Artificial Intelligence

Gregorics Tibor

Information

- Requirements
  - Exam with a theoretical and practical parts: up to 60 points
    50 – 60: excellent, 40 – 49: good, 30 – 39: satisfactory,
    20 – 29: pass, 0 – 19: fail,
  - Final mark can be modified with the sum of the result of
    - short tests at the beginning of each lesson: up to 12 points
    - optional assignments week after week: up to 24 points
- Contact
  - https://canvas.elte.hu/
  - gt@inf.elte.hu (Tibor Gregorics)
  - shrph25@gmail.com (Zoltán Milacski)

References

- Slides can be found in system canvas:
  - https://canvas.elte.hu/
- S. Russel, P. Norvig: Artificial Intelligence - in Modern Approach, Panem-Prentice Hall, 2003.
- N. Nilsson: Principles of Artificial Intelligence, Springer-Verlag, 1982.
- N. Nilsson: Artificial Intelligence: a new synthesis, Morgan Kaufmann Publishers, 1998.
- E. Rich, K. Knigth: Artificial Intelligence, McGraw-Hill, 1991.

INTRODUCTION

1. What is artificial intelligence(AI)?

Strong AI
Its aim is to reproduce the human intelligence with computers

Skepticism of AI
Computer cannot be more clever than human

Weak AI

The artificial intelligence collects, develops and researches the principles and the methods that are useful to help the human mental activities with computers.

AI is not a subarea of informatics but an intention to make computers solve interesting new problems which, at this moment, people do better.

specializati

I **feel** you **are bored with** me **lately.**

I **think** you **have been angry with** me **recently.**

Pattern: ***I <a> you <b> me <c>.***

1. Why do you think that you <a> I <b> you <c>?
2. Let us suppose that I <b> you <c>. Would that make a difference?

Recall:
If a sentence has already occurred, the computer tries to answer differently or it can say: „I am getting tired of replying the same sentence over and over.”

Carry on: What else do you want to talk about?
I see. Please continue. This is very interesting.

two-player games (
chatting program (ELIZA,1966)

Romantic ages

All kinds of problems
with general tools

1956 1960 1970 1980 1990 2000 2010

specialization

History

[Projects]():
- two-player games (chess)
- chatting program (ELIZA,1966)

[Methods]():
- GPS, resolution (1966)
- Lisp (1958)
- artificial neural networks
- evolutionary algorithms

Romantic ages

All kinds of problems
with general tools

1956 1960 1970 1980 1990 2000 2010

specialization

History

Classical ages

Specialized methods on specialized problems

Romantic ages

All kinds of problems with general tools

Projects:

SHRDLU (1972),
BACON, AM
DENDRAL (1969-78),
MYCIN(1976)

Methods:

heuristic search,
knowledge representation
Prolog

1960 1970 1980 1990 2000 2010

specialization

History

Industrial ages

expert systems
knowledge based systems

Classical ages

Specialized methods on specialized problems

Romantic ages

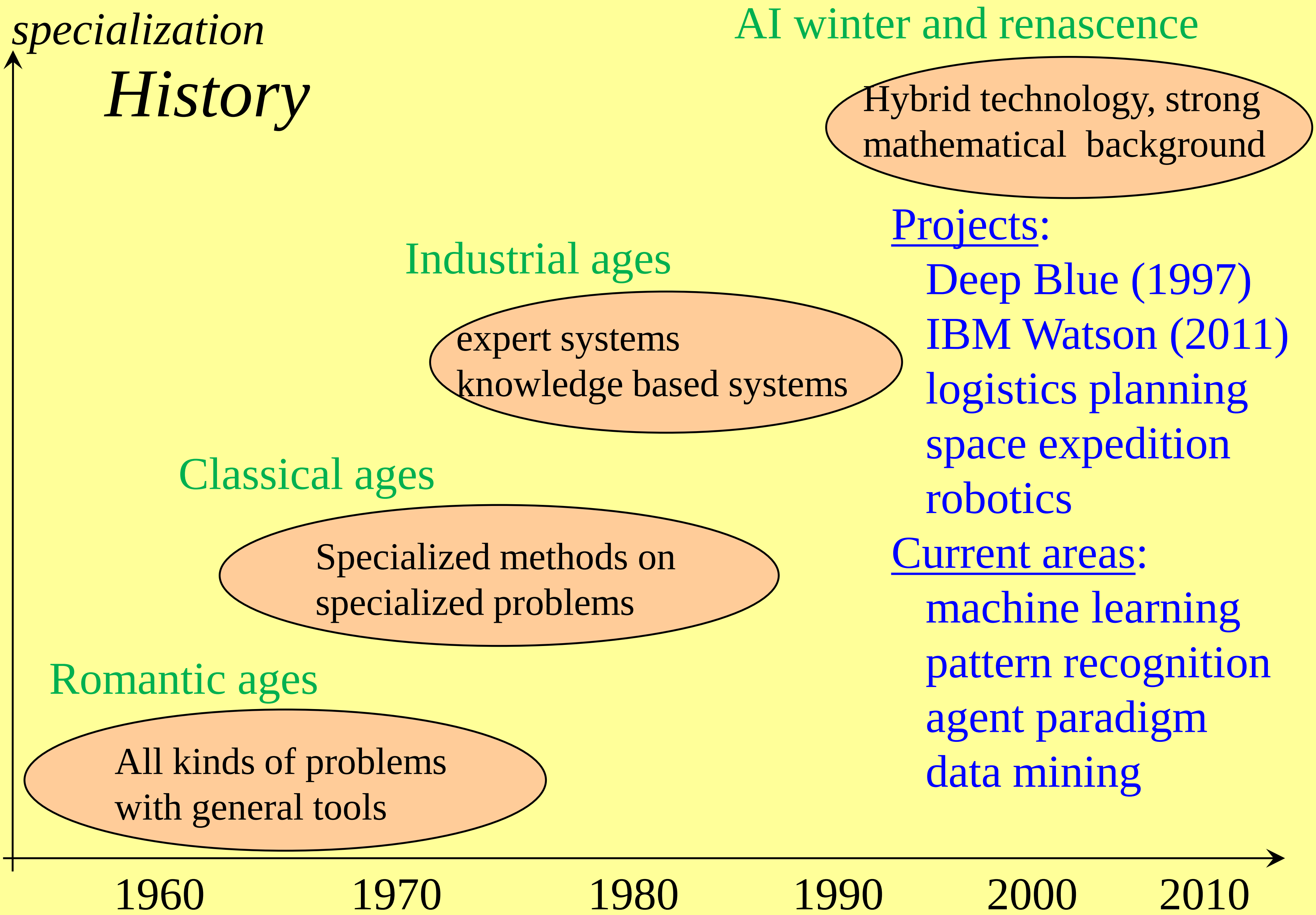
All kinds of problems with general tools

Projects: XCON(1982), PROSPECTOR (1979)

Methods:
- shells (IDE)
- knowledge-based technology
- non-monotone reasoning
- uncertainty management

1960 1970 1980 1990 2000 2010

History

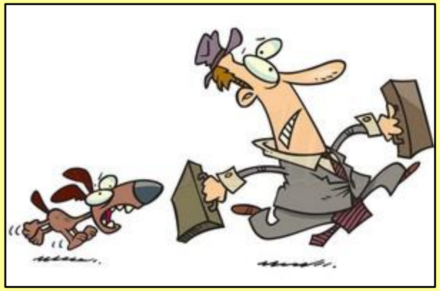

specialization
AI winter and renascence
Hybrid technology, strong mathematical background
Projects:
Deep Blue (1997)
IBM Watson (2011)
logistics planning
space expedition
robotics
Current areas:
machine learning
pattern recognition
agent paradigm
data mining
Industrial ages
expert systems
knowledge based systems
Classical ages
Specialized methods on specialized problems
Romantic ages
All kinds of problems with general tools
1960
1970
1980
1990
2000
2010

How can AI be recognised?

- **Problems** that are solved: difficult
  - Problem spaces of these problems are huge.
  - A traditional systematic search can not find the solution. Intuition and creativity (heuristics) is needed to avoid the combinatorial explosion.
- **Behavior** of the software: intelligent
  - Turing test vs. Chinese room argument
- **Methods** of the software: special
  - modeling techniques for problems
  - effective heuristic algorithms
  - machine learning methods

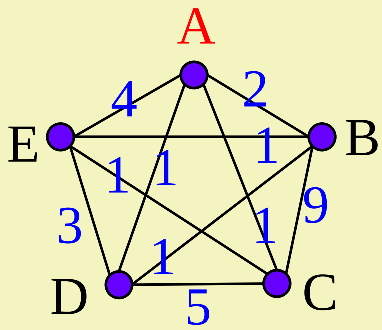
Features of intelligent systems
- language processing
- storing knowledge
- automatic reasoning
- learning

\+ machine vision, acting



Travelling salesman problem

The traveling salesman must visit every city in his territory exactly once and then return home covering the optimal total cost. (n cities and cost of each pair of cities are known)



n	(n-1)!
5	24
50	$6 \cdot 10^{62}$

possible solutions:

ABCDEA

ACBDEA ABDCEA ABCEDA

ABDECA

ADBECA …

problem space

7-11

$x=?,\ y=?,\ z=?,\ t=?$

$$\begin{cases} x + y + z + t = 7.11 \\ x \cdot y \cdot z \cdot t = 7.11 \end{cases}$$

$\Downarrow$

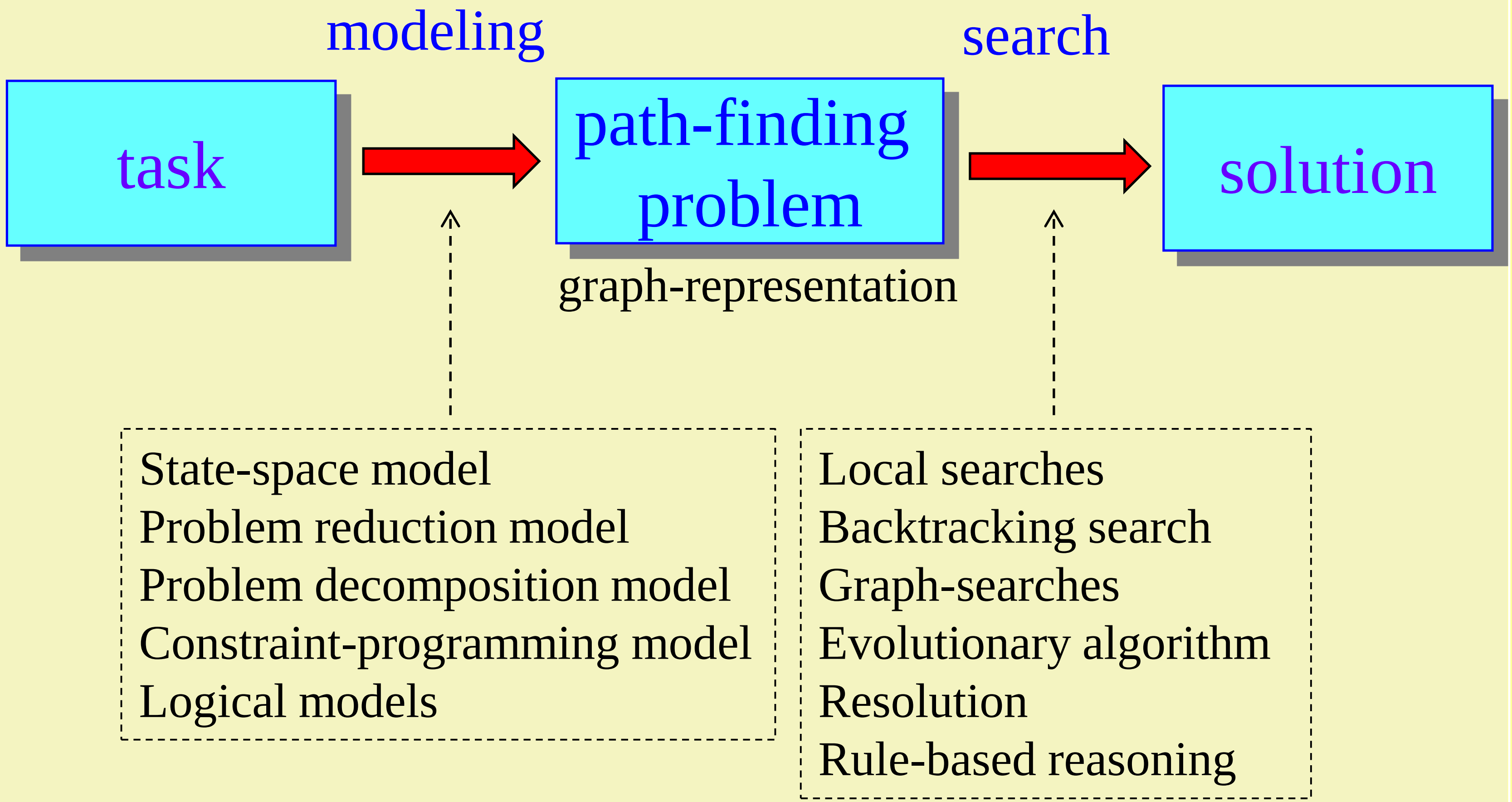
$x, y, z, t \in \{ 1 , \dots , 708 \}$

$$\begin{cases} x + y + z + t = 711 \\ x \cdot y \cdot z \cdot t = 711 \cdot 10^6 = 2^6 \cdot 3^2 \cdot 5^6 \cdot 79 \end{cases}$$

$x = 79$ $\quad$ $x = 2\cdot 79$ $\quad$ … $\quad$ $\cancel{x = 7\cdot 79}$ $\quad$ $x = 8\cdot 79$ $\quad$ $\cancel{x = 9\cdot 79 = 711}$

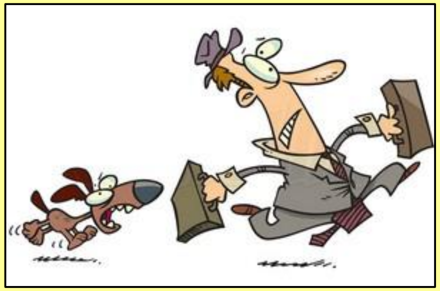
$$\begin{cases} y+z+t=632 \\ y\cdot z\cdot t=2^6\cdot 3^2\cdot 5^6 \end{cases} \qquad \begin{cases} y+z+t=553 \\ y\cdot z\cdot t=2^5\cdot 3^2\cdot 5^6 \end{cases} \qquad \dots \qquad \begin{cases} y+z+t=79 \\ y\cdot z\cdot t= 2^3\cdot 3^2\cdot 5^6 \end{cases}$$

2. Modeling & Search



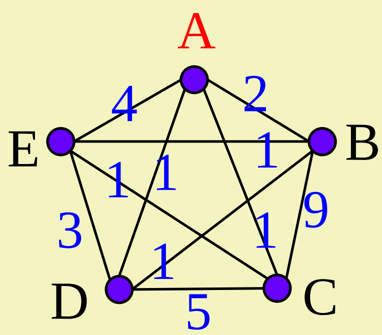
What does the modeling focus on?

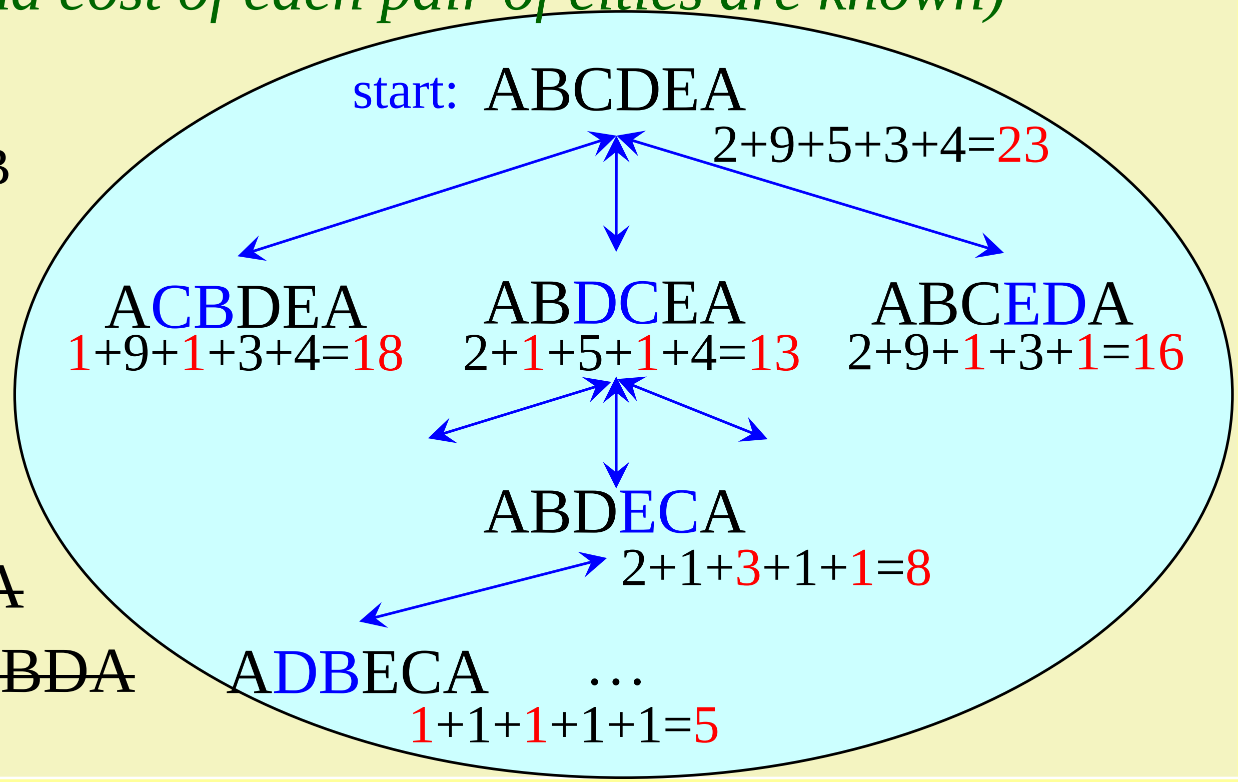
- ***Problem space***: contains the possible answers.
- ***Goal***: finding a correct answers (solutions).
- ***Ideas that help the search***:
  - Restricting the problem space: feasible answers
  - Selecting an initial item of the problem space.
  - Defining neighborhood relationships between the items of the problem space:
    - It helps to go through the problem space systematically.
  - Ranking the currently available items of the problem space during its traversal



Travelling salesman problem

The traveling salesman must visit every city in his territory exactly once and then return home covering the optimal total cost. (n cities and cost of each pair of cities are known)





~~AEDCBA~~

~~AEDBCA~~

~~ACEDBA~~

useless

~~ACEBDA~~

Path-finding problems

- Using an appropriate model, the problem space of a path-finding problem can be treated as an arc-weighted, directed graph, where the items of the problem space are symbolized by either nodes or paths. This graph might be infinite but the number of the outgoing arcs of one node is always finite, and there is a positive constant lower bound on the cost of the arcs (δ-graph).
- In order to solve these problems either a special goal node or a path driving form a start node to any goal node must be found. (Sometimes the optimal path is needed.)

Graph notations 1.

- nodes, arcs — $N,\ A\subseteq N\times N$ (infinite)
- arc from n to m — $(n,m)\in A\ (n,m\in N)$
- children of n — $\Gamma(n) = \{m\in N \mid \exists(n,m)\in A\}$
- parents of n — $\pi(n) \in \Pi(n) = \{m\in N \mid \exists(m,n)\in A\}$
- directed graph — $R=(N,A)$
- finite outgoing arcs — $|\,\Gamma(n)\,| < \infty\ (\forall n\in N)$
- cost of arc — $c:A\to\mathbb{R}$
- δ-property ($\delta\in\mathbb{R}^+$) — $c(n,m) \geq \delta > 0 \quad \forall(n,m)\in A$
- δ-graph — directed, arc-weighted, δ-property, finite outgoing arcs from a node

Graph notations 2.

- directed path $\alpha = (n,n_1),(n_1,n_2),\ldots,(n_{k-1},m)$
  $= <n,n_1,n_2,\ldots,n_{k-1},m>$

  $n \longrightarrow^{\alpha} m,\ n \longrightarrow m,\ n \longrightarrow M\ (M \subseteq N)$

  $\{n \longrightarrow m\},\ \{n \longrightarrow M\}\ (M \subseteq N)$

It is correct in spite of infinite number of paths since we have δ-graphs.

If no path, it is infinite.

- length of a path $|\alpha|$
- cost of a path $c^{\alpha}(n,m) := \Sigma_i\, c(n_{i-1},n_i)$
- optimal cost $c^*(n,m) := \min_{\alpha \in \{n \longrightarrow m\}} c^{\alpha}(n,m)$
  $c^*(n,M) := \min_{\alpha \in \{n \longrightarrow M\}} c^{\alpha}(n,m)$
- optimal path $n \longrightarrow^* m := \min_c \{\alpha \mid \alpha \in \{n \longrightarrow m\}\}$
  $n \longrightarrow^* M := \min_c \{\alpha \mid \alpha \in \{n \longrightarrow M\}\}$

Definition of graph-representation

- All path-finding problems can be described with a graph-representation. It is a triple (R, s, T) where
  - $R=(N, A, c)$ is a δ-graph (representation graph)
  - $s \in N$ is the start node
  - $T \subseteq N$ is the set of goal nodes.
- Solution of the problem:
  - finding a goal node: $t \in T$
  - finding a path $s \longrightarrow T$ or an optimal path $s \longrightarrow^{*} T$

directed path from s to one of the nodes of T

the cheapest directed path from s to one of the nodes of T

Search

- Only special path-finding algorithms can find solution path in a huge size graph.
  - It starts from the start node that is the first current node.
  - In each step it selects a new current node among the children of the previous current node(s) in non-deterministic way.
  - It can store the subpart of discovered part of the representation graph.
  - It stops when it detects the goal node.

Search-system

Procedure *Search-system*

1. DATA := *initial value*
2. **while** ¬*termination condition*(DATA) **loop**
3. SELECT R FROM rules that can be applied
4. DATA := R(DATA)
5. **endloop**

end

Search-system

Procedure *Search-system*
1. DATA := *initial value*
2. **while** ¬*termination condition*(DATA) **loop**
3. SELECT R FROM rules that can be applied
4. DATA := R(DATA)
5. **endloop**

end

global workspace
stores the part of knowledge acquired that is useful to preserve (*initial value*, *termination condition*)

searching rules
can change the content of the workspace (*precondition*, *effect*)

control strategy
selects an appropriate rule (*general principle + heuristics*)

Examination of search-system

- soundness
  - the answer is correct if the search terminates
- completeness
  - the search guarantees the solution if there exists a solution
- optimality
  - the search gives optimal solution
- time complexity
  - number of the iterations and running time of one iteration
- space complexity
  - size of the workspace