Knowledge Representation (KR)

- This material is covered in chapters 7– 9 and 12 (R&N, 3rd ed) (chapters 7—10 (R&N, 2nd ed)).
- Chapter 7 provides useful motivation for logic, and an introduction to some basic ideas. It also introduces propositional logic, which is a good background for first-order logic.
- What we cover here is mainly in Chapters 8 and 9. However, Chapter 8 contains some additional useful examples of how first-order knowledge bases can be constructed. Chapter 9 covers forward and backward chaining mechanisms for inference, while here we concentrate on resolution.
- Chapter 12 (3rd ed) (10 in 2nd ed.) covers some of the additional notions that have to be dealt with when using knowledge representation in AI.

Knowledge Representation
and
Reasoning

from the book of the same name

by

Ronald J. Brachman

and

Hector J. Levesque

Morgan Kaufmann Publishers, San Francisco, CA, 2004

1.

Introduction

What is knowledge?

Easier question: how do we talk about it?

We say "John knows that ..." and fill the blank with a <u>proposition</u>

– can be true / false, right / wrong

Contrast: "John fears that ..."

– same content, different attitude

Other forms of knowledge:

- know how, who, what, when, ...
- sensorimotor: typing, riding a bicycle
- affective: deep understanding

Belief: not necessarily true and/or held for appropriate reasons

and weaker yet: "John suspects that ..."

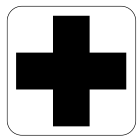
Here: no distinction

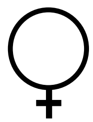
the main idea

taking the world to be one way and not another

What is representation?

Symbols standing for things in the world

 → first aid

 → women

"John" → John

"John loves Mary" → the proposition that John loves Mary

Knowledge representation:

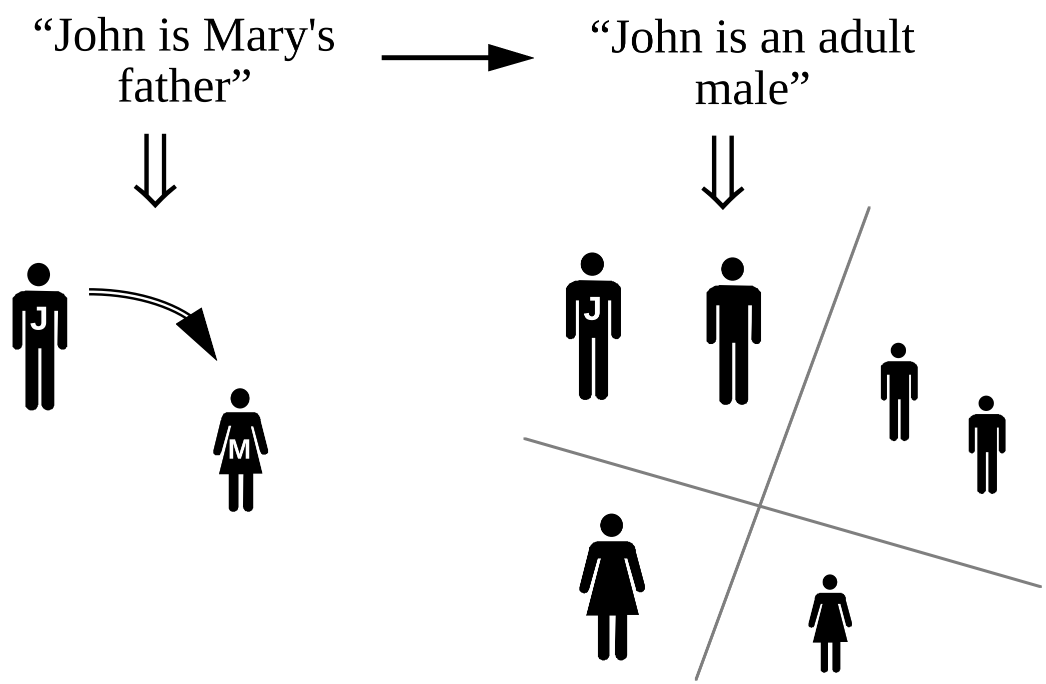
symbolic encoding of propositions believed
(by some agent)

What is reasoning?

Manipulation of symbols encoding propositions to produce representations of new propositions

Analogy: arithmetic

$$\begin{array}{ccccc} \text{“1011”} & + & \text{“10”} & \rightarrow & \text{“1101”} \\ \Downarrow & & \Downarrow & & \Downarrow \\ \text{eleven} & & \text{two} & & \text{thirteen} \end{array}$$



Why knowledge?

For sufficiently complex systems, it is sometimes useful to describe systems in terms of beliefs, goals, fears, intentions

e.g. in a game-playing program

“because it believed its queen was in danger, but wanted to still control the center of the board.”

more useful than description about actual techniques used for deciding how to move

“because evaluation procedure P using minimax returned a value of +7 for this position

= taking an intentional stance (Dan Dennett)

Is KR just a convenient way of talking about complex systems?

- sometimes anthropomorphizing is inappropriate

  e.g. thermostats

- can also be very misleading!

  fooling users into thinking a system knows more than it does

Why representation?

Note: intentional stance says nothing about what is or is not represented symbolically

e.g. in game playing, perhaps the board position is represented, but the goal of getting a knight out early is not

KR Hypothesis: (Brian Smith)

"Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantic attribution, play a formal but causal and essential role in engendering the behaviour that manifests that knowledge."

Two issues: existence of structures that

- we can interpret propositionally
- determine how the system behaves

Knowledge-based system: one designed this way!

Two examples

Example 1

```
printColour(snow) :- !, write("It's white.").
printColour(grass) :- !, write("It's green.").
printColour(sky) :- !, write("It's yellow.").
printColour(X) :- write("Beats me.").
```

Example 2

```
printColour(X) :- colour(X,Y), !,
        write("It's "), write(Y), write(".").
printColour(X) :- write("Beats me.").

colour(snow,white).
colour(sky,yellow).
colour(X,Y) :- madeof(X,Z), colour(Z,Y).
madeof(grass,vegetation).
colour(vegetation,green).
```

Both systems can be described intentionally.

Only the 2nd has a separate collection of symbolic structures à la KR Hypothesis

its <u>knowledge base</u> (or KB)

∴ a small knowledge-based system

KR and AI

Much of AI involves building systems that are knowledge-based

ability derives in part from reasoning over explicitly represented knowledge

- language understanding,
- planning,
- diagnosis,
- “expert systems”, etc.

Some, to a certain extent

game-playing, vision, etc.

Some, to a much lesser extent

speech, motor control, etc.

Current research question:

how much of intelligent behaviour is knowledge-based?

Challenges: connectionism, others

Why bother?

Why not “compile out” knowledge into specialized procedures?

- distribute KB to procedures that need it

  (as in Example 1)

- almost always achieves better performance

No need to think. *Just do it!*

- riding a bike
- driving a car
- playing chess?
- doing math?
- staying alive??

Skills (Hubert Dreyfus)

- novices think; experts *react*
- compare to current “expert systems”:

  knowledge-based !

Advantage

Knowledge-based system most suitable for *open-ended* tasks

can structurally isolate *reasons* for particular behaviour

Good for

- explanation and justification
  - “Because grass is a form of vegetation.”
- informability: debugging the KB
  - “No the sky is not yellow. It's blue.”
- extensibility: new relations
  - “Canaries are yellow.”
- extensibility: new applications
  - returning a list of all the white things
  - painting pictures

Cognitive penetrability

Hallmark of knowledge-based system:

- the ability to be *told* facts about the world and adjust our behaviour correspondingly
  - for example: read a book about canaries or rare coins

Cognitive penetrability (Zenon Pylyshyn)

- actions that are conditioned by what is currently believed
  - an example:
    - we normally leave the room if we hear a fire alarm
    - we do not leave the room on hearing a fire alarm
      if we believe that the alarm is being tested / tampered
      - can come to this belief in very many ways
    - so this action is cognitively penetrable
  - a non-example:
    - blinking reflex

Why reasoning?

Want knowledge to affect action

not do action A if sentence P is in KB

but do action A if world believed in satisfies P

Difference:

P may not be *explicitly* represented

Need to apply what is known in general
to the particulars of a given situation

Example:

"Patient x is allergic to medication m."

"Anybody allergic to medication m is also
allergic to m'."

Is it OK to prescribe m' for x ?

Usually need more than just DB-style retrieval of facts in the KB

Entailment

Sentences $P_1, P_2, \ldots, P_n$ entail sentence P iff the truth of P is implicit in the truth of $P_1, P_2, \ldots, P_n$.

If the world is such that it satisfies the P_i then it must also satisfy P.

Applies to a variety of languages (languages with truth theories)

Inference: the process of calculating entailments

- sound: get only entailments
- complete: get all entailments

Sometimes want unsound / incomplete reasoning

for reasons to be discussed later

Logic: study of entailment relations

- languages
- truth conditions
- rules of inference

Using logic

No universal language / semantics

- Why not English?
- Different tasks / worlds
- Different ways to carve up the world

No universal reasoning scheme

- Geared to language
- Sometimes want “extralogical” reasoning

Start with first-order predicate calculus (FOL)

- invented by philosopher Frege for the formalization of mathematics
- but will consider subsets / supersets and very different looking representation languages

Knowledge level

Allen Newell's analysis:

- Knowledge level: deals with language, entailment
- Symbol level: deals with representation, inference

Picking a logic has issues at each level

- Knowledge level:

  expressive adequacy,

  theoretical complexity, ...
- Symbol level:

  architectures,

  data structures,

  algorithmic complexity, ...

Next: we begin with FOL at the knowledge level

2.

The Language of First-order Logic

Declarative language

Before building system

before there can be learning, reasoning, planning, explanation ...

need to be able to express knowledge

Want a precise declarative language

- declarative: believe *P* = hold *P* to be <u>true</u>

  cannot believe *P* without some sense of what it would mean for the world to satisfy *P*

- precise: need to know exactly

  what strings of symbols count as sentences

  what it means for a sentence to be true

  (but without having to specify which ones are true)

Here: language of first-order logic

again: not the only choice

Alphabet

Logical symbols:

- Punctuation: (,), .
- Connectives: $\neg, \wedge, \vee, \forall, \exists, =$
- Variables: $x, x_1, x_2, \ldots, x', x'', \ldots, y, \ldots, z, \ldots$

  Fixed meaning and use

  like keywords in a programming language

Non-logical symbols

- Predicate symbols (like Dog)
- Function symbols (like bestFriendOf)

Note: not treating = as a predicate

  Domain-dependent meaning and use

  like identifiers in a programming language

Have arity: number of arguments

arity 0 predicates: propositional symbols

arity 0 functions: constant symbols

Assume infinite supply of every arity

Grammar

Terms

1. Every variable is a term.
2. If $t_1, t_2, ..., t_n$ are terms and f is a function of arity n, then $f(t_1, t_2, ..., t_n)$ is a term.

Atomic wffs (well-formed formula)

1. If $t_1, t_2, ..., t_n$ are terms and P is a predicate of arity n, then $P(t_1, t_2, ..., t_n)$ is an atomic wff.
2. If t_1 and t_2 are terms, then $(t_1 = t_2)$ is an atomic wff.

Wffs

1. Every atomic wff is a wff.
2. If α and β are wffs, and v is a variable, then $\neg\alpha$, $(\alpha\wedge\beta)$, $(\alpha\vee\beta)$, $\exists v.\alpha$, $\forall v.\alpha$ are wffs.

The propositional subset: no terms, no quantifiers

Atomic wffs: only predicates of 0-arity: $(p \wedge \neg(q \vee r))$

Notation

Occasionally add or omit (,), .

Use [,] and {,} also.

Abbreviations:

$(\alpha \supset \beta)$ for $(\neg\alpha \vee \beta)$

safer to read as disjunction than as "if ... then ..."

$(\alpha \equiv \beta)$ for $((\alpha\supset\beta) \wedge (\beta\supset\alpha))$

Non-logical symbols:

- Predicates: mixed case capitalized

  Person, Happy, OlderThan

- Functions (and constants): mixed case uncapitalized

  fatherOf, successor,
  johnSmith

Variable scope

Like variables in programming languages, the variables in FOL have a scope determined by the quantifiers

Lexical scope for variables

$$P(x) \wedge \exists x[P(x) \vee Q(x)]$$

free bound occurrences of variables

A sentence: wff with no free variables (closed)

Substitution:

$\alpha[v/t]$ means α with all free occurrences of the v replaced by term t

Note: written α^v_t elsewhere (and in book)

Also: $\alpha[t_1,\ldots,t_n]$ means $\alpha[v_1/t_1,\ldots,v_n/t_n]$

Semantics

How to interpret sentences?

- what do sentences claim about the world?
- what does believing one amount to?

Without answers, cannot use sentences to represent knowledge

Problem:

cannot fully specify interpretation of sentences because non-logical symbols reach outside the language

So:

make clear dependence of interpretation on non-logical symbols

Logical interpretation:

specification of how to understand predicate and function symbols

Can be complex!

DemocraticCountry, IsABetterJudgeOfCharacterThan, favouriteIceCreamFlavourOf, puddleOfWater27

The simple case

There are objects.

some satisfy predicate P; some do not

Each interpretation settles extension of P.

borderline cases ruled in separate interpretations

Each interpretation assigns to function f a mapping from objects to objects.

functions always well-defined and single-valued

The FOL assumption:

this is all you need to know about the non-logical symbols to understand which sentences of FOL are true or false

In other words, given a specification of

» what objects there are

» which of them satisfy P

» what mapping is denoted by f

it will be possible to say which sentences of FOL are true

Interpretations

Two parts: $\mathfrak{I} = \langle D, I \rangle$

D is the domain of discourse

can be *any* non-empty set

not just formal / mathematical objects

e.g. people, tables, numbers, sentences, unicorns, chunks of peanut butter, situations, the universe

I is an interpretation mapping

If P is a predicate symbol of arity n,

$I[P] \subseteq D \times D \times \ldots \times D$

an n-ary relation over D

If f is a function symbol of arity n,

$I[f] \in [D \times D \times \ldots \times D \rightarrow D]$

an n-ary function over D

for propositional symbols,

$I[p] = \{\}$ or $I[p] = \{\langle\rangle\}$

for constants, $I[c] \in D$

In propositional case, convenient to assume

$\mathfrak{I} = I \in$ [*prop. symbols* $\rightarrow$ {true, false}]

Denotation

In terms of interpretation $\mathfrak{I}$, terms will denote elements of the domain D.

will write element as $\|t\|_{\mathfrak{I}}$

For terms with variables, the denotation depends on the values of variables

will write as $\|t\|_{\mathfrak{I},\mu}$

where $\mu \in [\textit{Variables} \rightarrow D]$,
called a <u>variable</u> <u>assignment</u>

Rules of interpretation:

1. $\|v\|_{\mathfrak{I},\mu} = \mu(v)$.
2. $\| f(t_1, t_2, \ldots, t_n) \|_{\mathfrak{I},\mu} = H(d_1, d_2, \ldots, d_n)$

   where $H = I[f]$

   and $d_i = \|t_i\|_{\mathfrak{I},\mu}$, recursively

Satisfaction

In terms of an interpretation $\mathfrak{I}$, sentences of FOL will be either true or false.

Formulas with free variables will be true for some values of the free variables and false for others.

Notation:

will write as $\mathfrak{I},\mu \models \alpha$ "α is satisfied by $\mathfrak{I}$ and μ"

where $\mu \in [\mathit{Variables} \rightarrow D]$, as before

or $\mathfrak{I} \models \alpha$, when α is a sentence

"α is true under interpretation $\mathfrak{I}$"

or $\mathfrak{I} \models S$, when S is a set of sentences

"the elements of S are true under interpretation $\mathfrak{I}$"

And now the definition...

Rules of interpretation

1. $\Im,\mu \models P(t_1, t_2, ..., t_n)$ iff $\langle d_1, d_2, ..., d_n \rangle \in R$
   where $R = I[P]$
   and $d_i = \| t_i \|_{\Im,\mu}$, as on denotation slide
2. $\Im,\mu \models (t_1 = t_2)$ iff $\| t_1 \|_{\Im,\mu}$ is the same as $\| t_2 \|_{\Im,\mu}$
3. $\Im,\mu \models \neg\alpha$ iff $\Im,\mu \not\models \alpha$
4. $\Im,\mu \models (\alpha\wedge\beta)$ iff $\Im,\mu \models \alpha$ and $\Im,\mu \models \beta$
5. $\Im,\mu \models (\alpha\vee\beta)$ iff $\Im,\mu \models \alpha$ or $\Im,\mu \models \beta$
6. $\Im,\mu \models \exists v\alpha$ iff for some $d \in D$, $\Im,\mu\{d;v\} \models \alpha$
7. $\Im,\mu \models \forall v\alpha$ iff for all $d \in D$, $\Im,\mu\{d;v\} \models \alpha$
   where $\mu\{d;v\}$ is just like μ, except that $\mu(v)=d$.

For propositional subset:

$\Im \models p$ iff $I[p] \neq \{\}$ and the rest as above

Entailment defined

Semantic rules of interpretation tell us how to understand all wffs in terms of specification for non-logical symbols.

But some connections among sentences are independent of the non-logical symbols involved.

e.g. If α is true under $\Im$, then so is $\neg(\beta \wedge \neg\alpha)$,
no matter what $\Im$ is, why α is true, what β is, ...

$S \models \alpha$ iff for every $\Im$, if $\Im \models S$ then $\Im \models \alpha$.

Say that S <u>entails</u> α or α is a <u>logical consequence</u> of S:

In other words: for no $\Im$, $\Im \models S \cup \{\neg\alpha\}$. $S \cup \{\neg\alpha\}$ is <u>unsatisfiable</u>

Special case when S is empty: $\models \alpha$ iff for every $\Im$, $\Im \models \alpha$.

Say that α is <u>valid</u>.

Note: $\{\alpha_1, \alpha_2, ..., \alpha_n\} \models \alpha$ iff $\models (\alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n) \supset \alpha$

finite entailment reduces to validity

Why do we care?

We do not have access to user-intended interpretation of non-logical symbols

But, with <u>entailment</u>, we know that if *S* is true in the intended interpretation, then so is α.

If the user's view has the world satisfying *S,* then it must also satisfy α.

There may be other sentences true also; but α is logically guaranteed.

So what about ordinary reasoning?

Dog(fido) ➠ Mammal(fido) ??

Not entailment!

There are logical interpretations where $I[\text{Dog}] \not\subset I[\text{Mammal}]$

Key idea of KR:

include such connections <u>explicitly</u> in *S*
$\forall x[\text{Dog}(x) \supset \text{Mammal}(x)]$
Get: $S \cup \{\text{Dog(fido)}\} \models \text{Mammal(fido)}$

the rest is just details...

Knowledge bases

KB is set of sentences

explicit statement of sentences believed (including any assumed connections among non-logical symbols)

KB $\models \alpha$ α is a further consequence of what is believed

- explicit knowledge: KB
- implicit knowledge: $\{ \alpha \mid \text{KB} \models \alpha \}$

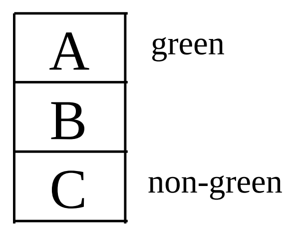
Often non trivial: explicit ➠ implicit

Example:

Three blocks stacked.

Top one is green.

Bottom one is not green.



Is there a green block directly on top of a non-green block?

A formalization

$S = \{\mathrm{On(a,b)},\ \mathrm{On(b,c)},\ \mathrm{Green(a)},\ \neg\mathrm{Green(c)}\}$

all that is required

$\alpha = \exists x \exists y[\mathrm{Green}(x) \wedge \neg\mathrm{Green}(y) \wedge \mathrm{On}(x,y)]$

Claim: $S \models \alpha$

Proof:

Let $\mathfrak{I}$ be any interpretation such that $\mathfrak{I} \models S$.

Case 1: $\mathfrak{I} \models \mathrm{Green(b)}$.

$\therefore\ \mathfrak{I} \models \mathrm{Green(b)} \wedge \neg\mathrm{Green(c)} \wedge \mathrm{On(b,c)}$.

$\therefore\ \mathfrak{I} \models \alpha$

Case 2: $\mathfrak{I} \not\models \mathrm{Green(b)}$.

$\therefore\ \mathfrak{I} \models \neg\mathrm{Green(b)}$

$\therefore\ \mathfrak{I} \models \mathrm{Green(a)} \wedge \neg\mathrm{Green(b)} \wedge \mathrm{On(a,b)}$.

$\therefore\ \mathfrak{I} \models \alpha$

Either way, for any $\mathfrak{I}$, if $\mathfrak{I} \models S$ then $\mathfrak{I} \models \alpha$.

So $S \models \alpha$. QED

Knowledge-based system

Start with (large) KB representing what is explicitly known

e.g. what the system has been told or has learned

Want to influence behaviour based on what is <u>implicit</u> in the KB (or as close as possible)

Requires reasoning

<u>deductive inference</u>:

process of calculating entailments of KB

i.e given KB and any α, determine if KB $\models \alpha$

Process is <u>sound</u> if whenever it produces α, then KB $\models \alpha$

does not allow for plausible assumptions that may be true in the intended interpretation

Process is <u>complete</u> if whenever KB $\models \alpha$, it produces α

does not allow for process to miss some α or be unable to determine the status of α