From children's games to the limits of computability

Urban Larsson, partly with Johan Wästlund
Chalmers University of Technology,
Logikseminariet 2012

March 30, 2012

Table of contents

Combinatorial games

- 2 players take turns in moving

Combinatorial games

- 2 players take turns in moving
- open/perfect information

Combinatorial games

- 2 players take turns in moving
- open/perfect information
- a set of rules

Combinatorial games

- 2 players take turns in moving
- open/perfect information
- a set of rules
- not necessarily the same for both players

Combinatorial games

- 2 players take turns in moving
- open/perfect information
- a set of rules
- not necessarily the same for both players
- but often simple, a child can understand them

Combinatorial games

- 2 players take turns in moving
- open/perfect information
- a set of rules
- not necessarily the same for both players
- but often simple, a child can understand them
- and a simple ending-condition

Combinatorial games

- 2 players take turns in moving
- open/perfect information
- a set of rules
- not necessarily the same for both players
- but often simple, a child can understand them
- and a simple ending-condition
- "checkmate".

Combinatorial game theory (CGT)

- Combinatorial games are special but very rich in structure.

Combinatorial game theory (CGT)

- Combinatorial games are special but very rich in structure.
- Fredrik's seminar: They encompass our number system.

Combinatorial game theory (CGT)

- Combinatorial games are special but very rich in structure.
- Fredrik's seminar: They encompass our number system.
- In fact CGT-players avoid to move in numbers.

Combinatorial game theory (CGT)

- Combinatorial games are special but very rich in structure.
- Fredrik's seminar: They encompass our number system.
- In fact CGT-players avoid to move in numbers.
- Here: Impartial games are never numbers...

Combinatorial game theory (CGT)

- Combinatorial games are special but very rich in structure.
- Fredrik's seminar: They encompass our number system.
- In fact CGT-players avoid to move in numbers.
- Here: Impartial games are never numbers...
- but always nimbers (Sprague 1934 and Grundy 1939).

Combinatorial game theory (CGT)

- Combinatorial games are special but very rich in structure.
- Fredrik's seminar: They encompass our number system.
- In fact CGT-players avoid to move in numbers.
- Here: Impartial games are never numbers...
- but always nimbers (Sprague 1934 and Grundy 1939).
- both players follow the same rules.

Combinatorial game theory (CGT)

- A player likes to move in a nimber if it is an $\mathcal{N}$-position

Combinatorial game theory (CGT)

- A player likes to move in a nimber if it is an $\mathcal{N}$-position
- that is unless it equals 0, 2nd player wins, a $\mathcal{P}$-position,

Combinatorial game theory (CGT)

- A player likes to move in a nimber if it is an $\mathcal{N}$-position
- that is unless it equals 0, 2nd player wins, a $\mathcal{P}$-position,
- a position is in $\mathcal{P}$ iff each option is in $\mathcal{N}$. Hence each terminal position is in $\mathcal{P}$.

Combinatorial game theory (CGT)

- A player likes to move in a nimber if it is an $\mathcal{N}$-position
- that is unless it equals 0, 2nd player wins, a $\mathcal{P}$-position,
- a position is in $\mathcal{P}$ iff each option is in $\mathcal{N}$. Hence each terminal position is in $\mathcal{P}$.
- This gives a recursive classification of all outcomes of a game.

Combinatorial game theory (CGT)

- A player likes to move in a nimber if it is an $\mathcal{N}$-position
- that is unless it equals 0, 2nd player wins, a $\mathcal{P}$-position,
- a position is in $\mathcal{P}$ iff each option is in $\mathcal{N}$. Hence each terminal position is in $\mathcal{P}$.
- This gives a recursive classification of all outcomes of a game.

Many classical heap games are impartial: Nim, Wythoff Nim, subtraction games, ...

A classical subtraction game

- An impartial subtraction game:

A classical subtraction game

- An impartial subtraction game:
- subtract 1 or 2 from a given number, starting from 21. The player who reaches 0 wins.

A classical subtraction game

- An impartial subtraction game:
- subtract 1 or 2 from a given number, starting from 21. The player who reaches 0 wins.
- 2nd player winning positions divisible by 3.

A classical subtraction game

- An impartial subtraction game:
- subtract 1 or 2 from a given number, starting from 21. The player who reaches 0 wins.
- 2nd player winning positions divisible by 3.
- Any game on one heap with a finite subtraction set $\mathcal{M}$ has ultimately periodic nimbers (Knuth 1981, Berlekamp-Conway-Guy 2001) at most exponential growth of length of period (in $\max \mathcal{M}$). Polynomial period lengths are obtained for special cases (Bulterman 1993, Flammenkamp 1996). Precise descriptions are not known in general, even for some subtraction-sets with only three numbers (Nhan Bao Ho arXiv 2012).

Generalized subtraction games

- Generalize: play on a fixed number of heaps of matches,

Generalized subtraction games

- Generalize: play on a fixed number of heaps of matches,
- a finite set $\mathcal{M}$ of move options (integer vectors).

Generalized subtraction games

- Generalize: play on a fixed number of heaps of matches,
- a finite set $\mathcal{M}$ of move options (integer vectors).
- Generalize "subtraction" to: For each move, the total number of matches decreases,

Generalized subtraction games

- Generalize: play on a fixed number of heaps of matches,
- a finite set $\mathcal{M}$ of move options (integer vectors).
- Generalize "subtraction" to: For each move, the total number of matches decreases,
- the number possibly increases on individual heaps.

Generalized subtraction games

- Generalize: play on a fixed number of heaps of matches,
- a finite set $\mathcal{M}$ of move options (integer vectors).
- Generalize "subtraction" to: For each move, the total number of matches decreases,
- the number possibly increases on individual heaps.
- But the game terminates. (Figures 1 and 2)

Generalized subtraction games

- Generalize: play on a fixed number of heaps of matches,
- a finite set $\mathcal{M}$ of move options (integer vectors).
- Generalize "subtraction" to: For each move, the total number of matches decreases,
- the number possibly increases on individual heaps.
- But the game terminates. (Figures 1 and 2)
- Why a finite number of moves?

Undecidable heap games

Theorem (Larsson, Wästlund)

It is algorithmically undecidable whether two heap games, defined via two finite move sets on the same number of heaps, have the same sets of $\mathcal{P}$-positions.

Undecidable heap games

Theorem (Larsson, Wästlund)

It is algorithmically undecidable whether two heap games, defined via two finite move sets on the same number of heaps, have the same sets of $\mathcal{P}$-positions.

Questions of undecidability are related to universal computation which requires "finite programs" as input.

Games as universal computers

Invariant games

Our heap games have invariant move options. Each move option is available from each position, provided there are enough tokens in each heap.

Games as universal computers

Invariant games

Our heap games have invariant move options. Each move option is available from each position, provided there are enough tokens in each heap.

Modular games are "nearly invariant" subtraction games...

The modular games are played on two heaps, a time-heap and a tape-heap. The move options are given by k move sets $\mathcal{M}_i, i \in \{0, \ldots, k-1\}$, the congruence class of the number of matches in the time-heap determines what the moves options are.

Games as universal computers

Invariant games

Our heap games have invariant move options. Each move option is available from each position, provided there are enough tokens in each heap.

Modular games are "nearly invariant" subtraction games...

The modular games are played on two heaps, a time-heap and a tape-heap. The move options are given by k move sets $\mathcal{M}_i, i \in \{0, \ldots, k-1\}$, the congruence class of the number of matches in the time-heap determines what the moves options are.

...capable of universal computation

Modular games are capable of universal computation because they can emulate cellular automata.

Cellular automata and modular games

Our one-dimensional cellular automaton (CA) takes as input a doubly infinite binary string. The initial string is "...0011...", $x_{i,0} = 0$ iff $i < 0$.

Cellular automata and modular games

Our one-dimensional cellular automaton (CA) takes as input a doubly infinite binary string. The initial string is "...0011...", $x_{i,0} = 0$ iff $i < 0$.

The string is updated via a boolean function f, which takes as input n consecutive digits, via the rule: cell

$$x_{i,t+1} = f(x_{i-n,t}, \ldots, x_{i,t}).$$

For our purpose require $f(0, \ldots, 0) = 0$. (Figures 3 and 4) Then CA(f) corresponds to the evolution of a given "finite program". Such abstract machines are capable of universal computation.

The construction of a modular game computer

- Let square brackets $[\cdot]$ denote $1 - \max(\cdot)$.

The construction of a modular game computer

- Let square brackets $[\cdot]$ denote $1 - \max(\cdot)$.
- For instance $0 = [xyz] = 1 - \max(x, y, z)$, iff at least one of $x, y, z = 1$ and $1 = [xyz]$ iff $x = y = z = 0$, and by convention $[\,] = 1$. (Think $P \leftrightarrow$ "1".)

The construction of a modular game computer

- Let square brackets $[\cdot]$ denote $1 - \max(\cdot)$.
- For instance $0 = [xyz] = 1 - \max(x, y, z)$, iff at least one of $x, y, z = 1$ and $1 = [xyz]$ iff $x = y = z = 0$, and by convention $[\,] = 1$. (Think $P \leftrightarrow$ "1".)
- Every Boolean function can be expressed in terms of nested brackets.

The construction of a modular game computer

- Let square brackets $[\cdot]$ denote $1 - \max(\cdot)$.
- For instance $0 = [xyz] = 1 - \max(x, y, z)$, iff at least one of $x, y, z = 1$ and $1 = [xyz]$ iff $x = y = z = 0$, and by convention $[\,] = 1$. (Think $P \leftrightarrow$ "1".)
- Every Boolean function can be expressed in terms of nested brackets.
- The set consisting of & and $\sim$ (negation) is a complete set of connectives in propositional logic.

The construction of a modular game computer

- Let square brackets $[\cdot]$ denote $1 - \max(\cdot)$.
- For instance $0 = [xyz] = 1 - \max(x, y, z)$, iff at least one of $x, y, z = 1$ and $1 = [xyz]$ iff $x = y = z = 0$, and by convention $[\,] = 1$. (Think $P \leftrightarrow$ "1".)
- Every Boolean function can be expressed in terms of nested brackets.
- The set consisting of & and $\sim$ (negation) is a complete set of connectives in propositional logic.
- For instance the function $f(x, y) = x \oplus y$ can be expressed as

$$x \oplus y = [[xy][[x][y]]].$$

(Figures 5 to 7)

The Gadget

From non-invariant

We wish to turn the modular games into invariant games. Simple rules: We should not need to count the number of matches in the time-heap in order to know how to play the game.

The Gadget

From non-invariant

We wish to turn the modular games into invariant games. Simple rules: We should not need to count the number of matches in the time-heap in order to know how to play the game.

to invariant

Introduce k more heaps, the gadget. Suppose that the number of matches in the time heap belongs to the i^{th} congruence class. Then a single match in the i^{th} heap of the gadget is moved to the j^{th} heap of the gadget if and only if $(i - j) \in \mathcal{M}_i$ matches are removed from the time-heap.

Reducing the halting problem to that of determining P-equivalence

The halting problem of a universal Turing machine is equivalent to the problem of determining whether a given finite string of "0"s and "1"s, say "101", appears in the update of a one-dimensional cellular automata.

Reducing the halting problem to that of determining P-equivalence

The halting problem of a universal Turing machine is equivalent to the problem of determining whether a given finite string of "0"s and "1"s, say "101", appears in the update of a one-dimensional cellular automata.

We reduce this problem to the question whether or not two modular games (and hence invariant games) have the same sets of $\mathcal{P}$-positions. Hence, the halting problem is not more complicated than the problem of equivalence of $\mathcal{P}$-positions for invariant games. The reduction works in both directions so that the problems are in fact equivalent. (Figures 8 and 9)

The Gadget

Comments

What if there is more than one match in the gadget? This can be dealt with by introducing a subfamily of invariant games which trivializes the $\mathcal{P}$-positions of the modular game to periodicity.

What if the size of the time-heap is in a different congruence class than the single match in the gadget? This either creates periodic $\mathcal{P}$-patterns or introduces nothing new.

A toy example: Goldbach's conjecture

Any very hard problem (for a CA) can be emulated by a modular game and thereby by a sub-family of all invariant games.

There is an even natural number $n > 3$ which is not a sum of two primes if and only if there is a position X which is in $\mathcal{P}$ for the k-heap invariant game $\mathcal{M}$ but not in $\mathcal{P}$ for $\mathcal{M}'$, where $\mathcal{M}$ and $\mathcal{M}'$ encode the program and X the number n.

A move-size dynamic variation of the game of Nim

Simple rules can be obtained by infinite move sets. For example: Bouton's game of Nim (Bouton 1902) and the move-size dynamic variation Imitation Nim (Larsson 2009): (Figures 10 to 12)

Dynamic heap games and the universality of rule 110

Can we emulate the structures of the universal rule 110 CA directly by some heap game?

The rules of game should emulate the patterns of the rule 110 CA via its set of $\mathcal{P}$-positions. The rule 110 CAs update function: keep a "1" if and only if the neighboring cell to the left or right is "0". Keep a "0" if and only if the cell to the left is "0".

Triangles and CAs

For the rule 110 CA Isosceles Right Triangles are being created in mysterious patterns. (Figure 13) Why IRTs? Is there another explanation? Is there a 2-player combinatorial game which generates such patterns? The rule 60 CA also generate IRTs and is simpler to analyze:

The rule 60 game

- A take away game emulating the behavior of rule 60.

The rule 60 game

- A take away game emulating the behavior of rule 60.
- A heap of matches and a heap of tokens.

The rule 60 game

- A take away game emulating the behavior of rule 60.
- A heap of matches and a heap of tokens.
- 2 players alternate in moving.

The rule 60 game

- A take away game emulating the behavior of rule 60.
- A heap of matches and a heap of tokens.
- 2 players alternate in moving.
- A player can remove any number of matches, at least one and at most the whole heap,

The rule 60 game

- A take away game emulating the behavior of rule 60.
- A heap of matches and a heap of tokens.
- 2 players alternate in moving.
- A player can remove any number of matches, at least one and at most the whole heap,
- and a number of tokens $0 \leq t \leq m_p$ limited by the number of matches m_p the previous player removed.

The rule 60 game

- Ending condition: who wins?

The rule 60 game

- Ending condition: who wins?
- The final match can be removed if and only if the heap of tokens is emptied in the same move.

The rule 60 game

- Ending condition: who wins?
- The final match can be removed if and only if the heap of tokens is emptied in the same move.
- Note $t \leq m_p$, so if there are less than m_p tokens left,

The rule 60 game

- Ending condition: who wins?
- The final match can be removed if and only if the heap of tokens is emptied in the same move.
- Note $t \leq m_p$, so if there are less than m_p tokens left,
- a winning move is to remove all tokens and all matches,

The rule 60 game

- Ending condition: who wins?
- The final match can be removed if and only if the heap of tokens is emptied in the same move.
- Note $t \leq m_p$, so if there are less than m_p tokens left,
- a winning move is to remove all tokens and all matches,
- but there are also other terminal positions:

The rule 60 game

- Ending condition: who wins?
- The final match can be removed if and only if the heap of tokens is emptied in the same move.
- Note $t \leq m_p$, so if there are less than m_p tokens left,
- a winning move is to remove all tokens and all matches,
- but there are also other terminal positions:
- (Figures 14 to 16)

Who wins the rule 60 game?

- The outcome of a position is resolved via the corresponding coordinates of the CA-cell together with the value of m_p. There are only a few cases to check: (Figure 17)

Who wins the rule 60 game?

- The outcome of a position is resolved via the corresponding coordinates of the CA-cell together with the value of m_p. There are only a few cases to check: (Figure 17)
- If the cell representing the heap-sizes is red ("1") then the outcome is N. (By the update rule of the CA at least one of the two cells just below is white—use it as in the next item.)

Who wins the rule 60 game?

- The outcome of a position is resolved via the corresponding coordinates of the CA-cell together with the value of m_p. There are only a few cases to check: (Figure 17)
- If the cell representing the heap-sizes is red ("1") then the outcome is N. (By the update rule of the CA at least one of the two cells just below is white—use it as in the next item.)
- A white ("0") cell is P iff the cell just below is red and the $m_p - 1$ above cells are white. ("If": then each cell just below the base of the IRT is red; "only if": otherwise there is a white cell just below the base of the IRT.)

Who wins the rule 60 game?

Theorem

Let the initial condition of the rule 60 CA be $a_{i,0} = 1$ *if and only if* $i > 0$*. Then, a position in the rule 60 game with* x *tokens,* y *matches and where the previous player removed* m_p *matches is a second player win if and only if* $a_{x,y} = \ldots = a_{x,y+m_p-1} = 0$ *and if* $y > 0$ *then* $a_{x,y-1} = 1$*.*

Who wins the rule 60 game?

Theorem

Let the initial condition of the rule 60 CA be $a_{i,0} = 1$ *if and only if* $i > 0$*. Then, a position in the rule 60 game with* x *tokens,* y *matches and where the previous player removed* m_p *matches is a second player win if and only if* $a_{x,y} = \ldots = a_{x,y+m_p-1} = 0$ *and if* $y > 0$ *then* $a_{x,y-1} = 1$*.*

This result generalizes to an infinitie family of games and CAs. The rule 110 game is very similar.

The rule 110 game

A token of the rule 110 game is either black “1” or white “0” corresponding to a finite part of the initial string of the CA.

The rule 110 game

A token of the rule 110 game is either black "1" or white "0" corresponding to a finite part of the initial string of the CA.

The move-dynamic rule is altered: $0 \leq m-1 \leq t \leq m+m_p$. In particular, it is allowed to remove 0 tokens iff $m = 1$.

The rule 110 game

A token of the rule 110 game is either black "1" or white "0" corresponding to a finite part of the initial string of the CA.

The move-dynamic rule is altered: $0 \leq m-1 \leq t \leq m+m_p$. In particular, it is allowed to remove 0 tokens iff $m = 1$.

The ending condition is modified: the m last matches may be removed iff there is no black token among the top m tokens. (Figure 18)

Undecidability of the rule 110 CA

Let $\underline{L}C\underline{R}$ denote a doubly infinite binary string with periodic Left and Right patterns and a central data pattern.

Theorem (Cook 2004)

It is algorithmically undecidable whether a given finite binary string occurs in the rule 110 CA with an $\underline{L}C\underline{R}$ initial condition.

Undecidability of the rule 110 CA

Let $\underline{L}C\underline{R}$ denote a doubly infinite binary string with periodic Left and Right patterns and a central data pattern.

Theorem (Cook 2004)

It is algorithmically undecidable whether a given finite binary string occurs in the rule 110 CA with an $\underline{L}C\underline{R}$ initial condition.

The finite string can be taken as 01101001101000 obtained from the so-called F-glider. (Figure 19)

Undecidability of the rule 110 CA

Way of proof, reduction process: Turing's halting problem $\leftrightarrow$ Post's tag system $\leftrightarrow$ a newly invented cyclic tag system $\leftrightarrow$ a rule 110 F-glider being produced through interactions of other gliders.

Undecidability of the rule 110 CA

Way of proof, reduction process: Turing's halting problem $\leftrightarrow$ Post's tag system $\leftrightarrow$ a newly invented cyclic tag system $\leftrightarrow$ a rule 110 F-glider being produced through interactions of other gliders.

Corollary (Cook 2004)

It is undecidable whether the central data pattern will ultimately become two dimensional periodic.

Undecidability of the rule 110 game

Theorem

It is algorithmically undecidable whether a given finite sequence of moves has alternating P-positions for a rule 110 game with an $\underline{L}C\underline{R}$-ending condition.

(Figure 20)

Undecidability of the rule 110 game

Theorem

It is algorithmically undecidable whether a given finite sequence of moves has alternating P-positions for a rule 110 game with an $\underline{L}C\underline{R}$-ending condition.

(Figure 20)

Theorem

It is undecidable whether the central pattern of second player winning positions will ultimately become two dimensional periodic for $\underline{L}C\underline{R}$-rule 110 games.