First-Order Logic

1

Truth in first-order logic

- Sentences are true with respect to a model and an interpretation
- Model contains objects (domain elements) and relations among them
- Interpretation specifies referents for

  constant symbols → objects

  predicate symbols → relations

  function symbols → functional relations
- An atomic sentence *predicate(term*$_1$*,...,term*$_n$*)* is true iff the objects referred to by *term*$_1$*,...,term*$_n$ are in the relation referred to by *predicate*

2

Review

- Everyone at UH is smart
  $\forall$x At(x,UH) $\wedge$ Smart(x)

Everyone is at UH and everyone is smart

$\forall$x At(x,UH) $\Rightarrow$ Smart(x)

3

A common mistake to avoid

- Typically, $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$:

  $\forall$x At(x,UH) $\wedge$ Smart(x)
  means “Everyone is at UH and everyone is smart”

4

Existential quantification

- $\exists$<*variables*> <*sentence*>

Someone at UH is smart:
$\exists x$ At(x,UH) $\Rightarrow$ Smart(x)
(Also true even if the person is not at UH)

$\exists x$ At(x,UH) $\wedge$ Smart(x)

- $\exists x$ P is true in a model m iff P is true with x being some possible object in the model
- Roughly speaking, equivalent to the disjunction of instantiations of P
- At(KingJohn,NUS) $\wedge$ Smart(KingJohn)
  $\vee$At(Richard,NUS) $\wedge$ Smart(Richard)
  $\vee$At(NUS,NUS) $\wedge$ Smart(NUS)
  $\vee$ ...

5

Another common mistake to avoid

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists x \text{ At(x,UH)} \Rightarrow \text{Smart(x)}$$

is true if there is anyone who is not at ULL!

6

Properties of quantifiers

- $\forall$x $\forall$y is the same as $\forall$y $\forall$x
- $\exists$x $\exists$y is the same as $\exists$y $\exists$x
- $\exists$x $\forall$y is not the same as $\forall$y $\exists$x
- $\exists$x $\forall$y Loves(x,y)
  - "There is a person who loves everyone in the world"
- $\forall$y $\exists$x Loves(x,y)
  - "Everyone in the world is loved by at least one person"

- Quantifier duality: each can be expressed using the other
- $\forall$x Likes(x,IceCream) $\quad\neg\exists$x $\neg$Likes(x,IceCream)
- $\exists$x Likes(x,Broccoli) $\quad\neg\forall$x $\neg$Likes(x,Broccoli)

7

Useful Metatheories

10

- Theorem (Deduction Theorem)
  If $\Sigma \cup \{\phi\}$ |- ψ then Σ |- $\phi \rightarrow \psi$
  i.e. to prove an implication, it is ok to take the antecedent as an additional hypothesis
- Theorem (Rule T)
  If Σ |- ϕ_1 , ..., Σ |- ϕ_n and $\{\phi_1, ..., \phi_n\}$ |- ψ
  then Σ |- ψ
  i.e. provability is transitive, i.e. if you can prove a set of sentences from a set of premises, and you can prove other sentences from those conclusions, then you can derive the final conclusions from the original premises
- Theorem (Refutation Theorem). If $\Sigma \cup \{\neg\phi\}$ is inconsistent, then Σ |- ϕ
  i.e. the basis for proofs by contradiction

8

11

Important results

- connections between semantics and proof theory (Godel, 1930)
  - soundness: if $|- \Phi$ then $|= \Phi$
  - completeness: if $|= \Phi$ then $|- \Phi$
- suggests a procedure for showing a formula Φ is valid, namely ....

9

12

Important results

- (Turing 1936) the validity problem for predicate logic is undecidable i.e. there is no algorithm which, given a wff Φ, halts (correctly) in finite time answering "yes Φ is valid" or "no Φ is not valid"
- however, the suggested procedure for showing a formula Φ is valid, has the property that if Φ is valid the procedure will halts (correctly) in finite time answering "yes Φ is valid" - we refer to this propert as being semi-decidable
- note, the validity problem for propositional logic is decidable, i.e. there is an algorithm which ….

10

Clausal Form

- clausal form = conjunctive normal form, with all quantifiers at the front, and no existential quant.
- given a sentence (say)
  $\Phi = \forall x \forall y \forall z \exists w P(x,y,z,w)$

  let g be a new function symbol and put
  $\Phi_S = \forall x \forall y \forall z P(x,y,z,g(x,y,z))$

  call Φ_S the Skolemisation of Φ .
- if $\Phi = \exists w P(w)$ then $\Phi_S = P(a)$ where a is a new constant symbol
- Why this interest in Skolemisation????

11

Skolemisation...

- Let Φ be a sentence and Φ_S its Skolemisation.

  Theorem
  (i) Every model of Φ_S is a model of Φ
  (ii) If Φ has a model then so does Φ_S
  (iii) Φ is satisfiable iff Φ_S is satisfiable
- i.e. although Φ and Φ_S are not equivalent, one is unsatisfiable, exactly when the other is
- see discussion K pg 197 at bottom
- so we can use the simpler one (no $\exists$) in setting up for resolution

12

Unification

- resolution in prop logic vs. resolution in pred logic?
  - because of Skolemisation, the key difference is the form of the literals
    - proposition letters vs. atomic formulae
      e.g. A & (B v ~C) P(x,y) & (Q(x) v ~R(x,y))
  - so ... how to deal with variables?
- C1 = {P(f(x), y), ~Q(a,b,x)}, C2 = {~P(f(g(c)), g(d))} should yield resolvent {~Q(a,b,g(c))}

  because C1 represents $\forall x \forall y$(P(f(x), y) $\vee$ ~Q(a,b,x)) and so, in particular, P(f(g(c)), g(d)) $\vee$ ~Q(a,b,g(c)) is deducible

13

Unification, continued

- a substitution is .... $\{t_1/x_1, t_2/x_2, ..., t_n/x_n\}$ ... where the x_i are variables and the t_i are terms
- term e.g. a, x, f(x,a), f(f(x,a), x) ....
- S = {P(a,x), Q(y,z,b)}, θ = {h(a)/x, g(b)/y, c/z}
  then Sθ = {P(a,h(a)), Q(g(b), c, b)}
- E = P(x,y,w,u), θ = {f(y)/x, g(z)/y, v/w},
  σ = {a/x, b/y, f(y)/z, w/v, c/u}
- E$\theta\sigma$ = (Eθ)σ = (P(f(y), g(z), v, u))$\boxed{\sigma =}$ P(f(b), g(f(y)), w, c)
  so can write the composition $\theta\sigma$ of the two substitutions as a substitution {f(b)/x, g(f(y))/y, c/u, f(y)/z, w/v}

also written $\sigma \circ \theta$

14

Unification, continued

- given S = $\{L_1, ..., L_n\}$, a unifier σ for S is a substitution such that $L_1\sigma = L_2\sigma = ... = L_n\sigma$
- {P(x,a), P(b,c)} is not unifiable
  {P(f(x), y), P(a,w)} is not unifiable
- {P(x,c), P(b,c)} is unifiable ..... {b/x} only unifier
- {P(f(x),y), P(f(a),w)} is unifiable .... several unifiers, eg {a/x, w/y}, also {a/x, a/y, a/w}, also {a/x, b/y, b/w} …
- a most general unifier (mgu) for S is …

  Theorem an mgu for set S is essentially unique (up to renaming)

15

Most General?

- consider E = {A(x,y,f(z)), A(a,g(z), w)}
- applying σ_1 = {a/x, g(a)/y, a/z, f(a)/w} yields ...
- Eσ_1 = {A(a, g(a), f(a)}
- applying σ_2 = {a/x, g(z)/y, z/z, f(z)/w} yields …
- Eσ_2 = {A(a, g(z), f(z))}
- applying σ_1 can be seen as the same as applying σ_2 first and then σ_3 = {a/z}
- i.e. Eσ_1 = E$\sigma_2\sigma_3$
- σ_2 is called "more general than" σ_1

16

Equality

- $term_1 = term_2$ is true under a given interpretation if and only if $term_1$ and $term_2$ refer to the same object
- E.g., definition of *Sibling* in terms of *Parent*:

  $\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(x = y) \wedge \exists m,f\ \neg(m = f) \wedge Parent(m,x) \wedge Parent(f,x) \wedge Parent(m,y) \wedge Parent(f,y)]$

17

Using FOL

The kinship domain:

- Brothers are siblings

  $\forall x,y\ Brother(x,y) \Leftrightarrow Sibling(x,y)$
- One's mother is one's female parent

  $\forall m,c\ Mother(c) = m \Leftrightarrow (Female(m) \wedge Parent(m,c))$
- "Sibling" is symmetric

  $\forall x,y\ Sibling(x,y) \Leftrightarrow Sibling(y,x)$

18

Using FOL

The set domain:

- $\forall s\ Set(s) \Leftrightarrow (s = \{\}) \vee (\exists x,s_2\ Set(s_2) \wedge s = \{x|s_2\})$
- $\neg\exists x,s\ \{x|s\} = \{\}$
- $\forall x,s\ x \in s \Leftrightarrow s = \{x|s\}$
- $\forall x,s\ x \in s \Leftrightarrow [\exists y,s_2\}\ (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$
- $\forall s_1,s_2\ s_1 \subseteq s_2 \Leftrightarrow (\forall x\ x \in s_1 \Rightarrow x \in s_2)$
- $\forall s_1,s_2\ (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$
- $\forall x,s_1,s_2\ x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$
- $\forall x,s_1,s_2\ x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$

19

Interacting with FOL KBs

- Suppose a wumpus-world agent is using an FOL KB and perceives a smell and a breeze (but no glitter) at $t=5$:

  `Tell`(KB,Percept([Smell,Breeze,None],5))
  `Ask`(KB,$\exists$a BestAction(a,5))

- I.e., does the KB entail some best action at $t=5$?
- Answer: *Yes*, {*a/Shoot*} ← substitution (binding list)
- Given a sentence *S* and a substitution σ,
- *S*σ denotes the result of plugging σ into *S*; e.g.,

  S = Smarter(x,y)
  σ = {x/Hillary,y/Bill}
  *S*σ = Smarter(Hillary,Bill)

- `Ask`(KB,S) returns some/all σ such that KB $\models$ σ

20

Knowledge base for the wumpus world

- Perception
  - ∀t,s,b Percept([s,b,Glitter],t) ⇒ Glitter(t)
- Reflex
  - ∀t Glitter(t) ⇒ BestAction(Grab,t)

21

Deducing hidden properties

- ∀x,y,a,b *Adjacent*([x,y],[a,b]) ⇔
  [a,b] ∈ {[x+1,y], [x-1,y],[x,y+1],[x,y-1]}

Properties of squares:
- ∀s,t *At*(Agent,s,t) ∧ Breeze(t) ⇒ Breezy(s)

Squares are breezy near a pit:

- Diagnostic rule---infer cause from effect
  ∀s Breezy(s) ⇒ \Exi{r} Adjacent(r,s) ∧ Pit(r)$
- Causal rule---infer effect from cause
  ∀r Pit(r) ⇒ [∀s Adjacent(r,s) ⇒ Breezy(s)$]

22

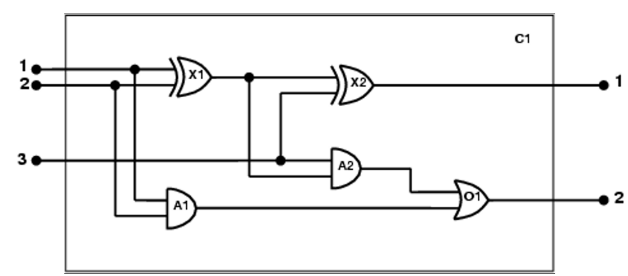
Knowledge engineering in FOL

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on a vocabulary of predicates, functions, and constants
4. Encode general knowledge about the domain
5. Encode a description of the specific problem instance
6. Pose queries to the inference procedure and get answers
7. Debug the knowledge base

23

The electronic circuits domain

One-bit full adder



24

The electronic circuits domain

1. Identify the task
   - Does the circuit actually add properly? (circuit verification)
2. Assemble the relevant knowledge
   - Composed of wires and gates; Types of gates (AND, OR, XOR, NOT)
   - Irrelevant: size, shape, color, cost of gates
3. Decide on a vocabulary
   - Alternatives:
     $Type(X_1) = XOR$
     $Type(X_1, XOR)$
     $XOR(X_1)$

25

The electronic circuits domain

4. Encode general knowledge of the domain
   - $\forall t_1,t_2\ Connected(t_1, t_2) \Rightarrow Signal(t_1) = Signal(t_2)$
   - $\forall t\ Signal(t) = 1 \vee Signal(t) = 0$
   - $1 \neq 0$
   - $\forall t_1,t_2\ Connected(t_1, t_2) \Rightarrow Connected(t_2, t_1)$
   - $\forall g\ Type(g) = OR \Rightarrow Signal(Out(1,g)) = 1 \Leftrightarrow \exists n\ Signal(In(n,g)) = 1$
   - $\forall g\ Type(g) = AND \Rightarrow Signal(Out(1,g)) = 0 \Leftrightarrow \exists n\ Signal(In(n,g)) = 0$
   - $\forall g\ Type(g) = XOR \Rightarrow Signal(Out(1,g)) = 1 \Leftrightarrow Signal(In(1,g)) \neq Signal(In(2,g))$
   - $\forall g\ Type(g) = NOT \Rightarrow Signal(Out(1,g)) \neq Signal(In(1,g))$

26

The electronic circuits domain

5. Encode the specific problem instance

$Type(X_1) = XOR$ $Type(X_2) = XOR$
$Type(A_1) = AND$ $Type(A_2) = AND$
$Type(O_1) = OR$

$Connected(Out(1,X_1),In(1,X_2))$	$Connected(In(1,C_1),In(1,X_1))$
$Connected(Out(1,X_1),In(2,A_2))$	$Connected(In(1,C_1),In(1,A_1))$
$Connected(Out(1,A_2),In(1,O_1))$	$Connected(In(2,C_1),In(2,X_1))$
$Connected(Out(1,A_1),In(2,O_1))$	$Connected(In(2,C_1),In(2,A_1))$
$Connected(Out(1,X_2),Out(1,C_1))$	$Connected(In(3,C_1),In(2,X_2))$
$Connected(Out(1,O_1),Out(2,C_1))$	$Connected(In(3,C_1),In(1,A_2))$

27

The electronic circuits domain

6. Pose queries to the inference procedure
   What are the possible sets of values of all the terminals for the adder circuit?
   $\exists i_1,i_2,i_3,o_1,o_2\ Signal(In(1,C\_1)) = i_1 \wedge Signal(In(2,C_1)) = i_2 \wedge Signal(In(3,C_1)) = i_3 \wedge Signal(Out(1,C_1)) = o_1 \wedge Signal(Out(2,C_1)) = o_2$
7. Debug the knowledge base
8. May have omitted assertions like $1 \neq 0$

28

Summary

- First-order logic:
  - objects and relations are semantic primitives
  - syntax: constants, functions, predicates, equality, quantifiers
- Increased expressive power: sufficient to define wumpus world

29