# Artificial Intelligence

## III Knowledge and Reasoning

## III.2 First-Order Logic

F. C. Langbein

School of Computer Science
and Informatics
Cardiff University

1.4

## Overview

- First-Order Logic
  - Syntax
  - Semantics
- Basic set theory
- Higher order logics



## First-Order Logic

- Propositional logic assumes world contains facts
- First-order logic (FOL) assumes world contains
  - Objects as elements in the world:
    people, houses, colours, numbers, theories, wars, . . .
  - Functions mapping $k$ objects onto one object:
    father_of, best_friend, one_more_than, head_of, . . .
  - $k$-ary relations between objects which are true or false:
    prime, colour, shape, brother, bigger, owns, . . .
- Variables enable use of knowledge templates
  - ➜ *Reusable* knowledge:
    instantiation with new objects gives new facts
- *Deterministic relational model*
  - Used in first-order logic reasoning, planning

# FOL Syntax: Basic Elements

- Constants: KingJohn, 2, CardiffUniversity, ... (to represent objects)
- Functions: Sqrt, LeftLegOf, ... (to represent functions)
- Predicates: Brother, $>$, ... (to represent relations between objects)
- Connectives: $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$ (to construct complex sentences)
- Variables: $x, y, a, b, \ldots$ (to express knowledge templates)
- Quantifiers: $\forall, \exists$ (to declare variables)
- Equality: $=$ (to determine object equality)



# FOL Syntax: Atomic Sentences

- `function`, `constant`, `predicate` are capitalised strings
- `variable` is non-capital character
- 
```
term = (function, "(", [ term, {",", term} ], ")") |
        constant | variable
```
  - LeftLegOf(Richard)
  - Length(LeftLegOf($x$))
- 
```
atom = (predicate,"(", [ term, {",", term} ], ")") |
        term, "=", term
```
  - Brother(KingJohn, RichardTheLionheart)
  - $>$ (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))
  - Richard $=$ RichardTheLionheart



# FOL Syntax: Complex Sentences

- Complex sentences are made from atomic sentences using connectives and quantifiers
  - 
```
sentence = (sentence,binary-connective,sentence)|
           (unary-connective,sentence)|
           (quantifier,variable,sentence)|
           atom
```
  - $\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$
  - $\forall x\ S(x), \quad \exists x\ S(x)$
    - $> (1,2) \vee \neg \geq (1,2)$
    - $\neg \leq$ (Length(LeftLegOf(KingJohn)), $0$)
    - $\forall x$ Sibling(KingJohn, $x$) $\Rightarrow$ Sibling($x$, KingJohn)
- Parenthesis for operator precedence

## FOL Semantics: Basic Truth

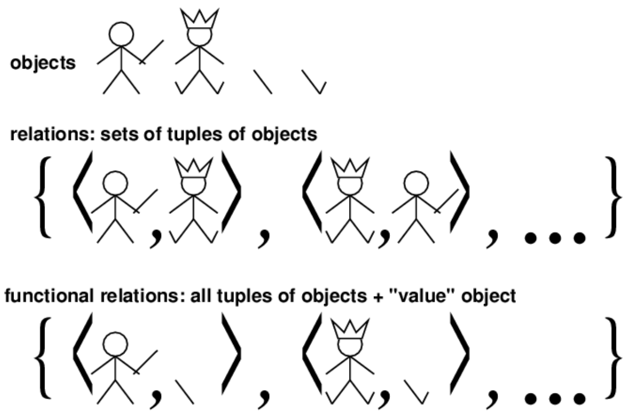
- Sentences are true in a model described by
  objects & functions & relations

- An atomic sentence $\textit{predicate}(\textit{term}_1, \ldots, \textit{term}_n)$ is *true* iff
  objects referred to by $\textit{term}_1, \ldots, \textit{term}_n$
  are in the relation referred to by *predicate*
  - Variables, equality and quantifiers not yet included

- *Interpretation* of model specifies referents
  - Constant symbols $\leftrightarrow$ objects in the world
  - Predicate symbols $\leftrightarrow$ relations in the world
  - Function symbols $\leftrightarrow$ functional relations in the world

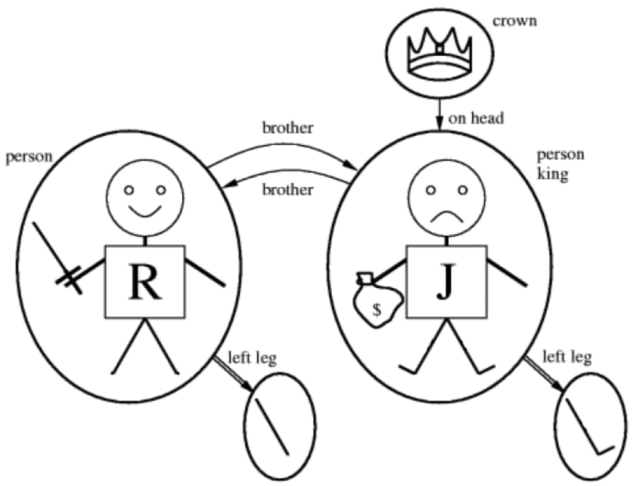


## FOL Model Example






## FOL Model Example



## Universal Quantification

- Universally quantified variables: $\forall \langle variable \rangle \quad \langle sentence \rangle$
  (sentence can use variable as term)
  - "Everyone at Cardiff is smart"
    $$\forall x \ \text{At}(x, \text{Cardiff}) \Rightarrow \text{Smart}(x)$$
  - $\forall x \ P$ is *conjunction of instantiations* of $P$
    $$\begin{aligned} &(\text{At}(\text{KingJohn}, \text{Cardiff}) \Rightarrow \text{Smart}(\text{KingJohn})) \\ \wedge\ &(\text{At}(\text{Richard}, \text{Cardiff}) \Rightarrow \text{Smart}(\text{Richard})) \\ \wedge\ &(\text{At}(\text{Cardiff}, \text{Cardiff}) \Rightarrow \text{Smart}(\text{Cardiff})) \\ \wedge\ &\ldots \end{aligned}$$
- Typically, $\Rightarrow$ is the main connective with $\forall$
  - *Common MISTAKE*: $\wedge$ is main connective with $\forall$
  - $\forall x \ \text{At}(x, \text{Cardiff}) \wedge \text{Smart}(x)$
    means "Everyone is at Cardiff and everyone is smart"!



## Existential Quantification

- Existentially quantified variables: $\exists \langle variable \rangle \quad \langle sentence \rangle$
  (sentence can use variable as term)
  - "Someone at Cambridge is smart"
    $$\exists x \ \text{At}(x, \text{Cambridge}) \wedge \text{Smart}(x)$$
  - $\exists x \ P$ is *disjunction of instantiations* of $P$
    $$\begin{aligned} &(\text{At}(\text{KingJohn}, \text{Cambridge}) \wedge \text{Smart}(\text{KingJohn})) \\ \vee\ &(\text{At}(\text{Richard}, \text{Cambridge}) \wedge \text{Smart}(\text{Richard})) \\ \vee\ &(\text{At}(\text{Cambridge}, \text{Cambridge}) \wedge \text{Smart}(\text{Cambridge})) \\ \vee\ &\ldots \end{aligned}$$
- Typically, $\wedge$ is the main connective with $\exists$
  - *Common MISTAKE*: $\Rightarrow$ is main connective with $\exists$
  - $\exists x \ \text{At}(x, \text{Cambridge}) \Rightarrow \text{Smart}(x)$
    is true if there is anyone who is not at Cambridge!



## Properties of Quantifiers

- $\forall x \ \forall y$ is the same as $\forall y \ \forall x$
- $\exists x \ \exists y$ is the same as $\exists y \ \exists x$
  - ➔ Write $\forall x, y, \ldots$ and $\exists x, y, \ldots$ for simplicity
- $\exists x \ \forall y$ is *not* the same as $\forall y \ \exists x$
  - $\exists x \ \forall y \ \text{Loves}(x, y)$
    - "There is someone who loves everyone in the world"
  - $\forall y \ \exists x \ \text{Loves}(x, y)$
    - "Everyone in the world is loved by at least one person"
- Quantifier duality: each can be expressed using the other
  - $\forall x \ \text{Likes}(x, \text{IceCream}) \equiv \neg \exists x \ \neg \text{Likes}(x, \text{IceCream})$
  - $\exists x \ \text{Likes}(x, \text{Broccoli}) \equiv \neg \forall x \ \neg \text{Likes}(x, \text{Broccoli})$

## Equality

- *term*$_1$ $=$ *term*$_2$ is true under a given interpretation iff *term*$_1$ and *term*$_2$ refer to the same object
- Examples
  - $1 = 2$: true if both constants refer to the same object
  - $2 = 2$: true in any model
  - $\forall x \; *(\text{Sqrt}(x), \text{Sqrt}(x)) = x$: true in certain models
- Example: definition of (full) sibling in terms of parent
  - $\forall x, y \; \text{Sibling}(x, y) \Leftrightarrow$
    $[\; \neg(x = y) \wedge$
    $(\; \exists m, f \;\; \neg(m = f) \wedge$
    $\text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge$
    $\text{Parent}(m, y) \wedge \text{Parent}(f, y) \;\; ) \;\; ]$



## Kinship Examples

- "Brothers are siblings"
  - $\forall x, y \; \text{Brother}(x, y) \Rightarrow \text{Sibling}(x, y)$
- "Sibling is symmetric"
  - $\forall x, y \; \text{Sibling}(x, y) \Rightarrow \text{Sibling}(y, x)$
- "One's mother is one's female parent"
  - $\forall x, y \; \text{Mother}(y) = x \Rightarrow \text{Female}(x) \wedge \text{Parent}(x, y)$
- "A first cousin is a child of a parent's sibling"
  - $\forall x, y \; \text{FirstCousin}(x, y) \Rightarrow$
    $(\exists p, s \; \text{Parent}(p, x) \wedge \text{Sibling}(s, p) \wedge \text{Parent}(s, y))$
    $\wedge \, x \neq y$



## Kinship Examples

- "There is at least one child"
  - $\exists x \; \text{Child}(x)$
- "There is at most one child"
  - $\forall x, y \; \text{Child}(x) \wedge \text{Child}(y) \Rightarrow x = y$
- "There is exactly one child"
  - $\exists x \; \text{Child}(x) \wedge (\forall x, y \; \text{Child}(x) \wedge \text{Child}(y) \Rightarrow x = y)$

## General Set Theory: Axioms

- General, not Zermelo-Fraenkel, sufficient for *finite* sets
- *Extensionality axiom*

$$\forall a, b, x\ (\mathrm{In}(x, a) \Leftrightarrow \mathrm{In}(x, b)) \Rightarrow a = b$$

  - Usually, $\mathrm{In}(x, a)$ is $x \in a$
- *Axiom scheme of specification* for any predicate $P$

$$\forall a\ \exists b\ \forall x\ \mathrm{In}(x, b) \Leftrightarrow \mathrm{In}(x, a) \wedge P(x)$$

  - Usually $b = \{x \in a : P(x)\}$
  - Special cases: $P =$ true or $P =$ false
  - ➔ *Empty set axiom*: $\exists e\ \forall x\ \neg\mathrm{In}(x, e)$
    - $e$ is unique because of extensionality axiom
- *Adjunction axiom*

$$\forall a, b\ \exists w\ \forall x\ \mathrm{In}(x, w) \Leftrightarrow \mathrm{In}(x, a) \vee x = b$$



## General Set Theory: Sets

- *List notation*: $\{x_1, x_2, \ldots, x_n\}$ denotes a set $A$ iff

$$\forall y\ \mathrm{In}(y, A) \Leftrightarrow y = x_1 \vee y = x_2 \vee \cdots \vee y = x_n$$

  - Uniqess of $A$ not specified
    - Extensionality guarantees uniqueness
- *Brace notation*: $\{x : P(x)\}$ denotes a set $A$ iff

$$(\forall y\ \mathrm{In}(y, A) \Leftrightarrow P(y)) \wedge (\forall b, z\ (\mathrm{In}(z, b) \Leftrightarrow P(z)) \Rightarrow b = A)$$

  - Uniqess of $A$ included
    - Specification guarantees that set $A$ exists for every $P$
    - Extensionality guarantees second conjunct is satisfied whenever first conjuct is satisfied
    - ➔ $A$ always exists and is unique



## General Set Theory: Subsets

- *Subset*

$$\forall a, b\ \mathrm{SubsetEq}(a, b) \Leftrightarrow (\forall x\ \mathrm{In}(x, a) \Rightarrow \mathrm{In}(x, b))$$

- *Not subset*

$$\forall a, b\ \mathrm{NotSubset}(a, b) \Leftrightarrow (\exists x\ \mathrm{In}(x, a) \wedge \neg\mathrm{In}(x, b))$$

- *True subset*

$$\forall a, b\ \mathrm{SubsetNotEq}(a, b) \Leftrightarrow (\forall x\ \mathrm{In}(x, a) \Rightarrow \mathrm{In}(x, b)) \wedge (\exists y\ \mathrm{In}(y, b) \wedge \neg\mathrm{In}(y, a))$$

## General Set Theory: Operations

- *Union*

$$\forall a, b, x \; \mathsf{In}(x, \mathsf{Union}(a, b)) \Leftrightarrow \mathsf{In}(x, a) \vee \mathsf{In}(x, b)$$

- *Intersection*

$$\forall a, b, x \; \mathsf{In}(x, \mathsf{Intersection}(a, b)) \Leftrightarrow \mathsf{In}(x, a) \wedge \mathsf{In}(x, b)$$

- *Difference* (relative complement)

$$\forall a, b, x \; \mathsf{In}(x, \mathsf{Difference}(a, b)) \Leftrightarrow \mathsf{In}(x, a) \wedge \neg \mathsf{In}(x, b)$$

- *Power set*

$$\forall a, x \; \mathsf{In}(x, \mathsf{PowerSet}(a)) \Leftrightarrow \mathsf{SubsetEq}(x, a)$$

  – Note, $|\mathsf{PowerSet}(a)| > |a|$ (Cantor, 1891)



## Russell's Paradox

- Let $R$ be the *set of all sets that do not contain themself*

$$R = \{a : \neg \mathsf{In}(a, a)\}$$

  - From brace notation for $R$ (first conjunct)

$$\forall x \; \mathsf{In}(x, R) \Leftrightarrow P(x) \quad \text{with } P(x) = \neg \mathsf{In}(x, x)$$

  - Hence

$$\forall x \; \mathsf{In}(x, R) \Leftrightarrow \neg \mathsf{In}(x, x)$$

  - Instantiate $x$ by $R$

$$\mathsf{In}(R, R) \Leftrightarrow \neg \mathsf{In}(R, R)$$

  ⇒ *Contradiction*

- Not allowing the above $R$ is insufficient
  – Many *reciprocal* $P$, e.g. $P(x) = \neg \exists z \; \mathsf{In}(x, z) \wedge \mathsf{In}(z, x)$

  ⇒ Be very vareful with things referring to themself!



## Higher Order Logics

- Propositional logic: facts
- Higher-order logic
  - First-order logic: quantify over objects
  - Second-order logic: quantify over predicates
  - Higher order $k > 2$ logic:
    order $k$ predicate has order $k - 1$ predicate arguments

    ⇒ Proofs become messy in higher-order logics
- *Type theory*: hierarchy of types assigned to entities
  - Alternative to naive set theory
    – Roughly, a type corresponds to a set
  - E.g. *typed $\lambda$-calculus*
    – Foundation of programming languages