Essential Thinking. The Art of Creative Thinking for Problem Solving

Antoni Ligęza

Faculty of EAIiIB
Department of Applied Computer Science
KRaKEn Research Group
https://kraken.edu.pl/

Kraków

Outline I

Outline II

References

1. Stuart J. Russel, Peter Norvig: *Artificial Intelligence. A Modern Approach.* Third Edition. Pearson, Prentice Hall, Boston, 2010. `http://aima.cs.berkeley.edu/`.
2. Ivan Bratko: *Prolog Programming for Artificial Intelligence*. Fourth Edition, 2011. Pearson, Addison Wesley, 2012. `http://www.pearsoned.co.uk/HigherEducation/Booksby/Bratko/`
3. Frank van Harmelen, Vladimir Lifschitz, Bruce Porter (Eds.): *Handbook of Knowledge Representation*. Elsevier B.V., Amsterdam, 2008.
4. Michael Negnevitsky: *Artificial Intelligence. A Guide to Intelligent Systems.* Addison-Wesley, Pearson Education Limited, Harlow, England, 2002.
5. Adrian A. Hopgood: *Intelligent Systems for Engineers and Scientists.* CRC Press, Boca Raton, 2001.
6. Joseph C. Giarratano, Gary D. Riley: *Expert Systems. Principles and Programming.* Fourth Edition, Thomson Course Technology, 2005.

References

1. George Polya: *How to Solve it?*. Princeton University Press, 1945; PWN 1993. `http://en.wikipedia.org/wiki/How_to_Solve_It`.
2. John Mason, Leone Burton, Kaye Stacey: *Thinking Mathematically*. Addison-Wesley, 1985; WSiP, 2005.
3. Mordechai Ben-Ari: *Mathematical Logic for Computer Science.* Springer-Verlag, London, 2012.
4. Michael R. Genesereth, Nils J. Nilsson: *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1987.
5. Peter Jackson: *Introduction to Expert Systems.* Addison-Wesley, Harlow, England, 1999.
6. Antoni Ligęza: *Logical Foundations for Rule-Based Systems.* Springer-Verlag, Berlin, 2006.

References

1. Krzysztof R. Apt: *Principles of Constraint Programming*. Cambridge University Press, Cambridge, UK, 2006.
2. Krzysztof R. Apt and Mark Wallace: *Constraint Logic Programming Using ECLiPSe*. Cambridge University Press, Cambridge, UK, 2006.
3. Rina Dechter: *Constraint Processing*. Morgan Kaufmann Publishers, San Francisco, CA, 2003.
4. Antoni Niederliński: *A Quick and Gentle Guide to Constraint Logic Programming via ECLiPSe*. PKJS, Gliwice, 2010 (`http://www.pwlzo.pl/`).
5. Roman Bartak: *On-line Guide to Constraint Programming*. http://kti.mff.cuni.cz/ bartak/constraints/index.html.
6. `http://en.wikibooks.org/wiki/Prolog/Constraint_Logic_Programming`.
7. `https://www.minizinc.org/`

Three Problems to Start

1. How to double the average speed? (A problem given by Max Wertheimer to Albert Einstein in 1934)
   `https://www.youtube.com/watch?v=q398AqtTEL8`
2. The Money Change Problem. Being given (unlimited) number of coins of several different types = positive integer values, say $m_1, m_2, \ldots, m_N$, where N is the number of types (e.g. 1 PLN, 2 PLN, 5 PLN; here $N = 3$) how to find the numbers (positive integers or zero) $c_1, c_2, \ldots, c_N$ such that
   $$S = \sum_{i=1}^{N} c_i m_i,$$
   where S is a given positive integer? In other words, how to make the change of some S with predefined coins? Is this always possible? And perhaps so that the number of coins is minimal?
3. Being given a limited, definite length fence mesh how to fence (enclose, inclose) the largest possible area?

What is worth learning?

A bit provocative position statement

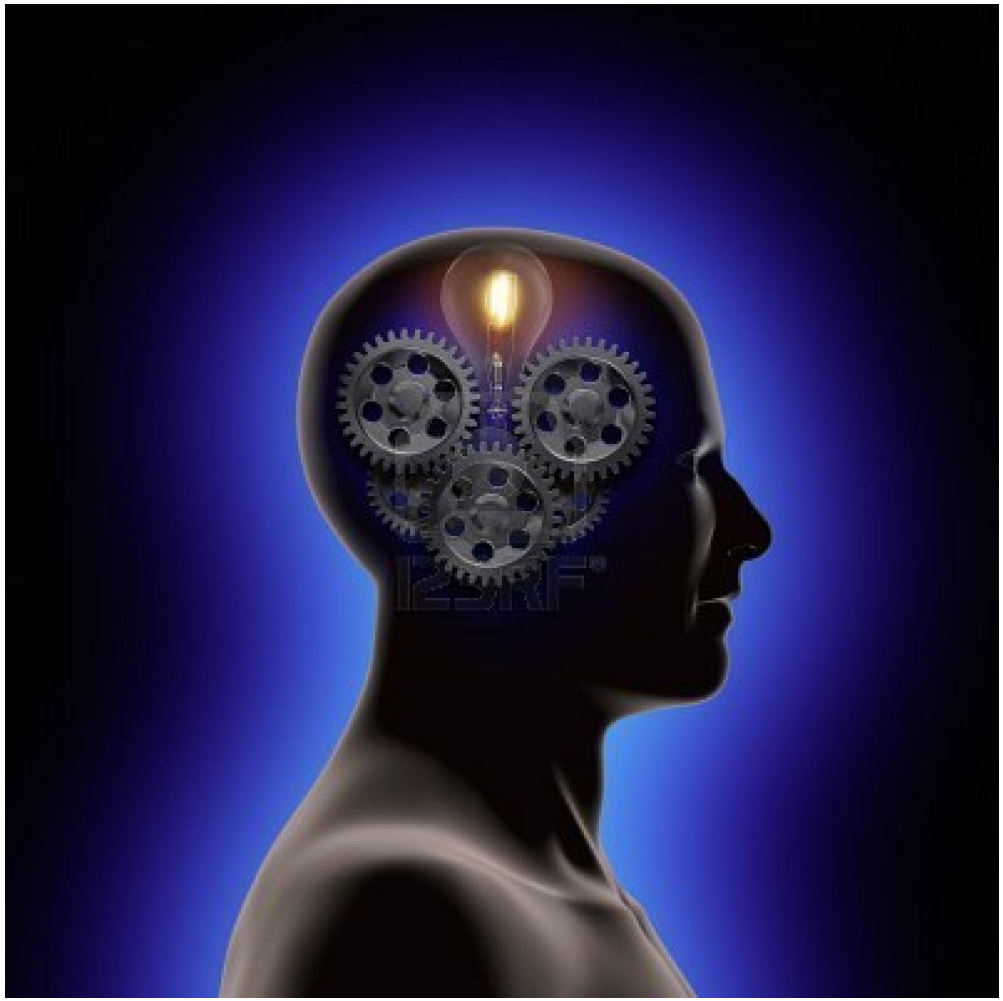
- Languages — enable communication and knowledge representation; Wie viele Sprachen du sprichst, sooft mal bist du Mensch; Goethe
- **Problem Solving** — analytical thinking; cross-curricular competencies,
- Learning — training your mind, persistent learning, quick learning, focused learning, learning on-demand, ...

Note that languages include also:

- Mathematics, including Arithmetics, Algebra, Geometry,...
- Logics,
- programming languages: Prolog, MiniZinc, Python, Julia, Java,...
- see: 10 languages over time
  `https://www.youtube.com/watch?v=Og847HVwRSI`
- music,
- ...???...

This *lecture* is all about **Creative Problem Solving**. Languages is a must!

Problem Solving — What is the Essence of it?



Knowledge + Intelligence $\Longrightarrow$ Problem Solving

Some inspiring questions

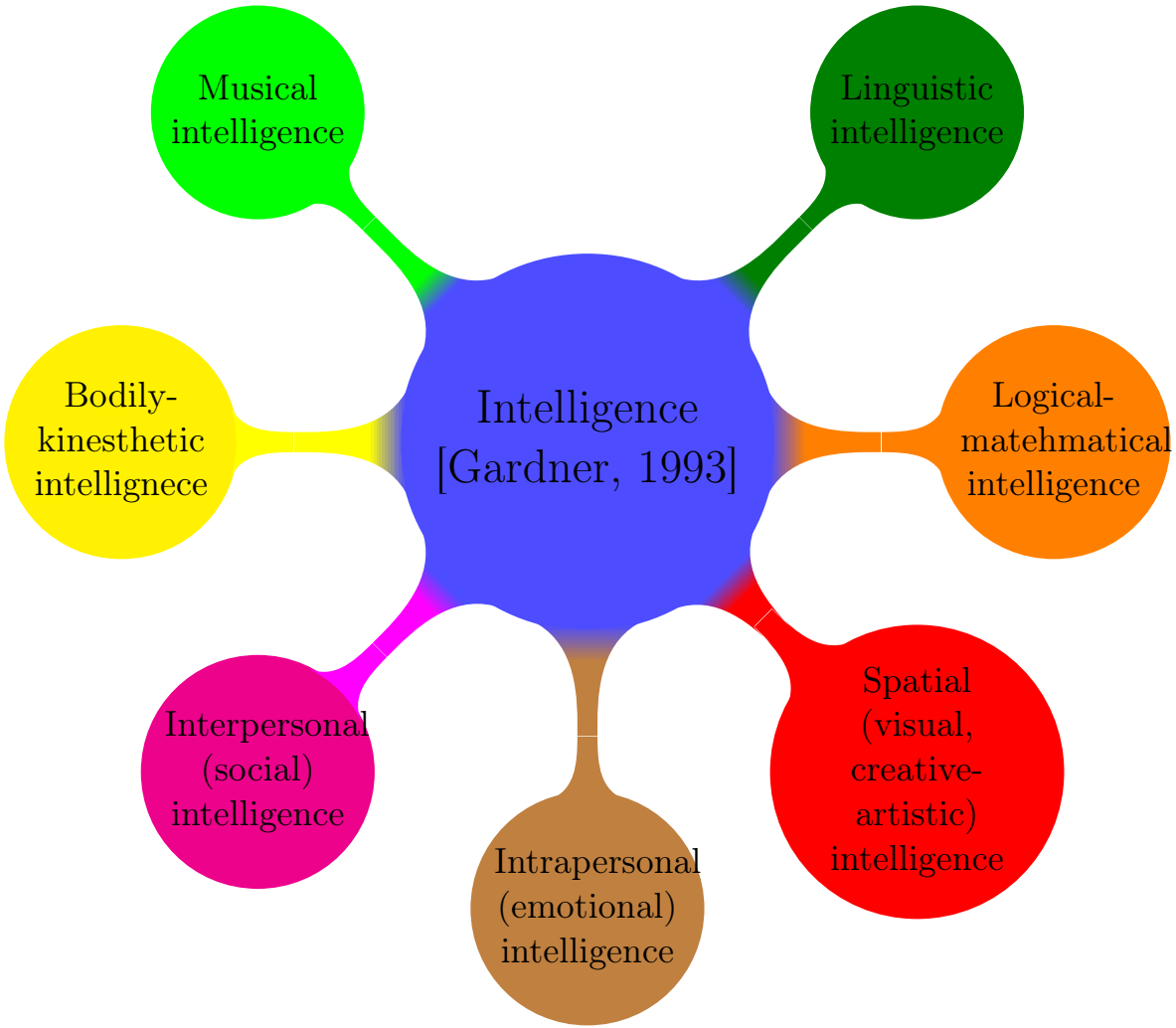
- What is the essence of thinking? How is it performed?
- Does only man think? What about animals and machines?
- What is the essence of intelligence?
- Can one learn/improve intelligence? Test/measure/evaluate?
- Can we have intelligent machines? More intelligent than people?

Some practical questions

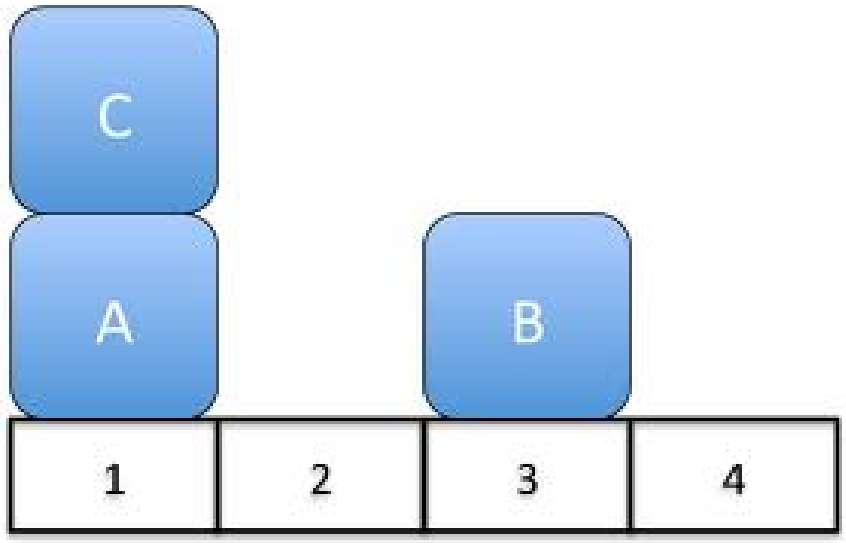
- What is the essence of problem solving? How is it performed?
- What is knowledge? What is the role of knowledge?
- Relationship between knowledge and intelligence?
- What is a problem? A solution?
- How to represent and process knowledge? Methods of reasoning?
- Can we have mechanical intelligence? Tools for problem solving?

Intelligence — What is the Essence of it?

1



What is a Problem? Means-Ends Analysis

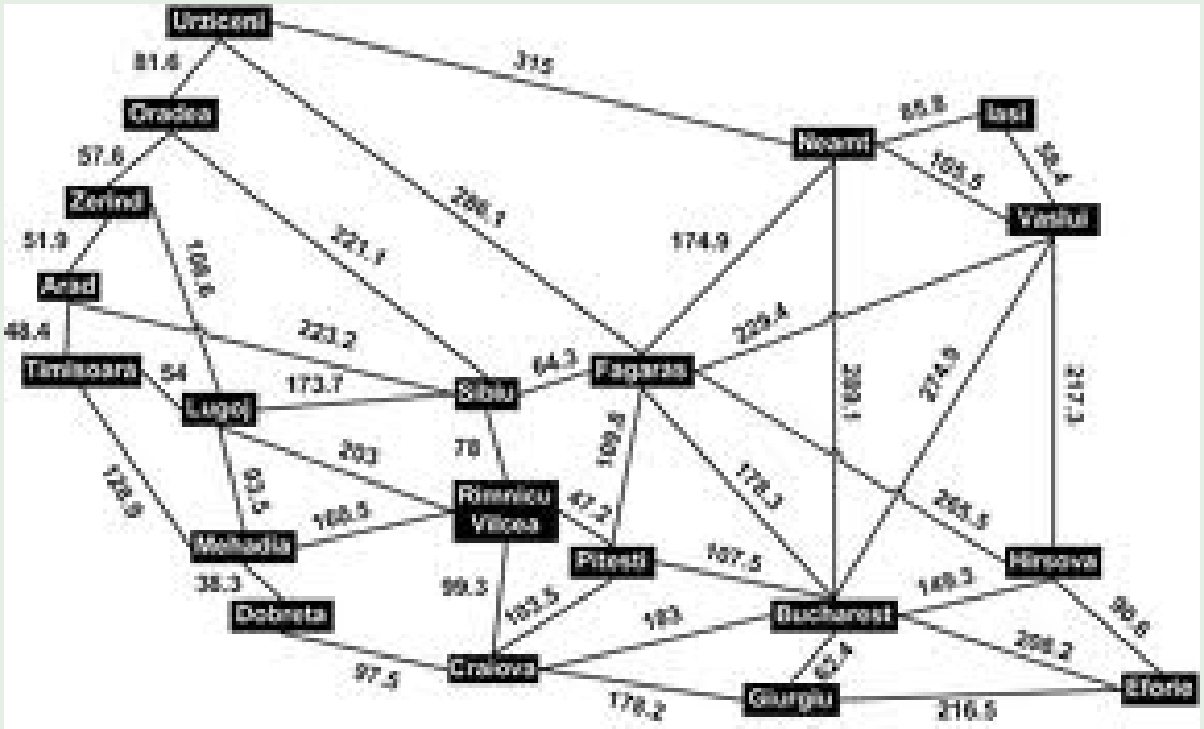


Nonlinear problem

- goal: ON(A,B) and ON(B,C),
- ON(B,C) — one-step, but wrong,
- ON(A,B) — two-steps, but also wrong.

A Generic Example — Graph Search

Path Finding for Route Planning

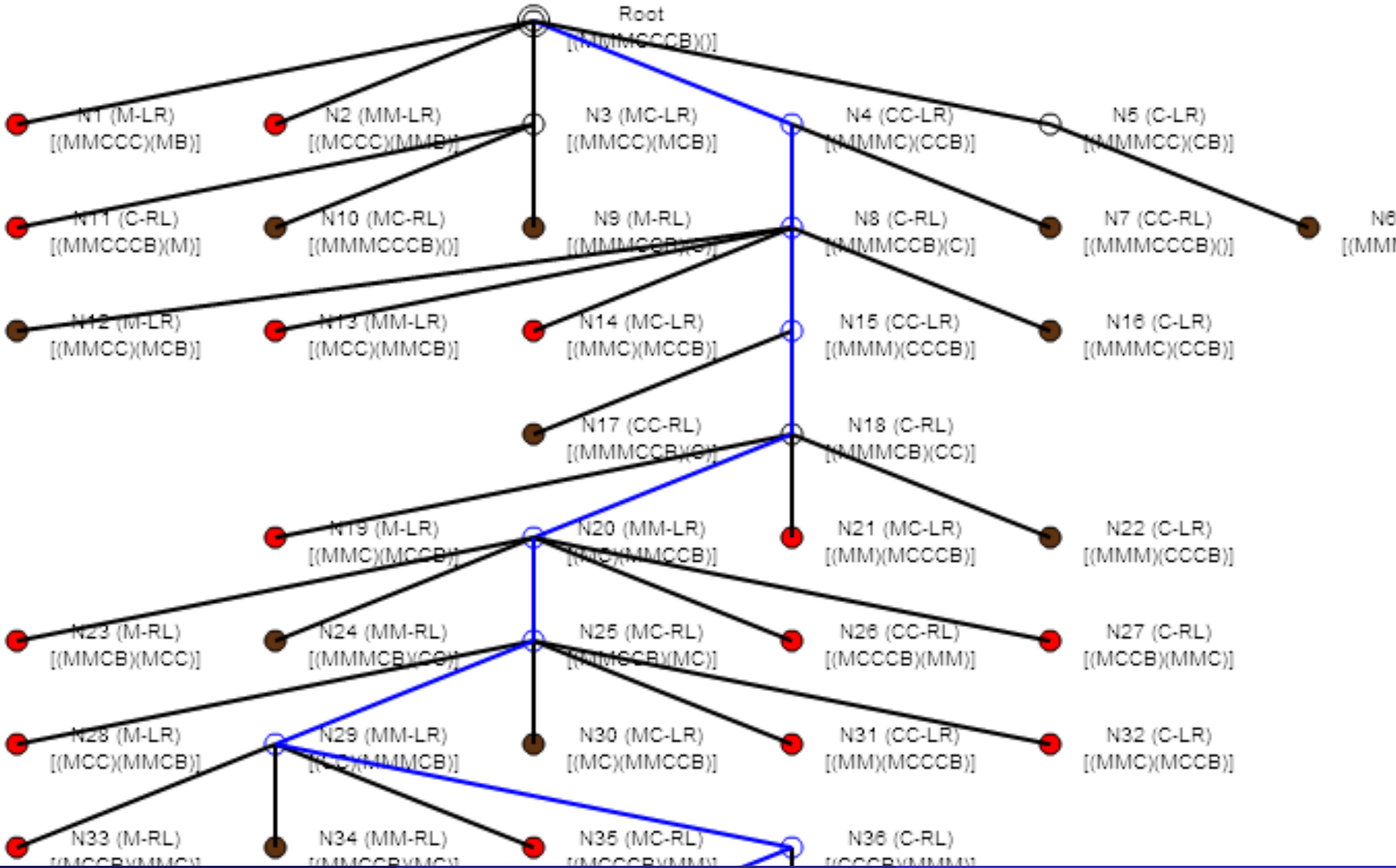


A Generic Example — Planning

Missionaries and Cannibals

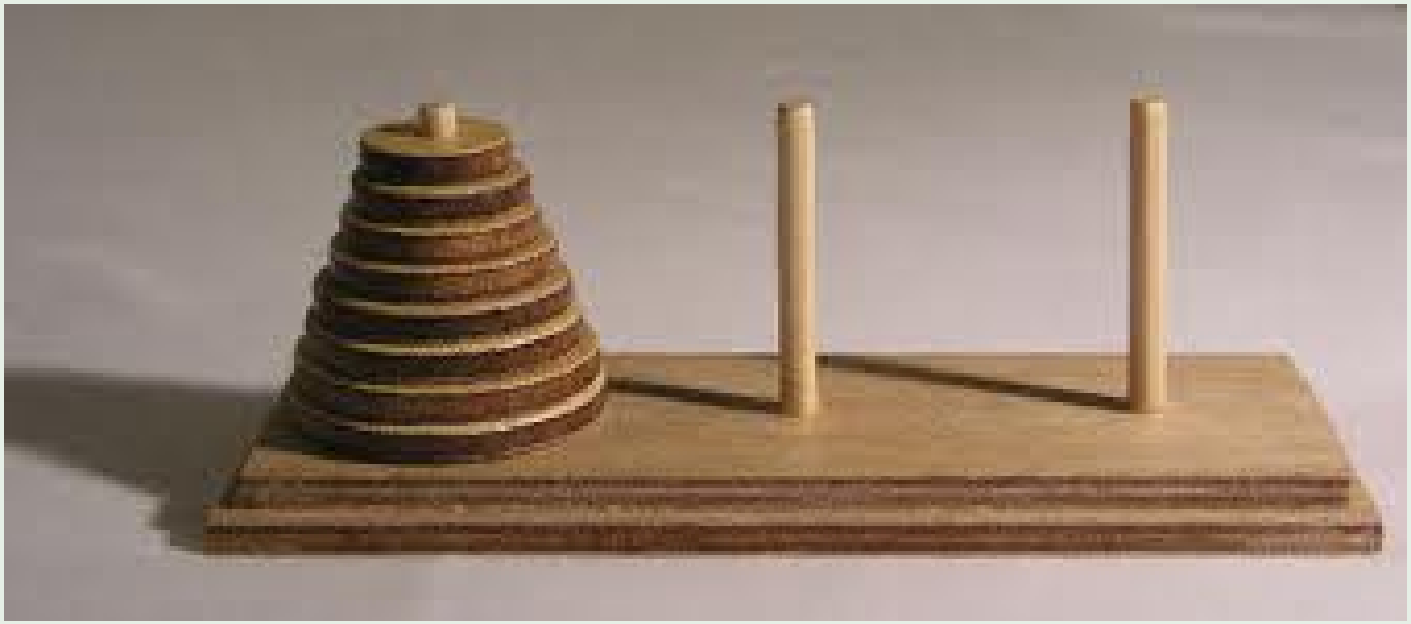
Try your skills:
`http://www.transum.org/software/River_Crossing/`

A Model and a Method: Graph/Three and Backtracking Search

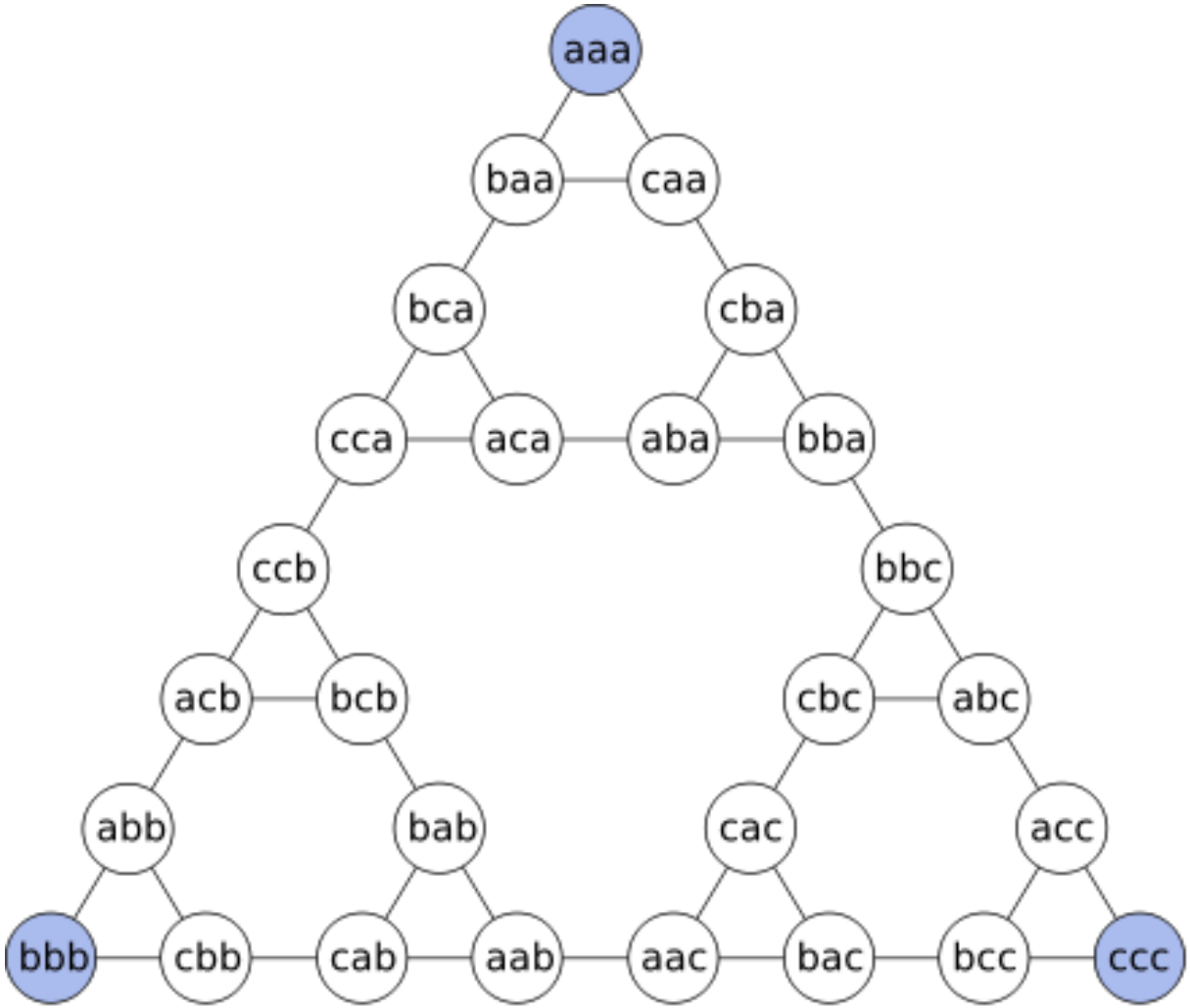


Examples — Search vs. Decomposition

Towers of Hanoi



Three generic examples



A Generic Problem Example

Let us give a try:
`http://www.japaneseiqtest.net/`

A Cryptoarithmetic Problem — Constraints Satisfaction

The rules:

- All letters are digits.
- Different letters are different digits.
- Letters on leading positions are non-zero.

A cryptoarithemtic problem

```
   SEND
+  MORE
------
  MONEY
```

Try your skills online: `http://r27.jp/quiz/send-more-money/`
Or follow the solution: `https://mindyourdecisions.com/blog/2018/09/06/send-more-money-a-great-puzzle/`

Einstein Riddle I

Einstein's riddle:

1. There are 5 houses in five different colors.
2. In each house lives a person with a different nationality.
3. These five owners drink a certain type of beverage, smoke a certain brand of cigar and keep a certain pet.
4. No owners have the same pet, smoke the same brand of cigar or drink the same beverage.

The question is: Who owns the fish?

Hints:

Einstein Riddle II

- the Brit lives in the red house
- the Swede keeps dogs as pets
- the Dane drinks tea
- the green house is on the left of the white house
- the green house's owner drinks coffee
- the person who smokes Pall Mall rears birds
- the owner of the yellow house smokes Dunhill
- the man living in the center house drinks milk
- the Norwegian lives in the first house
- the man who smokes blends lives next to the one who keeps cats
- the man who keeps horses lives next to the man who smokes Dunhill
- the owner who smokes BlueMaster drinks beer
- the German smokes Prince
- the Norwegian lives next to the blue house
- the man who smokes blend has a neighbor who drinks water

Einstein Riddle III

It is believed that Einstein wrote this riddle. He said that 98% of the world could not solve it.

Try your skills...

Or observe the solution:
`https://www.youtube.com/watch?v=ELVWdaNESkk`

Or try tools for Constraint Programming: Prolog or MiniZinc.

Towards a Problem Definition

A PROBLEM is when one wants to change the World to achieve some goal.

Components defining a PROBLEM

- A system in some environment,
- An ontology of the domain,
- A state-space,
- An initial state,
- A defined goal,
- A set of rules of transformation,
- A set of constraints,
- Perhaps some heuristics.

Lessons learned

Problem solving: search, backtracking, decomposition

- basic problem solving method is Backtracking Search,
- Decomposition (Problem Reduction) is power!
- a stable, appropriate search space must be defined,
- one can use a tree or a graph as search model,
- one can use a AND-OR tree or a AND-OR graph for decomposition,
- a search method is necessary: DFS, BFS, Dijkstra, UC, A*,...,
- appropriate formalism is power!
- constraints are useful!
- constraint propagation is power!
- heureka: important, but how does it work?

Analytical Thinking $\Leftrightarrow$ Brute Search

Analytical thinking vs. brute search

As far as now: Backtracking Search + Decomposition + Constraint Propagation work fine.

The spoiled chessboard problem

Another Example: Four-Digit Palindrom Case

Four Digit Palindrom

- a four digit palindrom: 1221, 7337, 2992,...
- observe: 1221:11=111, 7337:11=667, 2992:11=272,...
- Hypothesis: Every four-digit palindrom number is divisible by 11.

Analytical thinking vs. brute search

- is the hypothesis true or not?
- try several examples; try to invent a counterexample,
- try to induce regularity — or check all cases?
- prove or disprove!

Analytical + Inductive Thinking $\Leftrightarrow$ Brute Search

Lessons learned

Analytical thinking — problem solving

- basic problem solving method is search; but: combinatorial explosion!
- decomposition is power! But: interaction of subproblems!
- a stable, adequate search space must be defined; but: how to choose it?
- one can use a tree or a graph as search model; but repeated states vs.memory consumption!
- one can use a AND-OR tree or a AND-OR graph for decomposition,
- a search method is necessary; but which one?
- appropriate formalizm is power! But: how to find it?
- constraints are useful! How to explore them?
- constraint propagation is power!
- heureka: important; but: how does it work?

Analytical + Inductive Thinking + Backtrack Search + Experience + Learning

Problem Solving - what is necessary?

A word on toolkit

- language — its roles,
- knowledge representation formalism,
- knowledge processing tools — operators,
- problem statement,
- search space; state-space,
- constraints,
- heuristics,
- search strategy; memory vs. repeated search,
- domain ontology,
- the goal — explicit (exact state) or implicit (criterion),
- path to the goal vs. final solution.

Proof by Contradiction

Indirect Proof

There are:
- rational numbers: 1,2,3,...,1/2, 1/3, 3/5,...
- irrational numbers: π, e, $\sqrt{2}$, $\sqrt{3}$.

A Problem — prove that:

$$\sqrt{2}$$

is an irrational number.

See: `https://www.youtube.com/watch?v=sRDwsfNDXak`

Solution Existence Only: Proof by Cases

There are:
- rational numbers: 1,2,3,...,1/2, 1/3, 3/5,...
- irrational numbers: π, e, $\sqrt{2}$, $\sqrt{3}$.

Operations on irrational numbers can lead to rational numbers, e.g.

$$\sqrt{2} \cdot \sqrt{2} = 2$$

Problem: do there exists two irrational numbers, say a and b, such that:

$$a^b$$

is a rational number?
The answer should be: YES or NO.

Some example problems

Polynomial –> 0

Does there exists a polynomial function satisfying the following conditions:

- it is always strictly greater than zero:

$$P > 0$$

- for any (small) $\epsilon > 0$, there exists a value of the polynomial less then ϵ

$$\forall \epsilon > 0 \quad \exists P_\epsilon : \quad P_\epsilon < \epsilon$$

- Prove or
- Disprove.

Problem Solving - some questions to be raised

About the problem and solution

- does any solution exist?
- is the solution unique?
- should we search for the first solution or all of them?
- can the solutions be compared/evaluated?
- should we search for satisfactory or optimal solution?
- is the optimal solution unique?
- does an optimal solution exists? Pareto optimal solutions?

About solutions

- candidate solution,
- admissible solution, legal solution,
- satisfactory solution,
- semi-optimal, ϵ-optimal, dominant solution, optimal solution.

General Problem Solver

GPS: Towards General Intelligence

- creation: A. Newell,J.C. Shaw, H.A. Simon; 1957-1959.
- general-purpose problem solver,
- means-ends analysis,
- objects, transformations, differences,
- recursion.

GPS: how it works?

- Method 1: transform object A into object B;
- Methods 2: apply operator Q to A;
- Method 3: reduce the difference d between object A and B.

Means-Ends Analysis

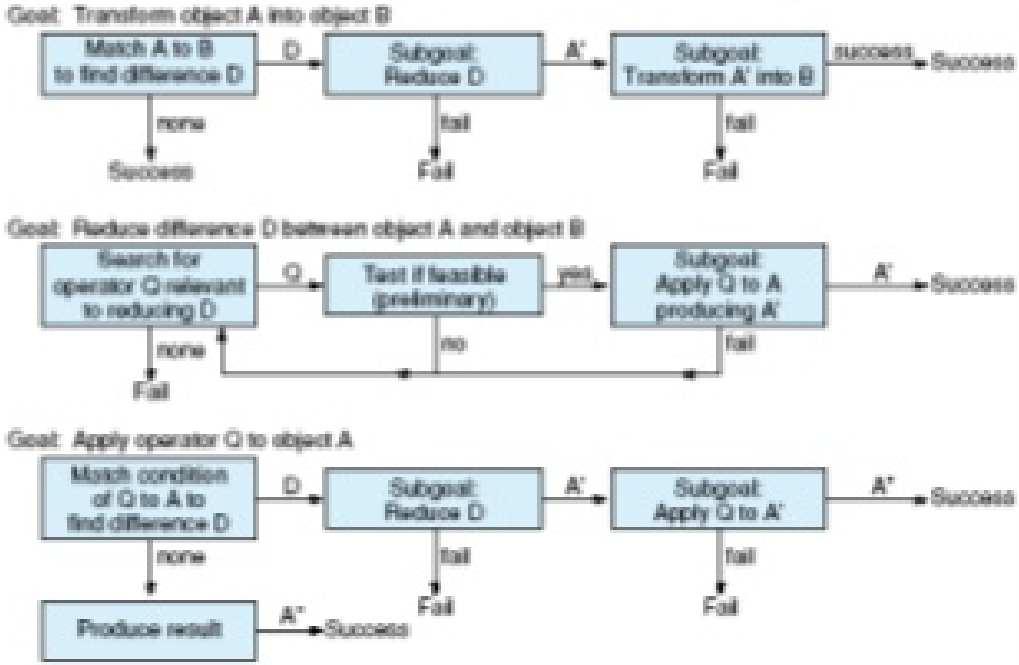
Principles of MEA

- explores the paradigm of goal-based problem solving,
- provides strategy of work at the conceptual level,
- is a universal method,
- roughly based on the concepts of distance and similarity.

MEA: main stages

- compare current state and goal state,
- find differences,
- find operator to reduce the difference,
- apply the operator; produce new state,
- repeat until success;
- backtracking and search are not excluded.

Fig 14.2 Flow chart and difference reduction table for the General Problem Solver, from Newell and Simon (1963b).



For the logic task of the text:

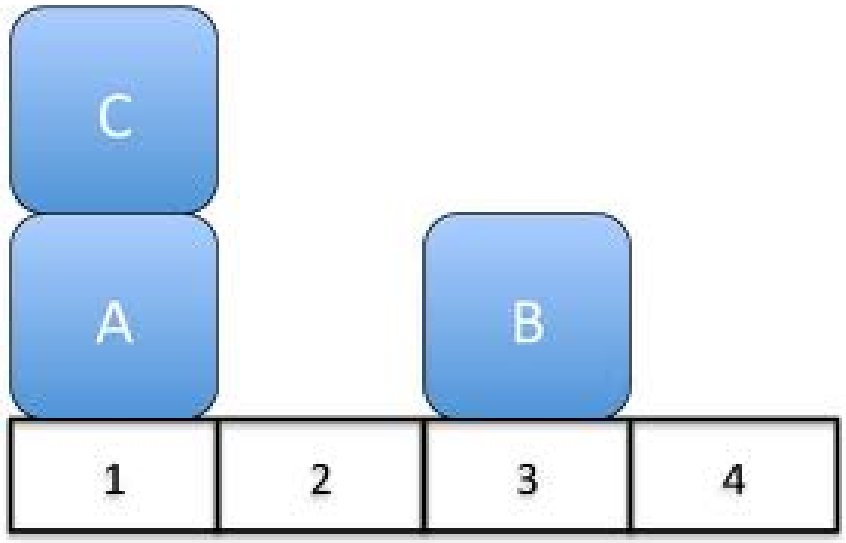
Feasibility test (preliminary)
- Is the main connective the same (e.g., A•B → B fails against P•Q)?
- Is the operator too big (e.g., (A•B)•(A•C) → A•(B•C) fails against P•Q)?
- Is the operator too easy (e.g., A → A•A applies to anything)?
- Are the side conditions satisfied (e.g., R8 applies only to main expressions)?

Table of connections

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
Add terms			X				X		X	X	X	X
Delete terms			X				X	X			X	X
Change connective					X	X	X					
Change sign					X							
Change lower sign		X			X	X						
Change grouping				X			X					
Change position	X	X										

X means some variant of the rule is relevant. GPS will pick the appropriate variant.

Problems with Means-Ends Analysis



Nonlinear problem

- goal: ON(A,B) and ON(B,C),
- ON(B,C) — one-step, but wrong,
- ON(A,B) — two-steps, but also wrong.

How to Solve It

Thanks to: George Polya: *How to Solve it?*. Princeton University Press, 1945; PWN 1993.

`http://en.wikipedia.org/wiki/How_to_Solve_It`.

G. Pólya: Four stages

- understand the problem,
- devise a plan,
- carry out the plan,
- revise/extend.

Auxiliary advice

- if failure, try simpler problem apply relaxation,
- if failure, try similar/related problem,
- partial solutions, auxiliary assumptions, auxiliary/less restrictions,
- approximate solutions.

Devise a Plan

Hints and Tips

- Guess and check,
- Make an orderly list,
- Eliminate possibilities,
- Use symmetry,
- Consider special cases,
- Use direct reasoning,
- Solve an equation.
- Look for a pattern,
- Draw a picture,
- Solve a simpler problem,
- Use a model,
- Work backward,
- Use a formula,
- Be creative,
- Use your head.

A List of Techniques — apart Search

Some Suggestions — Why Not Try?

- Analogy (Mapping to other problem),
- Generalization (Generalization),
- Induction (Induction from examples),
- Variation of the Problem (Modification, change, search),
- Auxiliary Problem (Subproblem, subgoal),
- Here is a problem related to yours and solved before (Pattern recognition, Pattern matching, Reduction; Case-Based Reasoning),
- Specialization (Specialization, instance),
- Decomposing and Recombining (Divide and Conquer),
- Working backward (Abduction; Backward chaining),
- Draw a Figure (Diagrammatic Reasoning),
- Auxiliary Elements (Extension).

Problem Solving: A List of Techniques

- Abstraction: solving the problem in a (simplified) model of the system
- Analogy: using a solution that solved an analogous problem
- Brainstorming: (for groups of people) suggesting a large number of solutions or ideas and combining and developing them until an optimum is found
- Divide and conquer: breaking down a complex problem into smaller ones
- Hypothesis testing: assuming a possible explanation to the problem and trying to prove (or, in some contexts, disprove) the assumption
- Lateral thinking: approaching solutions indirectly and creatively
- Means-ends analysis: choosing an action at each step to move closer to the goal
- Method of focal objects: synthesizing seemingly non-matching characteristics of different objects into something new
- Morphological analysis: assessing the output and interactions of an entire system
- Reduction: transforming the problem into another problem
- Research: employing/adapting existing solutions to similar problems
- Root cause analysis: eliminating the cause of the problem
- Trial-and-error: testing possible solutions until the right one is found
- Proof: prove that the problem cannot be solved; fail point = new start

Some Ideas about Tools

Tools - Why Not?

- Internet and Google,
- Excell,
- Wolfram Alpha `//www.wolframalpha.com/`,
- Prolog,
- MiniZinc,
- Toolboxes (e.g. Matlab),
- Specialized software,
- Mind Maps,
- ...???...

An Intuitive Introduction

```
?- [library(clpfd)].

?- X #> 3.
X in 4..sup.

?- X #\= 20.
X in inf..19\/21..sup.

?- 2*X #= 10.
X = 5.

?- X*X #= 144.
X in -12\/12.

?- 4*X + 2*Y #= 24, X + Y #= 9, [X,Y] ins 0..sup.
X = 3, Y = 6.

?- Vs = [X,Y,Z], Vs ins 1..3, all_different(Vs), X = 1, Y #\= 2.
Vs = [1, 3, 2], X = 1, Y = 3, Z = 2.
```

Basic Constraints

clp(fd) — intro

```
?- [library(clpfd)].

Expr1 #>= Expr2 % Expr1 is larger than or equal to Expr2

Expr1 #=< Expr2 % Expr1 is smaller than or equal to Expr2

Expr1 #= Expr2  % Expr1 equals Expr2

Expr1 #\= Expr2 % Expr1 is not equal to Expr2

Expr1 #> Expr2  % Expr1 is strictly larger than Expr2

Expr1 #< Expr2  % Expr1 is strictly smaller than Expr2
```

Basic Constraints

clp(fd) — intro

```
1 ?Var in Domain
2
3 ?Vars ins Domain
```

```
 Var is an element of Domain. Domain is one of:

   Integer
        Singleton set consisting only of Integer.
   Lower .. Upper
        All integers I such that Lower =< I =< Upper.
        Lower must be an integer or the atom inf, which de
        Upper must be an integer or the atom sup, which de
   Domain1 \/ Domain2
        The union of Domain1 and Domain2.
```

Basic Constraints

clp(fd) — intro

```
1 label(+Vars)
2
3 labeling(+Options, +Vars)
```

```
Labeling means systematically trying out values for
the finite domain variables Vars until all of them are grou
The domain of each variable in Vars must be finite.
Options is a list of options that let you exhibit
some control over the search process.
```

clp(fd) — intro

```
1 indomain(+Var)
```

```
Bind Var to all feasible values of its domain on backtracki
The domain of Var must be finite.
```

Basic Constraints

clp(fd) — intro

```
1 all_different(+Vars)
2
3 all_distinc(+Vars)

Vars are pairwise distinct.

The second command has stronger propagation
(can detect inconsistency).
```

The SEND+MORE=MONEY Example

Code

```
1  :- use_module(library(clpfd)).
2
3  puzzle([S,E,N,D] + [M,O,R,E] = [M,O,N,E,Y]) :-
4          Vars = [S,E,N,D,M,O,R,Y],
5          Vars ins 0..9,
6          all_different(Vars),
7                    S*1000 + E*100 + N*10 + D +
8                    M*1000 + O*100 + R*10 + E #=
9          M*10000 + O*1000 + N*100 + E*10 + Y,
0          M #\= 0, S #\= 0.
1
2  ?- puzzle(As+Bs=Cs), label(As).
3  As = [9, 5, 6, 7],
4  Bs = [1, 0, 8, 5],
5  Cs = [1, 0, 6, 5, 2] ;
6  false.
```

Bidirectional Factorial

Code

```
1  :- use_module(library(clpfd)).
2
3  n_factorial(0, 1).
4  n_factorial(N, F) :- N #> 0, N1 #= N - 1, F #= N * F1, n_factoria
5
6  ?- n_factorial(47, F).
7  F = 258623241511168180642964355153611979969197632389120000000000
8  false.
9
10 ?- n_factorial(N, 1).
11 N = 0 ;
12 N = 1 ;
13 false.
14
15 ?- n_factorial(N, 3).
16 false.
```

Summary: Types of Problems

An informal classification

- FORWARD CHAINING (deduction, rules, patterns),
- BACKWARD CHAINING (abduction, diagnostics, hypothetical reasoning),
- UPWARD INFERENCE (induction, model building),
- SEARCH — graph search, path finding,
- PLAN — plan generation,
- REDUCT — AND-OR graph search, AND-OR plans,
- GAME - adversarial search,
- CSP, CLP — search with constraints,
- SAT; logical modeling,
- OPT — optimal solution search,
- CI — Computational Intelligence problem (NN, Fuzzy, k-NN).

Finally: the type of problem defines the appropriate tools/methods!

A Generic Problem Example

http://www.japaneseiqtest.net/

Some example problems

Two eggs

You are given two eggs. There is a high building of n floors. An egg dropped from k-th floor brakes; but not for k' < k. Find k in a least number of trials.

Bicycle

What direction moves a bicycle, when the lower pedal is pulled backwards?

Formula

Find a formula for the sum: $a + aq + aq^2 + aq^3 + ...aq^n$. Finite/infinite case.

Two ships

Two ships move at constant speed. Find the smallest distance between them.

Fly problem

Two elephants are approaching each other from opposite direction with constant speed 2 km/h and 3 km/h. Initial distant is 1 km. A fly flies from one to the other of them and again with speed of 20km/h. What distance will it cover until the elephants meet?

Three generic examples: weighting

9 coins

- 9 identical coins; one is lighter
- how many weightings?

10 coins

- 10 identical coins; one is false
- how many weightings?

N coins

- N identical coins; one is lighter
- 3 weightings
- How big N?
- How big is N in case we know only that the coin is false?

Three generic examples: combinatorics

Pages

- Book pages numbered with 2989 digits
- how many pages?

Buckets

- Two buckets: 4 and 9 liters
- Produce exactly: 1, 2, 3, 4, 5, 6, 7, 8 liters

Squares on a Chessboard

- A chessboard 8x8 available
- How many squares can be found?
- What in the case of a NxN chessboard?

Three generic examples: planning

Desert: how many days?

- To cross a desert: 9 days (+ return)
- Two man; each can carry food for 12 days
- Food can be stored and retrieved

Raft + 3 + 2

- 3 man want to cross a river
- There are two boys with a raft of them
- The raft can carry one man only

Missionaries and Cannibals: 4 + 4

- 4 missionaries, 4 cannibals,
- a boat for two,
- M < C forbidden (M not 0)

Yet another example

3D Geometry Problem: Polyhedron Characteristics

- Give a polyhedron:
  - K - the number of edges,
  - N - the number of corners,
  - S - the number of walls.
- N+S-K=2
  - prove,
  - disprove

Examples

A cryptoarithemtic problem

$$\begin{array}{r} \texttt{SEND} \\ +\ \texttt{MORE} \\ \hline \texttt{MONEY} \end{array}$$

Examples

A cryptoarithemtic problem

```
   DONALD
+  GERALD
  ------
  ROBERT
```

Examples

A cryptoarithemtic problem

```
    CROSS
+   ROADS
------
  DANGER
```

Examples

A cryptoarithemtic problem

```
    TEN
+   TEN
 FORTY
------
 SIXTY
```