

---

# **Introduction to Artificial Intelligence**

## **First-order Logic**

**(Logic, Deduction, Knowledge Representation)**

**Bernhard Beckert**



**UNIVERSITÄT KOBLENZ-LANDAU**

**Wintersemester 2003/2004**

# Outline

---

- Why first-order logic?
- Syntax and semantics of first-order logic
- Fun with sentences
- Wumpus world in first-order logic

# Pros and Cons of Propositional Logic

---

- 😊 Propositional logic is **declarative**:  
pieces of syntax correspond to facts

# Pros and Cons of Propositional Logic

---

- 😊 Propositional logic is **declarative**:  
pieces of syntax correspond to facts
- 😊 Propositional logic allows partial / disjunctive / negated information  
(unlike most data structures and databases)

# Pros and Cons of Propositional Logic

---

- 😊 Propositional logic is **declarative**:  
pieces of syntax correspond to facts
- 😊 Propositional logic allows partial / disjunctive / negated information  
(unlike most data structures and databases)
- 😊 Propositional logic is **compositional**:  
meaning of  $B_{1,1} \wedge P_{1,2}$  is derived from meaning of  $B_{1,1}$  and of  $P_{1,2}$

# Pros and Cons of Propositional Logic

---

- 😊 Propositional logic is **declarative**:  
pieces of syntax correspond to facts
- 😊 Propositional logic allows partial / disjunctive / negated information  
(unlike most data structures and databases)
- 😊 Propositional logic is **compositional**:  
meaning of  $B_{1,1} \wedge P_{1,2}$  is derived from meaning of  $B_{1,1}$  and of  $P_{1,2}$
- 😊 Meaning in propositional logic is **context-independent**  
(unlike natural language, where meaning depends on context)

# Pros and Cons of Propositional Logic

---

- 😊 Propositional logic is **declarative**:  
pieces of syntax correspond to facts
- 😊 Propositional logic allows partial / disjunctive / negated information  
(unlike most data structures and databases)
- 😊 Propositional logic is **compositional**:  
meaning of  $B_{1,1} \wedge P_{1,2}$  is derived from meaning of  $B_{1,1}$  and of  $P_{1,2}$
- 😊 Meaning in propositional logic is **context-independent**  
(unlike natural language, where meaning depends on context)
- 😢 Propositional logic has very limited expressive power  
(unlike natural language)

**Example:**

Cannot say “pits cause breezes in adjacent squares”  
except by writing one sentence for each square

# First-order Logic

---

## Propositional logic

Assumes that the world contains **facts**

# First-order Logic

---

## Propositional logic

Assumes that the world contains **facts**

## First-order logic

Assumes that the world contains

- **Objects**  
people, houses, numbers, theories, Donald Duck, colors, centuries, ...

# First-order Logic

---

## Propositional logic

Assumes that the world contains **facts**

## First-order logic

Assumes that the world contains

- **Objects**  
people, houses, numbers, theories, Donald Duck, colors, centuries, ...
- **Relations**  
red, round, prime, multistoried, ...  
brother of, bigger than, part of, has color, occurred after, owns, ...

# First-order Logic

---

## Propositional logic

Assumes that the world contains **facts**

## First-order logic

Assumes that the world contains

- **Objects**  
people, houses, numbers, theories, Donald Duck, colors, centuries, ...
- **Relations**  
red, round, prime, multistoried, ...  
brother of, bigger than, part of, has color, occurred after, owns, ...
- **Functions**  
+, middle of, father of, one more than, beginning of, ...

# Syntax of First-order Logic: Basic Elements

---

## Symbols

**Constants**    *KingJohn, 2, Koblenz, C, ...*

**Predicates**    *Brother, >, =, ...*

**Functions**    *Sqrt, LeftLegOf, ...*

**Variables**    *x, y, a, b, ...*

**Connectives**     $\wedge$      $\vee$      $\neg$      $\Rightarrow$      $\Leftrightarrow$

**Quantifiers**     $\forall$      $\exists$

# Syntax of First-order Logic: Basic Elements

---

## Symbols

**Constants**    *KingJohn, 2, Koblenz, C, ...*

**Predicates**    *Brother, >, =, ...*

**Functions**    *Sqrt, LeftLegOf, ...*

**Variables**    *x, y, a, b, ...*

**Connectives**     $\wedge \quad \vee \quad \neg \quad \Rightarrow \quad \Leftrightarrow$

**Quantifiers**     $\forall \quad \exists$

## Note

The **equality predicate** is always in the vocabulary

It is written in infix notation

# Syntax of First-order Logic: Atomic Sentences

---

## Atomic sentence

*predicate* ( *term*<sub>1</sub>, …, *term*<sub>*n*</sub> )

or

*term*<sub>1</sub> = *term*<sub>2</sub>

# Syntax of First-order Logic: Atomic Sentences

---

## Atomic sentence

*predicate ( term<sub>1</sub>, …, term<sub>n</sub> )*

or

*term<sub>1</sub> = term<sub>2</sub>*

## Term

*function ( term<sub>1</sub>, …, term<sub>n</sub> )*

or

*constant*

or

*variable*

# Syntax of First-order Logic: Atomic Sentences

---

## Example

*Brother ( KingJohn, RichardTheLionheart )*

# Syntax of First-order Logic: Atomic Sentences

---

## Example

*Brother ( KingJohn, RichardTheLionheart )*

The diagram illustrates the components of the atomic sentence. The word "Brother" is underlined by a green bracket and labeled "predicate". The two arguments "KingJohn" and "RichardTheLionheart" are each underlined by a green bracket and labeled "constant".

# Syntax of First-order Logic: Atomic Sentences

---

## Example

*Brother ( KingJohn, RichardTheLionheart )*

The diagram shows the atomic sentence *Brother ( KingJohn, RichardTheLionheart )*. It is annotated with green curly braces and labels. The first brace, under 'Brother', is labeled 'predicate'. The next two braces, both under 'KingJohn' and 'RichardTheLionheart', are labeled 'constant'. The final two braces, also under 'KingJohn' and 'RichardTheLionheart', are labeled 'term'.

predicate      constant      constant  
                  term                  term

# Syntax of First-order Logic: Atomic Sentences

---

## Example

*Brother ( KingJohn, RichardTheLionheart )*

The diagram illustrates the structure of the atomic sentence "Brother ( KingJohn, RichardTheLionheart )". It uses green curly braces to group parts of the sentence. The first brace, spanning from the opening parenthesis to the closing parenthesis, is labeled "atomic sentence". Inside this, the second brace groups the predicate "Brother" with its arguments. This inner brace is divided into two segments: the first segment, from "Brother" to the first comma, is labeled "predicate"; the second segment, from the first comma to the closing parenthesis, is labeled "constant". The third brace, which covers the entire argument list "KingJohn, RichardTheLionheart", is labeled "constant". The fourth brace, which covers the entire argument "KingJohn", is labeled "term"; the fifth brace, which covers the entire argument "RichardTheLionheart", is also labeled "term".

# Syntax of First-order Logic: Atomic Sentences

---

## Example

> ( $\text{Length}(\text{LeftLegOf}(\text{Richard}))$ ,  $\text{Length}(\text{LeftLegOf}(\text{KingJohn}))$ )

# Syntax of First-order Logic: Atomic Sentences

---

## Example

>  $(\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$

The expression is annotated with green curly braces indicating its components:

- The first opening parenthesis is grouped by a brace labeled "predicate".
- The first closing parenthesis is grouped by a brace labeled "function".
- The second opening parenthesis is grouped by a brace labeled "function".
- The second closing parenthesis is grouped by a brace labeled "constant".
- The third opening parenthesis is grouped by a brace labeled "function".
- The third closing parenthesis is grouped by a brace labeled "function".
- The fourth opening parenthesis is grouped by a brace labeled "constant".

# Syntax of First-order Logic: Atomic Sentences

---

## Example

>  $(\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$

predicate      function      function      constant  
                        term                  function      function      constant  
                        term

# Syntax of First-order Logic: Atomic Sentences

---

## Example

>  $(\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$

The diagram illustrates the structure of the atomic sentence. It shows the following levels of abstraction:

- predicate**: The top level, represented by a green bracket under the opening parenthesis.
- function**: A middle level of grouping, indicated by green brackets under "Length" and "LeftLegOf".
- constant**: The bottom level of grouping, indicated by green brackets under "Richard" and "KingJohn".
- term**: A label for the entire argument structure of each term, indicated by a green bracket under the two arguments of each term.
- atomic sentence**: The overall structure, indicated by a large green bracket under the entire sentence.

# Syntax of First-order Logic: Complex Sentences

---

Built from atomic sentences using connectives

$$\neg S \quad S_1 \wedge S_2 \quad S_1 \vee S_2 \quad S_1 \Rightarrow S_2 \quad S_1 \Leftrightarrow S_2$$

(as in propositional logic)

# Syntax of First-order Logic: Complex Sentences

---

Built from atomic sentences using connectives

$$\neg S \quad S_1 \wedge S_2 \quad S_1 \vee S_2 \quad S_1 \Rightarrow S_2 \quad S_1 \Leftrightarrow S_2$$

(as in propositional logic)

Example

$$Sibling( KingJohn, Richard ) \Rightarrow Sibling( Richard, KingJohn )$$

# Syntax of First-order Logic: Complex Sentences

---

Built from atomic sentences using connectives

$$\neg S \quad S_1 \wedge S_2 \quad S_1 \vee S_2 \quad S_1 \Rightarrow S_2 \quad S_1 \Leftrightarrow S_2$$

(as in propositional logic)

Example

$$Sibling( KingJohn, Richard ) \Rightarrow Sibling( Richard, KingJohn )$$

predicate      term      term      predicate      term      term

# Syntax of First-order Logic: Complex Sentences

---

Built from atomic sentences using connectives

$$\neg S \quad S_1 \wedge S_2 \quad S_1 \vee S_2 \quad S_1 \Rightarrow S_2 \quad S_1 \Leftrightarrow S_2$$

(as in propositional logic)

Example

$$Sibling( KingJohn, Richard ) \Rightarrow Sibling( Richard, KingJohn )$$

atomic sentence      atomic sentence

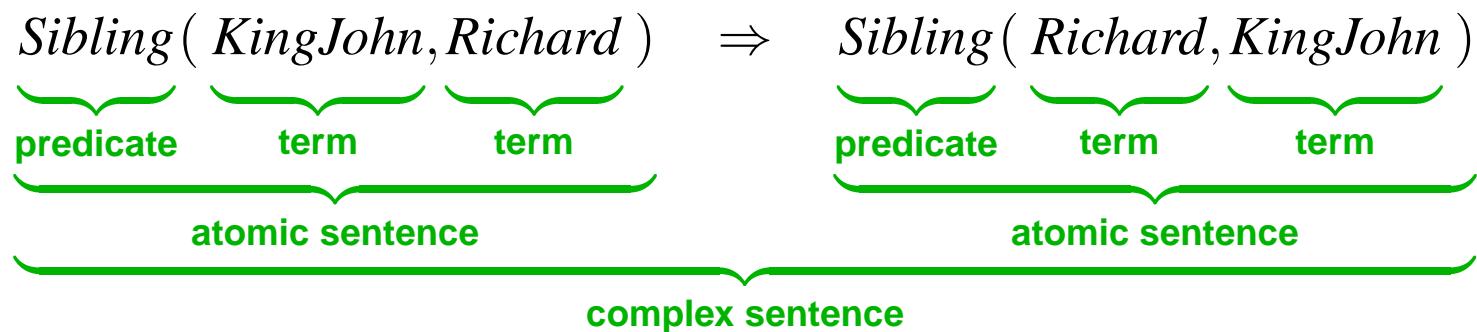
# Syntax of First-order Logic: Complex Sentences

## Built from atomic sentences using connectives

$\neg S$        $S_1 \wedge S_2$        $S_1 \vee S_2$        $S_1 \Rightarrow S_2$        $S_1 \Leftrightarrow S_2$

**(as in propositional logic)**

## Example



# Semantics in First-order Logic

---

## Models of first-order logic

**Sentences are true or false with respect to models, which consist of**

- a **domain** (also called universe)
- an **interpretation**

# Semantics in First-order Logic

---

## Models of first-order logic

**Sentences are true or false with respect to models, which consist of**

- a **domain** (also called universe)
- an **interpretation**

### Domain

**A non-empty (finite or infinite) set of arbitrary elements**

# Semantics in First-order Logic

---

## Models of first-order logic

Sentences are true or false with respect to models, which consist of

- a **domain** (also called universe)
- an **interpretation**

### Domain

A non-empty (finite or infinite) set of arbitrary elements

### Interpretation

Assigns to each

- constant symbol: a domain element
- predicate symbol: a relation on the domain (of appropriate arity)
- function symbol: a function on the domain (of appropriate arity)

# Semantics in First-order Logic

---

## Definition

An **atomic sentence**

$$\textit{predicate} ( \textit{term}_1, \dots, \textit{term}_n )$$

is true in a certain model (that consists of a domain and an interpretation)

iff

the domain elements that are the interpretations of  $\textit{term}_1, \dots, \textit{term}_n$   
are in the relation that is the interpretation of  $\textit{predicate}$

# Semantics in First-order Logic

---

## Definition

An **atomic sentence**

*predicate ( term<sub>1</sub>, … , term<sub>n</sub> )*

is true in a certain model (that consists of a domain and an interpretation)

iff

the domain elements that are the interpretations of term<sub>1</sub>, … , term<sub>n</sub>  
are in the relation that is the interpretation of predicate

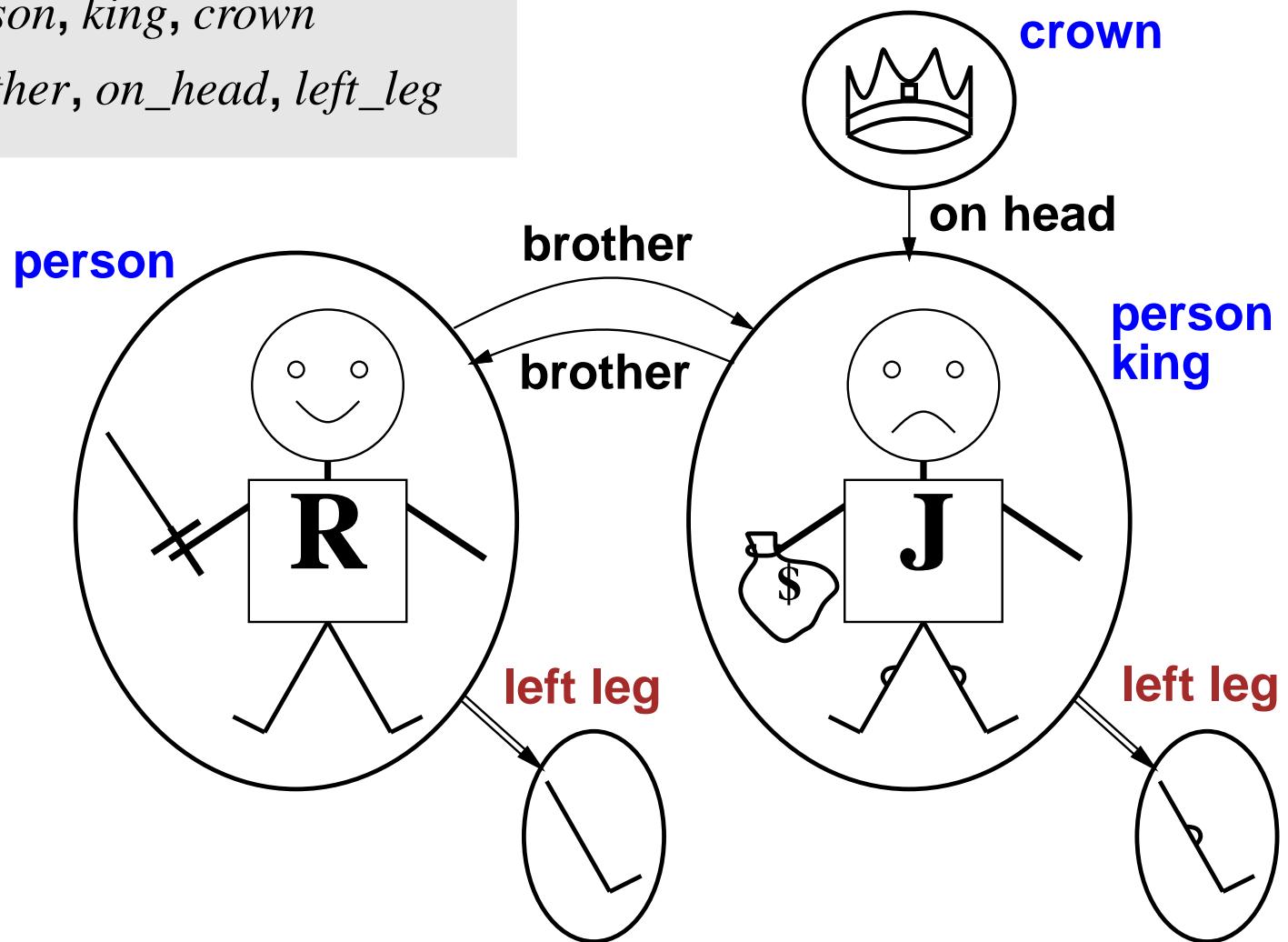
The truth value of a **complex sentence** in a model  
is computed from the truth-values of its atomic sub-sentences  
in the same way as in propositional logic

# Models for First-order Logic: Example

**Constants:** *KingJohn, Richard*

**Predicates:** *person, king, crown*

**Functions:** *brother, on\_head, left\_leg*



# Universal Quantification: Syntax

---

## Syntax

$\forall$  *variables sentence*

# Universal Quantification: Syntax

---

## Syntax

$\forall \text{ variables sentence}$

## Example

“Everyone studying in Koblenz is smart:

$$\forall \underbrace{x}_{\text{variables}} \underbrace{(StudiesAt(x, \text{Koblenz}) \Rightarrow Smart(x))}_{\text{sentence}}$$

# Universal Quantification: Semantics

---

## Semantics

$\forall xP$  is true in a model

iff

for all domain elements  $d$  in the model:

$P$  is true in the model when  $x$  is interpreted by  $d$

# Universal Quantification: Semantics

---

## Semantics

$\forall xP$  is true in a model

iff

for all domain elements  $d$  in the model:

$P$  is true in the model when  $x$  is interpreted by  $d$

## Intuition

$\forall xP$  is roughly equivalent to the conjunction of all instances of  $P$

# Universal Quantification: Semantics

---

## Semantics

$\forall x P$  is true in a model

iff

for all domain elements  $d$  in the model:

$P$  is true in the model when  $x$  is interpreted by  $d$

## Intuition

$\forall x P$  is roughly equivalent to the conjunction of all instances of  $P$

**Example**  $\forall x \text{StudiesAt}(x, \text{Koblenz}) \Rightarrow \text{Smart}(x)$  equivalent to:

$$\begin{aligned} & \text{StudiesAt}(\text{KingJohn}, \text{Koblenz}) \Rightarrow \text{Smart}(\text{KingJohn}) \\ \wedge \quad & \text{StudiesAt}(\text{Richard}, \text{Koblenz}) \Rightarrow \text{Smart}(\text{Richard}) \\ \wedge \quad & \text{StudiesAt}(\text{Koblenz}, \text{Koblenz}) \Rightarrow \text{Smart}(\text{Koblenz}) \\ \wedge \quad & \dots \end{aligned}$$

# A Common Mistake to Avoid

---

## Note

⇒ is the main connective with  $\forall$

## Common mistake

Using  $\wedge$  as the main connective with  $\forall$

# A Common Mistake to Avoid

---

## Note

⇒ is the main connective with  $\forall$

## Common mistake

Using  $\wedge$  as the main connective with  $\forall$

## Example

Correct:  $\forall x (StudiesAt(x, Koblenz) \Rightarrow Smart(x))$

“Everyone who studies at Koblenz is smart”

# A Common Mistake to Avoid

---

## Note

⇒ is the main connective with  $\forall$

## Common mistake

Using  $\wedge$  as the main connective with  $\forall$

## Example

Correct:  $\forall x (\text{StudiesAt}(x, \text{Koblenz}) \Rightarrow \text{Smart}(x))$

“Everyone who studies at Koblenz is smart”

Wrong:  $\forall x (\text{StudiesAt}(x, \text{Koblenz}) \wedge \text{Smart}(x))$

“Everyone studies at Koblenz and is smart”, i.e.,

“Everyone studies at Koblenz and everyone is smart”

# Existential Quantification: Syntax

---

## Syntax

$\exists$  *variables sentence*

# Existential Quantification: Syntax

---

## Syntax

$\exists$  *variables sentence*

## Example

“Someone studying in Landau is smart:

$$\exists \underbrace{x}_{\text{variables}} \underbrace{(StudiesAt(x, Landau) \wedge Smart(x))}_{\text{sentence}}$$

# Existential Quantification: Semantics

---

## Semantics

$\exists xP$  is true in a model

iff

there is a domain element  $d$  in the model such that:

$P$  is true in the model when  $x$  is interpreted by  $d$

# Existential Quantification: Semantics

---

## Semantics

$\exists x P$  is true in a model

iff

there is a domain element  $d$  in the model such that:

$P$  is true in the model when  $x$  is interpreted by  $d$

## Intuition

$\exists x P$  is roughly equivalent to the disjunction of all instances of  $P$

# Existential Quantification: Semantics

---

## Semantics

$\exists x P$  **is true in a model**

iff

**there is a domain element  $d$  in the model such that:**

**$P$  is true in the model when  $x$  is interpreted by  $d$**

## Intuition

$\exists x P$  **is roughly equivalent to the disjunction of all instances of  $P$**

**Example**      $\exists x \text{StudiesAt}(x, \text{Landau}) \wedge \text{Smart}(x)$    **equivalent to:**

$$\begin{aligned} & \text{StudiesAt}(\text{KingJohn}, \text{Landau}) \wedge \text{Smart}(\text{KingJohn}) \\ \vee \quad & \text{StudiesAt}(\text{Richard}, \text{Landau}) \wedge \text{Smart}(\text{Richard}) \\ \vee \quad & \text{StudiesAt}(\text{Landau}, \text{Landau}) \wedge \text{Smart}(\text{Landau}) \\ \vee \quad & \dots \end{aligned}$$

# Another Common Mistake to Avoid

---

## Note

$\wedge$  is the main connective with  $\exists$

## Common mistake

Using  $\Rightarrow$  as the main connective with  $\exists$

# Another Common Mistake to Avoid

---

## Note

$\wedge$  is the main connective with  $\exists$

## Common mistake

Using  $\Rightarrow$  as the main connective with  $\exists$

## Example

Correct:  $\exists x (StudiesAt(x, Landau) \wedge Smart(x))$

“There is someone who studies at Landau and is smart”

# Another Common Mistake to Avoid

---

## Note

$\wedge$  is the main connective with  $\exists$

## Common mistake

Using  $\Rightarrow$  as the main connective with  $\exists$

## Example

Correct:  $\exists x (StudiesAt(x, Landau) \wedge Smart(x))$

“There is someone who studies at Landau and is smart”

Wrong:  $\exists x (StudiesAt(x, Landau) \Rightarrow Smart(x))$

“There is someone who, if he/she studies at Landau, is smart”

This is true if there is anyone not studying at Landau

# Properties of Quantifiers

---

**Quantifiers of same type commute**

$$\forall x \forall y \quad \text{is the same as} \quad \forall y \forall x$$

$$\exists x \exists y \quad \text{is the same as} \quad \exists y \exists x$$

# Properties of Quantifiers

---

Quantifiers of different type do NOT commute

$\exists x \forall y$     is **not the same as**     $\forall y \exists x$

## 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”

(Both hopefully true but different)

# Properties of Quantifiers

---

Quantifiers of different type do NOT commute

$\exists x \forall y$     is **not the same as**     $\forall y \exists x$

## Example

$\forall x \exists y Mother(x, y)$

“Everyone has a mother”    **(correct)**

$\exists y \forall x Mother(x, y)$

“There is a person who is the mother of everyone”    **(wrong)**

# Properties of Quantifiers

---

## Quantifier duality

$$\forall x \text{Likes}(x, \text{IceCream}) \quad \text{is the same as} \quad \neg \exists x \neg \text{Likes}(x, \text{IceCream})$$

$$\exists x \text{Likes}(x, \text{Broccoli}) \quad \text{is the same as} \quad \neg \forall x \neg \text{Likes}(x, \text{Broccoli})$$

# Fun with Sentences

---

- “**Brothers are siblings**”

$$\forall x, y (\text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y))$$

# Fun with Sentences

---

- “**Brothers are siblings**”

$$\forall x, y (\text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y))$$

- “**Sibling**” is symmetric

$$\forall x, y (\text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x))$$

# Fun with Sentences

---

- “**Brothers are siblings**”

$$\forall x, y (\text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y))$$

- “**Sibling**” is symmetric

$$\forall x, y (\text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x))$$

- “**One’s mother is one’s female parent**”

$$\forall x, y (\text{Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)))$$

# Fun with Sentences

---

- “**Brothers are siblings**”

$$\forall x, y (\text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y))$$

- “**Sibling**” is symmetric

$$\forall x, y (\text{Sibling}(x, y) \Leftrightarrow \text{Sibling}(y, x))$$

- “**One’s mother is one’s female parent**”

$$\forall x, y (\text{Mother}(x, y) \Leftrightarrow (\text{Female}(x) \wedge \text{Parent}(x, y)))$$

- “**A first cousin is a child of a parent’s sibling**”

$$\forall x, y (\text{FirstCousin}(x, y) \Leftrightarrow \exists p, ps (\text{Parent}(p, x) \wedge \text{Sibling}(ps, p) \wedge \text{Parent}(ps, y)))$$

# Equality

---

## Semantics

$term_1 = term_2$  is true under a given interpretation

if and only if

$term_1$  and  $term_2$  have the same interpretation

# Equality

---

## Example

**Definition of (full) sibling in terms of *Parent***

$$\begin{aligned} \forall x, y \text{ } Sibling(x, y) \Leftrightarrow & (\neg(x = y) \wedge \\ & \exists m, f (\neg(m = f) \wedge \\ & Parent(m, x) \wedge Parent(f, x) \wedge \\ & Parent(m, y) \wedge Parent(f, y))) \end{aligned}$$

# Properties of First-order Logic

---

## Important notions

- validity
- satisfiability
- unsatisfiability
- entailment

are defined for first-order logic in the same way as for propositional logic

# Properties of First-order Logic

---

## Important notions

- validity
- satisfiability
- unsatisfiability
- entailment

are defined for first-order logic in the same way as for propositional logic

## Calculi

There are sound and complete calculi for first-order logic (e.g. resolution)

- Whenever  $KB \vdash \alpha$ , it is also true that  $KB \models \alpha$
- Whenever  $KB \models \alpha$ , it is also true that  $KB \vdash \alpha$

But these calculi **CANNOT** decide validity, entailment, etc.

# Properties of First-order Logic

---

In propositional logic

Validity, satisfiability, unsatisfiability are **decidable**

In first-order logic

The set of valid, and the set of unsatisfiable formulas are **enumerable**

The set of satisfiable formulas is **NOT EVEN enumerable**