**LECTURE 7**

**ONTOLOGIES IN FIRST ORDER LOGIC**

**SUBSETS OF FIRST ORDER LOGIC**

Because FOL is very general and it is based on very primitive concepts (constants, variables, function symbols, predicates and quantifiers), some people have chosen to study subsets of FOL that are most appropriate for higher-level abstractions such as taxonomies, time, space etc. In addition, by going to subsets of FOL, we try to avoid undecidability, and even better, have a subset for which efficient implementations of reasoning algorithms exist (as e.g., in Datalog)

A subset of first-order logic is defined by imposing certain limitations or constraints on the syntax and semantics of the logic. These limitations can be in terms of the allowed language constructs, axioms, inference rules, or the use of specific operators.

The purpose of defining subsets of first-order logic is to tailor the logic to specific needs, applications, or computational properties. By restricting the language, we can often achieve desirable properties, such as decidability (the ability to determine the truth or validity of formulas) or efficient reasoning algorithms.

**ONTOLOGIES**

Ontologies are formal representations of knowledge that capture concepts, relationships, and properties within a specific domain. They provide a structured and organized way to model and describe knowledge in a domain, enabling effective information sharing, reasoning, and knowledge management.

In an ontology, concepts are represented as classes or categories, relationships between concepts are represented as properties or predicates, and properties can have attributes or restrictions associated with them. The relationships and attributes defined in an ontology help define the semantics and structure of the domain it represents.

1. Conceptualization -: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose
2. Relationships and Properties: Ontologies define relationships between concepts, representing the various ways in which concepts are connected or related. Properties capture the attributes, roles, or characteristics associated with concepts and their relationships.
3. Formalization: Ontologies employ formal logic to specify the semantics of concepts, relationships, and properties. This enables reasoning and inference over the knowledge represented in the ontology, allowing for automated analysis and deduction.

The literature also offers us special formalisms for defining ontologies that

contain mainly taxonomic knowledge:

* Semantic networks
* Frames
* Description logics
* RDF, RDF(S) and OWL 2 for ontologies in the Semantic Web.

You can think about these formalisms as being object-oriented logics:

* They have special constructs for representing knowledge about individuals (or objects), categories (or classes) and relationships (or roles).
* Categories are organized into taxonomies.
* They have special reasoning methods to deal with these constructs.

**RESOURCE DESCRIPTION FRAMEWORK**

The Resource Description Framework (RDF) is a general [framework](https://www.techtarget.com/whatis/definition/framework) for representing interconnected data on the web. RDF statements are used for describing and exchanging [metadata](https://www.techtarget.com/whatis/definition/metadata), which enables standardized exchange of data based on relationships.

metadata is structured reference data that helps to sort.

RDF statements express relationships between resources, such as the following:

* documents
* physical objects
* people
* abstract concepts
* data objects

Collections of related RDF statements comprise a directed graph that maps the relationships among entities.

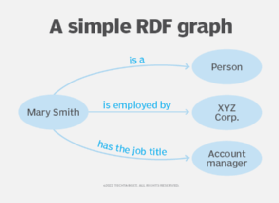
**How does RDF work?**

RDF is a standard way to make statements about resources. An RDF statement consists of three components, referred to as a *triple*:

1. **Subject** is a resource being described by the triple.
2. **Predicate** describes the relationship between the subject and the object.
3. **Object** is a resource that is related to the subject.

The subject and object are nodes that represent things. The predicate is an arc, because it represents the relationship between the nodes.

RDF is (roughly) limited to binary ground predicates,



**RDF SCHEMA**

RDF Schema is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources

RDF Schema is (roughly) limited to a subclass hierarchy and a property hierarchy, with domain and range definitions of these properties.

* They are designed with flexibility in mind

<https://redirect.cs.umbc.edu/courses/graduate/691/fall17/01/notes/05_rdfs/05rdfs.pdf>

**OWL 2**

OWL2, short for Web Ontology Language 2, is a widely-used ontology language designed to represent knowledge and facilitate semantic reasoning on the World Wide Web. It is an extension of its predecessor, OWL (OWL1), and provides enhanced features and expressiveness for representing ontologies.\

OWL2 incorporates a rich set of constructs and features that enable precise modeling of complex relationships and constraints within ontologies. It supports classes, properties, individuals, and relationships, allowing for the representation of various types of knowledge and domain-specific concepts. Furthermore, it includes powerful axioms, such as class and property hierarchies, cardinality restrictions, and logical relationships (e.g., conjunction, disjunction, negation), which enable sophisticated reasoning and inference.

OWL2 also offers support for modularity and reusability through the use of imports and ontology annotations. This allows ontologies to be divided into smaller modules and reused across different applications, fostering scalability and interoperability.

**TAXONOMIES**

Taxonomic information plays a central role in other Computer Science areas e.g., programming languages, databases and software engineering. Taxonomies are very important in modern applications: information retrieval in the Web, information integration, knowledge management, e-commerce, e-science, e-government etc.

A taxonomy is a hierarchical framework, or schema, for the organization of organisms, inanimate objects, events and/or concepts.

A taxonomy must :

* Follow a hierarchic format and provides names for each object in relation to other objects.
* May also capture the membership properties of each object in relation to other objects.
* Have specific rules used to classify or categorize any object in a domain. These rules must be complete, consistent, and unambiguous.
* Apply rigor in specification, ensuring any newly discovered object must fit into one and only one category or object.
* Inherit all the properties of the class above it, but can also have additional properties.

**UPPER ONTOLOGIES VS DOMAIN ONTOLOGIES**

1. Upper Ontologies: Upper ontologies, also known as foundational ontologies or top-level ontologies, provide a broad and general framework for representing knowledge that can be applied across multiple domains. They aim to capture universal concepts and categories that are common to many different areas of knowledge.

Examples of upper ontologies include:

* DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)
* BFO (Basic Formal Ontology)
* SUMO (Suggested Upper Merged Ontology)
* Cyc

1. Domain Ontologies: Domain ontologies, also known as application ontologies, are specific to a particular domain or subject area. They focus on capturing the concepts, entities, relationships, and constraints relevant to that domain. Domain ontologies are typically built on top of or aligned with an upper ontology to inherit the foundational concepts and provide more specific details.

Examples of domain ontologies include:

* SNOMED CT (Systematized Nomenclature of Medicine—Clinical Terms)
* Gene Ontology (GO)
* FOAF (Friend of a Friend)
* SWEET (Semantic Web for Earth and Environmental Terminology)

**CATEGORIES IN FIRST ORDER LOGIC**

FOL offers us two ways to talk about categories (or classes) of objects:

**Unary predicate symbols. -** Unary predicates are used to express properties or characteristics of individual objects in the domain of discourse.

For example: Person(x), FemalePerson(x)

**Constant symbols (through reification).** - In first-order logic (FOL), constant symbols are individual elements or objects in the domain of discourse that do not change their value. They represent specific, fixed entities and are used to denote specific elements or objects in the logical language.

Constant symbols are typically represented by uppercase letters or lowercase letters at the beginning of the alphabet. For example, "a" or "b" could be constant symbols representing specific objects or individuals.

For example: Persons, FemalePersons.

In this case, we also need predicates for membership and subclass: MemberOf and SubsetOf

In FOL we can use the following methods to talk about properties of an object or relationships among two objects:

**PROPERTIES OR RELATIONSHIPS IN FOL**

Binary predicate symbols - : Binary predicates express relationships or connections between two objects. For example:

SisterOf (x, y), HasMother(x, y), Legs(x, n)

* Likes(x, y)": Represents the relationship that person x likes person y.
* "IsParentOf(x, y)": Represents the relationship that person x is a parent of person y.
* "IsAdjacentTo(x, y)": Represents the relationship that object x is adjacent to object y.

Functions: Functions represent relationships or computations that involve objects and return a value. For example:

* "Sum(x, y)": Represents the function that calculates the sum of x and y.
* "Distance(x, y)": Represents the function that calculates the distance between x and y.
* "SquareRoot(x)": Represents the function that calculates the square root of x.
* BiologicalFatherOf (x) = y, Legs(x) = n

**1.MEMBERSHIP OF AN OBJECT IN A CATEGORY**

We have two ways to express that an object is a member of a category.

Example:

FemalePerson(Mary)

MemberOf (Mary, FemalePersons)

In the second option, predicate MemberOf has to be axiomatized appropriately.

**2 Objects in a relationship**- : SisterOf (Mary, John)

Examples of asserting that a property of an object has a value:

Legs(John,2), or by using a function symbol Legs(John) = 2

HasTelNo(John, 6975532334)

HasTelNo(John, 2107943525)

**2.CATEGORY IS A SUBCLASS OF ANOTHER CATEGORY**

We have two ways to express that a category is a subcategory (subclass)

of another category:

(∀x)(BasketBall(x) ⇒ Ball(x))

SubsetOf (BasketBalls, Balls)

For the second option, predicate SubsetOf has to be axiomatized appropriately

**NECESSARY CONDITIONS FOR MEMBERSHIP IN A CATEGORY**

Examples:

(∀x)(BasketBall(x) ⇒ Shape(x,Round))

(∀x)(MemberOf (x, BasketBalls) ⇒ Shape(x, Round))

**SUFFICIENT CONDITIONS FOR MEMBERSHIP IN A CATEGORY**

We can also express that all members of a category can be recognized by some properties (i.e., these are sufficient conditions for being a member of the category).

Example:

(∀x)(Orange(x) ∧ Round(x) ∧ Diameter(x) = 9.5!!∧

MemberOf (x, Balls) ⇒ MemberOf (x, BasketBalls))

Definition. An object is a triangle if and only if it is a polygon with three sides.

(∀x)(MemberOf (x, Triangle) ⇔ (MemberOf (x, Polygon)∧NoOfSides(x, 3))

**Category can be a member of category of categories**

Example: MemberOf (Dogs, DomesticatedSpecies)

ReproductiveCycle(FemaleDogs, 6months)

HasCardinality(MyDogs, 3)

**Other Set-Theoretic Relations Among Categories**

Often we want to say that two categories are disjoint or that they form an exhaustive decomposition of some other category or that they form a partition of some other category.

Examples:

Disjoint({Animals, Vegetables})

ExhaustiveDecomposition({Americans, Canadians, Mexicans},

NorthAmericans)

Partition({Males, Females},Animals)

**RELATIONSHIPS AMONG CATEGORIES**

The three predicates used above can be defined as follows:

(∀s)(Disjoint(s) ⇔ (∀c1, c2)(c1 ∈ s ∧ c2 ∈ s ∧ c1 /= c2 ⇒ Intersection(c1,c2) = {}))

(∀s, c)(ExhaustiveDecomposition(s, c) ⇔ (∀i)(i ∈ c ⇒ (∃c2)(c2 ∈ s ∧ i ∈ c2)))

(∀s, c)(Partition(s, c) ⇔ Disjoint(s) ∧ ExhaustiveDecomposition(s, c))

In the above formulas we use ∈ instead of MemberOf . Predicate

Intersection needs to be axiomatized appropriately.

The method of using unary predicates for representing knowledge about categories is weaker than the method of using constants.

In other words, the latter method can represent everything that the former one can and more (as we saw in the example with categories of categories).

The two methods can co-exist.

**Physical Composition**

The idea that one object is part of another is an important one in many domains (e.g., engineering design, e-commerce catalogs, human anatomy).

We use the general predicate PartOf to represent such information.

Example:

PartOf (Athens,Greece), PartOf (Greece,WesternEurope)

PartOf (WesternEurope,Europe), PartOf (Europe,Earth)

The relation PartOf is irreflexive and transitive:

(∀x)(¬PartOf (x, x))

(∀x, y, z)(PartOf (x, y) ∧ PartOf (y, z) ⇒ PartOf (x, z))

Thus we can conclude: PartOf (Athens,Earth)

Categories of composite objects are often characterized by the structure of those objects i.e., the parts and how the parts relate to the whole.

**DEFINING A BIPED**

(∀a)(Biped(a) ⇔

(∃l1, l2, b)(Leg(l1) ∧ Leg(l2) ∧ Body(b)∧

PartOf (l1, a) ∧ PartOf (l2, a) ∧ PartOf (b, a)∧

Attached(l1, b) ∧ Attached(l2, b)∧

l1 /= l2 ∧ (∀l3)(Leg(l3) ⇒ (l3 = l1 ∨ l3 = l2))))