First Order Logic - Syntax and Semantics

K. Subramani[1]

[1]Lane Department of Computer Science and Electrical Engineering
West Virginia University

30 January, 4 February, 6 February 2013

Outline

1. Motivation

Outline

Outline

First Order Logic

First Order Logic

Limitations of Propositional Logic

First Order Logic

Limitations of Propositional Logic

Propositional Logic has limited expressiveness.

First Order Logic

Limitations of Propositional Logic

Propositional Logic has limited expressiveness. For instance, how would you capture the assertion, “Property P is true of every positive number”?

First Order Logic

Limitations of Propositional Logic

Propositional Logic has limited expressiveness. For instance, how would you capture the assertion, “Property P is true of every positive number”? $P_1 \wedge P_2 \ldots P_\infty$ is neither compact nor useful.

First Order Logic

Limitations of Propositional Logic

Propositional Logic has limited expressiveness. For instance, how would you capture the assertion, “Property P is true of every positive number”? $P_1 \wedge P_2 \ldots P_\infty$ is neither compact nor useful.
First-order Logic (FOL) extends Propositional Logic (PL) with predicates, functions and quantifiers.

Syntax of FOL

Syntax of FOL

Basics

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

(ii) The universal quantifier $(\forall x)P(x)$ indicates that property P holds for all x in some domain.

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

(ii) The universal quantifier $(\forall x)P(x)$ indicates that property P holds for all x in some domain.

(iii) The existential quantifier $(\exists x)P(x)$ indicates that property P holds for some x in some domain.

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

(ii) The universal quantifier $(\forall x)P(x)$ indicates that property P holds for all x in some domain.

(iii) The existential quantifier $(\exists x)P(x)$ indicates that property P holds for some x in some domain.

(iv) A function is a mapping from the domain of interest to a range.

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

(ii) The universal quantifier $(\forall x)P(x)$ indicates that property P holds for all x in some domain.

(iii) The existential quantifier $(\exists x)P(x)$ indicates that property P holds for some x in some domain.

(iv) A function is a mapping from the domain of interest to a range.

(v) Variables are used as placeholders (0-ary predicates).

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

(ii) The universal quantifier $(\forall x)P(x)$ indicates that property P holds for all x in some domain.

(iii) The existential quantifier $(\exists x)P(x)$ indicates that property P holds for some x in some domain.

(iv) A function is a mapping from the domain of interest to a range.

(v) Variables are used as placeholders (0-ary predicates).

(vi) Constants are used to represent values that do not change.

Syntax of FOL

Basics

(i) Predicates are used to describe properties of objects. e.g., $P(x)$ could stand for the property that x is divisible by 3.

(ii) The universal quantifier $(\forall x)P(x)$ indicates that property P holds for all x in some domain.

(iii) The existential quantifier $(\exists x)P(x)$ indicates that property P holds for some x in some domain.

(iv) A function is a mapping from the domain of interest to a range.

(v) Variables are used as placeholders (0-ary predicates).

(vi) Constants are used to represent values that do not change.

(vii) Terms, atom, literal, formula.

Scope

Scope

Bound and Free variables

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x, y) \rightarrow (\exists y)R(x, y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x, y) \rightarrow (\exists y)R(x, y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Closures

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \to (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Closures

(i) Existential closure.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Closures

(i) Existential closure.

(ii) Universal closure.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Closures

(i) Existential closure.
(ii) Universal closure.

Subformulas and Subterms

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Closures

(i) Existential closure.
(ii) Universal closure.

Subformulas and Subterms

1. Subformulas and strict subformulas.

Scope

Bound and Free variables

Consider the expression:

$$(\forall x)[Q(x,y) \rightarrow (\exists y)R(x,y)]$$

The x occurrences are bound to the $(\forall x)$ quantifier. The first y is said to be a free variable, since it is not bound to any quantifier. The scope of a quantifier is the portion of the predicate formula to which it applies. A formula F is closed if it has no free variables.

Closures

(i) Existential closure.
(ii) Universal closure.

Subformulas and Subterms

1. Subformulas and strict subformulas.
2. Subterms and strict subterms.

Outline

Converting English to Predicate Logic

Converting English to Predicate Logic

Note

Not an easy task!

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous).

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). “Hang him not, let him free” and “Hang him, not let him free”!

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \rightarrow U(x)]$.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \to U(x)]$.

(ii) Some parrots are ugly.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \rightarrow U(x)]$.

(ii) Some parrots are ugly. $(\exists x)[P(x) \wedge U(x)]$.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \rightarrow U(x)]$.

(ii) Some parrots are ugly. $(\exists x)[P(x) \land U(x)]$.

(iii) All dogs chase all rabbits.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \rightarrow U(x)]$.

(ii) Some parrots are ugly. $(\exists x)[P(x) \wedge U(x)]$.

(iii) All dogs chase all rabbits. $(\forall x)[D(x) \rightarrow (\forall y)[R(y) \rightarrow C(x,y)]]$.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \rightarrow U(x)]$.

(ii) Some parrots are ugly. $(\exists x)[P(x) \wedge U(x)]$.

(iii) All dogs chase all rabbits. $(\forall x)[D(x) \rightarrow (\forall y)[R(y) \rightarrow C(x, y)]]$.

(iv) Some dogs chase all rabbits.

Converting English to Predicate Logic

Note

Not an easy task! More than one result possible, depending on semantics of English language (which is not unambiguous). "Hang him not, let him free" and "Hang him, not let him free"!

Example

(i) All parrots are ugly. $(\forall x)[P(x) \rightarrow U(x)]$.

(ii) Some parrots are ugly. $(\exists x)[P(x) \land U(x)]$.

(iii) All dogs chase all rabbits. $(\forall x)[D(x) \rightarrow (\forall y)[R(y) \rightarrow C(x, y)]]$.

(iv) Some dogs chase all rabbits. $(\exists x)[D(x) \land (\forall y)[R(y) \rightarrow C(x, y)]]$.

Exercise

Exercise

Let $S(x)$ denote “x is a student”, $I(x)$ denote “x is intelligent” and $M(x)$ denote “x likes music”.

Exercise

Exercise

Let $S(x)$ denote "x is a student", $I(x)$ denote "x is intelligent" and $M(x)$ denote "x likes music". Give predicate wffs for:

(i) All students are intelligent.

(ii) Some intelligent students like music.

(iii) Only intelligent students like music.

Exercise

Exercise

Let $S(x)$ denote “x is a student”, $I(x)$ denote “x is intelligent” and $M(x)$ denote “x likes music”. Give predicate wffs for:

(i) All students are intelligent.
(ii) Some intelligent students like music.
(iii) Only intelligent students like music.

Solution

(i) $(\forall x)[S(x) \rightarrow I(x)]$.

Exercise

Exercise

Let $S(x)$ denote "x is a student", $I(x)$ denote "x is intelligent" and $M(x)$ denote "x likes music". Give predicate wffs for:

(i) All students are intelligent.

(ii) Some intelligent students like music.

(iii) Only intelligent students like music.

Solution

(i) $(\forall x)[S(x) \rightarrow I(x)]$.

(ii) $(\exists x)[S(x) \wedge I(x) \wedge M(x)]$.

Exercise

Exercise

Let $S(x)$ denote "x is a student", $I(x)$ denote "x is intelligent" and $M(x)$ denote "x likes music". Give predicate wffs for:

(i) All students are intelligent.

(ii) Some intelligent students like music.

(iii) Only intelligent students like music.

Solution

(i) $(\forall x)[S(x) \rightarrow I(x)]$.

(ii) $(\exists x)[S(x) \land I(x) \land M(x)]$.

(iii) $(\forall x)[M(x) \rightarrow [S(x) \land I(x)]]$.

FOL in mathematics

FOL in mathematics

Example

FOL in mathematics

Example

(i) The length of one side of a triangle is less than the sum of the lengths of the other two sides.

FOL in mathematics

Example

(i) The length of one side of a triangle is less than the sum of the lengths of the other two sides.

(ii) Fermat's last theorem.

Semantics

Semantics

Main Points

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I,

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function.

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain,

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain,

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain,

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

(i) $F \ : \ x + y \geq z \rightarrow y \geq z - x$?

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

(i) $F \; : \; x + y \geq z \rightarrow y \geq z - x$?

(ii) $G \; : \; (\forall x) \, P(x, a)$?

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

(i) $F\ :\ x + y \geq z \rightarrow y \geq z - x$?

(ii) $G\ :\ (\forall x)\, P(x, a)$?

These questions are meaningless without the interpretation! Consider the following interpretation for G:

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

(i) $F \ : \ x + y \geq z \rightarrow y \geq z - x$?

(ii) $G \ : \ (\forall x)\, P(x, a)$?

These questions are meaningless without the interpretation! Consider the following interpretation for G: The domain is the set of natural numbers $N = \{0, 1, ..., \}$, $P(x, y)$ stands for $x \geq y$ and a is 0.

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

(i) $F \ : x + y \geq z \rightarrow y \geq z - x$?

(ii) $G \ : (\forall x) P(x, a)$?

These questions are meaningless without the interpretation! Consider the following interpretation for G: The domain is the set of natural numbers $N = \{0, 1, ..., \}$, $P(x, y)$ stands for $x \geq y$ and a is 0. Clearly, in this interpretation, the expression is **true**.

Semantics

Main Points

(i) As with propositional logic, a FOL formula evaluates to **true** or **false**.

(ii) Terms of a FOL formula evaluate to values from a specified domain.

(iii) A first order interpretation I, is a 2-tuple (D_I, α_I), where D_I is a non-empty set called the domain of interpretation and α_I is an assignment function. The assignment function maps constants and variables in the expression to objects in the domain, predicates in the domain to properties of objects in the domain, and functions in the expression to functions over the domain, respecting arity.

Example

What is the truth-value of

(i) $F \,:\, x + y \geq z \rightarrow y \geq z - x$?

(ii) $G \,:\, (\forall x)\, P(x, a)$?

These questions are meaningless without the interpretation! Consider the following interpretation for G: The domain is the set of natural numbers $N = \{0, 1, ..., \}$, $P(x, y)$ stands for $x \geq y$ and a is 0. Clearly, in this interpretation, the expression is **true**. Can you think of an interpretation in which G is **false**?

Inductive definition of semantics

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

(ii) $I \not\models \bot$.

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.
(ii) $I \not\models \bot$.

Atoms

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms.

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.
(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.
(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

(i) $\alpha_I[f(t_1, t_2, \ldots t_n)] =$

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

(i) $\alpha_I[f(t_1, t_2, \ldots t_n)] = \alpha_I[f(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

(i) $\alpha_I[f(t_1, t_2, \ldots t_n)] = \alpha_I[f(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

(ii) $\alpha_I[p(t_1, t_2, \ldots t_n)] =$

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.
(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

(i) $\alpha_I[f(t_1, t_2, \ldots t_n)] = \alpha_I[f(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.
(ii) $\alpha_I[p(t_1, t_2, \ldots t_n)] = \alpha_I[p(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

(i) $\alpha_I[f(t_1, t_2, \ldots t_n)] = \alpha_I[f(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

(ii) $\alpha_I[p(t_1, t_2, \ldots t_n)] = \alpha_I[p(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

Then

$$I \models p(t_1, t_2, \ldots t_n) \text{ iff}$$

Inductive definition of semantics

Goal

Given a FOL formula F and an interpretation $I : (D_I, \alpha_I)$, we want to compute if F evaluates to **true**, under that interpretation.

Truth Symbols

(i) $I \models \top$.

(ii) $I \not\models \bot$.

Atoms

Use the assignment function α_I to recursively evaluate arbitrary terms and arbitrary atoms. For instance,

(i) $\alpha_I[f(t_1, t_2, \ldots t_n)] = \alpha_I[f(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

(ii) $\alpha_I[p(t_1, t_2, \ldots t_n)] = \alpha_I[p(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]]$.

Then

$$I \models p(t_1, t_2, \ldots t_n) \text{ iff } \alpha_I[p(\alpha_I[t_1], \alpha_I[t_2], \ldots \alpha_I[t_n]] = \textbf{true}.$$

Completing the induction

Completing the induction

General unquantified FOL formulas

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Quantified Formulas

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Quantified Formulas

(i) $I \models (\forall x)\ F$ if and only if for every $v \in D$, $I \lhd \{x \mapsto v\} \models F$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Quantified Formulas

(i) $I \models (\forall x)\ F$ if and only if for every $v \in D$, $I \lhd \{x \mapsto v\} \models F$.

(ii) $I \models (\exists x)\ F$ if and only if there exists some $v \in D$, $I \lhd \{x \mapsto v\} \models F$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Quantified Formulas

(i) $I \models (\forall x)\ F$ if and only if for every $v \in D$, $I \lhd \{x \mapsto v\} \models F$.

(ii) $I \models\ (\exists x)\ F$ if and only if there exists some $v \in D$, $I \lhd \{x \mapsto v\} \models F$.

Example

Consider the formula $F\ :\ x + y > z \rightarrow y > z - x$.

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.

(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.

(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.

(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Quantified Formulas

(i) $I \models (\forall x)\ F$ if and only if for every $v \in D$, $I \triangleleft \{x \mapsto v\} \models F$.

(ii) $I \models\ (\exists x)\ F$ if and only if there exists some $v \in D$, $I \triangleleft \{x \mapsto v\} \models F$.

Example

Consider the formula $F\ :\ x + y > z \rightarrow y > z - x$. Is F **true**

Completing the induction

General unquantified FOL formulas

(i) $I \models \neg F$ iff $I \not\models F$.
(ii) $I \models F_1 \wedge F_2$ iff $I \models F_1$ and $I \models F_2$.
(iii) $I \models F_1 \vee F_2$ iff $I \models F_1$ or $I \models F_2$.
(iv) $I \models F_1 \rightarrow F_2$ iff if $I \models F_1$, then $I \models F_2$.

Quantified Formulas

(i) $I \models (\forall x)\ F$ if and only if for every $v \in D$, $I \lhd \{x \mapsto v\} \models F$.
(ii) $I \models (\exists x)\ F$ if and only if there exists some $v \in D$, $I \lhd \{x \mapsto v\} \models F$.

Example

Consider the formula $F\ :\ x + y > z \rightarrow y > z - x$. Is F **true** under the interpretation $I\ :\ (\mathbb{Z}, \alpha_I)$, where $\alpha_I\ :\ \{+ \mapsto +_{\mathbb{Z}}, - \mapsto -_{\mathbb{Z}}, > \mapsto >_{\mathbb{Z}}, x \mapsto 13_{\mathbb{Z}}, y \mapsto 42_{\mathbb{Z}}, z \mapsto 1_{\mathbb{Z}}\}$?