Symbolic Logic: An Introduction

THE UNIVERSITY OF UTAH

First order logic

First order logic

First order logic is a superset of propositional logic

- Propositional logic is the logic of facts which can be true or false
- First order logic is the logic of facts and relations about objects

First order logic

First order logic is a superset of propositional logic

- Propositional logic is the logic of facts which can be true or false
- First order logic is the logic of facts and relations about objects

Extends propositional logic in a few ways:

- Objects in the world
- Propositions about the objects
  - They can take arguments
- Functions map objects to other objects
- Variables denote objects
- Quantifiers and equality

Building blocks

What are the components that can be used to build first order logic sentences?

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

Variables: Stand in for unknown objects

$x, y, z, \ldots$

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

Variables: Stand in for unknown objects

$x, y, z, \ldots$

Connectives: The standard Boolean connectives ($\neg$, $\land$, $\lor$, $\rightarrow$, $\leftrightarrow$)

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

Variables: Stand in for unknown objects

$x, y, z, \ldots$

Connectives: The standard Boolean connectives ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$)

Equality: States that two objects are the same (=)

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

Variables: Stand in for unknown objects

$x, y, z, \ldots$

Connectives: The standard Boolean connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$)

Equality: States that two objects are the same (=)

Quantifiers: Express properties about a collection of objects

Universal ($\forall$), Existential ($\exists$)

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

“Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

Variables: Stand in for unknown objects

$x, y, z, \ldots$

Connectives: The standard Boolean connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$)

Equality: States that two objects are the same (=)

Quantifiers: Express properties about a collection of objects

Universal ($\forall$), Existential ($\exists$)

These are called logical symbols. They have fixed meaning

Building blocks

What are the components that can be used to build first order logic sentences?

Constants: Refer to objects in the world

"Salt Lake City", π, curiosity, "transformer neural network", "John Lennon", "Exit 312"…

Predicates: Refer to properties that apply to sets of arguments (which are objects)

Happy, IsFree, BrotherOf, GreaterThan,…

Functions: Uniquely map a set of arguments (which are objects) to a value (also an object)

NextIntegerOf, FatherOf, LeftLegOf,…

These are called non-logical symbols.
Their meaning is application-dependent

Variables: Stand in for unknown objects

$x, y, z, \ldots$

Connectives: The standard Boolean connectives ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$)

Equality: States that two objects are the same (=)

Quantifiers: Express properties about a collection of objects

Universal ($\forall$), Existential ($\exists$)

Terms

Terms are expressions that refer to objects

Terms

Terms are expressions that refer to objects

Terms can be a constant, a variable or the result of a function

$$\text{Term}: \quad function(term_1, term_2, \cdots, term_n)$$
$$\text{or } variable$$
$$\text{or } constant$$

Terms

Terms are expressions that refer to objects

Terms can be a constant, a variable or the result of a function

$$\text{Term}: \quad function(term_1, term_2, \cdots, term_n)$$
$$\text{or } variable$$
$$\text{or } constant$$

Examples:

- Constants: “Salt Lake City”, π, curiosity, “transformer neural network”, “John Lennon”, “Exit 312”…
- Variables: $x, y, z, \ldots$
- Functions: LeftLegOf(King John), FatherOf(John Lennon), …

Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional

Atomic Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional

Atomic sentences represent a single predicate about a collection of terms or equality of two terms

$$\text{AtomicSentence}: \quad Predicate(term_1, term_2, \cdots, term_n)$$
$$\text{or } term_1 = term_2$$

Atomic Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional

Atomic sentences represent a single predicate about a collection of terms or equality of two terms

$$\text{AtomicSentence}: \ Predicate(term_1, term_2, \cdots, term_n)$$
$$\text{or } term_1 = term_2$$

Examples:

- Brother(Richard, John)
- GreaterThan(13, π)
- OlderThan(FatherOf(Richard), MotherOf(Richard))
- FatherOf(Julian Lennon) = John Lennon

Compositional Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional

Compositional sentences are constructed using the Boolean connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

The rules are the same as with propositional logic

Compositional Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional

Compositional sentences are constructed using the Boolean connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

The rules are the same as with propositional logic

Compositional Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional

Compositional sentences are constructed using the Boolean connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

The rules are the same as with propositional logic

Examples:

- Brother(Richard, John) $\wedge$ Brother(John, Richard)
- $\neg$LessThan(13, π)
- OlderThan(John, Richard) $\wedge$ OlderThan(Richard, 30) $\rightarrow$ OlderThan(John, 30)
- Married(x, y) $\leftrightarrow$ Married(y, x)

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Quantifiers allow us to express properties of entire collections of objects

First order logic has two standard quantifiers: Universal and Existential

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Universal quantifier: Allows us to express the fact that a property holds for all members in a domain

$$\forall\, x, \mathrm{Predicate}(x, other\ args)$$

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Universal quantifier: Allows us to express the fact that a property holds for all members in a domain

$$\forall\, x, \mathrm{Predicate}(x, other\ args)$$

This allows us to compactly state facts like “All humans are mortal”

$$\forall x, \mathrm{Human}(x) \rightarrow \mathrm{Mortal}(x)$$

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Universal quantifier: Allows us to express the fact that a property holds for all members in a domain

$$\forall\, x, \text{Predicate}(x, \mathit{other\ args})$$

This allows us to compactly state facts like "All humans are mortal"

$$\forall x, \text{Human}(x) \rightarrow \text{Mortal}(x)$$

This is short hand for a large conjunction:

$$\begin{aligned}&(\text{Human}(\mathit{Richard}) \rightarrow \text{Mortal}(\mathit{Richard}))\\ &\wedge (\text{Human}(\pi) \rightarrow \text{Mortal}(\pi))\\ &\wedge (\text{Human}(\mathit{John\ Lennon}) \rightarrow \text{Mortal}(\mathit{John\ Lennon}))\\ &\wedge \cdots\end{aligned}$$

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Existential quantifier: Allows us to express the fact that a property holds for at least one member in a domain

$$\exists\ x, \text{Predicate}(x, other\ args)$$

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Existential quantifier: Allows us to express the fact that a property holds for at least one member in a domain

$$\exists\ x, \mathrm{Predicate}(x, other\ args)$$

This allows us to compactly state facts like “Bob has a pet dog”

$$\exists x, \mathrm{Pet}(x, Bob) \wedge \mathrm{Dog}(x)$$

Quantified Sentences

Sentences in first order logic representations of facts

- Can evaluate to `true` or `false`
- Can be atomic or compositional
- Can be quantified

Existential quantifier: Allows us to express the fact that a property holds for at least one member in a domain

$$\exists\, x, \text{Predicate}(x, other\ args)$$

This allows us to compactly state facts like “Bob has a pet dog”

$$\exists x, \text{Pet}(x, Bob) \land \text{Dog}(x)$$

This is short hand for a large disjunction:

$$\begin{aligned}&(\text{Pet}(Richard, Bob) \land \text{Dog}(Richard))\\ &\lor (\text{Pet}(\pi, Bob) \land \text{Dog}(\pi))\\ &\lor (\text{Pet}(x, John\ Lennon) \land \text{Dog}(John\ Lennon))\\ &\lor \cdots\end{aligned}$$

Other aspects of first order logic

Equality (=): Says that two terms refer to the same object

Example: $(\mathrm{FatherOf}(Luke) = Anakin) \land (Anakin = Vader)$

Other aspects of first order logic

Equality (=): Says that two terms refer to the same object

Example: $(\text{FatherOf}(Luke) = Anakin) \land (Anakin = Vader)$

Ground expressions: An expression with no variables

Examples of ground expressions:

- A ground term: $\text{FatherOf}(Luke)$
- A ground formula: $\text{Brother}(Bob, Richard) \land \text{Brother}(Richard, Bob)$

Not ground expressions: $\text{FatherOf}(x)$, $\forall x, \exists c \ \text{CoffeeShop}(c) \land \text{Near}(x, c)$

And there’s a lot more about first order logic

There are well-defined generalizations of ideas we saw with propositional logic:

- Interpretations to formally define semantics
- Inference rules, resolution, etc

We will not cover these in this class. Check out Russell & Norvig if you’re interested