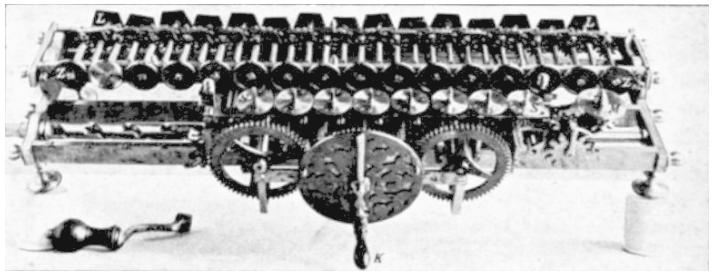
There are 19 matches on the table. Two players, Alice and Bob, alternate turns. Rules: remove 1 or 4 tokens. Last player wins. Can Alice secure a win in the first move? Let's play!



Can we compute who will win? Let's ask G.W. von Leibniz!

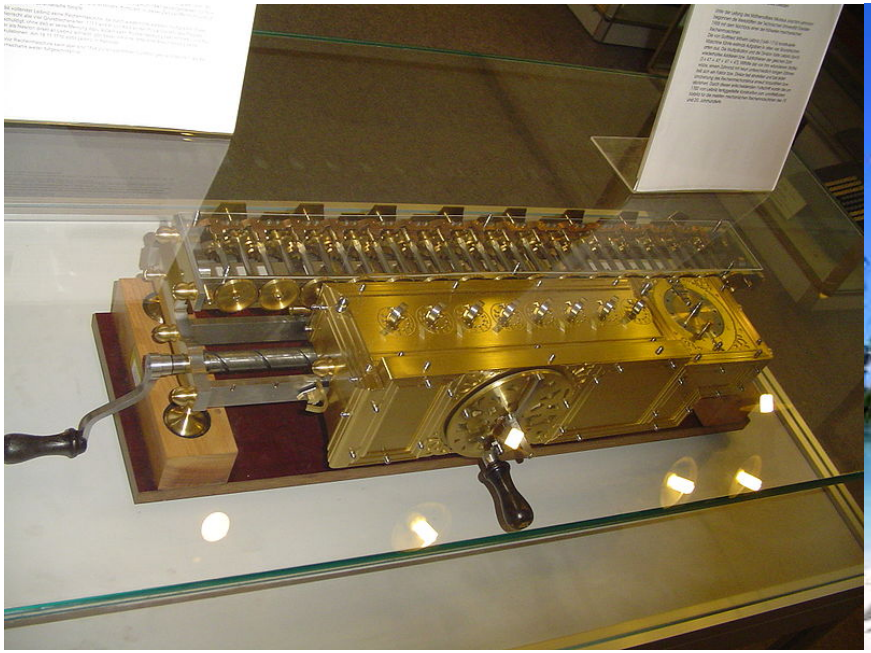
The Stepped Reckoner





"...having constructed a successful mechanical calculating machine, Leibniz dreamt of building a machine that could manipulate symbols in order to determine the truth value of any mathematical statement." (Wikipedia)

Leibniz dream: Stepped Reckoner $\leftrightarrow$ Easy Life?



2-player games as algorithms

Proposition

Alice will win, given perfect play.

2-player games as algorithms

Proposition

Alice will win, given perfect play.

Proof by induction

Divide 19 by 5. The remainder is 4 matches. Remove them. Bob moves from a heap of 15 matches,

2-player games as algorithms

Proposition

Alice will win, given perfect play.

Proof by induction

Divide 19 by 5. The remainder is 4 matches. Remove them. Bob moves from a heap of 15 matches, its remainder being 0 matches.

2-player games as algorithms

Proposition

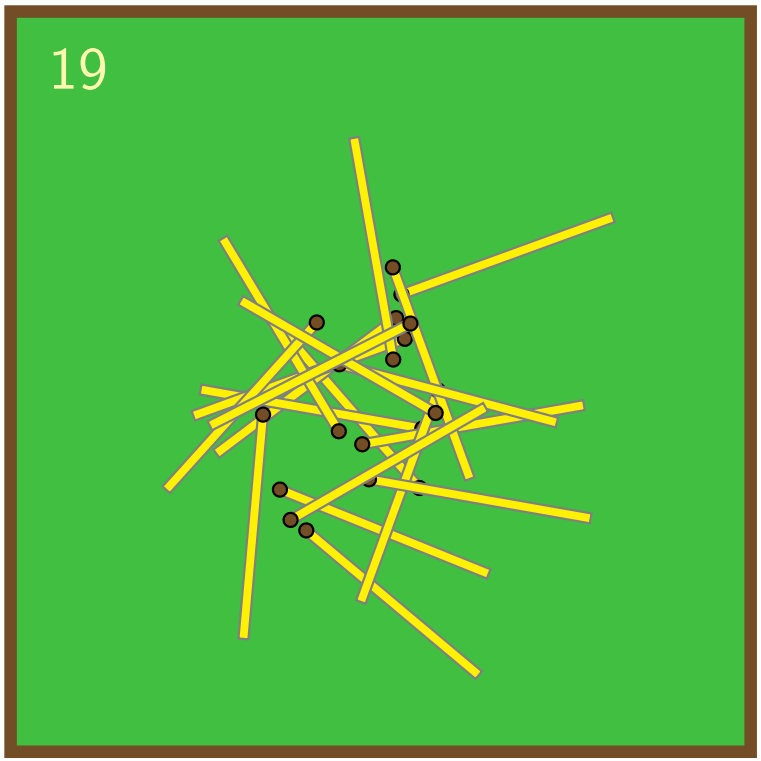
Alice will win, given perfect play.

Proof by induction

Divide 19 by 5. The remainder is 4 matches. Remove them. Bob moves from a heap of 15 matches, its remainder being 0 matches. By subtracting either 1 or 4, the new remainder will be either 4 or 1 matches...

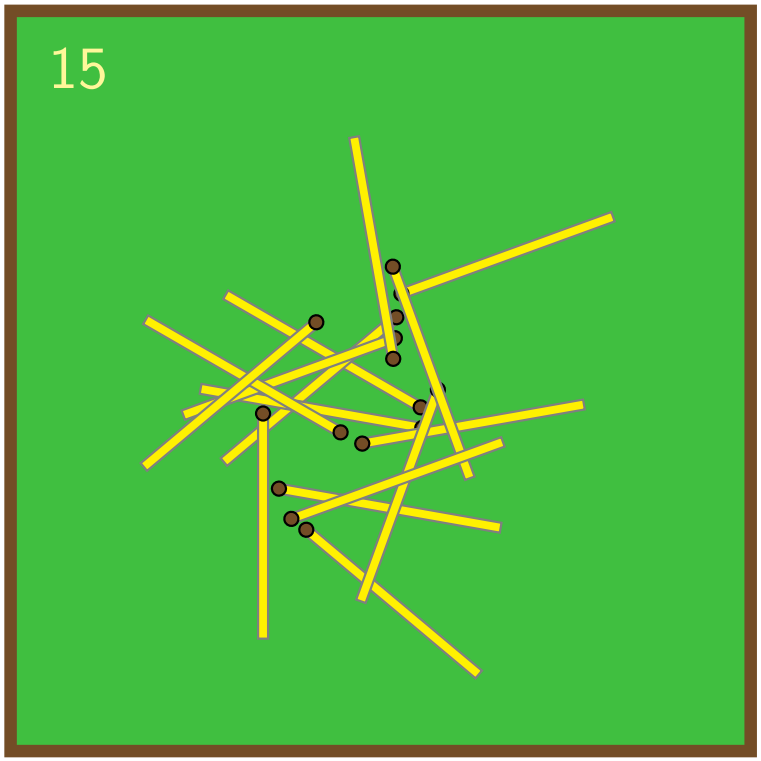
A heap game

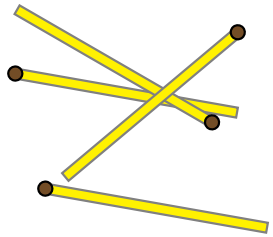
Alice starts from 19 matches.



A heap game

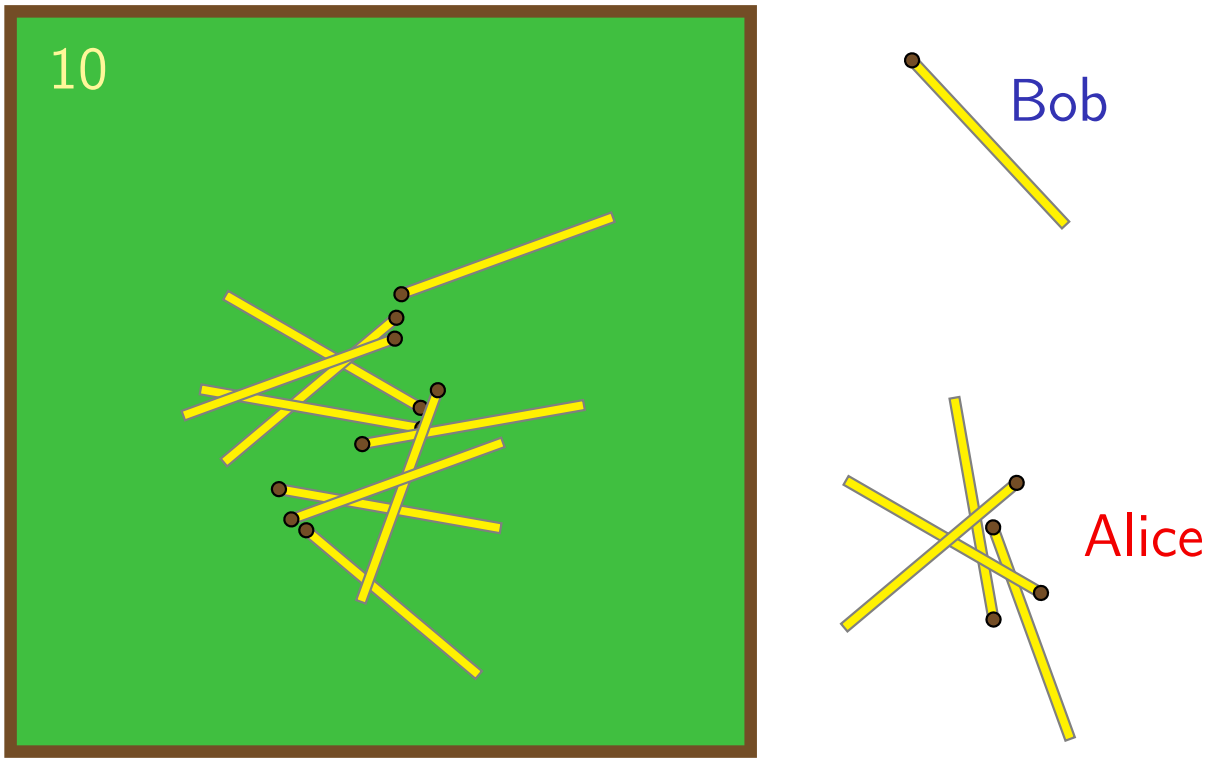
Removing 4 matches gives residue 0. Bob will move from 15.





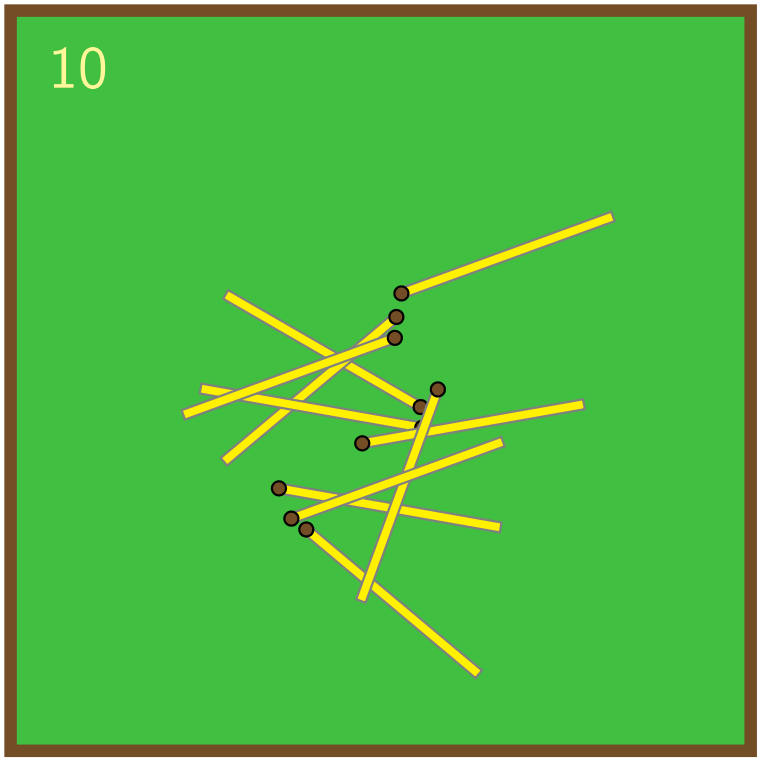
A heap game

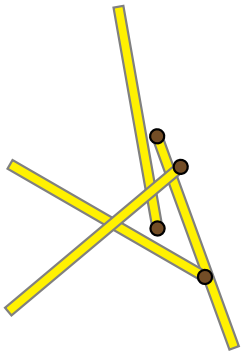
If Bob removes 1 match, then Alice will remove 4.



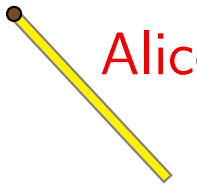
A heap game

And vice versa.





Bob



Alice

A heap game

If Alice rather starts from a heap of 18, then her Reckoner tells her to remove 1 match. In conclusion, the safe positions have residue either 0 or 2 when dividing by 5.

Theorem

Any one heap game with a finite number of move options can be step reckoned. The dividend is $\leq 2^x$, where x is the largest number of matches that can be removed.

Proof.

There are at most 2^x combinations (patterns) of safe and unsafe positions in a range of heap sizes $y, y+1, \ldots, y+x-1$. Whenever one such pattern reappears the future will become periodic. $\square$

What about Leibniz dream in general?

David Hilbert's Entscheidungsproblem (1928):

- Is there an algorithm that decides whether a given statement is provable from the axioms of first order logic?
- Is there an algorithm that evaluates whether two given propositions are equivalent?

What about Leibniz dream in general?

David Hilbert's Entscheidungsproblem (1928):

- Is there an algorithm that decides whether a given statement is provable from the axioms of first order logic?
- Is there an algorithm that evaluates whether two given propositions are equivalent?
- He believed "yes", but...

Church and Turing delivered bad news...



In the 1930s Alan Turing and Alonzo Church independently proved that such algorithms do not exist. Turing's approach was to reduce the halting problem of his universal "machine" to the Entscheidungsproblem. He had already established that the halting problem is algorithmically undecidable: there is no Turing machine that can take as input the code of another Turing machine and decide whether it will halt or not for a given input.

Church and Turing delivered bad news...



In the 1930s Alan Turing and Alonzo Church independently proved that such algorithms do not exist. Turing's approach was to reduce the halting problem of his universal "machine" to the Entscheidungsproblem. He had already established that the halting problem is algorithmically undecidable: there is no Turing machine that can take as input the code of another Turing machine and decide whether it will halt or not for a given input. Emil Post nearly proved it via his Tag Productions in the 1920s.

...so we must get back to work!

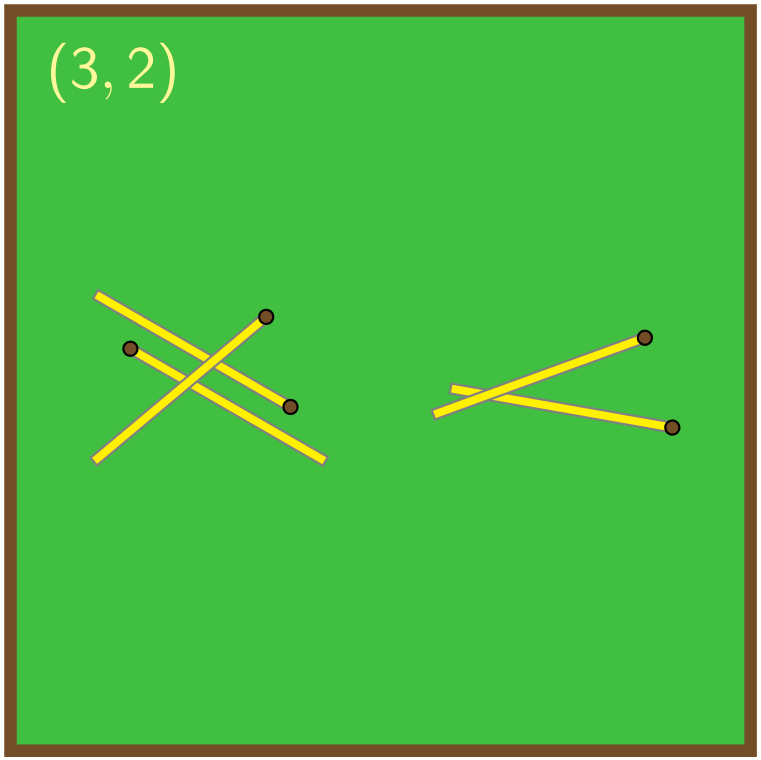
"Human computers", Wikipedia.

Decidability and heap games

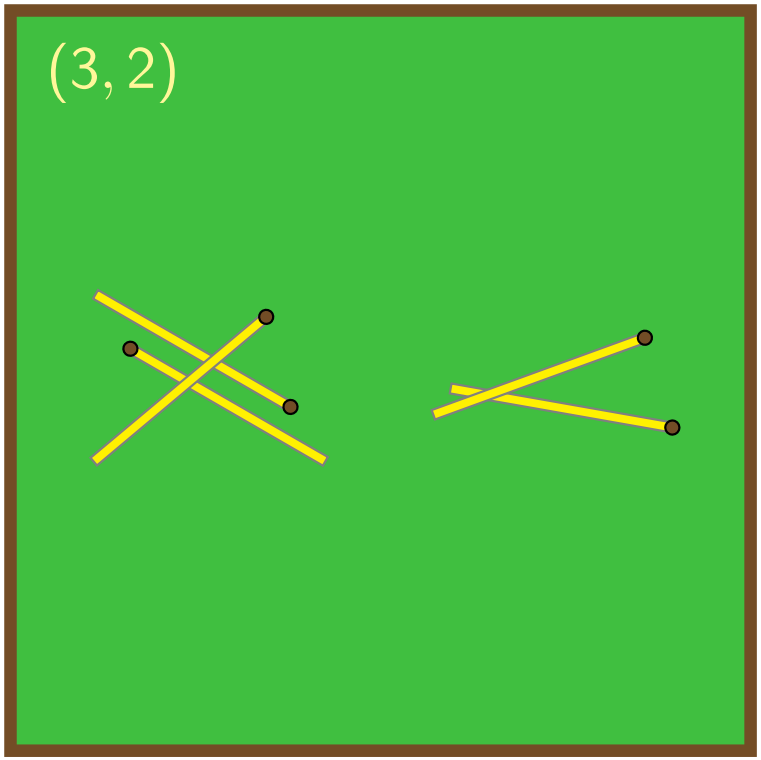
Play on several heaps. The total number of matches must decrease. It can increase on individual heaps. For example, play on two ordered heaps, with rules: either

- remove 1 match from Heap1 and 3 matches from Heap2, or
- remove 2 matches from Heap1 and add 1 match to Heap2.
- denote this game $\mathcal{M} = \{(-1, -3), (-2, 1)\}$.

Game rules $\mathcal{M} = \{(-1,-3),(-2,1)\}$

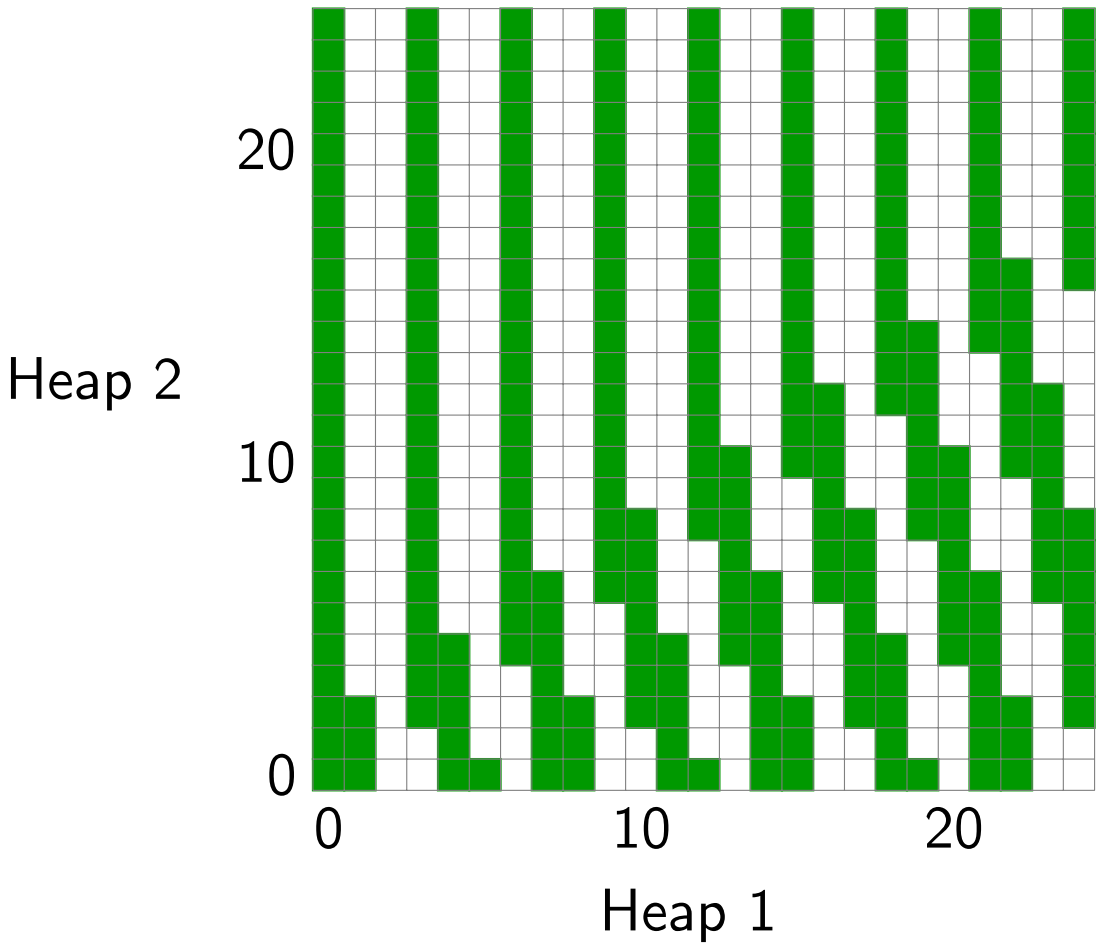


Game rules $\mathcal{M} = \{(-1,-3),(-2,1)\}$

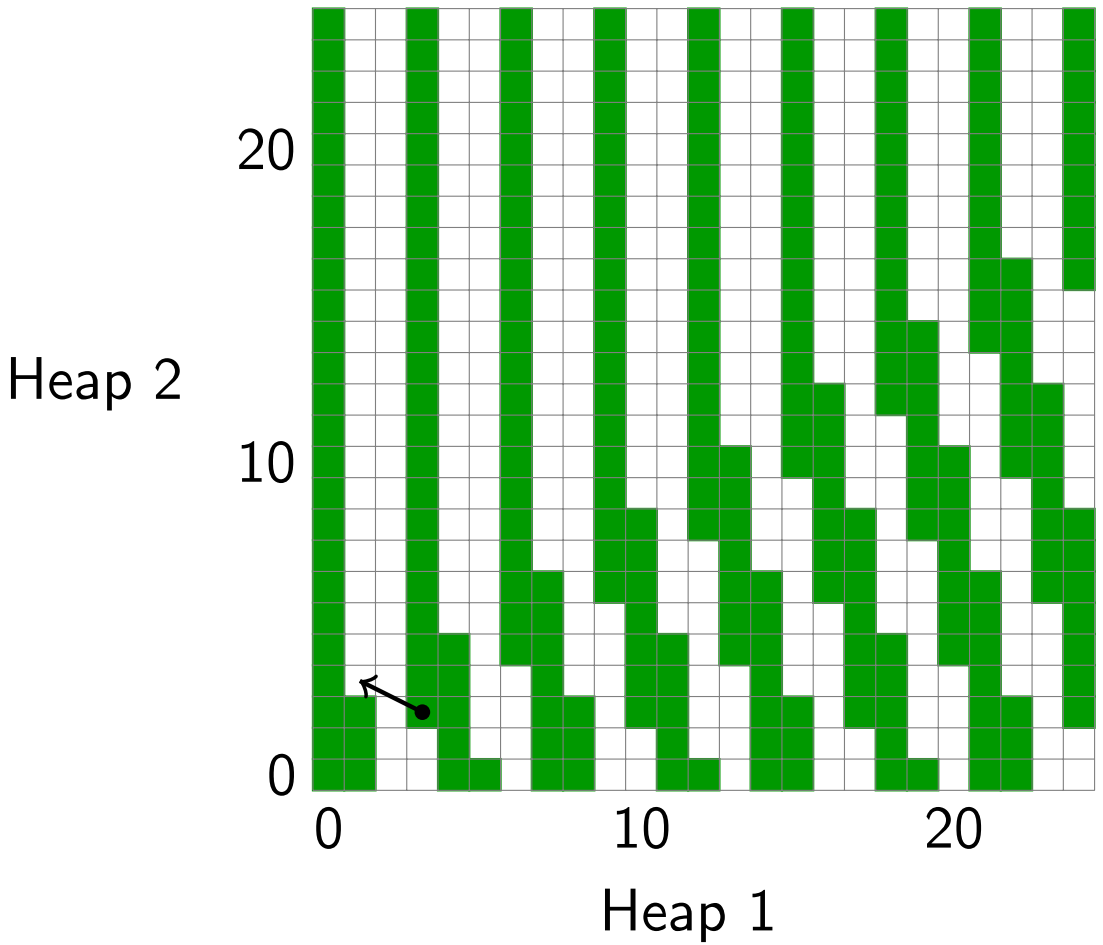


Any computer (with sufficient memory) can compute the status of any given starting position. Who wins from $(3,2)$? How?

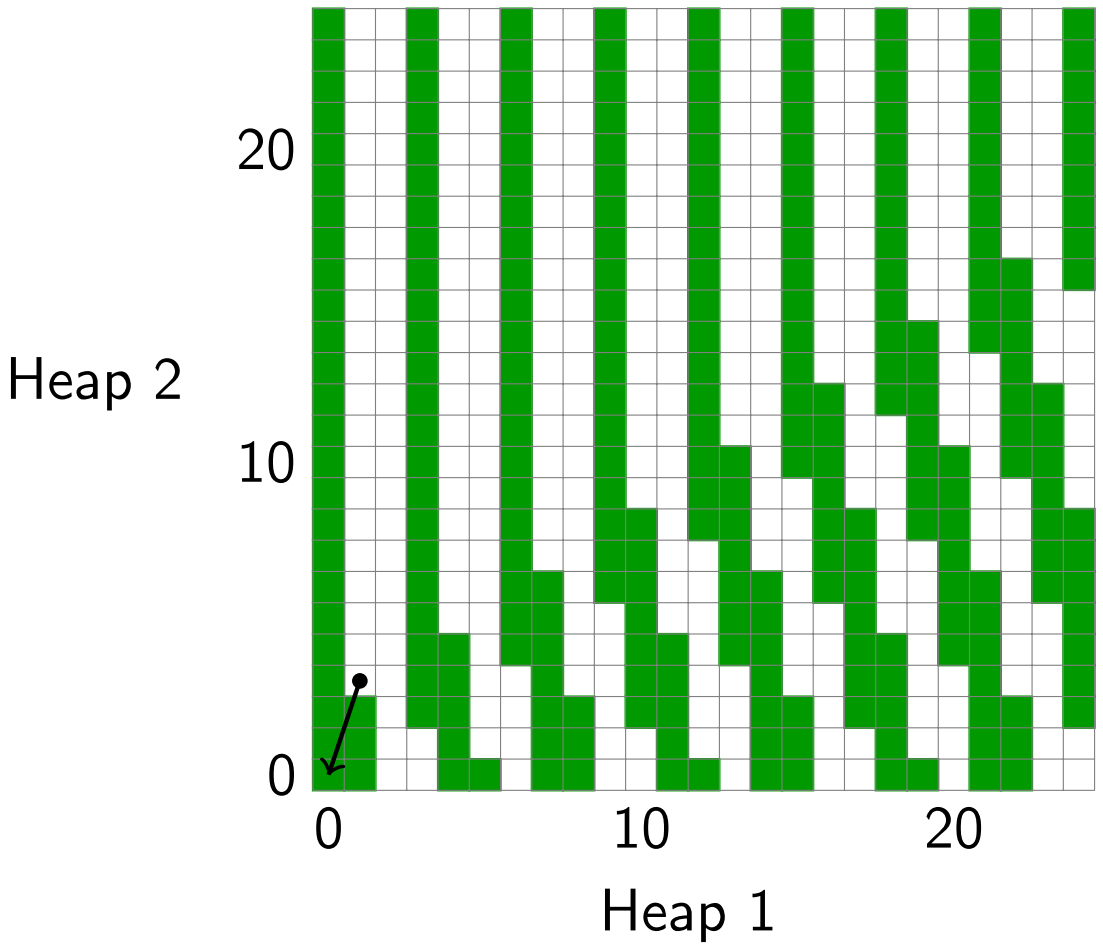
Safe positions of the game $\mathcal{M} = \{(-1, -3), (-2, 1)\}$



$(3, 2) \to (1, 3)$ is forced: safe goes to unsafe

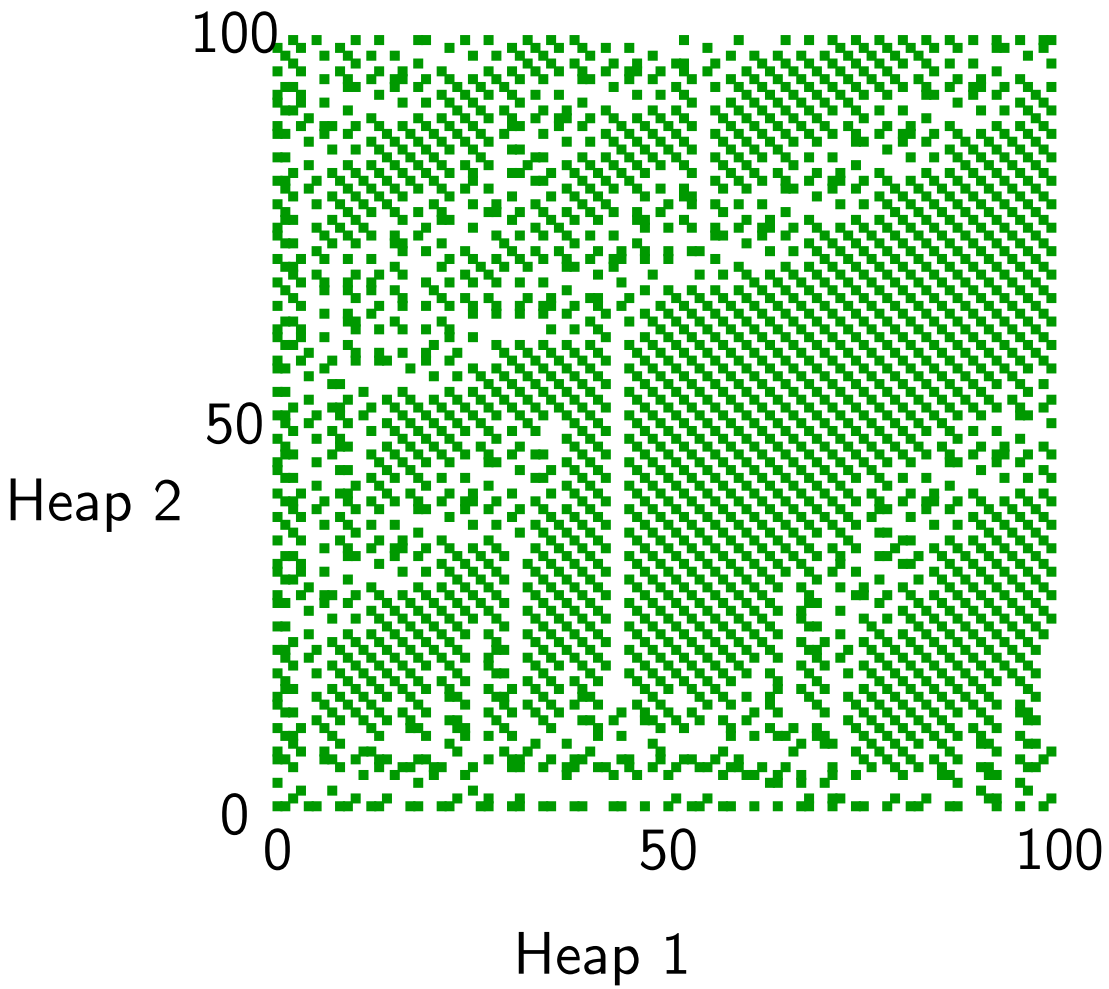


Bob's final move



Initial safe positions of the game

$\mathcal{M} = \{(0,-2), (-2,0), (2,-3), (-3,2), (-5,4), (-5,-2), (-4,-3), (-1,-4)\}$



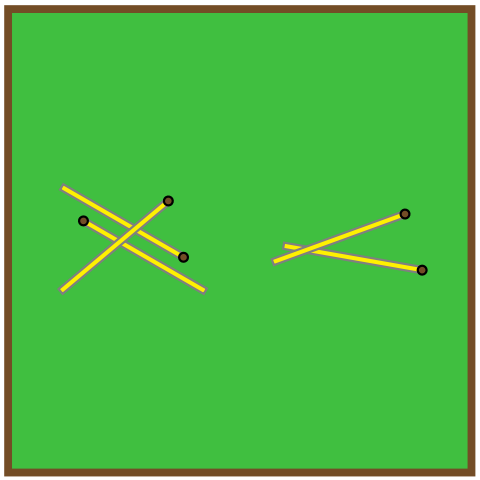
Decidability and heap games

Theorem (Larsson, Wästlund)

There is no algorithm that, given as input the number d of heaps and two finite sets $\mathcal{M}$ and $\mathcal{M}'$ of integer vectors specifying the rules of two d-heap games, decides whether or not the sets of safe positions of $\mathcal{M}$ and $\mathcal{M}'$ are the same.

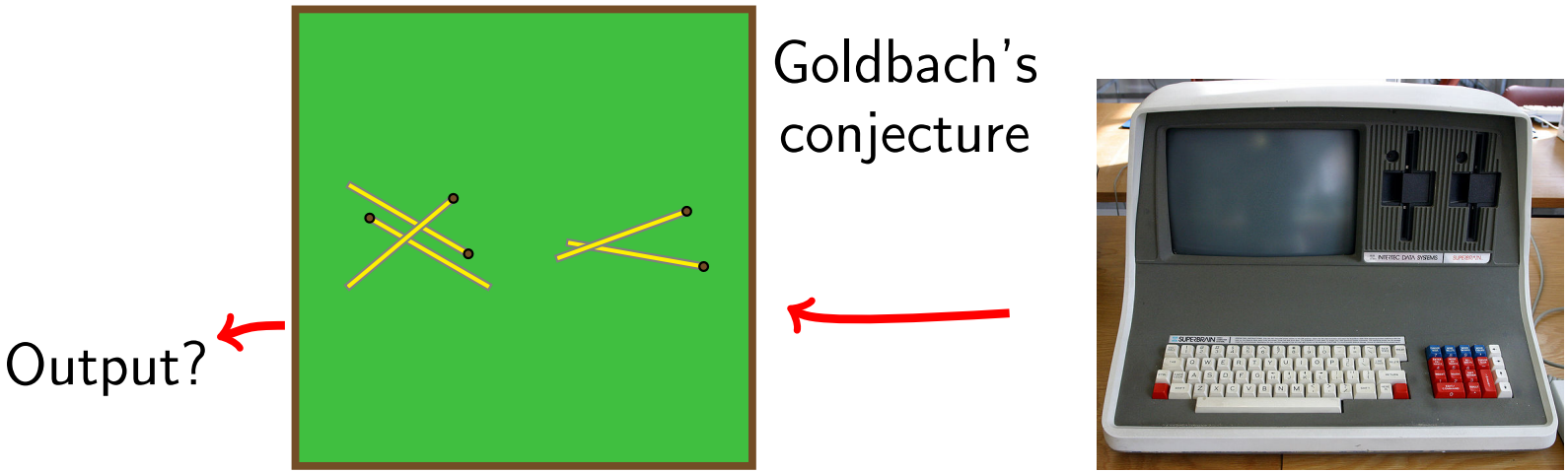
That is, no computer can ever be programmed to take as input any two rule-sets and decide whether the sets of safe starting positions are identical.

We have already seen that certain questions are programmable by any modern computer



Who wins and how

By our theorem, we also know that our games can "compute" arbitrarily hard problems: produce a given pattern if and only if the answer is "no" to a given problem



Wolfram's Rules 60 and 110

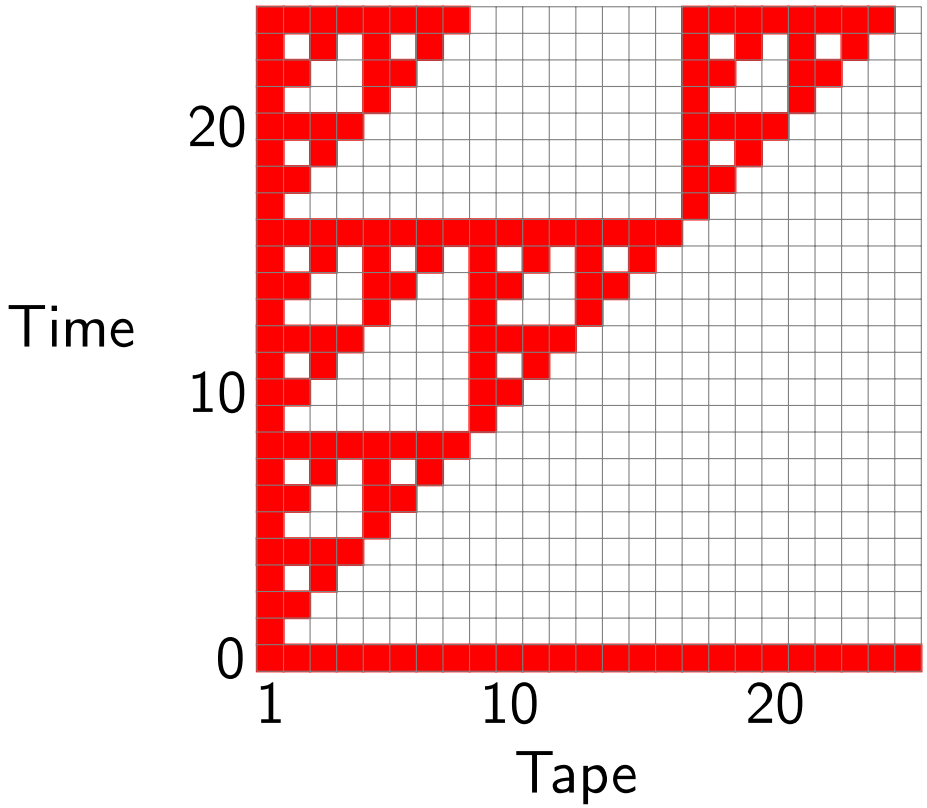
Rule 60

Given a doubly infinite bit-string, we update the cells simultaneously in discrete time steps. The content in a cell remains the same if and only if the cell immediately to the left contains a "0".

Rule 110

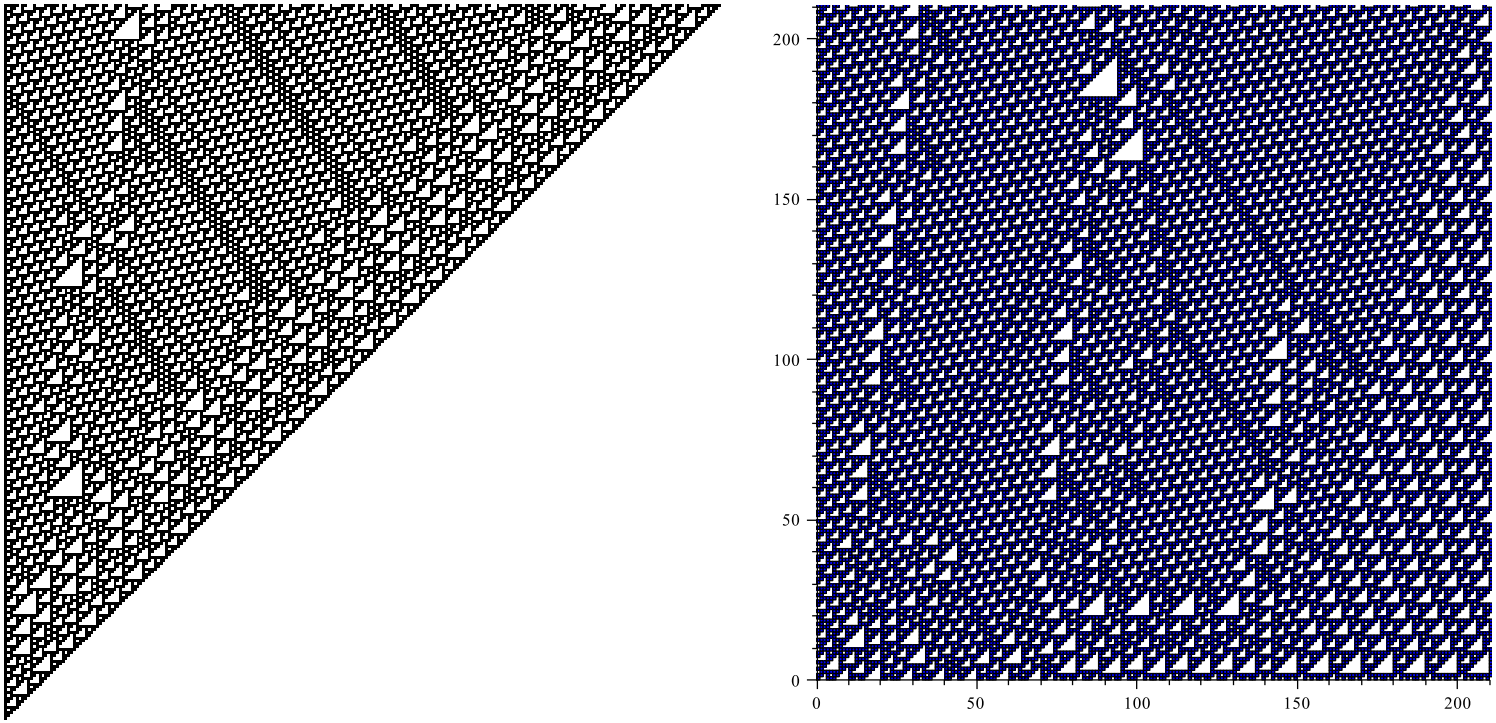
A given cell remains "0" if the cell immediately to the left is also "0". It remains "1" if at least one of the cells immediately to the left or right is "0". Otherwise the value flips.

The behavior of Wolfram's Rule 60 cellular automaton is decidable



The CA given by the update function $f(x, y) = x \oplus y$

The updates of Wolframs famous rule 110 cellular automaton



CA rule 110, time flows upwards. To the left the initial condition is a single "1". Matthew Cook (2004): given doubly periodic initial patterns and a central data pattern, the behavior is undecidable.

Safe positions as Isosceles Right Triangles

Idea for a Triangle Placing Game, also emulated via a game on two heaps

Find rules so that a game position is safe if and only if it consists of an IRT covering only 0s while its support covers only 1s.

Theorem (Larsson 2013)

There are games on just two heaps, with simple rules, that simulate the patterns of given CA rule 60 and rule 110 precisely, and hence the behavior of the latter game is undecidable.

The rules of the rule 60 game

- There is one finite heap of matches (the time-heap) and one finite heap of tokens (the tape-heap).

The rules of the rule 60 game

- There is one finite heap of matches (the time-heap) and one finite heap of tokens (the tape-heap).
- The current player removes at least one match (at most the whole heap) and at most as many tokens, as the number of matches removed by the previous player (possibly zero).

The rules of the rule 60 game

- There is one finite heap of matches (the time-heap) and one finite heap of tokens (the tape-heap).
- The current player removes at least one match (at most the whole heap) and at most as many tokens, as the number of matches removed by the previous player (possibly zero).
- It is not allowed to remove the remaining match(es) unless the tape-heap of tokens is empty.

The rules of the rule 60 game

- There is one finite heap of matches (the time-heap) and one finite heap of tokens (the tape-heap).
- The current player removes at least one match (at most the whole heap) and at most as many tokens, as the number of matches removed by the previous player (possibly zero).
- It is not allowed to remove the remaining match(es) unless the tape-heap of tokens is empty.
- Hence, if a player cannot remove a match (from the time-heap), the game ends and the other player wins.

Examples of the rule 60 game

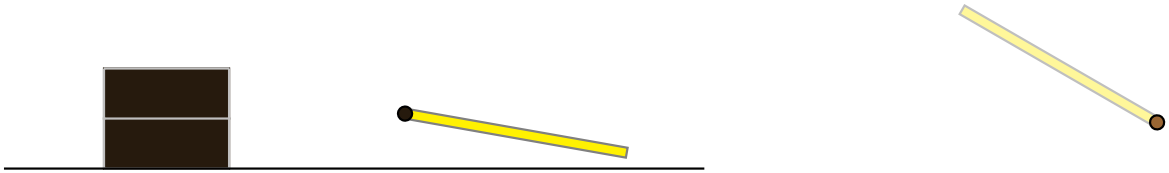


Figure: The previous player removed the rightmost match in the rule 60 game. Hence at most one token may be removed, which means that no move is possible and hence the previous player wins.

Examples of the rule 60 game

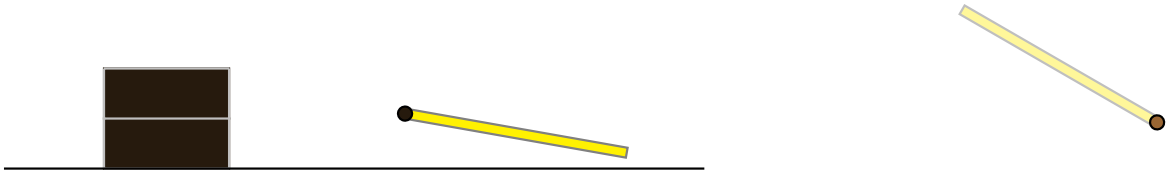


Figure: The previous player removed the rightmost match in the rule 60 game. Hence at most one token may be removed, which means that no move is possible and hence the previous player wins.

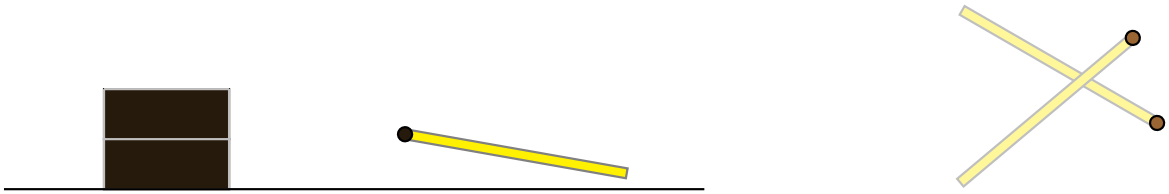


Figure: In this game, the next player wins by removing the last match together with both tokens.

Why “tape-heap"? The rule 110 game.

- We two-color the tokens, black or white: the final y matches can be removed if and only if the top y tokens are non-black.
- Thus, if there are no tokens left, then the last match can always be removed.
- The number of tokens a player can remove depends both on the current and the previous player's removal of matches.
- If the previous player removed m_p matches then $m - 1 \leq t \leq m_p + m$ tokens must be removed,
- together with $1 \leq m$ matches.

Example of rule 110 heap game

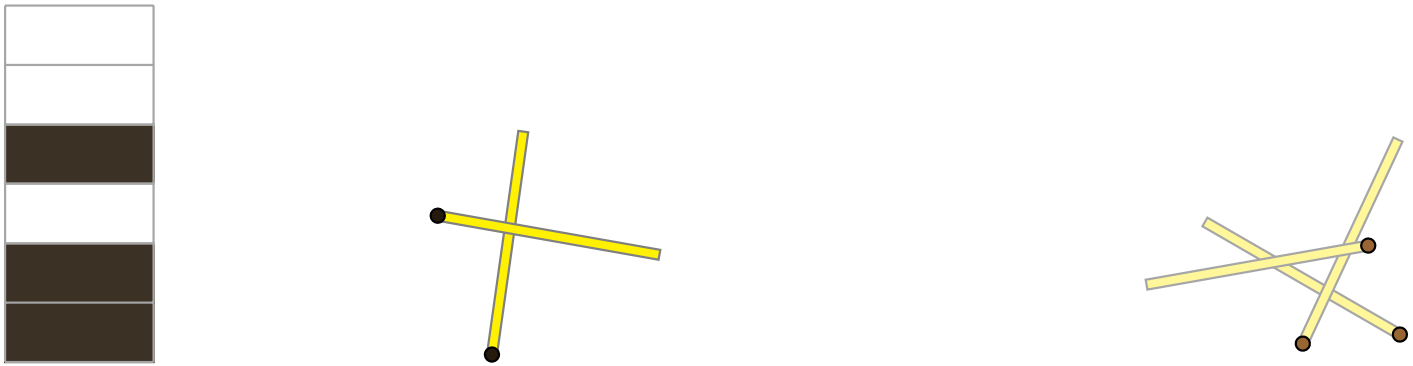


Figure: A rule 110 game variation on two heaps. The rightmost heap represents the previous player's removal of matches. Who wins the current game? Here $m_p = 3$ and $m \in \{1, 2\}$. If $m = 2$ then $1 \leq t \leq 5$, which violates the final condition. Hence only $m = 1$, $0 \leq t \leq 4$ is possible, which gives a win for the second player, by removing the last match together with $0, 1$ or 2 tokens as appropriate to avoid a top black token.

The board game variation of the rule 110 game