Revision

Note Title 22/11/2010

- Invariants
- Logic Puzzles
- Brute-force search, State-transition Diagrams
- Games
- Induction

Invariants

$E[xs := es]$ denotes expression E after the assignment $xs := es$.

Examples

$$x[x := x+1] = x+1$$

$$(x=y)[x,y := y,x] = (y=x)$$

$$(m+2\times n)[m,n := m+3, m\times n] = (m+3)+2\times(m\times n)$$

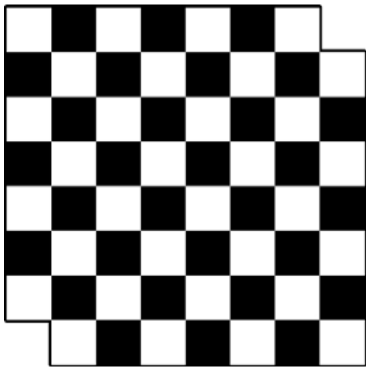
Invariant E is an *invariant* of $xs := es$ if

$$E = E[xs := es] .$$

Assignment	*Invariant*	*Applications*
$c := c+3$	$c \bmod 3$	tiling (eg mutilated chess board)
$c := c+2$	$c \bmod 2$ / even.c	
$c := c+k$	$c \bmod k$	

Mutilated Chess Board

chessboard



dominoes

Is it possible to tile the chessboard with (non-overlapping) dominoes ?

Assignment	Invariant	Applications
$c,d := c+2, d+3$	$3c-2d$	box problem, chameleon problem (not discussed)
$c,d := c+j, d+k$	$k \times c - j \times d$	
$s,n := s+2n+1, n+1$	$s-n^2$	summation (proof by induction)

Assignment	Invariant	Applications
$b, c \;:=\; \neg b, c+1$	$b = even.c$	river crossing

(b "the boat is at the left bank"
c " the number of crossings")

$$H(n+1, d) \qquad even.(n+1) = d$$

Towers of Hanoi

$$= \; H(n, \neg d) \; ; \; \cdots$$

(n number of disks
d direction)

Invariant: pattern, law, induction hypothesis,

Skill: practise identifying simple invariants as part of the process of familiarising yourself with a problem.

Logic Puzzles

Portia's Caskets.

Two caskets: silver and gold.
Exactly one has a picture of Portia in it.
Each casket has an inscription which may be true or false.

Silver: exactly one of these inscriptions is true.
Gold: the picture is not in this casket.

Boolean unknowns

calculated by formulating and solving system of simultaneous equations.

ps the picture is in the silver casket
pg the picture is in the gold casket

is inscription on the silver casket ("is true")
ig inscription on the gold casket ("is true")

Exactly one has a picture of Portia in it. $(ps \neq pg)$

Silver: exactly one of these inscriptions is true. $\wedge\ (is = (is \neq ig))$

Gold: the picture is not in this casket. $\wedge\ (ig = \neg pg)$

$(ps \neq pg) \;\wedge\; (is = (is \neq ig)) \;\wedge\; (ig = \neg pg)$

= { rewrite : "≡" is evaluated associatively }

$(ps \equiv pg \equiv false) \;\wedge\; (is \equiv is \equiv ig \equiv false)$
$\wedge\; (ig \equiv pg \equiv false)$

= { properties of boolean equality }

$(ps \equiv pg \equiv false) \;\wedge\; (ig \equiv false)$
$\wedge\; (ig \equiv pg \equiv false)$

= { properties of boolean equality }

$(ps \equiv false) \;\wedge\; (ig \equiv false) \;\wedge\; (pg \equiv true)$

2. Alf says "I won", "Bob was second"
Bob says "Alf was second", "I was third".
Exactly one of the statements made by each of Alf and Bob is true.

$(A1 \neq B2) \wedge (A2 \neq B3)$

= { distributivity:
$[\,(p \neq q) \wedge r \;=\; (p \wedge r \neq p \wedge q)\,]$
and symmetry of $\wedge$
(so $[\,r \wedge (p \neq q) = (r \wedge p \neq r \wedge q)\,]$ }

$A1 \wedge A2 \neq A1 \wedge B3 \neq B2 \wedge A2 \neq B2 \wedge B3$

= { $[\,Xp \wedge Xq = false \Leftarrow p \neq q\,]$
$[\,Xp \wedge Yp = false \Leftarrow X \neq Y\,]$ }

$A1 \wedge B3$

(Also: implicit use of associativity of $\neq$,
false is unit of $\neq$.)

Skill: Know your (Boolean) algebra.

Use equational reasoning, avoid case analysis.

Nervous Presidents

3 couples (president and bodyguard) want to cross a river.
Everyone can row. There is only one boat.
A president cannot be with another president's bodyguard unless his/her own bodyguard is also present.

The boat can carry two people at a time.
How can they get across?

3C ‖ — precondition

Notation	
‖	river
nC	n couples, eg. 2C ‖ 1C
nP	n presidents
nB	n bodyguards

‖ 3C — postcondition

3C ‖

Notation	
‖	river
nC	n couples, eg. 2C ‖ 1C
nP	n presidents
nB	n bodyguards

5

1C ‖ 1C ‖ 1C — unique symmetric crossing

2

‖ 3C

Notation

||	river
nC	n couples, eg. 2C || 1C
nP	n presidents
nB	n bodyguards

3C ||

3B || 3P

1C | 1C | 1C

3P || 3B

|| 3C

Skills:

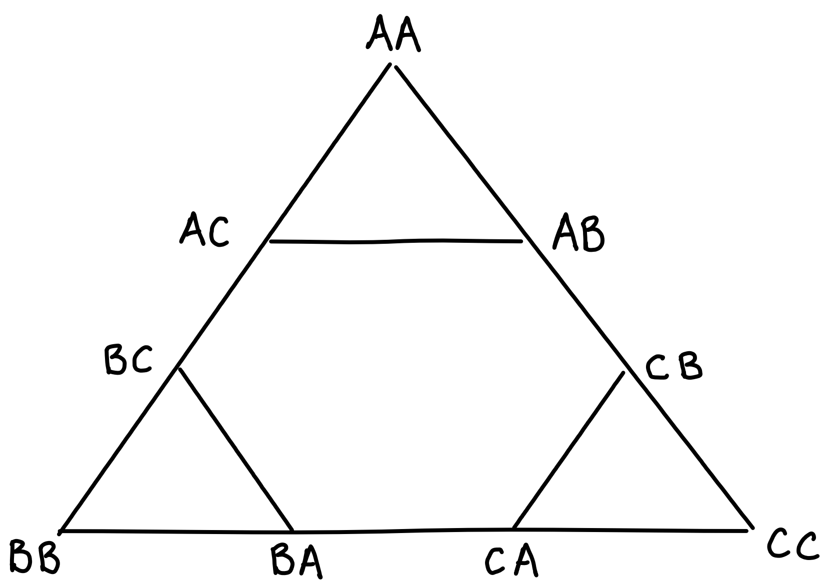
Ask the questions: what is the goal?
what is given?

Exploit structure in order to decompose the problem into simpler subproblems.

Combine concision with precision.

- Brute-force search, State-transition Diagrams

Tower of Hanoi (2 disks)



State Transition Diagram

Skills: Practise making precise

- valid states
- valid transitions

Look for and exploit inherent structure

Use brute force only as a last resort.

Games

impartial, 2-person combinatorial games with complete information.

assumptions: games guaranteed to terminate

a player in an end position loses.

end position: position from which no moves are possible

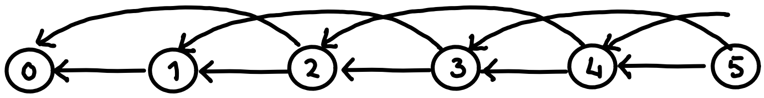
- Game graph

  Winning and Losing Positions

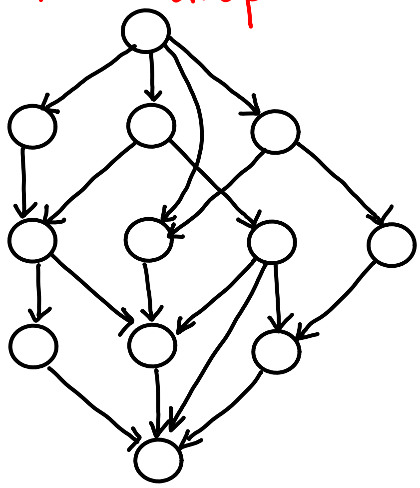
- Sum of Games

Matchstick Game

Move: remove 1 or 2 matches.



(Random) Game Graph

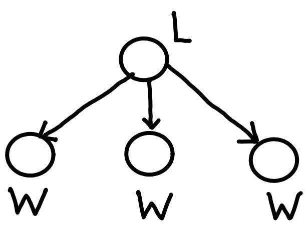


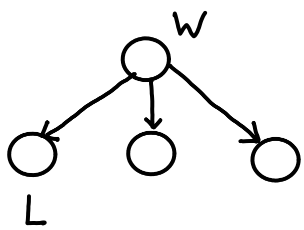
Positions are the nodes of the graph.

Moves are directed edges.

Losing Position: every move is to a winning position.
(guaranteed to lose against a perfect player)

Winning Position: there is a move to a losing position.
(guaranteed to be able to win)

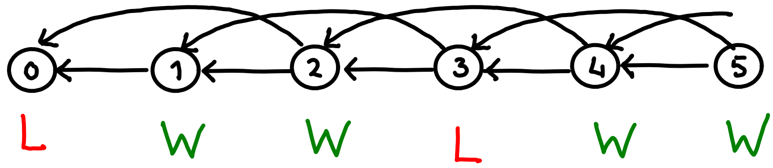




In principle (even for games like chess) it is possible to determine for each position whether it is winning or losing.

Matchstick Game

Move: remove 1 or 2 matches.



m matches is a losing position (losing.m)

$= \quad ((m \bmod 3) = 0)$

Winning strategy:

from a winning position $(((m \bmod 3) = 0) = \text{false})$

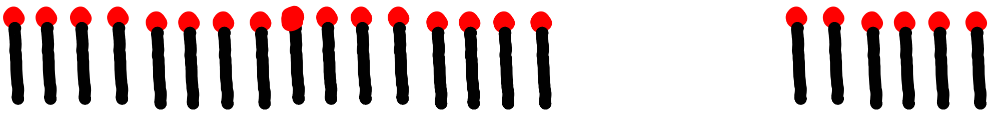
truthify $(m \bmod 3) = 0$ by removing $m \bmod 3$ matches.

How to win.

Identify a property of positions
(the *strategy invariant*) such that

- all end positions satisfy the property
- *every* move from a position satisfying the property *falsifies* the property.
- for every position that does not satisfy the property *there is* a move that *truthifies* the property.

Game Sum



A game with two piles of matches is the "sum" of two single-pile matchstick games.

(NB: rule for moving in left pile may differ from rule for moving in right pile.)

The *sum* of two games is a game defined as follows:

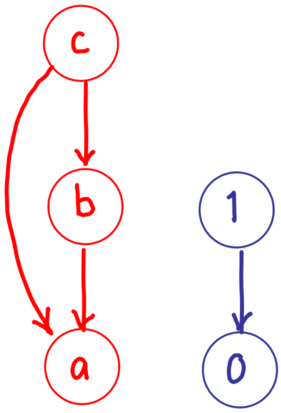
Call the two games the *left* game and the *right* game.

A *position* in the sum game is an ordered pair (ℓ, r) of positions, where ℓ is a position in the left game and r is a position in the right game.

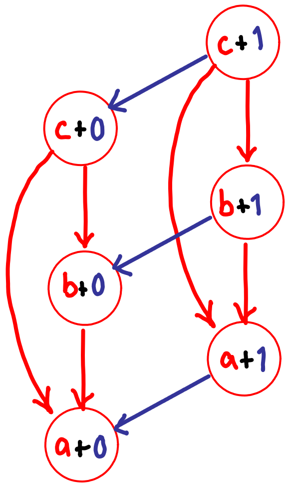
A *move* $(\ell, r) \mapsto (\ell', r')$ in the sum game satisfies

either: $\ell \mapsto \ell'$ is a move in the left game, and $r = r'$,

or : $\ell = \ell'$ and $r \mapsto r'$ is a move in the right game.



A Sum Game



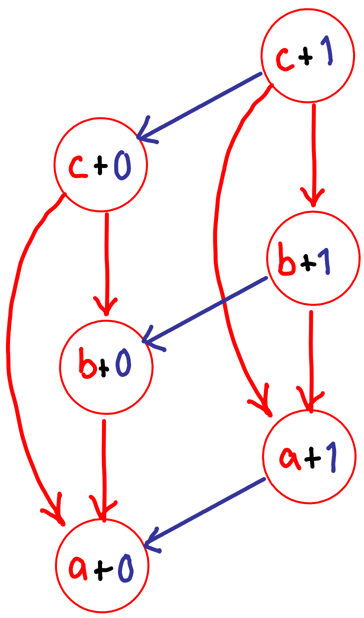
Game graph of sum game

If the number of positions in the two games is m and n, the number of positions in the sum game is $m \times n$.

Goal: Determine a winning strategy that can be computed in time proportional to $m+n$ (not $m \times n$).

Does the winning-position predicate distribute over sum?

Answer: no. (combination of two winning positions can be winning or losing).



$W.(c+1) = true$

$W.(b+1) = false$

$W.c = true$

$W.b = true$

$W.1 = true$

Mex Numbers

How to win. Assign a (natural) number to each position in the left and right games.

Use *strategy invariant* $mex.l = mex.r$

where $mex.p$ is the number assigned to position p

l is the position in the left game

r is the position in the right game.

- all end positions satisfy the property

  Assign number 0 to all end positions.

- *every* move from a position satisfying the property *falsifies* the property.

  $\{\ mex.l = mex.r\ \}$

  $l := l' \quad \square \quad r := r'$

  $\{\ mex.l \neq mex.r\ \}$

Require: if $s \mapsto t$ is a move, $mex.s \neq mex.t$

- for every position that does not satisfy the property *there is* a move that *truthifies* the property.

  $\{\ mex.l \neq mex.r\ \}$

  if $mex.l > mex.r \longrightarrow$ move in left game

  $\square$ $mex.r > mex.l \longrightarrow$ move in right game

  fi

  $\{\ mex.l = mex.r\ \}$

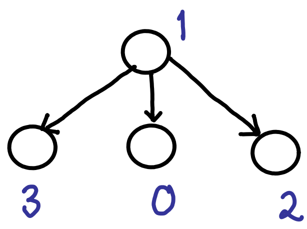
Require: for all m, $mex.l > m$, there is a move from l to a position l' such that $mex.l' = m$

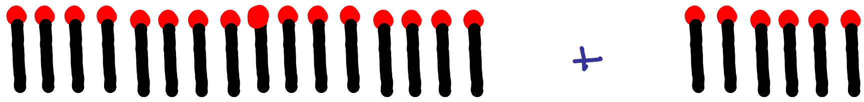
Similarly for r.

mex "minimal excludant"

mex number of a position is the smallest number not included in the numbers assigned to its successors

Example:





Left game: remove 1 thru M matches

mex no. of m matches $= m \bmod (M+1)$

Right game: remove 1 thru N matches

mex no. of n matches $= n \bmod (N+1)$

Strategy invariant: $m \bmod (M+1) = n \bmod (N+1)$

Winning strategy:

if $m \bmod (M+1) > n \bmod (N+1) \rightarrow$ remove $m \bmod (M+1) - n \bmod (N+1)$ matches from left pile

$\square$ $m \bmod (M+1) < n \bmod (N+1) \rightarrow$ remove $n \bmod (N+1) - m \bmod (M+1)$ matches from right pile

fi

Key: loser winner

$\{ m \bmod (M+1) = n \bmod (N+1) \}$

if $m \neq 0 \rightarrow$ decrease m by at most M

$\square$ $n \neq 0 \rightarrow$ decrease n by at most N

fi

$\{ m \bmod (M+1) \neq n \bmod (N+1) \}$

if $m \bmod (M+1) > n \bmod (N+1) \rightarrow$ remove $m \bmod (M+1) - n \bmod (N+1)$ matches from left pile

$\square$ $m \bmod (M+1) < n \bmod (N+1) \rightarrow$ remove $n \bmod (N+1) - m \bmod (M+1)$ matches from right pile

fi

$\{ m \bmod (M+1) = n \bmod (N+1) \}$

Skill: Don't guess!

Formulate requirements precisely.

Exploit requirements to calculate solution.

Induction

Principle of Mathematical Induction

Suppose P is a property of natural numbers.
Then P is everywhere true if

$P.0$ (basis)

for all n, $P.(n+1)$ follows from $P.n$. (induction step)

used to verify conjectures

Used to solve a *class* of problems
where each *instance* has a *size* (a natural number).

- Construct a solution to instances of size 0.
- Assuming that it is possible to construct solutions to instances of size n, construct a solution to instances of size $n+1$.

Skill: Abstraction: identify a given problem as an instance of a class of problems.

$$\boxed{\text{induction hypothesis} = \text{invariant}}$$

Final word:

PRACTISE !