AUTOMATA FORMAL LANGUAGES AND LOGIC

First Order Logic

Dr Pooja Agarwal
Department of Computer Science & Engineering

Outline

- Quantifiers
  - Universal quantification (∀)
  - Existential quantification (∃)
  - Nested quantifiers
  - Connections between ∀ and ∃
- Equality

Universal quantification (∀)

∀ : Universal Quantifiers

∀ x : for all x

"All kings are persons," is written in first-order logic as

$$\forall x\ King(x) \Rightarrow Person(x)$$

symbol x is called a **variable**.

$$\forall x\ P$$

Existential quantification (∃)

∃ : Existential Quantifiers

$\exists$ x : there exist x

We can make a statement about **some object** in the universe without naming it, by using an existential quantifier.

- **Example**

  King John has a crown on his head

  $\exists x\ Crown(x) \wedge OnHead(x, John)$

$\exists x\ P$

Nested quantifiers

- To express more complex sentences using multiple quantifiers.
- The simplest case is where the quantifiers are of the **same type.**
- **Example**:
  "Brothers are siblings" can be written as

- **$\forall x \forall y$ Brother $(x, y) \Rightarrow$ Sibling(x, y)**

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

Nested quantifiers - Example

1. "Everybody loves somebody"

$\forall x \exists y \, Loves(x, y)$ $\qquad$ $\forall x \, (\exists y \, Loves(x, y))$

2. "There is someone who is loved by everyone," we write

$\exists y \forall x \, Loves(x, y)$ $\qquad$ $\exists y \, (\forall x \, Loves(x, y))$

Nested quantifiers

When two quantifiers are used with the same variable name.

Example:

$\forall$ x (Crown(x) $\vee$ ($\exists$ x Brother (Richard, x)))

x in Brother(Richard, x) is existentially quantified.

$\forall$ x (Crown(x) $\vee$ ($\exists$ z Brother (Richard, z)))

Connections between ∀ and ∃

The two quantifiers are actually intimately connected with each other, through negation.

- Declaring

“Everyone dislikes Mango is **the same as** asserting there does not exist someone who likes Mango, and vice versa”

$\forall$ x ¬Likes(x, Mango) is equivalent to ¬$\exists$ x Likes(x, Mango)

Connections between ∀ and ∃

We can go one step further:

"Everyone likes ice cream"

means that there is no one who does not like ice cream:

∀ x Likes(x, IceCream) is equivalent to ¬∃ x ¬Likes(x, IceCream)

Connections between ∀ and ∃

The De Morgan rules for quantified and unquantified sentences are as follows:

$\forall x \neg P \equiv \neg \exists x P$

$\neg \forall x P \equiv \exists x \neg P$

$\forall x P \equiv \neg \exists x \neg P$

$\exists x P \equiv \neg \forall x \neg P$

$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

$P \wedge Q \equiv \neg(\neg P \vee \neg Q)$

$P \vee Q \equiv \neg(\neg P \wedge \neg Q)$

Equality

- First-order logic includes one more way to make atomic sentences, other than using a predicate and terms.

- We can use the **equality symbol** to signify the two terms refer to the same object.

Example:

$$Father(John) = Henry$$

- To say that Richard has at least two brothers, we would write

$$\exists\, x, y\ Brother(x, Richard) \land Brother(y, Richard) \land \neg(x = y)$$

Symbols and interpretations

Syntax of FOL

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term}, \ldots) \mid \mathit{Term} = \mathit{Term} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \land \mathit{Sentence} \\
& \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Leftrightarrow \mathit{Sentence} \\
& \mid & \mathit{Quantifier}\ \mathit{Variable}, \ldots\ \mathit{Sentence} \\
\\
\mathit{Term} & \rightarrow & \mathit{Function}(\mathit{Term}, \ldots) \\
& \mid & \mathit{Constant} \\
& \mid & \mathit{Variable} \\
\\
\mathit{Quantifier} & \rightarrow & \forall \mid \exists \\
\mathit{Constant} & \rightarrow & A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\mathit{Predicate} & \rightarrow & \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} & \rightarrow & \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots \\
\text{OPERATOR PRECEDENCE} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

THANK YOU

Pooja Agarwal
Department of Computer Science & Engineering
poojaagarwal@pes.edu