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: $, \wedge, \vee, \forall, \exists, =$
- 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)

  Domain-dependent meaning and use

  like identifiers in a programming language

Note: not treating = as a predicate

Have <u>arity</u>: 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 α, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $\exists v.\alpha$, $\forall v.\alpha$ are wffs.

The propositional subset: no terms, no quantifiers

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

Notation

Occasionally add or omit (,), .

Use [,] and {,} also.

Abbreviations:

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

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

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,

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

an n-ary relation over D

If f is a function symbol of arity n,

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

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 $\|t\|_{\mathfrak{I},}$

where $\in$ [*Variables* $\rightarrow D$],
called a <u>variable</u> <u>assignment</u>

Rules of interpretation:

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

   where $H = I[f]$

   and $d_i = \|t_i\|_{\mathfrak{I},}$, 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 $\mathfrak{I}, \models \alpha$ "α is satisfied by $\mathfrak{I}$ and "

where $\in$ [*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. $\Im$, $\models P(t_1, t_2, ..., t_n)$ iff $\langle d_1, d_2, ..., d_n\rangle \in R$

   where $R = I[P]$

   and $d_i = \| t_i \|_{\Im,}$, as on denotation slide

2. $\Im$, $\models (t_1 = t_2)$ iff $\| t_1 \|_{\Im,}$ is the same as $\| t_2 \|_{\Im,}$

3. $\Im$, $\models \alpha$ iff $\Im$, $\not\models \alpha$

4. $\Im$, $\models (\alpha \wedge \beta)$ iff $\Im$, $\models \alpha$ and $\Im$, $\models \beta$

5. $\Im$, $\models (\alpha \vee \beta)$ iff $\Im$, $\models \alpha$ or $\Im$, $\models \beta$

6. $\Im$, $\models \exists v\alpha$ iff for some $d \in D$, $\Im$, $\{d;v\} \models \alpha$

7. $\Im$, $\models \forall v\alpha$ iff for all $d \in D$, $\Im$, $\{d;v\} \models \alpha$

   where $\{d;v\}$ is just like , except that $(v)=d$.

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 $(\beta\wedge\ \ \alpha)$,
no matter what $\Im$ is, why α is true, what β is, ...

S |= α iff for every $\Im$, if $\Im$ |= S then $\Im$ |= α.

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

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

Special case when S is empty: |= α iff for every $\Im$, $\Im$ |= α.

Say that α is <u>valid</u>.

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

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 $I[\mathrm{Dog}] \not\subset I[\mathrm{Mammal}]$

Key idea of KR:

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

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: $\{ \alpha \mid \text{KB} \models \alpha \}$

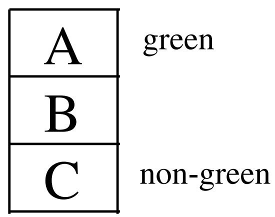
Often non trivial: explicit ➠ implicit

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 = {On(a,b), On(b,c), Green(a), Green(c)}

all that is required

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

Claim: $S \models \alpha$

Proof:

Let $\Im$ be any interpretation such that $\Im \models S$.

Case 1: $\Im \models$ Green(b).

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

$\therefore\ \Im \models \alpha$

Case 2: $\Im \not\models$ Green(b).

$\therefore\ \Im \models \ \mathrm{Green(b)}$

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

$\therefore\ \Im \models \alpha$

Either way, for any $\Im$, if $\Im \models S$ then $\Im \models \alpha$.

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