# 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)

**Note**: not treating = as a predicate

- Function symbols (like bestFriendOf)

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 $t_1, t_2, \ldots, t_n$ are terms and $f$ is a function of arity $n$, then $f(t_1, t_2, \ldots, t_n)$ is a term.

## Atomic wffs (well-formed formula)

1. If $t_1, t_2, \ldots, t_n$ are terms and $P$ is a predicate of arity $n$, then $P(t_1, t_2, \ldots, t_n)$ 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 $\alpha$ and $\beta$ 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

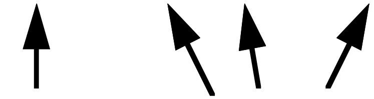
  fatherOf, successor,
  johnSmith

# Variable scope

Like variables in programming languages, the variables in FOL have a <u>scope</u> determined by the quantifiers

Lexical scope for variables

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



free bound occurrences of variables

A <u>sentence</u>: wff with no free variables (closed)

Substitution:

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

Note: written $\alpha_t^v$ 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 $n$,

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

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

Rules of interpretation:

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

   where $H = I[f]$

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

# Satisfaction

In terms of an interpretation $\Im$, 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 $\Im, \mu \models \alpha$ "$\alpha$ is satisfied by $\Im$ and $\mu$"

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

or $\Im \models \alpha$, when $\alpha$ is a sentence

"$\alpha$ is true under interpretation $\Im$"

or $\Im \models S$, when $S$ is a set of sentences

"the elements of $S$ are true under interpretation $\Im$"

And now the definition...

# Rules of interpretation

1. $\mathfrak{I},\mu \models P(t_1, t_2, ..., t_n)$ iff $\langle d_1, d_2, ..., d_n\rangle \in R$

   where $R = I[P]$

   and $d_i = \| t_i \|_{\mathfrak{I},\mu}$, as on denotation slide

2. $\mathfrak{I},\mu \models (t_1 = t_2)$ iff $\| t_1 \|_{\mathfrak{I},\mu}$ is the same as $\| t_2 \|_{\mathfrak{I},\mu}$
3. $\mathfrak{I},\mu \models \neg\alpha$ iff $\mathfrak{I},\mu \not\models \alpha$
4. $\mathfrak{I},\mu \models (\alpha\wedge\beta)$ iff $\mathfrak{I},\mu \models \alpha$ and $\mathfrak{I},\mu \models \beta$
5. $\mathfrak{I},\mu \models (\alpha\vee\beta)$ iff $\mathfrak{I},\mu \models \alpha$ or $\mathfrak{I},\mu \models \beta$
6. $\mathfrak{I},\mu \models \exists v\alpha$ iff for some $d \in D$, $\mathfrak{I},\mu\{d;v\} \models \alpha$
7. $\mathfrak{I},\mu \models \forall v\alpha$ iff for all $d \in D$, $\mathfrak{I},\mu\{d;v\} \models \alpha$

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

For propositional subset:

$\mathfrak{I} \models p$ 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 $\alpha$ is true under $\mathfrak{I}$, then so is $\neg(\beta \wedge \neg\alpha)$,
no matter what $\mathfrak{I}$ is, why $\alpha$ is true, what $\beta$ is, ...

$S \models \alpha$ iff for every $\mathfrak{I}$, if $\mathfrak{I} \models S$ then $\mathfrak{I} \models \alpha$.

Say that $S$ <u>entails</u> $\alpha$ or $\alpha$ is a <u>logical consequence</u> of $S$:

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

Special case when $S$ is empty: $\models \alpha$ iff for every $\mathfrak{I}$, $\mathfrak{I} \models \alpha$.

Say that $\alpha$ 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 $\alpha$.

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

There may be other sentences true also; but $\alpha$ 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 |= $\alpha$ $\alpha$ is a further consequence of what is believed

- explicit knowledge: KB
- implicit knowledge: $\{ \alpha \mid$ KB |= $\alpha \}$

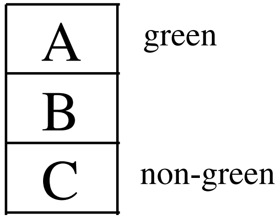
Often non trivial: explicit ➠ implicit

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

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

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 $\alpha$, determine if KB $\models \alpha$

Process is <u>sound</u> if whenever it produces $\alpha$, 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 $\alpha$

does not allow for process to miss some $\alpha$ or be unable to determine the status of $\alpha$