



# LoD

# The logic of Descriptions

## (HP2T)

# LoD – The Logic of descriptions

- **Intuition**
- Definition
- Domain
- Language – intuition
- The language of etype percepts
- The language of composite etype percepts
- The language of descriptions
- Theories – TBoxes
- Entailment
- Ask
- Key notions

# LoD – Why a logic of Descriptions?

- The **Logic of Descriptions** (LoD) allows us to represent etypes and dtypes and the properties which correlate them. It allows to reason about the etypes and dtypes which are populated in a LoE EG.
- It allows to construct a new **complex etype** based on the **properties** of the elements of an existing etype. For instance a foreigner may be described as a person who does not speak the local language, or a pet as an animal which lives in the house
- It allows to construct a new **complex composite etype** from simpler etypes, starting from the basic ones, as defined in LoE. For instance, it allows to define a parent as the union of father and mother.
- It allows to **constraint** the extension of etypes (via a **description**). For instance, it allows to say that a woman and a man are disjoint etypes, and that they are more specific than the etype person.
- It allows to **define** a new type by given a name (via a **definition**) to a previously constructed complex etype
- It allows to **reason** about how the meaning of complex as a function of the meaning of basic etypes.
- It allows to **extend** LoE graphs with the information about their etypes and to reason about these extended LoE Graphs (as part of the **LoDE** logic)

# LoD – Highlights

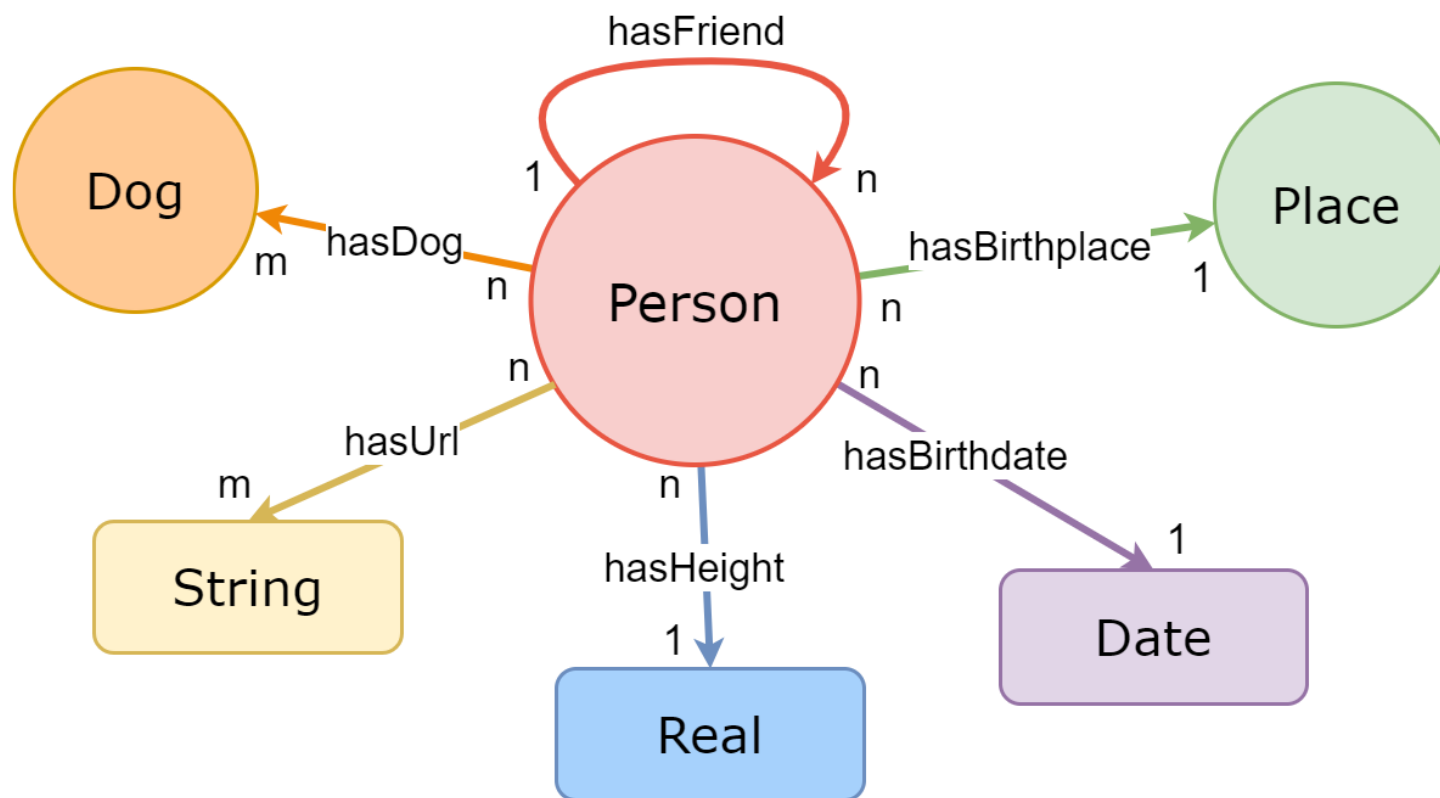
- LoD is the logic encoding the general knowledge about entities (how they are defined and how they can be described).
- LoD represents etypes and the properties and relations of etypes. It does not represent specific entities. Knowledge is at the type level, not at the entity level
- LoD is a world logic with a graph linguistic/ analogic representation
- Any LoE EG is built with reference to a LoD ETG. ETGs can be thought of specifying the schema of Egs.
- LoD is conceptually similar to the Terminology Box (Tbox) of Description Logics (DL). The move is from DBs to KGs.

# LoD – which percepts

In LoD we have the following ETG elements:

- An **entity type** (etype) is a class of entities (corresponding to the etype to which an entity belongs in a LoE EG).
- A **datatype** (dtype) is a class of (data) values (corresponding to the dtype to which a value belongs in a LoE EG).
- An **Object Property** describes a relation between two etypes (not between two entities, as in LoE).
- A **Data Property**, also called **Attribute**, describes a characteristic of an etype (not of an entity as in LoE).

# An example of LoD ETG



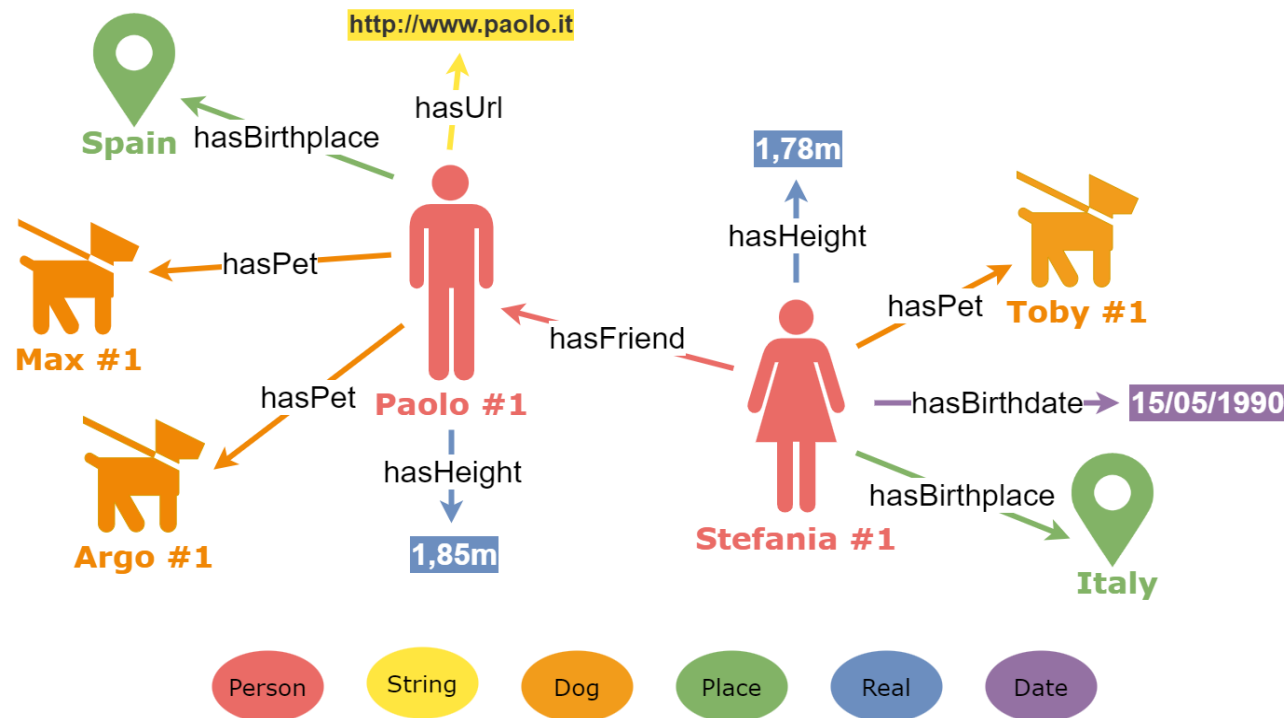
**Which percepts?**  
**Which facts?**

# Well-formedness conditions

An ETG, to be well-formed must satisfy the following conditions:

- Each node is associated one and ONLY one etype/ dtype.
- Each link is associated with one and only one data or object property.
- Data and object properties must have the correct etypes or datatypes (strong typing).
- No links are allowed starting from dtype nodes.

# An example of EG for the previous ETG



**Observation (ETG, EG).** An ETG defines all the etypes, dtypes, object properties and attributes used in an EG. An EG is an expansion (notion formally defined later on) where each node and link is expanded into all its elements. Compare with the previous ETG.



# LoD – which alphabet elements?

The **same** alphabet elements as percepts, that is:

- **Entity, etype, value, dtype, Attribute, Object Property** (as from ETG above).

**plus language elements needed to build knowledge statements**

- Etype specializations, using object and data properties
- Etype constructors, that is: intersection, union, complement.
- Defined etypes
- Equivalence/ subsumption /disjointness assertions of etypes (Semantically: set equality, subset relation, disjointness)

# LoD – The Logic of descriptions

- Intuition
- **Definition**
- Domain
- Language – intuition
- The language of etype percepts
- The language of composite etype percepts
- The language of descriptions
- Theories – TBoxes
- Entailment
- Ask
- Key notions

# LoD – The Logic of Descriptions - definition

We formally define LoD as follows

$$\text{LoD} = \langle \text{ETG}, \models_{\text{LoD}} \rangle$$

with

$$\text{ETG} = \langle L_{\text{LoD}}, D, I_{\text{LoD}} \rangle$$

When no confusion arises, we drop the subscripts.

# LoD – The Logic of descriptions

- Intuition
- Definition
- **Domain**
- Language – intuition
- The language of etype percepts
- The language of composite etype percepts
- The language of descriptions
- Theories – TBoxes
- Entailment
- Ask
- Key notions

# Domain


## Definition (Domain)

$$D = \langle \{C\}, \{R\} \rangle$$

where:

- $\{u\}$  is a set of **units**  $u_1, \dots, u_n$ , where  $u_1 \in U$ , with  $U = \{u\}$  is the **universe** of  $D$
- $\{C\}$  is a set of **classes**  $C_1, \dots, C_m$  of units, for some  $m$ , with  $C_i \subseteq U$
- $\{R\}$  is a set of **binary relations**  $R_1, \dots, R_p$  between units, for some  $p$ , with  $R_i \subseteq U \times U$

**Observation (Domain of interpretation).** The units in  $U$  are not part of the domain of interpretation of LoD. They are left implicit. The alphabet does not mention them.



# Classes

## Definition (Class)

$$\{C\} = E_T \cup D_T$$

where:

- $U = \{u\}$  is the **universe of interpretation**;
- $E = \{e\} \subseteq \{u\} = U$  is the **entity universe**;
- $V = \{v\} \subseteq \{u\} = U$  is the **value universe**;
- $\{e\}$  and  $\{v\}$  are disjoint.
- $E_T = \{E_i\}$  is a **set of etypes**  $E_i$ , with  $E_i = \{e\} \subseteq E$
- $D_T = \{D_i\}$  is a **set of dtypes**  $D_i$ , with  $D_i = \{v\} \subseteq V$

**Observation (LoD classes).** The same classes as LoE, but with no reference to the entities  $e$  and values  $v$  in  $U$ .

# Binary relations

## Definition (Binary relation)

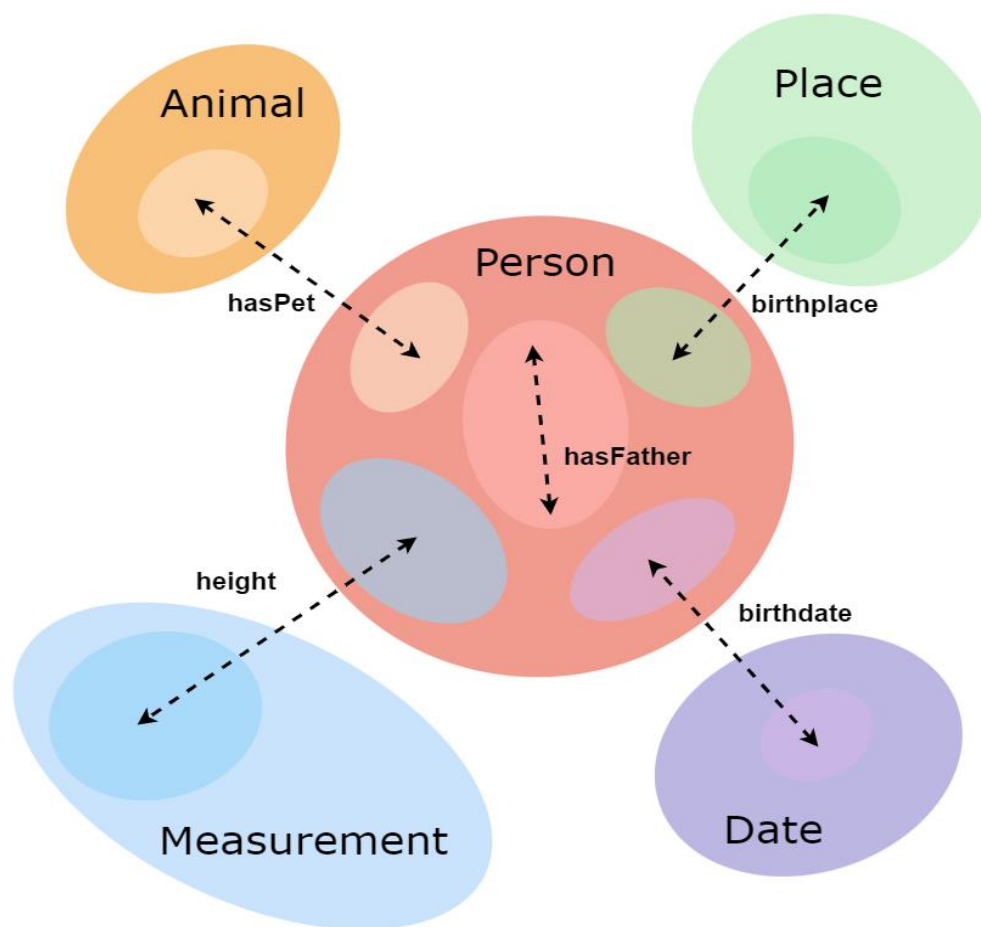
$$\{R\} = O_R \cup A_R$$

where:

- $O_R = \{O_i\}$  is a **set of object properties**  $O_i$ , with  $O_i \subseteq E_k \times E_j$
- $A_R = \{A_i\}$  is a **set of attributes**  $A_i$ , with  $A_i \subseteq E_k \times D_j$

**Observation (Object and data relations).** The same relations as LoE.

# An example of ETG – Venn diagram





# An example of domain of ETG (continued)

$$E_T = \{\text{Entity}, P, D, L, \dots\}$$

$$D_T = \{\text{Data}, \text{Real}, \text{String}, \dots\}$$

$$\{R\} = \{hF, hD, hH, hB, hL, hU, \dots\}$$

from which we construct the following facts in the domain:

$$D = \{P \subseteq \text{entity}, \text{Real} \subseteq \text{data}, hF(P, P), hD(P, D), hH(P, \text{Real}), \dots\}$$

with, e.g.,  $hF(P, P)$  standing for  $hF \subseteq P \times P$

# LoD – The Logic of descriptions

- Intuition
- Definition
- Domain
- **Language – intuition**
- The language of etype percepts
- The language of composite etype percepts
- The language of descriptions
- Theories – TBoxes
- Entailment
- Ask
- Key notions

# Language

**Intuition (Language).** The LoD language is progressively and compositionally extended in three steps:

- **Step 1 – The language of etype percepts.** It is an etype construction language. Composes etypes and properties to generate new etypes (e.g., a “mother” as a “parent” who is a “female”);
- **Step 2 – The language of composite etype percepts.** It is a composite etype construction language. Composes pairs of etypes to generate complex new etypes (e.g., the union of “mother” and “father”);
- **Step 3 – The language of Descriptions (LoD language).** It extends the alphabet by introducing names for defined etypes, as constructed in steps 1 and 2 (e.g., the new etype “parent” defined as the union of the etypes “mother” and “father”). Circular definitions are allowed.

**Observation (Language).** The LoD language is constructed in step 3 by exploiting the language extensions of the first two steps.

# Language - Observations

**Observation 1 (Percepts and facts).** Step 1 and step 2 only allow for the extension of the types of percepts. i.e., etypes, which can be talked about. Step 3 allows for assertions about what is the case in the domain.

**Observation 2 (LoD facts).** The LoD language (step 3) assertions do NOT necessarily describe single facts but complex composition of facts (e.g., the fact that cars are vehicles and that they have four vehicles).

**Observation (LoD percepts).** The LoD language (step 3) allows to **define new etypes**, that is, to introduce new terms which extend the language and which, therefore, generate in the domain of interpretation new simple percepts, i.e., etypes. This is the basic mechanism by which natural language and knowledge works. Thus, for instance:

- I can **define** a car as a street vehicle which is NOT a moto-bike, or a bus, or a track.
- I can **describe** a sport car as a car which have a certain shape and goes fast.

# Terminology – formulas

**Terminology (Formula, well-formed formula, wff).** We call

- formula, or
- well-formed formula, or
- wff

any string which is generated by a properly defined grammar. Assertions describe what is the case in a model, i.e., facts or compositions of facts. Assertions are special types of formulas. Not all formulas are assertions. Thus for instance:

- The formulas of etype and composite etype languages do not denote what is the case but, rather, complex percepts
- The formulas of the reasoning logic called LoP (for Logic of Proposition) denote truth or falsity.

# Language – Observations

**Observation 1 (Facts, percepts).** Facts define what is the case in a model as a composition of percepts, as they occur in the reference domain (see lectures before). These percepts and facts are not decomposable. We call them **atomic percepts** and **atomic facts**.

**Observation 2 (Complex percepts and facts).** We distinguish the percepts and facts denoted by the LoD language between **(atomic) percepts** and **facts** and **complex percepts** and **facts**, i.e., combinations of them.

**Observation 2 (Etype and composite complex percepts).** We distinguish **complex percepts** into (complex) **etyp e percepts**, obtained by defining a new etype based on its properties, and (complex) **composite etyp e percepts**, obtained by composing etypes.

**Observation 1 (Language and domains).** The LoD language allows for the construction of percepts (step 1, 2) and facts (step 3) which are not perceived. It creates a linguistic mental representation reality which does not map into the “reality” of what is being perceived, i.e. the analogic mental representation.

**Observation 2 (Language and domains).** The possibility of giving a name to the newly defined etypes allows for the generation of an unbound number of etypes and facts.

**Observation (Language heterogeneity).** LoD enables an expressiveness which approximates the complexity of human natural language. The mapping with the perceived reality further extended via a new many-to-many relation.

# Interpretation function – Observations

**Observation (Three LoD sub-languages composed in a single language).** Each of the three languages generates, via appropriate formation rules, more complex formulas using the formulas generated by the previous language as elements of the alphabet.

**Observation (Three nested interpretation functions).** Each language has its own interpretation function which uses the output of the interpretation function of the language one level below as its own input. We have nested interpretation functions.

**Observation (Domain in extension via language constructs).** The first two interpretation functions generate (formulas denoting) complex (etype and composite etype) percepts. The third generates complex assertions generating new facts.



# LoD – The Logic of descriptions

- Intuition
- Definition
- Domain
- Language – intuition
- **The language of etype percepts**
- The language of composite etype percepts
- The language of descriptions
- Entailment
- Ask
- Key notions



# Step 1 – The language of type percepts

## Definition (Language of etype percepts, $L_T$ )

$$L_T = \langle A_T, FR_T \rangle = \{p_T\}$$

where:

- $L_T$  is a **language** naming atomic etype percepts
- $A_T$  is an **alphabet**
- $FR_T$  is a **set of formation rules**
- $\{p_T\}$  is the set of type percepts  $p_T$  (dtypes plus atomic and complex etypes) obtained by the exhaustive application of  $FR_T$  to  $A_T$  (the transitive closure  $FR_T^+(A_T)$  of  $FR_T$  applied to  $A_T$ ).

# Alphabet

## Definition (Alphabet A)\*

$$A_T = \langle \{T\}, \{P\} \rangle$$

where:

- $\{T\} = \{E_i\} \cup \{D_i\}$  is a set of unary predicates standing for **etypes** and **dtypes**;
- $\{P\} = \{O_i\} \cup \{A_i\}$  is a set of binary **properties**, where  $O_i$  is an **object property**, also called a **role**, and  $A_i$  is an **attribute**.

**Observation (Alphabet of percepts).** Similarly to LoE,  $A_T$  is an alphabet which denotes percepts in the domain (but denoting a different set of percepts).

\*The elements of the alphabet are written in *italic* to distinguish them from percepts

# Formation rules – BNF

$$\langle p_T \rangle ::= \langle \text{etype} \rangle \mid \langle \text{dtype} \rangle \mid T \mid \perp$$
$$\langle \text{etype} \rangle ::= \exists \langle \text{objProp} \rangle . \langle \text{etype} \rangle \mid$$
$$\exists \langle \text{dataProp} \rangle . \langle \text{dtype} \rangle \mid$$
$$\forall \langle \text{objProp} \rangle . \langle \text{etype} \rangle \mid$$
$$\forall \langle \text{dataProp} \rangle . \langle \text{dtype} \rangle$$
$$\langle \text{etype} \rangle ::= E_1 \mid \dots \mid E_n$$
$$\langle \text{dtype} \rangle ::= D_1 \mid \dots \mid D_n$$
$$\langle \text{objProp} \rangle ::= O_1 \mid \dots \mid O_n$$
$$\langle \text{dataProp} \rangle ::= A_1 \mid \dots \mid A_n$$

**Observation (BNF).** This BNF does allow the iterative application of the formation rules on etypes (dtypes cannot be changed). It allows for the generation of etype percepts of any depth.

**Observation (BNF).** Entities are not mentioned (not part of the language). They are referred implicitly via the existential quantifier and also, somehow via the universal quantifier.

# Etype percepts – Example

- $\top$  - to be read **Top**
  - (*Intuition: the set of all entities*)
- $\perp$  - to be read **Bottom**
  - (*Intuition: the empty set of entities*)
- Person
  - (*Intuition: the set of entities which are called persons*)
- $\exists \text{hasFriend.Person}$ 
  - (*Intuition: the set of entities which have – at least – one friend who is a person*)
- $\forall \text{hasFriend.Person}$ 
  - (*Intuition: the set of entities whose friends are only persons, possibly none*)
- Integer
  - (*Intuition: the set of Integers*)
- $\exists \text{hasApple.Integer}$ 
  - (*Intuition: the set of entities which have at least one apple*)
- $\forall \text{hasApple.Integer}$ 
  - (*Intuition: the set of entities which have only not halved apples, possibly none*)

# Nested etype percepts – Example

- $\exists \text{talksTo.}(\exists \text{hasFriend.Person})$ 
  - **Intuition:** the set of entities which talk – at least once – to entities which have – at least – one friend who is a person
- $\forall \text{talksTo.}(\exists \text{hasFriend.Person})$ 
  - **Intuition:** the set of entities which talk only to the set of entities which have – at least – one friend who is a person
- $\exists \text{talksTo.}((\forall \text{hasFriend.Person})$ 
  - **Intuition:** the set of entities which talk – at least once – to the set of entities whose friends are only persons, possibly none
- $\forall \text{talksTo.}((\forall \text{hasFriend.Person})$ 
  - **Intuition:** the set of entities which talk only to the set of entities whose friends are only persons, possibly none

# Interpretation of etype percepts

$I_T(T) = U$ , with  $U$  the universe of interpretation

$I_T(\perp) = \emptyset$ , with  $\emptyset$  the empty set

$I_T(E_i) = E_i$

$I_T(D_i) = D_i$

$I_T(\exists P.T) = \{d \in U \mid \text{there is an } e \in U \text{ with } (d, e) \in I_T(P) \text{ and } e \in I_T(T)\}$

$I_T(\forall P.T) = \{d \in U \mid \text{for all } e \in U \text{ if } (d, e) \in I(P) \text{ then } e \in I_T(T)\}$

where  $I_T$  is the interpretation function of  $L_T$

**Observation (Interpretation function).** For an intensional view of the interpretation functions for etypes, dtypes, object properties and attributes, follow what done with LoE.

**Observation (Interpretation of nested etypes).** It is sufficient to interpret the application of the second external quantifier to the etype built via the application of the first quantifier.

# Interpretation of type percepts - Observations

**Observation (Existential type).**  $\exists P.T$  is an **existential etype**. Its interpretation

$$I(\exists P.T) = \{d \in U \mid \text{there is an } e \in U \text{ with } (d, e) \in I(P) \text{ and } e \in I(T)\}$$

is the set of all units  $d$  for which there exists a unit  $e$  in the codomain of  $P$  or type  $T$ .  $\exists P.T$  defines the etype which is in relation  $P$  with  $T$ , not necessarily only with  $T$  (there could be an  $e'$  not in  $I(T)$ ).

**Example.**  $\exists \text{hasFriend.Person}$  is the etype of all those entities who have *at least* a friend who is a person.



# Interpretation of etype percepts - Observations

**Observation (Universal etype).**  $\forall P.T$  is an **universal etype**. Its interpretation

$$I(\forall P.T) = \{d \in U \mid \text{for all } e \in U \text{ if } (d, e) \in I(P) \text{ then } e \in I(T)\}$$

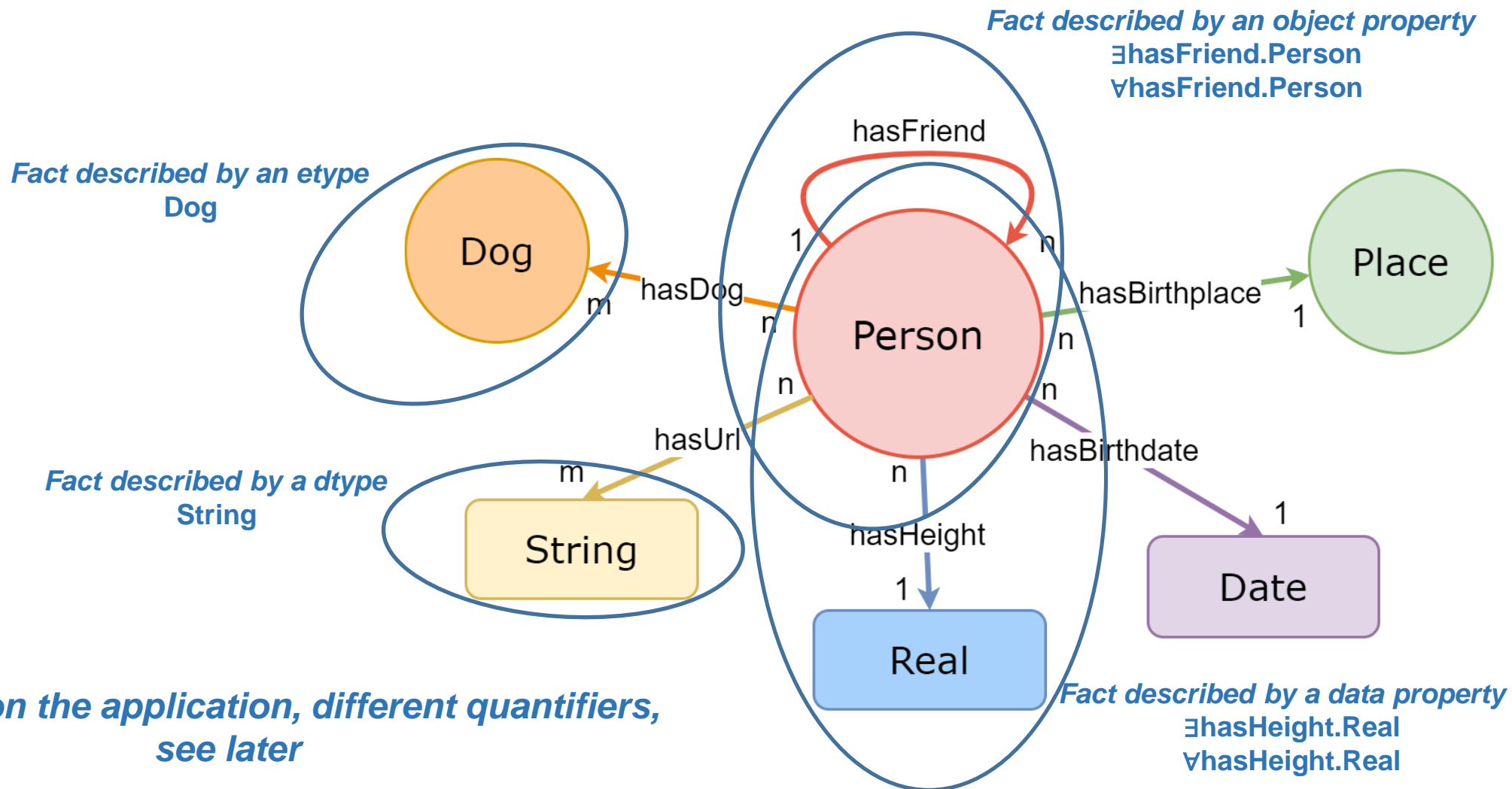
is the set of all units  $d$  for which all the units  $e$  in the codomain of  $P$  are of type  $T$ .

**Example.**  $\forall \text{hasFriend.Person}$  is the etype of all people whose friends are *only* persons.

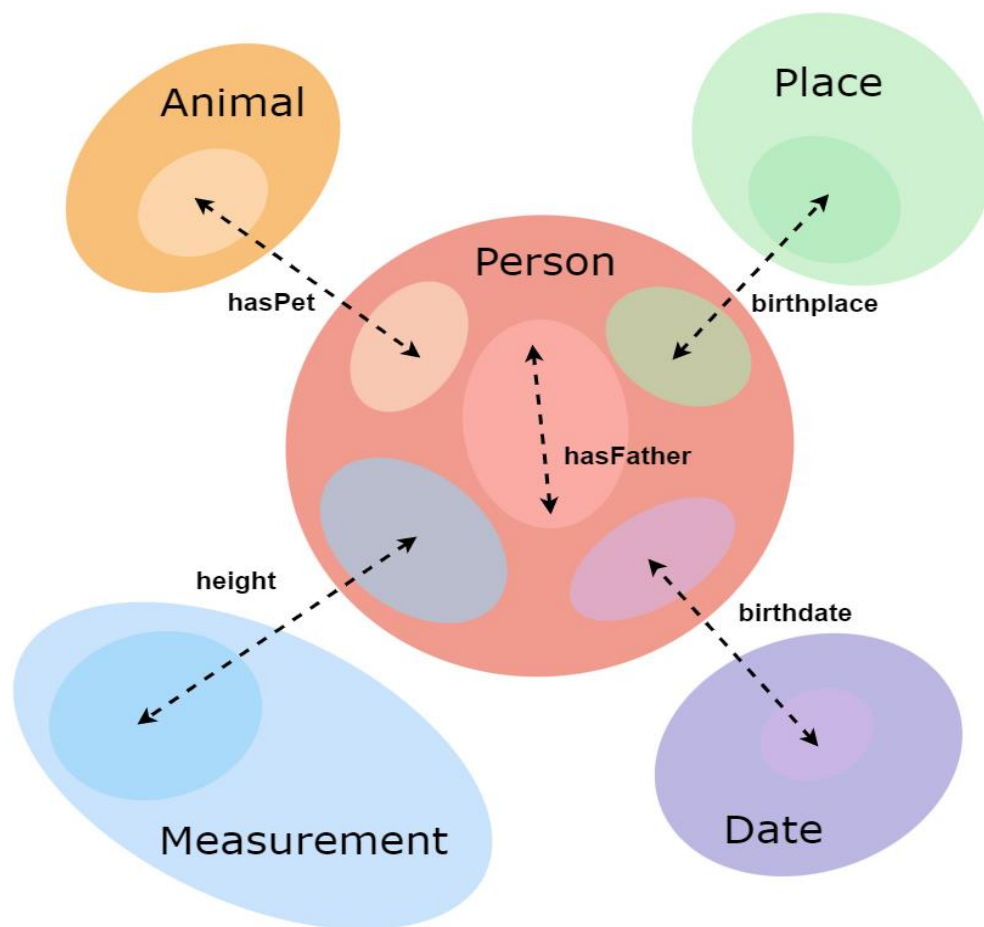
**Proposition.**  $\forall P.T$  does not imply  $\exists P.T$ .  $\forall P.T$  holds if  $I(\forall P.T) = \emptyset$ , while this is not the case with  $\exists P.T$ .



# Example – how LoD (names of) percepts represent ETG facts



# Interpretation function (Venn diagram)



**Most often, in informal  
world models we  
assume both  
universal and existential  
quantifiers**

**The first  
does not necessarily  
imply the second**

# LoD – The Logic of descriptions

- Intuition
- Definition
- Domain
- Language – intuition
- The language of etype percepts
- **The language of composite etype percepts**
- The language of descriptions
- Entailment
- Ask
- Key notions

## Step 2 – The language of composite etype percepts

**Definition (The language of composite etype percepts,  $L_C$ )**

$$L_C = \langle A_C, FR_C \rangle = \{p_C\}$$

where:

- $L_C$  is a **language** of composite etype percepts
- $A_C = \{p_T\}$ , the **alphabet**, consists of all the percepts  $p_T \in L_T$
- $FR_C$  is a **set of formation rules**
- $\{p_C\}$  is the set of composite etype percepts  $a_C$  obtained by the exhaustive application of  $FR_C$  to  $A_C$  (the transitive closure  $FR_C(A_C)$  of  $FR_C$  applied to  $A_C$ ).

# Formation rules – BNF

$$\begin{aligned} \langle p_C \rangle & ::= \langle p_C \rangle \sqcap \langle a_C \rangle \mid \\ & \quad \langle p_C \rangle \sqcup \langle p_C \rangle \mid \\ & \quad \neg \langle p_C \rangle \end{aligned}$$

$$\langle p_C \rangle ::= \langle p_T \rangle$$

**Notation (BNF).**  $\langle p_C \rangle$  is a nonterminal symbol and it stands for a  $p_C$  percept.  $\langle p_T \rangle$  is an  $L_C$  terminal symbol and it stands for an  $L_T$  percept. See the BNF of  $L_T$  to see how to expand it to a LoD terminal symbol.

**Observation (BNF).** This BNF does allow the iterative application of the formation rules. It allows to generate percepts of any depth.

# Composite etype percepts – Example

- $\text{Person} \sqcap \exists \text{hasFriend. Person}$ 
  - (***Intuition:** the set of entities which are persons and have a friend which is a person*)
- $\text{Person} \sqcup \text{Dog}$ 
  - (***Intuition:** the set of entities which are a person or a dog*)
- $\neg \exists \text{hasFriend. Person}$ 
  - (***Intuition:** the set of entities do not have a friend which is a person*)
- $\text{Person} \sqcap \neg (\exists \text{hasFriend. Person})$ 
  - (***Intuition:** the set of entities which are persons and which do not have a friend which is a person*)

# Composite etype percepts – Example etypes

*Consider the following concept names:*

Vehicle, Boat, Bicycle, Car, Device, Wheel, Engine, Axle, Rotation,  
Water, Human, Driver, Adult, Child

*Formalize the following natural language statements:*

- Nothing (empty set):  $\perp$
- Everything (All the interpretation domain):  $\top$
- Humans which are drivers :  $\text{Human} \sqcap \text{Driver}$
- Humans and vehicles:  $\text{Human} \sqcup \text{Vehicle}$
- Vehicles which are not boats:  $\text{Vehicle} \sqcap \neg \text{Boat}$
- Wheels or engines which are used in cars:  $(\text{Wheel} \sqcup \text{Engine}) \sqcap \text{Car}$
- Adults or children:  $\text{Adult} \sqcup \text{Child}$



# Composite etype percepts – Example roles

*Consider the previous concept names plus the following role names:*

hasPart, poweredBy, capableOf, travelsOn, controls

*Formalize in DL the following natural language statements:*

1. Those vehicles that have wheels and are powered by an engine
2. Those vehicles that have wheels and are powered by a human
3. Those vehicles that travel on water
4. Those objects which have no wheels
5. Those objects which do not travel on water
6. Those devices that have an axle and are capable of rotation
7. Those humans who control a vehicle
8. The drivers of cars



# Composite etype percepts – Example roles (cont)

1.  $\text{Vehicle} \sqcap \exists \text{hasPart.Wheel} \sqcap \exists \text{poweredBy.Engine}$
2.  $\text{Vehicle} \sqcap \exists \text{hasPart.Wheel} \sqcap \exists \text{poweredBy.Human}$
3.  $\text{Vehicle} \sqcap \exists \text{travelsOn.Water}$
4.  $\forall \text{hasPart.} \neg \text{Wheel}$
5.  $\forall \text{travelsOn.} \neg \text{Water}$
6.  $\text{Device} \sqcap \exists \text{hasPart.Axle} \sqcap \exists \text{capableOf.Rotation}$
7.  $\text{Human} \sqcap \exists \text{controls.Vehicle}$
8.  $\text{Driver} \sqcap \exists \text{controls.Car}$

# Interpretation of composite etype percepts

$$I_C(p_1 \sqcap p_2) = I_C(p_1) \cap I_C(p_2)$$

$$I_C(p_1 \sqcup p_2) = I_C(p_1) \cup I_C(p_2)$$

$$I_C(\neg p_1) = U \setminus I_C(p_1)$$

$$I_C(p_T) = I_T(p_T)$$

$$I_T(p_T) = p_T$$

where:

- $I_C$  is the interpretation function of  $L_C$
- $I_T$  is the interpretation function for  $L_T$ , the language of etype percepts.
- $p_1, p_2$  are composite etype percepts
- $p_T$  (in *italic*) is (the name of an) etype percept denoting the domain percept  $p_T$  (not in *italic*)

# LoD – The Logic of Descriptions

- Intuition
- Definition
- Domain
- Language
- Interpretation function
- Entailment
- Ask / Tell
- **Key notions**

# LoD – The Logic of descriptions

- Intuition
- Definition
- Domain
- Language – intuition
- The language of etype percepts
- The language of composite etype percepts
- The language of descriptions
- Theories – TBoxes
- Entailment
- Ask
- **Key notions**

# Key notions

- LoD as a logic of entities
- Entity graphs (EGs)
- Entities and values,
- Etypes and data types
- Object properties
- Data properties (alias attributes)
- LoD alphabet
- LoD formation rules
- LoD assertions
- LoD expressiveness
- Interpretation function
- Intensional definition of an interpretation function
- Extensional definition of an interpretation function
- Entailment relation
- LoD Tell
- LoD Instance checking
- OWA
- CWA
- LoD Instance retrieval



UNIVERSITY  
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione



# LoD

# The logic of Descriptions

# (HP2T)

# Correctness and completeness

**Proposition (Language correctness and completeness).**  $L_{LoD}$  is correct with respect to  $D_{LoD}$ .  $L_{LoD}$  is complete with respect to  $D_{LoD}$  if  $I_{LoD}$  is surjective.

**Evidence (Language correctness and completeness).** From the definition of  $L_{LoD}$  and  $I_{LoD}$  for any assertion there is a corresponding fact in the domain, not necessarily only one. We have completeness when the alphabet contains symbols for all the percepts in the domain.

**Proposition (Theory correctness and completeness).** A theory  $T_a = \{a\}$  such that, if  $a \in T_a$  then  $I(a) \in M$  is correct with respect to  $M$ .  $T_a$  is complete only if it is maximal.

**Evidence (Theory correctness and completeness).** By construction.