Knowledge Representation

Fabio G. Cozman

Escola Politécnica

Universidade de São Paulo

AI's current success:

more power, more data, some insights

AI's current success:
more power, more data, some insights

Knowledge Representation & Reasoning

Decision Making

Machine Learning

What is knowledge?

2020

Knowledge: true belief with explanation.

Platão, Theaetetus

What is knowledge?

Knowledge: true belief with explanation.

2020

Davis et al 1993

Platão, Theaetetus

What is knowledge?

2020

Davis et al 1993

Knowledge: true belief with explanation.

Knowledge in AI:
a very flexible thing...
including deterministic facts/rules,
and uncertain beliefs.

Platão, Theaetetus

Knowledge representation (Davis et al 1993):

- a model ("surrogate"),
- a set of ontological commitments,
- a basis for reasoning,
- a medium for computation.

What Is a Knowledge Representation?

Randall Davis, Howard Shrobe, and Peter Szolovits

■ Although knowledge representation is one of the central and, in some ways, most familiar concepts in AI, the most fundamental question about it—What is it?—has rarely been answered directly. Numerous papers have lobbied for one or another variety of representation, other papers have argued for various properties a representation should have, and still others have focused on properties that are important to the notion of representation in general.

In this article, we go back to basics to address the question directly. We believe that the answer can best be understood in terms of five important and distinctly different roles that a representation plays, each of which places different and, at times, conflicting demands on the properties a representation should have. We argue that keeping in mind all five of these roles provides a usefully broad perspective that sheds light on some long-standing disputes and can invigorate both research and practice in the field.

What is a knowledge representation? We argue that the notion can best be understood in terms of five distinct roles that it plays, each crucial to the task at hand:

First, a knowledge representation is most fundamentally a *surrogate*, a substitute for the thing itself, that is used to enable an entity to determine consequences by thinking rather than acting, that is, by reasoning about the world rather than taking action in it.

Second, it is a set of ontological commitments, that is, an answer to the question, In what terms should I think about the world?

Third, it is a fragmentary theory of intelligent reasoning expressed in terms of three components: (1) the representation's fundamental conception of intelligent reasoning, (2) the set of inferences that the representation sanctions, and (3) the set of inferences that it recommends.

Fourth, it is a medium for pragmatically efficient computation, that is, the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance that a representation provides for organizing information to facilitate making the recommended inferences.

Fifth, it is a medium of human expression, that is, a language in which we say things about the world.

Understanding the roles and acknowledging their diversity has several useful consequences. First, each role requires something slightly different from a representation; each accordingly leads to an interesting and different set of properties that we want a representation to have.

Second, we believe the roles provide a framework that is useful for characterizing a wide variety of representations. We suggest that the fundamental mind set of a representation can be captured by understanding how it views each of the roles and that doing so reveals essential similarities and differences.

Third, we believe that some previous disagreements about representation are usefully disentangled when all five roles are given appropriate consideration. We demonstrate the clarification by revisiting and dissecting the early arguments concerning frames and logic.

Finally, we believe that viewing representations in this way has consequences for both research and practice. For research, this view provides one direct answer to a question of fundamental significance in the field. It also suggests adopting a broad perspective on

Knowledge representation

- Often we can solve a problem by coding the algorithm that finds the solution.
  - To do so we must represent the aspects of the problem that are relevant.
- Sometimes it is better to store the relevant facts and rules and use them as needed.
  - The result is a *knowledge-based* system.
  - It uses a *knowledge-base*, and runs *inference* on it.

A distinctive feature of AI

- Concern about representation is key to AI.
  - In contrast, decision-making in Economics does not worry about the description of problems.

A distinctive feature of AI

- Concern about representation is key to AI.
  - In contrast, decision-making in Economics does not worry about the description of problems.

- How can...
  - a problem be described concisely and efficiently?
  - we guarantee that all features can be expressed?
  - we quantify the effect of some modeling choices?

Reasoning

- Once we have a representation for our objects of interest, we can reason about them.
- We can decide how to change them, we can extract some understanding from them.
- The operations are often referred to as "inference".

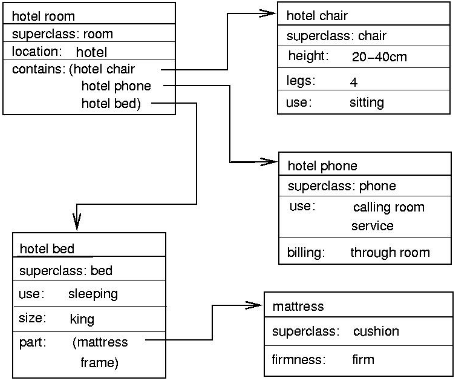
A bit of history: early efforts

- General problem solving by search: problem had to be properly represented.

- *General Problem Solver*: famous effort.
  - Separated declarative knowledge (Horn clausers) from search.
  - A *production system* with if-then rules.

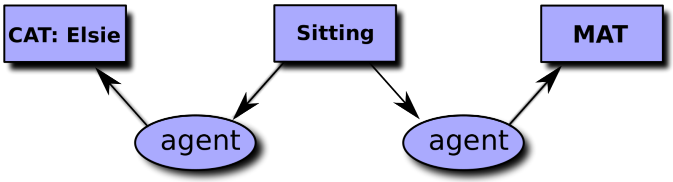
A bit of history: expert systems

- Fever of the eights.
- Idea: capture "expert knowledge" into a knowledge-base.
- Formalisms: production systems, frames.



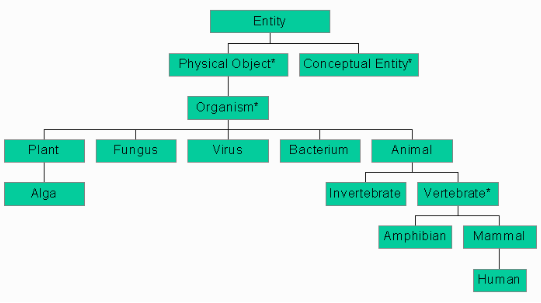
Conceptual graphs

- A graph-based interface to first-order logic.
- A reasoning model based on graphs.



Semantic networks

- Started as graphs in computational linguistics.
- Gradually were formalized using logic.



Expert system shells

- Many systems: PROSPECTOR, CADUCEUS, DENDRAL...
- The famous MYCIN rule-system (shell E-MYCIN):

```
(defrule 52
   if (site culture is blood )
      (gram organism is neg )
      (morphl organism is rod )
      (burn patient is serious)
   then 0.4
      (identity organism is pseudomonas )
)
```

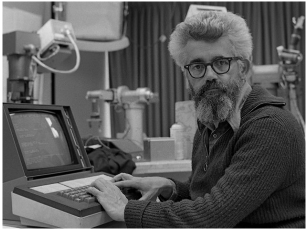
Neats and scruffies

Neats look for elegant solutions, typically with mathematical basis.

Scruffies want to build complex systems that work well.

Victory of the neats

- Gradually, most knowledge representation techniques have found a basis on formal languages.
  - Most techniques are based on logic (propositional, first-order, modal, etc).
- Serious work on complexity and expressivity.

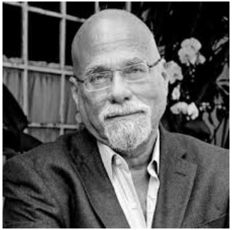


Two successful formalisms

- Description logics.
- Answer set programming.

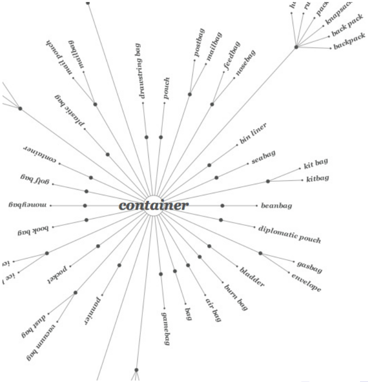
Victory of the scruffies

- Today some of the best tools are very complex and built without major consistency guarantees.
- Examples: WordNet, DBpedia, Freebase, NELL.



WordNet

- Giant multilingual lexical database.
- Free at https://wordnet.princeton.edu/.



DBpedia

- Giant database of facts extracted from Wikipedia.
- Free at https://wiki.dbpedia.org/.
- Data stored in OWL.
- Queries in SPARQL.

OWL in DBpedia

```
<owl:Class rdf:about="http://dbpedia.org/ontology/NationalAnthem">
  <rdfs:label xml:lang="en">National anthem</rdfs:label>
  <rdfs:label xml:lang="fr">Hymne national</rdfs:label>
  <rdfs:label xml:lang="nl">volkslied</rdfs:label>
  <rdfs:comment xml:lang="en">Patriotic musical composition which is the offcial national song.</rc
  <rdfs:subClassOf rdf:resource="http://dbpedia.org/ontology/MusicalWork"/>
  <prov:wasDerivedFrom rdf:resource="http://mappings.dbpedia.org/index.php/OntologyClass:NationalAr
</owl:Class>
```

SPARQL in DBpedia

Example: People who were born in Berlin before 1900.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX : <http://dbpedia.org/resource/>

SELECT ?name ?birth ?death ?person
WHERE { ?person dbo:birthPlace :Berlin .
?person dbo:birthDate ?birth .
?person foaf:name ?name . ?person dbo:deathDate ?death .
FILTER (?birth < "1900-01-01"^^xsd:date) . }
ORDER BY ?name
```

Knowledge graph: NELL

- More than 50M “beliefs” (3M with high confidence).

quaker ave is a highway [confidence 92.8]

Read the Web

Research Project at Carnegie Mellon University

Home | Project Overview | Resources & Data | Publications | People

NELL: Never-Ending Language Learning

Can computers learn to read? We think so. "Read the Web" is a research project that attempts to create a computer system that learns over time to read the web. Since January 2010, our computer system called NELL (Never-Ending Language Learner) has been running continuously, attempting to perform two tasks each day:

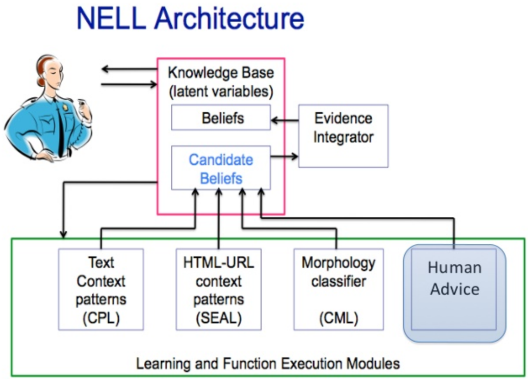
- First, it attempts to "read," or extract facts from text found in hundreds of millions of web pages (e.g., `playsInstrument(George_Harrison, guitar)`).
- Second, it attempts to improve its reading competence, so that tomorrow it can extract more facts from the web, more accurately.

Browse the Knowledge Base!

So far, NELL has accumulated over 50 million candidate beliefs by reading the web, and it is considering these at different levels of confidence. NELL has high confidence in 2,810,379 of these beliefs — these are displayed on this website. It is not perfect, but NELL is learning. You can track NELL's progress below or @cmunell on Twitter, browse and download its knowledge base, read more about our technical approach, or join the discussion group.

Recently-Learned Facts twitter Refresh

instance	iteration	date learned	confidence
jack_bond is a male	1111	06-jul-2018	90.6
quaker_ave is a highway	1111	06-jul-2018	92.8
ixquick _com is a website	1111	06-jul-2018	92.9
environmental_correspondent is a profession	1111	06-jul-2018	98.2
agco is a biotech company	1111	06-jul-2018	95.7
the sports leage ncaa uses the venue wien_stadium	1111	06-jul-2018	100.0
brewers is a sports team also known as atlanta_braves	1112	24-jul-2018	93.8
rose_garden is a stadium or event venue located in the city portland	1116	12-sep-2018	100.0
national is a company that has an office in the country ireland	1114	25-aug-2018	100.0
peppers is an agricultural product produced in _america	1116	12-sep-2018	99.2



Basic language: Propositional Logic

- Logic studies arguments and reasoning: main concern is to preserve truth.
- Propositional logic deals with *propositions* and *connectives*.

Propositions

- A proposition is a declarative statement.
  - John likes feijoada.
  - Every paulista is a corintiano.
- A proposition can be *true* or *false*.

Connectives

- Negation: $\neg A$.
- Conjunction: $A \wedge B$.
- Disjunction: $A \vee B$.
- (Material) implication: $A \to B$.
- Biconditional/equivalence: $A \leftrightarrow B$.

Syntax: well-formed formulas

- Atom is a symbol A, B, etc.
- Well-formed: $\top \quad | \quad \bot \quad | \quad A \quad | \quad \neg A$,
- Well-formed: $A \wedge B \mid A \vee B \mid A \rightarrow B \mid A \leftrightarrow B$.
- ... also, recursively...

Semantics

- A well-formed formula with n distinct propositions has 2^n distinct *interpretations*.
- An interpretation is an assignment of true/false to each proposition.

Semantics

- A well-formed formula with n distinct propositions has 2^n distinct *interpretations*.
- An interpretation is an assignment of true/false to each proposition.
- The semantics of any well-formed formula can be built out of truth tables.

Negation

A	$\neg A$
true	false
false	true

Conjunction and Disjunction

A	B	$A \wedge B$	$A \vee B$
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

Implication and Biconditional (not XOR)

A	B	$A \rightarrow B$	$A \leftrightarrow B$
false	false	true	true
false	true	true	false
true	false	false	false
true	true	true	true

Understanding material implication

- Suppose we have a triangle.
  If
  - A: one of its angles is a right angle,

  then
  - B: the square of the length of the larger side is equal to the sum of the squares of the lengths of the other sides.

Understanding material implication

- Suppose we have a triangle.
  If
  - A: one of its angles is a right angle,

  then
  - B: the square of the length of the larger side is equal to the sum of the squares of the lengths of the other sides.
- That is, $A \to B$.

In any case:

- $A \to B$
  is the *same thing* as
  $\neg A \vee B$.
- So implication is not necessary in a sense.

A well-formed formula is

- Valid (tautology) iff it is true for every interpretation.
- Satisfiable iff it is true for some interpretation.
- Unsatisfiable iff it is not satisfiable (always false).

Logical consequence

- A knowledge base KB is a set of well-formed formulas and propositions that are assumed true.
- A model of KG is an interpretation where all propositions are true and all formulas are satisfied.

Logical consequence

- A knowledge base KB is a set of well-formed formulas and propositions that are assumed true.
- A model of KG is an interpretation where all propositions are true and all formulas are satisfied.
- A is a *logical consequence* of KB, written KB $\models A$, iff A is true in every model of KB.

An example KB:

- sam_is_happy.
  ai_is_fun.
  worms_live_underground.
  night_time.
  bird_eats_apple.
  apple_is_eaten $\leftarrow$ bird_eats_apple.
  switch_1_is_up $\leftarrow$ sam_is_in_room $\land$
  night_time.

An example KB:

- sam_is_happy.
  ai_is_fun.
  worms_live_underground.
  night_time.
  bird_eats_apple.
  apple_is_eaten $\leftarrow$ bird_eats_apple.
  switch_1_is_up $\leftarrow$ sam_is_in_room $\wedge$ night_time.
- This KB has logical consequences bird_eats_apple. and apple_is_eaten. But not switch_1_is_up.

An important point:

- The implication

$$A \leftarrow B_1 \wedge B_2 \wedge \cdots \wedge B_n$$

is equivalent to

$$A \vee \neg B_1 \vee \neg B_2 \vee \cdots \vee \neg B_n.$$

- A disjunction of several possibly negated atoms is a *clause*.

The basis of current SAT-solvers: DPLL

- Go in a depth-first strategy:
  - Select a proposition A, select either A or $\neg A$.
  - Verify the consequences of the selection. If contradiction, stop the branch and backtrack.
  - Repeat this until an interpretation is found (or none can be found).
- Some of the best solvers do stochastic search as well.

DPLL works best with clauses:

- With clauses, detecting simplifications and contradictions is easy!
- If we only have definite clauses, this is easy!
- Any well-formed formula can be turned to clausal form with some effort.

Example

$$
\begin{tikzcd}[column sep=small]
 & (A \vee C) \wedge (\neg A \vee \neg B \vee C) \wedge (A \vee \neg C) \arrow[dl, "A = \top"'] \arrow[ddr, "A = \bot"] & \\
(\top \vee C) \wedge (\neg \top \vee \neg B \vee C) \wedge (\top \vee \neg C) \arrow[dd] & & \\
 & & (\bot \vee C) \wedge (\neg \bot \vee \neg B \vee C) \wedge (\bot \vee \neg C) \arrow[dd] \\
(\neg B \vee C) & & \\
 & & (C) \wedge (\neg C)
\end{tikzcd}
$$

Translating to clausal form

- Always possible with transformations:
  $\phi \leftrightarrow \psi$ to $(\phi \rightarrow \psi) \wedge (\phi \leftarrow \psi)$,
  $\phi \rightarrow \psi$ to $\neg\phi \vee \psi$,
  $\neg(\neg\phi)$ to ϕ,
  $\neg(\phi \wedge \psi)$ to $\neg\phi \vee \neg\psi$,
  $\neg(\phi \vee \psi)$ to $\neg\phi \wedge \neg\psi$,
  $\phi \vee (\psi' \wedge \psi'')$ to $(\phi \vee \psi') \wedge (\phi \vee \psi'')$.

Translating to clausal form

- Always possible with transformations:
  $\phi \leftrightarrow \psi$ to $(\phi \rightarrow \psi) \wedge (\phi \leftarrow \psi)$,
  $\phi \rightarrow \psi$ to $\neg\phi \vee \psi$,
  $\neg(\neg\phi)$ to ϕ,
  $\neg(\phi \wedge \psi)$ to $\neg\phi \vee \neg\psi$,
  $\neg(\phi \vee \psi)$ to $\neg\phi \wedge \neg\psi$,
  $\phi \vee (\psi' \wedge \psi'')$ to $(\phi \vee \psi') \wedge (\phi \vee \psi'')$.
- There are more efficient algorithms.

Example

$$
\begin{aligned}
\neg(A \leftrightarrow B) &\equiv \neg((A \rightarrow B) \wedge (A \leftarrow B)) \\
&\equiv \neg((\neg A \vee B) \wedge (\neg B \vee A)) \\
&\equiv \neg(\neg A \vee B) \vee \neg(\neg B \vee A) \\
&\equiv (A \wedge \neg B) \vee (B \wedge \neg A)
\end{aligned}
$$

Example

$$
\begin{aligned}
\neg(A \leftrightarrow B) &\equiv \neg((A \rightarrow B) \wedge (A \leftarrow B)) \\
&\equiv \neg((\neg A \vee B) \wedge (\neg B \vee A)) \\
&\equiv \neg(\neg A \vee B) \vee \neg(\neg B \vee A) \\
&\equiv (A \wedge \neg B) \vee (B \wedge \neg A) \\
&\equiv ((A \wedge \neg B) \vee B) \wedge ((A \wedge \neg B) \vee \neg A) \\
&\equiv (A \vee B) \wedge (\neg B \vee B) \wedge (A \vee \neg A) \wedge (\neg A \vee \neg B) \\
&\equiv (A \vee B) \wedge (\neg A \vee \neg B).
\end{aligned}
$$

What you should know:

- What is a knowledge-based system.
- What is a proposition, an atom, connectives.
- The syntax of propositional logic (well-formed formulas).
- The semantics of propositional logic: truth tables, interpretations, valid and satisfiable formulas, logical consequence.
- Definite clauses and conversion to clausal form.
- The DPLL algorithm (basic structure only).
- Simple translation to clausal form.

Further discussion in book by Poole and Mackworth, Chapter 5:

1. Definite clauses and knowledge-base construction (Section 5.3, Lecture 1).
2. Proofs (Section 5.3.2. Lectures 2 and 3).
3. User interaction, explanation generation and debugging (Section 5.4).
4. Abduction, nonmonotonic and default reasoning (Sections 5.6, 5.7).
5. Causal modeling (Section 5.8).

Further discussion in book by Russell and Norvig, Chapter 7:

1. Theorem proofs with propositional logic (resolution, etc).
2. Horn clauses.
3. Phase transitions in satisfiability.