Logics and Statistics for Language Modeling

Carlos Areces

areces@loria.fr
http://www.loria.fr/~areces/ls
INRIA Nancy Grand Est
Nancy, France

2009/2010

Today's Program

Today's Program

- First Order Logics.
  - Syntax
  - Models
  - Semantics

First Order Logic

First Order Logic

While propositional logic assumes that the world contains facts,

First Order Logic

While propositional logic assumes that the world contains facts, first-order logic (like natural language) assumes the world contains

First Order Logic

While propositional logic assumes that the world contains facts, first-order logic (like natural language) assumes the world contains

- Objects: or elements in the world like people, houses, numbers, cars, windows, doors, cups, trees, cats, ...

First Order Logic

While propositional logic assumes that the world contains facts, first-order logic (like natural language) assumes the world contains

- Objects: or elements in the world like people, houses, numbers, cars, windows, doors, cups, trees, cats, ...
- Functions: (one to one mappings) defined over objects in the world like father of, best briend, one more than, plus, ...

First Order Logic

While propositional logic assumes that the world contains facts, first-order logic (like natural language) assumes the world contains

- Objects: or elements in the world like people, houses, numbers, cars, windows, doors, cups, trees, cats, ...
- Functions: (one to one mappings) defined over objects in the world like father of, best briend, one more than, plus, ...
- Relations: between or about objects in the world like red, round, prime, brother of, bigger than, part of, taller than,...

Syntax of FOL: The Logical Language

Syntax of FOL: The Logical Language

In FOL we use a logical language which has a fixed interpretation:

Syntax of FOL: The Logical Language

In FOL we use a logical language which has a fixed interpretation:

- Boolean Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$

Syntax of FOL: The Logical Language

In FOL we use a logical language which has a fixed interpretation:

- Boolean Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
- Quantifiers: $\forall$, $\exists$

Syntax of FOL: The Logical Language

In FOL we use a logical language which has a fixed interpretation:

- Boolean Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
- Quantifiers: $\forall$, $\exists$
- Equality: $=$

Syntax of FOL: The Logical Language

In FOL we use a logical language which has a fixed interpretation:

- Boolean Connectives: $\neg$, $\rightarrow$, $\wedge$, $\vee$, $\leftrightarrow$
- Quantifiers: $\forall$, $\exists$
- Equality: $=$
- Punctuation:), (, .

Syntax of FOL: Basic elements

Syntax of FOL: Basic elements

A first-order language is defined in terms a set of constant symbols variables, predicate symbols, function symbols (a signature):

Syntax of FOL: Basic elements

A first-order language is defined in terms a set of constant symbols variables, predicate symbols, function symbols (a signature):

- Constant symbols: π, 2, Carlos, ID223, B230, LORIA, etc. They will stand by special elements in a given situation.

Syntax of FOL: Basic elements

A first-order language is defined in terms a set of constant symbols variables, predicate symbols, function symbols (a signature):

- Constant symbols: π, 2, Carlos, ID223, B230, LORIA, etc.
  They will stand by special elements in a given situation.
- Variables: x, y, z, a, b, etc.
  They stand for arbitrary objects in a given situation.

Syntax of FOL: Basic elements

A first-order language is defined in terms a set of constant symbols variables, predicate symbols, function symbols (a signature):

- Constant symbols: π, 2, Carlos, ID223, B230, LORIA, etc.
  They will stand by special elements in a given situation.
- Variables: x, y, z, a, b, etc.
  They stand for arbitrary objects in a given situation.
- Function symbols: Sqrt, LeftLegOf, LengthOf, Succ, etc.
  They will be the names of the functions in a given situation.

Syntax of FOL: Basic elements

A first-order language is defined in terms a set of constant symbols variables, predicate symbols, function symbols (a signature):

- Constant symbols: π, 2, Carlos, ID223, B230, LORIA, etc.
  They will stand by special elements in a given situation.
- Variables: x, y, z, a, b, etc.
  They stand for arbitrary objects in a given situation.
- Function symbols: Sqrt, LeftLegOf, LengthOf, Succ, etc.
  They will be the names of the functions in a given situation.
- Predicate symbols: Brother, $\geq$, Tall, etc.
  They will be the particular relations among or about the elements in a given situation.

Syntax of FOL: Basic elements

A first-order language is defined in terms a set of constant symbols variables, predicate symbols, function symbols (a signature):

- Constant symbols: π, 2, Carlos, ID223, B230, LORIA, etc.
  They will stand by special elements in a given situation.
- Variables: x, y, z, a, b, etc.
  They stand for arbitrary objects in a given situation.
- Function symbols: Sqrt, LeftLegOf, LengthOf, Succ, etc.
  They will be the names of the functions in a given situation.
- Predicate symbols: Brother, $\geq$, Tall, etc.
  They will be the particular relations among or about the elements in a given situation.

These are also called the Non Logical Language and we should specify their meaning in each particular situation.

Syntax of FOL: Signature

Syntax of FOL: Signature

A signature is a 4-uple $S = \langle VAR, CONS, RELS, FUNS \rangle$, where

Syntax of FOL: Signature

A signature is a 4-uple $S = \langle VAR, CONS, RELS, FUNS \rangle$, where

VAR is a set of variables (e.g., $VAR = \{x, y, z\}$)

Syntax of FOL: Signature

A signature is a 4-uple $S = \langle VAR, CONS, RELS, FUNS \rangle$, where

VAR is a set of variables (e.g., $VAR = \{x, y, z\}$)
$CONS$ is a set of constant symbols (e.g., $CONS = \{3, carlos, a\}$)

Syntax of FOL: Signature

A signature is a 4-uple $S = \langle VAR, CONS, RELS, FUNS \rangle$, where

VAR is a set of variables (e.g., $VAR = \{x, y, z\}$)
$CONS$ is a set of constant symbols (e.g., $CONS = \{3, carlos, a\}$)
$RELS$ is a set of relational symbols
(e.g. $RELS = \{R^3, BrotherOf^2, Tall^1\}$)

Syntax of FOL: Signature

A signature is a 4-uple $S = \langle VAR, CONS, RELS, FUNS \rangle$, where

VAR is a set of variables (e.g., $VAR = \{x, y, z\}$)
$CONS$ is a set of constant symbols (e.g., $CONS = \{3, carlos, a\}$)
$RELS$ is a set of relational symbols
(e.g. $RELS = \{R^3, BrotherOf^2, Tall^1\}$)
$FUNS$ is a set of function symbols (e.g. $FUNS = \{f^1, +^2\}$).

Syntax of FOL: Signature

A signature is a 4-uple $S = \langle VAR, CONS, RELS, FUNS \rangle$, where

VAR is a set of variables (e.g., $VAR = \{x, y, z\}$)
$CONS$ is a set of constant symbols (e.g., $CONS = \{3, carlos, a\}$)
$RELS$ is a set of relational symbols
(e.g. $RELS = \{R^3, BrotherOf^2, Tall^1\}$)
$FUNS$ is a set of function symbols (e.g. $FUNS = \{f^1, +^2\}$).

Elements in the signature will be our basic vocabulary when writing down FO formulas.

Syntax of FOL: Terms

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol($\text{term}_1, \ldots, \text{term}_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

x

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,. . . , term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO
$f(x)$	

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term_1,. . . , term_n)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO
$f(x)$	YES

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO
$f(x)$	YES
$+(3)$	

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO
$f(x)$	YES
$+(3)$	NO

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,. . . , term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO
$f(x)$	YES
$+(3)$	NO
$Tall(carlos)$	

Syntax of FOL: Terms

Assume we are given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$.

Terms:

a constant symbol	E.g.: Carlos, Graciela
a variable	E.g.: x
function symbol(term$_1$,..., term$_n$)	E.g.: LeftLegOf(Carlos)

Terms can be seen as 'complex names' for elements in the situation.

Examples and Non-Examples: Consider the signature in the previous slide:

$VAR = \{x, y, z\}$
$CONS = \{3, carlos, a\}$
$RELS = \{R^3, BrotherOf^2, Tall^1\}$
$FUNS = \{f^1, +^2\}$

x	YES
f	NO
$f(x)$	YES
$+(3)$	NO
$Tall(carlos)$	NO

Syntax of FOL: Atomic Formulas

Syntax of FOL: Atomic Formulas

Atomic formulas are the basic units about which we can claim truth or falsity in a given situation.

Syntax of FOL: Atomic Formulas

Atomic formulas are the basic units about which we can claim truth or falsity in a given situation.

Atomic Formulas:
$(\text{term}_1 = \text{term}_2)$

E.g. Length(LeftLegOf(Carlos)) = Lenght(RightLegOf(Carlos))

Syntax of FOL: Atomic Formulas

Atomic formulas are the basic units about which we can claim truth or falsity in a given situation.

Atomic Formulas:

($\text{term}_1 = \text{term}_2$)

E.g. Length(LeftLegOf(Carlos)) = Lenght(RightLegOf(Carlos))

predicate symbol($\text{term}_1, \ldots, \text{term}_2$)

E.g. Brother(Carlos,Graciela)

Syntax of FOL: Atomic Formulas

Atomic formulas are the basic units about which we can claim truth or falsity in a given situation.

Atomic Formulas:

(term$_1$ = term$_2$)

E.g. Length(LeftLegOf(Carlos)) = Lenght(RightLegOf(Carlos))

predicate symbol(term$_1$, . . . , term$_2$)

E.g. Brother(Carlos,Graciela)

Examples: The following are atomic formulas (over the appropriate signature)

$(+(2,2) = 4)$
$<(x,2)$
$f(y)$
$P(x, \pi, carlos, 4, xyz)$

Syntax of FOL: Atomic Formulas

Atomic formulas are the basic units about which we can claim truth or falsity in a given situation.

Atomic Formulas:

$(\text{term}_1 = \text{term}_2)$

E.g. Length(LeftLegOf(Carlos)) = Lenght(RightLegOf(Carlos))

predicate symbol$(\text{term}_1, \ldots, \text{term}_2)$

E.g. Brother(Carlos,Graciela)

Examples: The following are atomic formulas (over the appropriate signature)

$(+(2, 2) = 4)$
$<(x, 2)$
$f(y)$
$P(x, \pi, carlos, 4, xyz)$

Which is the appropriate signature?:

Syntax of FOL: Atomic Formulas

Atomic formulas are the basic units about which we can claim truth or falsity in a given situation.

Atomic Formulas:

(term$_1$ = term$_2$)

E.g. Length(LeftLegOf(Carlos)) = Lenght(RightLegOf(Carlos))

predicate symbol(term$_1$, ..., term$_2$)

E.g. Brother(Carlos,Graciela)

Examples: The following are atomic formulas (over the appropriate signature)

$(+(2,2) = 4)$
$<(x, 2)$
$f(y)$
$P(x, \pi, carlos, 4, xyz)$

Which is the appropriate signature?:
$VAR = \{x, y\}$
$CONS = \{2, 4, \pi, carlos, xyz\}$
$FUNS = \{+^2\}$
$RELS = \{<^2, f^1, P^5\}$

Syntax of FOL: Complex Formulas

Syntax of FOL: Complex Formulas

Complex formulas are built up from atomic formulas using the boolean connectives and the quantifiers:

$$FORMS := ATOM \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \exists x.(\varphi)$$

where φ is a formula, and x is a variable.

Syntax of FOL: Complex Formulas

Complex formulas are built up from atomic formulas using the boolean connectives and the quantifiers:

$$FORMS := ATOM \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \exists x.(\varphi)$$

where φ is a formula, and x is a variable.
($\forall$ is defined in terms of $\neg$ and $\exists$: $\forall x.(\varphi) \equiv \neg\exists x.(\neg\varphi)$.)

Syntax of FOL: Complex Formulas

Complex formulas are built up from atomic formulas using the boolean connectives and the quantifiers:

$$FORMS := ATOM \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \exists x.(\varphi)$$

where φ is a formula, and x is a variable.
($\forall$ is defined in terms of $\neg$ and $\exists$: $\forall x.(\varphi) \equiv \neg\exists x.(\neg\varphi)$.)

Examples:

((Brother(carlos,graciela) $\wedge$ Woman(graciela))$\rightarrow$ Sister(graciela,carlos))

Syntax of FOL: Complex Formulas

Complex formulas are built up from atomic formulas using the boolean connectives and the quantifiers:

$$FORMS := ATOM \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \exists x.(\varphi)$$

where φ is a formula, and x is a variable.
($\forall$ is defined in terms of $\neg$ and $\exists$: $\forall x.(\varphi) \equiv \neg\exists x.(\neg\varphi)$.)

Examples:

((Brother(carlos,graciela) $\wedge$ Woman(graciela))$\rightarrow$ Sister(graciela,carlos))

$$\forall x.(\forall y.(>(x,y) \vee ((x=y) \vee >(y,x))))$$

FOL: Semantics (Warning: Most Complex Slide Today!)

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects
  - I is the interpretation: a function asigning *meaning* to constant symbols, relation symbols and function symbols.

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects
  - I is the interpretation: a function asigning *meaning* to constant symbols, relation symbols and function symbols.
- And assingment is a function that to each variable assigns an element of D.

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects
  - I is the interpretation: a function asigning *meaning* to constant symbols, relation symbols and function symbols.
- And assingment is a function that to each variable assigns an element of D.

Formally, given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$ a model for S is $M = \langle D, I \rangle$ such that,

- for $c \in CONS$, $I(c) \in D$

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects
  - I is the interpretation: a function asigning *meaning* to constant symbols, relation symbols and function symbols.
- And assingment is a function that to each variable assigns an element of D.

Formally, given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$ a model for S is $M = \langle D, I \rangle$ such that,

- for $c \in CONS$, $I(c) \in D$
- for $R \in REL$ n-ary, $I(R) \subseteq D^n$ (an n-ary relation in D)

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects
  - I is the interpretation: a function asigning *meaning* to constant symbols, relation symbols and function symbols.
- And assingment is a function that to each variable assigns an element of D.

Formally, given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$ a model for S is $M = \langle D, I \rangle$ such that,

- for $c \in CONS$, $I(c) \in D$
- for $R \in REL$ n-ary, $I(R) \subseteq D^n$ (an n-ary relation in D)
- for $F \in FUN$ n-ary, $I(F) : D^n \mapsto D$ (an n-ary function in D)

FOL: Semantics (Warning: Most Complex Slide Today!)

- FOL formulas are true or false with respect to a model and an assignment for the model.
- A model is a pair $\langle D, I \rangle$ where
  - D is the domain: a non empty set of objects
  - I is the interpretation: a function asigning *meaning* to constant symbols, relation symbols and function symbols.
- And assingment is a function that to each variable assigns an element of D.

Formally, given a signature $S = \langle VAR, CONS, RELS, FUNS \rangle$ a model for S is $M = \langle D, I \rangle$ such that,

- for $c \in CONS$, $I(c) \in D$
- for $R \in REL$ n-ary, $I(R) \subseteq D^n$ (an n-ary relation in D)
- for $F \in FUN$ n-ary, $I(F) : D^n \mapsto D$ (an n-ary function in D)

An assignment g for M is a function $g : VAR \mapsto D$.

Satisfiability

Satisfiability

- Given a model $M = \langle D, I \rangle$ and an assignment g for M we want to define when a given FOL formula φ is true or false in M, g ($M, g \models \varphi$). We do this case by case.

Satisfiability

- Given a model $M = \langle D, I \rangle$ and an assignment g for M we want to define when a given FOL formula φ is true or false in M, g ($M, g \models \varphi$). We do this case by case.
- We first define the meaning of complex terms. The interpretation gives us the meaning for constants ($I(c)$) and funcions ($I(F)$). The assignment gives us the meaning of variables ($g(x)$) We can then define:

$$x^{I,g} = g(x)$$

Satisfiability

- Given a model $M = \langle D, I \rangle$ and an assignment g for M we want to define when a given FOL formula φ is true or false in M, g ($M, g \models \varphi$). We do this case by case.
- We first define the meaning of complex terms. The interpretation gives us the meaning for constants ($I(c)$) and funcions ($I(F)$). The assignment gives us the meaning of variables ($g(x)$) We can then define:

$$\begin{array}{rcl} x^{I,g} & = & g(x) \\ c^{I,g} & = & I(c) \end{array}$$

Satisfiability

- Given a model $M = \langle D, I \rangle$ and an assignment g for M we want to define when a given FOL formula φ is true or false in M, g ($M, g \models \varphi$). We do this case by case.
- We first define the meaning of complex terms. The interpretation gives us the meaning for constants ($I(c)$) and funcions ($I(F)$). The assignment gives us the meaning of variables ($g(x)$) We can then define:

$$\begin{aligned} x^{I,g} &= g(x) \\ c^{I,g} &= I(c) \\ (f(t_1, \ldots, t_n))^{I,g} &= I(f)(t_1^{I,g}, \ldots, t_n^{I,g}) \end{aligned}$$

Satisfiability

- Given a model $M = \langle D, I \rangle$ and an assignment g for M we want to define when a given FOL formula φ is true or false in M, g ($M, g \models \varphi$). We do this case by case.
- We first define the meaning of complex terms. The interpretation gives us the meaning for constants ($I(c)$) and funcions ($I(F)$). The assignment gives us the meaning of variables ($g(x)$) We can then define:

$$\begin{array}{rcl} x^{I,g} & = & g(x) \\ c^{I,g} & = & I(c) \\ (f(t_1, \ldots, t_n))^{I,g} & = & I(f)(t_1^{I,g}, \ldots, t_n^{I,g}) \end{array}$$

- Now we can define $M, g \models \varphi$ for arbitrary formulas...

Satisfiability

Satisfiability

- $M, g \models t_1 = t_2$ if and only if $t_1^{I,g} = t_2^{I,g}$

Satisfiability

- $M, g \models t_1 = t_2$ if and only if $t_1^{I,g} = t_2^{I,g}$
- $M, g \models R(t_1, \ldots t_n)$ if and only if $(t_1^{I,g}, \ldots, t_n^{I,g}) \in I(R)$

Satisfiability

- $M, g \models t_1 = t_2$ if and only if $t_1^{I,g} = t_2^{I,g}$
- $M, g \models R(t_1, \ldots t_n)$ if and only if $(t_1^{I,g}, \ldots, t_n^{I,g}) \in I(R)$
- $M, g \models \neg\varphi$ if and only if $M, g \not\models \varphi$

Satisfiability

- $M, g \models t_1 = t_2$ if and only if $t_1^{I,g} = t_2^{I,g}$
- $M, g \models R(t_1, \ldots t_n)$ if and only if $(t_1^{I,g}, \ldots, t_n^{I,g}) \in I(R)$
- $M, g \models \neg\varphi$ if and only if $M, g \not\models \varphi$
- $M, g \models (\varphi_1 \wedge \varphi_2)$ if and only if $M, g \models \varphi_1$ and $M, g \models \varphi_2$

Satisfiability

- $M, g \models t_1 = t_2$ if and only if $t_1^{I,g} = t_2^{I,g}$
- $M, g \models R(t_1, \ldots t_n)$ if and only if $(t_1^{I,g}, \ldots, t_n^{I,g}) \in I(R)$
- $M, g \models \neg\varphi$ if and only if $M, g \not\models \varphi$
- $M, g \models (\varphi_1 \wedge \varphi_2)$ if and only if $M, g \models \varphi_1$ and $M, g \models \varphi_2$
- $M, g \models \exists x.(\varphi)$ if and only if $M, g' \models \varphi$ for some asignment g' identical to g excepts perhaps in $g(x)$.

Exercises

- Define an adequate signature and write first order formulas for the following sentences.
  1. There is one triangle and two circles.
  2. Each object has a color: either red, blue or green.
  3. Circles are neither squares nor green.
  4. Circles are bigger than triangles.
- Give a formal definition of a model $\mathcal{M}$ s.t.:
  - $\mathcal{M}$ is a proper model for the signature used in the formulas above.
  - All the formulas above are true in $\mathcal{M}$.