

Introduction to Knowledge Representation

Dustin Smith

September 5, 2008

1 Note

Slides are at the bottom of the page! until I correct it—there’s a typo on slide 46, the conjunction and disjunction symbols are backwards (true: \wedge = and \vee = or). Latex should have caught this error :)

This section is largely based on Sowa’s *Conceptual Structures*; Sussman and Abelson’s *SICP*, Suppe’s *Introduction to Logic*, Partee et al’s *Mathematical Methods in Linguistics*, Cruse’s *Lexical Semantics* and *Thinking about Android Epistemology*.

2 The big questions

We’re concerned with the problem with getting knowledge into computers¹

Getting the knowledge We can load the knowledge into the computer, or we can build a learning system that builds its own representation from examples.

Using the knowledge Assuming we’ve got the knowledge, how do we *use* it to solve problems? (And what problems are worth solving?)

Knowledge is more than just a static set of facts; it involves the ability to use those “facts” to interact with the world.

Teddy Ruxpin needs an interface with the environment.

Information retrieval problem. Which knowledge is relevant

3 History

3.1 First combination of symbols: Ramon Lull

- 13th century Majorcan philosopher Ramon Lull (1235–1316), monk and missionary. Main work is the *Ars magna* comprising “The Tree of knowledge”. Built devices to show the *combinations* of God’s attributes. Lullian machines were discs with a common spindle, “God is Good *and* God is big *and* God is eternal. . .” missionary work in North Africa and Asia Minor. Stoned to death (as in lapidation, not Hendrix).

¹As we will see later, if we want to do a good job on either of these tasks we will need to “tighten the loop” and build systems that actively learn *in order to* solve problems.

3.2 George Boole's algebra of logic

- goal: provide a method for “correct reasoning”.
- Bertrand Russell and Rudolf Carnap showed how Frege's logic w/ set theory ,could describe the constructions of objects from the data of sensation. Russell and Whitehead tried to reduce mathematics to logic; Godel's showed these formal systems are incomplete.

4 Representing Items

4.1 Math

Math is useful because it distills the idea to its essence (simplification) and is a precise and universal way of describing structures. For example, which would you prefer?

- $x(y + z) = xy + xz$
- Any number that is multiplied by the sum of two numbers is equal to the sum of the products of it and each of the numbers in the first sum.

Set An a collection of any arbitrary elements. In math, often they are numbers.

- Sometimes the elements are never defined but left as abstractions that can be represented in different ways.
- Curly brace notation. Order doesn't matter.
- One way to list sets is just to describe their members.
 $\{5, 33, 2, 445\}$
 $\{\text{California, Colorado, Connecticut}\}$
- When its hard to enumerate, you can define some *rule* or *property*.
 $\{x \mid \text{tall}(x) \text{ and cold}(x) \}$
- The complete enumeration of the elements in a set is its **extension**. This can be tedious when sets get big, so we can resort to definitions.
 $\{x \mid x \neq 1 \}$
- But this can cause problems, *e.g.*, **Russell's paradox**:

$$U = \{x \mid x \notin x\}$$
$$U \in U?$$

- Set notations:

in	$a \in \{a, b, c\}$
not in	$d \notin \{a, b, c\}$
empty set	\emptyset
subset	$\{a, b, c\} \subset \{a, b, c\}$
superset	$\{a, b, c\} \supset \{a\}$
not subset	$\{a, b, c\} \not\subset \{a, b\}$
proper subset	$\{a, b, c\} \subsetneq \{a, b, c, d\}$
not proper subset	$\{a, b, c\} \not\subsetneq \{a, b, c\}$
subset	$\emptyset \subset \{a, b, c\}$

- Some Set theoretic equalities:
- Idempotent Laws ($X \cup X = X$)
- Commutative laws ($X \cup Y = Y \cup X$)
- Associative Laws $(X \cup Y) \cup Z = X \cup (Y \cup Z) \dots$

These make up **Boolean algebra**.

$D \subseteq A$
 $A \not\subseteq P$
 $f \in A$
 $f \in D$
 $f \notin P'$
 $D = \{x | \text{has} - \text{for} - \text{legs and howls}\}$

4.2 Describing properties of sets

- Universal quantifiers.

All, every, each, not

Logic has a long history, beginning with Aristotle (*c.* 4th BC). First began to be combined with math by Leibniz (17^{c.}).

There are many good books on the subject² .

4.3 Sentential Connectives

Sentences have truth values, and map to {True, False}. For example:

P = "it will rain today"

Q = "I will wear my umbrella hat"

²Patrick Suppes: **Introduction to Logic** [1]

4.3.1

Connective	Symbol	English
Not	$\neg P$	not P
And	$P \vee Q$	P and Q
Or	$P \wedge Q$	P or Q
Implies	$P \rightarrow Q$	If P then Q Q if P Q is a necessary condition of P P is a sufficient condition of Q
Biconditional	$P \leftrightarrow Q$	P is a sufficient and necessary condition of Q Q is a sufficient and necessary condition of P

4.3.2 Implication

IF it is raining | P \rightarrow Q
 THEN I will wear an umbrella hat |

- Only false if $P \rightarrow Q$, P and $\neg Q$:

Could be complete irrelevant:

- “It will rain” $\rightarrow 2 \times 9 = 18$

5 Inference in logic

Here is inference.

5.1 Truth tables

P	Q		P \rightarrow Q		$\neg P \vee Q$
T	T		T		T
T	F		F		F
T	T		T		T
F	F		T		T

- Rows in table grow 2^n with the number of variables.

5.2 Deduction

Applying truth-preserving transformations

Either a) check consistency—try to derive a contradiction (?) or b) deriving conclusion from premises.

5.3 History: Defining concepts

- Leibniz (1679)'s *Universal Characteristic*, concepts are combinations of primitive concepts (each a prime number) that can be combined to create products of primes (more abstract concepts).

2 = ALIVE

3 = MOSTLY-WATER

5 = IS-STRUGGLING-ARTIST

7 = WORKS-IN-E15-320G

11 = STUDIES-AI

DUSTIN = 2 x 3 x 7 x 11 = 462

CONCEPT % PROPERTY == 0 → true

CONCEPT % PROPERTY != 0 → false

5.4 Computer Science

- Abstraction

5.4.1 Object oriented programming

From Sun's Java OOP Tutorial:

“Software objects are conceptually similar to real-world objects: they too consist of state and related behavior. An object stores its state in fields (variables in some programming languages) and exposes its behavior through methods (functions in some programming languages). Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication. Hiding internal state and requiring all interaction to be performed through an object's methods is known as data encapsulation a fundamental principle of object-oriented programming.”

5.5 Linguistics and Cognitive Science

Natural languages, as opposed to programming languages, are used for communication between people. Linguists have divided these into different levels: 1. Prosody 2. Phonology 3. Morphology 4. Syntax 5. Semantics 6. Pragmatics

- not a puzzle with separate pieces, but these are all interacting parts and the distinctions may do more harm than good, particularly on the top three.
- **Category-based reasoning** involves classification (inferring an instance is a member of some kind) and then inheriting properties of that kind. Kind-referring words are known as *generics* in developmental linguistics. Typically refer to properties that are “essential, enduring, timeless” not accidental or tied to context (see Lyons 1977 via Susan Gelman: Learning words for kinds...). **Often generics are over-generalizations and do not need universal quantifiers.** (gelman 1998: “birds lay eggs” despite less than half of kinds of birds lay eggs)

(Endel Tulving (1972) classified memories into: **episodic** and **semantic**. Episodic is about individual activities and events. Semantic is dictionary-like knowledge like “All dogs have paws.”)

- Induction problems. (Prasada 2000: problem of generic knowledge) when is a feature accidental or essential? with limited number of examples.
- to learn words: children exploit inductive biases, formal morphosyntactic cues, contextual cues, and theory-based knowledge to solve the problem of learning generic language/knowledge.
- the knowledge is underspecified but adheres to their existing knowledge in appropriate ways. “birds lay eggs” -> birds are a kind of animal such that the mature female lays eggs” (Shipley 1993, 278)

5.6 Cognitive Science

Categorization and nouns People group objects into the world into categories on the basis of their “similarities”.

5.7 Lexical Semantics

- Semantic trait: mode of participation
- Criterial: Implication
- Expected: properties. “expressive paradox” (common sense level)

Combinations of words are not just constrained by the meanings but grammatical properties.

6 Learning Items

6.1 Francis Bacon’s *Novum Organum*

- 1660, he made a recipe for induction (conjunctions of features as a essential necessary and sufficient description of the phenomenon)
 1. Collect positive instances of the phenomenon, in a table
 2. Collect negative examples
 3. Organize table in varying degrees of fit, if appropriate
 4. Come up with a combination of features that are common for all *positive examples* but absent from *negative examples*.
- This is known currently as **concept learning!**

6.2 Computer Science

6.2.1 Find-S algorithm

Problems: (From Tom Mitchell’s lecture slides) - ”Can’t tell whether it has learned concept - fails with inconsistent training data - picks maximally specific h - there many be several concepts in H

6.3 Cognitive Science

6.4 Philosophy / Math

7 Representing Relations

7.1 Math

Ordered tuples.

Functions

Kinds of relations

7.2 Computer Science

7.3 Linguistics

1. Lexicalist hypothesis (Chomsky 1968)
2. Deep case hypothesis of (Fillmore 1967)

8 Representing Processes

(skipped)

9 Learning Relations

10 Learning Procedures

11 Combining Knowledge

11.1 Math

Generative grammars Hierarchy

11.2 Computer Science

Binding

- Human heap is 1 byte

12 Semantic Networks

- just another syntactic structured described by a graph grammar instead of an ordinary phrase-structure grammar.
- Partee (1971): “give abstract syntax-like structures without specifying a logic to operate on them”

12.1 Functions

Functions are a **relation** between two sets A and B , where: 1. A is the domain, and 2. Each element in A (domain) is paired with *only one* element in B (range).

13 Lastly

Note: Slides are available online.

References

[1] SUPPES, P. Introduction to logic. 330.