First Order Logic

The wumpus world example

Assertions in FOL

- A **domain** is just some part of the world about which we wish to express some knowledge.

- Sentences are added to a knowledge base using **TELL**, exactly as in propositional logic. Such sentences are called **assertions**.
- For example, we can assert that John is a king, Richard is a person, and all kings are persons:
  - TELL(KB, King(John)) .
  - TELL(KB, Person(Richard)).
  - TELL(KB, Vx King(x) = Person(x)).

- We can ask questions of the knowledge base using **ASK**. For example,
  - ASK (KB, King(John)) returns true.

- Questions asked with ASK are called **queries** or **goals**.

- If we want to know what value of x makes the sentence true, we will need a different function, which we call **ASKVARS**,
  - ASKVARS(KB, Person(x))
- and which yields a stream of answers.

- In this case there will be two answers: {x/John} and {x/Richard}. Such an answer is called a **substitution or binding list**.

The wumpus world

- Recall that wumpus world agent receives percepts.
- The percepts will be given to the agent program in the form of a list of five symbols;
- For example, if there is a stench and a breeze, but no glitter, bump, or scream, the agent program will get
  - Percept([Stench, Breeze, None, None, None]).
- The corresponding first-order sentence stored in the knowledge base must include both the percept and the time at which it occurred.
  - Percept([Stench, Breeze, Glitter, None, None),5)
- Percept is a binary predicate, and Stench and so on are constants placed in a list.

- **Actions:** Turn(Right), Turn(Left), Forward, Shoot, Grab, Climb.
- To determine which is best, the agent program executes the query
  - ASKVARS(KB, BestAction(a,5))

  which returns a binding list such as {a/Grab}.

$$\forall t,s,g,w,c\ \ Percept([s,Breeze,g,w,c],t) \Rightarrow Breeze(t)$$
$$\forall t,s,g,w,c\ \ Percept([s,None,g,w,c],t) \Rightarrow \neg Breeze(t)$$
$$\forall t,s,b,w,c\ \ Percept([s,b,Glitter,w,c],t) \Rightarrow Glitter(t)$$
$$\forall t,s,b,w,c\ \ Percept([s,b,None,w,c],t) \Rightarrow \neg Glitter(t)$$

□t Glitter(t) ⇒ BestAction(Grab,t).

- **Objects**: squares, pits, and the wumpus.

- We could name each square—Square1, 5 and so on, but adjacent would have to be an “extra” fact, and we would need one such fact for each pair of squares.

- Adjacency of any two squares can be defined as,
- $\square_{x,y,a,b}$ Adjacent([x,y],[a,b]=?

- **Objects**: squares, pits, and the wumpus.

- We could name each square—Square1, 5 and so on, but adjacent would have to be an “extra” fact, and we would need one such fact for each pair of squares.

- Adjacency of any two squares can be defined as

$$\forall x, y, a, b \ Adjacent([x, y], [a, b]) \Leftrightarrow \\ (x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1)) .$$

- It is simpler to use a unary predicate Pit that is true of squares containing pits.

- Finally, since there is exactly one wumpus, a constant Wumpus is just as good as a unary predicate.

- Agents Location: At(Agent,s,r) to mean that the agent is at square s at time 1.
- We can fix the wumpus to a specific location forever with
  - □t At(Wumpus, [1,3],t).

- Given its current location, the agent can infer properties of the square from properties of its current percept.
- For example, if the agent is at a square and perceives a breeze, then that square is breezy:

$$\forall s,t \ At(Agent,s,t) \wedge Breeze(t) \Rightarrow Breezy(s)$$

- Consider the following statements:

$$R_2: \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \vee P_{2,1}).$$
$$R_3: \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$

- **What they infer in natural language?**

- Given its current location, the agent can infer properties of the square from properties of its current percept.
- For example, if the agent is at a square and perceives a breeze, then that square is breezy:

$$\forall s,t \; At(Agent,s,t) \land Breeze(t) \Rightarrow Breezy(s)$$

- Consider the following statements:

$$R_2: \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \lor P_{2,1}).$$
$$R_3: \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \lor P_{2,2} \lor P_{3,1})$$

- **What they infer in natural language?:**
  - A square is breezy if and only if there is a pit in a neighboring square.

- Given its current location, the agent can infer properties of the square from properties of its current percept.
- For example, if the agent is at a square and perceives a breeze, then that square is breezy:

$$\forall s,t \;\; At(Agent,s,t) \wedge Breeze(t) \Rightarrow Breezy(s)$$

- Consider the following statements:

$$R_2: \quad B_{1,1} \quad \Leftrightarrow \quad (P_{1,2} \vee P_{2,1}).$$
$$R_3: \quad B_{2,1} \quad \Leftrightarrow \quad (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$

- **What they infer in natural language?**
  - A square is breezy if and only if there is a pit in a neighboring square.
- **How do you represent the above sentence in FOL?**

- Given its current location, the agent can infer properties of the square from properties of its current percept.
- For example, if the agent is at a square and perceives a breeze, then that square is breezy:

$$\forall s,t \;\; At(Agent,s,t) \land Breeze(t) \Rightarrow Breezy(s)$$

- Consider the following statements:

$$R_2: \quad B_{1,1} \;\Leftrightarrow\; (P_{1,2} \lor P_{2,1}).$$
$$R_3: \quad B_{2,1} \;\Leftrightarrow\; (P_{1,1} \lor P_{2,2} \lor P_{3,1})$$

- **What they infer in natural language?**
  - A square is breezy if and only if there is a pit in a neighboring square.
- **How do you represent the above sentence in FOL?**

$$\forall s \;\; Breezy(s) \;\Leftrightarrow\; \exists r \;\; Adjacent(r,s) \land Pit(r)$$

The knowledge engineering process

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

Example Problem: Full Adder Circuit