Conceptual modelling, ontology design, and semantic interoperability

Professional master on technologies for e-government
Lecture 3

Claudio Masolo
Laboratory for Applied Ontology, ISTC-CNR
`masolo@loa-cnr.it`

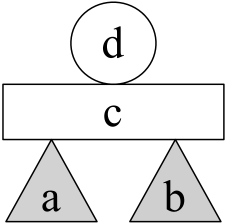
Trento, 08.10.2007

Outline

- ▷ Introduction to logic-based knowledge representation
- ▷ Knowledge representation and conceptual modeling

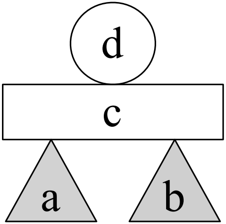
A simple example

Let us consider the situation illustrated in the figure:



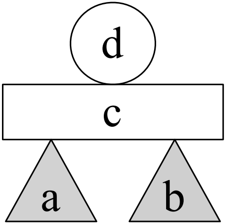
How this situation can be modeled?

Informal analysis



- ▷ there are different objects
- ▷ the objects have different shapes
- ▷ the objects have different colors
- ▷ the objects are 'spatially' related

Informal model #1



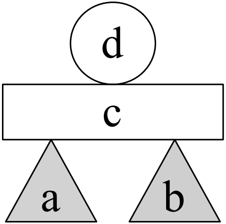
- 4 *individuals*: a, b, c, d
- 5 *properties*: being white, being gray, being triangular, being rectangular, being round
- 1 *relation*: supports

Using a *natural language* sentence, we can describe the previous situation as follows:

> "the gray triangle a and the white triangle b support the white rectangle c that supports the gray round d".

Note: *supports* is not a pure spatial relation.

Set theoretical model #1 ($\mathcal{M}_1$)

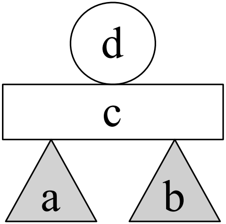


- $D_1 = \{a, b, c, d\}$
- $White_1 = \{c, d\}$
- $Gray_1 = \{a, b\}$
- $Triang_1 = \{a, b\}$
- $Rect_1 = \{c\}$
- $Round_1 = \{d\}$
- $Supp_1 = \{\langle a, c\rangle, \langle b, c\rangle, \langle c, d\rangle\}$

We obtain the following structure:

$\mathcal{M}_1 = \langle D_1, White_1, Gray_1, Triang_1, Rect_1, Round_1, Supp_1\rangle$

Informal analysis #2



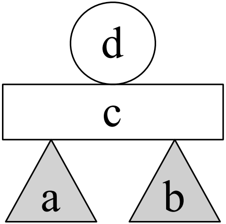
- 9 *individuals*: a, b, c, d, white, gray, triangular, rectangular, round
- 3 *properties*: being an object, being a color, being a shape
- 2 *relation*: supports, has shape, has color

Using a *natural language* sentence:

> "the object a has a gray color and a triangular shape and it support the object c that has a white color and a rectangular shape...".

(?) is which sense this NL sentence differs from the previous one?

Set theoretical model #2 ($\mathcal{M}_2$)



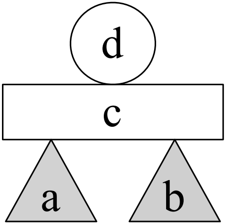
- $D_2 = \{a, b, c, d, wh, gr, tr, re, ro\}$
- $Object_2 = \{a, b, c, d\}$
- $Color_2 = \{wh, gr\}$
- $Shape_2 = \{tr, re, ro\}$
- $hasColor_2 = \{\langle a, gr\rangle, \langle b, gr\rangle, \langle c, wh\rangle, \langle d, wh\rangle\}$
- $hasShape_2 = \{\langle a, tr\rangle, \langle b, tr\rangle, \langle c, re\rangle, \langle d, ro\rangle\}$
- $Supp_2 = \{\langle a, c\rangle, \langle b, c\rangle, \langle c, d\rangle\}$

We obtain the following structure:

$\mathcal{M}_2 = \langle D_2, Object_2, Color_2, Shape_2, hasColor_2, hasShape_2, Supp_2\rangle$

Relational data model

Relational models organize data in tables that represent relations. In this framework, we can represent our example with 2 tables:



Object

ID	*Color*	*Shape*
a	*gr*	*tr*
b	*gr*	*tr*
c	*wh*	*re*
d	*wh*	*ro*

Supports

Supporter	*Supportee*
a	*c*
b	*c*
c	*d*

Set theoretical view on relational model ($\mathcal{M}_r$)

Object

ID	*Color*	*Shape*
a	gr	tr
b	gr	tr
c	wh	re
d	wh	ro

Supports

Supporter	*Supportee*
a	c
b	c
c	d

- $D_r = \{a, b, c, d, wh, gr, tr, re, ro\}$
- $ID_r = \{a, b, c, d\}$
- $Color_r = \{wh, gr\}$
- $Shape_r = \{tr, re, ro\}$
- $Supporter_r = \{a, b, c\}$
- $Supportee_r = \{c, d\}$
- $Object_r = \{\langle a, gr, tr\rangle, \langle b, gr, tr\rangle, \langle c, wh, re\rangle, \langle d, wh, ro\rangle\}$
- $Supp_r = \{\langle a, c\rangle, \langle b, c\rangle, \langle c, d\rangle\}$

$$\mathcal{M}_r = \langle D_r, ID_r, Color_r, Shape_r, Supporter_r, Supportee_r, Object_r, Supp_r\rangle$$

Modeling using FOL

You have already seen what first order logic is. Is it possible to model the previous situation using that formalism?

FOL is based on a clear distinction between:

- ▷ individual constants and individual variables
- ▷ predicates
  - we will call *property* a unary predicate, and
  - we will call *relation* a n-ary predicate with $n > 1$

Constants and predicates will be respectively noted using lower and upper case sans serif characters to clearly distinguish them from elements and sets that has been noted in *italic*

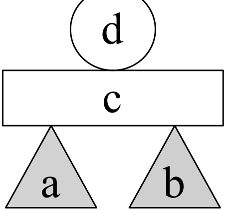
Instance-of

- ▷ The *instance-of* relation is one of the basic relations considered in conceptual modeling
- ▷ It links an *individual* (tuple) with a *property* (relation)
- ▷ In set theory it is represented by *membership*
  e.g. $b \in White$, $\langle a, gr \rangle \in hasColor$
- ▷ In FOL it is represented by *predication*
  e.g. $\mathsf{White}(\mathsf{b})$, $\mathsf{hasColor}(\mathsf{a}, \mathsf{gr})$

⇒ Classical FOL semantics links predication to membership:
- constants (variables) are mapped to elements in the domain
- predicates and relations are respectively mapped to sets of elements and sets of tuples

FOL allows one level, membership infinite levels of instantiation.

FOL theory that corresponds to $\mathcal{M}_1$

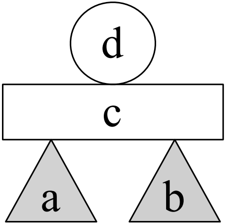


- 4 constants: a,b,c,d
- 5 properties: White_1, Gray_1, Triang_1, Rect_1, Round_1
- 1 relation: Supp_1

By means of these primitives we can write the FOL theory that 'corresponds' to $\mathcal{M}_1$

$$\begin{aligned}\mathcal{T}h_1\text{:}\ & \mathsf{Gray}_1(\mathsf{a}) \wedge \mathsf{Gray}_1(\mathsf{b}) \wedge \mathsf{White}_1(\mathsf{c}) \wedge \mathsf{White}_1(\mathsf{d}) \\ & \mathsf{Triang}_1(\mathsf{a}) \wedge \mathsf{Triang}_1(\mathsf{b}) \wedge \mathsf{Rect}_1(\mathsf{c}) \wedge \mathsf{Round}_1(\mathsf{d}) \\ & \mathsf{Supp}_1(\mathsf{a}, \mathsf{c}) \wedge \mathsf{Supp}_1(\mathsf{b}, \mathsf{c}) \wedge \mathsf{Supp}_1(\mathsf{c}, \mathsf{d})\end{aligned}$$

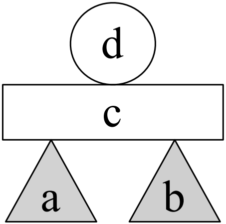
FOL theory that corresponds to $\mathcal{M}_2$



- 9 constants: a, b, c, d, wh, gr, tr, re, ro
- 3 properties: Object_2, Color_2, Shape_2
- 3 relations: $\mathsf{hasColor}_2$,$\mathsf{hasShape}_2$, Supp_2

$\mathcal{T}h_2$:
$\mathsf{Object}_2(\mathsf{a}) \wedge \mathsf{Object}_2(\mathsf{b}) \wedge \mathsf{Object}_2(\mathsf{c}) \wedge \mathsf{Object}_2(\mathsf{d})$
$\mathsf{Color}_2(\mathsf{wh}) \wedge \mathsf{Color}_2(\mathsf{gr}) \wedge \mathsf{Shape}_2(\mathsf{tr}) \wedge \mathsf{Shape}_2(\mathsf{re}) \wedge \mathsf{Shape}_2(\mathsf{ro})$
$\mathsf{hasColor}_2(\mathsf{a},\mathsf{gr}) \wedge \mathsf{hasColor}_2(\mathsf{b},\mathsf{gr}) \wedge \mathsf{hasColor}_2(\mathsf{c},\mathsf{wh}) \wedge \mathsf{hasColor}_2(\mathsf{d},\mathsf{wh})$
$\mathsf{hasShape}_2(\mathsf{a},\mathsf{tr}) \wedge \mathsf{hasShape}_2(\mathsf{b},\mathsf{tr}) \wedge \mathsf{hasShape}_2(\mathsf{c},\mathsf{re}) \wedge \mathsf{hasShape}_2(\mathsf{d},\mathsf{ro})$
$\mathsf{Supp}_2(\mathsf{a}, \mathsf{c}) \wedge \mathsf{Supp}_2(\mathsf{b}, \mathsf{c}) \wedge \mathsf{Supp}_2(\mathsf{c}, \mathsf{d})$

FOL theory that corresponds to $\mathcal{M}_r$



- 9 constants: a, b, c, d, wh, gr, tr, re, ro
- 5 properties: ID_r, Color_r, Shape_r, $\mathsf{Supporter}_r$, $\mathsf{Supportee}_r$
- 2 relations: Object_r, Supp_r

$\mathcal{T}h_r$: $\mathsf{ID}_r(\mathsf{a}) \wedge \mathsf{ID}_r(\mathsf{b}) \wedge \mathsf{ID}_r(\mathsf{c}) \wedge \mathsf{ID}_r(\mathsf{d})$
$\mathsf{Color}_r(\mathsf{wh}) \wedge \mathsf{Color}_r(\mathsf{gr}) \wedge \mathsf{Shape}_r(\mathsf{tr}) \wedge \mathsf{Shape}_r(\mathsf{re}) \wedge \mathsf{Shape}_r(\mathsf{ro})$
$\mathsf{Supporter}_r(\mathsf{a}) \wedge \mathsf{Supporter}_r(\mathsf{b}) \wedge \mathsf{Supporter}_r(\mathsf{c})$
$\mathsf{Supportee}_r(\mathsf{c}) \wedge \mathsf{Supportee}_r(\mathsf{d})$
$\mathsf{Object}_r(\mathsf{a,gr,tr}) \wedge \mathsf{Object}_r(\mathsf{b,gr,tr}) \wedge \mathsf{Object}_r(\mathsf{c,wh,re}) \wedge$
$\mathsf{Object}_r(\mathsf{d,wh,ro}) \wedge \mathsf{Supp}_r(\mathsf{a}, \mathsf{c}) \wedge \mathsf{Supp}_r(\mathsf{b}, \mathsf{c}) \wedge \mathsf{Supp}_r(\mathsf{c}, \mathsf{d})$

Comparing the models (and the theories)

- $\mathcal{M}_2$ and $\mathcal{M}_r$ both accept individuals like colors and shapes ($D_r = D_2$) while $\mathcal{M}_1$ considers only the objects of $\mathcal{M}_2$ and the *ID*s of $\mathcal{M}_r$ ($D_1 = Object_2 = ID_r$)
- 'being a color' and 'being a shape' are represented both in $\mathcal{M}_2$ ($Color_2$ and $Shape_2$) and in $\mathcal{M}_r$ ($Color_r$ and $Shape_r$) while in $\mathcal{M}_1$ we represent 'being gray', 'being white', 'being triangular', etc. that apply directly on the entities in the domain
- 'supports' is represented in the same way in all the models
- In $\mathcal{M}_2$ we have $hasColor_2$ and $hasShape_2$ to link objects to their *attributes*, while in $\mathcal{M}_r$ the relation $Object_2$ links one *ID* to all its attributes (it is a ternary relation in this case)

(?) Why $Object_r$ is a relation (with argument ID_r) and not a class?

Formal comparison

▷ Is it possible to introduce a more formal comparison, for example establishing mappings between the previous theories?

▷ Using the FOL we can follow at least two strategies:

- we consider a global theory that is the union of the theories we want to map and the mappings are expressed in that theory
- we consider mappings as external links among theories (we need a suitable language to express these mappings)

▷ We will follow the first strategy assuming (i) that all the theories have disjoint predicates, and (ii) that, for constant, the principle "same name, same constant" holds (instead of having disjoint names for constants and mapping axioms like $\mathsf{a}_1 = \mathsf{a}_2$)

Formal mappings (1/2)

Let us start with a theory $\mathcal{T}h$ that is the union of $\mathcal{T}h_1$, $\mathcal{T}h_2$, $\mathcal{T}h_r$

Can you see general/intuitive mappings between predicates?

- ▷ "colors are colors, shapes are shapes, and IDs are objects"
  $\forall x(\mathsf{Color}_2(x) \leftrightarrow \mathsf{Color}_r(x))$
  $\forall x(\mathsf{Shape}_2(x) \leftrightarrow \mathsf{Shape}_r(x))$
  $\forall x(\mathsf{Object}_2(x) \leftrightarrow \mathsf{ID}_r(x))$
- ▷ "supports is supports"
  $\forall x, y(\mathsf{Supp}_1(x, y) \leftrightarrow \mathsf{Supp}_2(x, y) \leftrightarrow \mathsf{Supp}_r(x, y))$
- ▷ "$\mathcal{T}h_1$ accepts only objects/IDs"
  $\forall x(\mathsf{White}_1(x) \rightarrow \mathsf{Object}_2(x))$
  and the same argument restrictions for all predicates in $\mathcal{T}h_1$

Formal mappings (2/2)

More complex mappings:

- ▷ $\forall x, y, z(\mathsf{Object}_r(x, y, z) \leftrightarrow (\mathsf{hasColor}_2(x, y) \wedge \mathsf{hasShape}_2(x, z)))$
- ▷ $\forall x, y(\mathsf{hasColor}_2(x, y) \leftrightarrow \exists z(\mathsf{Object}_r(x, y, z))$
- ▷ $\forall x(\mathsf{White}_1(x) \leftrightarrow (\mathsf{hasColor}_2(x, \mathsf{wh}))$
- ▷ $\forall x(\mathsf{White}_1(x) \leftrightarrow \exists z(\mathsf{Object}_r(x, \mathsf{wh}, z))$

- The last two mappings makes explicit the roles of the color and shape constants in theories $\mathcal{T}h_1$ and $\mathcal{T}h_r$

(?) Do you agree on the link $\forall x(\mathsf{Object}_2(x) \leftrightarrow \mathsf{ID}_r(x))$?
It is formally correct, but does it match your intuitions?

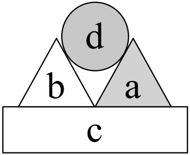
*Set-theoretical counterpart of FOL mappings

Is it possible to find a set theoretical counter-part of the FOL mappings?

- $\forall x(\mathsf{Color}_2(x) \leftrightarrow \mathsf{Color}_r(x))$
  $Color_2 = Color_r$
- $\forall x(\mathsf{White}_1(x) \rightarrow \mathsf{Object}_2(x))$
  $White_1 \subseteq Figures_2$
- $\forall x(\mathsf{White}_1(x) \leftrightarrow (\mathsf{hasColor}_2(x, \mathsf{wh}))$
  $White_1 = \{x | \langle x, wh\rangle \in hasColor_2\}$
- $\forall x, y(\mathsf{hasColor}_2(x, y) \leftrightarrow \exists z(\mathsf{Object}_r(x, y, z))$
  $hasColor_2 = \{\langle x, y\rangle | \langle x, y, z\rangle \in Object_r\}$

Extension vs. intension **(1/3)**

Let us now suppose that the situation changes:



▷ The previous models an theories do not represent anymore this new situation and new theories and models are needed, but do these two different situations have something in common?

(?) Is it possible to reuse something from FOL theories or from set-theoretical models?

- Sets are builded on instances while FOL predicates apply to constants and some mappings does not involve constants.

Extension vs. intension (2/3)

- ▷ The idea is to consider more abstract theories/models that represent *conceptualizations* of a domain independently from a specific *arrangement* of that domain.
  - In other terms, we want to represent in a formal way *knowledge* instead of just *data* about a domain.
- ▷ In lecture 1, the distinction between *intensional* vs. *extensional* relations has been characterized in terms of Montague's *possible worlds* semantics.
  - In this framework a relation is a function that to each possible world associates the extension of the relation, i.e. a relation is a set of set of tuples.

Extension vs. intension (3/3)

- ▷ From a modeling (and maybe theoretical) perspective, the Montague's approach is really demanding because it requires knowledge about all the possibles arrangements of the objects in the world.
- ▷ How is it then possible to capture the *invariances* we are interested in without assuming a sort of *omniscience*?
- Logic allows us this kind of abstraction. The idea is to represent *necessary* knowledge by means of axioms. These axioms are by definition valid in all the *models* of the theory, and models can represent different arrangements of the objects in the world, but we do not need to specify all these models from the beginning, we need just axioms.

*Limits of logic

(!) Problematic for representing data and dynamics of data.

- *Description Logics* introduce two levels: T-Box is about general knowledge while A-Box is about data. But the dynamics of data is still problematic.
- The explicit introduction of time partially solves the problem of dynamics but complicates a lot the theory.

(!) Problematic for representing *possible* knowledge and *typicality*.

- Modal logics and default / non-monotonic logics try to address these problems.

But let us start with the analysis of some abstractions introduced in the database community for which data are important.

Relations in relational data models (1/2)

▷ What is the difference between the following two tables?

$\mathbf{Object_1}$

ID	*Color*	*Shape*
a	gr	tr
b	gr	tr
c	wh	re
d	wh	ro

$\mathbf{Object_2}$

Shape	*ID*	*Color*
tr	a	gr
tr	b	gr
re	c	wh
ro	d	wh

▷ Following the set theoretical reading, the relations represented in the two tables are different (actually they are disjoint):

$$\mathbf{Object_1} = \{\langle a, gr, tr\rangle, \langle b, gr, tr\rangle, \langle c, wh, re\rangle, \langle d, wh, ro\rangle\}$$
$$\mathbf{Object_2} = \{\langle tr, a, gr\rangle, \langle tr, b, gr\rangle, \langle re, c, wh\rangle, \langle ro, d, wh\rangle\}$$
$$\mathbf{Object_1} \cap \mathbf{Object_2} = \emptyset$$

Relations in relational data models (2/2)

▷ Relational data modeling introduces tables to abstract from the mathematical notion of relation (a set of tuples)

a. *Relational schemes*, restrictions on the arguments of relations (*attributes* in data modeling), are more abstract than tuples

$\mathbf{Object_1}(ID, Color, Shape)$; $\mathbf{Object_2}(Shape, ID, Color)$

b. and the substitution of tuples with *mappings* from attributes to *values* allows for abstracting from the order of attributes

$\Rightarrow$ $\mathbf{Object_1}$ and $\mathbf{Object_2}$ become the same (abstract) relation with *scheme* $\mathbf{Object}(ID, Color, Shape)$. In our example:

$\mathbf{Object} = \{\mu_1, \mu_2, \mu_3, \mu_4\}$ where
$\mu_1(ID) = a, \mu_1(Color) = gr, \mu_1(Shape) = tr$
$\mu_2(ID) = b, \mu_2(Color) = gr, \mu_2(Shape) = tr$ ...

*Relations schemes and FOL

▷ Relational schemes represent quite general knowledge: they characterize the kind of entities that can be in a specific relation, and for this reason, intuitively they are independent from specific states of the world.

(?) How a relational scheme can be represented in FOL?

- Let us consider **Object**($ID, Color, Shape$); by means of one relation and three properties we can write the following axiom (called *argument restriction* of Object):

  $\forall x, y, z(\mathsf{Object}(x, y, z) \rightarrow \mathsf{ID}(x) \wedge \mathsf{Color}(y) \wedge \mathsf{Shape}(z))$

- Differently from the relational scheme, this formulation is not independent from the order of arguments; it is possible to simulate this independence but that is not central for our goal.

Keys

▷ Let us now consider the attribute *ID*, introduced in our relational model.

▷ The chosen name underline a particular role of this attribute, a *key* in relational data models:

A set S of attributes of an (abstract) relation R is a *key* iff:

1. R cannot have two tuples that agree in all the attributes in S
2. no proper subset of S has property (1)

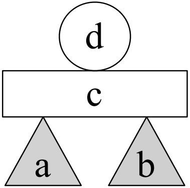
- *ID* is a key of the relation **Object**, i.e. it is **not possible** to have two rows with the same value in the column *ID*

Keys (2/4)

- ▷ What is the intuition behind that?
- • The intuition regards the fact that objects can have just a color and a shape, i.e. one object cannot have two shapes or colors
- ▷ We defined keys for relations but, more correctly, they are defined for *relational schemes* therefore keys do not depend on which tuples are present in the relation but only on the arguments, i.e. the notion of key is an abstract/conceptual one
- ▷ Keys encapsulate a *modal* notion: it is not possible that, there are no possible states of the world in which, two different tuples agree in all the attributes in the key

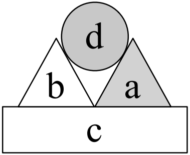
Keys

What key for **Supports**?



Supports

Supporter	*Supportee*
a	c
b	c
c	d



Supports

Supporter	*Supportee*
c	b
c	a
b	d
a	d

Entity-Relationship (ER) Diagrams

Before answering the previous questions, let us consider a graphical formalism designed to summarize in an intuitive way the *relational schemes*. In these diagrams:

- ▷ rectangles represent *entity sets*;
- ▷ circles represent *attributes* (linked to their entities sets by edges);
  - attributes part of the key (for their entity set) are underlined;
  - if an entity set has just an attribute the attribute is omitted;
- ▷ diamonds represent *relationships*.

But, while ER diagrams assume the distinction between *entity sets* and *relationships*, in relational data modeling we have only relational schemes! How these two languages can match?

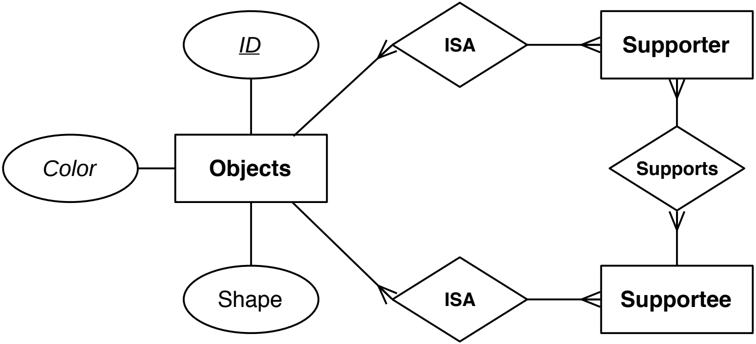
Coming back to our example

- ▷ Let us consider our two relational schemes:
  Object(ID, $Color$, $Shape$)
  Supports($Supporter$, $Supportee$)
- ▷ in order to build the ER diagrams for these schemes we need to decide if **Object** and **Supports** are entity sets with multiple attributes or relationships,

  or, in a similar way
- ▷ if ID, $Color$, $Shape$, $Supporter$, $Supportee$ are attributes or entity sets.

- As already observed, intuitively **Object** is much closer to an entity set than to a relationship, while the opposite is true for **Supports**. Therefore we can accept the following ER diagram.

ER diagram of our example



Remarks on the ER diagram: ISA relation

- ▷ There is one **ISA** relation between **Supporter** and **Object** and one between **Supportee** and **Object**
- ▷ The **ISA** relation links two entity sets:
- from a set theoretical and logical perspective the **ISA** relation is easy; it corresponds to *inclusion* ($\subseteq$) that 'corresponds' to the FOL *implication* ($\rightarrow$);
- in a relational data model the **ISA** relation is more difficult because we need to take into account how the sub-entity sets (**Supportee** and **Supporter**) inherit the attributes of the super-entity set (**Object**)

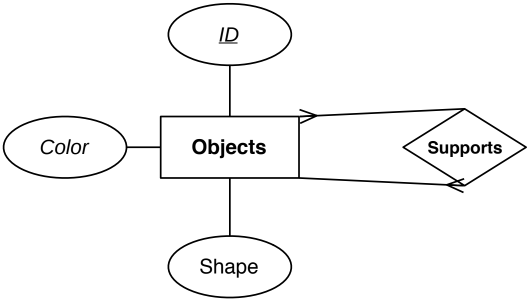
  In lecture 9 we will see that this is a quite interesting/open problem

*Remarks on the ER diagram: Roles

If **Supporter** and **Supportee** can be conceptually defined as:

$$\forall x(\mathsf{Supporter}_r(x) \leftrightarrow \exists y(\mathsf{Supp}_r(x, y)))$$
$$\forall x(\mathsf{Supportee}_r(x) \leftrightarrow \exists y(\mathsf{Supp}_r(y, x)))$$

why do we need them in the model?



Remarks on the ER diag.: Attributes/Keys

▷ The distinction between Attributes and Keys refers to the intuitive one between *partial* and *complete identifiers*.

- *Color* and *Shape* only partially identify an object, i.e. it is possible to have different objects with the same color and shape;
- *ID* completely identifies an object because it is not possible to have two different objects with the same ID.

The ER diagram in FOL (1/3)

Let us try to change the perspective and look for a possible encoding of the first ER diagram in FOL.

▷ In particular, let us use $\mathcal{T}h_2$ to capture previous ER diagrams.

- Attributes and entity sets correspond to unary predicates, relations to n-ary predicates, links between attributes and entity sets to binary relations.

- We start with basic *argument restrictions* (we do not write universal quantifier that applies to the whole formula):

  $\mathsf{hasID}(x, y) \rightarrow \mathsf{Object}(x) \wedge \mathsf{ID}(y)$
  $\mathsf{hasColor}(x, y) \rightarrow \mathsf{Object}(x) \wedge \mathsf{Color}(y)$
  $\mathsf{hasShape}(x, y) \rightarrow \mathsf{Object}(x) \wedge \mathsf{Shape}(y)$
  $\mathsf{Supports}(x, y) \rightarrow \mathsf{Object}(x) \wedge \mathsf{Object}(y)$

The ER diagram in FOL **(2/3)**

Now let us try to capture the notion of attribute and key

- Attribute:
  $\mathsf{Object}(x) \rightarrow \exists y(\mathsf{hasID}(x, y))$
  $\mathsf{Object}(x) \rightarrow \exists y(\mathsf{hasColor}(x, y))$
  $\mathsf{Object}(x) \rightarrow \exists y(\mathsf{hasShape}(x, y))$

- Key:
  $\mathsf{hasID}(x, y) \wedge \mathsf{hasID}(x', y) \rightarrow x = x'$

▷ On this topic there is a huge discussion.

- Are keys necessary in FOL theories?

- Or are objects, intended as entities and not as tuples of attributes, enough?

The ER diagram in FOL (3/3)

At the end let us try to capture the roles:

- Roles:
  $\mathsf{Supporter}(x) \leftrightarrow \exists y(\mathsf{Supports}(x, y))$
  $\mathsf{Supportee}(y) \leftrightarrow \exists x(\mathsf{Supports}(y, x))$
- we can easily infer:
  $\mathsf{Supporter}(x) \rightarrow \mathsf{Object}(x)$
  $\mathsf{Supportee}(x) \rightarrow \mathsf{Object}(x)$

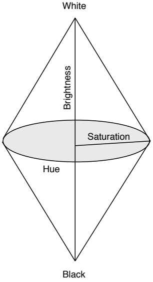
  and therefore, following the previous equivalencies, the attributes and keys of Object apply also to Supporter and Supportee

Note: Accepting the previous equivalencies, we can avoid Supporter and Supporter from the set of primitives of the theory considering them just as syntactic sugar.

Remarks on Attributes (1/3)

▷ The last FOL theory presupposes the existence of entities like colors, shapes, and names/IDs. Is this reasonable/founded?

- There are cognitive theories of *concepts* that assume spaces of colors or shapes, as for example the color splinter



Remarks on Attributes (2/3)

- Controversial axioms for attributes:
  $\mathsf{ID}(x) \rightarrow \exists y(\mathsf{hasID}(y, x))$
  $\mathsf{Color}(x) \rightarrow \exists y(\mathsf{hasColor}(y, x))$
  $\mathsf{Shape}(x) \rightarrow \exists y(\mathsf{hasShape}(y, x))$

  i.e. it is true that for each value of an attribute there is an object with this attribute value?

- In some cases the attributes are abstract construction (like cost) and therefore it is possible to build an infinite number of values; does this means that there is an infinite number of objects?

Remarks on Attributes (3/3)

- Let us consider the following mapping introduced before: $\mathsf{White}_1(x) \leftrightarrow \mathsf{hasColor}_2(x, \mathsf{wh})$
- This mapping makes explicit that if we want to represent colors with predicates
  1. we need one predicate for each color, therefore if the colors are infinite we need infinite predicates.
  2. we need second order to describe relations between colors (e.g. this color is similar to this other color) that are required, for example, to capture the structure of the color splinter.

Strengthening the theory (1/3)

- ▷ We saw that, w.r.t. conceptual distinctions, ER diagrams are more 'transparent' than relational data model even though all ER diagrams can be translated in relational data models.
- ▷ FOL is a substantial gain in expressive power but right now we have not really used this additional expressive power.
- ▷ E.g. for Supports we just considered the argument restriction

  $$\mathsf{Supports}(x, y) \rightarrow \mathsf{Object}(x) \wedge \mathsf{Object}(y)$$

  that characterize very weakly the concept of 'supporting': very different relations can be defined on obejcts.

(?) How can we better characterize Supports using FOL?

Strengthening the theory (2/3)

Let us start answering to some general questions:

$\triangleright$ Is it possible that one object supports itself?

- If **not** we can add the following axiom:
  $\neg\mathsf{Supports}(x, x)$ (*irreflexivity*)

$\triangleright$ Is it possible that two objects "support each other"?

- If **not** we can add the following axiom:
  $\mathsf{Supports}(x, y) \rightarrow \neg\mathsf{Supports}(y, x)$ (*asymmetry*)

$\triangleright$ If one object supports a second object that supports a third object then does the first object support the third one?

- If **yes** we can add the following axiom:
  $\mathsf{Supports}(x, y) \wedge \mathsf{Supports}(y, z) \rightarrow \mathsf{Supports}(x, z)$ (*transitivity*)

Strengthening the theory (3/3)

Let us continue looking for possible links with other properties and relations in the theory, for example we can introduce the following general constraints.

- Round objects cannot support other objects:
  $\mathsf{Round}(x) \rightarrow \neg\exists y(\mathsf{Supports}(x, y))$

  where Round can be syntactically defined as:
  $\mathsf{Round}(x) \triangleq \mathsf{hasShape}(x, \mathsf{ro})$

- Triangular objects can support other objects only together with at least another object:
  $\mathsf{Triang}(x) \wedge \mathsf{Supports}(x, y) \rightarrow \exists z(\mathsf{Supports}(z, y))$

  where Triang can be syntactically defined as:
  $\mathsf{Triang}(x) \triangleq \mathsf{hasShape}(x, \mathsf{tr})$

Reasoning in FOL

In FOL we can use the deductive system in order to prove some intended results but also to check for the equivalence or independence of axioms. For example, it is possible to prove that:

- ▷ $\mathsf{Supports}(x, y) \rightarrow \neg\mathsf{Round}(x)$ is an alternative but equivalent formulation of $\mathsf{Round}(x) \rightarrow \neg\exists y(\mathsf{Supports}(x, y))$
- ▷ or that from the *irreflexivity* of Supports its *asymmetry* follows.

- Sketched proof.
  By contradiction let us assume $\mathsf{Supports}(x, x)$.
  Then, for the reflexivity of implication, we have:
  $\mathsf{Supports}(x, x) \rightarrow \mathsf{Supports}(x, x)$.
  Contradiction.

Knowledge engineering

We have seen that:

- ▷ there are alternative languages with different expressive powers to represent knowledge
- ▷ it is possible to chose different primitives and to formally characterize these primitives in different ways
- ▷ it is possible to introduce axioms at different levels of generality
- ▷ different existential axioms represent different commitments on the entities assumed in the domain of quantification

Knowledge engineering is a discipline that develop methodologies and technics that help in the domain analysis, primitives' selection and formalization, etc.

*Evaluating the models/theories

- ▷ Which is the best model? Does that question make sense?
- ▷ Is ‘best’ a relative notion?
- ▷ How ca we talk of *adequacy* of models with respect some goals?
- ▷ How the *quality* of a model can be evaluated?
- ▷ A lot of these questions are still open research question.... but... here we are at least interested in:
  –are ontological *commitment* and ontological *parsimony* good evaluation parameter?
  – is common-sense adequacy important from models, and is it easy to understand this kind of adequacy?