First-Order Logic

Natalie Parde UIC CS 421

Basic Elements of First-Order Logic

Term: First-order logic device for representing objects

- Constants
- Functions
- Variables

Common across all types of terms:

 Each one can be thought of as a way of pointing to a specific object

Basic Elements of First-Order Logic

- Terms:
 - Constants: Specific objects in the world being described
 - Conventionally depicted as single capitalized letters (A, B) or words (Natalie, Usman)
 - Refer to exactly one object, although objects can have more than one constant that refers to them
 - Functions: Concepts that are syntactically equivalent to single-argument predicates
 - Can refer to specific objects without having to associate a named constant with them, e.g., LocationOf(Giordano's)
 - Variables: Provide the ability to make assertions and draw inferences without having to refer to a specific named object
 - Conventionally depicted as single lowercase letters

Basic Elements of First-Order Logic

- Predicates: Symbols that refer to the relations between a fixed number of objects in the domain
 - Can have one or more arguments
 - Serve(Giordano's, Italian)
 - Relates two objects
 - Restaurant(Giordano's)
 - Asserts a property of a single object
- Predicates can be put together using logical connectives
 - and ∧
 - or V
 - implies →
- They can also be negated
 - not ¬

Variables and Quantifiers

- Two basic operators in first-order logic are:
 - 3: The existential quantifier
 - Pronounced "there exists"
 - ∀: The universal quantifier
 - Pronounced "for all"
- These two operators make it possible to represent many more sentences!
 - a restaurant → ∃x Restaurant(x)
 - all restaurants → ∀x Restaurant(x)

We can combine these operators with other basic elements of first-order logic to build logical representations of complex sentences.

- Nikolaos likes Giordano's and Usman likes Artopolis.
 - Like(Nikolaos, Giordano's) Like(Usman, Artopolis)
- Mina likes fast restaurants.
 - $\forall x \; \text{Fast}(x) \rightarrow \text{Like}(\text{Mina}, x)$
- Not everybody likes IDOF.
 - $\exists x \ Person(x) \land \neg Like(x, IDOF)$

Semantics of First-Order Logic

Symbols for objects, properties, and relations acquire meaning based on their correspondences to "real" objects, properties, and relations in the external world

The model-theoretic approach employs a simple set of notions to define meaning based on truth-conditional mappings between expressions in a meaning representation and the state of affairs being modeled

We can determine truth based on the presence of specified terms and predicates.

Р	Q	¬Р	P∧Q	PVQ	P→Q
False	False	True	False	False	True
False	True	True	False	True	True
True	False	False	False	True	False
True	True	False	True	True	True

```
patron = {Natalie, Usman,
Nikolaos, Mina} = {a, b, c, d}
```

restaurants = {Giordano's, IDOF, Artopolis} = {e, f, g}

cuisines = {Italian, Mediterranean, Greek} = {i, j, k}

```
Fast = {f}
TableService = {e, g}
Likes = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}
Serve = {(e, i), (f, j), (g, k)}
```

Natalie likes Giordano's and Usman likes Giordano's.

```
patron = {Natalie, Usman,
Nikolaos, Mina} = {a, b, c, d}
```

restaurants = {Giordano's, IDOF, Artopolis} = {e, f, g}

cuisines = {Italian, Mediterranean, Greek} = {i, j, k}

```
Fast = {f}
TableService = {e, g}
Likes = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}
Serve = {(e, i), (f, j), (g, k)}
```

Natalie likes Giordano's and Usman likes Giordano's.

Likes(Natalie, Giordano's) ∧ Likes(Usman, Giordano's)

```
patron = {Natalie, Usman,
Shahla, Yatri} = {a, b, c, d}
```

restaurants = {Giordano's, IDOF, Artopolis} = {e, f, g}

cuisines = {Italian, Mediterranean, Greek} = {i, j, k}

```
Fast = {f}
TableService = {e, g}
Likes = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}
Serve = {(e, i), (f, j), (g, k)}
```

Natalie likes Giordano's and Usman likes Giordano's.

Likes(Natalie, Giordano's) ∧ Likes(Usman, Giordano's)

Likes(a, e) ∧ Likes(b, e)

```
patron = {Natalie, Usman,
Shahla, Yatri} = {a, b, c, d}
```

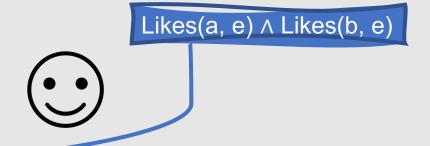
restaurants = {Giordano's, IDOF, Artopolis} = {e, f, g}

cuisines = {Italian, Mediterranean, Greek} = {i, j, k}

```
Fast = {f}
TableService = {e, g}
Likes = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}
Serve = {(e, i), (f, j), (g, k)}
```

Natalie likes Giordano's and Usman likes Giordano's.

Likes(Natalie, Giordano's) ∧ Likes(Usman, Giordano's)



```
patron = {Natalie, Usman,
Shahla, Yatri} = {a, b, c, d}
```

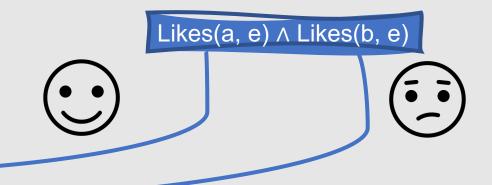
restaurants = {Giordano's, IDOF, Artopolis} = {e, f, g}

cuisines = {Italian, Mediterranean, Greek} = {i, j, k}

```
Fast = {f}
TableService = {e, g}
Likes = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}
Serve = {(e, i), (f, j), (g, k)}
```

Natalie likes Giordano's and Usman likes Giordano's.

Likes(Natalie, Giordano's) ∧ Likes(Usman, Giordano's)



```
patron = {Natalie, Usman,
Shahla, Yatri} = {a, b, c, d}
```

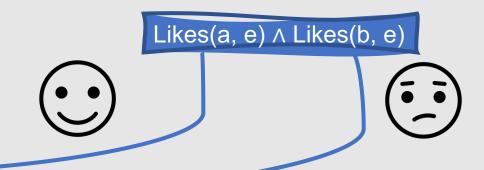
restaurants = {Giordano's, IDOF, Artopolis} = {e, f, g}

cuisines = {Italian, Mediterranean, Greek} = {i, j, k}

```
Fast = {f}
TableService = {e, g}
Likes = {(a, e), (a, f), (a, g), (b, g), (c, e), (d, f)}
Serve = {(e, i), (f, j), (g, k)}
```

Natalie likes Giordano's and Usman likes Giordano's.

Likes(Natalie, Giordano's) ∧ Likes(Usman, Giordano's)



False ...not valid!

A few additional notes....

- Formulas involving ∃ are true if there is *any* substitution of terms for variables that results in a formula that is true according to the model
- Formulas involving ∀ are true only if all substitutions of terms for variables result in formulas that are true according to the model

How do we infer facts not explicitly included in the knowledge base?

- Modus ponens: If a conditional statement is accepted (if p then q), and the antecedent (p) holds, then the consequent (q) may be inferred
- More formally:

$$\frac{\alpha}{a \Rightarrow \beta}$$

Example: Inference

GreekRestaurant(Artopolis) $\forall x \text{ GreekRestaurant}(x) \Rightarrow \text{Serves}(x, GreekFood)$

Serves(Artopolis, GreekFood)

conditional statement accepted ✓

Example: Inference

antecedent holds (our model says that Artopolis is a Greek restaurant) ✓

GreekRestaurant(Artopolis) $\forall x \text{ GreekRestaurant}(x) \Rightarrow \text{Serves}(x, GreekFood)$

Serves(Artopolis, GreekFood)

conditional statement accepted ✓

Example: Inference

antecedent holds (our model says that Artopolis is a Greek restaurant) ✓

GreekRestaurant(Artopolis) $\forall x \text{ GreekRestaurant}(x) \Rightarrow \text{Serves}(x, GreekFood)$

Serves(Artopolis, GreekFood)

conditional statement accepted ✓

consequent may be inferred e