

WIAWTTAS

Travel from thrillers to Tetris

What I always wanted to tell about science

dr. Walter Kusters, LIACS

10 September 2008

`www.liacs.nl/home/kusters/`

WIAWTTAS

In this lecture we will travel from thrillers to Tetris.

Meanwhile we meet:

recommender systems,
Data Mining,
association rules,
Artificial Intelligence (AI),
games and plays,
surreal numbers,
CT scans,
NP-completeness,
and many others.

WIAWTTAS

Crime novels

Crime novels deal with crime — in many ways. There are several sub-genres:

detective/ion Poe, Doyle, Christie (whodunnit)

psychological Rendell (whydunnit; inverted crime novels)

police procedural Sjöwall and Wahlöö, Simenon

private eye Hammett, Chandler

spy Ambler, Le Carré

thriller legal, historical, medical, literary, Da Vinci, . . .

. . .

Doyle (1893):

'Is there any other point to which you would draw my attention?'

'To the curious incident of the dog in the night-time.'

'The dog did nothing in the night-time.'

'That was the curious incident.'

Hammett (1927):

I first heard Personville called Poisonville by a red-haired mucker named Hickey Dewey in the Big Ship in Butte. He also called his shirt a shoit.

Rendell (1977):

Eunice Parchman killed the Coverdale family because she could not read or write.

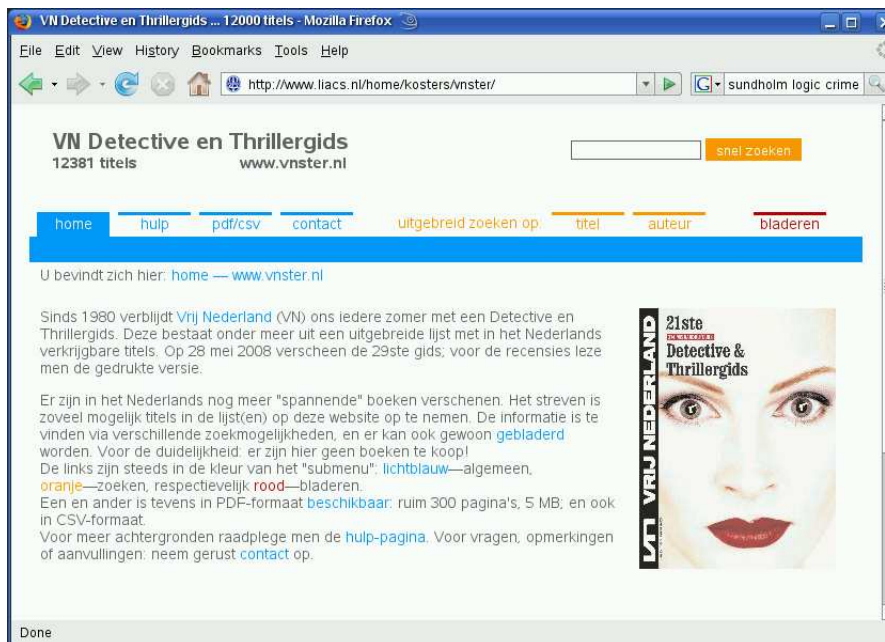
From a scientific point of view you can approach crime novels from many angles:

- Why do people like them?
- When is a novel good or bad?
- When is it literature?
- What are the boundaries between (sub-)genres?

But there are also computer science related issues like:

- What is the logic in a novel? (Sundholm)
- How to build recommender systems?

As a sort of escape from science, Hendrik Jan Hoogeboom and I built a website concerning crime novels, starting from the printed “Vrij Nederland Detective & Thrillergids”: an extensive ranking system.



`www.vnster.nl`

How to recommend?

WIAWTTAS

Data Mining

Data Mining tries to find interesting and (un)expected patterns in large amounts of (un)ordered data.

Cases (among others):

- Market basket analysis: What and how do we buy?
- Bio-informatics: What is common in DNA?

Problems (selection):

- need for expected results and for unexpected ones
- moving targets
- acquisition of data
- distance notion

The Data Mining process — or rather the **KDD** process, for **Knowledge Discovery in Databases** — is usually divided into the following steps:

1. data selection
2. cleaning, e.g., de-duplication and domain consistency
3. enrichment, e.g., data fusion
4. coding
5. Data Mining — the real thing
this is not intended to be recursive ;-)
6. reporting

There are *many* Data Mining techniques, or rather algorithms that can be used to mine data:

- (don't forget:) statistical methods
- all sorts of machine learning techniques from AI: evolutionary computation, neural networks, Bayesian networks, . . .
- all sorts of clustering and classification techniques (decision trees, . . .)
- association rules
- ad-hoc methods

WIAWTTAS

Association rules

We are interested in connections between sets of items: products, or molecules, or visits to websites. Rules should be like: “if you buy bread, you usually buy butter too”. And: can you identify a customer by the products he/she buys? (Yes, to a large extent.)

Given a database with n records (customers, transactions), where every record consists of several items (products). The **support** of a set of items is the number of records that contain that set, usually divided by n . A set with high support (above some threshold) is called **frequent**.

For example: 20% of the customers in a supermarket buy both bread, butter and cheese — and possibly more.

Now suppose that out of the α customers that buy a set A , β buy a set B too. Then we can say that the **association rule** $A \Rightarrow B$ has **confidence** α/β .

For example: it might be that 80% of the customers that buy butter and cheese, also buy bread.

Now we are interested in rules $A \Rightarrow B$ with both high confidence *and* high support, i.e., high support for the itemset $A \cup B$ (20% in the above case).

There is an extensive literature on association rules, in particular on the following aspects:

- **efficient** algorithms to find them
- how to select the **interesting** ones
- how to deal with non-Boolean attributes

For this last issue one can use **fuzzy logic**, where instead of 0/1 (not-buy vs. buy) intermediate values can occur. For example: age can be “young” to an extent of 0.35.

product = item customer = transaction	1	2	3	4	5	6	7	8	9
1	1	1	0	0	1	1	1	1	0
2	1	0	1	0	0	0	0	1	1
3	0	1	1	0	1	0	1	0	0
4	1	0	1	0	1	1	0	1	1
5	0	0	0	0	1	0	0	0	0
6	0	1	0	0	1	0	1	0	0

Even for this small database it is hard to see that the itemset $\{2, 5, 7\}$ is the only set with 3 items that is “bought” by at least 50% (some fixed bound, the so-called **support threshold**) of the customers.

Frequent itemsets naturally lead to association rules, like $\{2, 7\} \Rightarrow \{5\}$.

WIAWTTAS

Market basket analysis

Many stores try to analyze consumer behavior by doing so-called **market basket analysis**.

Some possible applications are:

- direct marketing: **recommender systems**
“Which customer shall we present some special offer?”
- analysis of current catalog
“Which products are of interest for a small shop, say in a railway station?”
- clustering and classification
“Can we group (or recognize) customers by means of their behavior?”

A more general goal might just be a better understanding of the consumers.

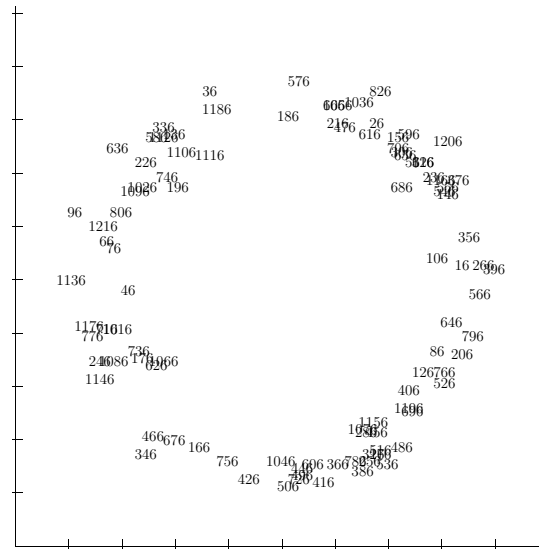
It turned out that it was quite easy to recognize customers by what they buy. In order to do so, one has to define a symmetric **distance measure** between customers.

For instance, one can simply let $distance(Cust_1, Cust_2)$ be equal to the number of products that the two customers $Cust_1$ and $Cust_2$ have in common. This can be refined by weights, or by “punishing” differences. We could *for example* define (in set notation; $|Z|$ is number of elements in Z ; $U \setminus V$ consists of elements in U that are not in V):

$$distance(X, Y) = |Universe| - |X \cap Y| + |X \setminus Y| + |Y \setminus X| .$$

You can “prove” this by the following experiment: if you visit a supermarket, observe the difference between your purchases and those from others, and remember the similarity between your current and previous baskets.

A large chain of shops wanted to obtain insight in their “positioning”, based on accumulated weekly sales. Pictures will look like:



Here we see 100 shops, clustered in the unit square, with respect to their common weekly sales.

In order to cluster, one has to define a distance notion. Now suppose that two shops have the following sales lists for this week:

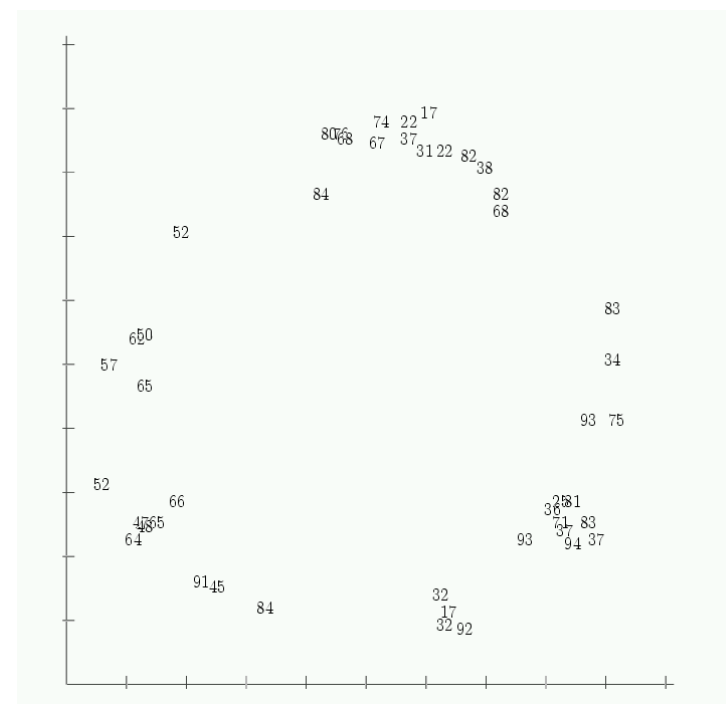
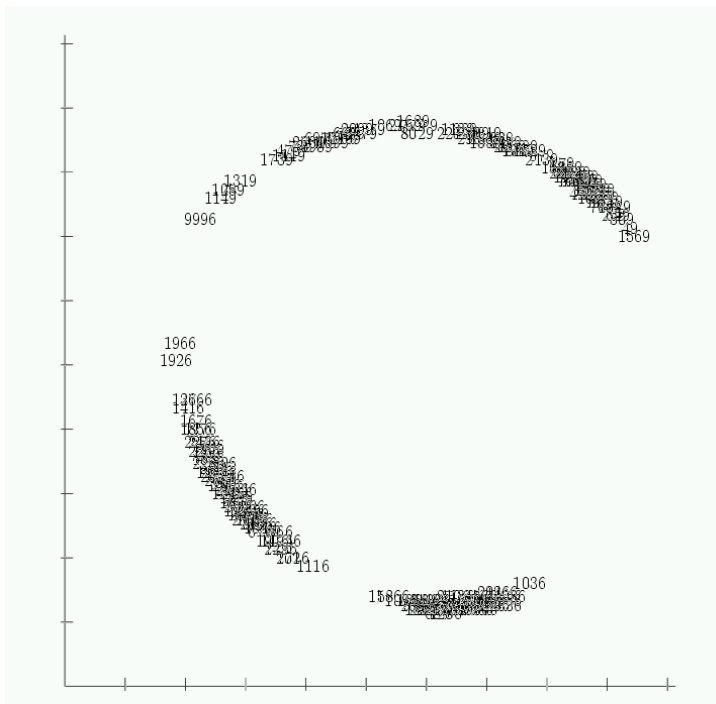
wine	bread	cheese	sausage	bananas
10	120	34	0	40
7	100	21	0	0

Their distance could be: $3 + 20 + 13 + 0 + 40 = 76$, or $3 + 20 + 13 = 36$, or $3/10 + 20/120 + 13/34$, or ...

It seems reasonable to normalize for the total sales of a shop.

Anyway, there are many possibilities!

Left we see a clear distinction between shops with and without restaurant. Right we look at zip code; in that case one clearly distinguishes “eating habits”.



In a supermarket there are *many* customers who can choose from a *large* stock. Association rules can be easily applied.

“Research” revealed that cigarette paper and tobacco are often bought together (what a surprise!), but also that special types of tobacco required special types of cigarette paper.

There is much interest in **hierarchies**. E.g., special brands in relation to more general categories.

The diapers—beer connection is a fairy-tale.

Discussion

How about privacy issues?

And why do *I* like crime novels?

Probably because of the suspense and the vivid dialogue, but also because of the **puzzles** in the “plot”: what is the explanation of the events so far?

By the way, this often results in perhaps artificial and sometimes rather far-fetched scenarios, e.g., so-called locked-room mysteries.

WIAWTTAS

Games

Games and puzzles give rise to complicated search algorithms, for example in chess, go, sudoku, ... In particular, if chance and contingency play a role (poker), you have serious problems: **Artificial Intelligence (AI)**!

Often an **evaluation function** is used, to judge situations.

Many programs **prune**: they try to eliminate parts of the search space.

We will examine the **minimax algorithm** from Von Neumann (1928) and the **α - β -algorithm** from 1956/58. And there many others ...

Games can be distinguished as follows:

	deterministic	chance
perfect information	chess, connect-four, go, checkers, sudoku, othello	monopoly, backgammon
incomplete information	battleships, mastermind	bridge, poker, scrabble

Deterministic: move consequences are fixed, no chances.

Perfect information: players know everything of a state.

Chess has an extensive literature.

Currently, the ultimate challenge seems the game **go**.

There are many different **strategies** to determine (an approximation of) the “game-theoretic value” of a game. **Shannon** (1950; entropy, information theory) conceived three types:

type A compute everything up to a certain depth (“game tree”), and use an evaluation function there

type B sometimes go deeper (if it is turbulent: “quiescence”); use “heuristic” function for guidance

type C goal oriented human search

A and B are brute-force, C is more “knowledge-based”.

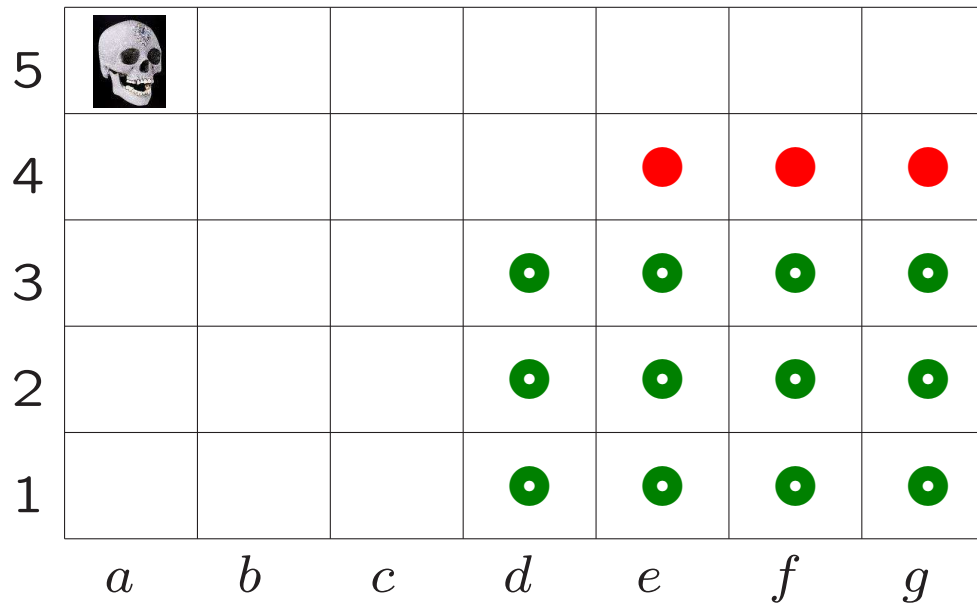
Some people distinguish three levels of game solving:

ultra-weak the game-theoretic value of the initial state is known: “one can win connect-four”

weak as above, and an optimal strategy is known: “start in the middle column, . . .”, see later

strong in every legal position an optimal strategy is known

The game of **Chomp** is played on a rectangular bar of chocolate, of which the players, taking turns, eat a rectangular part at the bottom right. He/she who eats the (poisonous) top left piece, has lost.



first bite: *d3*

second bite: *e4*

Claim: if you start, you can always win at Chomp! Indeed, if you can win by taking the piece at the bottom right, then it is fine. If not, the adversary clearly has a (for him/her) winning bite. You could have taken that same bite in the first place, and win!

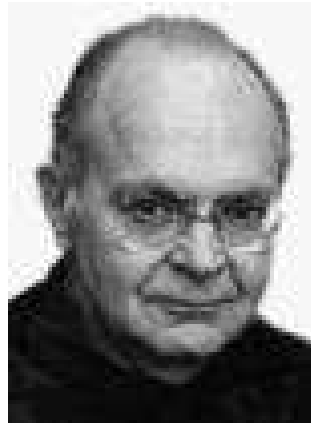
This argument is called **strategy stealing**.

So Chomp is solved ultra-weakly, we do not know the real winning move . . .

For example, for 2×2 and 2×3 Chomp the bottom right piece “wins” (more general for squares: take the piece to the bottom right of the poisonous one); for 3×4 Chomp take the piece from the middle row, third column.

WIAWTTAS

Mathematical plays



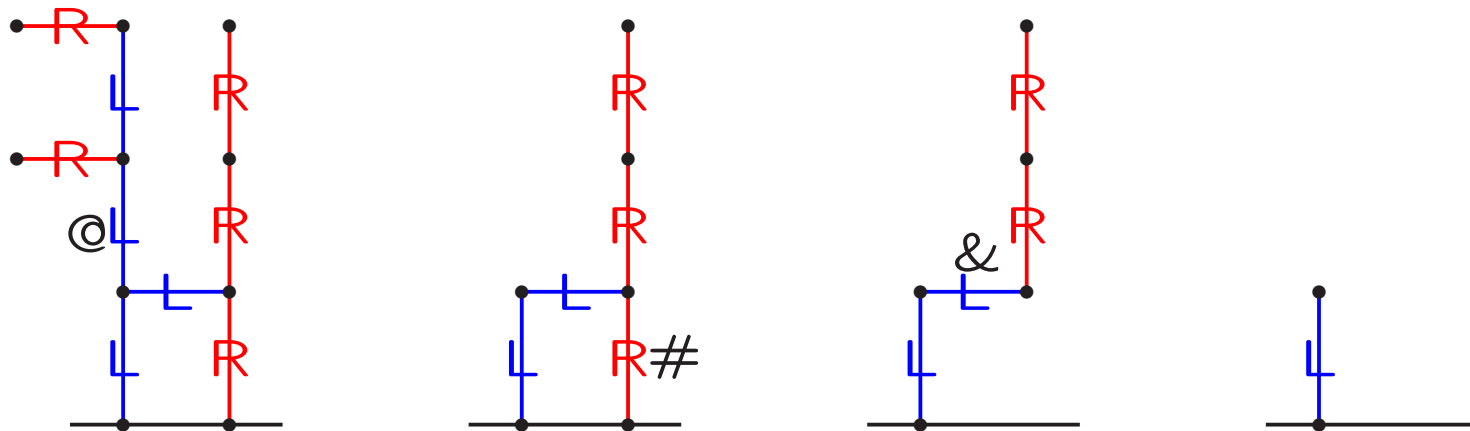
Donald E.(Ervin) Knuth
1938, US
NP; KMP
T_EX
change-ringing; 3:16
The Art of Computer
Programming



John H.(Horton) Conway
1937, UK → US
 C_0 , C_1 , C_2 , C_3
Doomsday algorithm
game of Life; Angel problem
Winning Ways for your
Mathematical Plays

Surreal numbers

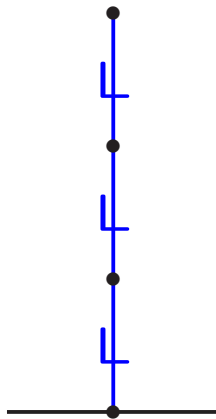
When playing **Hackenbush** **Left** and **Right** take turns, and remove a **bLue** or a **Red** line, respectively, after which all lines that lose contact with the ground are removed. *If you cannot play, you have lost!*



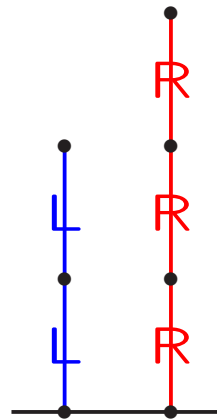
Left chooses @, **Right** chooses # (silly), **Left** chooses & and wins since **Right** cannot play.

By the way, **Right** can win here, whoever starts!

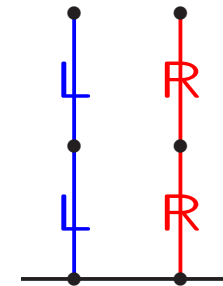
When playing Hackenbush, what is the **value of a position**?



value 3



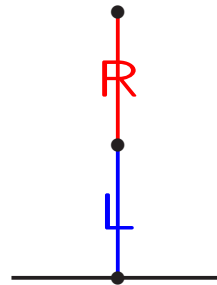
value $2 - 3 = -1$



value $2 - 2 = 0$

If the value is positive (> 0), **Left** can always win (whoever begins; in the example on the left with advantage 3), if the value is negative (< 0) **Right** can always win, and if the value is 0 the player to begin loses.

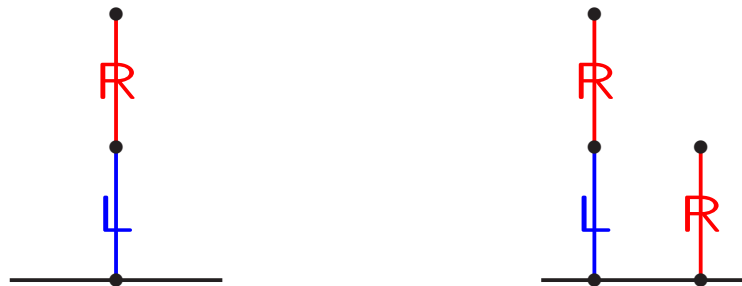
But what is the value of this position?



If **Left** begins, he/she wins immediately; if **Right** begins, **Left** has another move, and also wins. So **Left** always wins. Therefore, the value is > 0 .

Question: is the value equal to 1?

If the value in the left hand side position would be 1, de value of the right hand side position must be $1 + (-1) = 0$, and the player to begin should lose here. Is this true?



No: if **Left** begins, **Left** loses, and if **Right** begins **Right** can also win. So **Right** always wins (i.e., can always win), and therefore the right hand side position is < 0 , and the left one is between 0 and 1.

We denote the value of left hand side position by $\{ 0 \mid 1 \}$.



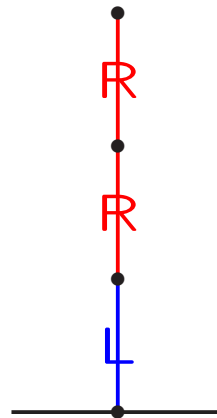
Note that the right hand side position does have value 0: the player to begin loses. And so we have:

$$\{ 0 \mid 1 \} + \{ 0 \mid 1 \} + (-1) = 0,$$

and apparently $\{ 0 \mid 1 \} = 1/2$.

We denote the value of a position where **Left** can play to (values of) positions from the set L and **Right** can play to (values of) positions from the set R by $\{ L \mid R \}$.

An example:



The value is $\{ 0 \mid \frac{1}{2}, 1 \} = \frac{1}{4}$.

The value appears to be the “simplest” number that is between left and right set.

In this way we define **surreal numbers**: “neat” tuples of sets of surreal numbers, defined earlier.

We start with $0 = \{ \emptyset \mid \emptyset \} = \{ \text{NOTHING} \mid \text{NOTHING} \} = \{ \mid \}$: the game where the player to begin has no opportunities at all, and therefore loses.

And then $1 = \{ 0 \mid \}$ and $-1 = \{ \mid 0 \}$.

And $42 = \{ 41 \mid \}$.

And $\frac{3}{8} = \{ \frac{1}{4} \mid \frac{1}{2} \}$.

And $\pi = \{ 3, 3\frac{1}{8}, 3\frac{9}{64}, \dots \mid 4, 3\frac{1}{2}, 3\frac{1}{4}, 3\frac{3}{16}, 3\frac{5}{32}, \dots \}$.

The real numbers (the set \mathbf{R}) are special surreal numbers (the set \mathbf{S}).



The so-called **Dali-function** $\delta : \mathbf{R} \rightarrow \mathbf{S}$ takes care of the “embedding”: $\delta(1) = \{ 0 \mid \} = 1$. Bu there is more ...

www.tondering.dk/claus/surreal.html

For example, we define:

$$\varepsilon = \{ 0 \mid \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots \},$$

an “incredibly small positive number”, and

$$\omega = \{ 0, 1, 2, 3, \dots \mid \} = \{ \mathbf{Z} \mid \emptyset \},$$

a “terribly large number, some sort of ∞ ”.

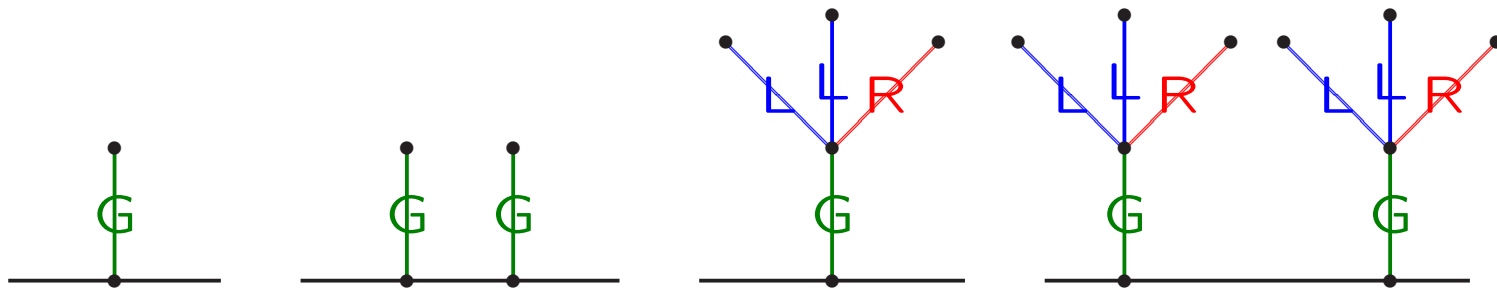
Then it turns out that:

$$\varepsilon \cdot \omega = 1$$

— if you have a well-defined multiplication ...

And then you also have $\omega + 1$, $\sqrt{\omega}$, ω^ω , $\varepsilon/2$, and so on!

For **Hotchpotch-Hackenbush** we also have **Green** lines, that can be taken by *both* players.

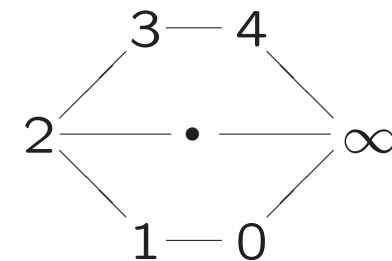
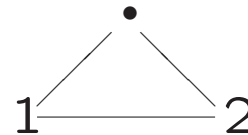
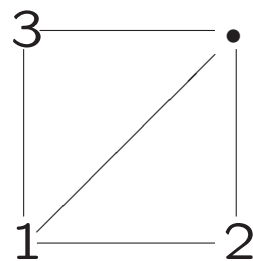
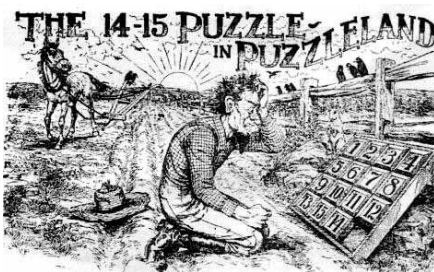


The leftmost position has value $* = \{ 0 \mid 0 \}$ (that appears to be not a surreal number!), because the player to begin can win by taking the **Green** line. The second from the left is $* + * = 0$ (player to begin loses).

The second from the right is again a win for the player to begin. The rightmost position is a win for **Left** (whoever begins), and so is > 0 .

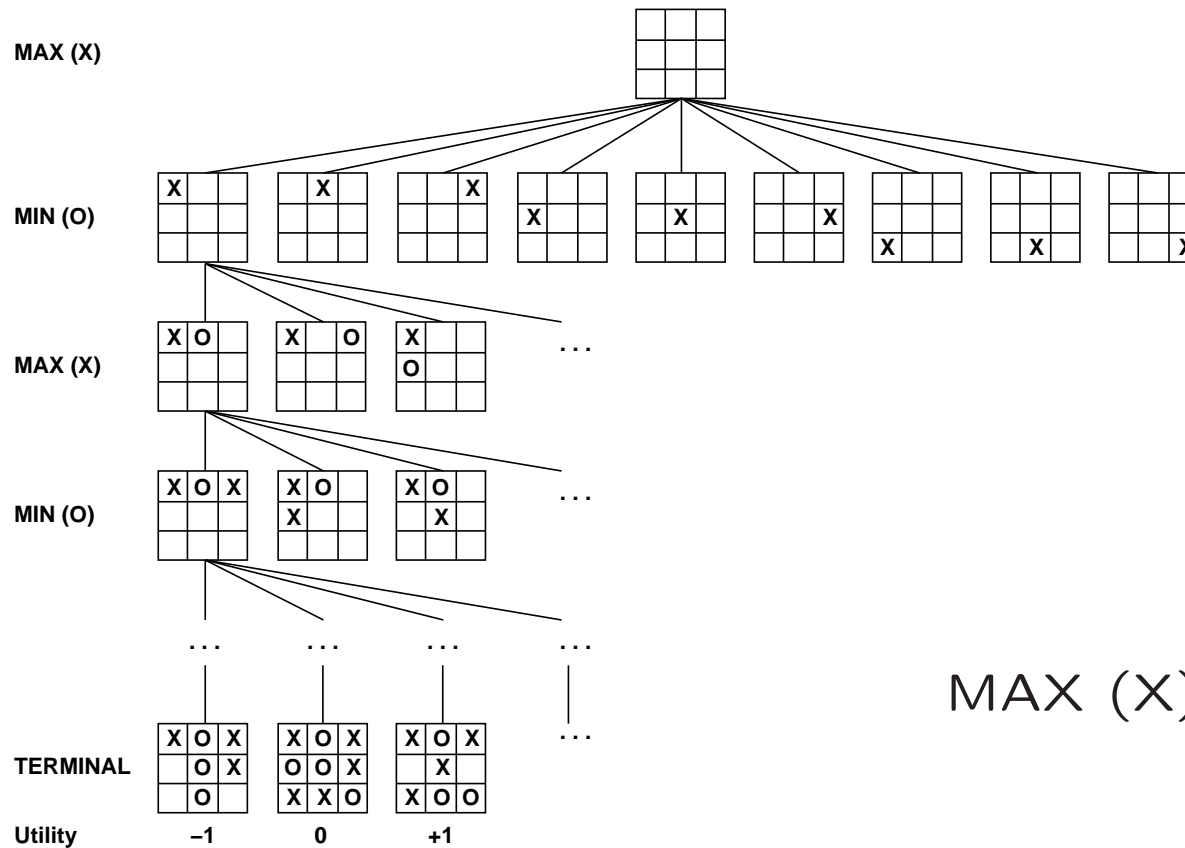
For a **sliding puzzle** one needs a **graph** where all nodes except one have a unique number. A number may slide to its neighbour if that has no number. Question: which positions can be reached from a fixed given one?

Examples: the 15-puzzle, and the graphs below. The right-most one is Wilson's Tricky Six Puzzle (cf. Rubik's cube).



WIAWTTAS

Games — Computer science



Tic-tac-toe

two players:
MAX (X) and MIN (O)

5478 states

Tic-tac-toe is an example of a two-person deterministic zero-sum game with full information, where the players take turns in doing “legal” moves. MAX must find a **strategy** that leads to a final winning position, whatever MIN does. This strategy must counter every possible move of MIN.

A **utility function** (= payoff function) gives the value of final states. Here we use: $-1/0/1$; for backgammon it is: $-192 \dots + 192$.

Symmetry arguments allow for removal of many states.

Maxy and **Mini** play the following simple game: first **Maxy** chooses a horizontal row, and then **Mini** chooses a vertical column.

3	12	8
2	4	6
14	5	2

For example: **Maxy** chooses row 3, and then **Mini** chooses column 2; that gives final result 5.

Maxy wants the heighest number possible, **Mini** the lowest.

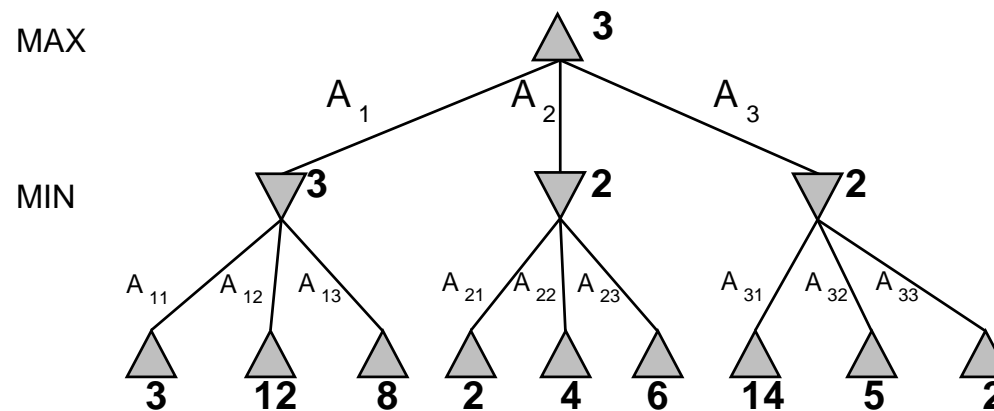
How to analyze this game?

If **Maxy** chooses row 1, **Mini** chooses column 1 (gives 3);
if **Maxy** chooses row 2, **Mini** chooses column 1 (gives 2);
if **Maxy** chooses row 3, **Mini** chooses column 3 (gives 2).
So **Maxy** chooses row 1! This is a so-called **brute force** argument: we examined each and every possibility.

3	12	8
2	?	?
14	5	2

Now we notice that the analysis (the **minimax-algorithm**) goes the same if we do not know the value of the two question marks. The **α - β -algorithm** remembers the best and worst possibilities, and prunes if there cannot be an improvement on those.

In tree shape:



The **minimax-algorithm** is recursive: in leaves one takes the evaluation function, in MAX-nodes the maximum of the children, in MIN-nodes the minimum of the children, and MAX- and MIN-nodes interchange.

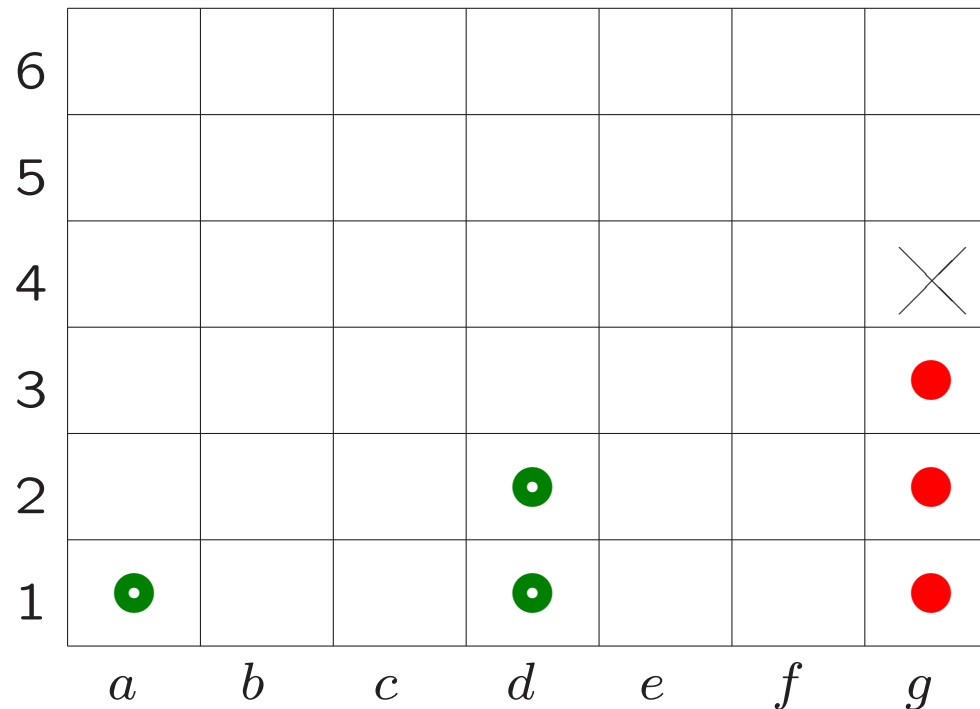
The tree above models **one move**, or **two ply**.

It is important to have **heuristics** (rules of thumb) that order the moves. Some examples:

null-move examine good adversary moves first (skip your own move, so to speak)

killer if a move prunes somewhere, it might also do so elsewhere

tabu-search remember (very) bad moves



Connect-four

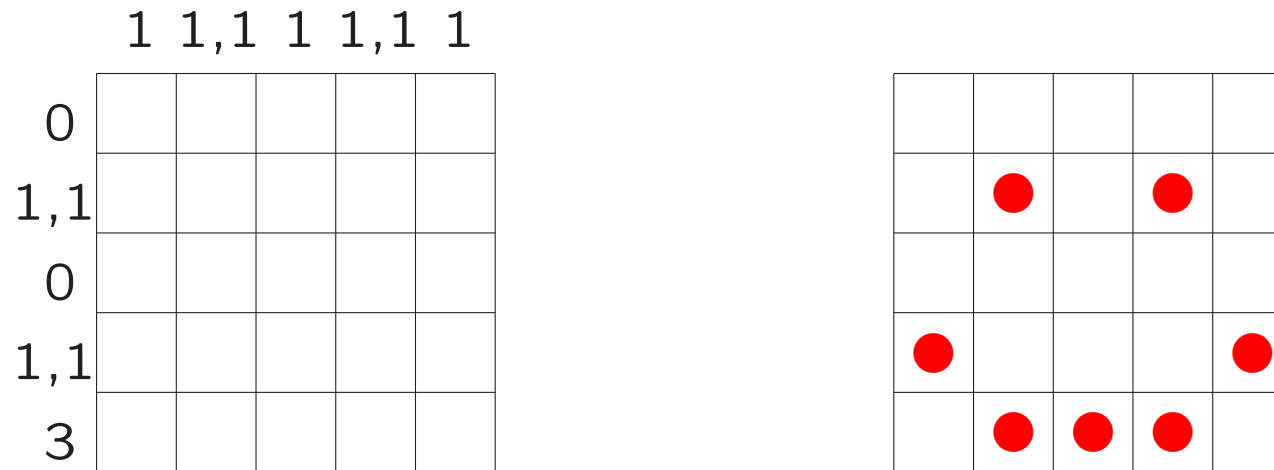
With **green** to play, *g4* is both for null-move and killer heuristic the proper move.

The winning series for **green** is: *d1!* – *d2* – *d3!* – *d4* – *d5!* – *b1* – *b2* (a ! means: unique winning move).

WIAWTTAS

Tomography

Nonograms (Japanese puzzles) look like this:



Next to each row and column the lengths of consecutive series of **red** squares are given.

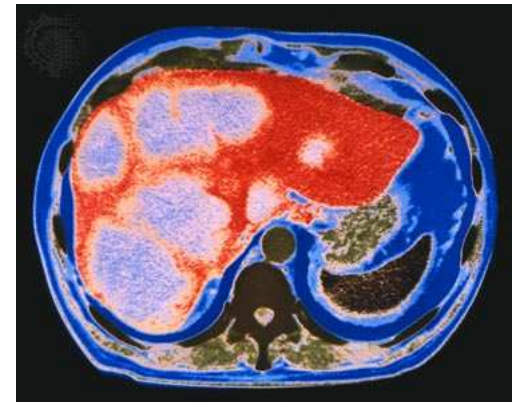
0000001010000001000101110

Tomography deals with the following problem:

How to reconstruct an object from its **projections**?

Examples:

- Solve nonograms
- How do organs look, given CT-scans?
- Where are the “holes” in a diamond?



WIAWTTAS

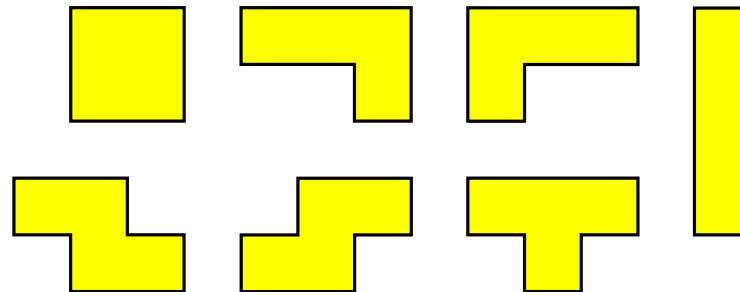
Tetris

There are several interesting questions attached to a game like **Tetris**:

- How to play well? (AI — Artificial Intelligence)
- How hard is it? (complexity)
- What might happen?

It has been shown that certain Tetris-problems are **NP-complete** (joint work with researchers from MIT), that you can reach almost all configurations, but that not all problems are “decidable”.

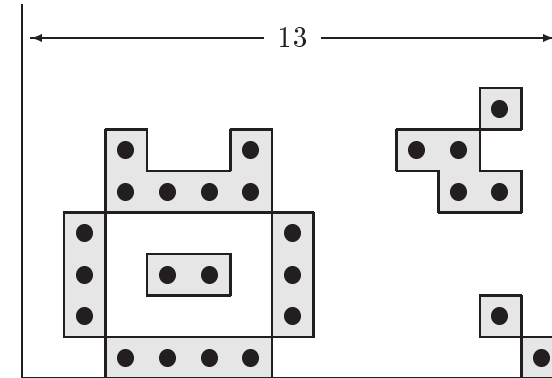
The 7 Tetris-pieces:



Random pieces fall down, and filled lines are cleared. The question “Is it possible, given a finite ordered series of these pieces, to clear a partially filled game board?” is NP-complete.

If someone clears the board, this is easy to verify: a “yes” is easy. If clearing is *not* possible however, up till now the only thing one can do to prove this is to check all possibilities, one by one: a “no” is difficult!

An “arbitrary” configuration:



This figure can be made by dropping 276 suitable Tetris-pieces in the appropriate way, see

<http://www.liacs.nl/home/kosters/tetris/>

Claim: on a game board of odd width every configuration is reachable.

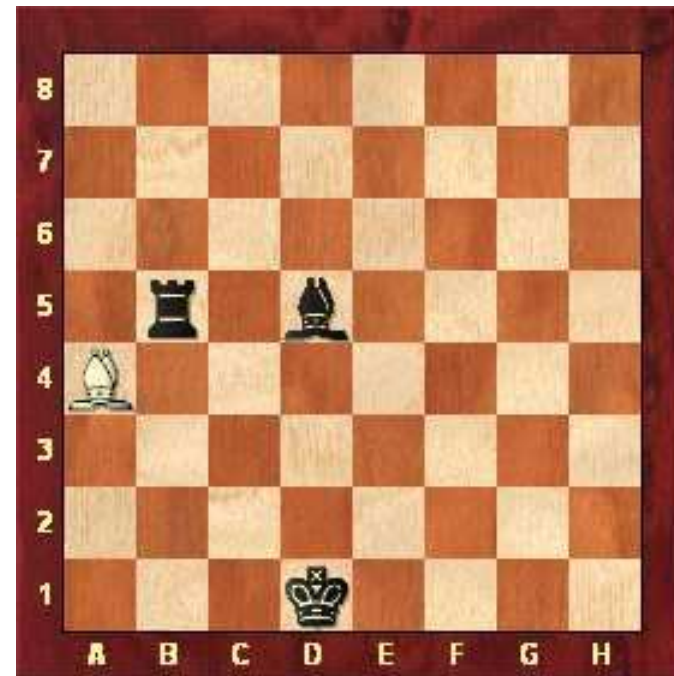
WIAWTTAS

Discussion

What is it good for?



Babson task
Yarosh, Tim Krabbé



retrograde analysis
Smullyan

See: The chess mysteries of Sherlock Holmes