Using FOL

The set domain:

- The only sets are the empty set and the set to which something is added:
  \[ \forall s \text{ Set}(s) \Leftrightarrow (s = \emptyset) \lor (\exists x, s_2 \text{ Set}(s_2) \land s = \{x|s_2\}) \]
- No element is added to the empty set:
  \[ \neg \exists x \text{ s } x|s| = \emptyset \]
- Adding an in already existing element to a set has no effect:
  \[ \forall x, x \in s \Leftrightarrow s = \{x|s\} \]
- The only elements of a set are those which have been added to it (x is element of s if s is equal to s₂ which has an added y where y=x or x element of s₂):
  \[ \forall x, x \in s \Leftrightarrow (\exists y, s_2 \text{ s } s = \{y|s_2\} \land (x = y \lor x \in s_2)) \]
- A set is a subset of a second set if all elements of the first are elements of the second:
  \[ \forall s_1, s_2 \text{ s}_1 \subseteq s_2 \Leftrightarrow (\forall x \in s_1 \Rightarrow x \in s_2) \]
- Two sets are equal if they are mutual subsets:
  \[ \forall s_1, s_2 \text{ s}_1 = s_2 \Leftrightarrow (s_1 \subseteq s_2 \land s_2 \subseteq s_1) \]
- An element is in the intersection of two sets if it is in both sets:
  \[ \forall x, s_1, s_2 \text{ s } x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \land x \in s_2) \]
- An element is in the union of two sets if it is in one of them:
  \[ \forall x, s_1, s_2 \text{ s } x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \lor x \in s_2) \]

Knowledge base for the wumpus world

- Perception and action:
  - Agents in Wumpus world receive a 5-element vector as input.
  - Sentence must include information about time-step.
  - A typical perception sentence (here binary predicate with list of input constants):
    \[ \text{Percept}([\text{Stench, Breeze, Glitter, None, None}],5) \]
  - Actions can be represented as:
    \[ \text{Turn(Right), Turn(Left), Forward, Shoot, Grab, Release, Climb} \]
  - Finding the right action at a time:
    \[ \exists a \text{ BestAction}(a,5) \]
  - Perception implicates facts about the current state:
    \[ \forall t, s, b \text{ Percept}([s,b,Glitter],t) \Rightarrow \text{Glitter}(t) \]
  - Reflex behavior for agent:
    \[ \forall t \text{ Glitter}(t) \Rightarrow \text{BestAction}(\text{Grab},t) \]
Deducing hidden properties

Environment:
- Using list terms with indices to denote the different squares:
  \([1,2]\) instead of \(\text{Square}_{1,2}\)
- \(\forall x,y,a,b \ \text{Adjacent}([x,y],[a,b]) \iff [a,b] \in \{[x+1,y], [x-1,y], [x,y+1], [x,y-1]\}\)
- \(\text{Wumpus}\)
- \(\text{Home}(X)\)
- \(\text{Pit}(X)\)
- Position of Agent at time:
  \(\text{At}(\text{Agent},s,t)\)
- Properties of squares:
  \(\forall s,t \ \text{At}(\text{Agent},s,t) \land \text{Breeze}(t) \implies \text{Breezy}(s)\)

Squares are breezy near a pit:
- Diagnostic rule---infer cause from effect
  \(\forall s \ \text{Breezy}(s) \implies \exists r \ \text{Adjacent}(r,s) \land \text{Pit}(r)\)
- Causal rule---infer effect from cause
  \(\forall r \ \text{Pit}(r) \implies [\forall s \ \text{Adjacent}(r,s) \implies \text{Breezy}(s)]\)

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 \(t=5\):
  \(\text{Tell}(\text{KB}, \text{Percept}([\text{Smell}, \text{Breeze}, \text{None}], 5))\)
  \(\text{Ask}(\text{KB}, \exists a \ \text{BestAction}(a,5))\)
- I.e., does the KB entail some best action at \(t=5\)?
- Answer: Yes, \(\{a/\text{Shoot}\}\) ← substitution (binding list)
- Given a sentence \(S\) and a substitution \(\sigma\),
- \(S\sigma\) denotes the result of plugging \(\sigma\) into \(S\); e.g.,
  \(S = \text{Smarter}(x,y)\)
  \(\sigma = \{x/\text{Hillary}, y/\text{Bill}\}\)
  \(S\sigma = \text{Smarter(\text{Hillary},\text{Bill})}\)
- \(\text{Ask}(\text{KB}, S)\) returns some/all \(\sigma\) such that \(\text{KB} \vdash \sigma\)
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

Using FOL

The X domain:

- What is this knowledge base’s domain? What “are” the relations?

\[
\forall x \ Z(x) \iff \neg Y1(x) \\
\forall x, y \ W34(x,y) \iff CZ23(y,x) \\
\forall x, y \ ZS2(x,y) \iff ZS2(y,x) \\
\forall x, y \ X1(y) = x \iff (Y1(x) \land W34(x,y)) \\
\forall x, y \ H21(x,y) \iff (Z1(y) \land S45(y,x)) \\
\forall x, y \ W123(x,y) \iff \exists z \ W34(z,x) \land W34(z,y) \\
\forall x, y \ ZS2(x,y) \iff x \neq y \land \exists z \ W34(z,x) \land W34(z,y)
\]

- Compare: \( \forall m, c \ Mother(c) = m \iff (Female(m) \land Parent(m,c)) \)
- Does this look more familiar?
  - The kinship domain
- Using human readable (and understandable) names helps to maintain knowledge base (but might lead to misinterpretation)
The electronic circuits domain

One-bit full adder

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($X_1$)
The electronic circuits domain

4. Encode general knowledge of the domain

1. Two connected in-/outputs have same signal:
   \( \forall t_1, t_2 \text{ Connected}(t_1, t_2) \Rightarrow \text{Signal}(t_1) = \text{Signal}(t_2) \)
2. Signal at in-/output has 0 oder 1 (never both):
   \( \forall t \text{ Signal}(t) = 1 \lor \text{Signal}(t) = 0 \)
   \( 1 \neq 0 \)
3. Connected is commutative:
   \( \forall t_1, t_2 \text{ Connected}(t_1, t_2) \Rightarrow \text{Connected}(t_2, t_1) \)
4. Output of OR-gate is 1 iff an input is 1
   \( \forall g \text{ Type}(g) = \text{OR} \Rightarrow \text{Signal(Out}(1,g)) = 1 \Leftrightarrow \exists n \text{ Signal(In}(n,g)) = 1 \)
5. Output of AND-gate is 0 iff an input is 0:
   \( \forall g \text{ Type}(g) = \text{AND} \Rightarrow \text{Signal(Out}(1,g)) = 0 \Leftrightarrow \exists n \text{ Signal(In}(n,g)) = 0 \)
6. Output of XOR-gate is 1 iff its inputs are unequal:
   \( \forall g \text{ Type}(g) = \text{XOR} \Rightarrow \text{Signal(Out}(1,g)) = 1 \Leftrightarrow \text{Signal(In}(1,g)) \neq \text{Signal(In}(2,g)) \)
7. Output of NOT-gate differs from its input:
   \( \forall g \text{ Type}(g) = \text{NOT} \Rightarrow \text{Signal(Out}(1,g)) \neq \text{Signal(In}(1,g)) \)

5. Encode the specific problem instance

Type\((X_1) = \text{XOR} \quad \text{Type}(X_2) = \text{XOR} \)
Type\((A_1) = \text{AND} \quad \text{Type}(A_2) = \text{AND} \)
Type\((O_1) = \text{OR} \)

Connected\((\text{Out}(1,X_1), \text{In}(1,X_2)) \quad \text{Connected}(\text{In}(1,C_1), \text{In}(1,X_1)) \)
Connected\((\text{Out}(1,X_1), \text{In}(2,A_2)) \quad \text{Connected}(\text{In}(1,C_1), \text{In}(1,A_1)) \)
Connected\((\text{Out}(1,A_2), \text{In}(1,O_1)) \quad \text{Connected}(\text{In}(2,C_1), \text{In}(2,X_1)) \)
Connected\((\text{Out}(1,A_2), \text{In}(2,O_1)) \quad \text{Connected}(\text{In}(2,C_1), \text{In}(2,A_2)) \)
Connected\((\text{Out}(1,X_2), \text{Out}(1,C_1)) \quad \text{Connected}(\text{In}(3,C_1), \text{In}(2,X_2)) \)
Connected\((\text{Out}(1,O_1), \text{Out}(2,C_1)) \quad \text{Connected}(\text{In}(3,C_1), \text{In}(1,A_2)) \)
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?
   \[
   \exists i_1, i_2, i_3, o_1, o_2 \text{ Signal(In}(1, C_1)) = i_1 \land \text{Signal(In}(2, C_1)) = i_2 \land \\
   \text{Signal(In}(3, C_1)) = i_3 \land \text{Signal(Out}(1, C_1)) = o_1 \land \\
   \text{Signal(Out}(2, C_1)) = o_2
   \]

7. Debug the knowledge base
   May have omitted assertions like \(1 \neq 0\)

Summary

• First-order logic:
  • objects and relations are semantic primitives
  • syntax: constants, functions, predicates, equality, quantifiers

• Increased expressive power: sufficient to define wumpus world