# CS 480 – Introduction to Artificial Intelligence

# Topic: First Order Logic representation
# Chapter: 8

**Mustafa Bilgic**

http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

# Motivation

- Propositional logic is not very expressive
- Consider the following English sentence
  - "Squares adjacent to pits are breezy"
- How do you represent this in propositional logic?
  - $B_{1,1} \Leftrightarrow P_{1,2} \vee P_{2,1}$
  - $B_{2,1} \Leftrightarrow P_{1,1} \vee P_{2,2} \vee P_{3,1}$
  - $B_{3,1} \Leftrightarrow \ldots$
  - …
  - $B_{1,2} \Leftrightarrow \ldots$
  - …

# First-Order Logic (FOL)

- Built around *objects* and *relationships*
- Can express facts and rules about *some* or *all* of the objects in the universe
  - Enables to represent general rules

# First-Order Logic (FOL)

- First-order logic models the world in terms of
  - **Objects**, which are things with individual identities
  - **Relations** that hold among sets of objects and properties of objects that distinguish them from other objects
  - **Functions**, which are a subset of relations where there is only one "value" for any given "input"
- Examples:
  - **Objects**: Students, lectures, companies, cars ...
  - **Relations**: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ... and unary relations (properties): blue, oval, even, large, ...
  - **Functions**: father-of, best-friend, second-half, one-more-than ...

# Syntax

1. Symbols
2. Terms
3. Atomic Sentences
4. Complex Sentences

# Symbols

- Quantifier $\rightarrow \forall \mid \exists$
- Constant $\rightarrow$ A | $X_1$ | John | …
- Variable $\rightarrow$ a | x | s| …
- Predicate $\rightarrow$ True | False | Loves | SisterOf | …
- Function $\rightarrow$ Mother | One-more-than | …

# Sentence

- Sentence →
  - AtomicSentence
  - ComplexSentence
- AtomicSentence →
  - Predicate
  - Predicate(Term, …)
  - Term = Term

# Term

- Term →
  - Function(Term, …)
  - Constant
  - Variable

# Complex Sentence

- ComplexSentence $\rightarrow$
  - $\neg$ Sentence
  - Sentence $\wedge$ Sentence
  - Sentence $\vee$ Sentence
  - Sentence $\Rightarrow$ Sentence
  - Sentence $\Leftrightarrow$ Sentence
  - Quantifier Variable, … Sentence

# All In One Page

- Sentence $\rightarrow$ AtomicSentence | ComplexSentence
- AtomicSentence $\rightarrow$ Predicate | Predicate(Term, …) | Term = Term
- ComplexSentence $\rightarrow$
  - $\neg$ Sentence
  - Sentence $\wedge$ Sentence
  - Sentence $\vee$ Sentence
  - Sentence $\Rightarrow$ Sentence
  - Sentence $\Leftrightarrow$ Sentence
  - Quantifier Variable, … Sentence
- Term $\rightarrow$ Function(Term, …) | Constant | Variable
- Quantifier $\rightarrow$ $\forall$ | $\exists$
- Constant $\rightarrow$ A | $X_1$ | John | …
- Variable $\rightarrow$ a | x | s| …
- Predicate $\rightarrow$ True | False | Loves | SisterOf | …
- Function $\rightarrow$ Mother | One-more-than | …
- Operator precedence: $\neg$, =, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

# QUANTIFIERS

- Universal quantification
  - $\forall x$ P($x$) means that P holds for all values of $x$ in the domain associated with that variable
  - E.g., $\forall x$ Dolphin($x$) $\Rightarrow$ Mammal($x$)
- Existential quantification
  - $\exists x$ P($x$) means that P holds for some value of $x$ in the domain associated with that variable
  - E.g., $\exists x$ Mammal($x$) $\wedge$ LaysEggs($x$)
  - Permits one to make a statement about some object without naming it

# Universal Quantification

- Universal quantifiers are often used with "implies" to form rules
  - $\forall x\ \text{King}(x) \Rightarrow \text{Person}(x)$
- Beware of using $\forall$ with $\wedge$:
  - $\forall x\ \text{King}(x) \wedge \text{Person}(x)$
  - What's wrong with this sentence?

# Existential Quantification

- Existential quantifiers are often used with $\wedge$:
  - $\exists y$ Crown(y) $\wedge$ OnHead($y$, John)
- Beware of using $\exists$ with $\Rightarrow$:
  - $\exists y$ Crown($y$) $\Rightarrow$ OnHead($y$, John)
  - What's wrong with this sentence?

# Quantifier Scope

- Switching the order of universal quantifiers *does not* change the meaning:
  - $\forall x\ \forall y\ P(x,y) \Leftrightarrow \forall y\ \forall x\ P(x,y)$
- Similarly, you can switch the order of existential quantifiers:
  - $\exists x\ \exists y\ P(x,y) \Leftrightarrow \exists y\ \exists x\ P(x,y)$
- Switching the order of universals and existential *does* change the meaning:
  - $\forall x\ \exists y\ \text{Loves}(x,y)$
    - Everyone loves someone
  - $\exists y\ \forall x\ \text{Loves}(x,y)$
    - Someone is loved by everyone

# Connections Between $\forall$ and $\exists$

- We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:
  - $\forall x\ P(x) \Leftrightarrow \neg\exists x\ \neg P(x)$
    - For all $x$, $P(x)$ is true $\Leftrightarrow$ It is not the case that there is an $x$ where $P(x)$ is not true
  - $\forall x\ \neg P(x) \Leftrightarrow \neg\exists x\ P(x)$
  - $\neg\forall x\ P(x) \Leftrightarrow \exists x\ \neg P(x)$
  - $\neg\forall x \neg\ P(x) \Leftrightarrow \exists x\ P(x)$

## Translating English into FOL

- All smart students are hard-working
  - $\forall x\ (\text{Smart}(x) \wedge \text{Student}(x)) \Rightarrow \text{HardWorking}(x)$
- No smart student is hard-working
  - $\forall x\ (\text{Smart}(x) \wedge \text{Student}(x)) \Rightarrow \neg\text{HardWorking}(x)$
  - $\neg\exists x\ \text{Smart}(x) \wedge \text{Student}(x) \wedge \text{HardWorking}(x)$
    - Show that the two above are logically equivalent

# None, At least one, At most one, Exactly one

- Bill has no sister
  - $\neg\exists x$ SisterOf($x$, Bill)
- Bill has at least one sister (Bill has a sister)
  - $\exists x$ SisterOf($x$, Bill)
- Bill has at most one sister
  - $\forall x,y$ SisterOf($x$, Bill) $\wedge$ SisterOf($y$, Bill) $\Rightarrow x=y$
- Bill has exactly one sister
  - $\exists x$ (Sister($x$, Bill) $\wedge \forall y$ (Sister($y$, Bill) $\Rightarrow x=y$))
- Bill has at least two sisters
  - $\exists\ x,y$ SisterOf($x$, Bill) $\wedge$ SisterOf($y$, Bill) $\wedge \neg(x=y)$

# Next

- Chapter 9: inference in FOL