Artificial Intelligence

Ez a prezentáció az ELTE Informatikai Karának
2014. évi Jegyzetpályázatának támogatásával készült.
This presentation is supported by Eötvös Loránd University
Faculty of Informatics.

Books

- 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 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 (AI) 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.

How can AI be recoanised?

Turing test
- natural language processing
- storing knowledge
- automatic reasoning
- learning

+ machine vision, acting

- Problems that are solved: difficult
  - Problem spaces of these problems are huge.
  - Finding solution requires intuition and creativity (heuristics) to avoid the combinatorial explosion.
- Behavior of the software: intelligent
  - It is like a human doing something.
- Methods of the software: special
  - Corresponding model (representation)
  - Effective algorithm (solving strategy) with heuristics

Does the result of a software be good enough (WAI), or its execution be like a human activity (SAI)?

Matching rules

I think you are <a> me <b>.

e.g. I think you are not listening to *me* nowadays.

Why do you think that I am <a> you <b>?

e.g. Why do you think that I am not listening to *you* nowadays*?*

Recalling rules

Why are you repeating this sentence?

„Go on” rules

What else do you want to talk about?

I see.

Projects:

chatting program (ELIZA,1966)

two-player games (chess)

Methods:

GPS, resolution (1966),

Lisp(1958)

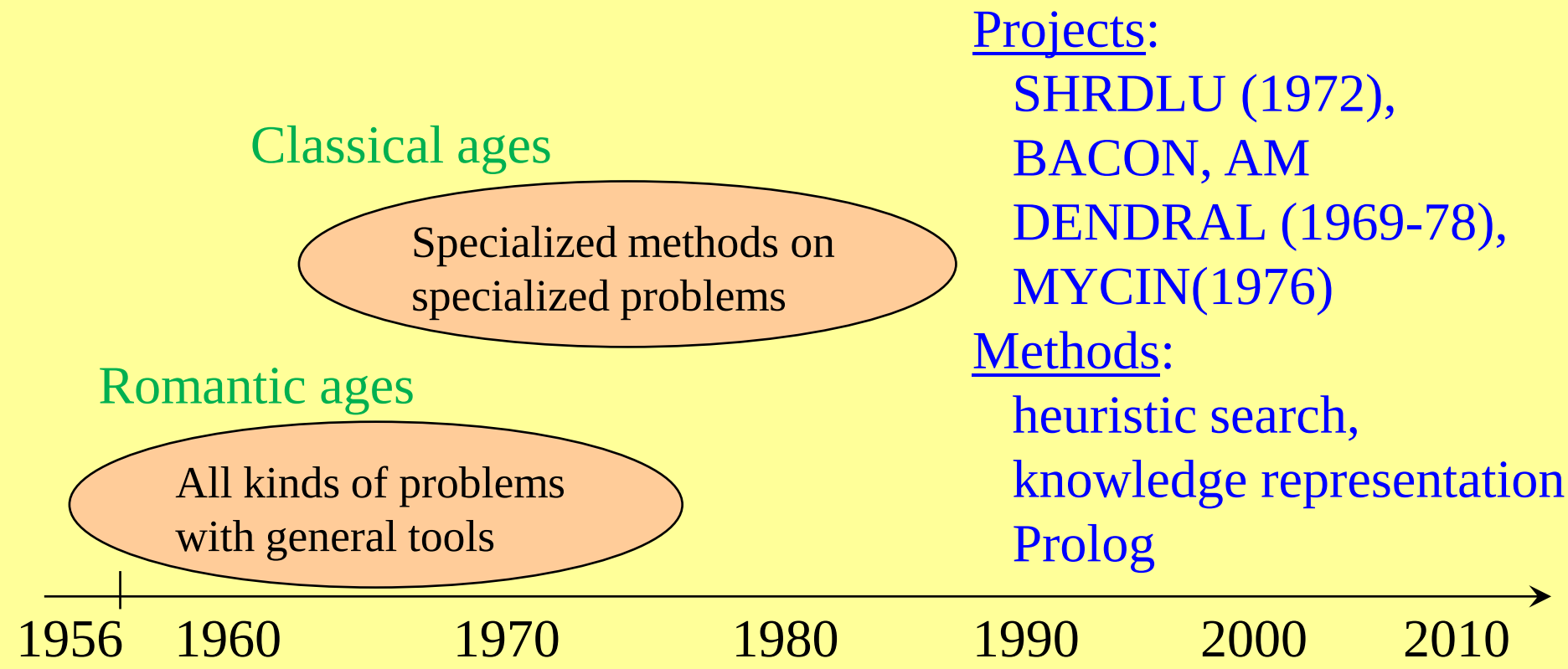
artificial neural networks

evolution algorithms

Romantic ages

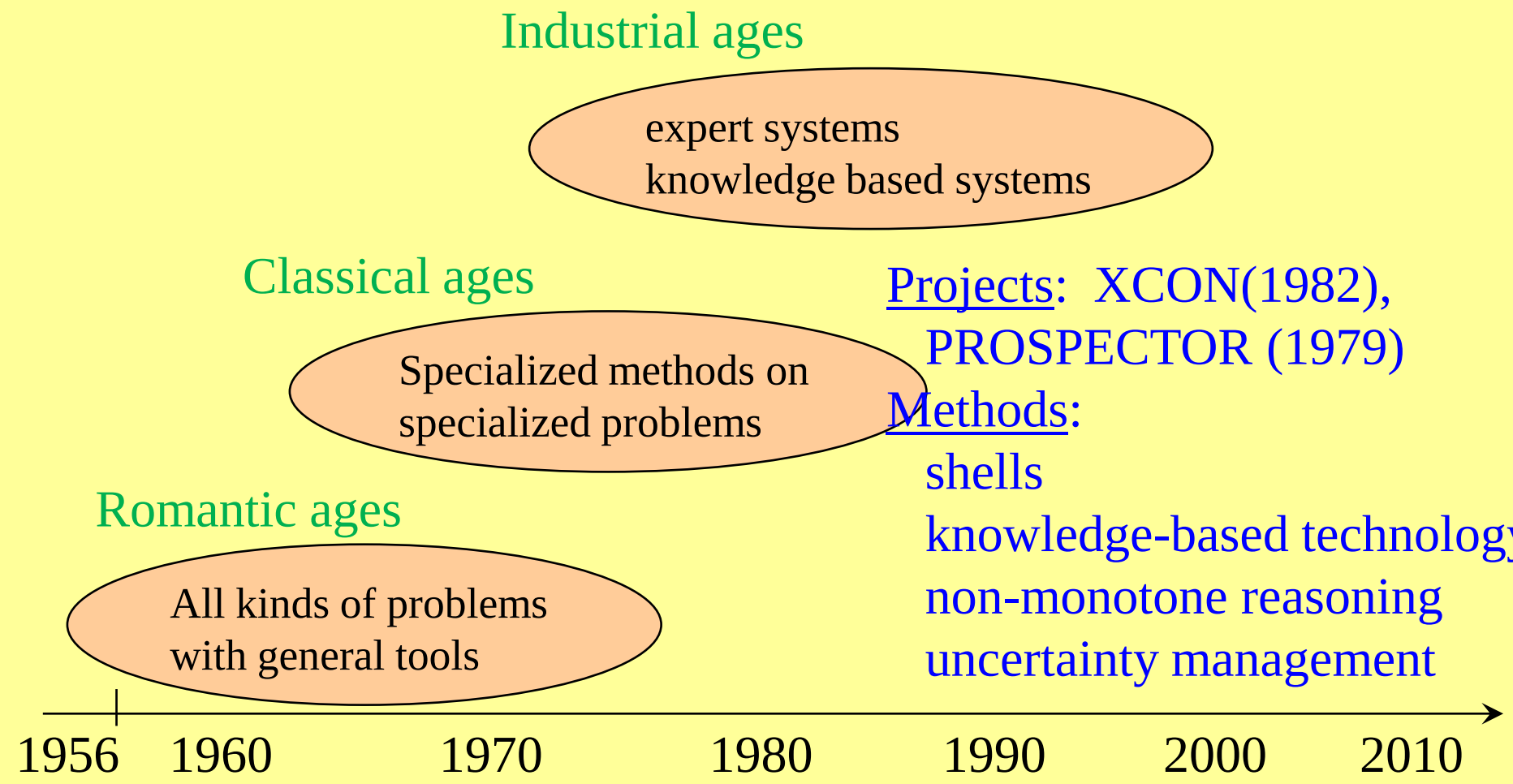
All kinds of problems with general tools

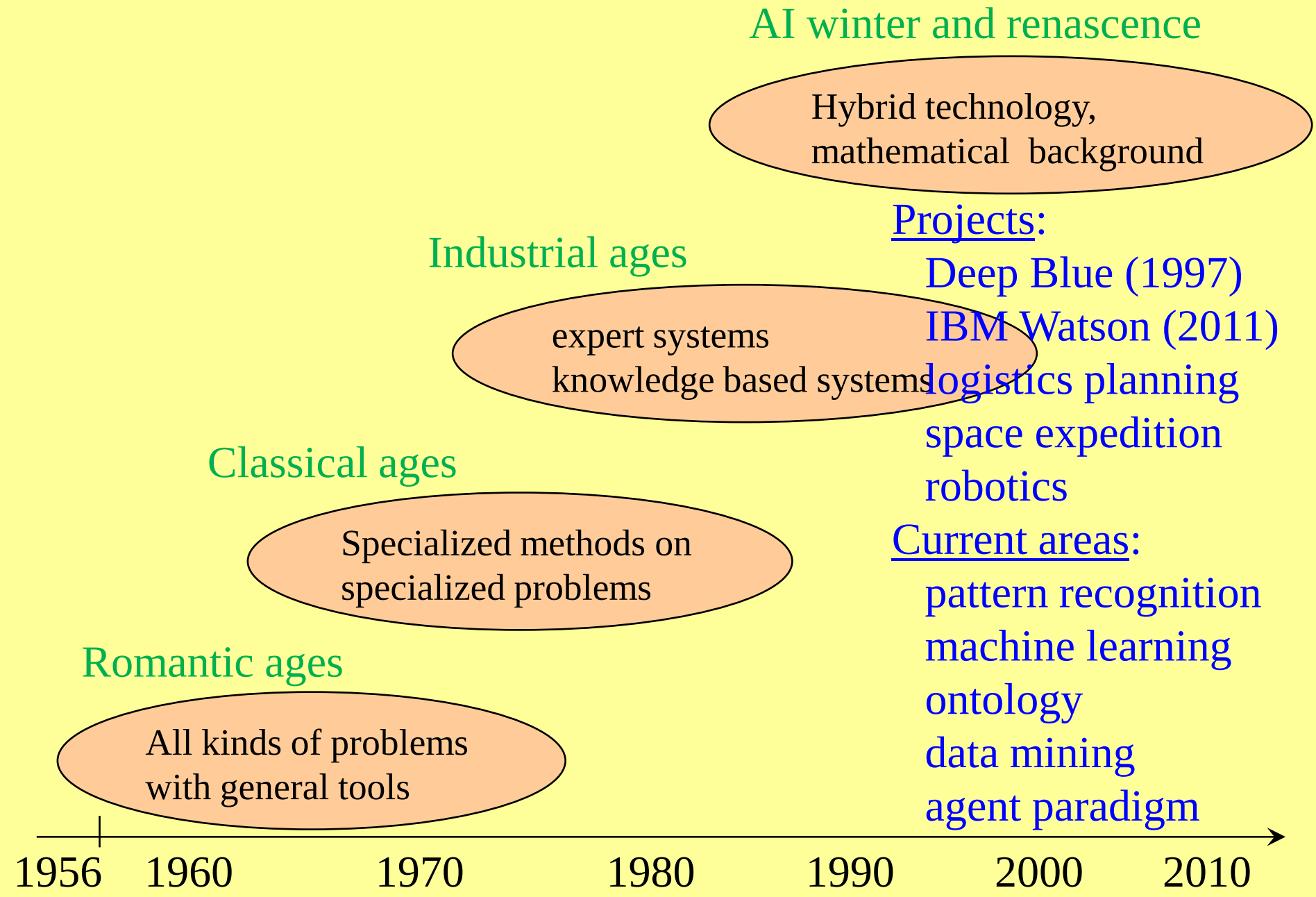
1956 1960 1970 1980 1990 2000 2010



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

Methods:
- heuristic search,
- knowledge representation
- Prolog


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
knowledge-based technology
non-monotone reasoning
uncertainty management
1956
1960
1970
1980
1990
2000
2010


AI winter and renascence
Hybrid technology,
mathematical background
Projects:
Deep Blue (1997)
IBM Watson (2011)
logistics planning
space expedition
robotics
Current areas:
pattern recognition
machine learning
ontology
data mining
agent paradigm
Industrial ages
expert systems
knowledge based systems
Classical ages
Specialized methods on
specialized problems
Romantic ages
All kinds of problems
with general tools
1956
1960
1970
1980
1990
2000
2010

2. MODELING THE PROBLEM

- ***Problem space***: contains the possible solutions / answers.
- ***Goal***: finding a correct solution
- ***Constraints***:
  - Restricting the problem space: feasible answers
  - Defining neighborhood relationships between the answers
    - Starting from an initial answer always the current answer is investigated and than one of its neighbors is selected to become the current one, and then its neighbors and so on.
    - Sorting the neighbors of the current answer.

Path-finding problems

- There are several ways to construct a model of a problem and many times this model transforms the problem to a path-finding problem.
- All path-finding problems can be mapped into an arc-weighted, directed graph where either the nodes or the paths symbolize the possible solutions of the problem space.
- In order to solve these problems either a special goal node must be found or a path starting form a start node to any goal node has to be looked for.

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	$\lvert \Gamma(n) \rvert < \infty \ (\forall n \in N)$
cost of arc	$c: A \rightarrow \mathbb{R}$
δ-property ($\delta \in \mathbb{R}^+$)	$c(n,m) \geq \delta > 0 \quad \forall (n,m) \in A$
δ-graph	arc-weighted, directed with finite outgoing arcs and δ-property

Graph notations 2.

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

  $n \overset{\alpha}{\rightarrow} m$, $n \rightarrow m$,

  $n \rightarrow M$ (one path from n to a node of $M \subseteq N$)

  $\{n \rightarrow m\}$ (all paths from n to m)

  $\{n \rightarrow M\}$ (all paths from n to nodes of $M \subseteq N$)

This is correct in δ-graphs.
If no path then 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 $c^*(n,m) := \min_{\alpha \in \{n \rightarrow m\}} c^{\alpha}(n,m)$

  $c^*(n,M) := \min_{\alpha \in \{n \rightarrow M\}} c^{\alpha}(n,m)$
- optimal path $n \rightarrow^* m := \min_c \{\alpha \mid \alpha \in \{n \rightarrow M\}\}$

Definition of graph-representation

- The graph-representation 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 of $\{s \rightarrow T\}$,
  - or an optimal path $s \rightarrow^* T$.

3. SEARCH

- Looking for a correct answer in the problem space needs a special (many times a path-finding) algorithm.
  - It starts from the initial answer.
  - After investigating an answer it steps onto one of the neighbor answers of the current one.
  - It stores the part of knowledge discovered during searching.
  - It detects the goal.

Search-system

Procedure *Search-system*

1. DATA ← *initial value (start node)*
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 discovered 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
  - its answers are correct
- completeness
  - it can solve the problem that has got solution
- optimality
  - it can find optimal solution
- time complexity
  - number of the steps and running time of one step
- space complexity
  - size of the workspace