Reasoning and Planning
Unit 1. Introduction

Pedro Cabalar

Depto. Computación
University of Corunna, SPAIN

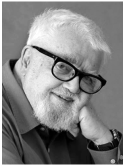
September 6, 2022

Outline

Debate

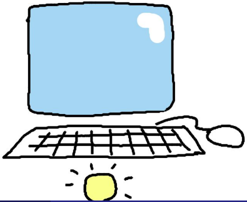
- Define Automated Reasoning
- Define Automated Planning
- Define Knowledge Representation (KR)
- Identify systems or applications where they might be used
- Pros and Cons

The origins of KR

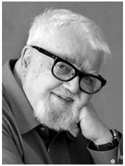


John McCarthy (1927-2011)

- Coins the term Artificial Intelligence
  "It is the science and engineering of making intelligent machines, especially intelligent computer programs"



The origins of KR

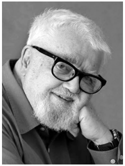


John McCarthy (1927-2011)

- *Programs with Commonsense* [1959].
  http://jmc.stanford.edu/articles/mcc59.html
  First AI reasoning system: Advice Taker.
- Keypoint: explicit representation of the domain using logical formulas. In McCarthy's words:

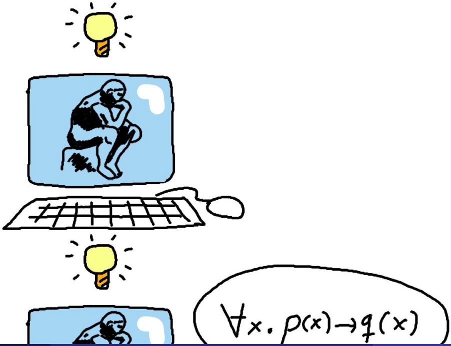
  > *"In order for a program to be capable of learning something it must first be capable of being told it"*

The origins of KR



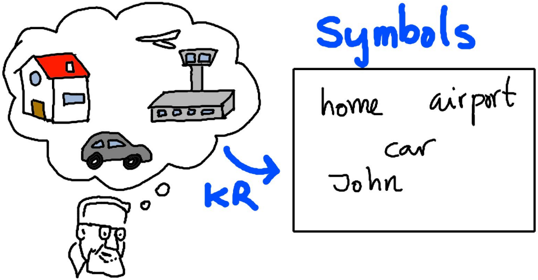
John McCarthy (1927-2011)

- Novel idea: using formal logic for commonsense reasoning



Reasoning about Actions and Change (RAC)

- Knowledge Representation (KR) plays a central role in AI.



- Automated reasoning: mechanization of thinking.
  Inference = manipulation of symbols in the machine.
- Example: Modus Ponens

$$\frac{at(John, car) \rightarrow can(go(home, airport, driving)) \qquad at(John, car)}{can(go(home, airport, driving))}$$

Reasoning about Actions and Change (RAC)

- Commonsense reasoning led to the KR area called Reasoning about Actions and Change.
- *Some philosophical problems from the standpoint of Artificial Intelligence* [McCarthy & Hayes 69]

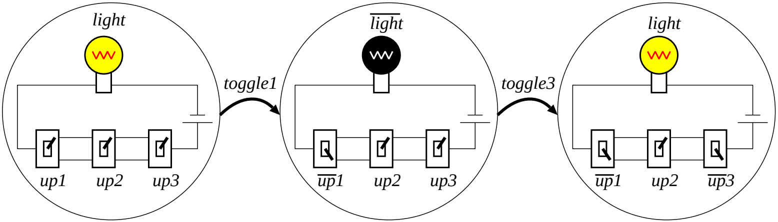
  http://jmc.stanford.edu/articles/mcchay69.html

  They introduce Situation Calculus = First Order Logic + 3 *sorts*:

  1. Fluents: system properties whose values may vary along time. These values configure the system state.
  2. Actions: possible operations that allow a state transition.
  3. Situations: terms that identify a given instant

RAC scenarios

- Typically, (discrete) dynamic systems: state transitions.
- A simple scenario: a lamp in a corridor with 3 switches.
- Fluents: $up1, up2, up3, light$ (Boolean).
- Actions: $toggle1, toggle2, toggle3$.
- State: a possible configuration of fluent values. Example: $\{\overline{up1}, up2, up3, \overline{light}\}$.
- Situation: a moment in time. We can just use $0, 1, 2, \ldots$



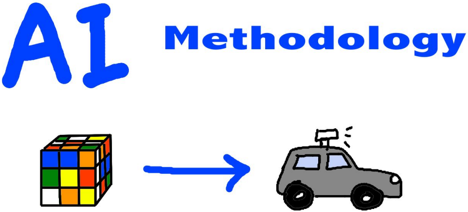
RAC goals

We want to solve some typical reasoning problems.

The most usual ones:

- Simulation: run a sequence of actions on an initial state
- Temporal explanation: fill gaps from partial observations
- Planning: obtain sequence of actions to reach some goal
- Diagnosis: explain unexpected observed results
- Verification: check system properties

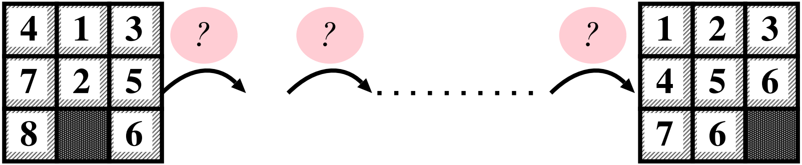
Example-based methodology



- Paraphrasing McCarthy's comment in a workshop:
  AI researchers start from examples and then try to generalize. Philosophers start from the most general case, and never use examples unless they are forced to.
- Advantage: focus on features under study using a synthetic, limited scenario (games, puzzles, etc)
- Real problems usually contain complex factors that happen to be irrelevant for the property under study.

Example-based methodology

- A classical (planning) example: the *N*-puzzle.



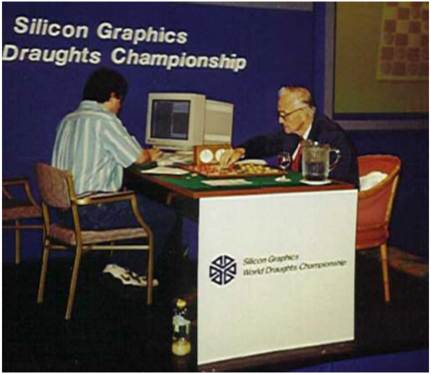
- Well known: the 8-puzzle has 181440 sates, the 15-puzzle more than 10^{13}.
- Complexity: NP-complete.

Example-based methodology

Alexander Kronrod (1921-1986)

- “Chess is the Drosophila of AI” [A. Kronrod 65]
- Games for AI can play the same role as fruit flies for Genetics.
- Competition: AI versus humans . . .

Example-based methodology



1994: Chinook [J. Schaeffer] checkers program beats world champion Marion Tinsley

Example-based methodology

1997: IBM Deep Blue beats Chess World Champion Garry Kasparov

Example-based methodology

2016: AlphaGo beats Go
World Champion Lee Sedol

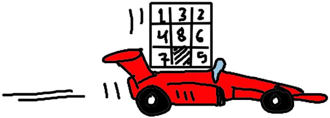
Example-based methodology

- Still, we don't have intelligent (rational) machines yet
- Warning: avoid too much focus on the toy problem. Remember we must be capable of generalizing the obtained results.
- Back to the chess example:

  "Unfortunately, the competitive and commercial aspects of making computers play chess have taken precedence over using chess as a scientific domain. It is as if the geneticists after 1910 had organized fruit fly races and concentrated their efforts on breeding fruit flies that could win these races." [McCarthy]

Example-based methodology

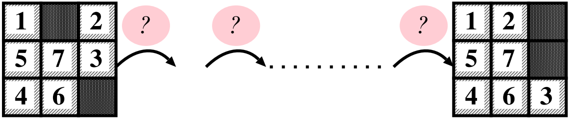
- Take the 8-puzzle example. Which is our main goal? Making a very fast solver for 8-puzzle?



- But what can we learn from that? Which is the application to other scenarios?
- We should perhaps wonder which other scenarios. Originally, AI goal was any scenario (General Problem Solver) but was too ambitious.
- It could perhaps suffice with similar scenarios. Small variations or elaborations.

Elaboration

- Example: assume we may allow now two holes.



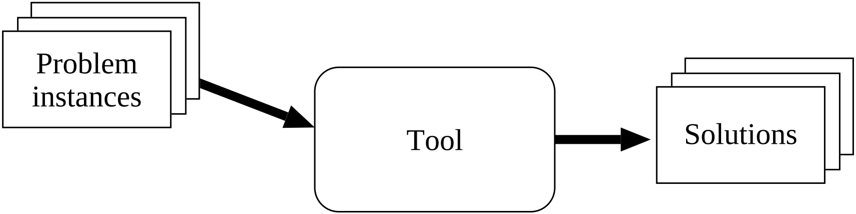
- Less steps to solve. We can even allow simultaneous movements.
- Can we easily adapt our solver to this elaboration?
- Think about an optimized heuristic search algorithm programmed in C, for instance.

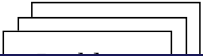
Keypoint: representation

- A much more flexible solution:
  add a description of the scenario as an input to our solver.

- In this way, variations of the scenario would mean changing the problem description . . . Knowledge Representation (KR) is crucial!

- An explicit representation of the domain rules allows Declarative Problem Solving:





Outline

Keypoint: representation

- Which are the desirable properties of a good KR?
  1. Simplicity
  2. Natural understanding: correspondence with human language
  3. Clear semantics
  4. Allows efficiently computable automated reasoning methods or at least, their complexity can be assessed
  5. Elaboration tolerance [McCarthy98]

  "A formalism is elaboration tolerant to the extent that it is convenient to modify a set of facts expressed in the formalism to take into account new phenomena or changed circumstances." [McCarthy98]

Elaboration tolerance

"Elaborating Missionaries and Cannibals Problem" [J. McCarthy]
http://jmc.stanford.edu/articles/missionaries1.html

> *3 missionaries and 3 cannibals come to a river and find a boat that holds two. If the cannibals ever outnumber the missionaries on either bank, the missionaries will be eaten. How shall they cross?*

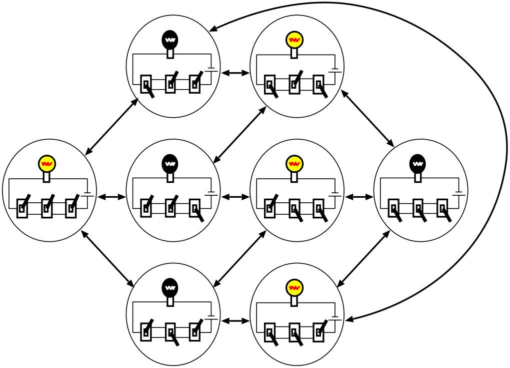
- McCarthy proposes 22 elaborations of the problem: MCP4=four on each group; MCP5=missionaries can't row; MCP10=there is an island; MCP11=Jesus Christ; MCP15=probabilities ...

Elaboration tolerance

- Students A and B encode the 8 puzzle as follows:
  - Student A:
    $at(1,1,8)\;\; at(1,2,6)\; at(1,3,hole)\;\; \ldots$
  - Student B:
    $row(8)=1\; col(8)=1$
    $row(6)=1\; col(6)=2$
    $row(hole)=1\; col(hole)=3$
- Add more holes: which solution is more elaboration tolerant? Solution A requires no changes!
- The real problem comes when our KR formalism has no way to find an elaboration tolerant solution

Elaboration tolerance

- Example of representation: an automaton is simple, and has a clear semantics . . .
- But fails in elaboration tolerance! A small change (say, adding new switches or lamps) means a complete rebuilding



Keypoint: representation

- A practical alternative: use rules to describe the local effects of each performed action.
- For each switch $X \in \{1,2,3\}$

Action	precondition	$\Rightarrow$ effect(s)
$toggle(X)$:	$up(X)$	$\Rightarrow \overline{up(X)}$
$toggle(X)$:	$\overline{up(X)}$	$\Rightarrow up(X)$
$toggle(X)$:	$light$	$\Rightarrow \overline{light}$
$toggle(X)$:	$\overline{light}$	$\Rightarrow light$

- This language is similar to STRIPS [Fikes & Nilsson 71] still used in planning systems.

Logical Knowledge Representation

- Can we just use classical logic instead?

$$toggle(X) : up(X) \quad \Rightarrow \quad \overline{up(X)}$$
$$toggle(X, T) \wedge up(X, true, T-1) \quad \rightarrow \quad up(X, false, T)$$

  where we include as new arguments, the temporal indices $T > 0$, $T - 1$ plus the fluent values *true*, *false*.

- Problem: when $toggle(1)$, what can we conclude about $up(2)$ and $up(3)$?
  They should remain unchanged! However, our logical theory provides no information (we also have models where their value change).

- We would need much more formulae

$$\begin{array}{rcl} toggle(1,T) \wedge up(2,true,T-1) & \rightarrow & up(2,true,T) \\ toggle(1,T) \wedge up(2,false,T-1) & \rightarrow & up(2,false,T) \end{array}$$

$$\begin{array}{rcl} toggle(1,T) \wedge up(3,true,T-1) & \rightarrow & up(3,true,T) \\ toggle(1,T) \wedge up(3,false,T-1) & \rightarrow & up(3,false,T) \\ & \vdots & \end{array}$$

and so on, for any fluent and value that are unrelated to $toggle(1)$.

Default reasoning

- Frame problem: adding a simple fluent or action means reformulating all these formulae! [McCarthy & Hayes 69]
- We need a kind of default reasoning.
  Inertia rule: fluents remain unchanged *by default*
- "By default" = when no evidence on the contrary is available. We must extract conclusions from absence of information.
- Unfortunately, Classical Logic is not well suited for this purpose because

  $$\Gamma \vdash \alpha \text{ implies } \Gamma \cup \Delta \vdash \alpha$$

  This is called monotonic consequence relation.
- But $\Gamma \vdash \alpha$ by default could mean that adding Δ, $\Gamma \cup \Delta \not\vdash \alpha$.
  We need Nonmonotonic Reasoning (NMR).

Default reasoning

- An example: suppose $up(2, true, 0)$ and we perform $toggle(1, 0)$. Inertia should allow us to conclude that switch 2 is unaffected:

$$\Gamma \vdash up(2, true, 1)$$

- Elaboration: we are said now that $toggle(1)$ affects $up(2)$ in the following way:

$$toggle(1, T) \wedge up(2, true, T-1) \rightarrow up(2, false, T) \tag{1}$$

We will need retract our previous conclusion

$$\Gamma \cup (1) \not\vdash up(2, true, 1)$$

Other typical representational problems

- Qualification problem: preconditions are affected by conditions that qualify an action.
- Example: when can we toggle the switch? Elaborations: switch is not broken, switch has not been stuck, we must be close enough, etc.
- The explicit addition of any imaginable “disqualification” is unfeasible. Again: by default, toggle works when nothing prevents it.

Other typical representational problems

- Elaboration: there is a light sensor that activates an alarm, if the latter is connected. The alarm causes locking the door.
- In STRIPS, this means relating indirect effects *alarm* to each possible action $toggle(X)$.

Action	precondition	$\Rightarrow$ effect(s)
$toggle(X)$:	$\overline{light}, connected$	$\Rightarrow alarm$
$toggle(X)$:	$\overline{light}, connected$	$\Rightarrow lock$

Problem: there may be other new ways to turn on a light, or to activate the alarm. We will be forced to relate *lock* to the performed actions!

Other typical representational problems

- This is called ramification problem: postconditions are affected by interactions due to indirect effects.
- *lock* is an indirect effect of toggling a switch ($toggle \mapsto light \mapsto alarm \mapsto lock$).
- We would need something like:

$$\begin{aligned} light(true, T) \wedge connected(true, T) &\rightarrow alarm(true, T) \\ alarm(true, T) &\rightarrow lock(true, T) \end{aligned}$$

Outline

1. History and Motivation
2. Knowledge Representation goals
3. Knowledge Representation areas

KR is a well-established field

Main conferences including KR, Reasoning and Planning

- IJCAI: Intl. Joint Conf. on Artificial Intelligence
- AAAI: Conf. on Artificial Intelligence
- ECAI: European Conf. on Artificial Intelligence
- KR: Intl. Conf. on Principles of Knowledge Representation and Reasoning
- ICAPS: Intl. Conf. on Automated Planning and Scheduling
- IJCAR: Intl. Joint Conf. on Automated Reasoning
- JELIA: European Conf. on Logics in Artificial Intelligence
- LPNMR: Intl. Conf. on Logic Programming and Non-Monotonic Reasoning LPNMR'13 cellebrated in Corunna!
- Workshop on Logical Formalizations of Commonsense Reasoning

KR is a well-established field

These are some of the usual topics in KR call for papers:

- Reasoning about actions and change, dynamic logic
- Epistemic reasoning (knowledge and belief)
- Belief revision and update
- Explanation finding, diagnosis, causal reasoning, abduction
- Nonmonotonic logics, default logics, conditional logics
- (Constraint) logic programming, answer set programming
- Qualitative reasoning, spatial reasoning and temporal reasoning
- Argumentation
- Computational aspects of KR, complexity
- Description logics, ontology languages, contextual reasoning
- Inconsistency, paraconsistent logics
- Preference modeling and representation
- Philosophical foundations of KR
- Uncertainty, vagueness, many-valued and fuzzy logics, relational probability