Semantic Data

Chapter 1 : Semantics and knowledge representation

Jean-Louis Binot

Sources and recommanded readings

- There are no additional references required for this chapter.

- Sources and useful additional readings :
  - The discussion on inheritance and its examples is inspired from Brachman and Levesque *(Knowledge Representation and Reasoning, 2004)*.
  - Poole and Mackworth *(Artificial intelligence: foundations of computational agents*, 2nd *edition, 2017)* provide an interesting and recent coverage of various aspects of AI, including knowledge representation.
  - Wood's seminal paper : *What's in a link, foundations for semantic networks (1975)*, although not recent, provides interesting reading.

- University courses having partially inspired ideas and examples for this chapter :
  - *Semantic Web Technologies, H. Paulheim, Universität Manheim.*

Agenda

1. Semantics and IT
2. Knowledge Representation
3. Semantic networks and frames

Elements of a language (reminder)

- **Vocabulary :** the set of symbols available to form sentences of a language.
- **Syntax :** the set of principles and rules that define well-formed sentences.
- **Semantics :** the set of principles and rules defining the relationship between the symbols and sentences of the language and what they mean (represent).
- **Pragmatics :** the study of a language from the point of view of its usage in context.

These elements are relevant for formal languages as well as natural languages, although they may take different forms.

Semantics

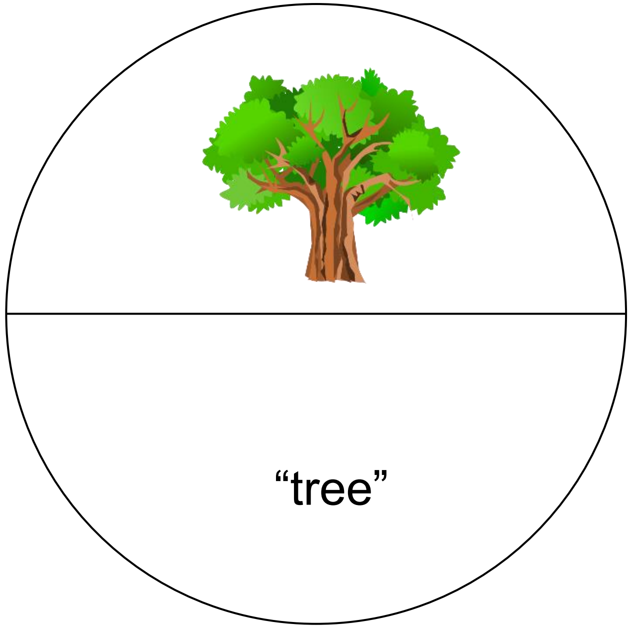
- ❑ Semantics is the linguistic and philosophical study of meaning.

  It is concerned with the relationship between the signs of the language and what they represent.

- ❑ Ferdinand de Saussure's (1857-1913) idea of a linguistic sign :

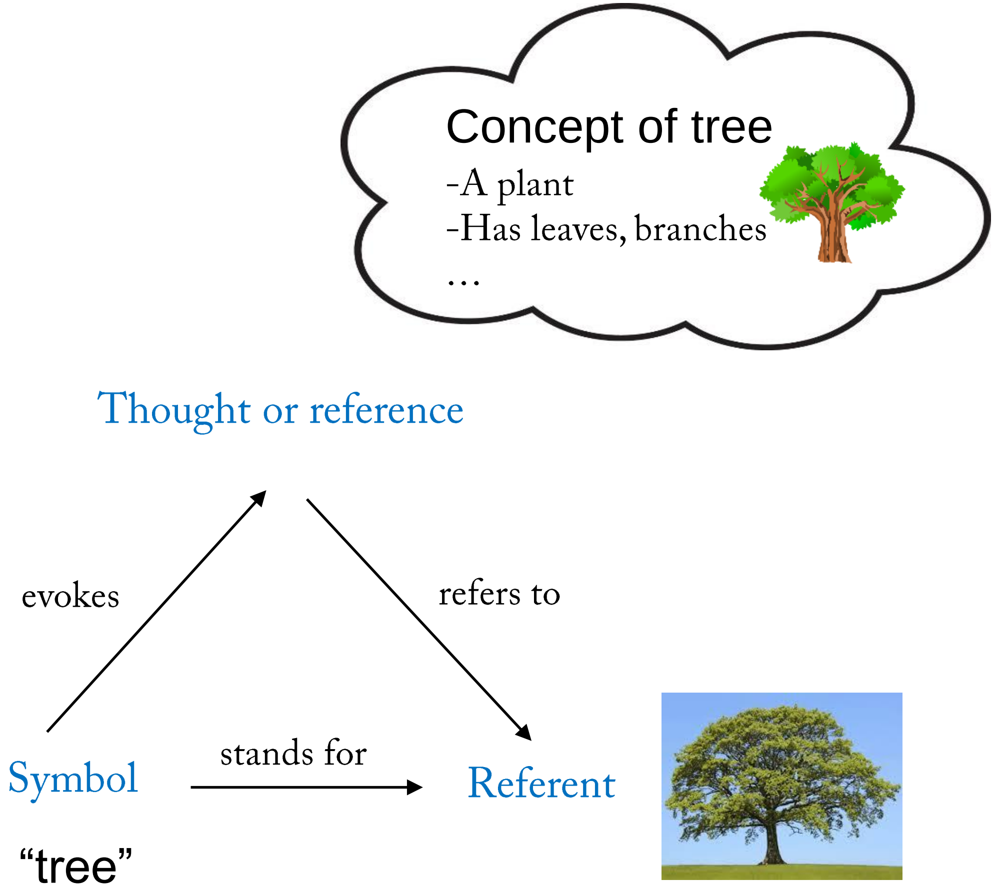
  A sign is formed of two aspects which cannot be separated :

  - The signifier (*signifiant*) – the (acoustic) image of the sign;
  - The signified (*signifié*) – the concept brought in mind by hearing the sign.



Meaning is linked to concepts

- Communication has been explained by the triangle of meaning :
  - The sender of a message uses symbols to communicate.
  - The symbol evokes a concept in the mind of the receiver.
  - The receiver uses his understanding of the concept to identify a world referent.
- What is a concept ?



Concept of tree

-A plant

-Has leaves, branches

…

Thought or reference

evokes

refers to

Symbol

stands for

Referent

“tree”

The triangle of meaning
(Ogden and Richards, 1923)

Concept extension and intension ./.

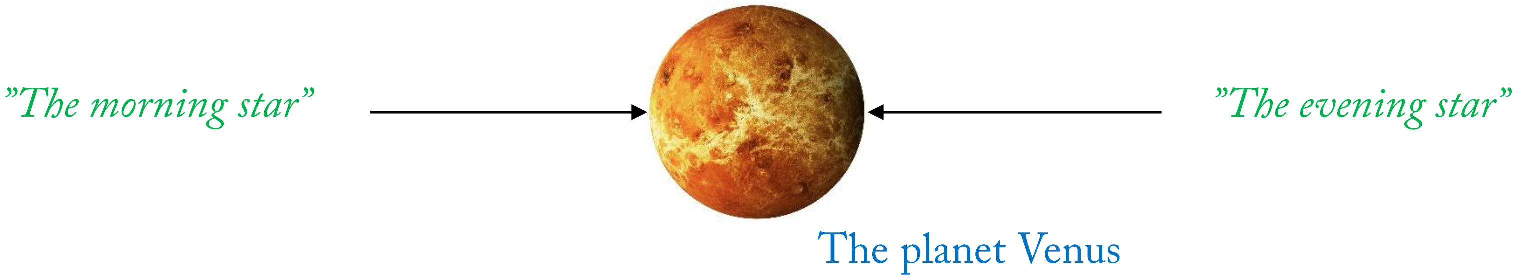
We view a concept as a collection of objects grouped according to their properties.

Since the work of Carnap *(1947)*, a concept is regarded as having two aspects :

- Its intension: the definition (or description) of the concept.
  - It is made of the relations and properties which are a necessary part of that definition.

  A table is a piece of furniture with a flat top with one or more legs, providing an even surface for …

  - Often considered equivalent to what the logician Frege(*) called *sense (1892)*.
- Its extension or referential meaning : the set of objects corresponding to its definition.



"The morning star"

"The evening star"

The planet Venus

*: Frege is regarded as the inventor of modern predicate logic.

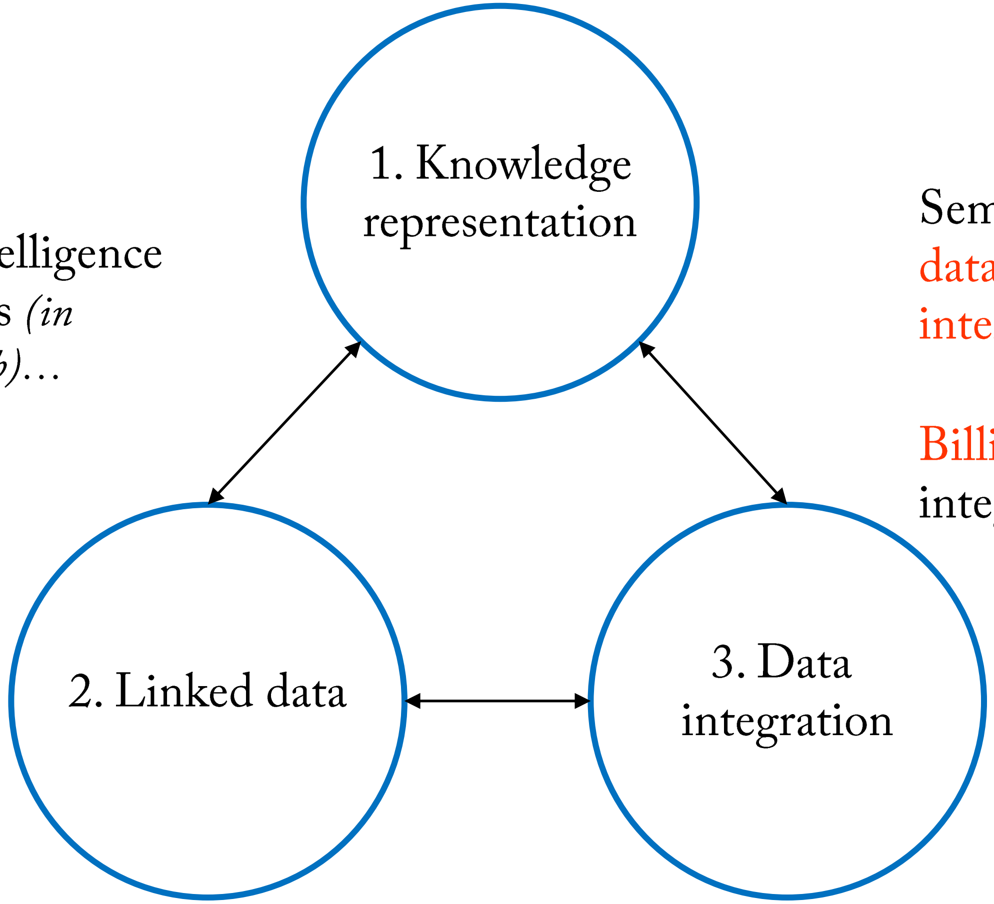
Fine, but …

Why should we, engineers, computer scientists or data scientists care about semantics ?

Why should computer/data scientists care about semantics ?

Semantics is a core part of knowledge representation.

Applications : Artificial Intelligence solutions, knowledge graphs *(in connection with the semantic web)…*



Semantics plays a key role in big data, data integration and system interoperability.

Billions of dollars lost per year in data integration and interoperability costs !

Semantics enables linked data through the semantic web.

Applications : rich content for search engines, virtual assistants, recommendation engines, knowledge graphs …

Applications : multiple domains (health, energy, biogenetics, pharma, media, supply chain …).

Some active industry compagnies

Search and virtual assistants

Google

amazon alexa

Knowledge graphs

Google

aws

Microsoft Bing

IBM Watson

ebay

Uber

Service compagnies

CAMBRIDGE SEMANTICS

SEMANTIC WEB COMPANY

arns GROUP

ODASE

PoolParty Semantic Suite
Software

TopQuadrant™
W3C MEMBER

User applications

NOVARTIS

Walmart

pitney bowes

EU Open Data Portal
Access to European Union open data

NASA

GENEONTOLOGY
Unifying Biology

EPIM10

Montefiore
Inspired Medicine

BBC

Biogen

Graph databases

neo4j

Amazon Neptune
Fast, reliable graph database built for the cloud

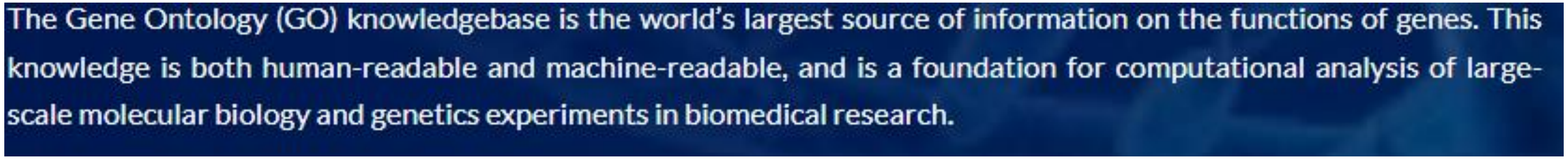
18c Oracle Spatial and Graph

AllegroGraph
Franz Inc.

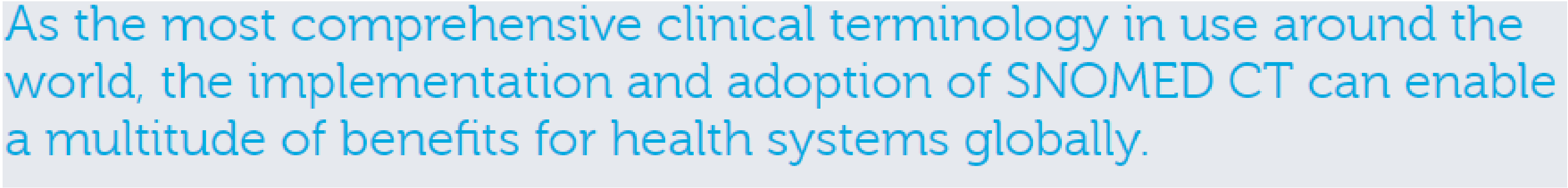
IBM
IBM Graph

Azure Cosmos DB

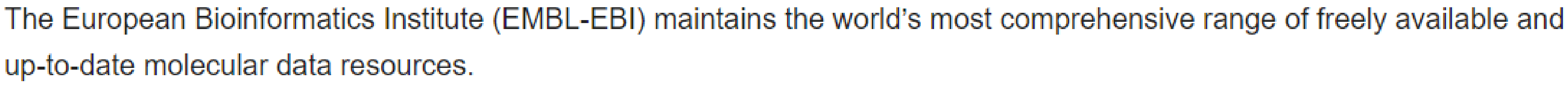
Examples of industrial scale ontology usage


The Gene Ontology (GO) knowledgebase is the world's largest source of information on the functions of genes. This knowledge is both human-readable and machine-readable, and is a foundation for computational analysis of large-scale molecular biology and genetics experiments in biomedical research.

(http://geneontology.org/)


As the most comprehensive clinical terminology in use around the world, the implementation and adoption of SNOMED CT can enable a multitude of benefits for health systems globally.

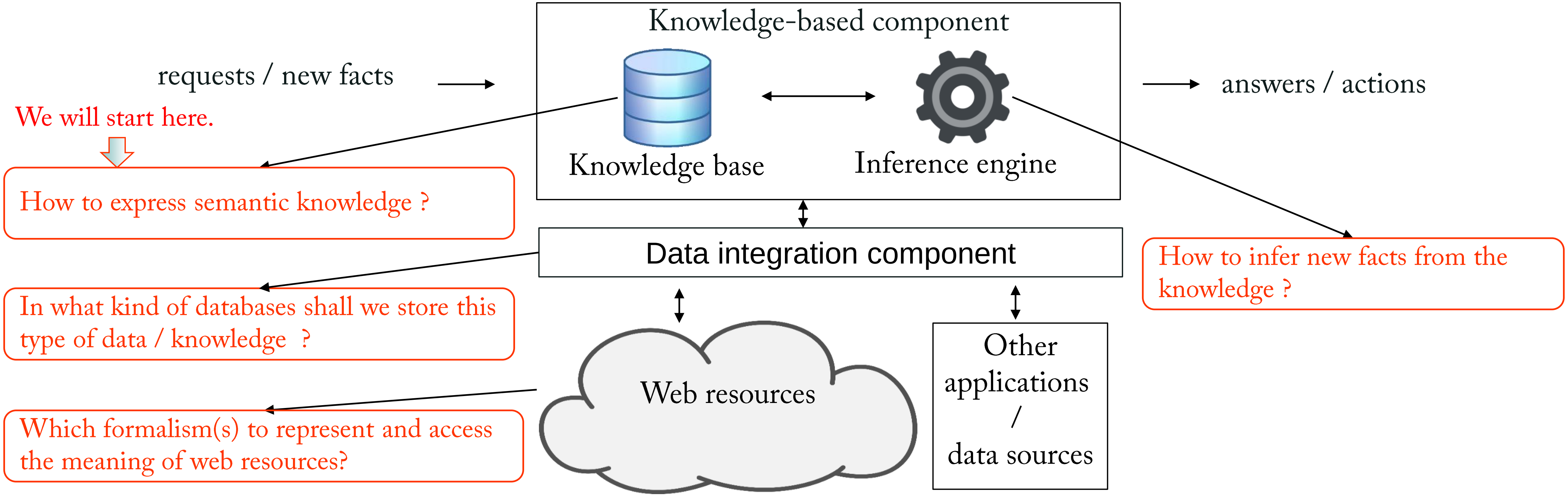
(https://www.snomed.org/snomed-ct/why-snomed-ct)


The European Bioinformatics Institute (EMBL-EBI) maintains the world's most comprehensive range of freely available and up-to-date molecular data resources.

(https://www.ebi.ac.uk/services)

A framework of reference for the questions of this course

We will consider the following questions regarding an application managing semantic data(*) :



requests / new facts

Knowledge-based component

Knowledge base

Inference engine

answers / actions

Data integration component

Web resources

Other applications / data sources

We will start here.

How to express semantic knowledge ?

In what kind of databases shall we store this type of data / knowledge ?

Which formalism(s) to represent and access the meaning of web resources?

How to infer new facts from the knowledge ?

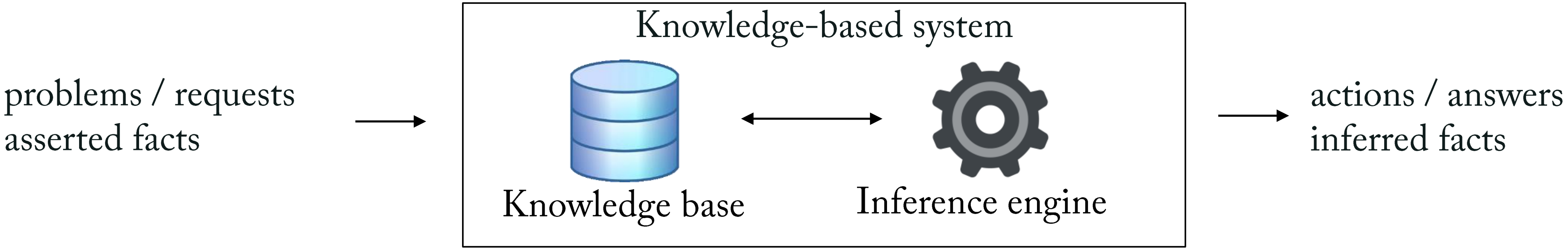
* : Depending on specific needs, more emphasis may be put on some components of such an application.

Agenda

1	Semantics and IT
2	Knowledge Representation
3	Semantic networks and frames

Knowledge representation

In 1958, McCarthy published a paper entitled Programs with Common Sense … (His system) embodied the central principles of knowledge representation and reasoning: … to have a formal, explicit representation of the world and its workings and to be able to manipulate that representation with deductive processes (Russel and Norvig 2010).



problems / requests
asserted facts

Knowledge-based system

Knowledge base

Inference engine

actions / answers
inferred facts

Knowledge : the information about a domain that can be used to solve tasks in that domain (Artificial intelligence: foundations of computational agents, Poole and Mackworth 2017).

Knowledge representation : a scheme or approach to represent knowledge and specify its form. Examples include semantic networks, logic, frames, rules, ontologies …

Example of inferences articulated around concepts

Conceptual knowledge ⇨

Capitals of countries are cities.

Each country has exactly one capital.

A city cannot be the capital of more than one country.

“Madrid is the capital of Spain”

How much information can you get from this sentence ?

Madrid is the capital of Spain.

Madrid is a city.

Barcelona is not the capital of Spain.

Madrid is not a country.

Spain is a country.

Madrid is located in Spain.

Madrid is not the capital of France.

…

(example from Paulheim, Semantic Web Technologies)

Inferences ?

- ❑ To infer : *to form an opinion or guess that something is true because of the information that you have (Cambridge online dictionary).*
- ❑ Inference : *a conclusion reached on the basis of evidence and reasoning (Google online Dictionary, Oxford Languages).*

Knowledge base versus data base

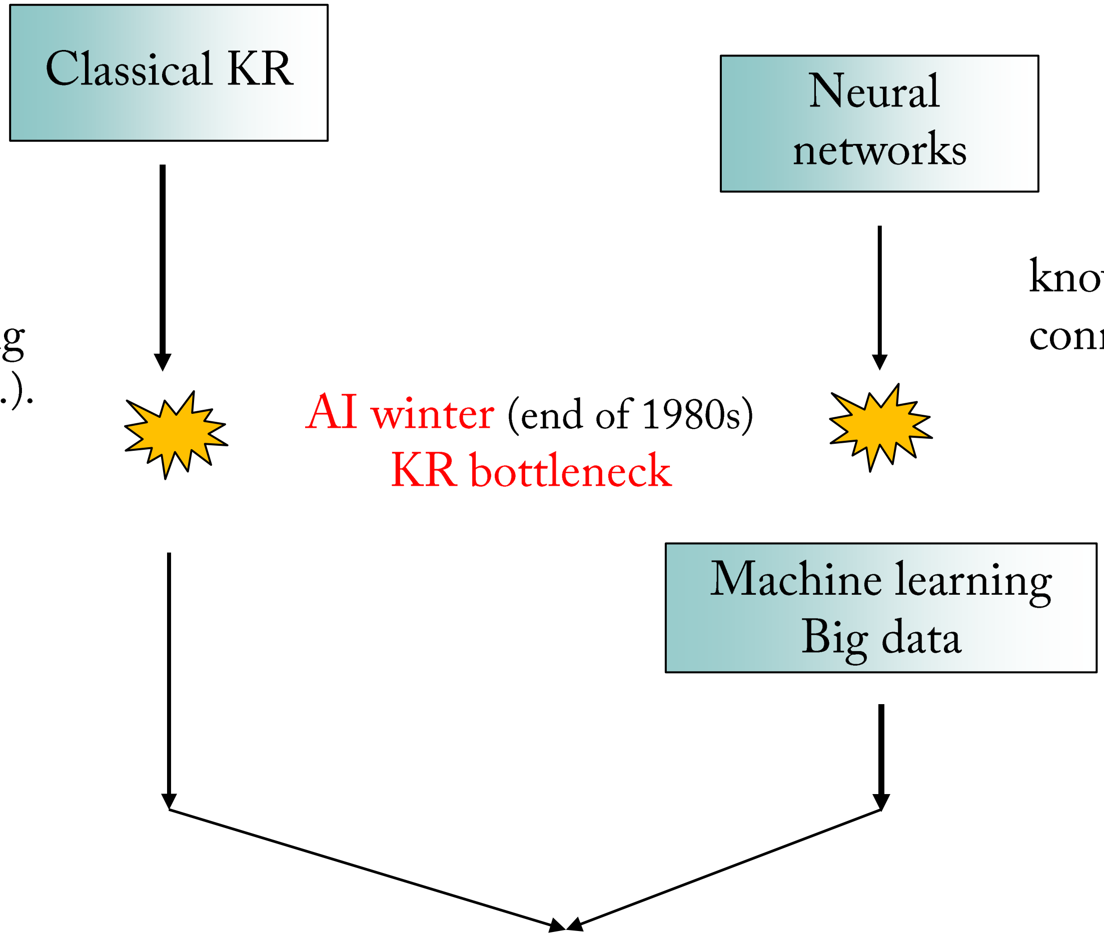
- Knowledge is not the same as data.
- A knowledge base is more than a database (is a special kind of database) in that it tries to structure information in a way emulating human knowledge in order to infer new data and to solve tasks.

(Databases have moved towards knowledge bases with rule extensions but do not address knowledge content.)

Explicit versus embodied knowledge

- In deep learning approaches, one may talk about inference when applying a trained network to new data.
  The knowledge is embodied in the network.
- In symbolic reasoning, we use explicit knowledge to derive new facts, conclusions, plans for action.

Knowledge representation : a simplified AI perspective



Classical KR

Neural networks

Symbolic approach :
explicit knowledge is needed for understanding, reasoning and acting (KR formalisms, rule-based systems …).

Connectionist approach :
knowledge is captured in the strength of connections in networks.

AI winter (end of 1980s)
KR bottleneck

Machine learning
Big data

Semantics, logic

Statistics, probability, neural networks

The current view is that the connectionist and symbolic approaches are complementary, not competing (*Russel and Norvig 2010*).

Two questions

1. What types of knowledge can we consider ?

2. What formalisms can we use to represent explicit knowledge ?

Types of knowledge ?

- Epistemology *(from the Greek ἐπιστήμη, epistēmē, knowledge, and λόγος, logos, reason) :*

  the field of philosophy concerned with the study of the nature of knowledge (what it is « to know »).

  Basic distinction between knowing-how and knowing-that *(Ryle, The concept of mind, 1949).*
  In AI terms, it more or less corresponds to a distinction between :

  - Declarative knowledge (knowledge-that): about descriptive statements, or propositions.
  - Procedural knowledge (knowledge-how): about how to perform tasks, about skills.

- Cognitive psychology *(scientific study of mental processes) :*

  - Declarative knowledge : cognizance of objects, events, ideas …
  - Structural knowledge : how concepts within a domain are interrelated.

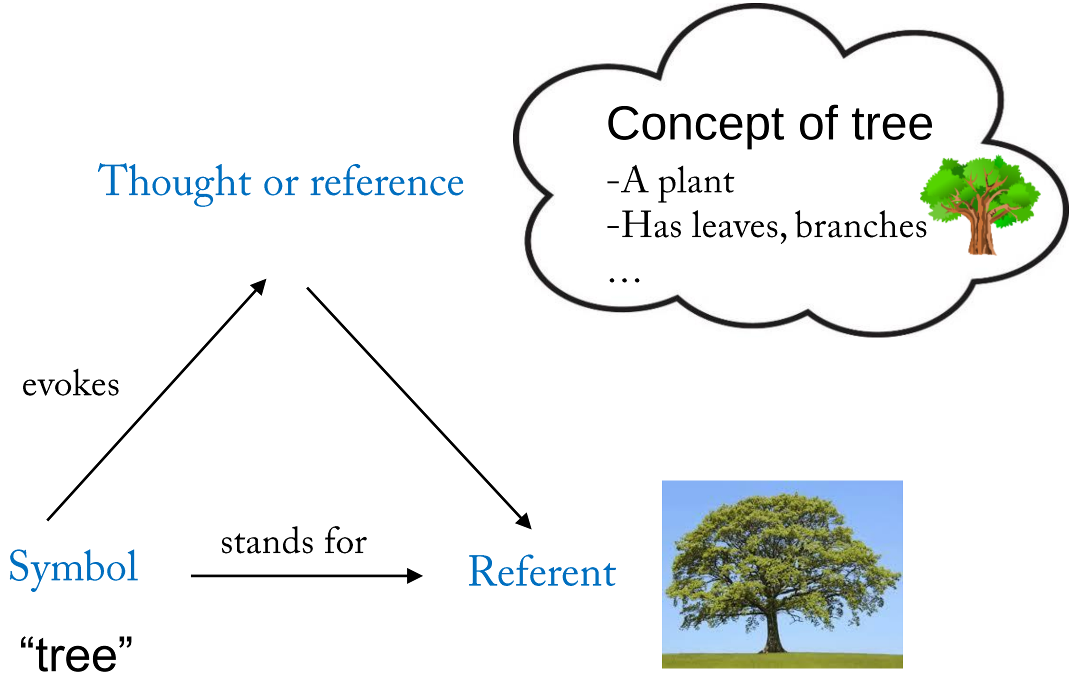
  Semantic knowledge

  - Procedural knowledge : how learners use of apply declarative knowledge.

  (Jonassen et al. 1993).

Semantic knowledge

- Semantic knowledge is :
  - Knowledge that a person acquires.
  - Linked to the meaning of words.



Concept of tree

-A plant

-Has leaves, branches

…

Thought or reference

evokes

Symbol

“tree”

stands for

Referent

- **A large part** of human knowledge is semantic knowledge.
- **A large part** of this knowledge is organized around concepts or categories.

(Russel and Norvig 2010)

Knowledge representation formalisms ?

What we will cover.

- ❑ Semantic networks : graph-based representations of semantic knowledge.
- ❑ First order predicate logic : the language of reference of formal logic.

  Proposition logic is not powerful enough.
- ❑ Description logics : logic-based representations of conceptual knowledge.
- ❑ Ontologies : knowledge bases of semantic knowledge.
- ❑ Frame-based systems : object-oriented structures to represent stereotyped situations.
- ❑ Rule-based systems : conditional rules to reason about implications and consequences.
- ❑ Horn logic : a subset of first order logic; the basis for the Prolog language.
- ❑ Bayesian networks : probabilistic reasoning about uncertain knowledge.
- ❑ …

How do we define a formalism / formal language ?

- Syntax (reminder) :
  - Through a grammar : (recursive) rules defining complex syntactic structures in terms of atomic ones (e.g., a Backus-Naur definition).

- Semantics : many ways, falling in a few broad categories(*) :
  - Procedural : the semantics of a language are defined by reference to the behavior of a procedure, or an abstract machine, that will use or execute a given expression in the language.
  - Declarative : the semantics are defined by declarative definitions in an appropriate formalism.

  The logic programming language Prolog, for example, has both declarative and procedural semantics.

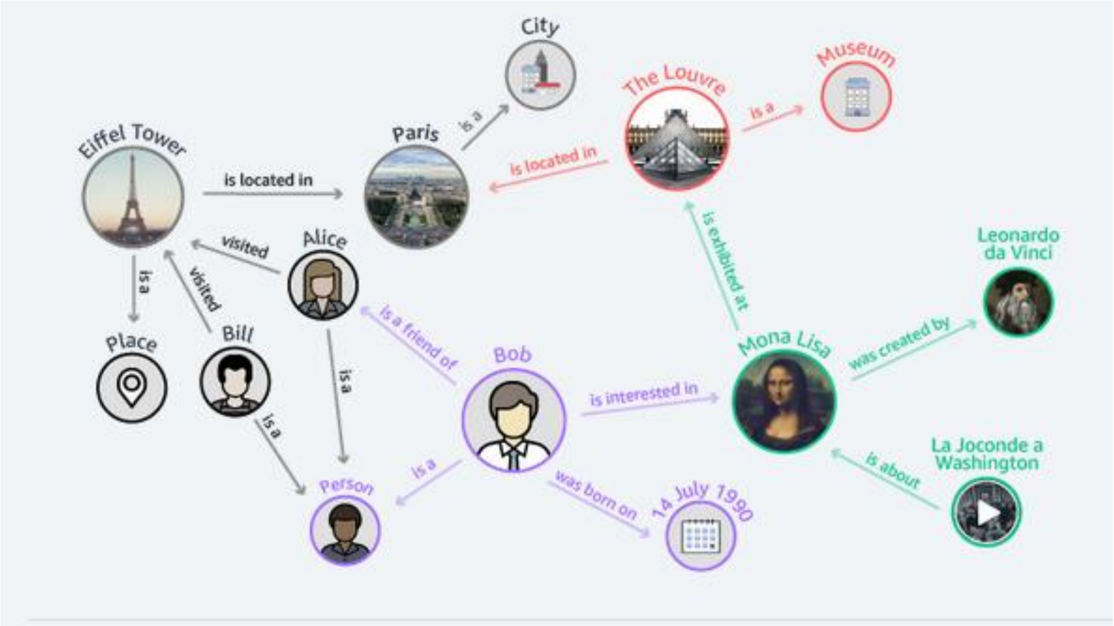
- We aim for declarative semantics :
  - Uniform behavior for interpretation of meaning and inferences.
  - More easily adoptable and sharable among a large community.

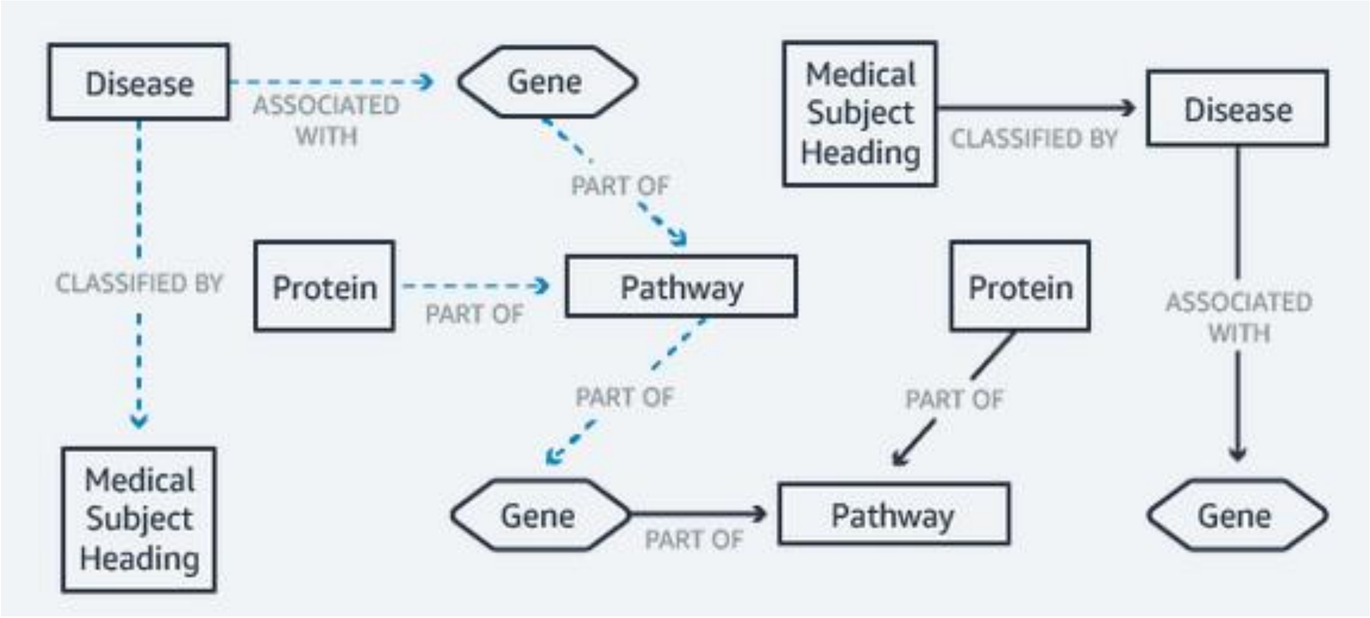
(*)*: for programming languages, operational, denotational and axiomatical semantics are usually distinguished, but they are outside the scope of this course.*

Agenda

1. Semantics and IT
2. Knowledge Representation
3. Semantic networks and frames

Going back to the examples of the introduction





- These are fragments of knowledge graphs.

  (From Amazon Neptune : new graph database by Amazon Web Services, 2018).

- These are also semantic networks !

Semantic networks : definition

Graph structure for representing knowledge in patterns of interconnected nodes and arcs.

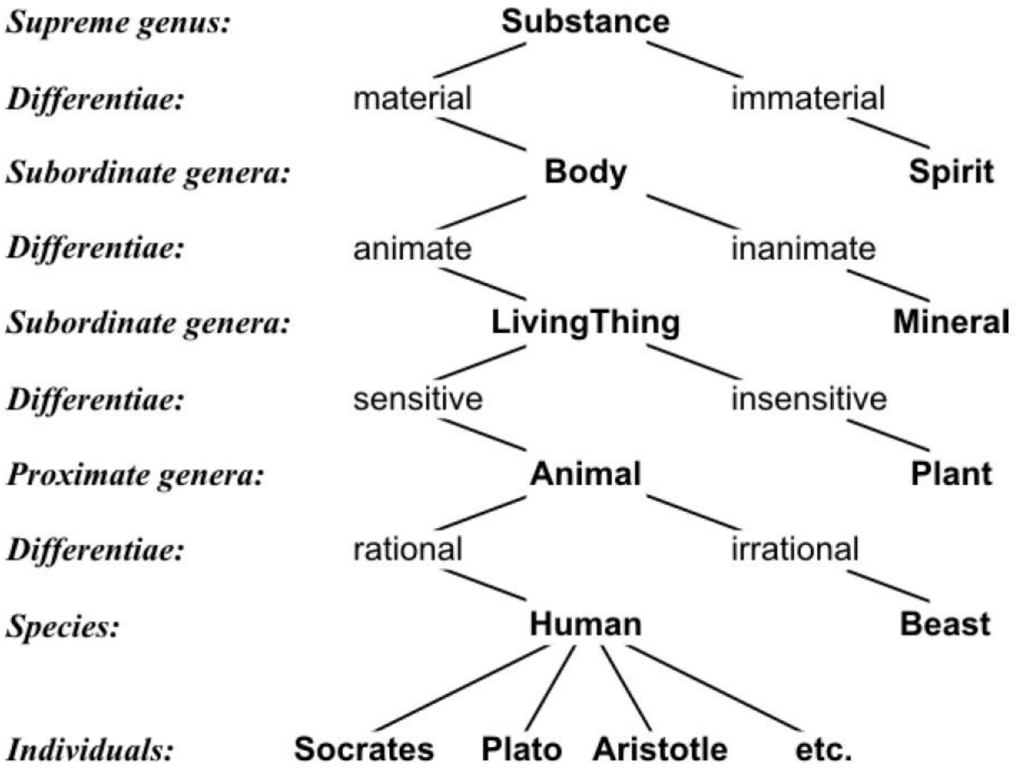
Computer implementations of semantic networks were first developed for artificial intelligence and machine translation, but earlier versions have long been used in philosophy, psychology, and linguistics.

(Sowa, 1992)

How old are semantic networks ?

Note : notations for semantic networks have varied widely. We have chosen a notation for this course with the objective to present the main concepts and issues.

Earliest efforts



Tree of Porphyry drawn by the logician Peter of Spain (1239) *(Sowa 1992)*

Aristotle's categories *(350 BC)* are a major part of the Organon (his set of books about logic).

They attempt to enumerate all possible kinds of things which can be observed and said in language.

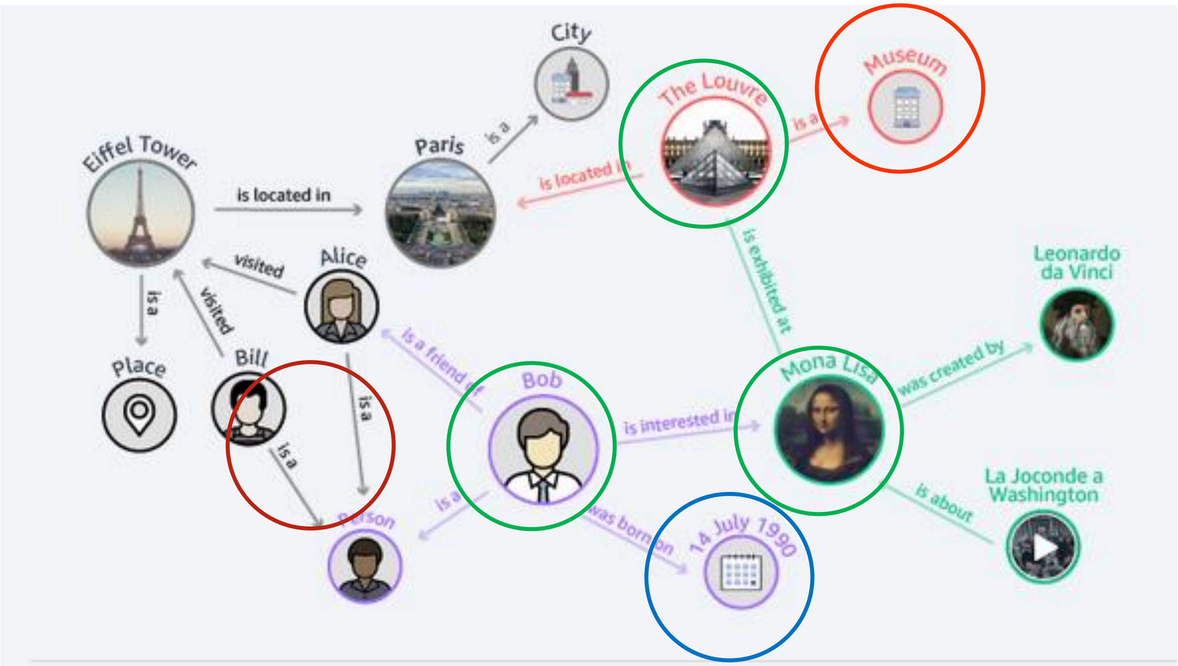
Porphyry's tree : the oldest known semantic network, used by the Greek philosopher Porphyry *(3d century AD)* to represent Aristotle's categories.

Historical perspective

- Porphyry's tree (*3d century AD*) – philosophy.
  - The first known attempt to visualize a concept hierarchy.
- Pierce's existential graphs *(Pierce 1909)* – philosophy / logic.
  - A visual notation for logical expressions.
- Richens' first semantic nets for computers *(Richen 1956)* : for machine translation.
- Quillian's semantic memory model *(Quillian 1968)* : psychology / computer sciences.
  - A representation attempting to model a part of human memory to encode dictionary word senses and sentence meanings; has strongly influenced early AI semantic networks.
- Various formalisms used in AI *(1970s).*
- Basic ideas and techniques spread to other computer science areas :
  - ER modeling, software design (UML…), OO programming (Java...), etc.
  - But these do not have the same objectives, principles and constraints as knowledge representation.

Semantic networks : basics

- A semantic network is a directed labelled graph :
  - Directed : the arcs between the nodes have a direction.
  - Labelled : both nodes and arcs have labels attached to them.
- Nodes represent concepts, objects, values, events.
- Arcs represent relationships between them.
- Concepts are usually structured into hierarchies.
  - Specialization/generalization hierarchy (the "*infamous*" ISA links).
  - Other types of hierarchies, e.g. part-of hierarchies (cf. previous example knowledge graph on biomedical knowledge).



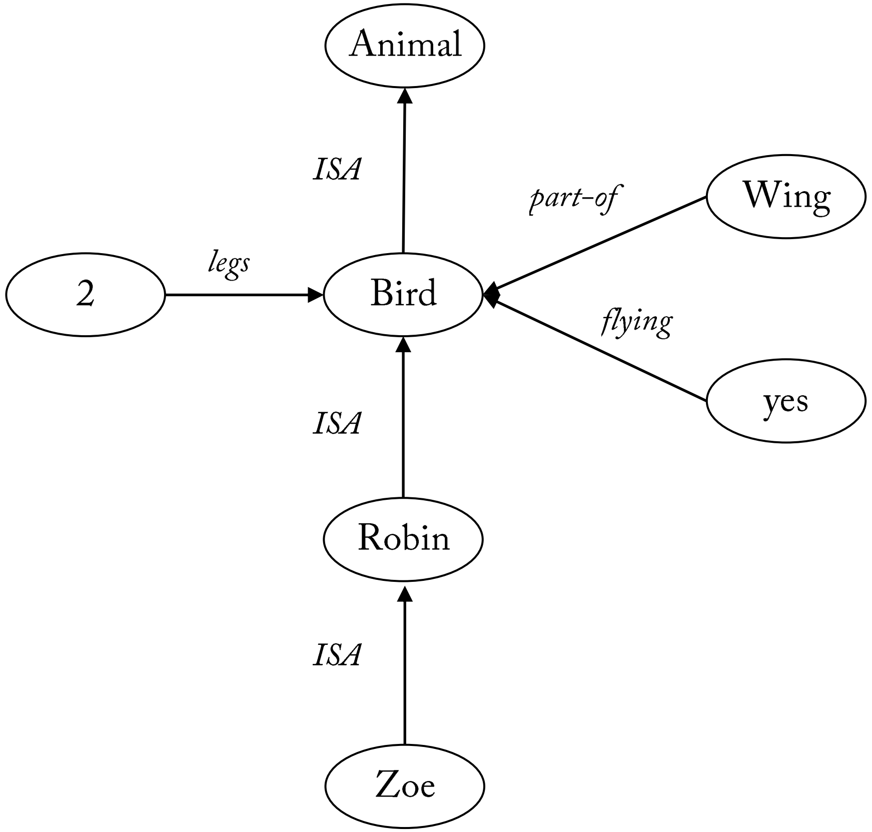
Specialization hierarchies

- ❑ Specialization hierarchies may seem well-known.
- ❑ They still open, however, a number of complex issues.
- ❑ We are looking at them from a knowledge level perspective.

  A knowledge representation is not a data structure. Our questions :

  - What does the representation mean ?
  - What reasoning processes / inferences are supported ?
  - What are their limitations ?

- ❑ A simple example :
  - Is there a problem with this network ?



Well-known example going back to (*Quillian 1968*).

(For the sake of simplicity, we also represent property values as nodes.)

Specialization hierarchies ./.

As a concept is a collection of individuals, we must distinguish between instance and subclass :

- *Zoe* is a member of the set of entities representing the extension of the class *Robin*.
- Every member of the extension of the class *Bird* is in the extension of the class *Animal*.

❑ The ISA relationship does not make that distinction !

We replace it by two different and more precise relationships :

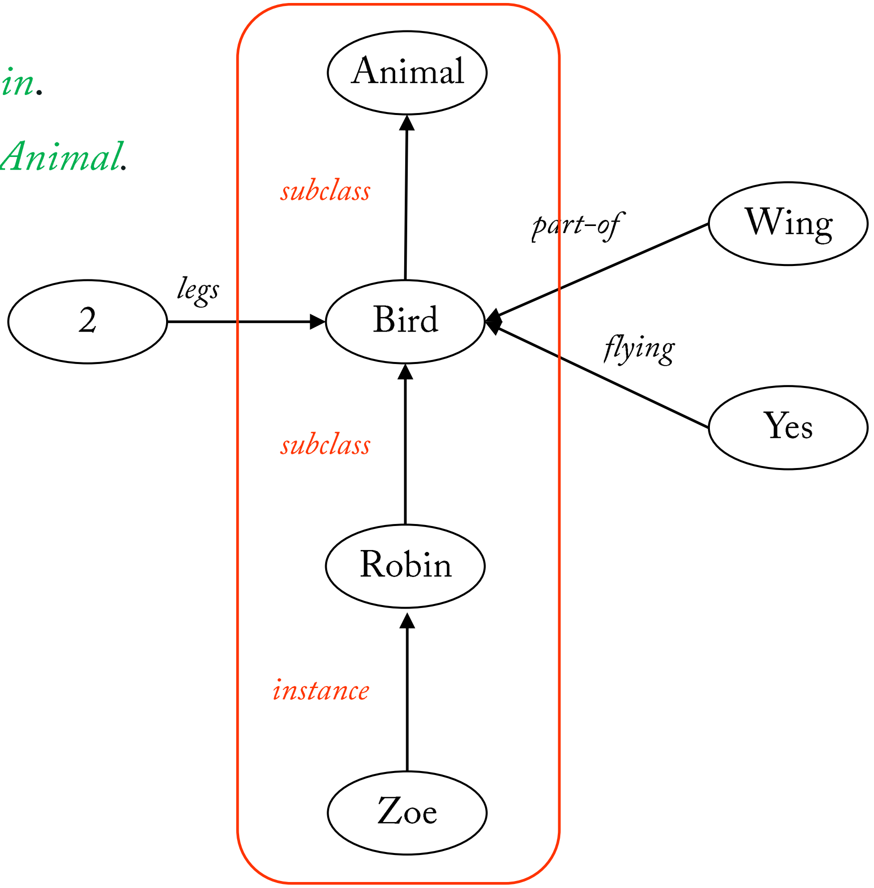
- subclass corresponds to set inclusion : $\subseteq$

Bird is a subclass of Animal.

- instance corresponds to set membership : $\in$

Zoe as an individual is an instance of the class *Robin*.

❑ These relations form the generalization / specialization structure.

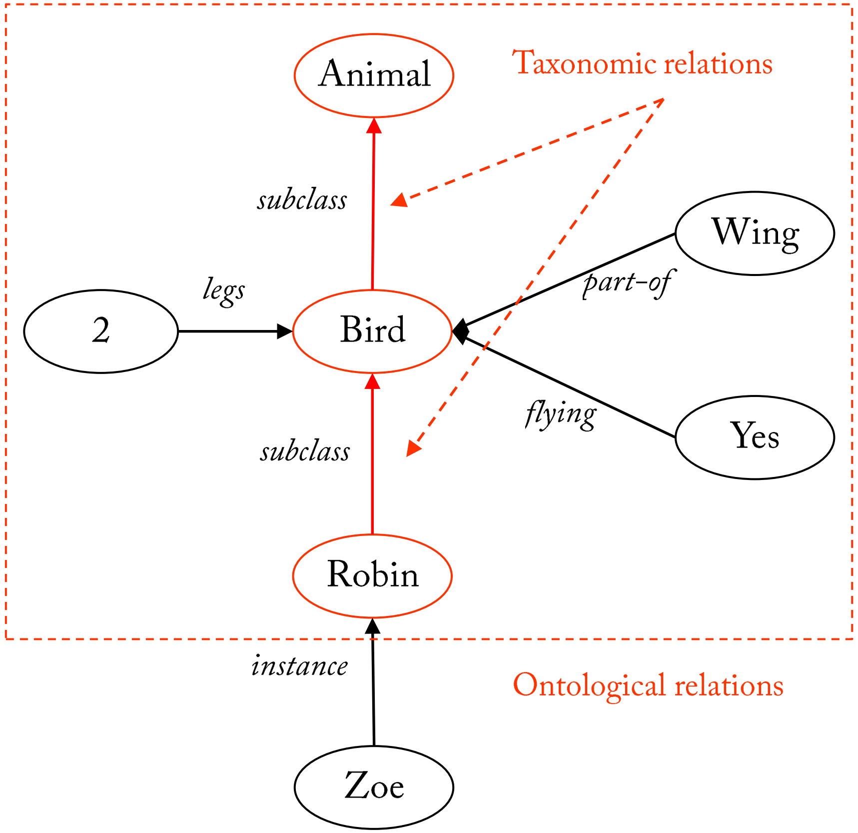


Taxonomies and ontologies

Taxonomies and ontologies are structuration tools.

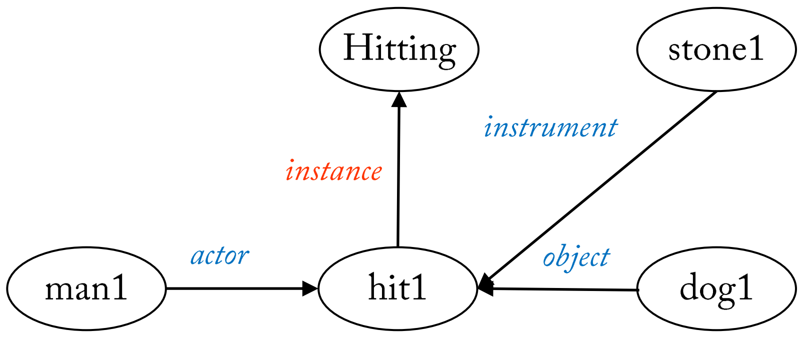
They are used to organize the representation of the semantic knowledge about the world or about specific domains :

- ❑ The generalization/specialization hierarchy forming the backbone of the representation provides a taxonomy of that domain.
- ❑ The richer set of relations is a representation of an ontology of that domain.
  - Instances are usually not considered as parts of a taxonomy or ontology.



Representing events

- How to represent in a network events with more than two arguments ?
  - Consider the sentence *A dog was hit by a man with a stone*.
- The solution, called reification(*), is to turn the relation into an object.
  - The event is represented by an event node, instance of the appropriate event class.
  - This node is linked to the instances involved in the event by specific relationships describing their role.
  - These relationships are called semantic cases and differ from syntactic roles.



* : from the latin “res”, meaning “thing”.

Reasoning with semantic networks

Main reasoning processes supported by semantic networks as a KR formalism :

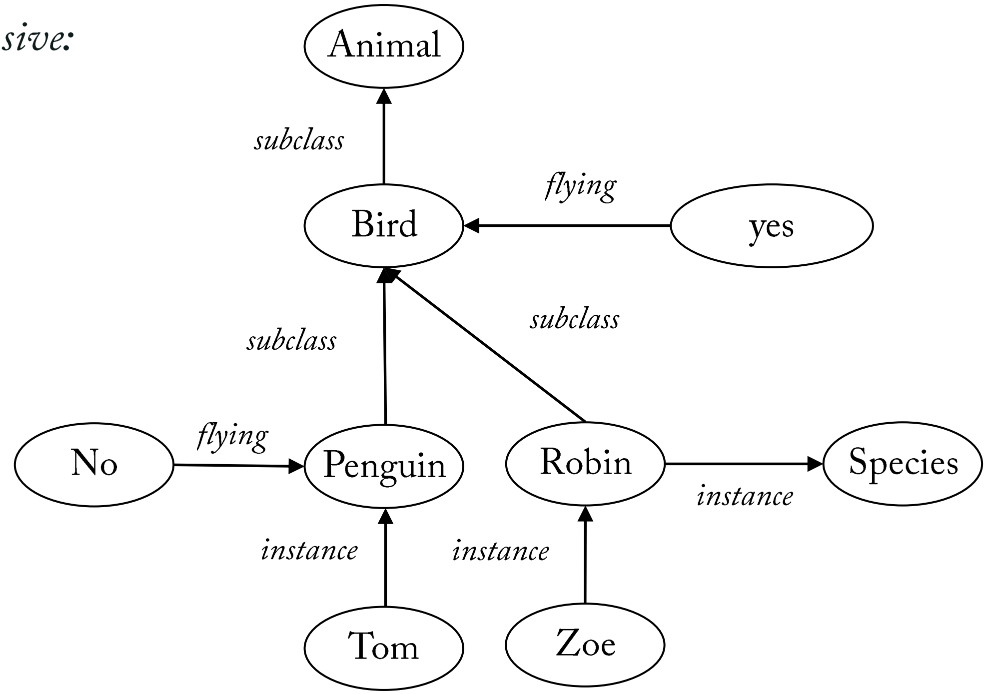
- Spreading activation
  - Identifying semantic relationships by finding the shortest path (semantic distance) between nodes.
  - Mostly used in historical cognitive approaches to natural language processing.
- Subsumption checking
  - Validating a classification by checking if a class is included in another.
- Inheritance
  - Inheriting information down a generalization/specialization hierarchy.

Subsumption checking and inheritance are fundamental for knowledge representation.

(Note : modern large-scale knowledge graphs also use machine learning inference techniques).

Subsumption

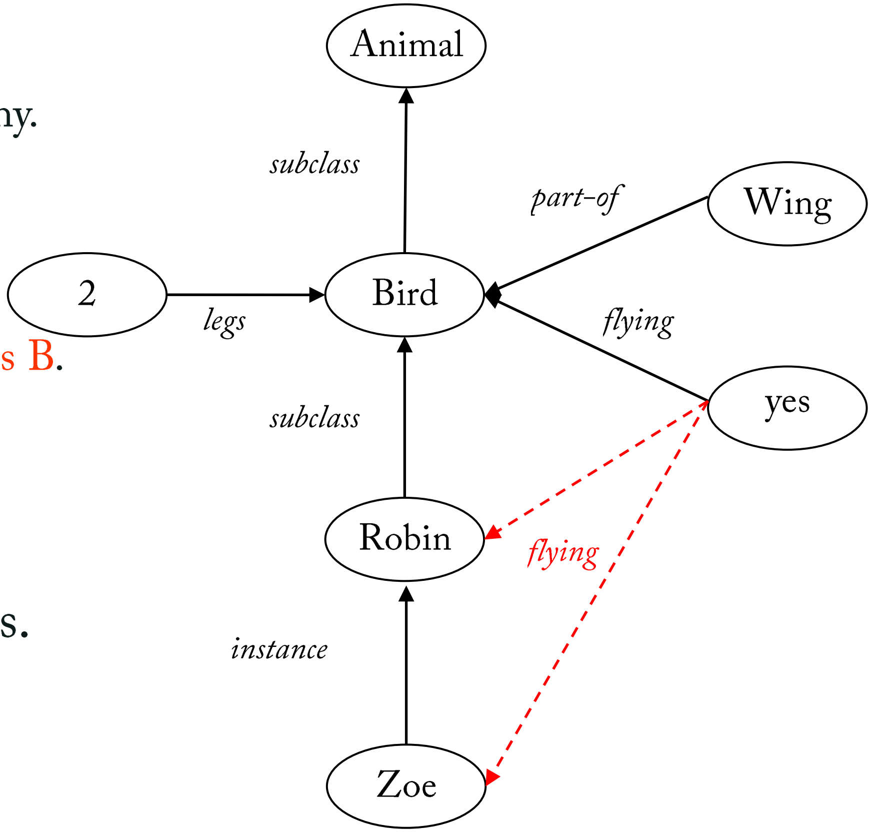
- The subsumption relationship is the inverse of subclass.
  - To subsume : *to include or place within something larger or more comprehensive: encompass as a subordinate …* (Merriam Webster).
- Class A subsumes class B iff $B \subseteq A$ (note the sense of the inclusion).
  - All instances of B are necessarily instances of A.
- The subsumption relation is transitive :
  - If A subsumes B and B subsumes C, A subsumes C.
  - *Bird* subsumes *Robin*; *Animal* subsumes *Bird* and *Robin*.
- The instance relation is not transitive.
  - *Zoe* is an instance of *Robin* but not an instance of *Species*.



Inheritance

- Properties can be inherited by individuals down the generalization hierarchy.
  - By being a *Robin*, hence a *Bird*, *Zoe* inherits the property of *flying*.
  - Information is expressed only once, at the most appropriate level in the hierarchy.
- Inheritance is strongly linked to subsumption.
  - To conclude that class B inherits from class A, one must check that A subsumes B.
- Is inheritance always possible ? Simple ?
- No. Subsumption and inheritance are sources of many complexities.
  - These depend on the type of network and inheritance considered.

(The discussion that follows is strongly inspired from Brachman and Levesque 2004).

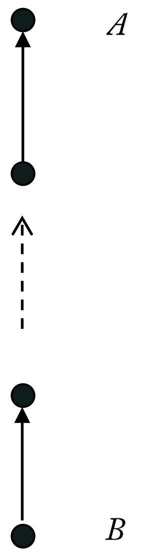


Checking subsumption

- How can we check if : class A is subsuming class B ?
  - It is not an obvious question in every knowledge representation language.
    In description logics, for example, subsumption has to be checked by logical inferences.
- In a semantic network, checking subsumption looks simple in theory(*) :
  one checks the graph.

  Is A subsuming B ? involves asking :
  - Is there a path from B to A using the generalization relation (subclass) ?
  - Or: is A in the transitive closure(**) of that generalization relation from B ?

  Any graph traversal algorithm may be used.
- Let us consider a few specific cases in the next slides.



* : *Assuming that the network has been constructed to avoid the issues discussed later.*

** : *Reminder : for a graph G = (V, E), its transitive closure $G^+ = (V, E^+)$ contains an edge $(v, w) \in E^+$ iff there is a path (v, w) in G.*

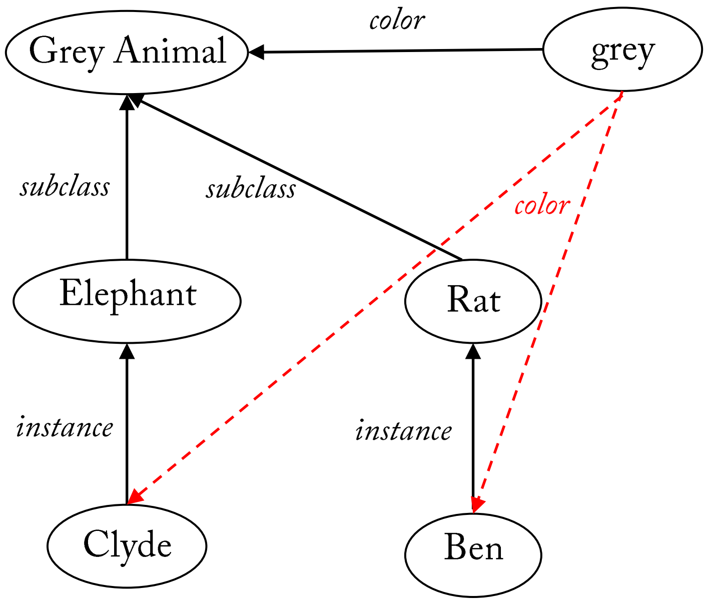
Strict inheritance in a tree

- In strict inheritance, instances of a sub-concept must have all the features inherited from all super-concepts.

Strict inheritance in a tree-structured network is simple :

- All conclusions from all reachable nodes in the subsumption hierarchy are inherited.
  - *Clyde* and *Ben* are both of color *grey*.

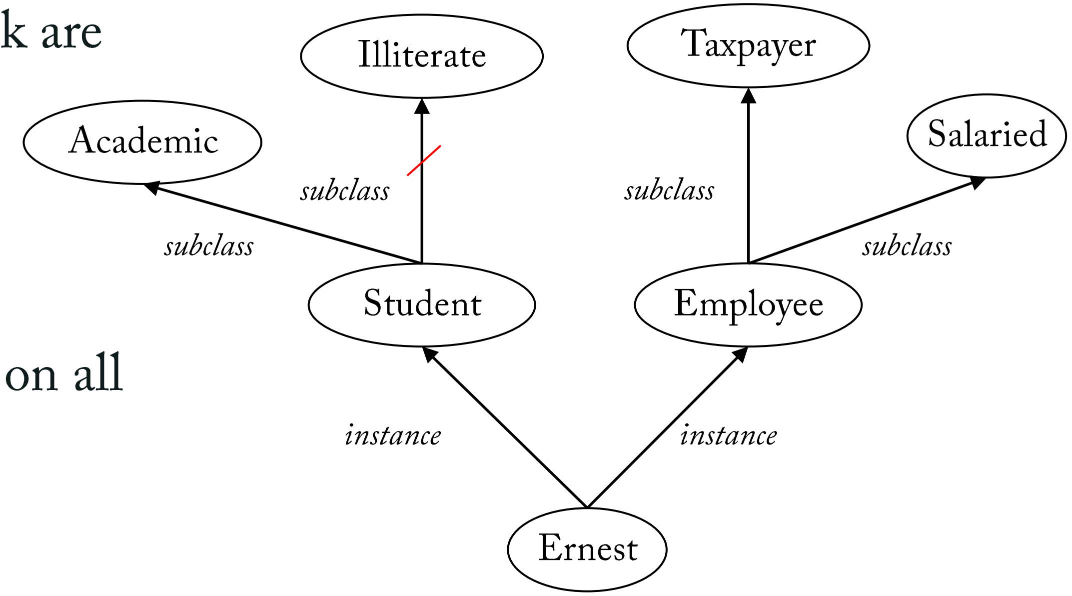
- Any graph-traversal procedure computing transitive closure on all subclass paths will derive correct conclusions.



(from Brachman and Levesque 2004)

Strict inheritance in a direct acyclic graph

- Inheritance in a directed acyclic graph with multiple parents is often called multiple inheritance.
- Strict multiple inheritance behaves as in the tree case : all conclusions from all reachable nodes in the network are inherited.
  - *Ernest* is an *Academic*, a *Taxpayer* and a *Salaried*.
  - A *Student* is not *Illiterate*(*).
- Again, any procedure computing transitive closure on all subclass paths will derive correct conclusions.



(from Brachman and Levesque 2004)

** : links may have polarities : / marks a negative link.*

Issues with semantic networks

Semantic networks present a number of issues and difficulties :

1. Multiple defeasible inheritance.
    a. Ambiguities and inconsistencies.
    b. Non-monotonic reasoning.
2. Distinction between assertions and descriptions.
3. Procedural semantics.

Defeasible inheritance

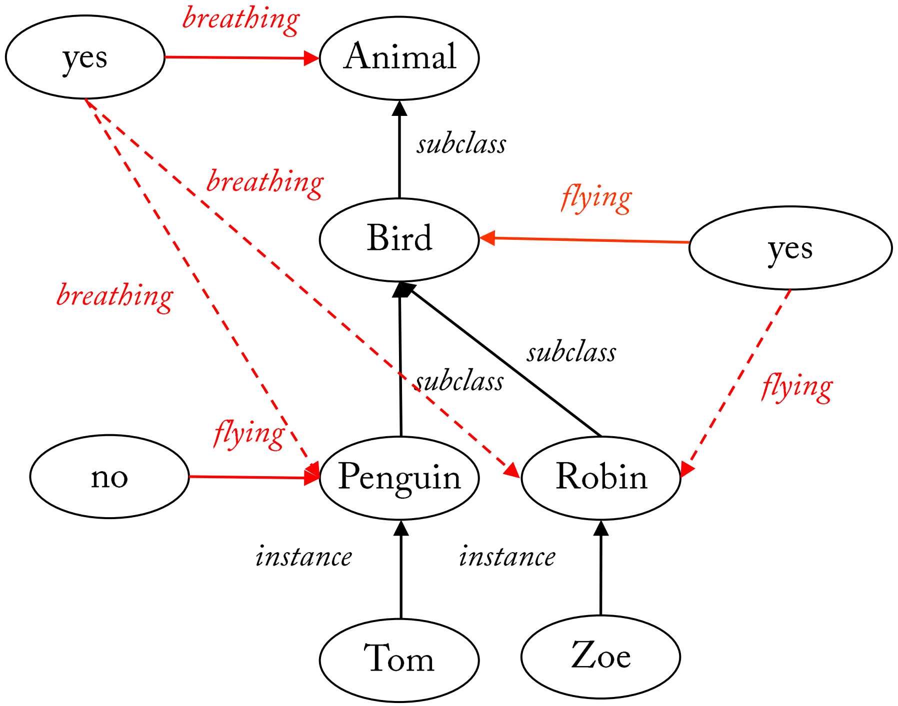
- When modeling a domain of knowledge, we will face exceptions.

  Penguins are birds but they cannot fly.

- In defeasible inheritance, properties are inherited by default :
  - Information at a lower level overrides values inherited from a higher level.
  - This rule prevents inconsistencies in tree-based networks :

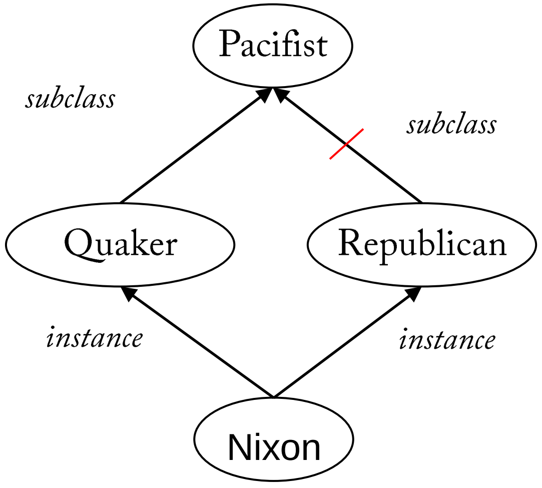
    Penguins are breathing but not flying; robins are breathing and flying.

- Is that always working ?
- No! Several issues :
  - Multiple inheritance.
  - Non monotonic reasoning.



Issue 1a: multiple defeasible inheritance

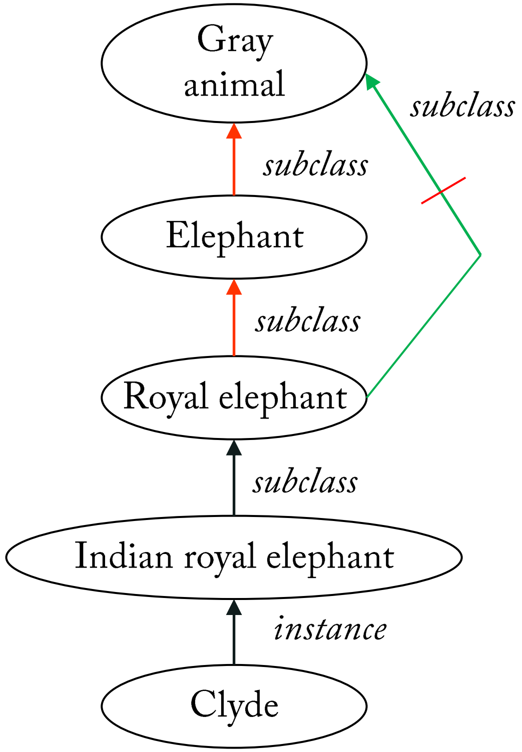
- Defeasible inheritance with multiple predecessors lead to ambiguities and conflicting inferences.
  - Is Nixon a pacifist or not ?
- This issue is complex but pervasive in the real world.
- In this example, the network is truly ambiguous.
- However, we may look for heuristics to reduce cases of ambiguities.



The « Nixon diamond »

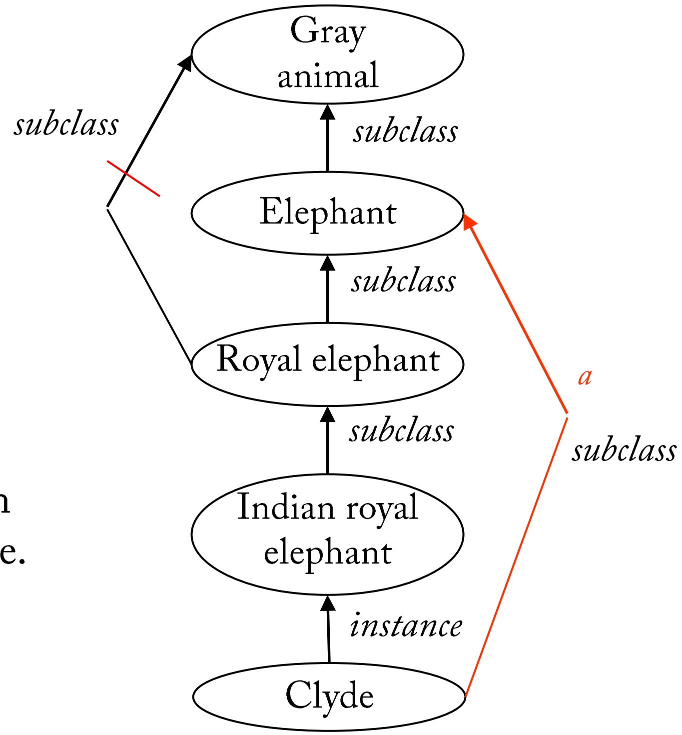
Defeasible inheritance: the shortest path strategy

- A path is an argument in support of a conclusion (inference).
  - Not all paths count in generating conclusions : some paths are preempted by others.
  - Those who are not preempted are admissible.
- The inheritance problem can be recast as : what are the admissible paths (conclusions) ?
- Shortest path heuristic : prefer conclusions from shorter paths in the network.
  - In the *Clyde* example the red path is preempted by the green one.
  - Our initial rule ("inherit from the most specific subsuming class") is a shortest path heuristic.

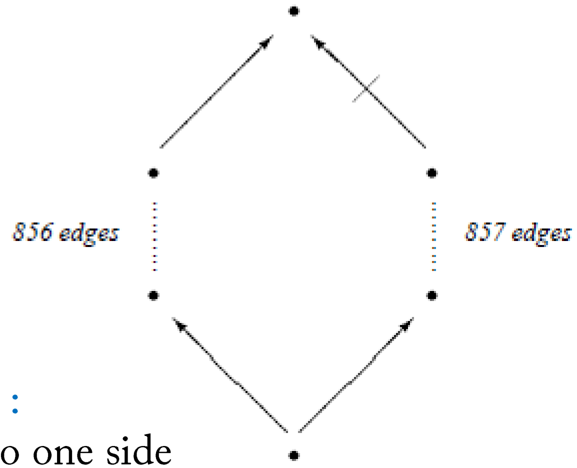


Problems with the shortest path heuritic

1. The presence of redundant (already implied) links can create incorrect answers.
2. The length of the path may not be relevant :
    - Some paths may describe parts of a domain in many details, other be very sketchy.



Example 1 :
the redundant edge a
changes the conclusion
about the color of Clyde.



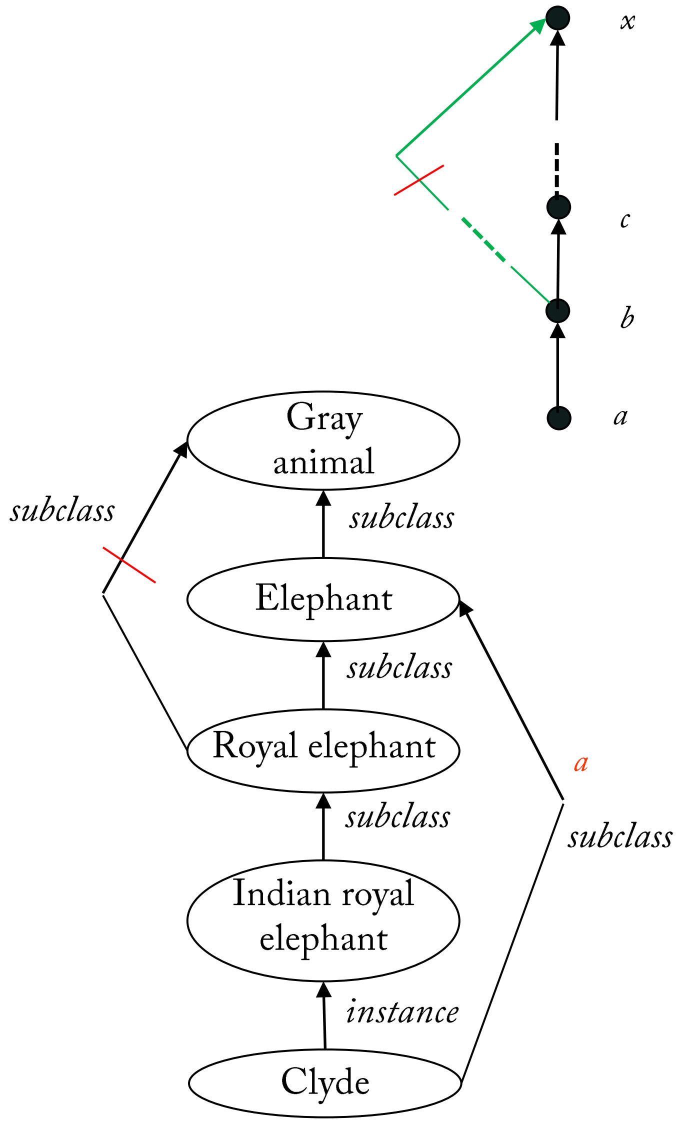
Example 2 :
Adding two edges to one side
should not change the
conclusion ! This network
should be ambiguous !

Defeasible inheritance: the inferential distance strategy

- ❑ The shortest path is not the only possible heuristic.
- ❑ Inferential distance heuristic :
  - Not linear but topology-based.
  - A node a is nearer to b than to c if there is a path from a to c through b.
  - Conclusions inherited from b should then prevent those inherited from c.
  - This handles correctly examples with redundant links.

    Under that heuristic, Royal elephant is more specific than Elephant.
- ❑ This heuristic also has issues :
  - a, b and c are not necessarily adjacent.

  What if path from a through b to c has itself some of its edges preempted, or if some edges are redundant ?



Approaches for multiple defeasible inheritance

- Several formal theories were proposed, none has emerged as agreed upon.
- Possible approaches :
  - Believe the first answer encountered or choose arbitrarily (“credulous” reasoner).
  - Conclude that two answers are possible and try to apply rules to prefer one of them (“model preference" reasoner).
  - Conclude that none of the answers can be deduced and the status is “unknown” (“skeptical" reasoner).
  - Build the network to avoid such conflicts.
  - Exclude multiple inheritance (as it is done in JAVA).
- Some networks will remain ambiguous (cf. the Nixon diamond).

Issue 1b : non monotonic reasoning

- Defeasible inheritance leads to non monotonic reasoning :

  some facts do not remain true when new assertions are added.

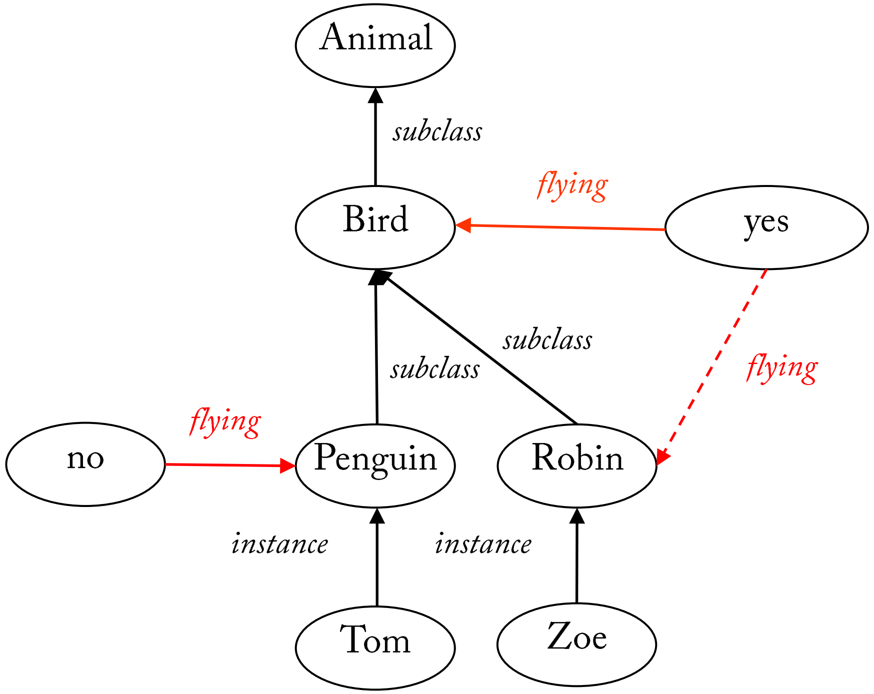
  - Example :

    Without the link stating that penguins cannot fly, we can infer that Tom can fly.

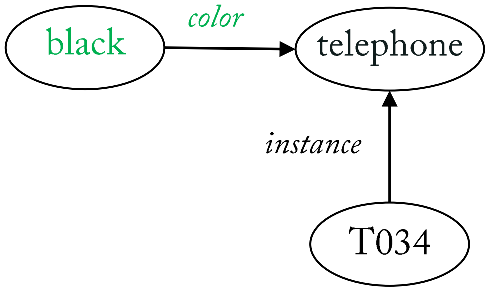
    As soon as we add that link the conclusion changes.

- This topic is still subject to much research today (e.g., default logics).

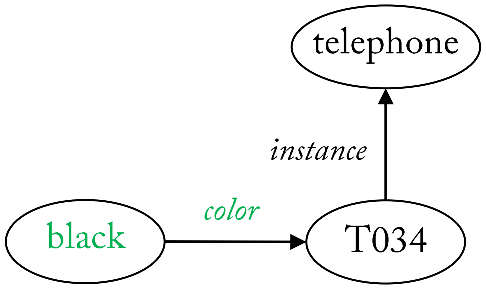
- We will use only monotonic reasoning in this course.



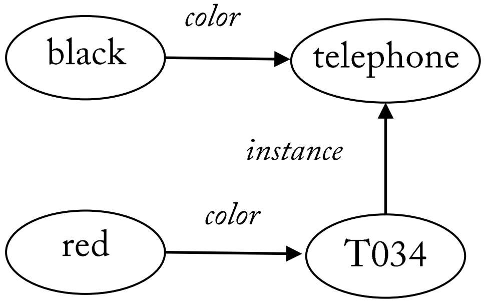
Issue 2: distinguishing assertions from descriptions



Example 1



Example 2



Example 3

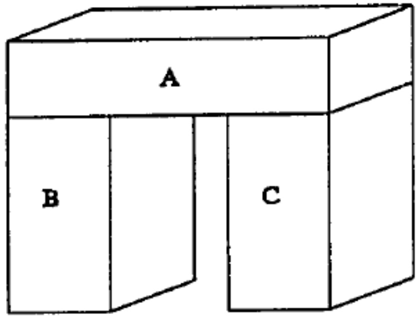
What links are part of the definition of a concept ?

- Example 1 : the *color black* is descriptional : it is part of the intension (description) of the concept *telephone*.
- Example 2 : the same link is assertional: it states a fact about a specific entity.
- Example 3 : the color black is a default property overridden through defeasible inheritance.

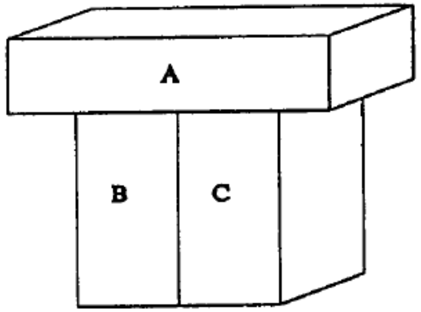
⇒ We need a way to distinguish assertions from concept descriptions. Semantic networks are poor at that.

(Adapted from Woods, 1975)

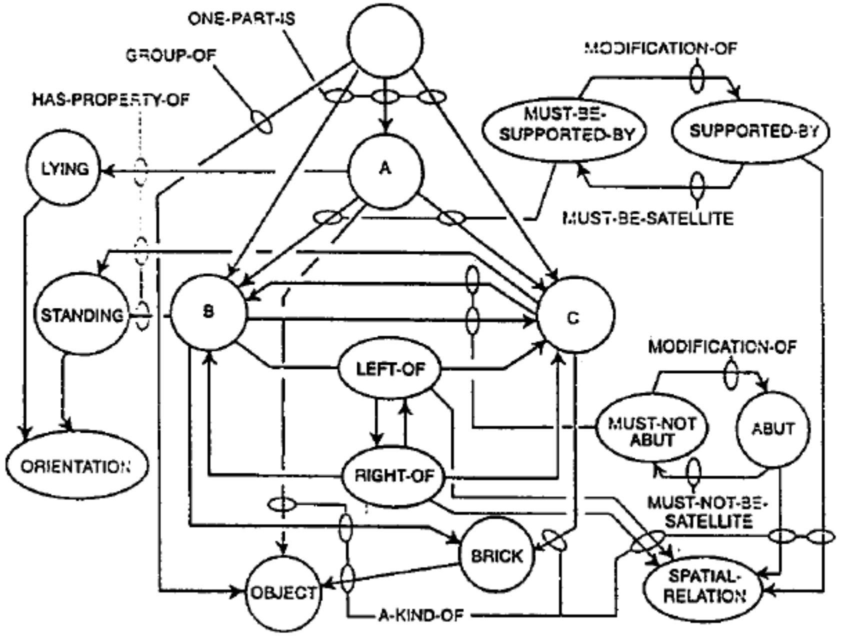
Issue 3 : procedural semantics



Example



Non-example



"*Winston's (1975) semantic net for the 'block-arch' concept.*

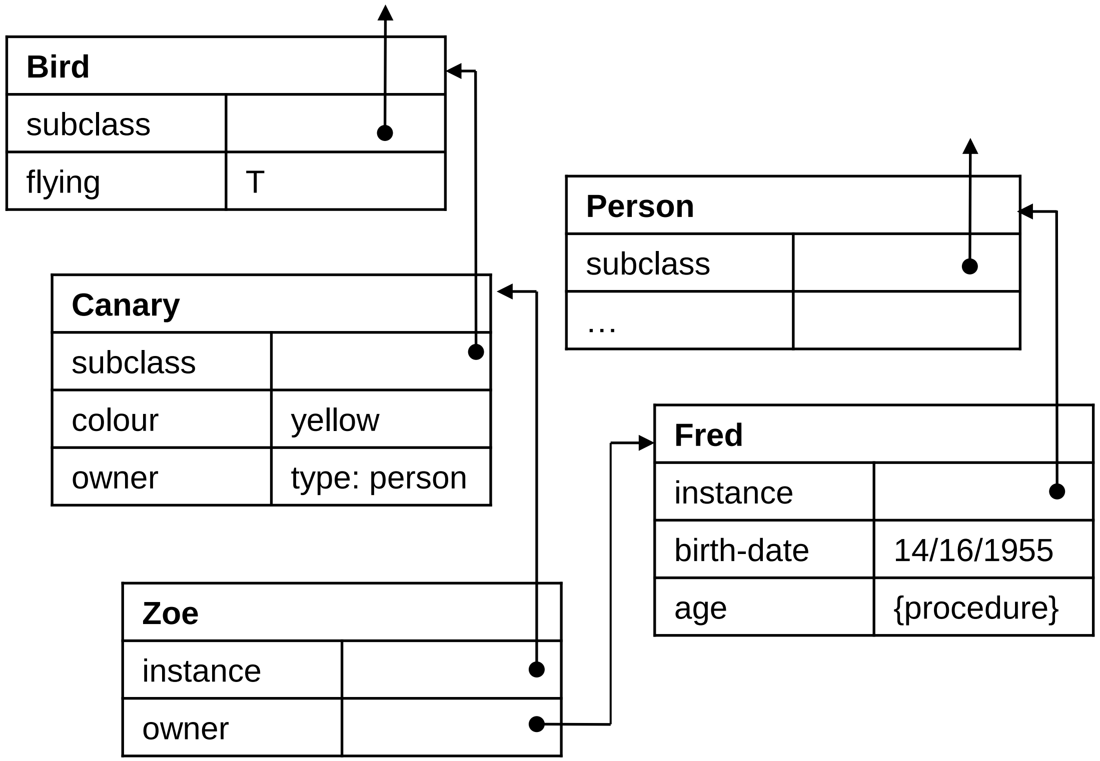
A whole zoo of new objects and relations.
There are new things called *properties* as well as a set (the unlabeled circle). A-KIND-OF links can apply to objects like BLOCKS and to relations like LEFT-OF. MUST indicates a definitional condition and NOT is of course logical…" *(Lehman 1992).*

Would you care to explain what this network means ?

Too many notations ! The semantics of the network is embedded in its interpretation program.
We need a way to define declaratively the semantics of the knowledge representation language.

Frames

- Frames were introduced by Marvin Minsky in his seminal work : *A framework for representing knowledge* (Minsky 1975).
- A frame structures knowledge into substructures by representing “stereotyped situations“ (e.g. being in a living room), using slots and values.
  - These frames can be selected from memory to quickly recognize features of a concrete situation.
- Frames can represent specific entities or concepts. Slots can be filled by :
  - Values, with possible constraints on what values a slot can take. Values can have defaults and be inherited.
  - Procedures (procedural knowledge) computing values or implementing an agent’s reaction to a particular situation.
  - Pointers to other frames.



Bird	
subclass	●
flying	T

Person	
subclass	●
…	

Canary	
subclass	●
colour	yellow
owner	type: person

Fred	
instance	●
birth-date	14/16/1955
age	{procedure}

Zoe	
instance	●
owner	●

(Informal example).

Relationships between frames and semantic networks

- Although there are differences between semantic networks and frames, including in the cognitive intuitions driving them, they have also strong common points :
    - Both are convenient to represent **structured** knowledge about classes, individuals and their relationships.
    - Both can be represented by network structures (directed labeled graphs).
    - Semantic networks were often implemented using frames.
- The main features and issues identified for semantic networks also apply to frames.

The concepts of semantic networks and frames (slots, values, hierarchies, inheritance, defaults, procedural attachment) have found their place in object-oriented programming, with some key differences :

- OO programs must be executable; classes are design objects; multiple inheritance excluded.
- Knowledge representation deals with world-related concepts and models of a complex real world.

Summary

- Semantic networks and frames have useful features for knowledge representation :
  - A clear visualization of the structure of knowledge;
  - Straightforward and usually efficient implementations (links = pointers).

But :

- They have no widely agreed syntax.
- They do not have clear formal semantics :
  - Inheritance depends on the specific heuristic chosen. If not restricted, it can lead to ambiguities, conflicts and significant complexity.
  - There is no easy way to make the distinction between concept descriptions and factual assertions.
  - The semantic interpretation will depend on the program encoding the network.
- While remembering the concepts introduced, we need to explore other formalisms.

References for this chapter

- *[Brachman and Levesque 2004] : Brachman R. and Levesque H.*, Knowledge Representation and Reasoning, *Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.*
- *[Carnap 1947] : Carnap, R.*, Meaning and Necessity, *Chicago University Press, Chicago,1947.*
- *[Frege, 1892] : Frege G.*, Über Sinn und Bedeutung, *Zeitschrift für Philosophie und philosophische Kritik, 1892.*
- *[Jonassen et al. 1993] : Jonassen, D., Beissner, K. and Yacci, M.*, Structural Knowledge: Techniques for Representing, Conveying, and Acquiring Structural Knowledge, *Hillsdale, Erlbaum, 1993.*
- *[Lehman 1992]: Lehman F.*, Semantic Networks, *Computers & mathematics with applications, v 23, Elsevier, 1992.*
- *[Minsky 1975]: Minsky M.*, A framework for representing knowledge, *in (Winston, ed.) "The psychology of computer vision", McGraw Hill, 1975.*
- *[Newell and Simon 1976] : A. Newell and H. A. Simon*, Computer science as empirical enquiry: symbols and search. *Communications of the ACM 19, pp. 113–126.*
- *[Odgen and Richards 1923] : Ogden C. K. & Richards I. A.*, The Meaning of Meaning, *8th Ed. New York, Harcourt, Brace & World, Inc, 1923.*
- *[Pierce 1909] : Peirce, C. S.*, Manuscript 514, *transcribed by Michel Balat with commentary by J. F. Sowa. http://www.jfsowa.com/peirce/ms514.htm.*
- *[Poole and Mackworth 2017]: Poole D. and Mackworth K.*, Artificial intelligence: foundations of computational agents, 2nd edition, *Cambridge University Press, 2017.*
- *[Quillian 1968]: Quillian, M. R.*, Semantic memory, *in Semantic information processing, 227–270, 1968.*
- *[Richens 1956]*, Preprogramming for mechanical translation, *Mechanical Translation 3 (1), July 1956, 20–25.*
- *[Russel and Norvig 2010]: Russel S. and Norvig, P.:* Artificial Intelligence: A Modern Approach (3rd Edition), *Pearson, 2010.*
- *[Ryle 1949] : Ryle, G.*, The Concept of Mind, *Hutchinson's University Library, London, 1949.*
- *[Sowa 1992]:* Semantic Networks, *Encyclopedia of Artificial Intelligence, second edition, Wiley,1992*
- *[Woods 1975]: Woods, W.A.*, What's in a link: foundations for semantic networks, *in (Bobrow and Collins, eds.) "Representation and understanding", Academic Press, 1975.*

THANK YOU