First-Order Logic

Predicate Logic

Resolution

Predicate logic

In propositional logic, a symbol that represents a sentence is **atomic**: it cannot be broken up to find information about its components. For example, consider the sentences:

P_1: "Linda is Mary's mother" P_2: "Mary is Anne's mother"

We can combine these two sentences in many ways to create other sentences, but

we cannot extract any relation between Linda and Anne.

For example, we cannot infer from the above two sentences that Linda is the grandmother of Anne. To do so, we need **Predicate logic: the logic that defines the relation between the parts in a proposition**.

In predicate logic, a sentence is divided into a ***predicate*** and ***arguments***. For example, each of the following propositions can be written as predicates with two arguments:

P_1: “Linda is Mary’s mother”	becomes	mother (Linda, Mary)
P_2: “Mary is Anne’s mother”	becomes	mother (Mary, Anne)

The **relationship of motherhood** in each of the above sentences is defined by the predicate *mother*. If the object Mary in both sentences refers to the same person, we can **infer a new relation** between Linda and Anne:

grandmother **(Linda, Anne)**

RULES OF THE GAME

A sentence in predicate language is defined as follows:

1. A predicate with *n* arguments is a sentence.
2. Any of the two constant values (true and false) is a sentence.
3. If P is a sentence, then $\neg P$ is a sentence.
4. If P and Q are sentences, then $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$, and $P \leftrightarrow Q$ are sentences.

Example

1. The sentence "John works for Ann's sister" can be written as: Works [John, sister(Ann)] in which the function sister(Ann) is used as an argument.
2. The sentence "John's father loves Ann's sister" can be written as: loves[father(John), sister(Ann)]

Predicate logic allows us to use **quantifiers**. Two quantifiers are common in predicate logic:

$\forall$ and $\exists$

1. The first, which is read as "for all", is called the universal quantifier: it states that something is true for every object that its variable represents.
2. The second, which is read as "there exists", is called the existential quantifier: it states that something is true for one or more objects that its variable represents.

Examples

1. The sentence "All men are mortals" can be written as:

$$\forall x[\text{man}(x) \rightarrow \text{mortal}(x)]$$

2. The sentence "Frogs are green" can be written as:

$$\forall x[\text{frog}(x) \rightarrow \text{green}(x)]$$

3. The sentence "Some flowers are red" can be written as:

$$\exists x[\text{flower}(x) \wedge \text{red}(x)]$$

Examples Continued

4. The sentence "John has a book" can be written as:

$$\exists x[\text{book}(x) \wedge \text{has}(\text{John}, x)]$$

5. The sentence "No frog is yellow" can be written as:

$$\forall x[\text{frog}(x) \rightarrow \neg\text{yellow}(x)]$$

or

$$\neg\exists x[\text{frog}(x) \wedge \text{yellow}(x)]$$

Deduction

In predicate logic, if there is no quantifier, the verification of an argument is the same as that which we discussed in propositional logic. However, the verification becomes more complicated if there are quantifiers. For example, the following argument is completely valid.

All men are mortals	Premise 1:
Socrates is a man	Premise 2:
Therefore, Socrates is mortal	Conclusion

Verification of this simple argument is not difficult. We can write this argument as shown next:

$$\forall x\,[\text{man}(x) \rightarrow \text{mortal}(x)]\,,\ \text{man}(\text{Socrates}) \ \vdash\ \text{mortal}(\text{Socrates})$$

Since the first premise talks about all men, we can replace one instance of the class man (Socrates) in that premise to get the following argument:

$$\text{man}(\text{Socrates}) \rightarrow \text{mortal}(\text{Socrates})\,,\ \text{man}(\text{Socrates}) \ \vdash\ \text{mortal}(\text{Socrates})$$

Which is reduced to $M1 \rightarrow M2, M1 \vdash M2$, in which M1 is man(Socrates) and M2 is mortal(Socrates). The result is an argument in propositional logic and can be easily validated.

Rule Base Example

```
R1:  IF animal has hair
     THEN animal is a mammal

R5:  IF animal eats meat
     THEN animal is carnivore

R9:  IF animal is mammal
     AND animal is carnivore
     AND animal has tawney colour
     AND animal has dark spots
     THEN animal is cheetah
```

In FO Logic

- We can write the above rules in first-order logic as follows (there are other ways).

$$
\begin{aligned}
L1.\quad & \forall x \cdot has\_hair(x) \Rightarrow mammal(x) \\
L5.\quad & \forall x \cdot eats(x, meat) \Rightarrow carnivore(x) \\
L9.\quad & \forall x \cdot (mammal(x) \wedge carnivore(x) \wedge \\
& \quad colour(x, tawney) \wedge dark\_spots(x)) \Rightarrow \\
& \quad cheetah(x)
\end{aligned}
$$

- Similarly for the other rules.

Working Memory

- Assume that we have the following information in working memory.

```
cyril has hair,
cyril eats meat,
cyril has tawney colour,
cyril has dark spots
```

- This can be written in first-order logic as follows.

$$
\begin{array}{ll}
F1. & has\_hair(cyril) \\
F2. & eats(cyril, meat) \\
F3. & colour(cyril, tawney) \\
F4. & dark\_spots(cyril)
\end{array}
$$

Goal

- Assume we want to show that

  `cyril is a cheetah`

- This can be written in first-order logic as

$$cheetah(cyril)$$

Reasoning

- To show that

$$cheetah(cyril)$$

follows from the above first-order formula we must show

$$L1, L5, L9, F1, F2, F3, F4 \models cheetah(cyril)$$

- We show

$$L1 \wedge L5 \wedge L9 \wedge F1 \wedge F2 \wedge F3 \wedge F4 \wedge \neg cheetah(cyril)$$

is unsatisfiable. We abbreviate cyril by c

Proof

1. $\neg has\_hair(x) \vee mammal(x)$
2. $\neg eats(y, meat) \vee carnivore(y)$
3. $\neg mammal(z) \vee \neg carnivore(z) \vee \neg colour(z, tawney) \vee \neg dark\_spots(z) \vee cheetah(z)$
4. $has\_hair(c)$
5. $eats(c, meat)$
6. $colour(c, tawney)$
7. $dark\_spots(c)$
8. $\neg cheetah(c)$
9. $\neg mammal(c) \vee \neg carnivore(c) \vee \neg colour(c, tawney) \vee \neg dark\_spots(c)$ $[3, 8, \{z \mapsto c\}]$
10. $\neg mammal(c) \vee \neg carnivore(c) \vee$
    $\neg colour(c, tawney)$ $[7, 9]$
11. $\neg mammal(c) \vee \neg carnivore(c)$ $[6, 10]$
12. $\neg mammal(c) \vee \neg eats(c, meat)$ $[2, 11, \{y \mapsto c\}]$
13. $\neg mammal(c)$ $[5, 12]$
14. $\neg has\_hair(c)$ $[1, 13, \{x \mapsto c]$
15. **false** $[4, 14]$