

# Fundamentals of Artificial Intelligence

## Chapter 12: Knowledge Representation

Roberto Sebastiani

DISI, Università di Trento, Italy – [roberto.sebastiani@unitn.it](mailto:roberto.sebastiani@unitn.it)  
[http://disi.unitn.it/rseba/DIDATTICA/fai\\_2022/](http://disi.unitn.it/rseba/DIDATTICA/fai_2022/)

Teaching assistant: **Mauro Dragoni** – [dragoni@fbk.eu](mailto:dragoni@fbk.eu)  
<http://www.maurodragoni.com/teaching/fai/>

M.S. Course “Artificial Intelligence Systems”, academic year 2022-2023

Last update: Friday 2<sup>nd</sup> December, 2022, 14:48

Copyright notice: Most examples and images displayed in the slides of this course are taken from [\[Russell & Norvig, “Artificial Intelligence, a Modern Approach”, 3<sup>rd</sup> ed., Pearson\]](#), including explicitly figures from the above-mentioned book, so that their copyright is detained by the authors. A few other material (text, figures, examples) is authored by (in alphabetical order): [Pieter Abbeel](#), [Bonnie J. Dorr](#), [Anca Dragan](#), [Dan Klein](#), [Nikita Kitaev](#), [Tom Lenaerts](#), [Michela Milano](#), [Dana Nau](#), [Maria Simi](#), who detain its copyright. These slides cannot be displayed in public without the permission of the author.

# Outline

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects
- 3 Reasoning about Knowledge
- 4 Reasoning about Categories
  - Semantic Networks (hints)
  - Description Logics

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects
- 3 Reasoning about Knowledge
- 4 Reasoning about Categories
  - Semantic Networks (hints)
  - Description Logics

# Generalities

Q: What content do we put into an agent's KB?

- how do we organize such content?
- how do we represent facts about the world?
- A whole AI field: Knowledge Representation, KR
  - often combined with Automated Reasoning on KB
- ⇒ Knowledge Representation & Reasoning, KRR
- KR: use FOL to represent the most important aspects of the real world, such as: action, space, time, knowledge, belief
- Topics:
  - ontologies and ontological engineering
  - objects and categories, composite objects, measurements, ...
  - actions and change, events, temporal intervals, ...
  - reasoning about knowledge & beliefs
  - reasoning about categories
  - default reasoning
  - ...

# Knowledge Engineering and Ontological Engineering

## Knowledge Engineering

- The activity to **formalize a specific problem or task domain**
- Relevant questions to be addressed:
  - What are the relevant facts, objects, relations ... ?
  - Which is the right level of abstraction?
  - What are the queries to the KB (inferences)?

## Ontological Engineering

- The activity to **build general-purpose ontologies**
  - should **be applicable in any special-purpose domain** (with the addition of domain-specific axioms)
  - In non trivial domains, reasoning and problem solving could involve several areas of knowledge simultaneously
    - ⇒ **different areas of knowledge must be combined**
- Several attempts to build general-purpose ontologies
  - CYC, DBpedia, TextRunner, ...
  - not very successful so far

# Outline

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects**
- 3 Reasoning about Knowledge
- 4 Reasoning about Categories
  - Semantic Networks (hints)
  - Description Logics

# Categories and Objects

## Categories, Objects, Members and Subclasses

- **KR requires the organisation of objects into categories**
  - interaction at the level of the object
  - reasoning at the level of categories
  - ex: typically we want to buy a basketball, rather than a particular basketball instance
- **Categories play a role in predictions about objects**
  - agent infers the presence of certain objects from perceptual input
  - infers category from the perceived properties of the objects,
  - uses category information to make predictions about the objects
- Categories can be represented in two ways by FOL
  - **predicates** (ex  $\text{Basketball}(x)$ ): **relations**
  - **reification** of categories into objects (ex  $\text{Basketballs}$ ): **sets**
    - ⇒ allows categories to be argument of predicates/functions
- Membership of a category as set membership
  - ex:  $\text{Member}(b, \text{Basketballs})$  (abbr.  $b \in \text{Basketballs}$ )
- **Subcategories** (aka **subclasses**) are (strict) subsets
  - ex:  $\text{Subset}(\text{Basketballs}, \text{Balls})$  (abbr.  $\text{Basketballs} \subset \text{Balls}$ )

# Categories and Objects [cont.]

## Inheritance and Taxonomies

- A subcategory inherits the properties of the category
  - ex:  
if  $\forall x.(x \in \text{Food} \rightarrow \text{Edible}(x))$ ,  $\text{Fruit} \subset \text{Food}$ ,  $\text{Apples} \subset \text{Fruit}$   
then  $\forall x.(x \in \text{Apple} \rightarrow \text{Edible}(x))$
- A member inherits the properties of the category
  - if  $a \in \text{Apples}$ , then  $\text{Edible}(a)$
- Subclass relation organize categories into taxonomies (aka taxonomic hierarchies)
  - ex: taxonomy of >10M living&extinct species
  - ex: Dewey Decimal System: taxonomy of all fields of knowledge



# Categories and Objects [cont.]

## FOL Reasoning about Categories

- FOL allows to state facts about categories:
  - an object is a member of a category  
 $BB_9 \in \textit{Basketballs}$
  - a category is a subclass of another category  
 $\textit{Basketballs} \subset \textit{Balls}$
  - all members of a category have some properties  
 $\forall x.(x \in \textit{Basketballs} \rightarrow \textit{Spherical}(x))$
  - members of a category can be recognized by some properties  
 $\forall x.((\textit{Orange}(x) \wedge \textit{Round}(x) \wedge \textit{Diameter}(x) = 9.5'' \wedge x \in \textit{Balls}) \rightarrow x \in \textit{Basketballs})$
  - category as a whole has some properties  
 $\textit{Dogs} \in \textit{DomesticatedSpecies}$
- New categories can be defined by providing **necessary and sufficient conditions** for membership
  - $\forall x.(x \in \textit{Bachelors} \leftrightarrow (\textit{Unmarried}(x) \wedge x \in \textit{Adults} \wedge x \in \textit{Males}))$

# Categories and Objects [cont.]

## Derived relations

- Two or more categories in a set  $s$  are **disjoint** iff they have no members in common
  - $Disjoint(s) \leftrightarrow (\forall c_1 c_2. ((c_1 \in s \wedge c_2 \in s \wedge c_1 \neq c_2) \rightarrow Intersection(c_1, c_2) = \emptyset))$
  - ex:  
 $Disjoint(\{Animals, Vegetables\}), Disjoint(\{Insects, Birds, Mammals, Reptiles\}),$
- A set of categories  $s$  is an **exhaustive decomposition** of a category  $c$  iff all members of  $c$  are covered by categories in  $s$ 
  - $ExhaustiveDecomposition(s, c) \leftrightarrow \forall i. (i \in c \leftrightarrow (\exists c_2. (c_2 \in s \wedge i \in c_2)))$
  - ex:  $E.D.(\{Americans, Canadians, Mexicans\}, NorthAmericans)$
- A disjoint exhaustive decomposition is a **partition**
  - $Partition(s, c) \leftrightarrow (Disjoint(s) \wedge ExhaustiveDecomposition(s, c))$
  - ex:  $Partition(\{NorthernItalians, CentralItalians, SouthernItalians, InsularItalians\}, Italians)$

## Digression: Natural Kinds

- Many categories have no clear-cut definition (ex: **chair**, **bush**, ...)
  - Ex: tomatoes are sometimes green, red, yellow, black; they are mostly round
- One useful solution: category “**Typical(.)**”, s.t.  $Typical(c) \subseteq c$ 
  - ⇒ most knowledge about natural kinds will actually be about their typical instances
    - ex:  $\forall x.(x \in Typical(Tomatoes) \rightarrow (Red(x) \wedge Round(x)))$

⇒ We can write down useful facts about categories without providing exact definitions

### Note

Quine (1953) challenged the utility of the notion of strict definition.

- Ex: “bachelor”: **is the Pope a bachelor?**
  - ⇒ technically yes, but misleading

# Physical Composition

- *PartOf*(.,.) relation: **One object may be part of another**
  - *PartOf*(Bucharest, Romania)
  - *PartOf*(Romania, EasternEurope)
  - *PartOf*(EasternEurope, Europe)
- *PartOf*(.,.) is reflexive and transitive:
  - $\forall x. PartOf(x, x)$
  - $\forall x, y, z. ((PartOf(x, y) \wedge PartOf(y, z)) \rightarrow PartOf(x, z))$ $\Rightarrow PartOf(Bucharest, Europe)$
- Categories of **composite objects** are often characterized by structural relations among parts.  
Ex: **Biped**

$$\begin{aligned} Biped(a) \Rightarrow \exists l_1, l_2, b \quad & Leg(l_1) \wedge Leg(l_2) \wedge Body(b) \wedge \\ & PartOf(l_1, a) \wedge PartOf(l_2, a) \wedge PartOf(b, a) \wedge \\ & Attached(l_1, b) \wedge Attached(l_2, b) \wedge \\ & l_1 \neq l_2 \wedge [\forall l_3 \quad Leg(l_3) \wedge PartOf(l_3, a) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)] \end{aligned}$$

(© S. Russell & P. Norwig, AIMA)

- Other concepts & relations: **PartPartition**, **BunchOf**...

# Measurements

## Quantitative Measurements

- Objects may have “quantitative” properties
  - e.g. *height*, *mass*, *cost*, ...
- Values that we assign to these properties are *measures*
- Can be represented by *unit functions*
  - ex  $Length(L_1) = Inches(1.5) \wedge Inches(1.5) = Centimeters(3.81)$
- Conversion between units:
  - $\forall i. Centimeters(2.54 \times i) = Inches(i)$
- Measures can be used to describe objects:
  - ex:  $Diameter(Basketball_{12}) = Inches(9.5)$
  - ex:  $ListPrice(Basketball_{12}) = \$(19)$
  - ex:  $\forall d.(d \in Days \rightarrow Duration(d) = Hours(24))$

# Measurements [cont.]

## Qualitative Measurements

- Some measures have no scale
  - ex: *beauty*, *deliciousness*, *difficulty*,...
- Most important aspect of measures: they are **orderable**
  - Ex: *Deliciousness(SacherTorte) > Deliciousness(BrussellSprout)*
  - Ex: *Beauty(PaulNewmann) > Beauty(MartyFeldman)*
  - Ex: *Difficulty(Prove\_P ≠ NP) > Difficulty(SolvePuzzle)*
- Allow for reasoning by exploiting transitivity of monotonicity:
  - $\forall e_1 e_2. ((e_1 \in \text{Exercises} \wedge e_2 \in \text{Exercises} \wedge \text{Wrote}(\text{Norvig}, e_1) \wedge \text{Wrote}(\text{Russell}, e_2)) \rightarrow \text{Difficulty}(e_1) > \text{Difficulty}(e_2))$
  - $\forall e_1 e_2. ((e_1 \in \text{Exercises} \wedge e_2 \in \text{Exercises} \wedge \text{Difficulty}(e_1) > \text{Difficulty}(e_2)) \rightarrow \text{ExpectedScore}(e_1) < \text{ExpectedScore}(e_2))$
  - $\forall e_1 e_2. (\text{ExpectedScore}(e_1) < \text{ExpectedScore}(e_2) \rightarrow \text{Pick}(e_1, e_2) = e_2)$
  - Then:  $(\text{Wrote}(\text{Norvig}, E_1) \wedge \text{Wrote}(\text{Russell}, E_2)) \models \text{Pick}(E_1, E_2) = E_2$
- **Qualitative physics**: a subfield of AI that investigates how to reason about physical systems without numerical computations

# Objects vs Stuff

- There are **countable objects**
  - e.g, **apples, holes, theorems, ...**
- ... and **mass objects**, aka **stuff** or **substances**
  - e.g. **butter, water, energy, ...**

⇒ Intuitive meaning “an amount/quantity of...”

- ex:  $b \in \text{butter}$ : “b is an amount/quantity of butter”
- Any part of stuff is still stuff:
  - ex:  $\forall b, p. ((b \in \text{Butter} \wedge \text{PartOf}(p, b)) \rightarrow p \in \text{Butter})$
- Can define sub-categories, which are stuff
  - ex:  $\text{UnsaltedButter} \subset \text{Butter}$
- Stuff has a number of **intrinsic properties**, shared by its subparts
  - e.g., color, fat content, density ...
  - ex:  $\forall b. (b \in \text{Butter} \rightarrow \text{MeltingPoint}(b, \text{Centigrade}(30)))$
- Stuff has no **extrinsic properties**
  - e.g., weight, length, shape, ...

# Outline

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects
- 3 Reasoning about Knowledge**
- 4 Reasoning about Categories
  - Semantic Networks (hints)
  - Description Logics



# Agents' Attitudes

- Intelligence is intrinsically social: agents need to negotiate and coordinate with other agents
- In multi-agents scenarios, to predict what other agents will do, **we need methods to model mental states of other agents**
  - representations of other agents' knowledge (and beliefs, goals)
- Agent's **Propositional attitudes**: Knows, Believes, Wants,...
  - ex "Lois **Knows** that Superman can fly"

## Problem

Propositional attitudes do not behave as regular predicates

- issue: **Referential opacity** vs. **referential transparency**

# Referential opacity vs. Referential transparency

- Consider the assertion “Lois knows that Superman can fly”
- Consider the FOL formalization:  $Knows(Lois, CanFly(Superman))$
- Minor Problem:  $CanFly(Superman)$  is a formula
  - ⇒ cannot occur as argument of a predicate
  - ⇒ **must apply reification** ⇒ make it a term
- Major Problem (Referential Transparency of FOL):
  - since **Superman is Clark Kent** (but Lois doesn't know it!), FOL allows to conclude “Lois knows that Clark Kent can fly”:  
 $Superman = Clark \wedge Knows(Lois, CanFly(Superman))$   
 $\models_{FOL} Knows(Lois, CanFly(Clark))$   
⇒ **Wrong inference!** (Lois doesn't know Clark Kent can fly!)
- Hint: FOL predicates transparent to equality reasoning:  
 $t = s \wedge P(s, \dots) \models_{FOL} P(t, \dots)$
- Need a logic which is **opaque** to equality reasoning (aka **Referential Opacity**):  
**Modal Logics**

# Modal Logics

- Modal logics include **special modal operators** that take **formulas** (not terms!) as arguments
  - “A knows P” is represented with  $K_A P$  ( $P$  formula, not term!)
  - ex: “Lois knows that Superman can fly”:  $K_{Lois} CanFly(Superman)$
  - ex: “Lois knows Clark Kent knows if he is Superman or not”:  
 $K_{Lois}(K_{Clark} Identity(Superman, Clark) \vee K_{Clark} \neg Identity(Superman, Clark))$
- Properties in all modal logics:
  - $K_A(P \wedge Q) \iff K_A P \wedge K_A Q$
  - $K_A P \vee K_A Q \models K_A(P \vee Q)$ , but  $K_A(P \vee Q) \not\models K_A P \vee K_A Q$  (e.g.  $K_A(P \vee \neg P) \not\models K_A P \vee K_A \neg P$ )
- The following axiom holds in all (normal) modal logics:  
 $K : (K_A \phi \wedge K_A(\phi \rightarrow \psi)) \rightarrow K_A \psi$  (**distribution axiom**): “A is able to perform propositional inference”
- The following axioms hold in some (normal) modal logics:  
 $T : K_A \phi \rightarrow \phi$  (**knowledge axiom**): “A knows only true facts”  
 $4 : K_A \phi \rightarrow K_A K_A \phi$  (**positive-introspection axiom**): “If A knows fact  $\phi$ , then [s]he knows [s]he knows it”  
 $5 : \neg K_A \phi \rightarrow K_A \neg K_A \phi$  (**negative-introspection axiom**):  
“If A doesn’t know  $\phi$ , then [s]he knows [s]he doesn’t know it”
- **Referential Opacity**:  $Superman = Clark \wedge K_{Lois} CanFly(Superman) \not\models K_{Lois} CanFly(Clark)$
- Reasoning in (propositional) Modal logics is NP-hard (most often even PSPACE-hard)

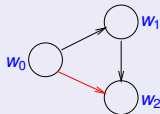
# Semantics of Modal Logics

- A model (**Kripke model**) is a **collection of possible worlds  $w_i$** 
  - possible worlds are connected in a graph by **accessibility relations**
  - one relation for each distinct modal operator  $K_A$
- $w_1$  is **accessible from  $w_0$  wrt.  $K_A$**  if **everything which holds in  $w_1$  is consistent with what A knows in  $w_0$**  (written “ $Acc(K_A, w_0, w_1)$ ” or “ $w_0 \xrightarrow{K_A} w_1$ ”)
  - $\Rightarrow K_A\varphi$  holds in  $w_0$  iff  $\varphi$  holds in every world  $w_i$  accessible from  $w_0$ 
    - the more is known in  $w_0$ , the less worlds are accessible from  $w_0$
    - remark: **two possible worlds may differ also for what an agent knows there**
- Different modal logics differ by different properties of  $Acc(K_A, \dots)$ 
  - **T** :  $K_A\varphi \rightarrow \varphi$  holds iff  $Acc(K_A, \dots)$  **reflexive**:  $w \xrightarrow{K_A} w$
  - **4** :  $K_A\varphi \rightarrow K_A K_A\varphi$  holds iff  $Acc(K_A, \dots)$  **transitive**:  $w_0 \xrightarrow{K_A} w_1$  and  $w_1 \xrightarrow{K_A} w_2 \implies w_0 \xrightarrow{K_A} w_2$
  - **5** :  $\neg K_A\varphi \rightarrow K_A\neg K_A\varphi$  holds iff  $Acc(K_A, \dots)$  **euclidean**:  $w_0 \xrightarrow{K_A} w_1$  and  $w_0 \xrightarrow{K_A} w_2 \implies w_1 \xrightarrow{K_A} w_2$

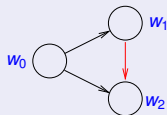
T: reflexive



4: transitive



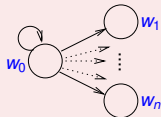
5: euclidean



# Semantics of Modal Logics: Some Remarks

Assume the knowledge of  $A$  is correct:  $T : K_A\varphi \rightarrow \varphi$  (“Everything which  $A$  knows holds”)

- $\not\models \varphi \rightarrow K_A\varphi$ :  $A$  does not know everything which holds!  
 $\implies A$  does not know exactly in which world [s]he is
- The less worlds are accessible, the more precise is the knowledge of  $A$ 
  - uncertainty on some information makes accessible worlds different  
 $\implies A$  does not know the world [s]he is
  - complete knowledge: current world is the only successor of itself  
 $\implies A$  knows exactly the world [s]he is



Uncertainty



Complete knowledge

Notice the difference:

- $K_A\neg P$ : agent  $A$  knows that  $P$  does not hold (in all accessible worlds  $P$  is false)
  - $\neg K_AP$ : agent  $A$  does not know if  $P$  holds (in some accessible worlds  $P$  is false)
- $\implies K_A\neg P \models \neg K_AP$ , but  $\neg K_AP \not\models K_A\neg P$

# Semantics of Modal Logics: Example

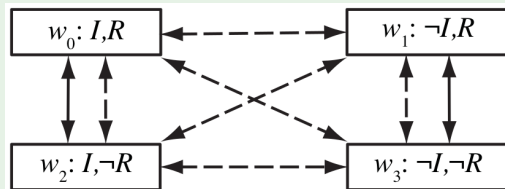
Accessibility relations:  $K_{Superman}$  (solid arrows) and  $K_{Lois}$  (dotted arrows).

- Legend:

- R: “the weather report says tomorrow will rain”
- I: “Superman’s secret identity is Clark Kent.”
- Ex:  $K_{Lois}(K_{Clark}I \vee K_{Clark}\neg I)$ : “Lois Knows that Clark Knows if he is Superman or not.”

- Superman knows his own identity:  $K_{Superman}I \vee K_{Superman}\neg I$ , and

(a) neither Superman nor Lois has seen the weather report, she knows Superman knows if he is Clark  
 $(\neg K_{Lois}R \wedge \neg K_{Lois}\neg R) \wedge (\neg K_{Superman}R \wedge \neg K_{Superman}\neg R) \wedge K_{Lois}(K_{Superman}I \vee K_{Superman}\neg I)$



(a)

(self-loop arrows not reported)

(© S. Russell & P. Norwig, AIMA)

# Semantics of Modal Logics: Example

Accessibility relations:  $K_{Superman}$  (solid arrows) and  $K_{Lois}$  (dotted arrows).

- **Legenda:**

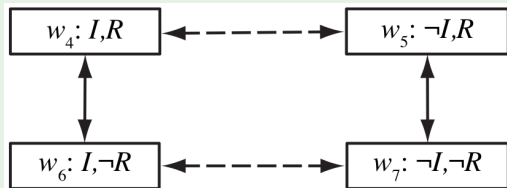
- R: “the weather report says tomorrow will rain”
- I: “Superman’s secret identity is Clark Kent.”
- Ex:  $K_{Lois}(K_{Clark}I \vee K_{Clark}\neg I)$ : “Lois Knows that Clark Knows if he is Superman or not.”

- **Superman knows his own identity:**  $K_{Superman}I \vee K_{Superman}\neg I$ , and

**(b) Lois has seen the weather report,** Superman has not, but he knows that Lois has seen it

$(K_{Lois}R \vee K_{Lois}\neg R) \wedge (\neg K_{Superman}R \wedge \neg K_{Superman}\neg R)$

$K_{Lois}(K_{Superman}I \vee K_{Superman}\neg I) \wedge K_{Superman}(K_{Lois}R \vee K_{Lois}\neg R)$



(b)

(self-loop arrows not reported)

# Semantics of Modal Logics: Example

Accessibility relations:  $K_{Superman}$  (solid arrows) and  $K_{Lois}$  (dotted arrows).

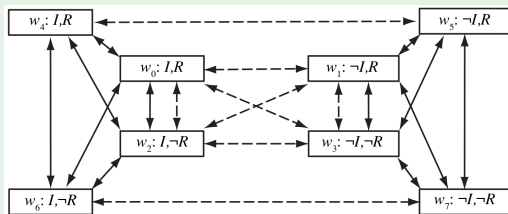
- **Legenda:**

- R: “the weather report says tomorrow will rain”
- I: “Superman’s secret identity is Clark Kent.”
- Ex:  $K_{Lois}(K_{Clark}I \vee K_{Clark}\neg I)$ : “Lois Knows that Clark Knows if he is Superman or not.”

- **Superman knows his own identity:  $K_{Superman}I \vee K_{Superman}\neg I$ , and**  
**(c) Lois may or may not have seen the weather report, Superman has not:**

$$((\neg K_{Lois}R \wedge \neg K_{Lois}\neg R) \vee (K_{Lois}R \vee K_{Lois}\neg R)) \wedge (\neg K_{Sup.}R \wedge \neg K_{Sup.}\neg R)$$

$$K_{Lois}(K_{Superman}I \vee K_{Superman}\neg I)$$



(c)

(self-loop arrows not reported)



# Semantics of Modal Logics: Example

Accessibility relations:  $K_{Superman}$  (solid arrows) and  $K_{Lois}$  (dotted arrows).

- **Legenda:**

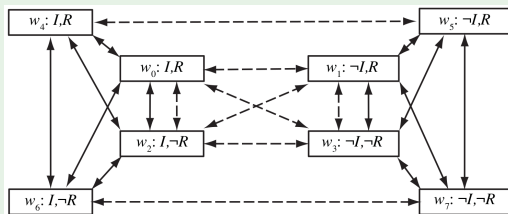
- R: “the weather report says tomorrow will rain”
- I: “Superman’s secret identity is Clark Kent.”
- Ex:  $K_{Lois}(K_{Clark}I \vee K_{Clark}\neg I)$ : “Lois Knows that Clark Knows if he is Superman or not.”

- **Superman knows his own identity:**  $K_{Superman}I \vee K_{Superman}\neg I$ , and

(c) Lois may or may not have seen the weather report, Superman has not:

$((\neg K_{Lois}R \wedge \neg K_{Lois}\neg R) \vee (K_{Lois}R \vee K_{Lois}\neg R)) \wedge (\neg K_{Sup.}R \wedge \neg K_{Sup.}\neg R)$

$K_{Lois}(K_{Superman}I \vee K_{Superman}\neg I)$



(c)

(self-loop arrows not reported)

## Exercise

Consider the previous example.

- For each scenario (a), (b) and (c) define doubly-nested knowledge in terms of

$$\begin{aligned} &[\neg]K_{Lois}[\neg]K_{Lois}[\neg]I, \\ &[\neg]K_{Lois}[\neg]K_{Lois}[\neg]R, \\ &[\neg]K_{Sup.}[\neg]K_{Sup.}[\neg]I, \\ &[\neg]K_{Sup.}[\neg]K_{Sup.}[\neg]R \end{aligned}$$

## Exercise

Consider (normal) modal logics (i.e., axioms K, T, 4 and 5 hold).

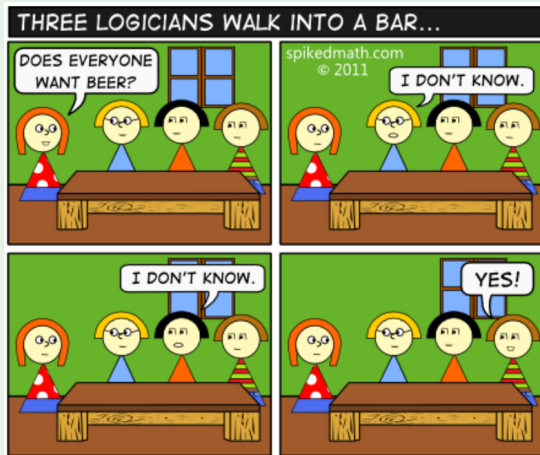
Let  $\text{IsRed}(\text{Pen})$ ,  $\text{IsOnTable}(\text{Pen})$  be possible facts, let  $Mary$ ,  $John$  be agents and let  $K_{Mary}$ ,  $K_{John}$  denote the modal operators “Mary knows that...” and “John knows that...” respectively.

For each of the following facts, say if it is true or false.

- If  $K_{Mary} \neg \text{IsRed}(\text{Pen})$  holds, then  $\neg K_{Mary} \text{IsRed}(\text{Pen})$  holds
- If  $\neg K_{Mary} \text{IsRed}(\text{Pen})$  holds, then  $K_{Mary} \neg \text{IsRed}(\text{Pen})$  holds
- If  $K_{John} \text{IsRed}(\text{Pen})$  and  $\text{IsRed}(\text{Pen}) \leftrightarrow \text{IsOnTable}(\text{Pen})$  hold, then  $K_{John} \text{IsOnTable}(\text{Pen})$  holds
- If  $K_{Mary} \text{IsRed}(\text{Pen})$  and  $K_{Mary} (\text{IsRed}(\text{Pen}) \rightarrow K_{John} \text{IsRed}(\text{Pen}))$  hold, then  $K_{Mary} K_{John} \text{IsRed}(\text{Pen})$  holds

# Exercise

- Why does the third logician answers “Yes”?
- Formalize and solve the problem by means of modal logic (K+T+4+5)



(Courtesy of Maria Simi, UniPI)

# Outline

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects
- 3 Reasoning about Knowledge
- 4 Reasoning about Categories**
  - Semantic Networks (hints)
  - Description Logics

# Outline

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects
- 3 Reasoning about Knowledge
- 4 Reasoning about Categories**
  - Semantic Networks (hints)**
  - Description Logics

# Reasoning Systems for Categories

## Q. How to organize and reason with categories?

### • Semantic Networks

- allow to visualize knowledge bases
- efficient algorithms for category membership inference
- limited expressivity
- many variants

### • Description Logics (DLs)

- formal language for constructing and combining category definitions
- (relatively) efficient algorithms to decide subset and superset relationships between categories
- many DLs
  - up to very high expressivity
  - up to very high complexity (e.g., DOUBLY-EXPTIME)

# Semantic Networks

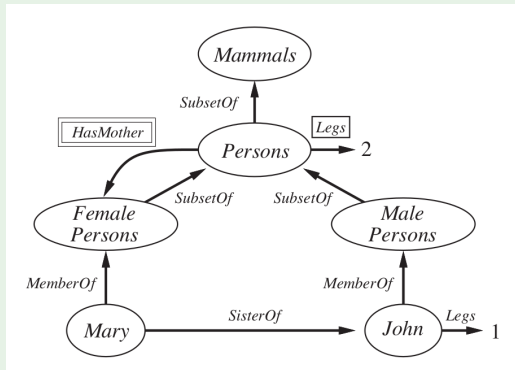
- Allow for representing **individual objects**, **categories of objects**, and **relations among objects**
- A **Semantic Network** is a graph where:
  - nodes, with a label, correspond to **concepts**
  - arcs, labelled and directed, correspond to **binary relations between concepts** (aka **roles**)
- Two kinds of nodes:
  - **Generic concepts**, corresponding to **categories/classes**
  - **Individual concepts**, corresponding to **individuals**
- Two special relations are always present, with different names
  - **IS-A**, aka **SubsetOf/SubclassOf** (**subclass**)
  - **InstanceOf** aka **MemberOf** (**membership**)
- **Inheritance detection straightforward**
- Ability to represent **default values** for categories
- Limited expressive power: **cannot represent negation, disjunction, nested function symbols, existential quantification**



# Semantic Networks: Example

## ● Notice

- “HasMother” is a relation between persons (individuals) (categories do not have mothers)
- “HasMother” (double-boxed notation) means  
 $\forall x.(x \in \text{Persons} \rightarrow [\forall y.(\text{HasMother}(x, y) \rightarrow y \in \text{FemalePersons})])$
- “Legs” is a property of single persons (individuals)
- “Legs” (single-boxed notation) means:  
 $\forall x.(x \in \text{Persons} \rightarrow \text{Legs}(x, 2))$

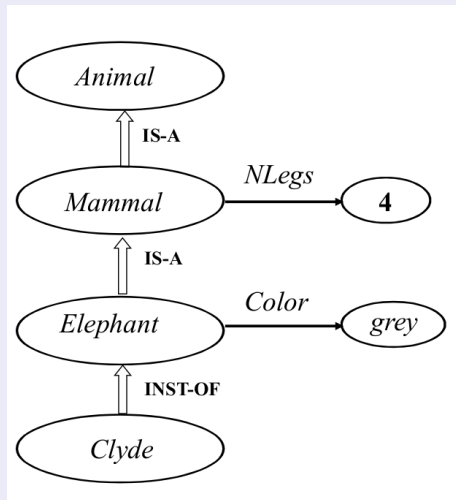


# Inheritance in Semantic Networks

- Inheritance conveniently implemented as [link traversal](#)

Q. How many legs has Clyde?

⇒ follow the INST-OF/IS-A chain until find the property NLegs

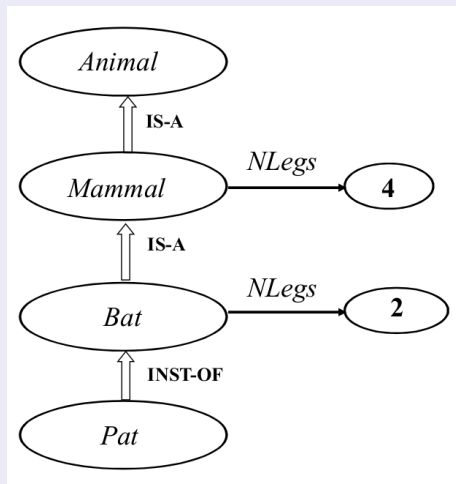


(Courtesy of Maria Simi, UniPI)

# Inheritance with Exceptions

The presence of exceptions does not create any problem with S.N.

- How many legs has Pat?
  - Just take **the most specific information**: the first that is found going up the hierarchy
- ⇒ ability to represent **default values** for categories



(Courtesy of Maria Simi, UniPI)

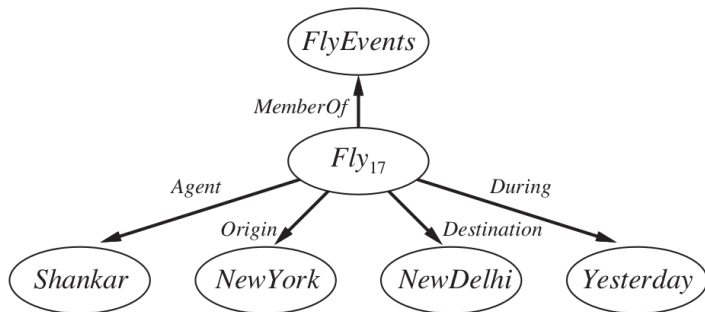
# Encoding N-Ary Relations

- Semantic networks allow only binary relations

Q. How to represent n-ary relations?

⇒ Reify the proposition as an event belonging to an appropriate event category

- ex “*Fly<sub>17</sub>*” for *Fly(Shankar, NewYork, NewDelhi, Yesterday)*



# Outline

- 1 Ontologies and Ontological Engineering
- 2 Categories and Objects
- 3 Reasoning about Knowledge
- 4 Reasoning about Categories**
  - Semantic Networks (hints)
  - Description Logics**

# Description Logics

- Designed to describe **definitions** and **properties** about categories
- Principal inference tasks:
  - **Subsumption**: check if one category is a subset (sub-category) of another
  - **Classification**: check whether an object belongs to a category
  - **Consistency**: check if category membership criteria are satisfiable
- Defaults and exceptions are lost

# Concepts, Roles, Individuals

- **Concepts**, corresponding to **unary relations**
  - $\top$ ,  $\perp$ : universal and empty concepts
  - **atomic concepts**: ex: *Female, Male, Article, Journalist, ...*
  - operators for the construction of complex concepts:  
and ( $\sqcap$ ), or ( $\sqcup$ ), not ( $\neg$ ), all ( $\forall$ ), some ( $\exists$ ), at least ( $\geq n$ ), at most ( $\leq n$ ), ...
  - ex: mothers (i.e., women who have children) of at least three female children:  
*Woman  $\sqcap$   $\exists$ hasChildren.Person  $\sqcap$   $\geq 3$  hasChild.Female*
  - ex: articles that have authors and whose authors are all journalists:  
*Article  $\sqcap$   $\exists$ hasAuthor.\sqcap  $\forall$ hasAuthor.Journalist*
- **Roles** corresponding to **binary relations**
  - ex: *hasAuthor, hasChild*
  - can be combined with operators for constructing complex roles
  - *hasChildren  $\equiv$  hasSon  $\sqcup$  hasDaughter*
- **Individuals** (used in assertions only)
  - ex: *Mary, John*

# T-Boxes and A-Boxes

- Terminologies (T-Boxes): sets of
  - concepts definitions ( $C_1 \equiv C_2$ )  
ex: *Father*  $\equiv$  *Man*  $\sqcap$   $\exists$ *hasChild*.*Person*
  - or concept generalizations ( $C_1 \sqsubseteq C_2$ )  
ex: *Woman*  $\sqsubseteq$  *Person*
- Assertions (A-Boxes): assert
  - individuals as concept members  $i : C$ ,  
where i is an individual and C is a concept  
ex: *mary* : *Person*, *john* : *Father*
  - individual pairs as relation members  $\langle i, j \rangle : R$ ,  
where i,j are individuals and R is a relation  
ex:  $\langle$ *john*, *mary* $\rangle$  : *hasChild*



## T-Box: Example (Logic $\mathcal{ALCN}$ )

Woman	$\equiv$	Person $\sqcap$ Female
Man	$\equiv$	Person $\sqcap$ $\neg$ Woman
Mother	$\equiv$	Woman $\sqcap$ $\exists$ hasChild.Person
Father	$\equiv$	Man $\sqcap$ $\exists$ hasChild.Person
Parent	$\equiv$	Father $\sqcup$ Mother
Grandmother	$\equiv$	Mother $\sqcap$ $\exists$ hasChild.Parent
MotherWithManyChildren	$\equiv$	Mother $\sqcap$ $\geq 3$ hasChild .Person
MotherWithoutDaughter	$\equiv$	Mother $\sqcap$ $\forall$ hasChild. $\neg$ Woman
Wife	$\equiv$	Woman $\sqcap$ $\exists$ hasHusband. Man

# Reasoning Services for DLs

- Design and management of ontologies
  - consistency checking of concepts, creation of hierarchies
- Ontology integration
  - Relations between concepts of different ontologies
  - Consistency of integrated hierarchies
- Queries
  - Determine whether facts are consistent wrt ontologies
  - Determine if individuals are instances of concepts
  - Retrieve individuals satisfying a query (concept)
  - Verify if a concept is more general than another (subsumption)

## Querying a DL Ontology: Example

All the children of John are females. Mary is a child of John.  
Tim is a friend of professor Blake. Prove that Mary is a female.

- $\mathcal{A} \stackrel{\text{def}}{=} \{ \text{john} : \forall \text{hasChild}.\text{female}, (\text{john}, \text{mary}) : \text{hasChild},$   
 $(\text{blake}, \text{tim}) : \text{hasFriend}, \text{blake} : \text{professor} \}$
- Query:  $\text{mary} : \text{female}$  (or: is  $\mathcal{A} \sqcap \text{mary} : \neg \text{female}$  unsatisfiable?)
- Yes

# Exercise

Given:

- a set of basic concepts: {Person, Male, Doctor, Engineer}
- a set of relations: {hasChild}

with their obvious meaning. Write a  $\mathcal{T}$ -box in  $\mathcal{ALCN}$  defining the following concepts

- Female, Man, Woman (with their standard meaning)
- femaleDoctorWithoutChildren: female doctor with no children
- fatherOfFemaleDoctor: father of at least two female doctors
- motherOfDoctorsOrEngineers: woman whose children are all engineers or <sup>a</sup> doctors

---

<sup>a</sup>non-exclusive or.