COMP44 Knowledge Representation and Reasoning

First-Order Logic 2

Maurice Pagnucco

School of Computer Science and Engineering

COMP4418, Week 2

First-Order Logic

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 true
  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)

What does it mean to have a language?

- syntax
- **semantics**
- pragmatics

Here: language of first-order logic
    again: not the only choice

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!

IsABetterJudgeOfCharacterThan,
favouriteIceCreamFlavourOf,
puddleOfWater27

Simple Case

There are objects
    some satisfy predicate *P*: some do not
Each interpretation settles ext..... of *P*
    borderline cases ruled in ..... interpretations
Each interpretation assigns to ..... *f* a mapping from objects to objects
    functions always well-defined and single-valued
Main 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 and which are not

Interpretations

Two parts: $I = \langle D, \Phi \rangle$

D is the domain of discourse

- can be any set
- not just formal / mathematical objects
- e.g. people, tables, numbers, sentences, chunks of peanut butter, situations, the universe

Φ is an *interpretation* mapping

- If P is a predicate symbol of arity n, $\Phi(P) \subseteq [D \times D \times \ldots \times D]$
  an n-ary relation over D
  Can view interpretation of predicates in terms of characteristic function
  $\Phi(P) \in [D \times D \times \ldots \times D \to \{0, 1\}]$
- If f is a function symbol of arity n,
  $\Phi(f) \in [D \times D \times \ldots \times D \to D]$
  an n-ary function over D
- For constants, $\Phi(c) \in D$

UNSW

Denotation

In terms of interpretation *I*, terms will denote elements of *D*.

    will write element as *I*

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

    will write as $I, \mu||t||$

    where $\mu \in [$*Variables* $\rightarrow D]$ called a variable assignment

Rules of interpretation:

1. $I, \mu||v|| = \mu(v)$

2. $I, \mu||f(t_1, t_2, \ldots, t_n)|| = H(d_1, d_2, \ldots, d_n)$

    where $H = \Phi(f)$
    and d_i recursively

Satisfaction

In terms of *I*, wffs will be true for some 程序代写代做 CS编程辅导 and false for others

will write as $I, \mu \models \alpha$ "α is satisfied by I and μ" where $\mu \in [Variables \rightarrow D]$, as before

or $I \models \alpha$, when α is a senter

or $I \models S$, when S s a set of (all sentences in S are true in I).

Rules of interpretation:

1. $I, \mu \models P(t_1, t_2, \ldots, t_n)$ iff $\langle d_1, d_2, \ldots, d_n \rangle$ where $R = \Phi(P)$ and $d_i = I, \mu||t_i||$, as on previous slide

2. $I, \mu \models (t_1 = t_2)$ iff $I, \mu||t_1||$ is the same as $I, \mu||t_2||$

3. $I, \mu \models \neg\alpha$ iff $I, \mu \not\models \alpha$

4. $I, \mu \models (\alpha \land \beta)$ iff $I, \mu \models \alpha$ and $I, \mu \models \beta$

5. $I, \mu \models (\alpha \lor \beta)$ iff $I, \mu \models \alpha$ or $I, \mu \models \beta$

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

7. $I, \mu \models \forall v, \alpha$ iff for all $d \in D$, $I, \mu\{d; v\} \models \alpha$ where $\mu\{d; v\}$ is just like μ, except on v, where $\mu(v) = d$.

For propositional subset: $I \models p$ iff $\Phi(p) = 1$ and the rest as above

UNSW

Logical Consequence

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

But some connections among wffs are independent of non-logical symbols involved.   e.g. If α is true under I then so is $\neg(\beta \wedge \neg\alpha)$ no matter what I is, why α is true, what β is,...   a function of logical symbols only

S *entails* α or α is a *logical consequence* of S:

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

In other words: for no I, $I \models S \cup \{\neg\alpha\}$.

Say that $S \cup \{\neg\alpha\}$ is *unsatisfiable*

Special case: S is empty $\models \alpha$ iff for every $I, I \models \alpha$. Say α is *valid*.

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

finite entailment reduces to validity

Why do we care?

We do not have access to user intended interpretation of non-logical symbols
But, with *entailment*, we know that if *S* is true in the intended interpretation, then
so is α.

- If the user's view has the ... satisfying *S*, then it must also satisfy α
- There may be other sentences true also; but α is logically guaranteed.

So what about:

Dog(fido) ⇒ *Mammal(fido)??*
Not entailment!
There are logical interpretations where $\Phi(Dog) \not\subset \Phi(Mammal)$
Key idea of KR:
    include such connections explicitly in S
    $\forall x[Dog(x) \rightarrow Mammal(x)]$
    Get: $S \cup \{Dog(fido)\} \models Mammal(fido)$
The rest is just the details...

Knowledge Bases

KB is set of sentences

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

$KB \models \alpha$

- α is a further consequence of what is believed
- explicit knowledge: KB
- implicit knowledge: $\{\alpha | KB \models \alpha\}$

Often non trivial: explicit $\rightarrow$ implicit

A	green
B	
C	non-green

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 Formalisation

$S = \{On(a, b), On(b, c), Green(a), \neg Green(c)\}$
  all that is required
$\alpha = \exists x \exists y[Green(x) \wedge \neg Green(y) \wedge On(x, y)]$
Claim: $S \models \alpha$
Proof:
Let I be any interpretation such that $I \models S$.
Case 1: $I \models Green(b)$.
  $\therefore I \models Green(b) \wedge \neg Green(c) \wedge On(b, c)$.
  $\therefore I \models \alpha$
Case 2: $I \not\models Green(b)$.
  $\therefore I \models \neg Green(b)$
  $\therefore I \models Green(a) \wedge \neg Green(b) \wedge On(a, b)$.
  $\therefore I \models \alpha$
Either way, for any I, if $I \models S$ then $I \models \alpha$
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
Want to influence behaviour by using what is *implicit* in the KB (or as close as possible)
Requires reasoning

- *deductive inference:*
    process of calculating entailments of KB
    i.e given KB and any α determine if $KB \models \alpha$

- Process is *sound* if whenever it produces α then $KB \models \alpha$
    does not allow for plausible assumptions that may be true in intended interpretation

- Process is *complete* 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