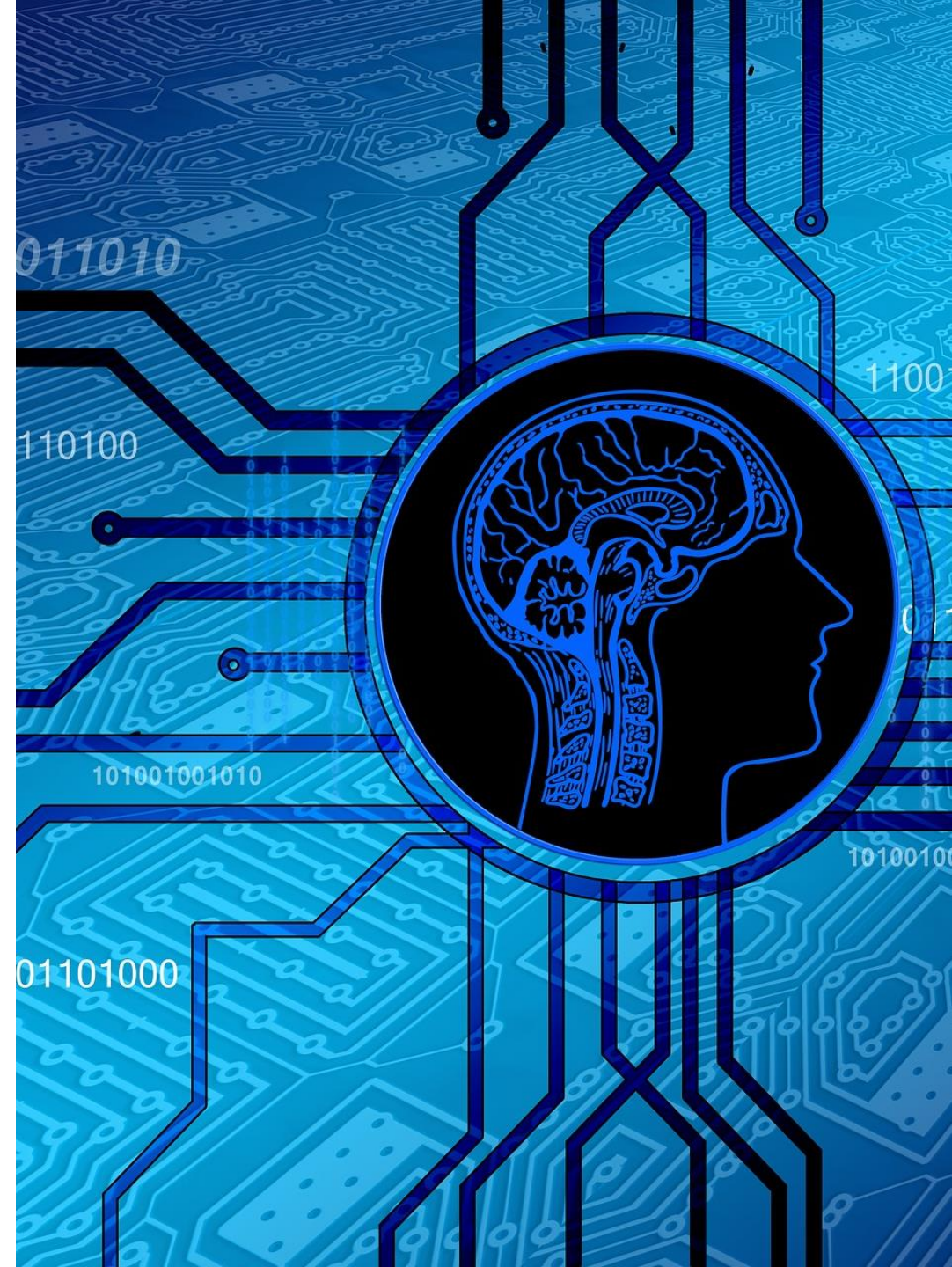


First-order Logic

Informatics 2D: Reasoning and Agents
Lecture 10

Adapted from slides provided by Dr Petros Papapanagiotou



Pros and cons of Propositional Logic

✓ Declarative

✓ Partial/disjunctive/negated information
◦ (unlike most data structures and databases!)

✓ Compositional

The meaning of $B_{1,1} \wedge P_{1,2}$ is derived from that of $B_{1,1}$ and of $P_{1,2}$

⊗ Meaning is context-independent
◦ (unlike natural language,
where meaning depends on context)

⊗ Very limited expressive power
◦ (unlike natural language)
◦ for example, we cannot say "*pits cause breezes in adjacent squares*", except by writing one sentence for each square

First-order logic (FOL)

- Propositional logic assumes the world contains **atomic facts**.
 - Non-structured propositional symbols, usually finitely many.
- FOL assumes the world contains:

Objects

- people, houses, numbers, colours, football games, wars, ...

Relations

- red, round, prime, brother of, bigger than, part of, comes between, ...

Functions

- father of, best friend, one more than, plus, ...

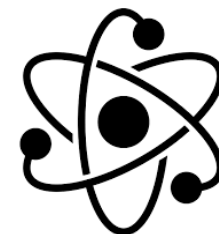
Syntax of FOL: Basic elements

Constants	• <i>KingJohn, 2, UoE,...</i>
Predicates	• <i>Brother, >,...</i>
Functions	• <i>Sqrt, LeftLegOf,...</i>
Variables	• <i>x, y, a, b,...</i>
Connectives	• $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
Equality	• $=$
Quantifiers	• \forall, \exists

Syntax of FOL: Basic elements



Constants	• <i>KingJohn</i> _{/0} , <i>2</i> _{/0} , <i>UoE</i> _{/0} , ...
Predicates	• <i>Brother</i> _{/2} , <i>></i> _{/2} , ...
Functions	• <i>Sqrt</i> _{/1} , <i>LeftLegOf</i> _{/1} , <i>+</i> _{/2} , ...
Variables	• <i>x</i> , <i>y</i> , <i>a</i> , <i>b</i> , ...
Connectives	• \neg , \Rightarrow , \wedge , \vee , \Leftrightarrow
Equality	• $=$
Quantifiers	• \forall , \exists



Atomic formulae

Atomic formula = *predicate* ($term_1, \dots, term_n$)
or $term_1 = term_2$

Term = *function* ($term_1, \dots, term_n$)
or *constant* or *variable*

Examples:

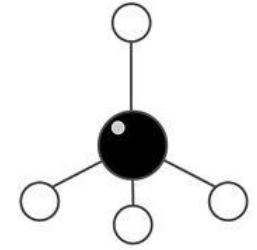
- *Brother*(*KingJohn*, *Richard*)
- $>(\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})))$

predicate

functions

constants

Complex formulae



Complex formulae are made from atomic formulae using connectives

$$\neg P \quad P \wedge Q \quad P \vee Q \quad P \Rightarrow Q \quad P \Leftrightarrow Q$$

Examples:

$$\text{Sibling}(\text{KingJohn}, \text{Richard}) \Rightarrow \text{Sibling}(\text{Richard}, \text{KingJohn})$$

$$>(1,2) \vee \leq(1,2)$$

$$>(1,2) \wedge \neg >(1,2)$$

Semantics of first-order logic



Formulae are mapped to an **interpretation**.



An interpretation is called a model of a set of formulae when all the formulae are **true** in the interpretation.

Semantics of first-order logic

- An **interpretation** contains objects (**domain elements**) and relations between them. Mapping is as follows :

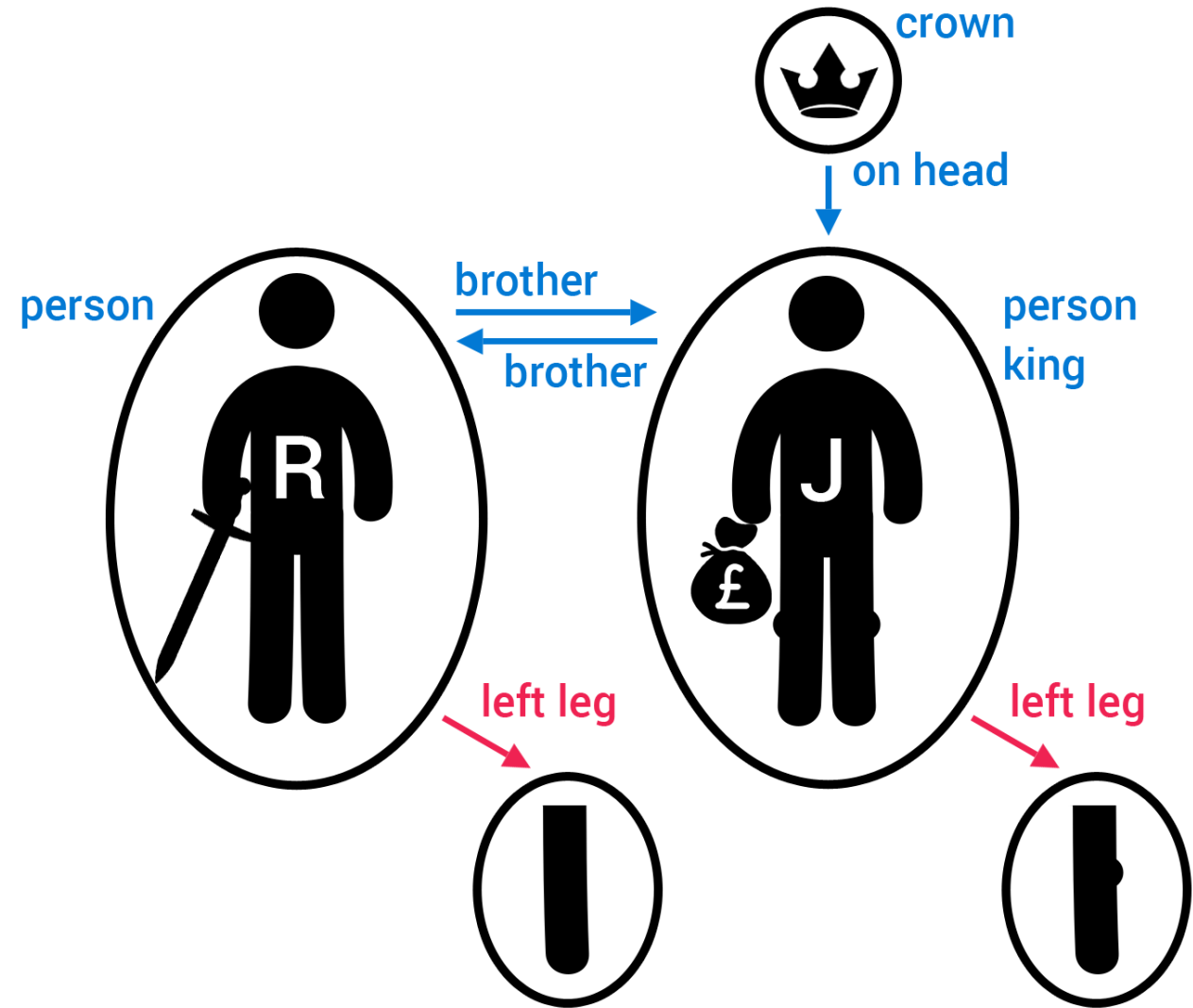
constant symbols \mapsto **objects**

predicate symbols \mapsto **relation**

function symbols \mapsto **functions**

- An atomic formula *predicate(term₁,...,term_n)* is **true**
iff the **objects** referred to by *term₁,...,term_n*
are in the **relation** referred to by *predicate*.

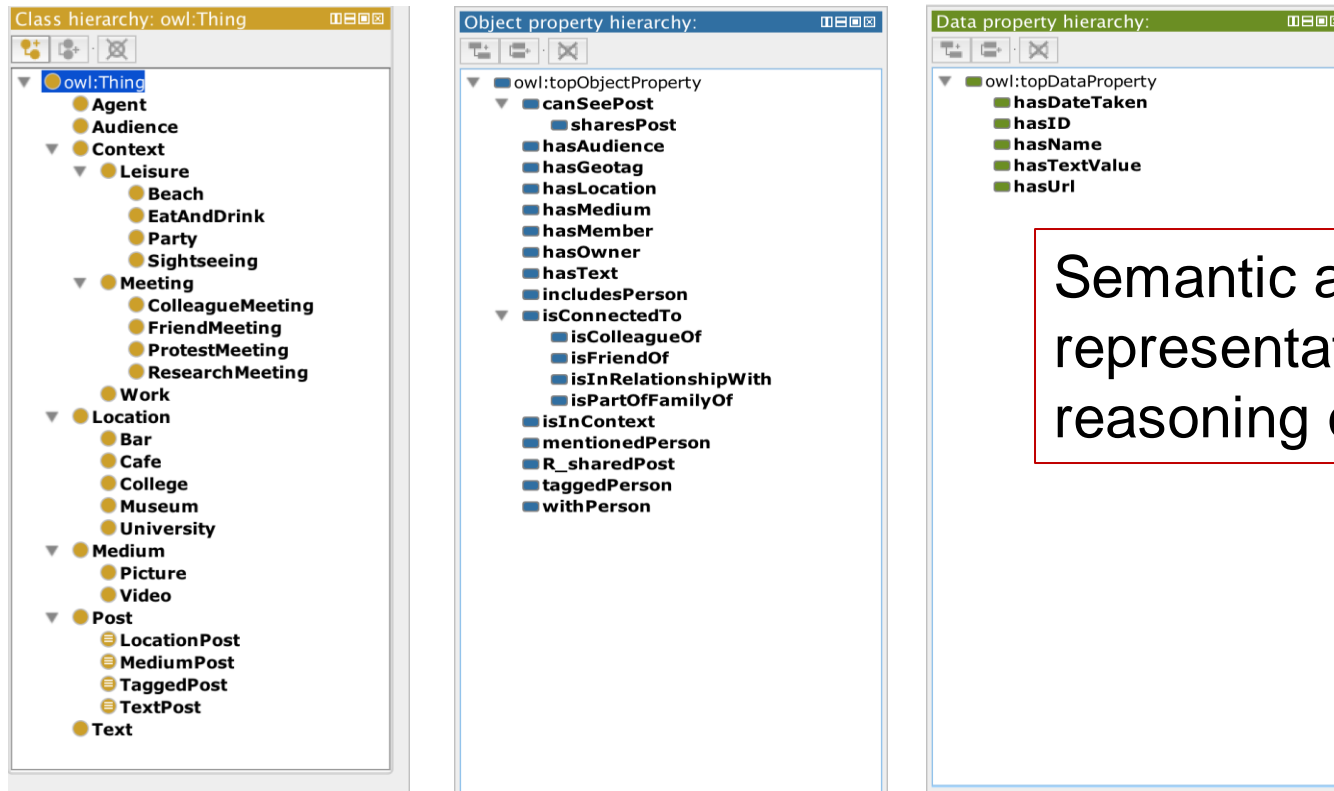
Interpretations for FOL: Example



Brother(KingJohn, Richard)

>(Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))

A Social Network **Ontology** based on Description Logics



Semantic approaches rely on a knowledge representation, such as an **ontology**, for reasoning on the content.



PriGuard: A Semantic Approach to Detect Privacy Violations in Online Social Networks. Nadin Kökciyan and Pinar Yolum. IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 28, no. 10, pp. 2724-2737. 2016.

Adding further inference: Rules

$$I_1: \text{isFriendOf}(\text{?user}, \text{?tp}) \rightarrow \text{inFRContext}(\text{?user}, \text{?tp})$$

$$I_2: \text{workAt}(\text{?user}, \text{?office}), \text{workAt}(\text{?tp}, \text{?office}), \\ \text{hasRole}(\text{?tp}, \text{?role}), \text{isColleagueOf}(\text{?user}, \text{?tp}) \\ \rightarrow \text{inPRContext}(\text{?user}, \text{?tp})$$

$$I_3: \text{Emergency}(\text{?em}), \text{isInEmergency}(\text{?user}, \text{?em}), \\ \text{hasRole}(\text{?tp}, \text{:em-responder}) \rightarrow \text{inEMContext}(\text{?user}, \text{?tp})$$

$$P_1: \text{owns}(\text{:bob}, \text{:bob-loc}), \text{inPRContext}(\text{:bob}, \text{?tp}) \\ \rightarrow \text{allow}(\text{?tp}, \text{:bob-loc})$$

$$P_2: \text{owns}(\text{:bob}, \text{:bob-loc}), \text{inFRContext}(\text{:bob}, \text{?tp}) \\ \rightarrow \text{disallow}(\text{?tp}, \text{:bob-loc})$$

$$A_1: \text{owns}(\text{:bob}, \text{:bob-loc}), \text{inPRContext}(\text{:bob}, \text{:alice}), \\ \text{hasAR}(\text{:alice}, \text{:bob-loc}) \rightarrow \text{allow}(\text{:alice}, \text{:bob-loc})$$

$$A_2: \text{owns}(\text{:bob}, \text{:bob-mobile}), \text{inEMContext}(\text{:bob}, \text{:alice}), \\ \text{hasAR}(\text{:alice}, \text{:bob-mobile}) \rightarrow \text{allow}(\text{:alice}, \text{:bob-mobile})$$

$$A_3: \text{owns}(\text{:bob}, \text{:bob-loc}), \text{inFRContext}(\text{:bob}, \text{:alice}), \\ \text{hasAR}(\text{:alice}, \text{:bob-loc}) \rightarrow \text{disallow}(\text{:alice}, \text{:bob-loc})$$



Contextual Integrity for Argumentation-based Privacy Reasoning. Gideon Ogunniye and Nadin Kökciyan. The International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2023, Accepted.



Universal quantification

$$\forall \langle \text{variables} \rangle. \langle \text{formula} \rangle$$

- But will often write $\forall x, y. P$ for $\forall x. \forall y. P$
- Example: *Everyone at UoE is smart*: $\forall x. \text{At}(x, \text{UoE}) \Rightarrow \text{Smart}(x)$
- $\forall x. P$ is **true** in an interpretation m iff P is **true** with x being **each** possible object in the interpretation.
- Roughly speaking, equivalent to the **conjunction** of **instantiations** of P
 - $\text{At}(\text{KingJohn}, \text{UoE}) \Rightarrow \text{Smart}(\text{KingJohn})$
 - $\wedge \text{At}(\text{Richard}, \text{UoE}) \Rightarrow \text{Smart}(\text{Richard})$
 - $\wedge \text{At}(\text{UoE}, \text{UoE}) \Rightarrow \text{Smart}(\text{UoE}) \wedge \dots$



Existential quantification

$\exists \langle \text{variables} \rangle. \langle \text{formula} \rangle$

- But will often write $\exists x, y. P$ for $\exists x. \exists y. P$
- Example: *Someone at UoE is smart*: $\exists x. \text{At}(x, \text{UoE}) \wedge \text{Smart}(x)$

➤ $\exists x. P$ is **true** in an interpretation m iff P is **true** with x being **some** possible object in the interpretation.

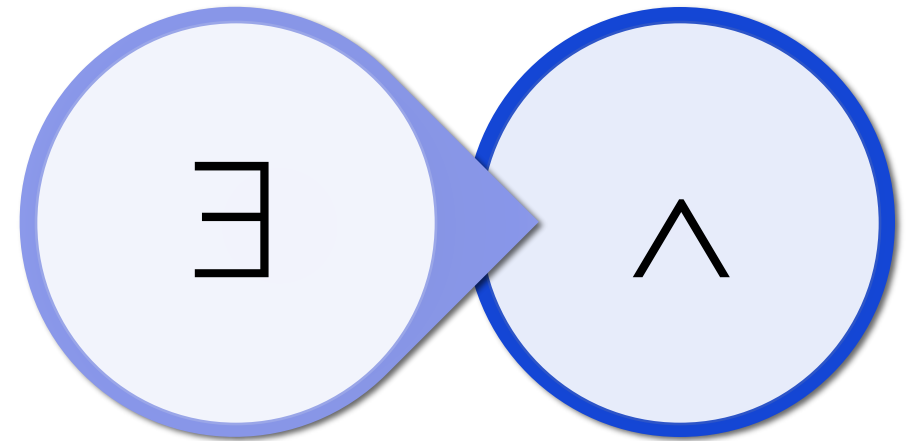
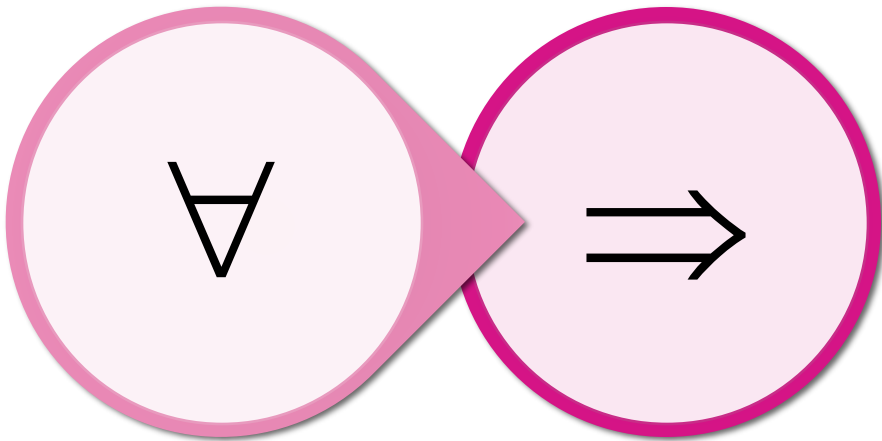
➤ Roughly speaking, equivalent to the **disjunction** of **instantiations** of P

$\text{At}(\text{KingJohn}, \text{UoE}) \wedge \text{Smart}(\text{KingJohn})$

✓ $\text{At}(\text{Richard}, \text{UoE}) \wedge \text{Smart}(\text{Richard})$

✓ $\text{At}(\text{UoE}, \text{UoE}) \wedge \text{Smart}(\text{UoE}) \vee \dots$

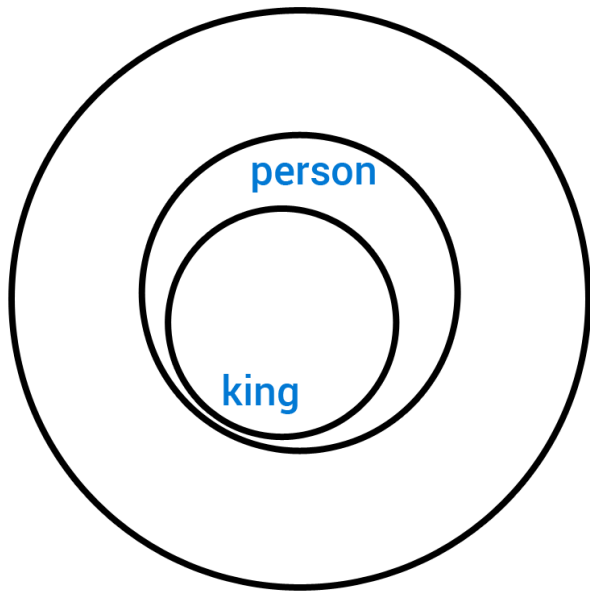
Rule of thumb



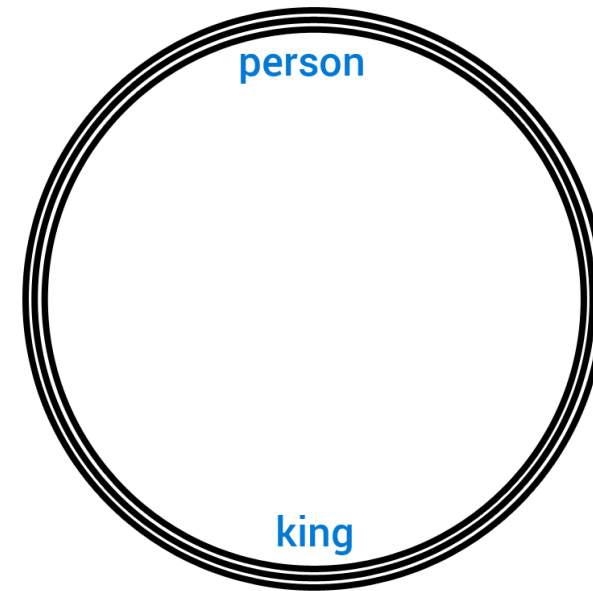


Common mistakes

$\forall x. \text{King}(x) \Rightarrow \text{Person}(x)$



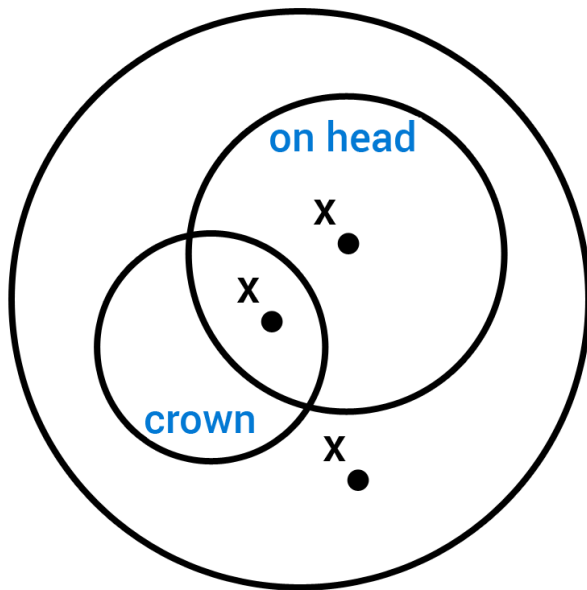
$\forall x. \text{King}(x) \wedge \text{Person}(x)$



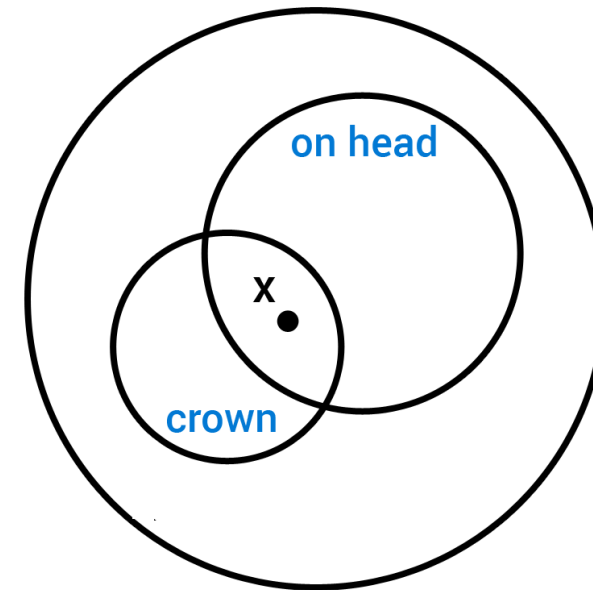


Common mistakes

$\exists x. \text{Crown}(x) \Rightarrow \text{OnHead}(x, \text{John})$



$\exists x. \text{Crown}(x) \wedge \text{OnHead}(x, \text{John})$



Properties of quantifiers



➤ $\forall x. \forall y.$ is the same as $\forall y. \forall x.$

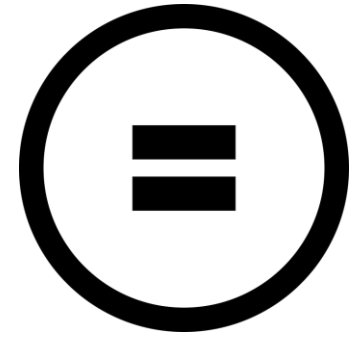
$\exists x. \exists y.$ is the same as $\exists y. \exists x.$

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

- $\exists x. \forall y. \text{Loves}(x, y)$: "There is a person 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
if and only if $term_1$ and $term_2$ refer to the same object.

➤ Example: Definition of *Sibling* in terms of *Parent*:

$$\forall x, y. Sibling(x, y) \Leftrightarrow (\neg(x = y) \wedge$$

$$\exists m, f. \neg(m = f) \wedge$$

$$Parent(m, x) \wedge Parent(f, x) \wedge Parent(m, y) \wedge Parent(f, y))$$

Example: Kinship domain

Brothers are siblings.

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

One's mother is one's female parent.

- $\forall m, c. \text{Mother}(c) = m \Leftrightarrow (\text{Female}(m) \wedge \text{Parent}(m, c))$

"Sibling" is symmetric.

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

"Parent" and "Child" are inverse relations.

- $\forall x, y. \text{Parent}(x, y) \Leftrightarrow \text{Child}(y, x)$

Example: Set domain

$$\forall s. \text{Set}(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2. \text{Set}(s_2) \wedge s = \{x|s_2\})$$

$$\neg \exists x, s. \{x|s\} = \{\}$$

$$\forall x, s. x \in s \Leftrightarrow s = \{x|s\}$$

$$\forall x, s. x \in s \Leftrightarrow [\exists y, s_2. (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$$

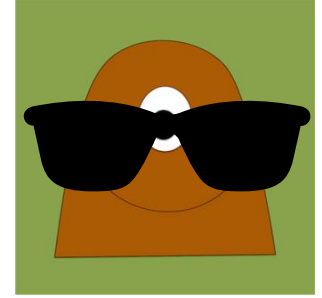
$$\forall s_1, s_2. s_1 \subseteq s_2 \Leftrightarrow (\forall x. x \in s_1 \Rightarrow x \in s_2)$$

$$\forall s_1, s_2. (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$$

$$\forall x, s_1, s_2. x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$$

$$\forall x, s_1, s_2. x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$$

Interacting with FOL KBs



- Suppose a Wumpus-world agent using a FOL KB perceives:
a **smell** and a **breeze** (but **no glitter**) at $t=5$:

`Tell(KB, Percept([Smell, Breeze, None], 5))`

`Ask(KB, $\exists a.$ BestAction($a, 5$))`

*i.e., does the KB entail some **best action** at $t=5$?*

Substitution

- Given a sentence S and a substitution σ ,

- $S\sigma$ denotes the result of “plugging” σ into S ; e.g.,

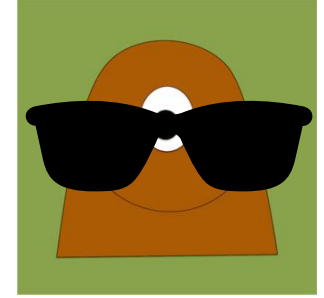
$$S = \text{Smarter}(x, y)$$

$$\sigma = \{x/\text{Agent}_1, y/\text{Wumpus}_1\}$$

$$S\sigma = \text{Smarter}(\text{Agent}_1, \text{Wumpus}_1)$$

- $\text{Ask}(\text{KB}, S)$ returns some/all σ such that $\text{KB} \models S\sigma$

Interacting with FOL KBs



- Suppose a Wumpus-world agent using a FOL KB perceives:
a **smell** and a **breeze** (but **no glitter**) at $t=5$:

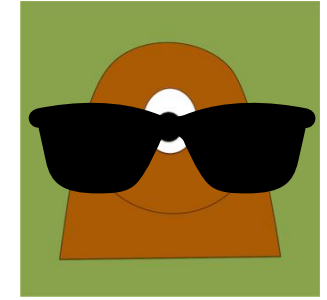
`Tell(KB, Percept([Smell, Breeze, None], 5))`

`Ask(KB, $\exists a.$ BestAction(a , 5))`

*i.e., does the KB entail some **best action** at $t=5$?*

Answer: Yes, $\{a/Shoot\}$ \leftarrow **substitution** (binding list)

KB for the Wumpus world






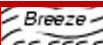
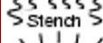






 Perception

$\forall t, s, b. \text{Percept}([s, b, \text{Glitter}], t) \Rightarrow \text{Glitter}(t)$



Reflex

$\forall t. \text{Glitter}(t) \Rightarrow \text{BestAction}(\text{Grab}, t)$

4	 Stench		 Breeze	PIT
3		 Breeze  Stench  Gold	PIT	 Breeze
2	 Stench		 Breeze	
1		 Breeze	PIT	 Breeze
	1	2	3	4

Deducing hidden properties

➤ $\forall x, y, a, b. \text{Adjacent}([x, y], [a, b]) \Leftrightarrow [a, b] \in \{ [x+1, y], [x-1, y], [x, y+1], [x, y-1] \}$

$\forall s, t. \text{At}(\text{Agent}, s, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(s)$

➤ Squares are breezy near a pit:

◦ **Diagnostic** rule: infer cause from effect

$\forall s. \text{Breezy}(s) \Rightarrow \exists r. \text{Adjacent}(r, s) \wedge \text{Pit}(r)$

◦ **Causal** rule: infer effect from cause

$\forall r. \text{Pit}(r) \Rightarrow (\forall s. \text{Adjacent}(r, s) \Rightarrow \text{Breezy}(s))$

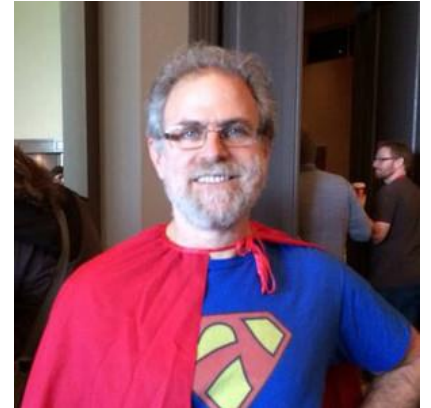
Why?

➤ Universal ontology language.

- e.g., databases, semantic web, knowledge graphs

➤ At the core of:

- programming language semantics and type theory.
- formal verification and advanced (> propositional) automated reasoning.
- theorem proving, including in mathematics, physics, cryptography, and beyond.
- logic programming and its derivations, expert systems, rule-based systems.



➤ Renewed interest in the context of explainable AI (XAI) and the “third-wave of AI”.



Phil Wadler “What does logic have to do with Java?” 2009