KR-IST: Lecture 7b
First-order Logic

Chris Thornton

November 22, 2013

From reasoning to knowledge representation

With any set of implication rules (e.g., the celebrity rulebase), search-like processes can be used to determine implied facts and conclusions.

The combination of rulebase and inference method can be viewed as a *representation of knowledge* for the domain, i.e., a **knowledge base** (**KB**).

A system which packages up knowledge represented this way is a **knowledge-based** or **expert system**.

From knowledge representation to logic

Rule-based methods of knowledge representation are also known as **logics**.

With a history stretching back 2000 years, the study of logic and formal reasoning was a kind of pre-computation AI.

AI methods of knowledge representation (**KR**) are generally based on adapted systems of formal logic.

What is a logic?

A logic is a formal language for representing facts and properties of a world in a precise, unambiguous way.

What is a logic?

A logic is a formal language for representing facts and properties of a world in a precise, unambiguous way.

- **Sentences** in the language are constructed according to formal rules and relationships.

What is a logic?

A logic is a formal language for representing facts and properties of a world in a precise, unambiguous way.

- **Sentences** in the language are constructed according to formal rules and relationships.
- A **semantics** identifies the formal meaning of any sentence.

What is a logic?

A logic is a formal language for representing facts and properties of a world in a precise, unambiguous way.

- **Sentences** in the language are constructed according to formal rules and relationships.
- A **semantics** identifies the formal meaning of any sentence.
- An **inference method** allows new sentences to be generated from existing sentences.

What is a logic?

A logic is a formal language for representing facts and properties of a world in a precise, unambiguous way.

- **Sentences** in the language are constructed according to formal rules and relationships.
- A **semantics** identifies the formal meaning of any sentence.
- An **inference method** allows new sentences to be generated from existing sentences.

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$
- Or: $\vee$

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\land$
- Or: $\lor$
- Not: $\neg$

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$
- Or: $\vee$
- Not: $\neg$
- Implies: $\Rightarrow$

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$
- Or: $\vee$
- Not: $\neg$
- Implies: $\Rightarrow$
- Therefore: $\vdash$

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$
- Or: $\vee$
- Not: $\neg$
- Implies: $\Rightarrow$
- Therefore: $\vdash$
- wrapping parentheses: (...)

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$
- Or: $\vee$
- Not: $\neg$
- Implies: $\Rightarrow$
- Therefore: $\vdash$
- wrapping parentheses: (...)
- logical constants: *true*, *false*, *unknown*

Propositional logic

The simplest logic of all.

Allows facts about the world to be represented as sentences formed from:

- propositional symbols: P, Q, R, S...
- And: $\wedge$
- Or: $\vee$
- Not: $\neg$
- Implies: $\Rightarrow$
- Therefore: $\vdash$
- wrapping parentheses: (...)
- logical constants: *true*, *false*, *unknown*

Propositional logic examples

Propositional logic examples

- 'It is humid': Q

Propositional logic examples

- 'It is humid': Q
- 'If it is humid, then it is hot': $Q \Rightarrow P$

Propositional logic examples

- 'It is humid': Q
- 'If it is humid, then it is hot': $Q \Rightarrow P$
- 'If it is hot and humid, then it is raining': $(P \wedge Q) \Rightarrow R$

Propositional logic examples

- 'It is humid': Q
- 'If it is humid, then it is hot': $Q \Rightarrow P$
- 'If it is hot and humid, then it is raining': $(P \wedge Q) \Rightarrow R$

Truth tables

The meaning of logical relationships is defined using **truth tables**.

A truth table shows how truth values combine under the relevant relationship.

\`AND'

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

\`OR'

P	Q	P ∨ Q
T	T	T
T	F	T
F	T	T
F	F	F

Basic rules of inference

The two main rules of inference are

Basic rules of inference

The two main rules of inference are

- Modus ponens

Basic rules of inference

The two main rules of inference are

- Modus ponens

$$P \Rightarrow Q$$
$$P$$
$$\vdash Q$$

Basic rules of inference

The two main rules of inference are

- Modus ponens

  $P \Rightarrow Q$
  P
  $\vdash Q$

- Modus Tolens

Basic rules of inference

The two main rules of inference are

- Modus ponens

$$P \Rightarrow Q$$
$$P$$
$$\vdash Q$$

- Modus Tolens

$$P \Rightarrow Q$$
$$\neg\, Q$$
$$\vdash \neg\, P$$

Soundness and completeness

In logic, sentences have a *value*.

This is normally a *truth value*, i.e., true or false.

The associated inference method is then said to be

Soundness and completeness

In logic, sentences have a *value*.

This is normally a *truth value*, i.e., true or false.

The associated inference method is then said to be

- **sound** if it does not generate false facts (i.e., contradictions), and

Soundness and completeness

In logic, sentences have a *value*.

This is normally a *truth value*, i.e., true or false.

The associated inference method is then said to be

- **sound** if it does not generate false facts (i.e., contradictions), and
- **complete** if it is able to produce every sentence that is logically entailed by any existing set of sentences.

Soundness and completeness

In logic, sentences have a *value*.

This is normally a *truth value*, i.e., true or false.

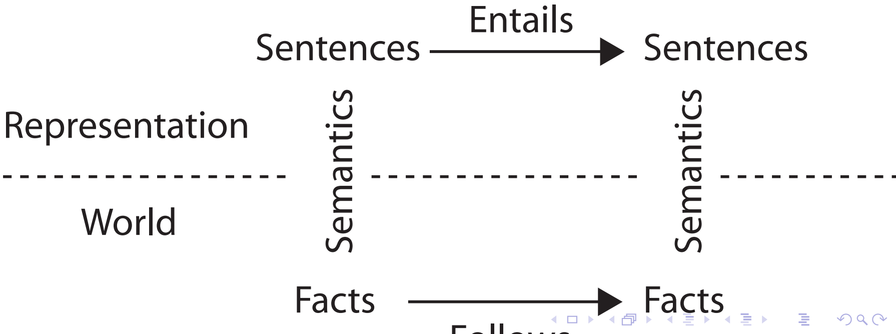
The associated inference method is then said to be

- **sound** if it does not generate false facts (i.e., contradictions), and
- **complete** if it is able to produce every sentence that is logically entailed by any existing set of sentences.

Semantics

A semantics maps sentences to facts in the world; e.g., the mapping determines which objects in the world are referenced by which objects in the language. This is called a **referential semantics**.

The way one fact follows another should be mirrored by the way one sentence is *entailed* by another.



Problems with propositional logic

In Propositional logic, we have no way to represent properties of objects.

We cannot represent property-based generalisations.

For example, it is impossible to represent this **categorical syllogism** in Propositional logic:

> Every person is mortal
> Tony Blair is a person
> Therefore Tony Blair is mortal

First-order logic

First-order logic (FOL) (also known as **first-order predicate calculus** or **FOPC**) adds

First-order logic

First-order logic (FOL) (also known as **first-order predicate calculus** or **FOPC**) adds

- predicates which can represent properties, e.g., mortal(person), or relationships, e.g., likes(fred, sausages),

First-order logic

First-order logic (FOL) (also known as **first-order predicate calculus** or **FOPC**) adds

- predicates which can represent properties, e.g., mortal(person), or relationships, e.g., likes(fred, sausages),
- existentially quantified variables, e.g., $\exists$ at least one X such that...

First-order logic

First-order logic (FOL) (also known as **first-order predicate calculus** or **FOPC**) adds

- predicates which can represent properties, e.g., mortal(person), or relationships, e.g., likes(fred, sausages),
- existentially quantified variables, e.g., ∃ at least one X such that...
- universallly quantified variables, e.g., ∀ X it is the case that...

First-order logic

First-order logic (FOL) (also known as **first-order predicate calculus** or **FOPC**) adds

- predicates which can represent properties, e.g., mortal(person), or relationships, e.g., likes(fred, sausages),
- existentially quantified variables, e.g., $\exists$ at least one X such that...
- universallly quantified variables, e.g., $\forall$ X it is the case that...

Using predicates to represent relationships

In order to represent a relationship between individual objects, we can use a predicate specifying the objects as its arguments.

Using predicates to represent relationships

In order to represent a relationship between individual objects, we can use a predicate specifying the objects as its arguments.

- 'Alison likes Richard and chocolate'

Using predicates to represent relationships

In order to represent a relationship between individual objects, we can use a predicate specifying the objects as its arguments.

- 'Alison likes Richard and chocolate'
- likes(alison, richard) $\wedge$ likes(alison, chocolate)

Using predicates to represent relationships

In order to represent a relationship between individual objects, we can use a predicate specifying the objects as its arguments.

- ‘Alison likes Richard and chocolate’
- likes(alison, richard) $\wedge$ likes(alison, chocolate)

Using predicates within a rule

Using predicates within a rule

- ‘If Richard is a friend of Alison then Alison likes Richard’

Using predicates within a rule

- 'If Richard is a friend of Alison then Alison likes Richard'
- friends(alison, richard) $\Rightarrow$ likes(alison, richard)

Using predicates within a rule

- ‘If Richard is a friend of Alison then Alison likes Richard’
- friends(alison, richard) $\Rightarrow$ likes(alison, richard)

Using variables with predicates to capture generalisations

We can capture generalisations by asserting that any instance of a given class has the relevant property. For example

Using variables with predicates to capture generalisations

We can capture generalisations by asserting that any instance of a given class has the relevant property. For example

- ‘Every elephant is grey’

Using variables with predicates to capture generalisations

We can capture generalisations by asserting that any instance of a given class has the relevant property. For example

- 'Every elephant is grey'
- $\forall$ X: elephant(X) $\Rightarrow$ grey(X)

Using variables with predicates to capture generalisations

We can capture generalisations by asserting that any instance of a given class has the relevant property. For example

- 'Every elephant is grey'
- $\forall$ X: elephant(X) $\Rightarrow$ grey(X)

Using quantifiers and variables

We can use quantification to distinguish general and specific assertion.

Using quantifiers and variables

We can use quantification to distinguish general and specific assertion.

- ‘There is a white aligator’: $\exists$ X: alligator(X) $\wedge$ white(X)

Using quantifiers and variables

We can use quantification to distinguish general and specific assertion.

- 'There is a white aligator': $\exists$ X: alligator(X) $\wedge$ white(X)
- 'Alison eats everything that she likes': $\forall$ X: likes(alison, X) $\Rightarrow$ eats(alison, X)

Using quantifiers and variables

We can use quantification to distinguish general and specific assertion.

- 'There is a white aligator': $\exists$ X: alligator(X) $\wedge$ white(X)
- 'Alison eats everything that she likes': $\forall$ X: likes(alison, X) $\Rightarrow$ eats(alison, X)
- 'There exists some bird that doesn't fly': $\exists$ X: bird(X) $\Rightarrow \neg$ flies(X).

Using quantifiers and variables

We can use quantification to distinguish general and specific assertion.

- ‘There is a white aligator’: $\exists$ X: alligator(X) $\wedge$ white(X)
- ‘Alison eats everything that she likes’: $\forall$ X: likes(alison, X) $\Rightarrow$ eats(alison, X)
- ‘There exists some bird that doesn’t fly’: $\exists$ X: bird(X) $\Rightarrow \neg$ flies(X).
- ‘Every person has something that they love’: $\forall$ X: person(X) $\Rightarrow \exists$ Y: loves(X,Y)

Using quantifiers and variables

We can use quantification to distinguish general and specific assertion.

- 'There is a white aligator': $\exists$ X: alligator(X) $\wedge$ white(X)
- 'Alison eats everything that she likes': $\forall$ X: likes(alison, X) $\Rightarrow$ eats(alison, X)
- 'There exists some bird that doesn't fly': $\exists$ X: bird(X) $\Rightarrow \neg$ flies(X).
- 'Every person has something that they love': $\forall$ X: person(X) $\Rightarrow \exists$ Y: loves(X,Y)

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

- 'Every gardener likes the sun.' $\forall$ x: gardener(x) $\Rightarrow$ likes(x,Sun)

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

- 'Every gardener likes the sun.' $\forall$ x: gardener(x) $\Rightarrow$ likes(x,Sun)
- 'You can fool some of the people all of the time.' $\exists$ x: (person(x) $\wedge$ $\forall$ t: (time(t) $=\rangle$ can-fool(x,t)))

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

- 'Every gardener likes the sun.' $\forall$ x: gardener(x) $\Rightarrow$ likes(x,Sun)
- 'You can fool some of the people all of the time.' $\exists$ x: (person(x) $\wedge$ $\forall$ t: (time(t) $=\rangle$ can-fool(x,t)))
- 'You can fool all of the people some of the time.' $\forall$ x: (person(x) $\Rightarrow$ $\exists$ t: (time(t) $\wedge$ can-fool(x,t)))

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

- 'Every gardener likes the sun.' $\forall$ x: gardener(x) $\Rightarrow$ likes(x,Sun)
- 'You can fool some of the people all of the time.' $\exists$ x: (person(x) $\wedge$ $\forall$ t: (time(t) $=\rangle$ can-fool(x,t)))
- 'You can fool all of the people some of the time.' $\forall$ x: (person(x) $\Rightarrow$ $\exists$ t: (time(t) $\wedge$ can-fool(x,t)))
- 'All purple mushrooms are poisonous.' $\forall$ x: (mushroom(x) $\wedge$ purple(x)) $\Rightarrow$ poisonous(x)

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

- 'Every gardener likes the sun.' $\forall$ x: gardener(x) $\Rightarrow$ likes(x,Sun)
- 'You can fool some of the people all of the time.' $\exists$ x: (person(x) $\wedge$ $\forall$ t: (time(t) $=\rangle$ can-fool(x,t)))
- 'You can fool all of the people some of the time.' $\forall$ x: (person(x) $\Rightarrow$ $\exists$ t: (time(t) $\wedge$ can-fool(x,t)))
- 'All purple mushrooms are poisonous.' $\forall$ x: (mushroom(x) $\wedge$ purple(x)) $\Rightarrow$ poisonous(x)
- 'No purple mushroom is poisonous.' $\neg$ $\exists$ x: purple(x) $\wedge$ mushroom(x) $\wedge$ poisonous(x) or, equivalently, $\forall$ x: (mushroom(x) $\wedge$ purple(x)) $\Rightarrow$ $\neg$ poisonous(x)

More examples

Usually, there are several ways to render a sentence in FOL. There's no 'one right answer'.

Consider these

- 'Every gardener likes the sun.' $\forall$ x: gardener(x) $\Rightarrow$ likes(x,Sun)
- 'You can fool some of the people all of the time.' $\exists$ x: (person(x) $\wedge$ $\forall$ t: (time(t) $=\rangle$ can-fool(x,t)))
- 'You can fool all of the people some of the time.' $\forall$ x: (person(x) $\Rightarrow$ $\exists$ t: (time(t) $\wedge$ can-fool(x,t)))
- 'All purple mushrooms are poisonous.' $\forall$ x: (mushroom(x) $\wedge$ purple(x)) $\Rightarrow$ poisonous(x)
- 'No purple mushroom is poisonous.' $\neg$ $\exists$ x: purple(x) $\wedge$ mushroom(x) $\wedge$ poisonous(x) or, equivalently, $\forall$ x: (mushroom(x) $\wedge$ purple(x)) $\Rightarrow$ $\neg$ poisonous(x)

Examples cont.

Examples cont.

- ‘There are exactly two purple mushrooms.’ $\exists$ x $\exists$ y: mushroom(x) $\wedge$ purple(x) $\wedge$ mushroom(y) $\wedge$ purple(y) $\wedge$ $\neg$ (x=y) $\wedge$ $\forall$ z: (mushroom(z) $\wedge$ purple(z)) $\Rightarrow$ ((x=z) $\vee$ (y=z))

Examples cont.

- 'There are exactly two purple mushrooms.' $\exists$ x $\exists$ y: mushroom(x) $\wedge$ purple(x) $\wedge$ mushroom(y) $\wedge$ purple(y) $\wedge$ $\neg$ (x=y) $\wedge$ $\forall$ z: (mushroom(z) $\wedge$ purple(z)) $\Rightarrow$ ((x=z) $\vee$ (y=z))
- 'Deb is not tall.' $\neg$ tall(Deb)

Examples cont.

- 'There are exactly two purple mushrooms.' $\exists$ x $\exists$ y: mushroom(x) $\wedge$ purple(x) $\wedge$ mushroom(y) $\wedge$ purple(y) $\wedge$ $\neg$ (x=y) $\wedge$ $\forall$ z: (mushroom(z) $\wedge$ purple(z)) $\Rightarrow$ ((x=z) $\vee$ (y=z))
- 'Deb is not tall.' $\neg$ tall(Deb)
- 'X is above Y if X is on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.' $\forall$ x $\forall$ y: above(x,y) $\Leftrightarrow$ (on(x,y) $\vee$ $\exists$ z (on(x,z) $\wedge$ above(z,y)))

Examples cont.

- 'There are exactly two purple mushrooms.' $\exists$ x $\exists$ y: mushroom(x) $\wedge$ purple(x) $\wedge$ mushroom(y) $\wedge$ purple(y) $\wedge$ $\neg$ (x=y) $\wedge$ $\forall$ z: (mushroom(z) $\wedge$ purple(z)) $\Rightarrow$ ((x=z) $\vee$ (y=z))
- 'Deb is not tall.' $\neg$ tall(Deb)
- 'X is above Y if X is on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.' $\forall$ x $\forall$ y: above(x,y) $\Leftrightarrow$ (on(x,y) $\vee$ $\exists$ z (on(x,z) $\wedge$ above(z,y)))

Problems with FOL

FOL is a powerful language for representing knowledge.

But its expressiveness complicates the derivation of inferences. (It gets easier if we exliminate existential quantification and assume 'negation by failure'.)

Also, in FOL you cannot construct sentences which make assertions about other sentences. For example, you cannot say things like 'there exists a property such that...'

For this task, you need a **higher-order logic**.

Special-purpose logics

Other flavours of logic offer different forms of sentence valuation.

For example

Special-purpose logics

Other flavours of logic offer different forms of sentence valuation.

For example

- **fuzzy logic**: evaluation in terms of probability;

Special-purpose logics

Other flavours of logic offer different forms of sentence valuation.

For example

- **fuzzy logic**: evaluation in terms of probability;
- **modal logic**: evaluation in terms of a propositional attitude such as belief. Good for sentences containing ‘should’, ‘must’ etc.

Special-purpose logics

Other flavours of logic offer different forms of sentence valuation.

For example

- **fuzzy logic**: evaluation in terms of probability;
- **modal logic**: evaluation in terms of a propositional attitude such as belief. Good for sentences containing ‘should’, ‘must’ etc.
- **temporal logic**: evaluation in terms of truth at a particular moment in time.

Special-purpose logics

Other flavours of logic offer different forms of sentence valuation.

For example

- **fuzzy logic**: evaluation in terms of probability;
- **modal logic**: evaluation in terms of a propositional attitude such as belief. Good for sentences containing 'should', 'must' etc.
- **temporal logic**: evaluation in terms of truth at a particular moment in time.

The Frame problem

A fundamental difficulty for sentential representation is the **frame problem**.

This affects all varieties of knowledge representation but is particularly apparent where evaluation is in terms of truth, and rules are used to define the results of *actions*.

Frame problem example

Suppose we have

$$\text{paint}(X, C) \Rightarrow \text{color}(X, C)$$
$$\text{move}(X, P) \Rightarrow \text{position}(X, P)$$

and it is known that

paint(tony, blue).
move(tony, garden).

We should then be able to infer that

$$\text{colour}(\text{tony}, \text{blue}) \wedge \text{position}(\text{tony}, \text{garden})$$

But the inference is, in fact, logically *unsound*

There is the possibility that the colour of tony gets changed by the move action.

Addressing the frame problem

The most obvious way to protect against the frame problem is to add rules which capture the *non-effects* of actions.

Such rules are known as **frame axioms**.

For example

$$\text{move(X, P)} \wedge \text{color-before-move(X, C)} \Rightarrow \text{color(X, C)}.$$

asserts the fact that moving an object will not affect its colour.

However, this is not satisfactory.

Since *most* actions do not affect *most* properties of a situation, in a domain comprising m actions and n properties, we are going to need approximately $m \times n$ frame axioms.

The Epistemological Frame Problem

The underlying puzzle is how a cognitive creature with many beliefs about the world can update those beliefs when it performs an act so that they remain roughly faithful to the word.

Imagine being the designer of a robot that has to carry out an everyday task, such as making a cup of tea. Now, suppose the robot has to take a tea-cup from the cupboard. The present location of the cup is represented as a sentence in its database of facts alongside those representing innumerable other features of the ongoing situation, such as the ambient temperature, the configuration of its arms, the current date, the colour of the tea-pot, and so on. Having grasped the cup and withdrawn it from the cupboard, the robot needs to update this database. The location of the cup has clearly changed, so that's one fact that demands revision. But which other sentences require modification?

Summary

Summary

- From reasoning to knowledge representation

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic
- First-order logic

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic
- First-order logic
- Using variables with predicates and quantifiers to capture generalisations

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic
- First-order logic
- Using variables with predicates and quantifiers to capture generalisations
- Problems with FOL

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic
- First-order logic
- Using variables with predicates and quantifiers to capture generalisations
- Problems with FOL
- Special-purpose logics

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic
- First-order logic
- Using variables with predicates and quantifiers to capture generalisations
- Problems with FOL
- Special-purpose logics
- The Frame problem

Summary

- From reasoning to knowledge representation
- From knowledge representation to logic
- Propositional logic
- Truth tables
- Basic rules of inference
- Soundness and completeness
- Semantics
- Problems with propositional logic
- First-order logic
- Using variables with predicates and quantifiers to capture generalisations
- Problems with FOL
- Special-purpose logics
- The Frame problem

Questions

Questions

- What does it mean to say that predicate calculus is a ‘first-order’ logic?

Questions

- What does it mean to say that predicate calculus is a ‘first-order’ logic?
- What is the difference between implication and conjunction?

Questions

- What does it mean to say that predicate calculus is a ‘first-order’ logic?
- What is the difference between implication and conjunction?

Exercises

Exercises

- Represent the following assertions using propositional logic: Fred enjoys swimming or Fred enjoys dancing; Fred enjoys swimming and Fred enjoys dancing; Fred doesn't dance; If Fred enjoys dancing then Fred enjoys swimming.

Exercises

- Represent the following assertions using propositional logic: Fred enjoys swimming or Fred enjoys dancing; Fred enjoys swimming and Fred enjoys dancing; Fred doesn't dance; If Fred enjoys dancing then Fred enjoys swimming.
- Represent the same assertions using first-order logic.

Exercises

- Represent the following assertions using propositional logic: Fred enjoys swimming or Fred enjoys dancing; Fred enjoys swimming and Fred enjoys dancing; Fred doesn't dance; If Fred enjoys dancing then Fred enjoys swimming.
- Represent the same assertions using first-order logic.
- State the inference known as modus ponens. State the inference known as modus tollens.

Exercises

- Represent the following assertions using propositional logic: Fred enjoys swimming or Fred enjoys dancing; Fred enjoys swimming and Fred enjoys dancing; Fred doesn't dance; If Fred enjoys dancing then Fred enjoys swimming.
- Represent the same assertions using first-order logic.
- State the inference known as modus ponens. State the inference known as modus tollens.
- Write out the truth table for the exclusive-or relation.

Exercises

- Represent the following assertions using propositional logic: Fred enjoys swimming or Fred enjoys dancing; Fred enjoys swimming and Fred enjoys dancing; Fred doesn't dance; If Fred enjoys dancing then Fred enjoys swimming.
- Represent the same assertions using first-order logic.
- State the inference known as modus ponens. State the inference known as modus tollens.
- Write out the truth table for the exclusive-or relation.
- List the principle limitations of first-order logic and give an example of a situation in which they would be significant.

Exercises

- Represent the following assertions using propositional logic: Fred enjoys swimming or Fred enjoys dancing; Fred enjoys swimming and Fred enjoys dancing; Fred doesn't dance; If Fred enjoys dancing then Fred enjoys swimming.
- Represent the same assertions using first-order logic.
- State the inference known as modus ponens. State the inference known as modus tollens.
- Write out the truth table for the exclusive-or relation.
- List the principle limitations of first-order logic and give an example of a situation in which they would be significant.

Exercises cont.

Exercises cont.

- Consider this categorical syllogism.

Exercises cont.

- Consider this categorical syllogism.

  Every knowledge representation is formal
  Propositional logic is a knowledge representation
  Therefore propositional logic is formal

Exercises cont.

- Consider this categorical syllogism.

  Every knowledge representation is formal
  Propositional logic is a knowledge representation
  Therefore propositional logic is formal

- Produce the most accurate propositional representation of this assertion you can think of.

Exercises cont.

- Consider this categorical syllogism.

  Every knowledge representation is formal
  Propositional logic is a knowledge representation
  Therefore propositional logic is formal

- Produce the most accurate propositional representation of this assertion you can think of.
- Produce the best predicate logic representation you can think of.

Exercises cont.

- Consider this categorical syllogism.

  Every knowledge representation is formal
  Propositional logic is a knowledge representation
  Therefore propositional logic is formal

- Produce the most accurate propositional representation of this assertion you can think of.
- Produce the best predicate logic representation you can think of.

Exercises cont.

Exercises cont.

- Represent the first three verses of the song ‘House of the Rising Sun’ in FOL.

Exercises cont.

- Represent the first three verses of the song 'House of the Rising Sun' in FOL.

  There is a house in New Orleans
  They call the Rising Sun
  And it's been the ruin of many a poor boy
  And God I know I'm one

  My mother was a tailor
  She sewed my new bluejeans
  My father was a gamblin' man
  Down in New Orleans

  Now the only thing a gambler needs
  Is a suitcase and trunk
  And the only time he's satisfied

Exercises cont.

Exercises cont.

- Using first-order logic, represent as accurately as possible the information contained in these comments on the availability of mortgages during the credit crunch.

Exercises cont.

- Using first-order logic, represent as accurately as possible the information contained in these comments on the availability of mortgages during the credit crunch.

To be considered for the best mortgage deals during the current
difficult conditions, you must borrow substantially less than the
full purchase price, have a perfect credit record and be able to act fast.

Only people who have built up savings over
several years and have shown their ability to live on less than their salary are able to get a mortgage.

It is first-time buyers who are hardest hit by the need to

Exercises cont.

Exercises cont.

- Use FOL to represent the information contained in the lyric to ‘When a man loves a woman’.

Exercises cont.

- Use FOL to represent the information contained in the lyric to 'When a man loves a woman'.

  When a man loves a woman
  Can't keep his mind on nothing else
  He'll trade the world
  For the good things he's found
  If she's bad, he can't see it
  She can do no wrong
  Turn his back on his best friend
  If he put her down

  When a man loves a woman
  Spend his very last time
  Tryin' to hold on to what he needs
  He'd give up all his comfort