First-Order Logic

First-Order Language

- Syntax
- Semantics

Syntax

- Alphabet
- Terms
- Formulas

Alphabet

A first-order alphabet consists of the following disjoint sets of symbols:

- A countable set of variables $\mathcal{V}$.
- For each $n \geq 0$, a set of n-ary function symbols $\mathcal{F}^n$. Elements of $\mathcal{F}^0$ are called constants.
- For each $n \geq 0$, a set of n-ary predicate symbols $\mathcal{P}^n$.
- Logical connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$.
- Quantifiers $\exists$, $\forall$.
- Parenthesis '(', ')', and comma ','.

Sometimes the truth constants $\mathbb{T}$ and $\mathbb{F}$, and square brackets are also included in the alphabet.

Alphabet

Notation:

- x, y, z for variables.
- f, g for function symbols.
- a, b, c for constants.
- p, q for predicate symbols.

Terms

Definition

- A variable is a term.
- If $t_1, \ldots, t_n$ are terms and $f \in \mathcal{F}^n$, then $f(t_1, \ldots, t_n)$ is a term.
- Nothing else is a term.

Terms

Definition

- A variable is a term.
- If $t_1, \ldots, t_n$ are terms and $f \in \mathcal{F}^n$, then $f(t_1, \ldots, t_n)$ is a term.
- Nothing else is a term.

- Alternative notation for compound terms: $f[t_1, \ldots, t_n]$.
- s, t, r are used to denote terms.

Terms

Definition

- A variable is a term.
- If $t_1, \ldots, t_n$ are terms and $f \in \mathcal{F}^n$, then $f(t_1, \ldots, t_n)$ is a term.
- Nothing else is a term.

- Alternative notation for compound terms: $f[t_1, \ldots, t_n]$.
- s, t, r are used to denote terms.

Example

- $plus(plus(x, 1), x)$ is a term, where $plus$ is a binary function symbol, 1 is a constant, x is a variable.

Terms

Definition

- A variable is a term.
- If $t_1, \dots, t_n$ are terms and $f \in \mathcal{F}^n$, then $f(t_1, \dots, t_n)$ is a term.
- Nothing else is a term.

- Alternative notation for compound terms: $f[t_1, \dots, t_n]$.
- s, t, r are used to denote terms.

Example

- $plus(plus(x, 1), x)$ is a term, where $plus$ is a binary function symbol, 1 is a constant, x is a variable.
- $father(father(John))$ is a term, where $father$ is a unary function symbol and $John$ is a constant.

Formulas

Definition

- If $t_1, \ldots, t_n$ are terms and $p \in \mathcal{P}^n$, then $p(t_1, \ldots, t_n)$ is a formula. It is called an atomic formula.
- $\mathbb{T}$ and $\mathbb{F}$ are formulas (when the alphabet contains these symbols). They are also atomic formulas.
- If A is a formula, $(\neg A)$ is a formula.
- If A and B are formulas, then $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$, and $(A \Leftrightarrow B)$ are formulas.
- If A is a formula, then $(\exists x.A)$ and $(\forall x.A)$ are formulas.
- Nothing else is a formula.

Formulas

Definition

- If $t_1, \ldots, t_n$ are terms and $p \in \mathcal{P}^n$, then $p(t_1, \ldots, t_n)$ is a formula. It is called an atomic formula.
- $\mathbb{T}$ and $\mathbb{F}$ are formulas (when the alphabet contains these symbols). They are also atomic formulas.
- If A is a formula, $(\neg A)$ is a formula.
- If A and B are formulas, then $(A \vee B)$, $(A \wedge B)$, $(A \Rightarrow B)$, and $(A \Leftrightarrow B)$ are formulas.
- If A is a formula, then $(\exists x.A)$ and $(\forall x.A)$ are formulas.
- Nothing else is a formula.

- Alternative notation: $p[t_1, \ldots, t_n]$, $\underset{x}{\forall} A$, $\underset{x}{\exists} A$.
- A, B are used to denote formulas.

Eliminating Parentheses

- Excessive use of parentheses often can be avoided by introducing binding order.
- $\neg, \forall, \exists$ bind stronger than $\vee$.
- $\vee$ binds stronger than $\wedge$.
- $\wedge$ binds stronger than $\Rightarrow$ and $\Leftrightarrow$.
- Furthermore, omit the outer parentheses and associate $\vee, \wedge, \Rightarrow, \Leftrightarrow$ to the right.

Eliminating Parentheses

Example

The formula

$$(\forall y.(\forall x.((p(x)) \wedge (\neg r(y))) \Rightarrow ((\neg q(x)) \vee (A \vee B)))))$$

due to binding order can be rewritten into

$$(\forall y.(\forall x.(p(x) \wedge \neg r(y) \Rightarrow \neg q(x) \vee (A \vee B))))$$

which thanks to the convention of the association to the right and omitting the outer parentheses further simplifies to

$$\forall y.\forall x.(p(x) \wedge \neg r(y) \Rightarrow \neg q(x) \vee A \vee B).$$

Free and Bound Variables

- A variable is free in a formula A if it is not quantified in A.
- Otherwise, it is bound.
- In $\forall x.p(x,y)$, the variable x is bound and y is free.
- In $\forall x.\big(p(x) \Rightarrow \exists y.q(f(x,z))\big)$, the variables x and y are bound and z is free.
- In $p(x) \Rightarrow \forall x.q(x)$, the variable x is both free and bound.

Example

Identify constants, variables (free, bound), quantifiers, function symbols, predicate symbols, atoms, terms, formulas:

1. $\forall x.\ x + 1 \geq x$.

Example

Identify constants, variables (free, bound), quantifiers, function symbols, predicate symbols, atoms, terms, formulas:

1. $\forall x.\ x + 1 \geq x$.
2. $\neg \exists x.\ eq(0, f(x))$.

Example

Identify constants, variables (free, bound), quantifiers, function symbols, predicate symbols, atoms, terms, formulas:

1. $\forall x.\ x + 1 \geq x$.
2. $\neg \exists x.\ eq(0, f(x))$.
3. $\forall x.\ \exists y. \Big(eq(y, f(x)) \ \wedge \ \forall z. \big(eq(z, f(x)) \Rightarrow eq(y, z) \big) \Big)$.

Example

Translating English sentences into first-order logic formulas:

1. Every rational number is a real number.

Assume:

- $rational\_number, real\_number, prime\_number$: unary predicate symbols.
- $<$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. Every rational number is a real number.

$$\forall x.(rational\_number(x) \Rightarrow real\_number(x))$$

Assume:

- $rational\_number, real\_number, prime\_number$: unary predicate symbols.
- $<$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. Every rational number is a real number.

$$\forall x.(rational\_number(x) \Rightarrow real\_number(x))$$

2. There exists a number that is prime.

Assume:

- $rational\_number, real\_number, prime\_number$: unary predicate symbols.
- $<$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. Every rational number is a real number.

$$\forall x.(rational\_number(x) \Rightarrow real\_number(x))$$

2. There exists a number that is prime.

$$\exists x.\, prime\_number(x)$$

Assume:

- $rational\_number, real\_number, prime\_number$: unary predicate symbols.
- $<$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. Every rational number is a real number.

   $$\forall x.(rational\_number(x) \Rightarrow real\_number(x))$$

2. There exists a number that is prime.

   $$\exists x.\, prime\_number(x)$$

3. For every number x, there exists a number y such that $x < y$.

Assume:

- $rational\_number, real\_number, prime\_number$: unary predicate symbols.
- $<$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. Every rational number is a real number.

$$\forall x.(rational\_number(x) \Rightarrow real\_number(x))$$

2. There exists a number that is prime.

$$\exists x.\, prime\_number(x)$$

3. For every number x, there exists a number y such that $x < y$.

$$\forall x.\, \exists y.\, x < y$$

Assume:

- $rational\_number, real\_number, prime\_number$: unary predicate symbols.
- $<$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. There is no natural number whose immediate successor is 0.

Assume:

- $zero$: constant
- $succ$, $pred$: unary function symbols.
- $\doteq$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. There is no natural number whose immediate successor is 0.

$$\neg \exists x. zero \doteq succ(x)$$

Assume:

- $zero$: constant
- $succ$, $pred$: unary function symbols.
- $\doteq$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. There is no natural number whose immediate successor is 0.

$$\neg\exists x.zero \doteq succ(x)$$

2. For each natural number there exists exactly one immediate successor natural number.

Assume:

- $zero$: constant
- $succ$, $pred$: unary function symbols.
- $\doteq$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. There is no natural number whose immediate successor is 0.

$$\neg \exists x. zero \doteq succ(x)$$

2. For each natural number there exists exactly one immediate successor natural number.

$$\forall x. \exists y. (y \doteq succ(x) \wedge \forall z. (z \doteq succ(x) \Rightarrow y \doteq z))$$

Assume:

- $zero$: constant
- $succ, pred$: unary function symbols.
- $\doteq$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. There is no natural number whose immediate successor is 0.

$$\neg \exists x. zero \doteq succ(x)$$

2. For each natural number there exists exactly one immediate successor natural number.

$$\forall x. \exists y. (y \doteq succ(x) \wedge \forall z. (z \doteq succ(x) \Rightarrow y \doteq z))$$

3. For each nonzero natural number there exists exactly one immediate predecessor natural number.

Assume:

- $zero$: constant
- $succ, pred$: unary function symbols.
- $\doteq$: binary predicate symbol.

Example

Translating English sentences into first-order logic formulas:

1. There is no natural number whose immediate successor is 0.

$$\neg\exists x.zero \doteq succ(x)$$

2. For each natural number there exists exactly one immediate successor natural number.

$$\forall x.\exists y.(y \doteq succ(x) \wedge \forall z.(z \doteq succ(x) \Rightarrow y \doteq z))$$

3. For each nonzero natural number there exists exactly one immediate predecessor natural number.

$$\forall x.(\neg(x \doteq zero) \Rightarrow \exists y.(y \doteq pred(x) \wedge \forall z.(z \doteq pred(x) \Rightarrow y \doteq z)))$$

Assume:

- $zero$: constant
- $succ$, $pred$: unary function symbols.
- $\doteq$: binary predicate symbol.

Semantics

- Meaning of a first-order language consists of an universe and an appropriate meaning of each symbol.
- This pair is called structure.
- Structure fixes interpretation of function and predicate symbols.
- Meaning of variables is determined by a variable assignment.
- Interpretation of terms and formulas.

Structure

- Structure: a pair (D, I).
- D is a nonempty universe, the domain of interpretation.
- I is an interpretation function defined on D that fixes the meaning of each symbol associating
  - to each $f \in \mathcal{F}^n$ an n-ary function $f_I : D^n \to D$, (in particular, $c_I \in D$ for each constant c)
  - to each $p \in \mathcal{P}^n$ different from $\doteq$, an n-ary relation p_I on D.

Variable Assignment

- A structure $\mathcal{S} = (D, I)$ is given.
- Variable assignment $\sigma_{\mathcal{S}}$ maps each $x \in \mathcal{V}$ into an element of D: $\sigma_{\mathcal{S}}(x) \in D$.
- Given a variable x, an assignment $\vartheta_{\mathcal{S}}$ is called an x-variant of $\sigma_{\mathcal{S}}$ iff $\vartheta_{\mathcal{S}}(y) = \sigma_{\mathcal{S}}(y)$ for all $y \neq x$.

Interpretation of Terms

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.

Interpretation of Terms

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Value of a term t under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$, $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(t)$:
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(x) = \sigma_{\mathcal{S}}(x)$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(f(t_1, \ldots, t_n)) = f_I(Val_{\mathcal{S},\sigma_{\mathcal{S}}}(t_1), \ldots, Val_{\mathcal{S},\sigma_{\mathcal{S}}}(t_n))$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_\mathcal{S}$ are given.
- Value of an atomic formula under $\mathcal{S}$ and $\sigma_\mathcal{S}$ is one of *true*, *false* (truth values):

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Value of an atomic formula under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ is one of *true*, *false* (truth values):
  - $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(s \doteq t) = \mathit{true}$ iff $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(s) = \mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(t)$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Value of an atomic formula under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ is one of *true*, *false* (truth values):
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(s \doteq t) = true$ iff $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(s) = Val_{\mathcal{S},\sigma_{\mathcal{S}}}(t)$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(p(t_1, \ldots, t_n)) = true$ iff $(Val_{\mathcal{S},\sigma_{\mathcal{S}}}(t_1), \ldots, Val_{\mathcal{S},\sigma_{\mathcal{S}}}(t_n)) \in p_I$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Value of an atomic formula under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ is one of true, false (truth values):
  - $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(s \doteq t) = \mathit{true}$ iff $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(s) = \mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(t)$.
  - $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(p(t_1, \ldots, t_n)) = \mathit{true}$ iff
    $(\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(t_1), \ldots, \mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(t_n)) \in p_I$.
  - $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(\mathbb{T}) = \mathit{true}$, $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(\mathbb{F}) = \mathit{false}$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ are also either *true* or *false* (truth values):

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ are also either *true* or *false* (truth values):
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\neg A) = true$ iff $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = false$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ are also either *true* or *false* (truth values):
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\neg A) = true$ iff $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = false$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \vee B) = true$ iff
    $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = true$ or $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B) = true$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ are also either *true* or *false* (truth values):
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\neg A) = true$ iff $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = false$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \vee B) = true$ iff
    $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = true$ or $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \wedge B) = true$ iff
    $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = true$ and $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B) = true$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_\mathcal{S}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_\mathcal{S}$ are also either *true* or *false* (truth values):
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(\neg A) = true$ iff $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = false$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \vee B) = true$ iff
    $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = true$ or $Val_{\mathcal{S},\sigma_\mathcal{S}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \wedge B) = true$ iff
    $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = true$ and $Val_{\mathcal{S},\sigma_\mathcal{S}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \Rightarrow B) = true$ iff
    $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = false$ or $Val_{\mathcal{S},\sigma_\mathcal{S}}(B) = true$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_\mathcal{S}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_\mathcal{S}$ are also either *true* or *false* (truth values):
  - $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(\neg A) = \mathit{true}$ iff $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A) = \mathit{false}$.
  - $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A \vee B) = \mathit{true}$ iff
    $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A) = \mathit{true}$ or $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(B) = \mathit{true}$.
  - $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A \wedge B) = \mathit{true}$ iff
    $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A) = \mathit{true}$ and $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(B) = \mathit{true}$.
  - $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A \Rightarrow B) = \mathit{true}$ iff
    $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A) = \mathit{false}$ or $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(B) = \mathit{true}$.
  - $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A \Leftrightarrow B) = \mathit{true}$ iff $\mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(A) = \mathit{Val}_{\mathcal{S},\sigma_\mathcal{S}}(B)$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_\mathcal{S}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_\mathcal{S}$ are also either *true* or *false* (truth values):
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(\neg A) = true$ iff $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = false$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \lor B) = true$ iff
    $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = true$ or $Val_{\mathcal{S},\sigma_\mathcal{S}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \land B) = true$ iff
    $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = true$ and $Val_{\mathcal{S},\sigma_\mathcal{S}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \Rightarrow B) = true$ iff
    $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = false$ or $Val_{\mathcal{S},\sigma_\mathcal{S}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(A \Leftrightarrow B) = true$ iff $Val_{\mathcal{S},\sigma_\mathcal{S}}(A) = Val_{\mathcal{S},\sigma_\mathcal{S}}(B)$.
  - $Val_{\mathcal{S},\sigma_\mathcal{S}}(\exists x.A) = true$ iff
    $Val_{\mathcal{S},\vartheta_\mathcal{S}}(A) = true$ for some x-variant $\vartheta_\mathcal{S}$ of $\sigma_\mathcal{S}$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ and a variable assignment $\sigma_{\mathcal{S}}$ are given.
- Values of compound formulas under $\mathcal{S}$ and $\sigma_{\mathcal{S}}$ are also either *true* or *false* (truth values):
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\neg A) = true$ iff $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = false$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \vee B) = true$ iff
    $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = true$ or $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \wedge B) = true$ iff
    $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = true$ and $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \Rightarrow B) = true$ iff
    $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = false$ or $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B) = true$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A \Leftrightarrow B) = true$ iff $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = Val_{\mathcal{S},\sigma_{\mathcal{S}}}(B)$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\exists x.A) = true$ iff
    $Val_{\mathcal{S},\vartheta_{\mathcal{S}}}(A) = true$ for some x-variant $\vartheta_{\mathcal{S}}$ of $\sigma_{\mathcal{S}}$.
  - $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\forall x.A) = true$ iff
    $Val_{\mathcal{S},\vartheta_{\mathcal{S}}}(A) = true$ for all x-variants $\vartheta_{\mathcal{S}}$ of $\sigma_{\mathcal{S}}$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ is given.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ is given.
- The value of a formula A under $\mathcal{S}$ is either *true* or *false*:
  - $Val_{\mathcal{S}}(A) = true$ iff $Val_{\mathcal{S}}, \sigma_{\mathcal{S}}(A) = true$ for all $\sigma_{\mathcal{S}}$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ is given.
- The value of a formula A under $\mathcal{S}$ is either *true* or *false*:
  - $\mathit{Val}_{\mathcal{S}}(A) = \mathit{true}$ iff $\mathit{Val}_{\mathcal{S},\sigma_{\mathcal{S}}}(A) = \mathit{true}$ for all $\sigma_{\mathcal{S}}$.
- $\mathcal{S}$ is called a model of A iff $\mathit{Val}_{\mathcal{S}}(A) = \mathit{true}$.

Interpretation of Formulas

- A structure $\mathcal{S} = (D, I)$ is given.
- The value of a formula A under $\mathcal{S}$ is either *true* or *false*:
  - $Val_{\mathcal{S}}(A) = true$ iff $Val_{\mathcal{S}, \sigma_{\mathcal{S}}}(A) = true$ for all $\sigma_{\mathcal{S}}$.
- $\mathcal{S}$ is called a model of A iff $Val_{\mathcal{S}}(A) = true$.
- Written $\models_{\mathcal{S}} A$.

Example

- Formula: $\forall x.(p(x) \Rightarrow q(f(x), a))$

Example

- Formula: $\forall x.(p(x) \Rightarrow q(f(x), a))$
- Define $\mathcal{S} = (D, I)$ as
  - $D = \{1, 2\}$,
  - $a_I = 1$,
  - $f_I(1) = 2, f_I(2) = 1$,
  - $p_I = \{2\}$,
  - $q_I = \{(1, 1), (1, 2), (2, 2)\}$.

Example

- Formula: $\forall x.(p(x) \Rightarrow q(f(x), a))$
- Define $\mathcal{S} = (D, I)$ as
  - $D = \{1, 2\}$,
  - $a_I = 1$,
  - $f_I(1) = 2, f_I(2) = 1$,
  - $p_I = \{2\}$,
  - $q_I = \{(1, 1), (1, 2), (2, 2)\}$.
- If $\sigma_{\mathcal{S}}(x) = 1$, then $Val_{\mathcal{S},\sigma_{\mathcal{S}}}(\forall x.(p(x) \Rightarrow q(f(x), a))) = true$.

Example

- Formula: $\forall x.(p(x) \Rightarrow q(f(x), a))$
- Define $\mathcal{S} = (D, I)$ as
  - $D = \{1, 2\}$,
  - $a_I = 1$,
  - $f_I(1) = 2, f_I(2) = 1$,
  - $p_I = \{2\}$,
  - $q_I = \{(1, 1), (1, 2), (2, 2)\}$.
- If $\sigma_{\mathcal{S}}(x) = 1$, then $Val_{\mathcal{S},\sigma_S}(\forall x.(p(x) \Rightarrow q(f(x), a))) = true$.
- If $\sigma_{\mathcal{S}}(x) = 2$, then $Val_{\mathcal{S},\sigma_S}(\forall x.(p(x) \Rightarrow q(f(x), a))) = true$.

Example

- Formula: $\forall x.(p(x) \Rightarrow q(f(x), a))$
- Define $\mathcal{S} = (D, I)$ as
  - $D = \{1, 2\}$,
  - $a_I = 1$,
  - $f_I(1) = 2, f_I(2) = 1$,
  - $p_I = \{2\}$,
  - $q_I = \{(1, 1), (1, 2), (2, 2)\}$.
- If $\sigma_{\mathcal{S}}(x) = 1$, then $Val_{\mathcal{S},\sigma_S}(\forall x.(p(x) \Rightarrow q(f(x), a))) = true$.
- If $\sigma_{\mathcal{S}}(x) = 2$, then $Val_{\mathcal{S},\sigma_S}(\forall x.(p(x) \Rightarrow q(f(x), a))) = true$.
- Hence, $\models_{\mathcal{S}} A$.

Example

Find the truth value of the formula $\forall x.\ \exists y.\ x + y > c$, in the structure $\mathcal{S} = (D, I)$ defined as:

- $D = \{0, 1\}$.
- $c_I = 0$.
- $+_I = +_{\mathbb{Z}}$ (addition on integers).
- $>_I = >_{\mathbb{Z}}$ (strictly greater than).

Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.

Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.

Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.

Validity, Unsatisfiability

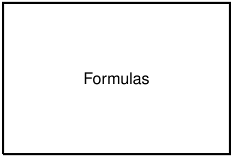
- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.
- The notions extend to (multi)sets of formulas.

Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.
- The notions extend to (multi)sets of formulas.
- For $\{A_1, \ldots, A_n\}$, just formulate them for $A_1 \wedge \cdots \wedge A_n$.

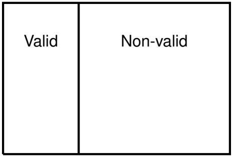
Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.
- The notions extend to (multi)sets of formulas.
- For $\{A_1, \ldots, A_n\}$, just formulate them for $A_1 \wedge \cdots \wedge A_n$.



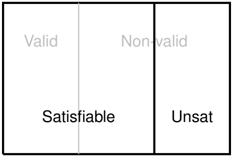
Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.
- The notions extend to (multi)sets of formulas.
- For $\{A_1, \ldots, A_n\}$, just formulate them for $A_1 \wedge \cdots \wedge A_n$.



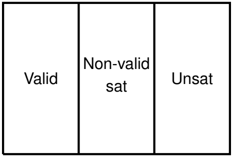
Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.
- The notions extend to (multi)sets of formulas.
- For $\{A_1, \ldots, A_n\}$, just formulate them for $A_1 \wedge \cdots \wedge A_n$.

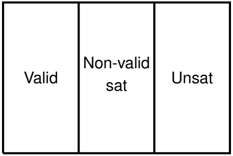


Validity, Unsatisfiability

- A formula A is valid, if $\models_{\mathcal{S}} A$ for all $\mathcal{S}$.
- Written $\models A$.
- A formula A is unsatisfiable, if $\models_{\mathcal{S}} A$ for no $\mathcal{S}$.
- If A is valid, then $\neg A$ is unsatisfiable and vice versa.
- The notions extend to (multi)sets of formulas.
- For $\{A_1, \dots, A_n\}$, just formulate them for $A_1 \wedge \cdots \wedge A_n$.



Validity, Unsatisfiability



- $\forall x.p(x) \Rightarrow \exists y.p(y)$ is valid.
- $p(a) \Rightarrow \neg\exists x.p(x)$ is satisfiable non-valid.
- $\forall x.p(x) \wedge \exists y.\neg p(y)$ is unsatisfiable.

Logical Consequence

Definition

A formula A is a logical consequence of the formulas $B_1, \ldots, B_n$, if every model of $B_1 \wedge \cdots \wedge B_n$ is a model of A.

Logical Consequence

Definition

A formula A is a logical consequence of the formulas $B_1, \ldots, B_n$, if every model of $B_1 \wedge \cdots \wedge B_n$ is a model of A.

Example

- $mortal(socrates)$ is a logical consequence of $\forall x.(person(x) \Rightarrow mortal(x))$ and $person(socrates)$.
- $cooked(apple)$ is a logical consequence of $\forall x.(\neg cooked(x) \Rightarrow tasty(x))$ and $\neg tasty(apple)$.
- $genius(einstein)$ is not a logical consequence of $\exists x.person(x) \wedge genius(x)$ and $person(einstein)$.

Logical Equivalence

Definition

Two formulas are logically equivalent if they are a logical consequence of each other (i.e., they have exactly the same models).

Normal Forms

Definition

A formula A is in a negation normal form (NNF) if

- A does not contain $\Rightarrow$ and $\Leftrightarrow$, and
- if A contains a subformula $\neg B$, then B is an atomic formula.

Normal Forms

Definition

A formula A is in a negation normal form (NNF) if

- A does not contain $\Rightarrow$ and $\Leftrightarrow$, and
- if A contains a subformula $\neg B$, then B is an atomic formula.

Example

- $p(a) \vee \exists x. \neg q(x)$ is in NNF.
- $p(a) \vee \neg \exists x. q(x)$ is not in NNF.
- $\neg p(a) \vee \neg q(b)$ is in NNF.
- $\neg (p(a) \wedge q(b))$ is not in NNF.
- $\exists x. (p(x) \wedge \neg q(x))$ is in NNF.
- $\exists x. (p(x) \Rightarrow \neg q(x))$ is not in NNF.

Normal Forms

Definition

A formula is in prenex normal form (PNF), if it has the form $Q_1\, x_1.Q_2\, x_2. \cdots Q_n\, x_n.M$, $n \geq 0$, where

- each Q_i is either $\forall$ or $\exists$,
- $x_1, \ldots x_n$ are distinct variables,
- M (called the matrix) does not contain quantifiers.

Normal Forms

Definition

A formula is in prenex normal form (PNF), if it has the form $Q_1\, x_1 . Q_2\, x_2 . \cdots Q_n\, x_n . M$, $n \geq 0$, where

- each Q_i is either $\forall$ or $\exists$,
- $x_1, \ldots x_n$ are distinct variables,
- M (called the matrix) does not contain quantifiers.

Example

- $\exists x.(p(x) \Rightarrow \neg q(a))$ is in PNF.
- $p(a) \vee \neg \exists x.q(x)$ is not in PNF.
- $\neg p(a) \vee \neg q(b)$ is in NNF.
- $\forall x.(\neg(p(x) \wedge \exists y.q(y)))$ is not in PNF.
- $\forall x.\exists y.\forall z.(\neg(p(x,y) \wedge q(y,z)))$ is in PNF.
- $\forall x.\neg \exists y.\forall z.(\neg(p(x,y) \wedge q(y,z)))$ is not in PNF.

Normal Forms

Propositions:

- For each formula A there exists a formula B in NNF such that A and B are logically equivalent.
  - Such a B is called an NNF of A.
  - An NNF of a formula is not unique.

Normal Forms

Propositions:

- For each formula A there exists a formula B in NNF such that A and B are logically equivalent.
  - Such a B is called an NNF of A.
  - An NNF of a formula is not unique.
- For each formula A there exists a formula B in PNF such that A and B are logically equivalent.
  - Such a B is called a PNF of A.
  - A PNF of a formula is not unique.

Transformation into a Prenex Normal Form

Idea:

Define transformations

$$A \rightsquigarrow B \rightsquigarrow C,$$

where B is an NNF of A, and C is a PNF of B and of A.

Transformation into a Prenex Normal Form

First set of rules, to get an NNF of a formula.
Rules are applied exhaustively.

1. Eliminate of $\Leftrightarrow$ and $\Rightarrow$:

$$A \Leftrightarrow B \quad \leadsto \quad (A \Rightarrow B) \wedge (B \Rightarrow A)$$
$$A \Rightarrow B \quad \leadsto \quad \neg A \vee B$$

2. Eliminate double negation and push the negation inside:

$$\neg(\neg A) \quad \leadsto \quad A$$
$$\neg(A \vee B) \quad \leadsto \quad \neg A \wedge \neg B$$
$$\neg(A \wedge B) \quad \leadsto \quad \neg A \vee \neg B$$
$$\neg \forall x.A \quad \leadsto \quad \exists x.\neg A$$
$$\neg \exists x.A \quad \leadsto \quad \forall x.\neg A$$

Transformation into a Prenex Normal Form

Second set of rules, to get an PNF of a formula in NNF.
Rules in 4 are applied exhaustively.

3. Rename of all bound variables.
4. Move the quantifiers to the left of the entire formula:

$$
\begin{array}{lcl}
Qx.A[x] \vee B & \rightsquigarrow & Qx.(A[x] \vee B) \\
Qx.A[x] \wedge B & \rightsquigarrow & Qx.(A[x] \wedge B) \\
Qx.B \vee A[x] & \rightsquigarrow & Qx.(B \vee A[x]) \\
Qx.B \wedge A[x] & \rightsquigarrow & Qx.(B \wedge A[x])
\end{array}
$$

where B does not contain free occurrences of x.
($A[x]$ means that x occurs freely in A.)

Transformation into a Conjunctive Normal Form

- CNF: A formula of the form $\mathcal{C}_1 \wedge \cdots \wedge \mathcal{C}_n$, where each $\mathcal{C}_i$ is a disjunction of literals: $L^i_1 \vee \cdots \vee L^i_k$.
- If $Q_1 x_1. \cdots Q_n x_n.M$ is in prenex normal form, then M is in NNF.
- M consists of disjunctions and conjunctions of literals.
- CNF is obtained from NNF by distributing disjunction over conjunction:

$$\begin{array}{rcl} A \vee (B \wedge C) & \rightsquigarrow & (A \vee B) \wedge (A \vee C) \\ (B \wedge C) \vee A & \rightsquigarrow & (B \vee A) \wedge (C \vee A) \end{array}$$

Examples

Prove that

$$\forall x.p(x) \Rightarrow q$$

is logically equivalent to

$$\exists x.(p(x) \Rightarrow q)$$

by bringing them to PNF with the matrix in CNF:

Skolemization

Replace existentially quantified variables by Skolem functions:

- The formula $Q_1 x_1. \cdots Q_n x_n.M$ is in PNF and M is in CNF.
- Skolemization is performed by repeatedly applying the following rule:

$$\forall x_1. \cdots \forall x_n.\exists y.Q_1 z_1. \cdots Q_m z_m. M[y] \quad \rightsquigarrow$$
$$\forall x_1. \cdots \forall x_n.Q_1 z_1. \cdots Q_m z_m. M[f(x_1, \ldots, x_n)]$$

where f is a new function symbol of arity n with $n \geq 0$.

Example

$$\forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u))$$

Example

$$
\begin{aligned}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow \quad & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u))
\end{aligned}
$$

Example

$$
\begin{array}{rl}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u))
\end{array}
$$

Example

$$
\begin{aligned}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.(\forall z.(\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u))
\end{aligned}
$$

Example

$$
\begin{array}{ll}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\forall z.(\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.\forall z.((\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u))
\end{array}
$$

Example

$$
\begin{aligned}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\leadsto\quad & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\quad & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\quad & \forall x.\exists y.(\forall z.(\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\quad & \forall x.\exists y.\forall z.((\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\quad & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \wedge \neg p(y,z)) \vee q(x,y,u))
\end{aligned}
$$

Example

$$
\begin{array}{ll}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.(\forall z.(\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.\forall z.((\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \wedge \neg p(y,z)) \vee q(x,y,u)) \\
\rightsquigarrow & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \vee q(x,y,u)) \wedge (\neg p(y,z) \vee q(x,y,u)))
\end{array}
$$

Example

$$
\begin{aligned}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\leadsto\ & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\ & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\ & \forall x.\exists y.(\forall z.(\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\ & \forall x.\exists y.\forall z.((\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\leadsto\ & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \wedge \neg p(y,z)) \vee q(x,y,u)) \\
\leadsto\ & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \vee q(x,y,u)) \ \wedge (\neg p(y,z) \vee q(x,y,u))) \\
\leadsto\ & \forall x.\forall z.\exists u.((\neg p(x,z) \vee q(x,f_1(x),u)) \wedge \\
& \qquad\qquad (\neg p(f_1(x),z) \vee q(x,f_1(x),u)))
\end{aligned}
$$

Example

$$
\begin{aligned}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.(\neg\exists z.(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.(\forall z.\neg(p(x,z) \vee p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.(\forall z.(\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.\forall z.((\neg p(x,z) \wedge \neg p(y,z)) \vee \exists u.q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \wedge \neg p(y,z)) \vee q(x,y,u)) \\
\rightsquigarrow\ & \forall x.\exists y.\forall z.\exists u.((\neg p(x,z) \vee q(x,y,u)) \wedge (\neg p(y,z) \vee q(x,y,u))) \\
\rightsquigarrow\ & \forall x.\forall z.\exists u.((\neg p(x,z) \vee q(x,f_1(x),u)) \wedge \\
& \qquad\qquad (\neg p(f_1(x),z) \vee q(x,f_1(x),u))) \\
\rightsquigarrow\ & \forall x.\forall z.((\neg p(x,z) \vee q(x,f_1(x),f_2(x,z))) \wedge \\
& \qquad\qquad (\neg p(f_1(x),z) \vee q(x,f_1(x),f_2(x,z))))
\end{aligned}
$$

Example

$$\begin{aligned}
& \forall x.\exists y.(\exists z.(p(x,z) \vee p(y,z)) \Rightarrow \exists u.q(x,y,u)) \\
\rightsquigarrow\quad & \forall x.\forall z.((\neg p(x,z) \vee q(x, f_1(x), f_2(x,z))) \wedge \\
& \qquad (\neg p(f_1(x),z) \vee q(x, f_1(x), f_2(x,z)))).
\end{aligned}$$

A Property of Skolem Normal Form

Theorem

Let A be a formula and B be its Skolem normal form. Then A is unsatisfiable iff B is unsatisfiable.

A Property of Skolem Normal Form

Theorem

Let A be a formula and B be its Skolem normal form. Then A is unsatisfiable iff B is unsatisfiable.

- A and B are, in general, not equivalent.

A Property of Skolem Normal Form

Theorem

Let A be a formula and B be its Skolem normal form. Then A is unsatisfiable iff B is unsatisfiable.

- A and B are, in general, not equivalent.
- Example:
  - $A = \exists x.p(x)$, $B = p(a)$.
  - $\mathcal{S} = (\{1, 2\}, I)$.
  - $a_I = 1$.
  - $p_I = \{2\}$.
  - Then $Val_{\mathcal{S}}(A) = true$ but $Val_{\mathcal{S}}(B) = false$.