AI 2002
Artificial Intelligence

Dr. Hashim Yasin

Connection between ∀ and ∃

- Asserting that **"Everyone dislikes parsnips"** is the same as asserting there does not exist someone who likes them, and vice versa:

$$\forall x \;\; \neg Likes(x, Parsnips) \quad \text{is equivalent to} \quad \neg \exists x \;\; Likes(x, Parsnips)$$

- We can go one step further: **"Everyone likes ice cream"** means that there is no one who does not like ice cream:

$$\forall x \;\; Likes(x, IceCream) \quad \text{is equivalent to} \quad \neg \exists x \;\; \neg Likes(x, IceCream)$$

Connection between ∀ and ∃

- **∀ is really conjunction** over the universe of objects while **∃ is a disjunction**.
- **Quantifiers obey De Morgan's rules**. The De Morgan rules for quantified and unquantified sentences are as follows:

$\forall x \;\; \neg Likes(x, Parsnips)$ is equivalent to $\neg \exists x \;\; Likes(x, Parsnips)$

$\forall x \;\; Likes(x, IceCream)$ is equivalent to $\neg \exists x \;\; \neg Likes(x, IceCream)$

$$\begin{array}{llll}
\neg\exists x \;\; P \equiv \forall x \;\; \neg P & \qquad & \neg(P \lor Q) & \equiv \neg P \land \neg Q \\
\neg\forall x \;\; P \equiv \exists x \;\; \neg P & & \neg(P \land Q) & \equiv \neg P \lor \neg Q \\
\neg\exists x \;\; \neg P \equiv \forall x \;\; P & & P \land Q & \equiv \neg(\neg P \lor \neg Q) \\
\neg\forall x \;\; \neg P \equiv \exists x \;\; P & & P \lor Q & \equiv \neg(\neg P \land \neg Q)
\end{array}$$

Inference in First Order Logic

Inference Rules for Quantifiers

Universal Instantiation

$$\boxed{\frac{\forall v \quad \alpha}{\text{SUBST}(\{v/g\}, \alpha)}}$$

for any variable v and ground term g

- $\text{SUBST}(\theta, \alpha)$ denotes the result of applying the substitution θ to the sentence α.

Inference Rules for Quantifiers

Existential Instantiation

- The variable is replaced by a single ***new constant symbol***.
- For any sentence $\boldsymbol{\alpha}$, variable $\boldsymbol{v}$, and constant symbol $\boldsymbol{K}$ that ***does not appear elsewhere in the knowledge base***,

$$\boxed{\frac{\exists v \quad \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}\ .}$$

Inference Rules for Quantifiers

Existential Instantiation

E.g., $\exists x \;\; Crown(x) \wedge OnHead(x, John)$ yields

$$Crown(C_1) \wedge OnHead(C_1, John)$$

- C_1 is the constant which does not appear elsewhere in the knowledge base. Such a constant is called **Skolem constant** and the process is called **Skolemization**.

FOL to Propositional Inference

- Suppose the **KB** consisits of following sentances:

$$\begin{aligned}&\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)\\ &King(John)\\ &Greedy(John)\\ &Brother(Richard, John)\end{aligned}$$

- Instantiating the universal sentence in ***all*** possible ways, we have

$$\begin{aligned}&King(John) \land Greedy(John) \;\Rightarrow\; Evil(John)\\ &King(Richard) \land Greedy(Richard) \;\Rightarrow\; Evil(Richard)\\ &King(John)\\ &Greedy(John)\\ &Brother(Richard, John)\end{aligned}$$

This KB is propositionalized

Unification

Unification

- The **Unify algorithm** takes two sentences and returns a **unifier** for them if one exists:

$$\text{UNIFY}(p, q) = \theta \text{ where } \text{SUBST}(\theta, p) = \text{SUBST}(\theta, q)\ .$$

- Suppose we have a query **AskVars(Knows(John, x)): whom does John know?**
- To answer this question, we have to find all sentences in the knowledge base that unify with **Knows(John, x).**

Unification

- Here are the results of the unification

$$\text{UNIFY}(Knows(John, x),\ Knows(John, Jane)) = \{x/Jane\}$$
$$\text{UNIFY}(Knows(John, x),\ Knows(y, Bill)) = \{x/Bill, y/John\}$$
$$\text{UNIFY}(Knows(John, x),\ Knows(y, Mother(y))) = \{y/John, x/Mother(John)\}$$
$$\text{UNIFY}(Knows(John, x),\ Knows(x, Elizabeth)) = fail\ .$$

- The last unification fails because **x** cannot take on the values **John** and **Elizabeth** at the same time
- **Knows(x, Elizabeth)** means **"Everyone knows Elizabeth,"** and we can infer that **John knows Elizabeth**

Unification --- Standardizing Apart

- This problem arises because two sentences happen to use the **same variable** name, **x**
- The problem can be avoided by **standardizing apart** which means **renaming its variables** to avoid name clashes.
- **Standardizing apart** eliminates overlap of variables.
- For example, we can rename **x** in **Knows(x, Elizabeth)** to $\mathbf{x_{17}}$ (a new variable name) without changing its meaning.

$$\text{UNIFY}(Knows(John, x),\ Knows(x_{17}, Elizabeth)) = \{x/Elizabeth, x_{17}/John\}$$

Unification Example

- $O(F(y), y)$ and $O(F(x), J)$.

- $Q\big(y, G(A, B)\big)$ and $Q(G(x, x), y)$.

Unification Example

- $O(F(y), y)$ and $O(F(x), J)$.

<u>Progressive unification:</u>

$O\ (\underline{F\ (y)}, y), O\ (\underline{F(x)}, J) :$ {} *needs recursion*

$O\ (F\ (\underline{y}), y), O\ (F(\underline{x}), J) : \{y/x\}$

$O\ (F\ (x), \underline{x}), O\ (F\ (x), \underline{J}) : \{y/x, x/J\} = \{y/J, x/J\}$

$O\ (F\ (J), J), O\ (F\ (J), J) : \{y/x, x/J\} = \{y/J, x/J\}$

Unification Example

- $Q\big(y, G(A, B)\big)$ and $Q(G(x, x), y)$.

<u>Progressive unification:</u>

Q(<u>y</u>, G(A, B)), Q(<u>G(x, x)</u>, y) : {<u>y/G(x, x)</u>},

Q(G(x, x), <u>G(A, B)</u>), Q(G(x, x), <u>G(x, x)</u>) : {y/G(x, x)} *needs recursion*

Q(G(x, x), G(<u>A</u>, B)), Q(G(x, x), G(<u>x</u>, x)) : {y/G(x, x), <u>x/A</u>}

Q(G(A, A), G(A, <u>B</u>)), Q(G(A, A), G(A, <u>A</u>)) : {y/G(x, x), x/A}

Cannot unify *constant A* with *constant B*.

First Order Logic ... Examples

FOL Examples

Kinship Domain:

One's husband is one's male spouse:

$$\forall w, h \;\; Husband(h, w) \;\Leftrightarrow\; Male(h) \land Spouse(h, w)\ .$$

Male and female are disjoint categories:

$$\forall x \;\; Male(x) \;\Leftrightarrow\; \neg Female(x)\ .$$

Parent and child are inverse relations:

$$\forall p, c \;\; Parent(p, c) \;\Leftrightarrow\; Child(c, p)\ .$$

A grandparent is a parent of one's parent:

$$\forall g, c \;\; Grandparent(g, c) \;\Leftrightarrow\; \exists p \;\; Parent(g, p) \land Parent(p, c)\ .$$

A sibling is another child of one's parents:

$$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; x \neq y \land \exists p \;\; Parent(p, x) \land Parent(p, y)$$

Reading Material

- **Artificial Intelligence,** A Modern Approach

  Stuart J. Russell and Peter Norvig
  - **Chapter 8 & 9.**