

# Propositional Logic: Strength and Weakness

---

- ♠ Programming languages lack a general mechanism for deriving facts from other facts.
- ♠ They lack the expressiveness required to handle partial information.

# Propositional Logic: Strength and Weakness

---

- ♠ Programming languages lack a general mechanism for deriving facts from other facts.
- ♠ They lack the expressiveness required to handle partial information.
- ♦ Propositional logic addresses the above issues.

# Propositional Logic: Strength and Weakness

---

- ♠ Programming languages lack a general mechanism for deriving facts from other facts.
- ♠ They lack the expressiveness required to handle partial information.
- ◆ Propositional logic addresses the above issues.
- ◆ It also has *compositionality* – the meaning of a sentence is a function of the meanings of its parts.

$\neg \text{rain} \vee \neg \text{outside} \vee \text{wet}$

# Propositional Logic: Strength and Weakness

---

- ♠ Programming languages lack a general mechanism for deriving facts from other facts.
- ♠ They lack the expressiveness required to handle partial information.
- ◆ Propositional logic addresses the above issues.
- ◆ It also has *compositionality* – the meaning of a sentence is a function of the meanings of its parts.

$\neg \text{rain} \vee \neg \text{outside} \vee \text{wet}$

- ♠ It lacks the expressive power to describe an environment with *many objects*.

# Propositional Logic: Strength and Weakness

---

- ♠ Programming languages lack a general mechanism for deriving facts from other facts.
- ♠ They lack the expressiveness required to handle partial information.
- ♦ Propositional logic addresses the above issues.
- ♦ It also has *compositionality* – the meaning of a sentence is a function of the meanings of its parts.

$\neg \text{rain} \vee \neg \text{outside} \vee \text{wet}$

- ♠ It lacks the expressive power to describe an environment with *many objects*.

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$$B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{1,3} \vee P_{2,2})$$

⋮

// Squares adjacent to pits are breezy.

# Propositional Logic: Strength and Weakness

---

- ♠ Programming languages lack a general mechanism for deriving facts from other facts.
- ♠ They lack the expressiveness required to handle partial information.
- ♦ Propositional logic addresses the above issues.
- ♦ It also has *compositionality* – the meaning of a sentence is a function of the meanings of its parts.

$\neg \text{rain} \vee \neg \text{outside} \vee \text{wet}$

- ♠ It lacks the expressive power to describe an environment with *many objects*.

$$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$$

$$B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{1,3} \vee P_{2,2})$$

⋮

// Squares adjacent to pits are breezy.

- ♠ Propositional logic assumes the world contains *facts only*.

# Combining Formal and Natural Languages

---

## First-order logic

- ◆ built around objects and relations
  - **Objects:** people, houses, cars, trees, colors, days, ...
  - **Relations:**
    - ◆ unary properties such as `big`, `windy`, ...
    - ◆  $n$ -ary properties such as `bigger than`, `parent of`, `on`, `owns`, ...
  - **Functions:** `square of`, `best friend`, `age`, ...
- ◆ capable of expressing facts about some or all objects

# Formal Languages

---

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value



# Alphabet of First-Order Logic

---

## ◆ Logical symbols

- connectives:  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ ,  $\neg$
- parenthesis: (, ) and punctuation ,
- equality: =
- quantifiers:  $\forall$  (universal quantification),  $\exists$  (existential quantification)
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$

# Alphabet of First-Order Logic

---

## ◆ Logical symbols

- connectives:  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ ,  $\neg$
- parenthesis: (, ) and punctuation ,
- equality: =
- quantifiers:  $\forall$  (universal quantification),  $\exists$  (existential quantification)
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$

## ◆ Non-logical symbols

- constants: Socrates, Turing, 1, earth, ...

# Alphabet of First-Order Logic

---

## ◆ Logical symbols

- connectives:  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ ,  $\neg$
- parenthesis: (, ) and punctuation ,
- equality: =
- quantifiers:  $\forall$  (universal quantification),  $\exists$  (existential quantification)
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$

## ◆ Non-logical symbols

- constants: Socrates, Turing, 1, earth, ...
- predicate symbols: *true*, *false*

# Alphabet of First-Order Logic

---

## ◆ Logical symbols

- connectives:  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ ,  $\neg$
- parenthesis: (, ) and punctuation ,
- equality: =
- quantifiers:  $\forall$  (universal quantification),  $\exists$  (existential quantification)
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$

## ◆ Non-logical symbols

- constants: Socrates, Turing, 1, earth, ...
- predicate symbols:
  - true*, *false*
  - Father*( $x, y$ ) //  $x$  is father of  $y$
  - Female*( $x$ ) //  $x$  is female

# Alphabet of First-Order Logic

---

## ◆ Logical symbols

- connectives:  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ ,  $\neg$
- parenthesis: (, ) and punctuation ,
- equality: =
- quantifiers:  $\forall$  (universal quantification),  $\exists$  (existential quantification)
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$

## ◆ Non-logical symbols

- constants: Socrates, Turing, 1, earth, ...
- predicate symbols:
  - true*, *false*
  - Father*( $x, y$ ) //  $x$  is father of  $y$
  - Female*( $x$ ) //  $x$  is female
- function symbols:
  - gcd*( $x, y$ ) // greatest common divisor of  $x$  and  $y$
  - FatherOf*( $x$ ) // father of  $x$

# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...

# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$

# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$
- functions:  $\text{gcd}(x, y)$ ,  $\text{FatherOf}(x)$ , ...

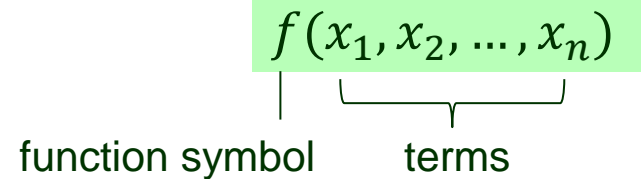


# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$
- functions:  $\text{gcd}(x, y)$ ,  $\text{FatherOf}(x)$ , ...

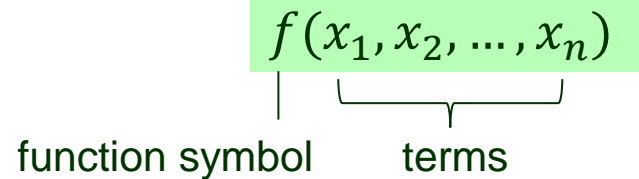


# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$
- functions:  $\text{gcd}(x, y)$ ,  $\text{FatherOf}(x)$ , ...



## ◆ Atomic sentences

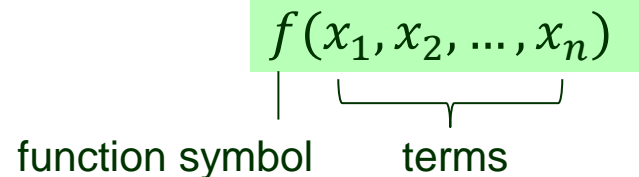
- predicates: *true*, *false*

# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$
- functions:  $\text{gcd}(x, y)$ ,  $\text{FatherOf}(x)$ , ...



## ◆ Atomic sentences

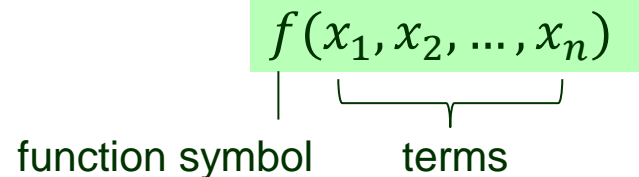
- predicates: *true*, *false*  
*Mother*(Aphrodite, Harmonia)  
*Male*(John)

# Terms and Atomic Sentences

---

## ◆ Terms

- constants: Socrates, Turing, 1, earth, ...
- variables:  $x, y, z, \dots$ ;  $x_1, x_2, \dots$
- functions:  $\text{gcd}(x, y)$ ,  $\text{FatherOf}(x)$ , ...



## ◆ Atomic sentences

- predicates: *true*, *false*  
*Mother*(Aphrodite, Harmonia)  
*Male*(John)
- term equalities

$$\text{FatherOf}(\text{Apollo}) = \text{Zeus}$$

# Complex Sentences

---

- made of atomic sentences using logical connectives

$Father(x, y) \Rightarrow Male(x)$

$Female(x) \vee \neg Mother(x, y)$

$Likes(Mary, John) \Leftrightarrow Likes(John, Mary)$

$(Parent(x, y) \wedge Parent(y, z)) \Rightarrow GrandParent(x, z)$

# Complex Sentences

---

- made of atomic sentences using logical connectives

$Father(x, y) \Rightarrow Male(x)$

$Female(x) \vee \neg Mother(x, y)$

$Likes(Mary, John) \Leftrightarrow Likes(John, Mary)$

$(Parent(x, y) \wedge Parent(y, z)) \Rightarrow GrandParent(x, z)$

- universal quantification

$\forall x Circle(x) \Rightarrow Ellipse(x)$  // Every circle is an ellipse.

$\neg \forall x Likes(x, sushi)$  // Not everyone likes sushi.

$\forall x Integer(x) \Rightarrow (Even(x) \vee Odd(x))$  // Every integer is either even or odd.

# Complex Sentences

---

- made of atomic sentences using logical connectives

$Father(x, y) \Rightarrow Male(x)$

$Female(x) \vee \neg Mother(x, y)$

$Likes(Mary, John) \Leftrightarrow Likes(John, Mary)$

$(Parent(x, y) \wedge Parent(y, z)) \Rightarrow GrandParent(x, z)$

- universal quantification

$\forall x Circle(x) \Rightarrow Ellipse(x)$  // Every circle is an ellipse.

$\neg \forall x Likes(x, sushi)$  // Not everyone likes sushi.

$\forall x Integer(x) \Rightarrow (Even(x) \vee Odd(x))$  // Every integer is either even or odd.

- existential quantification

$\exists x Star(x) \wedge \neg (x = Sun)$  // There are stars other than the sun.

$\exists x Whale(x) \wedge (Age(x) = 200)$  // Some whales live to 200 years.

# Syntax of First-Order Logic

---

*Sentence*  $\rightarrow$  *AtomicSentence* | *ComplexSentence*

*AtomicSentence*  $\rightarrow$  *Predicate* | *Predicate*(*Term*, ...) | *Term* = *Term*

*ComplexSentence*  $\rightarrow$  ( *Sentence* )

|  $\neg$  *Sentence*

| *Sentence*  $\wedge$  *Sentence*

| *Sentence*  $\vee$  *Sentence*

| *Sentence*  $\Rightarrow$  *Sentence*

| *Sentence*  $\Leftrightarrow$  *Sentence*

| *Quantifier* *Variable*, ... *Sentence*

*Term*  $\rightarrow$  *Function*(*Term*, ...)

| *Constant*

| *Variable*

*Quantifier*  $\rightarrow$   $\forall$  |  $\exists$

*Constant*  $\rightarrow$  *A* | *X*<sub>1</sub> | *John* | ...

*Variable*  $\rightarrow$  *a* | *x* | *s* | ...

*Predicate*  $\rightarrow$  *True* | *False* | *After* | *Loves* | *Raining* | ...

*Function*  $\rightarrow$  *Mother* | *LeftLeg* | ...

OPERATOR PRECEDENCE :  $\neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow$



# Translating English Sentences into FOL

---

1. Everyone makes mistakes.

$$\forall x \exists y \text{ Human}(x) \Rightarrow (\text{Does}(x, y) \wedge \text{Mistake}(y))$$

2. One person is an ancestor of another if he is a parent of the other or he is an ancestor of an ancestor of the other.

$$\forall x \forall y \left( \text{Parent}(x, y) \vee \exists z (\text{Ancestor}(x, z) \wedge \text{Ancestor}(z, y)) \right) \Rightarrow \text{Ancestor}(x, y)$$