Mathematical Logics

14. Practical Class: First Order Logics

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

November 14, 2013

1. Introduction
   - Well formed formulas
   - Free and bounded variables
2. FOL Formalization
   - Simple Sentences
   - FOL Interpretation
   - Formalizing Problems
     - Graph Coloring Problem
     - Data Bases

FOL Syntax

Alphabet and formation rules

- Logical symbols:
  $\perp, \wedge, \vee, \rightarrow, \neg, \forall, \exists, =$
- Non Logical symbols:
  a set $c_1, .., c_n$ of constants
  a set $f_1, .., f_m$ of functional symbols
  a set $P_1, .., P_m$ of relational symbols
- Terms T:
  $T := c_i | x_i | f_i(T, .., T)$
- Well formed formulas W:
  $$W := T = T | P_i(T, ..T) | \perp | W \wedge W | W \vee W | \\ W \rightarrow W | \neg W | \forall x.W | \exists x.W$$

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;
- $p(g(b))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;
- $p(g(b))$;
- $\neg r(x, a)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;
- $p(g(b))$;
- $\neg r(x, a)$;
- $q(x, p(a), b)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;
- $p(g(b))$;
- $\neg r(x, a)$;
- $q(x, p(a), b)$;
- $p(g(f(a), g(x, f(x))))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;
- $p(g(b))$;
- $\neg r(x, a)$;
- $q(x, p(a), b)$;
- $p(g(f(a), g(x, f(x))))$;
- $q(f(a), f(f(x)), f(g(f(z), g(a, b))))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $q(a)$;
- $p(y)$;
- $p(g(b))$;
- $\neg r(x, a)$;
- $q(x, p(a), b)$;
- $p(g(f(a), g(x, f(x))))$;
- $q(f(a), f(f(x)), f(g(f(z), g(a, b))))$;
- $r(a, r(a, a))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;
- $\forall x.\neg p(x)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;
- $\forall x. \neg p(x)$;
- $\neg r(p(a), x)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;
- $\forall x.\neg p(x)$;
- $\neg r(p(a), x)$;
- $\exists a.r(a, a)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;
- $\forall x.\neg p(x)$;
- $\neg r(p(a), x)$;
- $\exists a.r(a, a)$;
- $\exists x.q(x, f(x), b) \rightarrow \forall x.r(a, x)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;
- $\forall x.\neg p(x)$;
- $\neg r(p(a), x)$;
- $\exists a.r(a, a)$;
- $\exists x.q(x, f(x), b) \rightarrow \forall x.r(a, x)$;
- $\exists x.p(r(a, x))$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Say whether the following strings of symbols are well formed formulas or terms:

- $r(a, g(a, a))$;
- $g(a, g(a, a))$;
- $\forall x.\neg p(x)$;
- $\neg r(p(a), x)$;
- $\exists a.r(a, a)$;
- $\exists x.q(x, f(x), b) \rightarrow \forall x.r(a, x)$;
- $\exists x.p(r(a, x))$;
- $\forall r(x, a)$;

FOL Syntax

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Exercises

Say whether the following strings of symbols are well formed formulas or terms:

- $a \rightarrow p(b)$;
- $r(x, b) \rightarrow \exists y.q(y, y, y)$;
- $r(x, b) \vee \neg\exists y.g(y, b)$;
- $\neg y \vee p(y)$;
- $\neg\neg p(a)$;
- $\neg\forall x.\neg p(x)$;
- $\forall x \exists y.(r(x, y) \rightarrow r(y, x))$;
- $\forall x \exists y.(r(x, y) \rightarrow (r(y, x) \vee (f(a) = g(a, x))))$;

Free variables

A free occurrence of a variable x is an occurrence of x which is not bounded by a $\forall x$ or $\exists x$ quantifier.

Free variables

A free occurrence of a variable x is an occurrence of x which is not bounded by a $\forall x$ or $\exists x$ quantifier.

A variable x is free in a formula ϕ (denoted by $\phi(x)$) if there is at least a free occurrence of x in ϕ.

Free variables

A free occurrence of a variable x is an occurrence of x which is not bounded by a $\forall x$ or $\exists x$ quantifier.

A variable x is free in a formula ϕ (denoted by $\phi(x)$) if there is at least a free occurrence of x in ϕ.

A variable x is bounded in a formula ϕ if it is not free.

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Find free and bounded variables in the following formulas:

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Find free and bounded variables in the following formulas:

- $p(x) \wedge \neg r(y, a)$

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Find free and bounded variables in the following formulas:

- $p(x) \wedge \neg r(y, a)$
- $\exists x . r(x, y)$

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Find free and bounded variables in the following formulas:

- $p(x) \wedge \neg r(y, a)$
- $\exists x.r(x, y)$
- $\forall x.p(x) \rightarrow \exists y.\neg q(f(x), y, f(y))$

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Find free and bounded variables in the following formulas:

- $p(x) \wedge \neg r(y, a)$
- $\exists x.r(x, y)$
- $\forall x.p(x) \rightarrow \exists y.\neg q(f(x), y, f(y))$
- $\forall x \exists y.r(x, f(y))$

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Examples

Find free and bounded variables in the following formulas:

- $p(x) \wedge \neg r(y, a)$
- $\exists x.r(x, y)$
- $\forall x.p(x) \rightarrow \exists y.\neg q(f(x), y, f(y))$
- $\forall x \exists y.r(x, f(y))$
- $\forall x \exists y.r(x, f(y)) \rightarrow r(x, y)$

Free variables

Non Logical symbols

constants a, b; functions f^1, g^2; predicates p^1, r^2, q^3.

Exercises

Find free and bounded variables in the following formulas:

- $\forall x.(p(x) \rightarrow \exists y.\neg q(f(x), y, f(y)))$
- $\forall x(\exists y.r(x, f(y)) \rightarrow r(x, y))$
- $\forall z.(p(z) \rightarrow \exists y.(\exists x.q(x, y, z) \vee q(z, y, x)))$
- $\forall z \exists u \exists y.(q(z, u, g(u, y)) \vee r(u, g(z, u)))$
- $\forall z \exists x \exists y(q(z, u, g(u, y)) \vee r(u, g(z, u)))$

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$ x free

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$ x free
- $\exists x.(Sum(x, 3) = 12)$

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$ x free
- $\exists x.(Sum(x, 3) = 12)$ no free variables

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$ x free
- $\exists x.(Sum(x, 3) = 12)$ no free variables
- $\exists x.(Sum(x, y) = 12)$

Free variables

Intuitively..

Free variables represents individuals which must be instantiated to make the formula a meaningful proposition.

- $Friends(Bob, y)$ y free
- $\forall y.Friends(Bob, y)$ no free variables
- $Sum(x, 3) = 12$ x free
- $\exists x.(Sum(x, 3) = 12)$ no free variables
- $\exists x.(Sum(x, y) = 12)$ y free

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."
- $\forall x.bought(Frank, x) \rightarrow \forall x.bought(Susan, x)$

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."
- $\forall x.bought(Frank, x) \rightarrow \forall x.bought(Susan, x)$
  "If Frank bought everything, so did Susan."

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."
- $\forall x.bought(Frank, x) \rightarrow \forall x.bought(Susan, x)$
  "If Frank bought everything, so did Susan."
- $\forall x \exists y.bought(x, y)$

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."
- $\forall x.bought(Frank, x) \rightarrow \forall x.bought(Susan, x)$
  "If Frank bought everything, so did Susan."
- $\forall x \exists y.bought(x, y)$
  "Everyone bought something."

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."
- $\forall x.bought(Frank, x) \rightarrow \forall x.bought(Susan, x)$
  "If Frank bought everything, so did Susan."
- $\forall x \exists y.bought(x, y)$
  "Everyone bought something."
- $\exists x \forall y.bought(x, y)$

FOL: Intuitive Meaning

Examples

- $bought(Frank, dvd)$
  "Frank bought a dvd."
- $\exists x.bought(Frank, x)$
  "Frank bought something."
- $\forall x.(bought(Frank, x) \rightarrow bought(Susan, x))$
  "Susan bought everything that Frank bought."
- $\forall x.bought(Frank, x) \rightarrow \forall x.bought(Susan, x)$
  "If Frank bought everything, so did Susan."
- $\forall x \exists y.bought(x, y)$
  "Everyone bought something."
- $\exists x \forall y.bought(x, y)$
  "Someone bought everything."

FOL: Intuitive Meaning

Example

Which of the following formulas is a formalization of the sentence:

"There is a computer which is not used by any student"

FOL: Intuitive Meaning

Example

Which of the following formulas is a formalization of the sentence:

"There is a computer which is not used by any student"

- $\exists x.(Computer(x) \wedge \forall y.(\neg Student(y) \wedge \neg Uses(y, x)))$
- $\exists x.(Computer(x) \rightarrow \forall y.(Student(y) \rightarrow \neg Uses(y, x)))$
- $\exists x.(Computer(x) \wedge \forall y.(Student(y) \rightarrow \neg Uses(y, x)))$

Formalizing English Sentences in FOL

Common mistake..

- "Everyone studying at DIT is smart."
  $\forall x.(At(x, DIT) \rightarrow Smart(x))$

Formalizing English Sentences in FOL

Common mistake..

- "Everyone studying at DIT is smart."

  $\forall x.(At(x, DIT) \rightarrow Smart(x))$

  and NOT

  $\forall x.(At(x, DIT) \wedge Smart(x))$

Formalizing English Sentences in FOL

Common mistake..

- "Everyone studying at DIT is smart."

  $\forall x.(At(x, DIT) \rightarrow Smart(x))$

  and NOT

  $\forall x.(At(x, DIT) \wedge Smart(x))$

  "Everyone studies at DIT and everyone is smart"

Formalizing English Sentences in FOL

Common mistake..

- "Everyone studying at DIT is smart."
  $\forall x.(At(x, DIT) \rightarrow Smart(x))$
  and NOT
  $\forall x.(At(x, DIT) \wedge Smart(x))$
  "Everyone studies at DIT and everyone is smart"
- "Someone studying at DIT is smart."
  $\exists x.(At(x, DIT) \wedge Smart(x))$

Formalizing English Sentences in FOL

Common mistake..

- "Everyone studying at DIT is smart."

  $\forall x.(At(x, DIT) \rightarrow Smart(x))$

  and NOT

  $\forall x.(At(x, DIT) \wedge Smart(x))$

  "Everyone studies at DIT and everyone is smart"

- "Someone studying at DIT is smart."

  $\exists x.(At(x, DIT) \wedge Smart(x))$

  and NOT

  $\exists x.(At(x, DIT) \rightarrow Smart(x))$

Formalizing English Sentences in FOL

Common mistake..

- "Everyone studying at DIT is smart."
  $\forall x.(At(x, DIT) \rightarrow Smart(x))$
  and NOT
  $\forall x.(At(x, DIT) \wedge Smart(x))$
  "Everyone studies at DIT and everyone is smart"
- "Someone studying at DIT is smart."
  $\exists x.(At(x, DIT) \wedge Smart(x))$
  and NOT
  $\exists x.(At(x, DIT) \rightarrow Smart(x))$
  which is true if there is anyone who is not at DIT.

Formalizing English Sentences in FOL

Common mistake.. (2)

Quantifiers of different type do NOT commute

Formalizing English Sentences in FOL

Common mistake.. (2)

Quantifiers of different type do NOT commute

$\exists x \forall y . \phi$ is not the same as $\forall y \exists x . \phi$

Formalizing English Sentences in FOL

Common mistake.. (2)

Quantifiers of different type do NOT commute

$\exists x \forall y.\phi$ is not the same as $\forall y \exists x.\phi$

Example

- $\exists x \forall y.Loves(x, y)$
  "There is a person who loves everyone in the world."

Formalizing English Sentences in FOL

Common mistake.. (2)

Quantifiers of different type do NOT commute

$\exists x \forall y.\phi$ is not the same as $\forall y \exists x.\phi$

Example

- $\exists x \forall y.Loves(x, y)$
  "There is a person who loves everyone in the world."
- $\forall y \exists x.Loves(x, y)$
  "Everyone in the world is loved by at least one person."

Formalizing English Sentences in FOL

Examples

- All Students are smart.

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$
- There exists a smart student

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$
- There exists a smart student
  $\exists x.(Student(x) \wedge Smart(x))$

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$
- There exists a smart student
  $\exists x.(Student(x) \wedge Smart(x))$
- Every student loves some student

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$
- There exists a smart student
  $\exists x.(Student(x) \wedge Smart(x))$
- Every student loves some student
  $\forall x.(Student(x) \rightarrow \exists y.(Student(y) \wedge Loves(x, y)))$

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$
- There exists a smart student
  $\exists x.(Student(x) \wedge Smart(x))$
- Every student loves some student
  $\forall x.(Student(x) \rightarrow \exists y.(Student(y) \wedge Loves(x, y)))$
- Every student loves some other student.

Formalizing English Sentences in FOL

Examples

- All Students are smart.
  $\forall x.(Student(x) \rightarrow Smart(x))$
- There exists a student.
  $\exists x.Student(x)$
- There exists a smart student
  $\exists x.(Student(x) \wedge Smart(x))$
- Every student loves some student
  $\forall x.(Student(x) \rightarrow \exists y.(Student(y) \wedge Loves(x, y)))$
- Every student loves some other student.
  $\forall x.(Student(x) \rightarrow \exists y.(Student(y) \wedge \neg(x = y) \wedge Loves(x, y)))$

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$
- Bill takes either Analysis or Geometry (but not both).

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$
- Bill takes either Analysis or Geometry (but not both).
  $Takes(Bill, Analysis) \leftrightarrow \neg Takes(Bill, Geometry)$

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$
- Bill takes either Analysis or Geometry (but not both).
  $Takes(Bill, Analysis) \leftrightarrow \neg Takes(Bill, Geometry)$
- Bill takes Analysis and Geometry.

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$
- Bill takes either Analysis or Geometry (but not both).
  $Takes(Bill, Analysis) \leftrightarrow \neg Takes(Bill, Geometry)$
- Bill takes Analysis and Geometry.
  $Takes(Bill, Analysis) \wedge Takes(Bill, Geometry)$

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$
- Bill takes either Analysis or Geometry (but not both).
  $Takes(Bill, Analysis) \leftrightarrow \neg Takes(Bill, Geometry)$
- Bill takes Analysis and Geometry.
  $Takes(Bill, Analysis) \wedge Takes(Bill, Geometry)$
- Bill doesn't take Analysis.

Formalizing English Sentences in FOL

Examples

- There is a student who is loved by every other student.
  $\exists x.(Student(x) \wedge \forall y.(Student(y) \wedge \neg(x = y) \rightarrow Loves(y, x)))$
- Bill is a student.
  $Student(Bill)$
- Bill takes either Analysis or Geometry (but not both).
  $Takes(Bill, Analysis) \leftrightarrow \neg Takes(Bill, Geometry)$
- Bill takes Analysis and Geometry.
  $Takes(Bill, Analysis) \wedge Takes(Bill, Geometry)$
- Bill doesn't take Analysis.
  $\neg Takes(Bill, Analysis)$

Formalizing English Sentences in FOL

Examples

- No students love Bill.

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg \exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$
- Bill has at most one sister.

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$
- Bill has at most one sister.
  $\forall x \forall y.(SisterOf(x, Bill) \wedge SisterOf(y, Bill) \rightarrow x = y)$

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$
- Bill has at most one sister.
  $\forall x \forall y.(SisterOf(x, Bill) \wedge SisterOf(y, Bill) \rightarrow x = y)$
- Bill has (exactly) one sister.

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$
- Bill has at most one sister.
  $\forall x \forall y.(SisterOf(x, Bill) \wedge SisterOf(y, Bill) \rightarrow x = y)$
- Bill has (exactly) one sister.
  $\exists x.(SisterOf(x, Bill) \wedge \forall y.(SisterOf(y, Bill) \rightarrow x = y))$

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$
- Bill has at most one sister.
  $\forall x \forall y.(SisterOf(x, Bill) \wedge SisterOf(y, Bill) \rightarrow x = y)$
- Bill has (exactly) one sister.
  $\exists x.(SisterOf(x, Bill) \wedge \forall y.(SisterOf(y, Bill) \rightarrow x = y))$
- Bill has at least two sisters.

Formalizing English Sentences in FOL

Examples

- No students love Bill.
  $\neg\exists x.(Student(x) \wedge Loves(x, Bill))$
- Bill has at least one sister.
  $\exists x.SisterOf(x, Bill)$
- Bill has no sister.
  $\neg\exists x.SisterOf(x, Bill)$
- Bill has at most one sister.
  $\forall x \forall y.(SisterOf(x, Bill) \wedge SisterOf(y, Bill) \rightarrow x = y)$
- Bill has (exactly) one sister.
  $\exists x.(SisterOf(x, Bill) \wedge \forall y.(SisterOf(y, Bill) \rightarrow x = y))$
- Bill has at least two sisters.
  $\exists x \exists y.(SisterOf(x, Bill) \wedge SisterOf(y, Bill) \wedge \neg(x = y))$

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$
- Only one student failed Geometry.

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$
- Only one student failed Geometry.
  $\exists x.(Student(x) \wedge Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \rightarrow x = y))$

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$
- Only one student failed Geometry.
  $\exists x.(Student(x) \wedge Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \rightarrow x = y))$
- No student failed Geometry but at least one student failed Analysis.

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$
- Only one student failed Geometry.
  $\exists x.(Student(x) \wedge Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \rightarrow x = y))$
- No student failed Geometry but at least one student failed Analysis.
  $\neg\exists x.(Student(x) \wedge Failed(x, Geometry)) \wedge \exists x.(Student(x) \wedge Failed(x, Analysis))$

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$
- Only one student failed Geometry.
  $\exists x.(Student(x) \wedge Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \rightarrow x = y))$
- No student failed Geometry but at least one student failed Analysis.
  $\neg\exists x.(Student(x) \wedge Failed(x, Geometry)) \wedge \exists x.(Student(x) \wedge Failed(x, Analysis))$
- Every student who takes Analysis also takes Geometry.

Formalizing English Sentences in FOL

Examples

- Every student takes at least one course.
  $\forall x.(Student(x) \rightarrow \exists y.(Course(y) \wedge Takes(x, y)))$
- Only one student failed Geometry.
  $\exists x.(Student(x) \wedge Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \rightarrow x = y))$
- No student failed Geometry but at least one student failed Analysis.
  $\neg\exists x.(Student(x) \wedge Failed(x, Geometry)) \wedge \exists x.(Student(x) \wedge Failed(x, Analysis))$
- Every student who takes Analysis also takes Geometry.
  $\forall x.(Student(x) \wedge Takes(x, Analysis) \rightarrow Takes(x, Geometry))$

Formalizing English Sentences in FOL

Exercises

Define an appropriate language and formalize the following sentences in FOL:

- someone likes Mary.
- nobody likes Mary.
- nobody loves Bob but Bob loves Mary.
- if David loves someone, then he loves Mary.
- if someone loves David, then he (someone) loves also Mary.
- everybody loves David or Mary.

Formalizing English Sentences in FOL

Exercises

Define an appropriate language and formalize the following sentences in FOL:

- there is at least one person who loves Mary.
- there is at most one person who loves Mary.
- there is exactly one person who loves Mary.
- there are exactly two persons who love Mary.
- if Bob loves everyone that Mary loves, and Bob loves David, then Mary doesn't love David.
- Only Mary loves Bob.

Formalizing English Sentences in FOL

Example

Define an appropriate language and formalize the following sentences in FOL:

- "A is above C, D is on E and above F."
- "A is green while C is not."
- "Everything is on something."
- "Everything that has nothing on it, is free."
- "Everything that is green is free."
- "There is something that is red and is not free."
- "Everything that is not green and is above B, is red."

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "A is above C, D is above F and on E."

  $\phi_1 : Above(A, C) \wedge Above(E, F) \wedge On(D, E)$

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "A is above C, D is above F and on E."
  $\phi_1 : Above(A, C) \wedge Above(E, F) \wedge On(D, E)$
- "A is green while C is not."
  $\phi_2 : Green(A) \wedge \neg Green(C)$

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "A is above C, D is above F and on E."

  $\phi_1 : Above(A, C) \wedge Above(E, F) \wedge On(D, E)$

- "A is green while C is not."

  $\phi_2 : Green(A) \wedge \neg Green(C)$

- "Everything is on something."

  $\phi_3 : \forall x \exists y . On(x, y)$

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "A is above C, D is above F and on E."
  $\phi_1 : Above(A, C) \wedge Above(E, F) \wedge On(D, E)$
- "A is green while C is not."
  $\phi_2 : Green(A) \wedge \neg Green(C)$
- "Everything is on something."
  $\phi_3 : \forall x \exists y. On(x, y)$
- "Everything that has nothing on it, is free."
  $\phi_4 : \forall x. (\neg \exists y. On(y, x) \rightarrow Free(x))$

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "Everything that is green is free."
  $\phi_5 : \forall x.(Green(x) \rightarrow Free(x))$

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "Everything that is green is free."
  $\phi_5 : \forall x.(Green(x) \rightarrow Free(x))$
- "There is something that is red and is not free."
  $\phi_6 : \exists x.(Red(x) \wedge \neg Free(x))$

Formalizing English Sentences in FOL

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Example

- "Everything that is green is free."
  $\phi_5 : \forall x.(Green(x) \rightarrow Free(x))$
- "There is something that is red and is not free."
  $\phi_6 : \exists x.(Red(x) \wedge \neg Free(x))$
- "Everything that is not green and is above B, is red."
  $\phi_7 : \forall x.(\neg Green(x) \wedge Above(x, B) \rightarrow Red(x))$

An interpretation $\mathcal{I}_1$ in the Blocks World

Non Logical symbols

Constants: A, B, C, D, E, F;

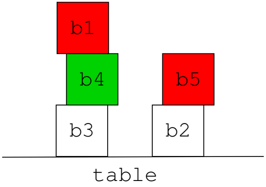
Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

An interpretation $\mathcal{I}_1$ in the Blocks World

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

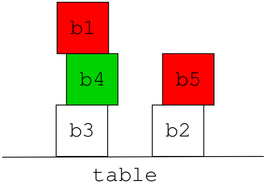


An interpretation $\mathcal{I}_1$ in the Blocks World

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.



Interpretation $\mathcal{I}_1$

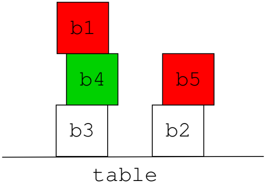
- $\mathcal{I}_1(A) = b_1$, $\mathcal{I}_1(B) = b_2$, $\mathcal{I}_1(C) = b_3$, $\mathcal{I}_1(D) = b_4$, $\mathcal{I}_1(E) = b_5$, $\mathcal{I}_1(F) = table$

An interpretation $\mathcal{I}_1$ in the Blocks World

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.



Interpretation $\mathcal{I}_1$

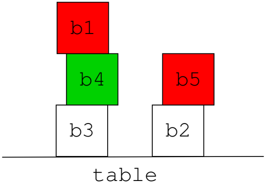
- $\mathcal{I}_1(A) = b_1$, $\mathcal{I}_1(B) = b_2$, $\mathcal{I}_1(C) = b_3$, $\mathcal{I}_1(D) = b_4$, $\mathcal{I}_1(E) = b_5$, $\mathcal{I}_1(F) = table$
- $\mathcal{I}_1(On) = \{\langle b_1, b_4\rangle, \langle b_4, b_3\rangle, \langle b_3, table\rangle, \langle b_5, b_2\rangle, \langle b_2, table\rangle\}$

An interpretation $\mathcal{I}_1$ in the Blocks World

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: On^2, $Above^2$, $Free^1$, Red^1, $Green^1$.



Interpretation $\mathcal{I}_1$

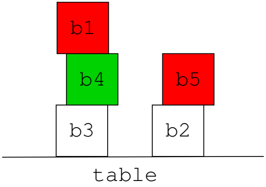
- $\mathcal{I}_1(A) = b_1$, $\mathcal{I}_1(B) = b_2$, $\mathcal{I}_1(C) = b_3$, $\mathcal{I}_1(D) = b_4$, $\mathcal{I}_1(E) = b_5$, $\mathcal{I}_1(F) = table$
- $\mathcal{I}_1(On) = \{\langle b_1, b_4\rangle, \langle b_4, b_3\rangle, \langle b_3, table\rangle, \langle b_5, b_2\rangle, \langle b_2, table\rangle\}$
- $\mathcal{I}_1(Above) = \{\langle b_1, b_4\rangle, \langle b_1, b_3\rangle, \langle b_1, table\rangle, \langle b_4, b_3\rangle, \langle b_4, table\rangle, \langle b_3, table\rangle, \langle b_5, b_2\rangle, \langle b_5, table\rangle, \langle b_2, table\rangle\}$

An interpretation $\mathcal{I}_1$ in the Blocks World

Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: On^2, $Above^2$, $Free^1$, Red^1, $Green^1$.



Interpretation $\mathcal{I}_1$

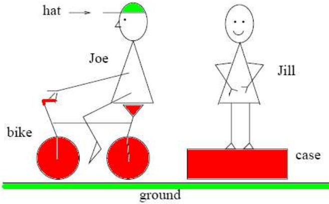
- $\mathcal{I}_1(A) = b_1$, $\mathcal{I}_1(B) = b_2$, $\mathcal{I}_1(C) = b_3$, $\mathcal{I}_1(D) = b_4$, $\mathcal{I}_1(E) = b_5$, $\mathcal{I}_1(F) = table$
- $\mathcal{I}_1(On) = \{\langle b_1, b_4\rangle, \langle b_4, b_3\rangle, \langle b_3, table\rangle, \langle b_5, b_2\rangle, \langle b_2, table\rangle\}$
- $\mathcal{I}_1(Above) = \{\langle b_1, b_4\rangle, \langle b_1, b_3\rangle, \langle b_1, table\rangle, \langle b_4, b_3\rangle, \langle b_4, table\rangle, \langle b_3, table\rangle, \langle b_5, b_2\rangle, \langle b_5, table\rangle, \langle b_2, table\rangle\}$
- $\mathcal{I}_1(Free) = \{\langle b_1\rangle, \langle b_5\rangle\}$, $\mathcal{I}_1(Green) = \{\langle b_4\rangle\}$, $\mathcal{I}_1(Red) = \{\langle b_1\rangle, \langle b_5\rangle\}$

A different interpretation $\mathcal{I}_2$

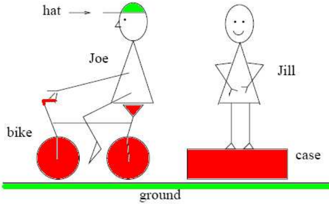
Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.



A different interpretation $\mathcal{I}_2$



Non Logical symbols

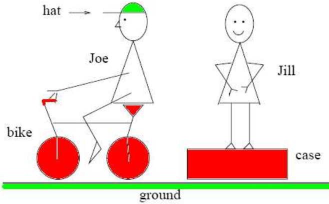
Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Interpretation $\mathcal{I}_2$

- $\mathcal{I}_2(A) = hat$, $\mathcal{I}_2(B) = Joe$, $\mathcal{I}_2(C) = bike$, $\mathcal{I}_2(D) = Jill$, $\mathcal{I}_2(E) = case$, $\mathcal{I}_2(F) = ground$

A different interpretation $\mathcal{I}_2$



Non Logical symbols

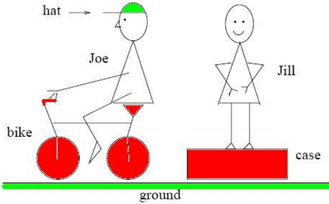
Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Interpretation $\mathcal{I}_2$

- $\mathcal{I}_2(A) = hat$, $\mathcal{I}_2(B) = Joe$, $\mathcal{I}_2(C) = bike$, $\mathcal{I}_2(D) = Jill$, $\mathcal{I}_2(E) = case$, $\mathcal{I}_2(F) = ground$
- $\mathcal{I}_2(On) = \{\langle hat, Joe\rangle, \langle Joe, bike\rangle, \langle bike, ground\rangle, \langle Jill, case\rangle, \langle case, ground\rangle\}$

A different interpretation $\mathcal{I}_2$



Non Logical symbols

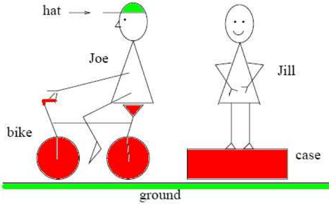
Constants: A, B, C, D, E, F;

Predicates: On^2, $Above^2$, $Free^1$, Red^1, $Green^1$.

Interpretation $\mathcal{I}_2$

- $\mathcal{I}_2(A) = hat$, $\mathcal{I}_2(B) = Joe$, $\mathcal{I}_2(C) = bike$, $\mathcal{I}_2(D) = Jill$, $\mathcal{I}_2(E) = case$, $\mathcal{I}_2(F) = ground$
- $\mathcal{I}_2(On) = \{\langle hat, Joe\rangle, \langle Joe, bike\rangle, \langle bike, ground\rangle, \langle Jill, case\rangle, \langle case, ground\rangle\}$
- $\mathcal{I}_2(Above) = \{\langle hat, Joe\rangle, \langle hat, bike\rangle, \langle hat, ground\rangle, \langle Joe, bike\rangle, \langle Joe, ground\rangle, \langle bike, ground\rangle, \langle Jill, case\rangle, \langle Jill, ground\rangle, \langle case, ground\rangle\}$

A different interpretation $\mathcal{I}_2$



Non Logical symbols

Constants: A, B, C, D, E, F;

Predicates: $On^2, Above^2, Free^1, Red^1, Green^1$.

Interpretation $\mathcal{I}_2$

- $\mathcal{I}_2(A) = hat$, $\mathcal{I}_2(B) = Joe$, $\mathcal{I}_2(C) = bike$, $\mathcal{I}_2(D) = Jill$, $\mathcal{I}_2(E) = case$, $\mathcal{I}_2(F) = ground$
- $\mathcal{I}_2(On) = \{\langle hat, Joe\rangle, \langle Joe, bike\rangle, \langle bike, ground\rangle, \langle Jill, case\rangle, \langle case, ground\rangle\}$
- $\mathcal{I}_2(Above) = \{\langle hat, Joe\rangle, \langle hat, bike\rangle, \langle hat, ground\rangle, \langle Joe, bike\rangle, \langle Joe, ground\rangle, \langle bike, ground\rangle, \langle Jill, case\rangle, \langle Jill, ground\rangle, \langle case, ground\rangle\}$
- $\mathcal{I}_2(Free) = \{\langle hat\rangle, \langle Jill\rangle\}$, $\mathcal{I}_2(Green) = \{\langle hat\rangle, \langle ground\rangle\}$, $\mathcal{I}_2(Red) = \{\langle bike\rangle, \langle case\rangle\}$

FOL Satisfiability

Example

For each of the following formulas, decide whether they are satisfied by $\mathcal{I}_1$ and/or $\mathcal{I}_2$:

- $\phi_1 : Above(A, C) \wedge Above(E, F) \wedge On(D, E)$
- $\phi_2 : Green(A) \wedge \neg Green(C)$
- $\phi_3 : \forall x \exists y . On(x, y)$
- $\phi_4 : \forall x . (\neg \exists y . On(y, x) \rightarrow Free(x))$
- $\phi_5 : \forall x . (Green(x) \rightarrow Free(x))$
- $\phi_6 : \exists x . (Red(x) \wedge \neg Free(x))$
- $\phi_7 : \forall x . (\neg Green(x) \wedge Above(x, B) \rightarrow Red(x))$

FOL Satisfiability

Example

For each of the following formulas, decide whether they are satisfied by $\mathcal{I}_1$ and/or $\mathcal{I}_2$:

- $\phi_1 : Above(A, C) \wedge Above(E, F) \wedge On(D, E)$
- $\phi_2 : Green(A) \wedge \neg Green(C)$
- $\phi_3 : \forall x \exists y . On(x, y)$
- $\phi_4 : \forall x . (\neg \exists y . On(y, x) \rightarrow Free(x))$
- $\phi_5 : \forall x . (Green(x) \rightarrow Free(x))$
- $\phi_6 : \exists x . (Red(x) \wedge \neg Free(x))$
- $\phi_7 : \forall x . (\neg Green(x) \wedge Above(x, B) \rightarrow Red(x))$

Sol.

- $\mathcal{I}_1 \models \neg\phi_1 \wedge \neg\phi_2 \wedge \neg\phi_3 \wedge \phi_4 \wedge \neg\phi_5 \wedge \neg\phi_6 \wedge \phi_7$
- $\mathcal{I}_2 \models \phi_1 \wedge \phi_2 \wedge \neg\phi_3 \wedge \phi_4 \wedge \neg\phi_5 \wedge \phi_6 \wedge \phi_7$

FOL Satisfiability

Example

Consider the following sentences:

- (1) All actors and journalists invited to the party are late.
- (2) There is at least a person who is on time.
- (3) There is at least an invited person who is neither a journalist nor an actor.

Formalize the sentences and prove that (3) is not a logical consequence of (1) and (2)

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$
- There is at least a person who is on time.

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$
- There is at least a person who is on time.
  (2) $\exists x.\neg l(x)$

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$
- There is at least a person who is on time.
  (2) $\exists x. \neg l(x)$
- There is at least an invited person who is neither a journalist nor an actor.

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$
- There is at least a person who is on time.
  (2) $\exists x.\neg l(x)$
- There is at least an invited person who is neither a journalist nor an actor.
  (3) $\exists x.(i(x) \wedge \neg a(x) \wedge \neg j(x))$

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$
- There is at least a person who is on time.
  (2) $\exists x.\neg l(x)$
- There is at least an invited person who is neither a journalist nor an actor.
  (3) $\exists x.(i(x) \wedge \neg a(x) \wedge \neg j(x))$

It's sufficient to find an interpretation $\mathcal{I}$ for which the logical consequence does not hold:

FOL Satisfiability

Example

Consider the following sentences:

- All actors and journalists invited to the party are late.
  (1) $\forall x.((a(x) \vee j(x)) \wedge i(x) \rightarrow l(x))$
- There is at least a person who is on time.
  (2) $\exists x.\neg l(x)$
- There is at least an invited person who is neither a journalist nor an actor.
  (3) $\exists x.(i(x) \wedge \neg a(x) \wedge \neg j(x))$

It's sufficient to find an interpretation $\mathcal{I}$ for which the logical consequence does not hold:

	l(x)	a(x)	j(x)	i(x)
Bob	F	T	F	F
Tom	T	T	F	T
Mary	T	F	T	T

FOL Satisfiability

Exercise

Let $\Delta = \{1, 3, 5, 15\}$ and $\mathcal{I}$ be an interpretation on Δ interpreting the predicate symbols E^1 as 'being even', M^2 as 'being a multiple of' and L^2 as 'being less then', and s.t. $\mathcal{I}(a) = 1, \mathcal{I}(b) = 3, \mathcal{I}(c) = 5, \mathcal{I}(d) = 15$.
Determine whether $\mathcal{I}$ satisfies the following formulas:

$\exists y.E(y)$ $\quad \forall x.\neg E(x)$ $\quad \forall x.M(x, a)$ $\quad \forall x.M(x, b)$ $\quad \exists x.M(x, d)$

$\exists x.L(x, a)$ $\quad \forall x.(E(x) \rightarrow M(x, a))$ $\quad \forall x \exists y.L(x, y)$ $\quad \forall x \exists y.M(x, y)$

$\forall x.(M(x, b) \rightarrow L(x, c))$ $\quad \forall x \forall y.(L(x, y) \rightarrow \neg L(y, x))$

$\forall x.(M(x, c) \vee L(x, c))$

Graph Coloring Problem

Provide a propositional language and a set of axioms that formalize the graph coloring problem of a graph with at most n nodes, with connection degree $\leq m$, and with less then $k+1$ colors.

- node degree: number of adjacent nodes
- connection degree of a graph: max among all the degree of its nodes
- Graph coloring problem: given a non-oriented graph, associate a color to each of its nodes in such a way that no pair of adjacent nodes have the same color.

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x
- A unary predicate node, where node(x) means that x is a node

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x
- A unary predicate node, where node(x) means that x is a node
- A binary predicate edge, where edge(x, y) means that x is connected to y

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x
- A unary predicate node, where node(x) means that x is a node
- A binary predicate edge, where edge(x, y) means that x is connected to y

FOL Axioms

Two connected node are not equally colored:

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x
- A unary predicate node, where node(x) means that x is a node
- A binary predicate edge, where edge(x, y) means that x is connected to y

FOL Axioms

Two connected node are not equally colored:

$$\forall x \forall y.(\text{edge}(x, y) \rightarrow (\text{color}(x) \neq \text{color}(y)) \quad (1)$$

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x
- A unary predicate node, where node(x) means that x is a node
- A binary predicate edge, where edge(x, y) means that x is connected to y

FOL Axioms

Two connected node are not equally colored:

$$\forall x \forall y.(\text{edge}(x, y) \rightarrow (\text{color}(x) \neq \text{color}(y)) \quad (1)$$

A node does not have more than k connected nodes:

Graph Coloring: FOL Formalization

FOL Language

- A unary function color, where color(x) is the color associated to the node x
- A unary predicate node, where node(x) means that x is a node
- A binary predicate edge, where edge(x, y) means that x is connected to y

FOL Axioms

Two connected node are not equally colored:

$$\forall x \forall y.(\text{edge}(x, y) \rightarrow (\text{color}(x) \neq \text{color}(y))) \tag{1}$$

A node does not have more than k connected nodes:

$$\forall x \forall x_1 \ldots \forall x_{k+1}. \left(\bigwedge_{h=1}^{k+1} \text{edge}(x, x_h) \rightarrow \bigvee_{i,j=1, j\neq i}^{k+1} x_i = x_j \right) \tag{2}$$

Graph Coloring: Propositional Formalization

Prop. Language

- For each $1 \leq i \leq n$ and $1 \leq c \leq k$, color_{ic} is a proposition, which intuitively means that *"the i-th node has the c color"*
- For each $1 \leq i \neq j \leq n$, edge_{ij} is a proposition, which intuitively means that *"the i-th node is connected with the j-th node"*.

Prop. Axioms

- for each $1 \leq i \leq n$, $\bigvee_{c=1}^{k} \text{color}_{ic}$
  "each node has at least one color"
- for each $1 \leq i \leq n$ and $1 \leq c, c' \leq k$, $\text{color}_{ic} \rightarrow \neg\text{color}_{ic'}$
  "every node has at most 1 color"
- for each $1 \leq i, j \leq n$ and $1 \leq c \leq k$, $\text{edge}_{ij} \rightarrow \neg(\text{color}_{ic} \wedge \text{color}_{jc})$
  "adjacent nodes do not have the same color"
- for each $1 \leq i \leq n$, and each $J \subseteq \{1..n\}$, where $|J| = m$,
  $\bigwedge_{j \in J} \text{edge}_{ij} \rightarrow \bigwedge_{j \not\in J} \neg\text{edge}_{ij}$
  "every node has at most m connected nodes"

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ doesn't contain functional symbols (relational language), there is a strict analogy between FOL and databases.

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ doesn't contain functional symbols (relational language), there is a strict analogy between FOL and databases.

- relational symbols of $\mathcal{L}$ correspond to database schema (tables)

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ doesn't contain functional symbols (relational language), there is a strict analogy between FOL and databases.

- relational symbols of $\mathcal{L}$ correspond to database schema (tables)
- Δ corresponds to the set of values which appear in the tables

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ doesn't contain functional symbols (relational language), there is a strict analogy between FOL and databases.

- relational symbols of $\mathcal{L}$ correspond to database schema (tables)
- Δ corresponds to the set of values which appear in the tables
- the interpretation $\mathcal{I}$ corresponds to the tuples that belongs to each relation

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ doesn't contain functional symbols (relational language), there is a strict analogy between FOL and databases.

- relational symbols of $\mathcal{L}$ correspond to database schema (tables)
- Δ corresponds to the set of values which appear in the tables
- the interpretation $\mathcal{I}$ corresponds to the tuples that belongs to each relation
- formulas on $\mathcal{L}$ corresponds to queries over the database

Analogy with Databases

When the language $\mathcal{L}$ and the domain of interpretation Δ are finite, and $\mathcal{L}$ doesn't contain functional symbols (relational language), there is a strict analogy between FOL and databases.

- relational symbols of $\mathcal{L}$ correspond to database schema (tables)
- Δ corresponds to the set of values which appear in the tables
- the interpretation $\mathcal{I}$ corresponds to the tuples that belongs to each relation
- formulas on $\mathcal{L}$ corresponds to queries over the database
- interpretation of formulas of $\mathcal{L}$ corresponds to answers

Analogy with Databases

FOL	DB
friends	CREATE TABLE FRIENDS (friend1 : INTEGER friend2 : INTEGER)

Analogy with Databases

FOL	**DB**
friends	`CREATE TABLE FRIENDS (friend1 : INTEGER friend2 : INTEGER)`
$friends(x, y)$	`SELECT friend1 AS x friend2 AS y FROM FRIENDS`

Analogy with Databases

FOL	DB
friends	`CREATE TABLE FRIENDS (friend1 : INTEGER` `friend2 : INTEGER)`
$friends(x, y)$	`SELECT friend1 AS x friend2 AS y` `FROM FRIENDS`

$friends(x, x)$	`SELECT friend1 AS x`
	`FROM FRIENDS`
	`WHERE friend1 = friend2`

Analogy with Databases

FOL	DB
friends	`CREATE TABLE FRIENDS (friend1 : INTEGER` `friend2 : INTEGER)`
$friends(x, y)$	`SELECT friend1 AS x friend2 AS y` `FROM FRIENDS`

$friends(x, x)$	`SELECT friend1 AS x`
	`FROM FRIENDS`
	`WHERE friend1 = friend2`

$friends(x, y) \wedge x = y$	`SELECT friend1 AS x friend2 AS y`
	`FROM FRIENDS`
	`WHERE friend1 = friend2`

Analogy with Databases

FOL	DB
friends	`CREATE TABLE FRIENDS (friend1 : INTEGER friend2 : INTEGER)`
$friends(x, y)$	`SELECT friend1 AS x friend2 AS y FROM FRIENDS`
$friends(x, x)$	`SELECT friend1 AS x FROM FRIENDS WHERE friend1 = friend2`
$friends(x, y) \wedge x = y$	`SELECT friend1 AS x friend2 AS y FROM FRIENDS WHERE friend1 = friend2`
$\exists x.friends(x, y)$	`SELECT friend2 AS y FROM FRIENDS`

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`
2. Give `Names` of students studying in a university in `Trento`
3. Give `Names` of students living in their origin town
4. Give `(Name, University)` pairs for each student studying in `Italy`
5. Give all `Country` that have at least one university for each town.

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`
   $\exists y \exists z. Students(x, y, z, Trento)$

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`

   $\exists y \exists z. Students(x, y, z, Trento)$

2. Give `Names` of students studying in a university in `Trento`

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`
   $\exists y \exists z. Students(x, y, z, Trento)$
2. Give `Names` of students studying in a university in `Trento`
   $\exists y \exists z \exists v. (Students(x, y, z, v) \wedge Universities(y, Trento))$

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`
   $\exists y \exists z. Students(x, y, z, Trento)$
2. Give `Names` of students studying in a university in `Trento`
   $\exists y \exists z \exists v. (Students(x, y, z, v) \wedge Universities(y, Trento))$
3. Give `Names` of students living in their origin town

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

1. Give `Names` of students living in `Trento`
   $\exists y \exists z . Students(x, y, z, Trento)$
2. Give `Names` of students studying in a university in `Trento`
   $\exists y \exists z \exists v . (Students(x, y, z, v) \wedge Universities(y, Trento))$
3. Give `Names` of students living in their origin town
   $\exists y \exists z . Students(x, y, z, z)$

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

4. Give `(Name, University)` pairs for each student studying in `Italy`

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

4 Give `(Name, University)` pairs for each student studying in `Italy`

$\exists z \exists v \exists w.(Students(x, y, z, v) \wedge Universities(y, w) \wedge Town(w, Italy))$

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

4. Give `(Name, University)` pairs for each student studying in `Italy`
   $\exists z \exists v \exists w.(Students(x, y, z, v) \wedge Universities(y, w) \wedge Town(w, Italy))$
5. Give all `Country` that have at least one university for each town.

Analogy with Databases

Example

Consider the following database schema:

- `Students(Name, University, OriginT, LiveT)`
- `Universities(Name, Town)`
- `Town(Name, Country)`

Express each of the following queries in FOL formulas with free variables.

4 Give `(Name, University)` pairs for each student studying in `Italy`
$\exists z \exists v \exists w.(Students(x, y, z, v) \wedge Universities(y, w) \wedge Town(w, Italy))$

5 Give all `Country` that have at least one university for each town.
$\forall x.(Town(x, y) \rightarrow \exists z.Universities(z, x))$

Analogy with Databases

Exercise

Consider the following database schema

- `Lives(Name,Town)`
- `Works(Name,Company,Salary)`
- `Company_Location(Company,Town)`
- `Reports_To(Name,Manager)`

(you may use the abbreviations `L(N,T)`, `W(N,C,S)`, `CL(C,T)`, and `R(N,M)`). Express each of the following queries in first order formulas with free variables.

1. Give `(Name,Town)` pairs for each person working for `Fiat`.
2. Find all people who live and work in the same town.
3. Find the maximum salary of all people who work in `Trento`.
4. Find the names of all companies which are located in every city that has a branch of `Fiat`