AUTOMATA FORMAL LANGUAGES AND LOGIC

First Order Logic

Dr Pooja Agarwal
Department of Computer Science & Engineering

AUTOMATA FORMAL LANGUAGES AND LOGIC

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**.

∀ x P

Existential quantification (∃)

∃ : Existential Quantifiers

∃ 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

  ∃ x Crown(x) ∧ OnHead(x, John)

∃ 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”

$\forall x \exists y \text{ Loves}(x, y)$ $\forall x (\exists y \text{ Loves}(x, y))$

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

$\exists y \forall x \text{ Loves}(x, y)$ $\exists y (\forall x \text{ Loves}(x, y))$

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"

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

$$\exists x, y \text{ Brother } (x,\text{Richard }) \wedge \text{ Brother } (y,\text{Richard }) \wedge \neg(x=y)$$

AUTOMATA FORMAL LANGUAGES AND LOGIC

Symbols and interpretations

Syntax of FOL

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

THANK YOU

Pooja Agarwal
Department of Computer Science & Engineering

poojaagarwal@pes.edu