Taxonomic representation and Reasoning

Description Logics

Contents

Semantic networks

Networks

- associative (psycologic model: at a different level, but they relate to neural networks)
- causal (bayesian networks)
- semantic (Quinlan '68)

- language: graphs with different bindings and annotations
- semantics: subset of FOL (the declarative part) otherwise informal/procedural
- inference: specialized methods for visiting the graph

Inheritance networks

nodes = objetcs or classes

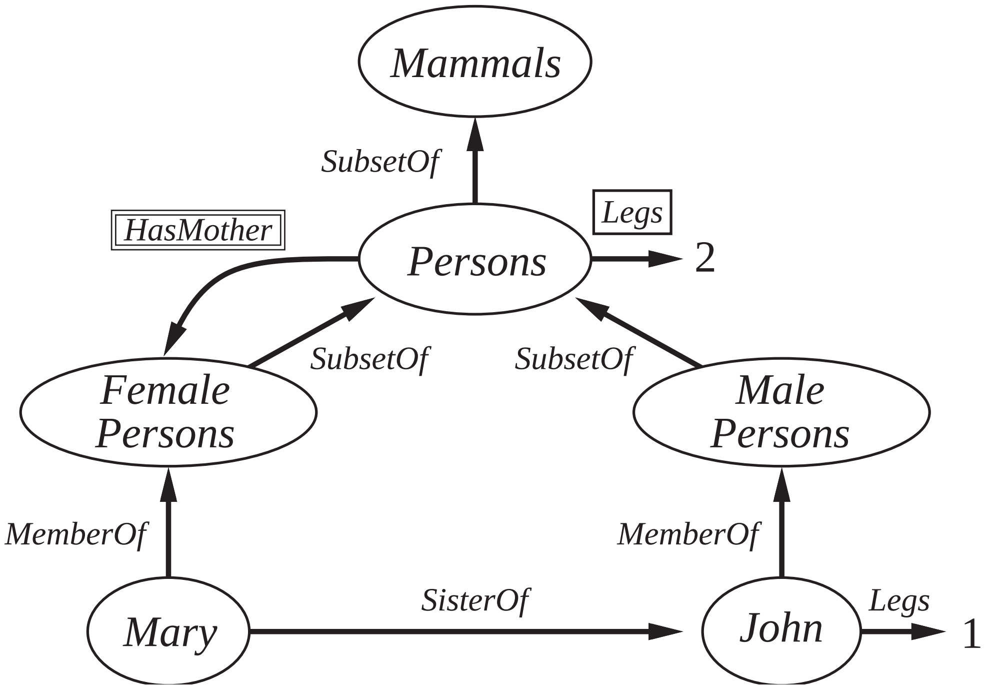
arcs = relations (in particular $is - a$, $istance - of$)

$Cats \stackrel{Subset}{\longrightarrow} Mammals$
$fufi \stackrel{Member}{\longrightarrow} Cats$
$Cats \subseteq Mammals$
$Cat\ is - a\ Mammal$

$\forall x \ \ Cat(x) \rightarrow Mammal(x)$
$Cat(fufi)$

from the network I can infer that:
$fufi$ is a mammal

Semantic Networks



Relationships between objects of different classes.

Frames

Minsky 75: Everything that characterizes an objects belongs to a "frame".

A frame is a prototype of the class elements, but since frames are connected to form networks, a set of frames is very similar to a semantic network, with nodes having a richer structure.

- language: graph/OO specification
- semantics: part in FOL (plus procedural aspects)
- inference: search in ad-hoc representations (graphs)

Systems: KEE, KRSS, ...

Example of frame definition

Frame: Course in KB `University`
  Superclasses:
  Memberslot: ENROLLED
    ValueClass: Stud
    Cardinality.Min: 2
    Cardinality.Max: 30
  Memberslot: TAUGHTBY
    ValueClass: (UNION Grad Prof)
    Cardinality.Min: 1
    Cardinality.Max: 1

Example of instance definition

rc Instance-of: AdvC in KB `University`

Memberslot: TAUGHTBY

ValueClass: **nardi**

Memberslot: ENROLLS

ValueClass: **s1,s2**

Common between semantic networks and frames

- ♢ node properties (own)
- ♢ inherited properties ($is - a$).
- ♢ class descriptions and instances ($instance - of$)
- ♢ default values and defeasible inheritance (*)

Additional features of frames

♢ Complex relationships between frames (slot values can be defined in terms of other frames)

♢ Logical operators in the slot definition

♢ Numerical restrictions

♢ Multiple inheritance

♢ Procedural Attachments ($if - needed$, $if - added$)

Formal reasoning framework → **Description Logics**

Summarizing

Taxonomic representations are based on the idea of organizing knowledge in classes classes of objects:

- objects are characterized by their properties
- inheritance is key to the representation
- complex relations hold between objects

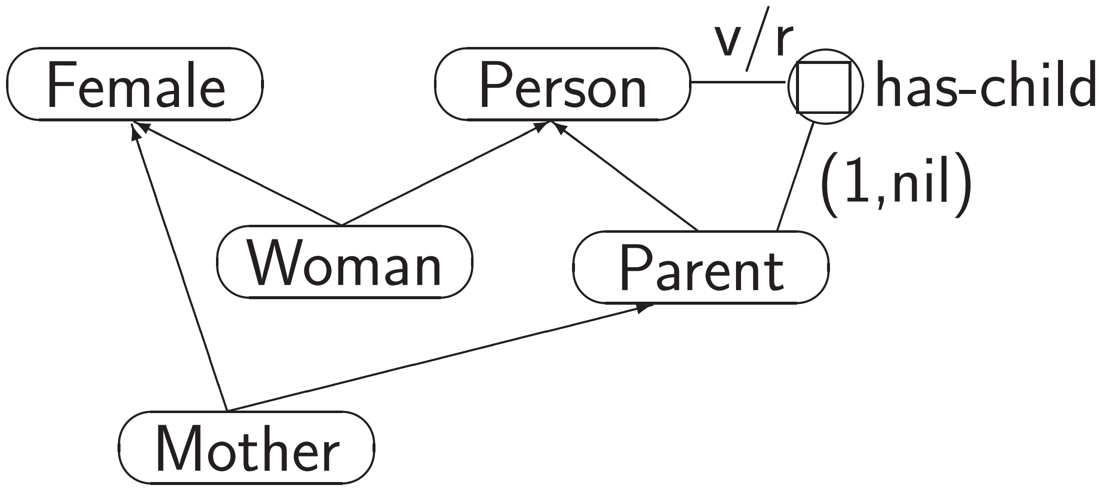
Taxonomic Knowledge Representation

- epistemological adequacy
- computational adequacy

Thesis

Taxonomic knowledge representations are **efficient** both epistemologically and computationally

An example



Relations in a network

- ISA: Mother ISA Female(inheritance)
- Role restrictions: Parent "A parent is a person having at least a child, and all his/her children are persons"

Deduction

Discovery of implicit relations:
let Woman represent the female persons, we have that
Mother ISA Woman

"Simple" inferences are simple to "see" in the network, but we need how the can be exactly computed.

The beginning ...

Brachman and Levesque's thesis

"There exists a fundamental tradeoff between the expressive power of a knowledge representation language and the complexity of reasoning in that language'

hHow to characterize the complexity of deduction?

Provide *formal definition* of the inference problem in order characterize its computational complexity.

Description Logics as a fragment of FOL

Description Logics
(Terminological/Taxonomic/Concept)

The representation in logic of semantic networks and frames is relatively straightforward:

- **classes** or **concepts** are unary predicates
- **relations** between classes or **roles** are binary predicates

Logical reconstruction of networks

- Definition of a knowledge representation language to denote the network structures (abstract syntax, LISP-like syntax, natural language and graphical interfaces)
- Definition of the meaning of the expressions of the language (interpretation structures)
- Definition of the inference problem as logical consequence
- Definition of inference algorithms

Syntax

The basic step of the construction is the choice of two disjoint sets of symbols (alphabets) for *primitive (atomic) concepts*, and *primitive (atomic) roles*.

Terms are then defined using the alphabets by means of *constructors/operators*

Ex. conjunction operator, $C \sqcap D$,
restricts the set to those individual objects belonging to both C and D.

Note: **no variables in the syntax**, the expressions implicitly characterise a set of individual objects (in this case the intersection)

LISP-like syntax (Classic) `(AND Person Female)`

Set constructors

- *intersection* $C \sqcap D$
- *union* $C \sqcup D$
- *complement* $\neg C$

"The persons that are not female" and
"The set of either male or female"

$$\text{Person} \sqcap \neg\text{Female} \qquad \text{and} \qquad \text{Female} \sqcup \text{Male}.$$

Female, Person, are atomic concepts.

Role restrictions

- **universal role restriction**, $\forall R.C$, requires that all those individual objects that are in the relation R belong to C.
  "individuals whose children are all female": ∀hasChild.Female
- **existential role restriction**, $\exists R.C$, requires the existence of an individual in the relation R that belongs to C.
  "individual with a female child": ∃hasChild.Female.

hasChild is an atomic role. The individual object corresponding to the second argument of a role is called **role filler**. the female children are role fillers.

$$\exists \text{hasChild.Person} \sqcap \forall \text{hasChild.Person}.$$

in the role restrictions there is another implicitly quantified variable: $\forall y.R(x, y) \rightarrow C(y)$

Number restrictions

number restrictions: denote sets of individuals with atleast/atmost the specified number of role fillers.

$$(\leqslant 3 \ \mathsf{hasChild})$$

"the individuals with at most three children"

$$(\leqslant 3 \ \mathsf{hasChild}) \sqcap (\geqslant 2 \ \mathsf{hasChild})$$

"the individuals with 2 or 3 children".

Constructors for role expressions

Role intersection: denotes the intersection of roles.

"has-Daughter": hasChild $\sqcap$ hasFemaleRelative

Woman $\sqcap$ ($\leqslant 2$ (hasChild $\sqcap$ hasFemaleRelative))

"women with at most 2 daughters".

More role constructors later on.

Examples of concept expressions

1. $\mathsf{Stud} \sqcap \neg\mathsf{Grad}$
2. $\mathsf{Grad} \sqcup \mathsf{Prof}$
3. $\mathsf{Course} \sqcap \forall\mathsf{ENROLLED}.\mathsf{Grad} \sqcap$
   $(\geq 2\,\mathsf{ENROLLED}) \sqcap (\leq 20\,\mathsf{ENROLLED})$
4. $\mathsf{Course} \sqcap \exists\mathsf{ENROLLED}.\mathsf{Grad} \sqcap$
   $\exists\mathsf{ENROLLED}.\mathsf{UndGrad}$
5. $\mathsf{PowerPlant} \sqcap (\geq 1\,\mathsf{LOC}) \sqcap (\leq 1\,\mathsf{LOC}) \sqcap$
   $\forall\mathsf{LOC}.\mathsf{NewYork} \sqcap \exists\mathsf{FAIL}.\mathsf{Mechanical}$

Correspondence with frame systems

Frame: AdvC in KB `University`
  Superclasses: Course
  Memberslot: ENROLLED
    ValueClass: Grad
    Cardinality.Min: 2
    Cardinality.Max: 20

DL definition of AdvC:

$$\mathsf{Course} \sqcap \forall \mathsf{ENROLLED}.\mathsf{Grad} \sqcap$$
$$(\geq 2\, \mathsf{ENROLLED}) \sqcap (\leq 20\, \mathsf{ENROLLED})$$

Syntax of concept expressions

C, D	$\longrightarrow$	$A \mid$	(primitive concept)
		$\top \mid$	(top)
		$\bot \mid$	(bottom)
		$C \sqcap D \mid$	(intersection)
		$C \sqcup D \mid$	(union)
		$\neg C \mid$	(complement)
		$\forall R.C \mid$	(universal quantif.)
		$\exists R.C \mid$	(existential quantif.)
		$(\geq n\, R) \mid$	(atleast num. restriction)
		$(\leq n\, R)$	(atmost num. restriction)

Syntax of role expressions

$$
\begin{array}{rcll}
R & \longrightarrow & P \mid & \text{(primitive role)} \\
 & & Q \sqcap R \mid & \text{(role conjunction)}
\end{array}
$$

Semantics

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is defined by:

- a set $\Delta^{\mathcal{I}}$ (the *domain* di $\mathcal{I}$)
- a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$) mapping every concept in a subset of $\Delta^{\mathcal{I}}$ and every role in a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The domain can be infinite and the **open world assumption** makes description logics substantially different from data bases.

Semantics of concept expressions

Primitive concepts are subsets of the interpretation domain: $A \subseteq \Delta^{\mathcal{I}}$

$$
\begin{aligned}
\bot^{\mathcal{I}} &= \{\,\} \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\forall R. C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.\, (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R. C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b.\, (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \\
(\geq n\, R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}| \geq n\} \\
(\leq n\, R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}| \leq n\}.
\end{aligned}
$$

Semantics of role expressions

Primitive roles are pairs of elements of the interpretation domain
$R \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

$$(Q \sqcap R)^{\mathcal{I}} = Q^{\mathcal{I}} \cap R^{\mathcal{I}}$$

Semantics of concepts

An interpretation $\mathcal{I}$ is a *model* of a concept C if $C^{\mathcal{I}}$ is non empty.

A concept is *satisfiable* if it has a model, *unsatisfiable* otherwise.

Semantics in FOL

C can be phrased as $\phi_C(x)$ such that for every interpretation $\mathcal{I}$ the set of elements $\Delta^{\mathcal{I}}$ satisfying $\phi_C(x)$ coincides with $C^{\mathcal{I}}$:

$$
\begin{aligned}
\phi_{C \sqcap D}(y) &= \phi_C(y) \wedge \phi_D(y) \\
&\ldots \\
\phi_{\exists R.C}(y) &= \exists x.\, R(y,x) \wedge \phi_C(x) \\
\phi_{\forall R.C}(y) &= \forall x.\, R(y,x) \rightarrow \phi_C(x) \\
\phi_{(\geq n\, R)}(x) &= \exists y_1, \ldots, y_n.\; R(x,y_1) \wedge \cdots \wedge R(x,y_n) \wedge \bigwedge_{i<j} y_i \neq y_j \\
\phi_{(\leq n\, R)}(x) &= \forall y_1, \ldots, y_{n+1}.\; R(x,y_1) \wedge \cdots \wedge R(x,y_{n+1}) \rightarrow \\
&\quad \bigvee_{i<j} y_i = y_j.
\end{aligned}
$$

Taxonomic reasoning

The main inference in DL is **subsumption**:

$$C \sqsubseteq D$$

D (the *subsumer*) is more general than (superset of) C (the *subsumee*).

C is *subsumed* by D if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every interpretation $\mathcal{I}$

Woman $\sqcap$ $\exists$hasChild.Female $\sqsubseteq$ Parent

Other reasoning problems

Concept satisfiability, equivalence and subsumption can be reduced to subsumption.

A complete analysis of the computational complexity of reasoning in different languages shows **sources** of complexity.

Main reasoning techniques:

- structural subsumption;
- tableaux

Knowledge Bases

Two components:

- *intensional* (TBOX)
- *extensional* (ABOX)

$$\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$$

$\mathcal{T}$ is a set of definitions: $C \equiv D$
$\mathcal{A}$ is a set of assertions: $C(a) \;\; P(a, b)$

Terminologies

The TBOX is also called **terminology** and it includes **concept definitions**.

$$\text{Woman} \equiv \text{Person} \sqcap \text{Female}$$

Definitions are interpreted as logical equivalences and thus represent necessary and sufficient conditions, for the defined concept.

- one definition for every concept name
- *acyclic* definitions

TBOX reasoning

Classification, which allows to position a new concept definition in the taxonomy

In practice, the classification of C determines:

- the most specific concepts subsuming C
- the most general concepts that are subsumed by C

A TBOX: family

$$
\begin{aligned}
\mathsf{Woman} &\equiv \mathsf{Person} \sqcap \mathsf{Female} \\
\mathsf{Man} &\equiv \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists\mathsf{hasChild}.\mathsf{Person} \\
\mathsf{Parent} &\equiv \mathsf{Father} \sqcup \mathsf{Mother} \\
\mathsf{GrandMother} &\equiv \mathsf{Mother} \sqcap \exists\mathsf{hasChild}.\mathsf{Parent} \\
\mathsf{MotherWithManyChildren} &\equiv \mathsf{Mother} \sqcap (\geq 3\,\mathsf{hasChild}) \\
\mathsf{MotherWithoutDaughter} &\equiv \mathsf{Mother} \sqcap \forall\mathsf{hasChild}.\neg\mathsf{Woman} \\
\mathsf{Wife} &\equiv \mathsf{Woman} \sqcap \exists\mathsf{hasHusband}.\mathsf{Man}
\end{aligned}
$$

The ABOX

The ABox includes the extensional knowledge:

$\text{Female} \sqcap \text{Person}(\text{ANNA})$

$\text{hasChild}(\text{ANNA}, \text{JACOPO})$

$\neg\text{Female} \sqcap \text{Person}(\text{JACOPO})$

$\text{hasChild}(\text{ANNA}, \text{MICHELA})$

$\text{hasChild}(\text{DANIELE}, \text{JACOPO})$

$\text{hasChild}(\text{DANIELE}, \text{MICHELA})$

Semantics of the ABOX

- individual object intepretation $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
- **UNA**: Unique Name Assumption $a^{\mathcal{I}} \neq b^{\mathcal{I}}$

An assertion $C(a)$ is **satisfiable** in $\mathcal{I}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$

An interpretation $\mathcal{I}$ satisfies an ABOX if it satisfies all the assertions in the ABOX and it is called **model** of the ABOX.

In order to apply the constraints in the TBOX we must expand the concepts in the ABOX.

ABOX reasoning

- **instance checking**, checks whether an individual is an instance of the concept (fundamental)
  $\mathcal{A} \models C(a)$ if every model of $\mathcal{A}$ is also a model of $C(a)$
- **knowledge base consistency**, checks whether the KB admits a model
- **realization**, determines the most specific concept a given individual is an instance of
- **retrieval**, finds all the individuals that are instances of a given concept

ABOX reasoning requires a significant extension of the techniques for TBOX reasoning

A knowledge base (Σ_1): TBOX

$$\mathsf{AdvC} \equiv \mathsf{Course} \sqcap \forall \mathsf{ENROLLED.Grad} \sqcap$$
$$(\geq 2\, \mathsf{ENROLLED}) \sqcap (\leq 20\, \mathsf{ENROLLED})$$
$$\mathsf{BasC} \equiv \mathsf{Course} \sqcap \forall \mathsf{ENROLLED.UndGrad},$$
$$\mathsf{IntC} \equiv \mathsf{Course} \sqcap \exists \mathsf{ENROLLED.Grad} \sqcap$$
$$\exists \mathsf{ENROLLED.UndGrad},$$
$$\exists \mathsf{TEACHES.Course} \sqsubseteq \mathsf{Grad} \sqcup \mathsf{Prof},$$
$$\mathsf{Grad} \equiv \mathsf{Stud} \sqcap \exists \mathsf{DEGREE.Bachelor},$$
$$\mathsf{UndGrad} \equiv \mathsf{Stud} \sqcap \neg \mathsf{Grad}$$

A knowledge base (Σ_1): ABOX

Prof(bob), UndGrad(peter),
Stud(susy), Stud(mary), Bachelor(bs),
Course(cs1), Course(cs2), IntC(ee1),
TEACHES(bob, ee1), TEACHES(john, cs2),
TEACHES(john, cs1), DEGREE(mary, bs),
ENROLLED(cs1, susy), ENROLLED(cs1, mary),
ENROLLED(cs2, susy), ENROLLED(cs2, peter),
ENROLLED(ee1, peter)

Querying the knowledge base

The queries to a KB Σ are expressed as:

$$C(a)$$

- YES $C(a)$ is true in every model of Σ
- NO $C(a)$ is false in every model of Σ
- $UNKNOWN$ otherwise.

Examples

1 $\Sigma_1 \models \exists$ENROLLED.Grad(ee1) ?
Answer: YES.

2 $\Sigma_1 \models$ Grad $\sqcup$ Prof(john) ?
Answer: YES.

3 $\Sigma_1 \models \forall$TEACHES.IntC(bob) ?
Answer: $UNKNOWN$.

4 $\Sigma_1 \models \exists$TEACHES.IntC(john) ?
Answer: YES.

Open World semantics

Ontology Web Language

OWL is the standard language for representing ontologies.

- ♢ `Protege` is a standard tool to define ontologies

- ♢ Different reasoners can be used within Protege.

Major impact of DL and ontologies: support for data integration