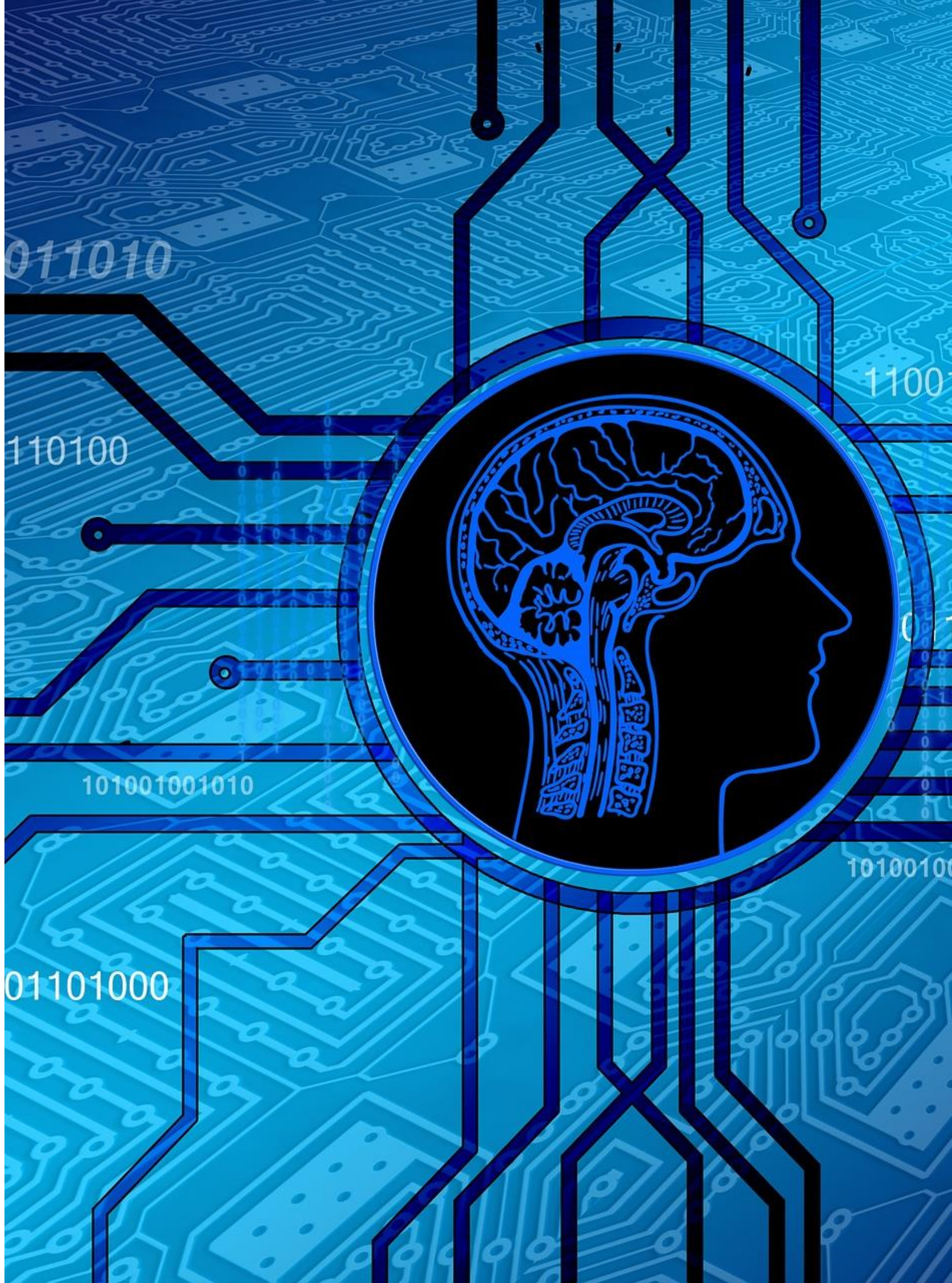
# First-order Logic

*Petros Papapanagiotou*

Informatics 2D: Reasoning and Agents
**Lecture 11**



# Pros and cons of propositional logic

Declarative

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

# Syntax of FOL: Basic elements

Arity!

- Constants: *KingJohn/0, 2 /0, UoE /0, …*
- Predicates: *Brother/2, >/2, …*
- Functions: *Sqrt/1, LeftLegOf/1, +/2, …*
- Variables: *x, y, a, b, …*
- Connectives: $\neg, \Rightarrow, \wedge, \vee, \Leftrightarrow$
- Equality: $=$
- Quantifiers: $\forall, \exists$

# Atomic formulae

**Atomic formula** = *predicate* (*term*$_1$,...,*term*$_n$)
or *term*$_1$ = *term*$_2$

**Term** = *function* (*term*$_1$,...,*term*$_n$)
or *constant* or *variable*

Examples:

- *Brother*(*KingJohn*,*RichardTheLionheart*)
- >( *Length*( *LeftLegOf*( *Richard* )), *Length*( *LeftLegOf*( *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:

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

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

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

# 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.

An *interpretation* contains objects (domain elements) and relations between them.
Mapping specifies referents for :

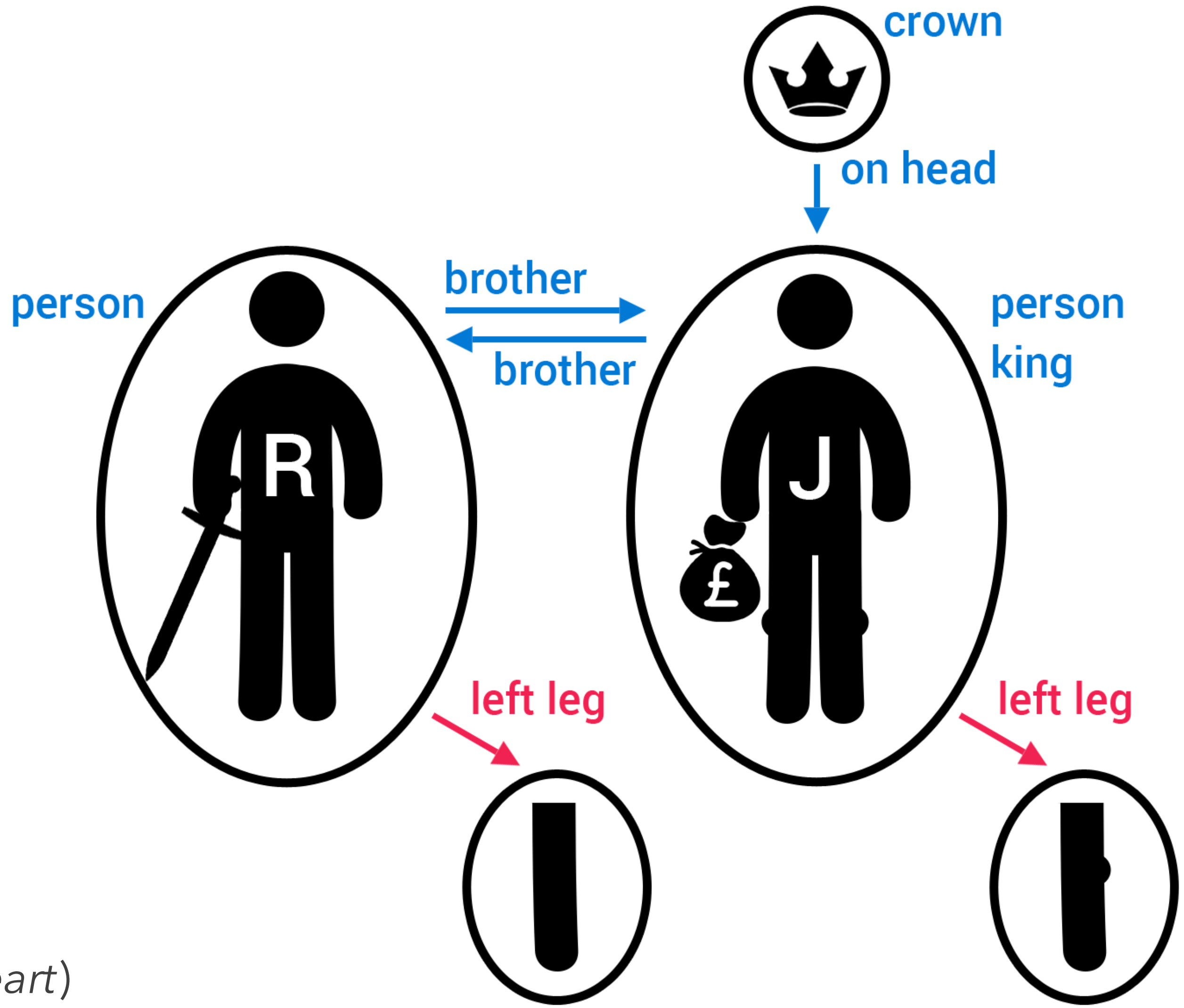
$$\begin{array}{lcl} \text{constant symbols} & \mapsto & \text{objects} \\ \text{predicate symbols} & \mapsto & \text{relation} \\ \text{function symbols} & \mapsto & \text{functions} \end{array}$$

An atomic formula $predicate(term_1,\ldots,term_n)$ is **true**

iff the objects referred to by $term_1,\ldots,term_n$

are in the relation referred to by *predicate*.

# Interpretations for FOL: Example




*Brother*(*KingJohn*,*RichardTheLionheart*)

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

# Universal quantification

$$\forall \langle variables \rangle.\ \langle formula \rangle$$

- But will often write $\forall\, x,y.\ P$ for $\forall x.\ \forall y.\ P$
- Example: *Everyone at UoE is smart*: $\forall x.\ \mathrm{At}(x, \mathrm{UoE}) \Rightarrow \mathrm{Smart}(x)$

$\forall x.\ P$ 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 $P$

$$\begin{aligned} & \mathrm{At}(\mathrm{KingJohn}, \mathrm{UoE}) \Rightarrow \mathrm{Smart}(\mathrm{KingJohn}) \\ \wedge\ & \mathrm{At}(\mathrm{Richard}, \mathrm{UoE}) \Rightarrow \mathrm{Smart}(\mathrm{Richard}) \\ \wedge\ & \mathrm{At}(\mathrm{UoE}, \mathrm{UoE}) \Rightarrow \mathrm{Smart}(\mathrm{UoE}) \\ \wedge\ & \ldots \end{aligned}$$

# Existential quantification

$\exists$<variables>. <formula>

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

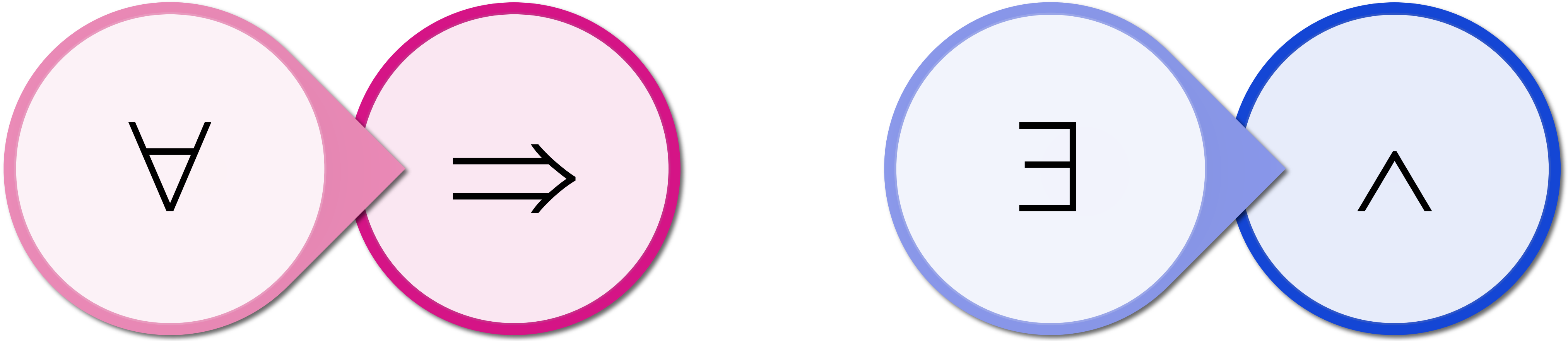
At(KingJohn, UoE) $\wedge$ Smart(KingJohn)

$\vee$ At(Richard, UoE) $\wedge$ Smart(Richard)

$\vee$ At(UoE, UoE) $\wedge$ Smart(UoE)

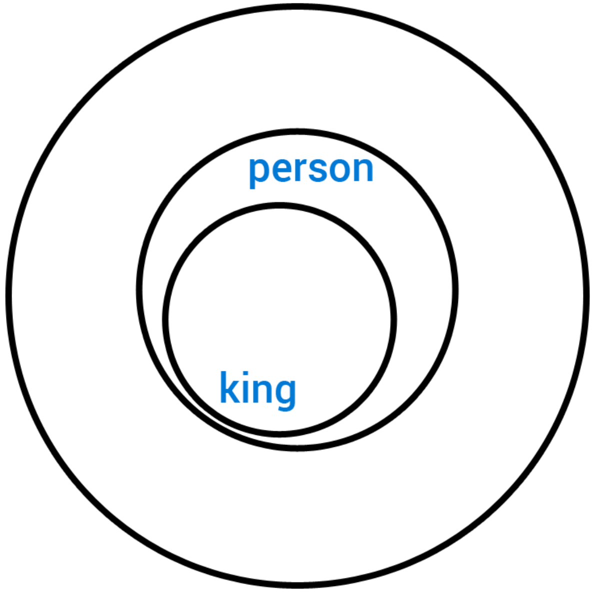
$\vee$ ...

# Rule of thumb

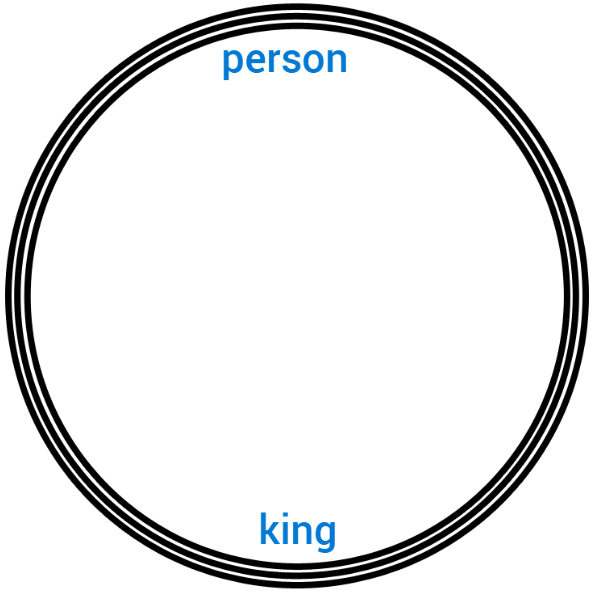


# Common mistakes

$\forall x.\ King(x) \Rightarrow Person(x)$

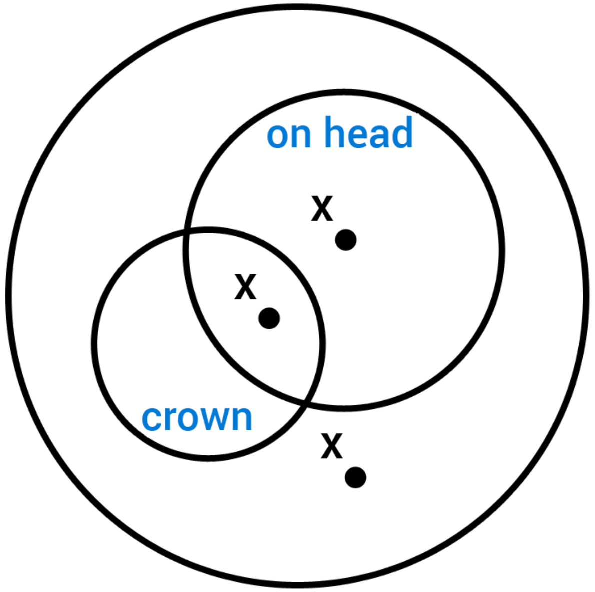



$\forall x.\ King(x) \land Person(x)$

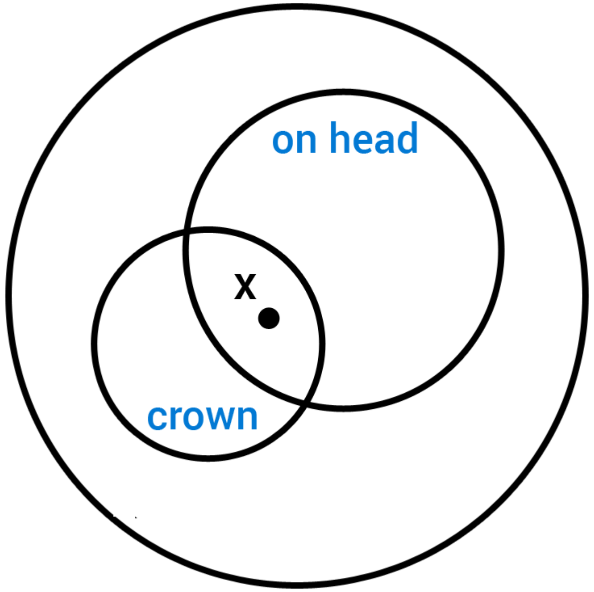


# Common mistakes

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




$\exists x.\ \mathrm{Crown}(x) \land \mathrm{OnHead}(x, \mathit{John})$



# 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 $\forall y.\exists x.$

- $\exists x.\ \forall y.$ 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"*

Quantifier duality: each can be expressed using the other:

- $\forall x.$ Likes($x$, IceCream) $\equiv$ $\neg\exists x.\ \neg$Likes($x$, IceCream)
- $\exists x.$ Likes($x$, Broccoli) $\equiv$ $\neg\forall x.\ \neg$Likes($x$, 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*:

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

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

$$\mathrm{Parent}(m, x) \wedge \mathrm{Parent}(f, x) \wedge \mathrm{Parent}(m, y) \wedge \mathrm{Parent}(f, y))$$

# Example: The kinship domain

*Brothers are siblings.*

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

*One's mother is one's female parent.*

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

*“Sibling” is symmetric.*

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

*“Parent” and “Child” are inverse relations.*

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

# Example: The Set domain

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

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

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

$\forall x,s.\ x \in s \Leftrightarrow [\ \exists y,s_{2.}\ (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$

$\forall s_1,s_{2.}\ s_1 \subseteq s_2 \Leftrightarrow (\forall x.\ x \in s_1 \Rightarrow x \in s_2)$

$\forall s_1,s_{2.}\ (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$

$\forall x,s_1,s_{2.}\ x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$

$\forall x,s_1,s_{2.}\ x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$

# 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, ∃a. BestAction(a, 5))

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

- Answer: *Yes*, {*a/Shoot*} ← substitution (binding list)

# Substitution

Given a sentence $S$ and a substitution σ,

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

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

$$\sigma = \{x/\text{Obama}, y/\text{Palin}\}$$

$$S\sigma = \text{Smarter}(\text{Obama}, \text{Palin})$$

`Ask`(KB, S) returns some/all σ such that KB ⊨ Sσ

# Knowledge base for the wumpus world

Perception

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

Reflex

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

# Deducing hidden properties

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

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

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.

- Onto-logy: from the Greek **όν** (= being, that which is) + **λόγος** (= discourse, speaking)
- 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