Context-Aware Computing 3

Context Representation and Reasoning

Mobile Computing

Minho Shin

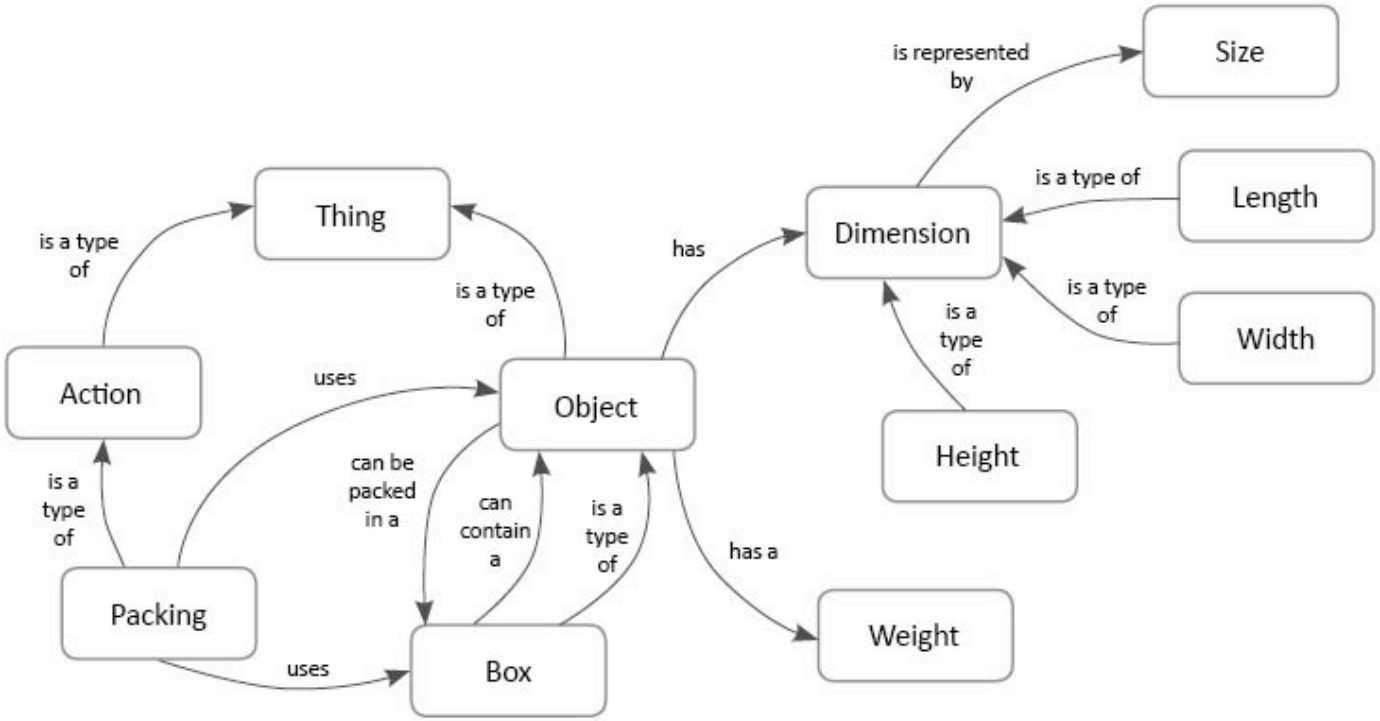
2012. 11

Context Representation & Reasoning

- How can we make inferences on the user context based on lower context data?
  - Logic programming
  - Ontology-based
  - Case-based
- Knowledge Representation & Reasoning

ONTOLOGY FOR CR&R

Background: Ontology



An object is a type of thing. An object has dimensions. Length is a type of dimension. Width is a type of dimension. Height is a type of dimension. A dimension is represented by a size. An object has a weight.

An action is a type of thing. Packing is a type of action. The packing action uses object and a box. An object is packed in a box. A box contains an object.

Ontology in Philosophy

- The term Ontology has its origin in philosophy
- Ontology: the branch of philosophy which deals with the nature and the organization of reality
- In this sense the Ontology tries to answer to the question:
  - What is being?
  - What are the features common to all beings?
- Recently adopted in several fields of computer science and information science ⇒ several meaning have been assigned to the ”Ontology” term

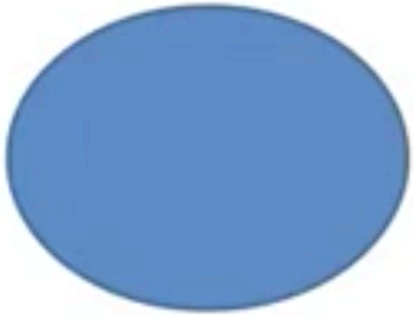
Ontology

- Ontology vs. ontology
- *Ontology*
  - Ontology = ontos (being) + logica (study of)
  - An old discipline introduced by Aristotle, which attempts to address questions such as: 'What is being?' and 'What characteristics do all beings have in common?'.
- *ontology*
  - a system of categories that account for a certain view of the world
  - An individual is in possession of some concepts in his or her mental model

Ontology

- Definition in Computer Science
  - A data model that represents knowledge
  - as a set of concepts within a domain
  - and the relationships between these concepts
- Formal: machine readable

CLASSES



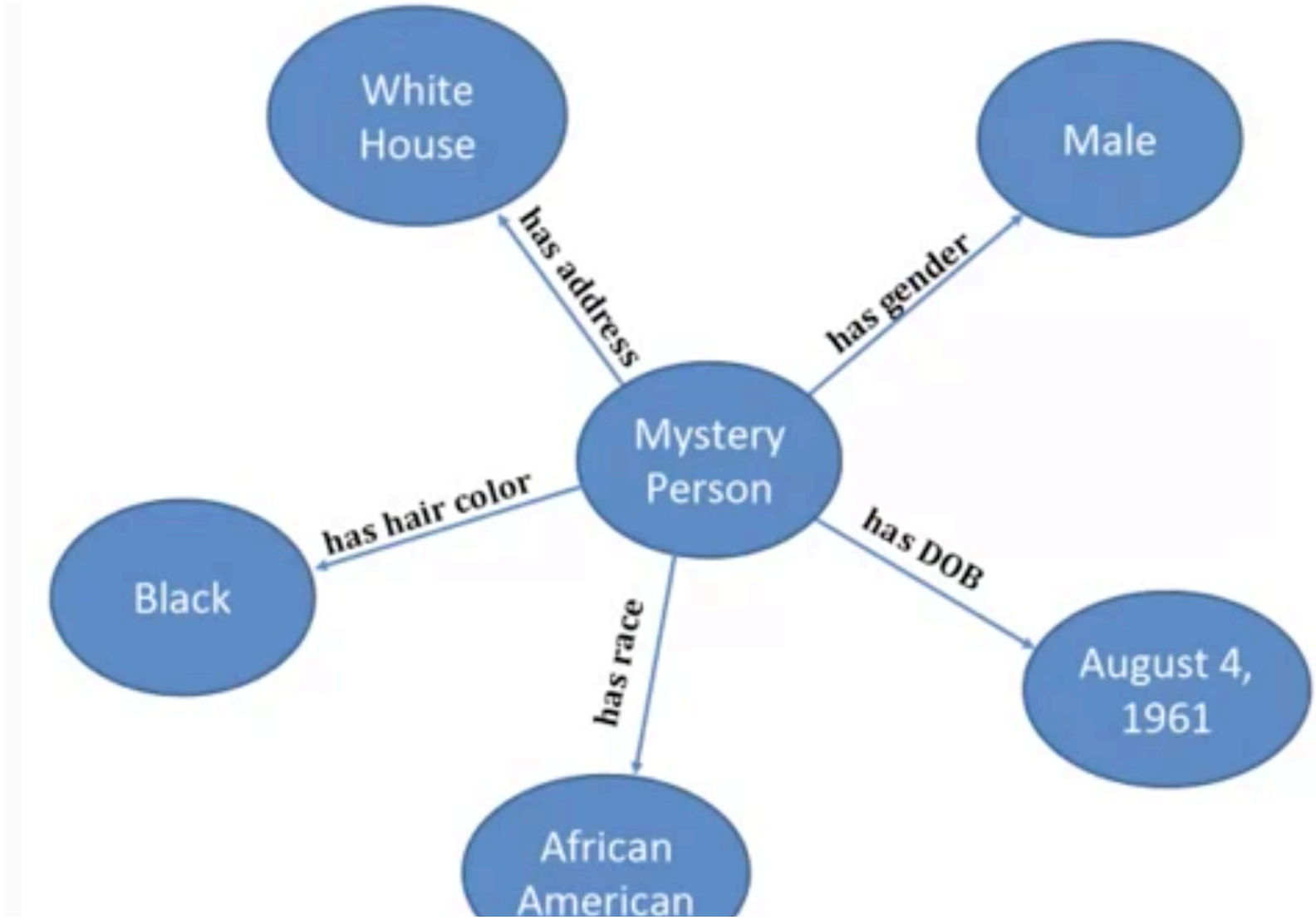
RELATIONSHIPS

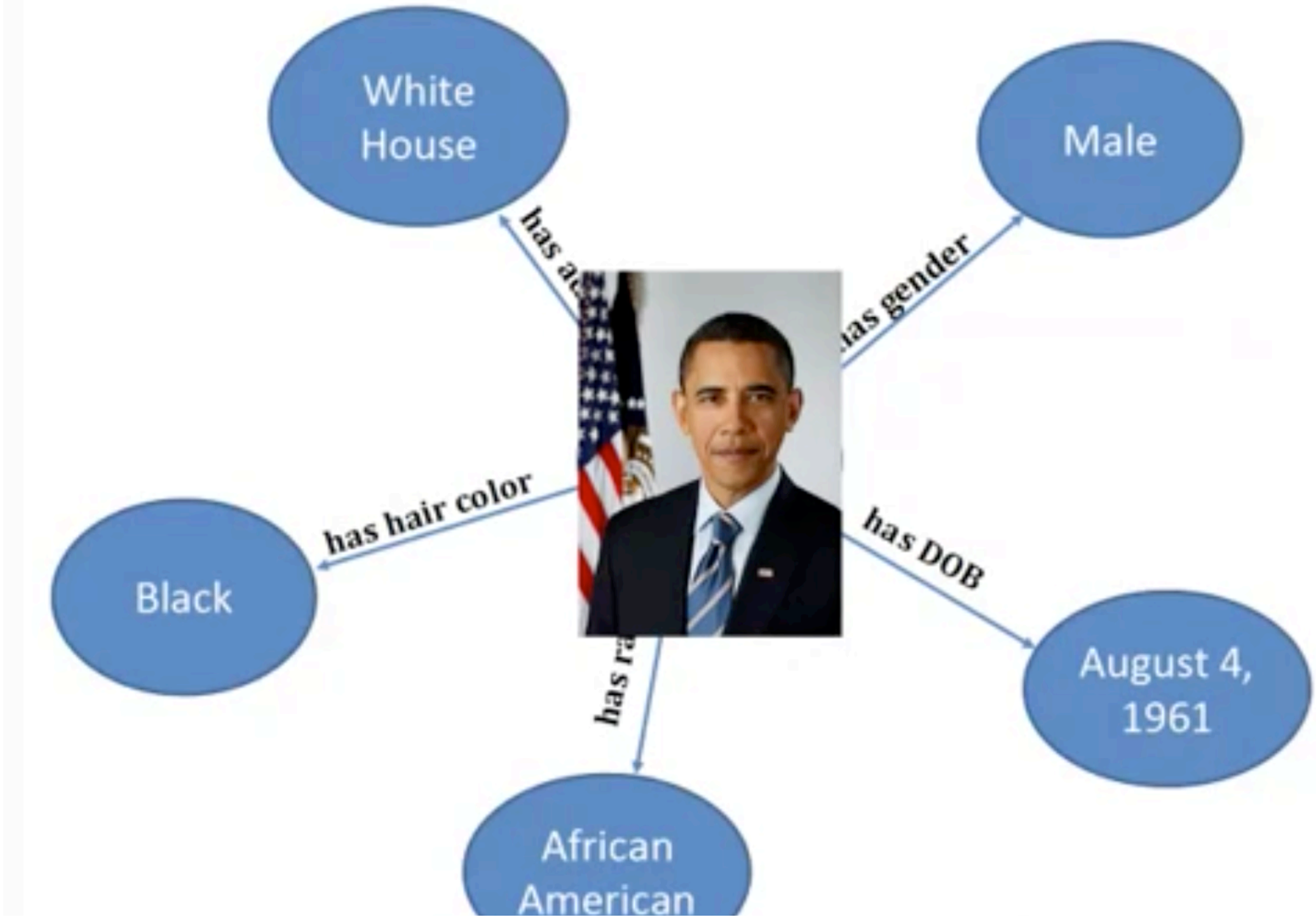
CLASSES

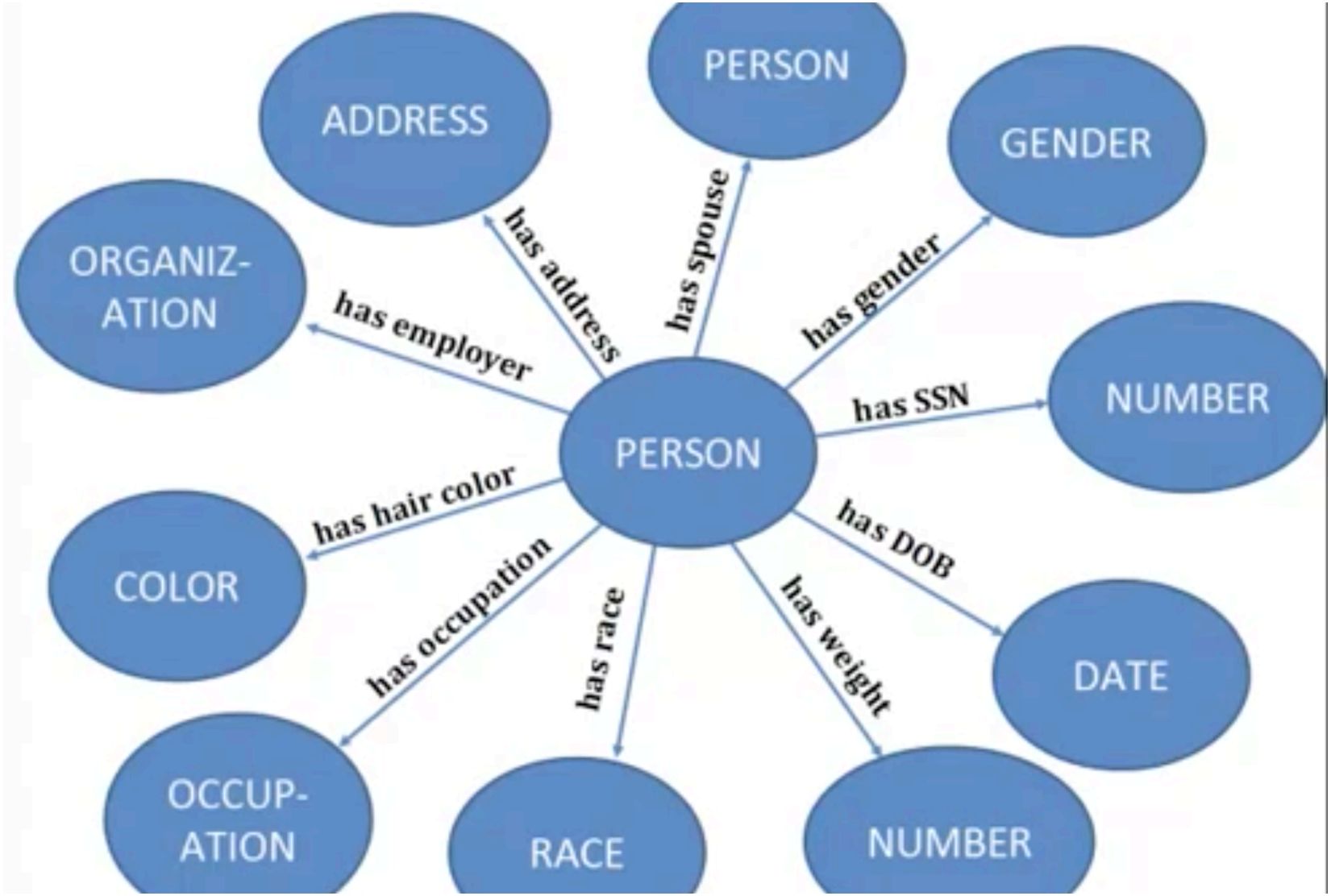
RELATIONSHIPS

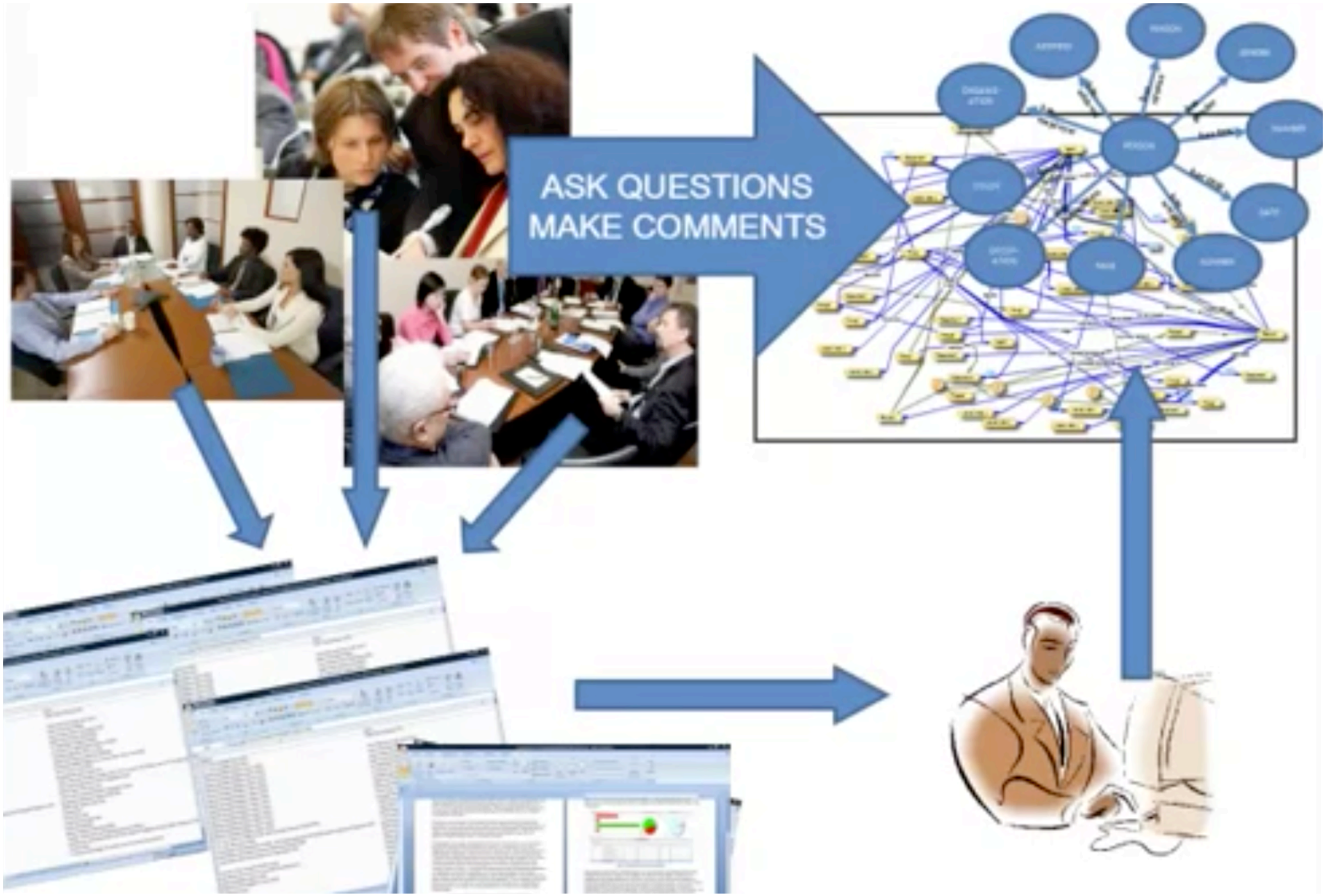

Person
has employer
ORGANIZATION


SUBJECT
PREDICATE
OBJECT
has employer
Todd
Spry, Inc
TRIPLE

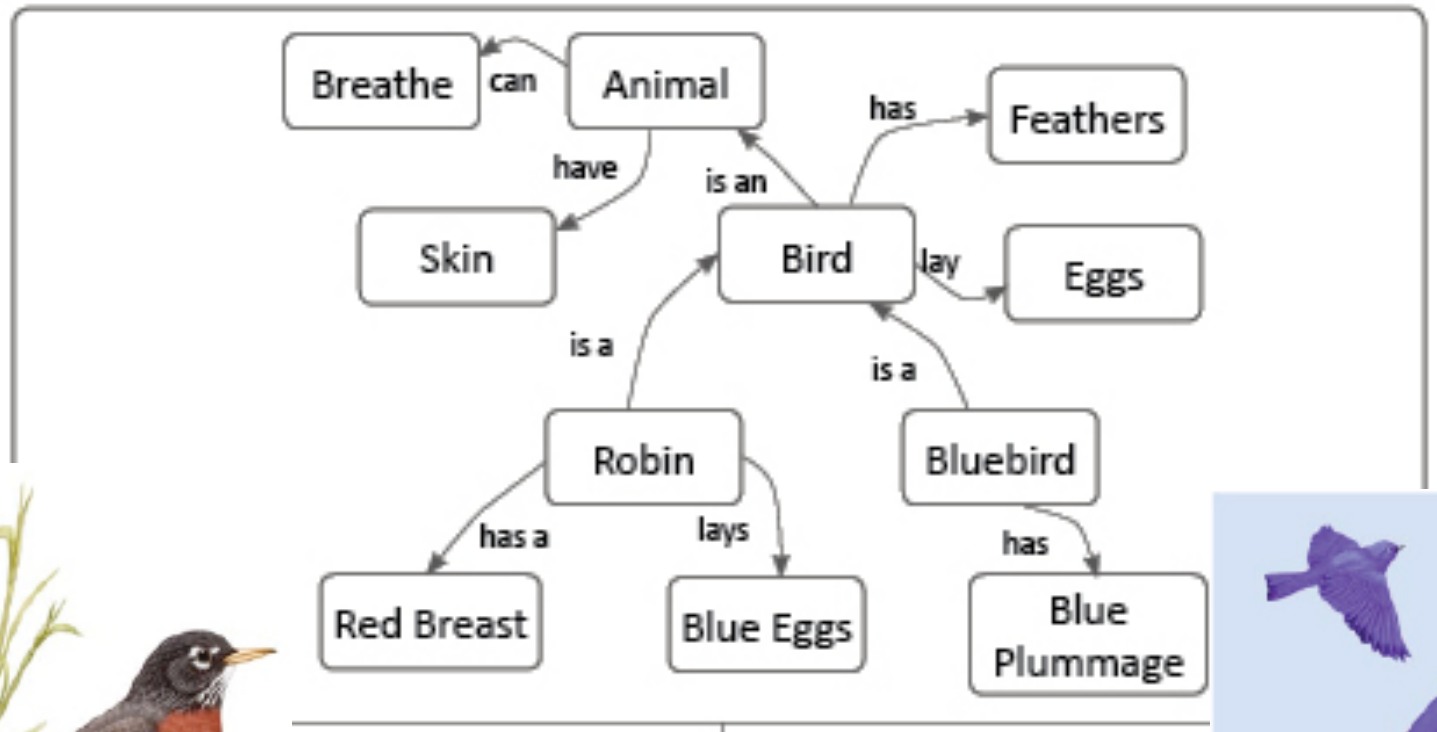

White House
has address
Male
has gender
Mystery Person
has hair color
Black
has race
has DOB
August 4, 1961
African American


White House
has at
Male
has gender
has hair color
Black
has DOB
August 4, 1961
African American


ADDRESS
PERSON
GENDER
ORGANIZ-
ATION
has address
has spouse
has gender
has employer
PERSON
has SSN
NUMBER
has hair color
COLOR
has occupation
has race
has weight
has DOB
DATE
OCCUP-
ATION
RACE
NUMBER


ASK QUESTIONS
MAKE COMMENTS

Bird


Breathe
can
Animal
has
Feathers
have
is an
Skin
Bird
lay
Eggs
is a
is a
Robin
Bluebird
has a
lays
has
Red Breast
Blue Eggs
Blue Plummage

Basic Elements of Ontology

- *Individuals*
  - ”Ground level” components of an ontology.
  - Concrete objects of a domain i.e. people, animals, automobiles, molecules...
  - Abstract individuals i.e. numbers and words.
- *Concepts*
  - Collections of objects.
  - They may contain individuals, other classes, or a combination of both.
  - Molecule, the class of all molecules
  - Vehicle, the class of all vehicles
  - Car, the class of all cars

Basic Elements of Ontology

- *Attributes*: describe the objects in the ontology.
  - Ex.: the Ford Explorer object has attributes:
    - Number-of-doors: 4
    - Transmission: 6-speed
- *Relationships*: make explicit the links between objects.
  - An attribute whose value is another object in the ontology,
    - Ex.: The attribute Successor:Ford Explorer of Ford Bronco means that Explorer is the modeled that replaced Bronco.
  - A mathematical relation
    - Ex.: Successor(Ford Bronco,Ford Explorer)
- Much of the power of ontologies comes from the ability to describe these relations.


What is an Ontology
Like
Share
More info
RDF
Resource Description Framework
Web Ontology Language
1:15 / 4:35

DESCRIPTION LOGIC FOR CR&R

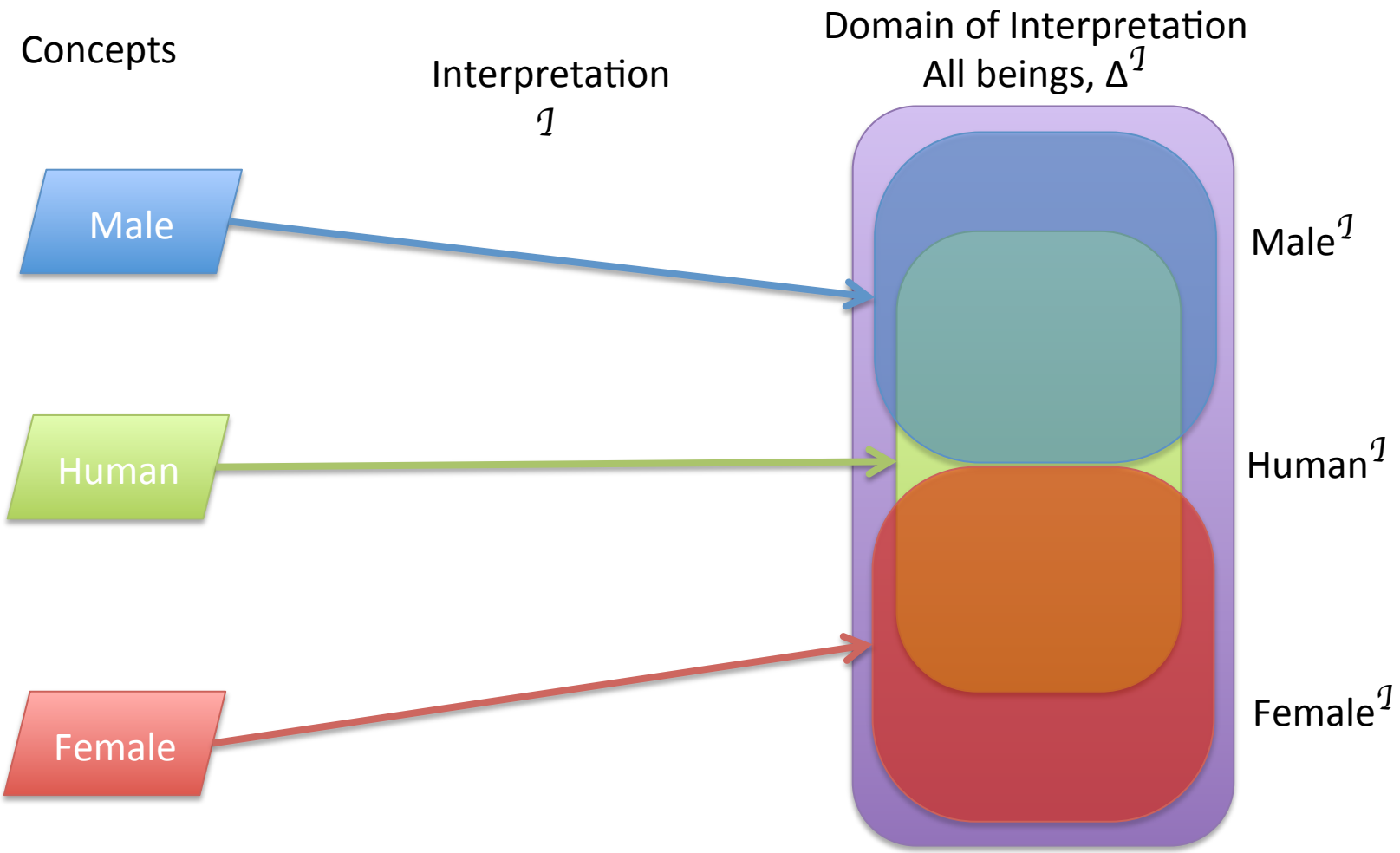
Description Logic

- A formalism for structured representation of knowledge
- Knowledge (of a domain) should be structured by
  - Class: a set of objects with common properties
  - Role: relationship among classes
- DL Knowledge Base = TBox + ABox

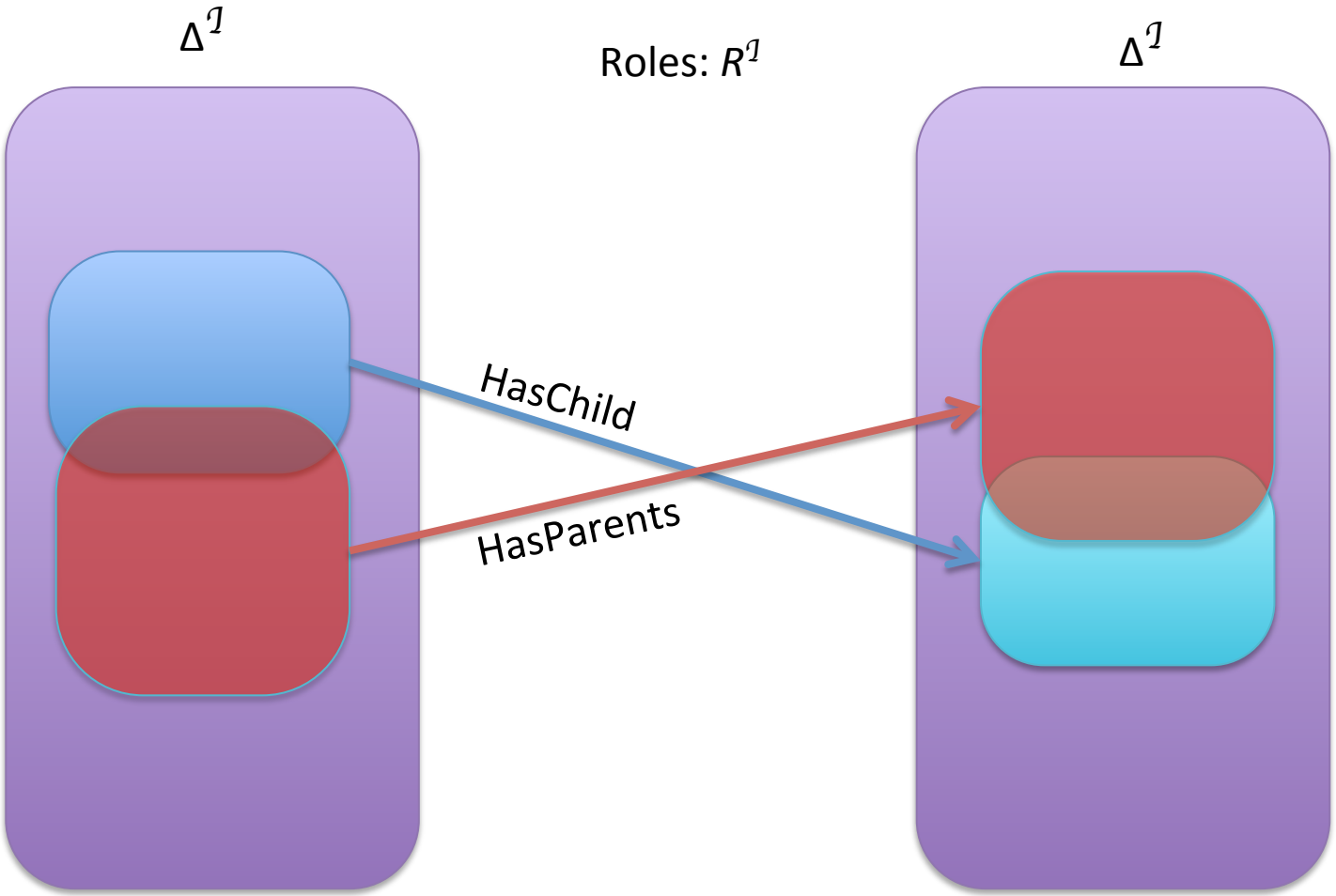
TBox and ABox

- TBox
  - Terminological Box
  - General knowledge (Vocabulary) about Classes &Roles
  - Corresponds to an ontology
- ABox
  - Assertional Box
  - Specific knowledge about individual objects
- Example:
  - Knowledge about family

Interpretation: Concept



Interpretation: Role

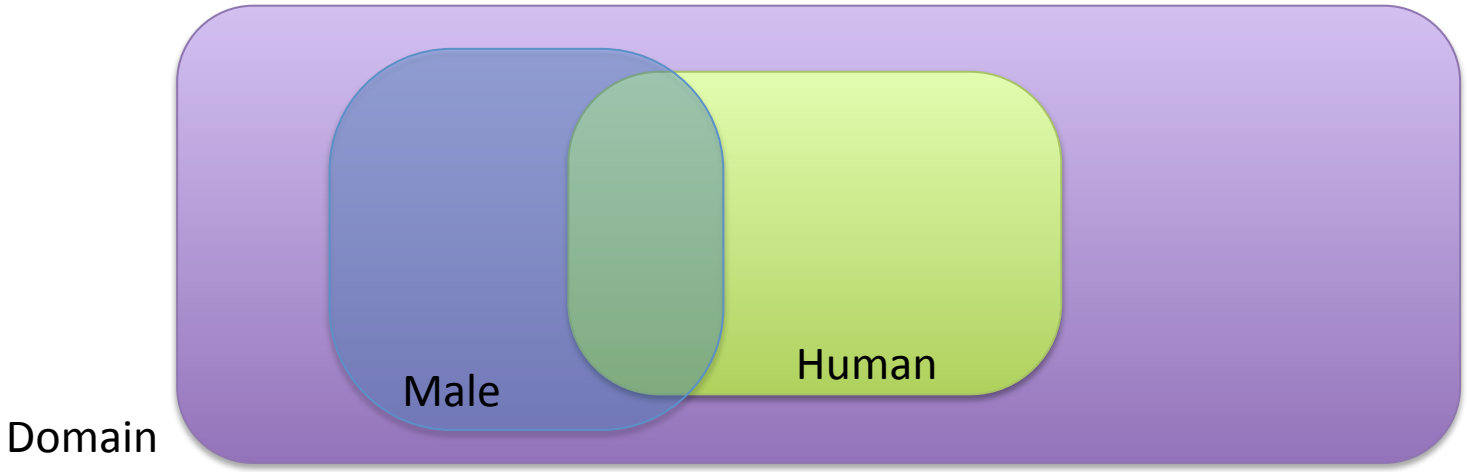


Concepts and Roles

- Domain
  - set of all the individuals of interest
- Concept
  - A subset of domain
  - Interpretation of a concept C: $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
- Role
  - Binary relation between sets
  - Interpretation of a rold R: $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Construction of Knowledge

- Atomic (primitive) concepts/roles
  - A concept that doesn't need a definition
  - Only interpretation is known
  - N_C = {Female, Male, Human}
  - N_R = {HasChild, HasParent, HasGrandParent, HasUncle}



Construction of Knowledge

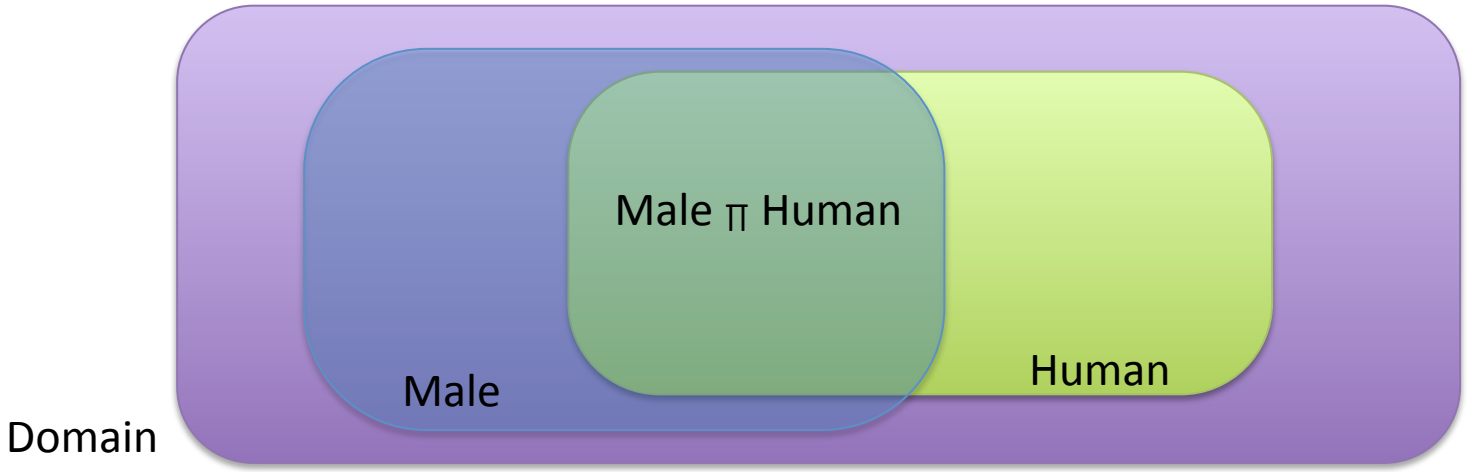
- Complex concepts/roles
  - Negation
  - Conjunction
  - Disconjunction
  - Full Existence Restriction
  - Universal Restriction
  - At Most Restriction
  - At Least Restriction
  - Qualified At Most Restriction
  - Qualified At Least Restriction
  - Inverse Of
  - Inclusion

Constructing Concepts

- Negation
  - ¬Male
  - Complement set: Male^{C}, or $\Delta^{\mathcal{I}}$-Male
  - Female ≡ ¬Male

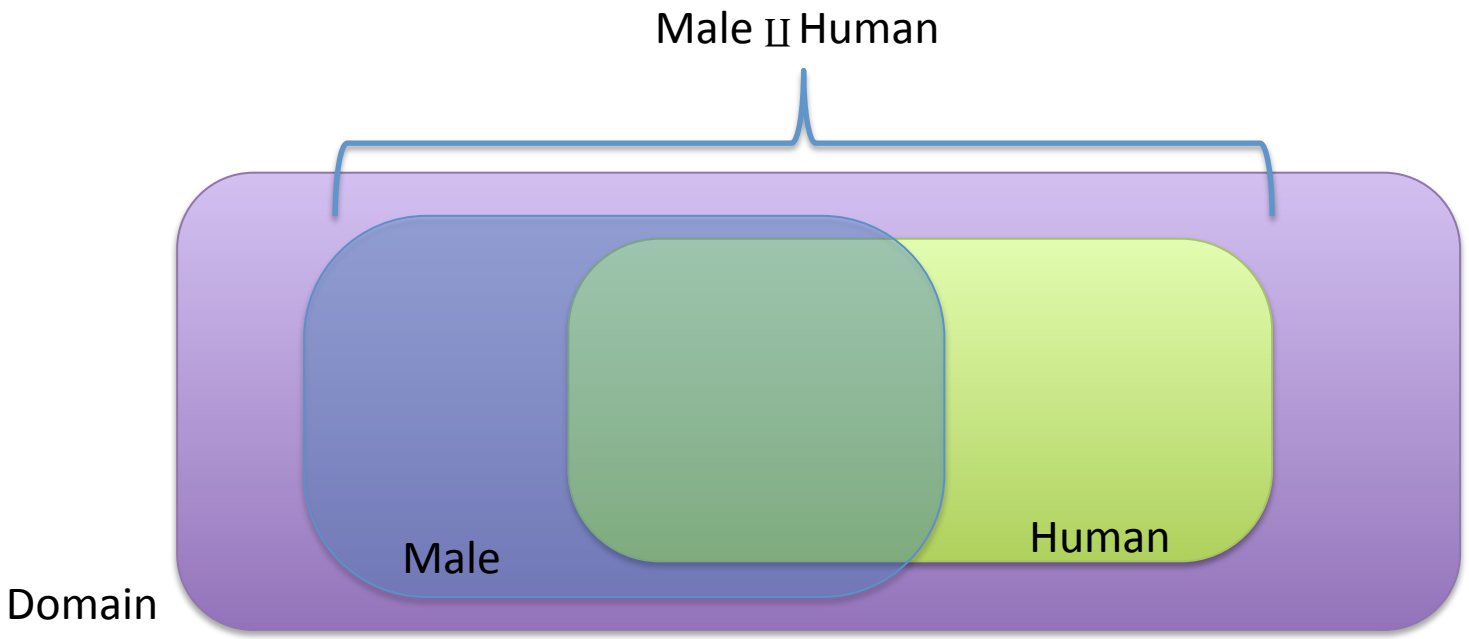
Constructing Concepts

- Conjunction
  - Male ∏ Human : $\text{Male}^{\mathcal{I}} \cap \text{Human}^{\mathcal{I}}$
  - Man≡ Male ∏ Human



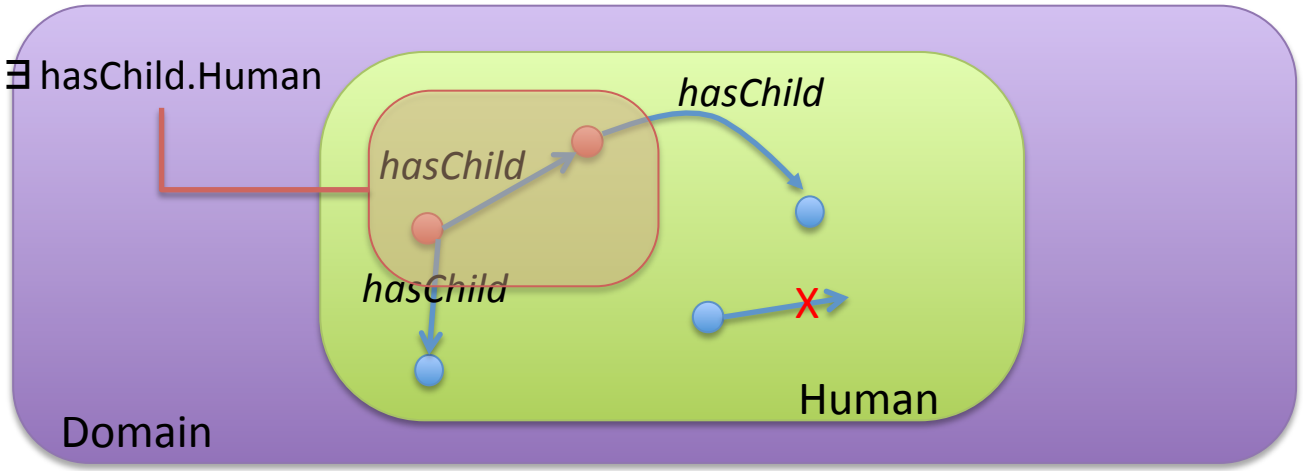
Constructing Concepts

- Disconjunction
  - Male $\amalg$ Human: Male$^{\mathcal{I}}$ $\cup$ Human$^{\mathcal{I}}$



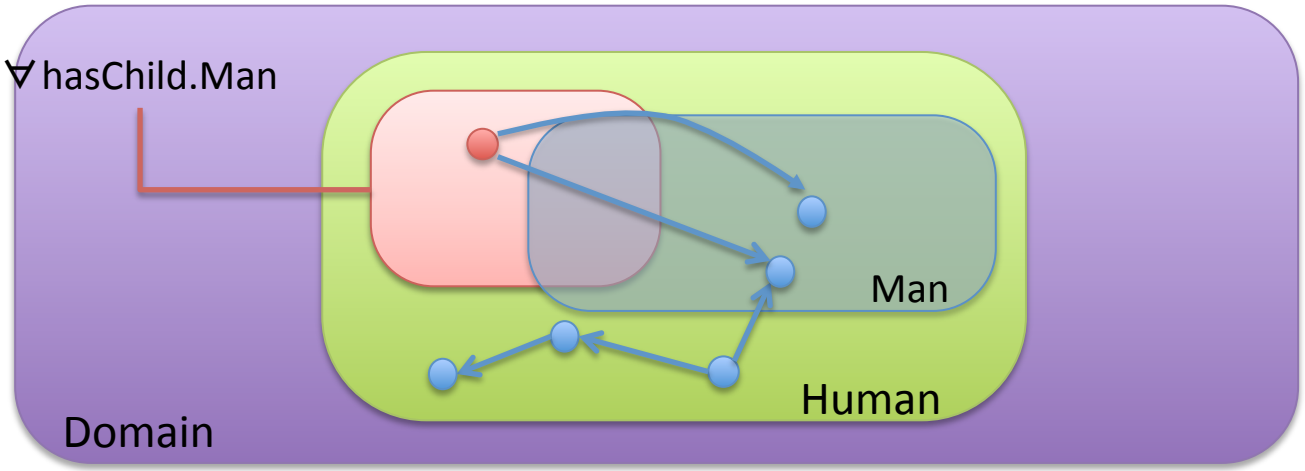
Constructing Concepts

- Full Existence Restriction
  - ∃ R.C: $\{a \in \Delta^{\mathcal{I}} \mid \exists b \in C^{\mathcal{I}},\ (a,b) \in R^{\mathcal{I}}\}$
    - Those that have a *b* in *C* such that *R(a,b)* is true
  - ∃ hasChild.Human
    - Those who have a child that is human



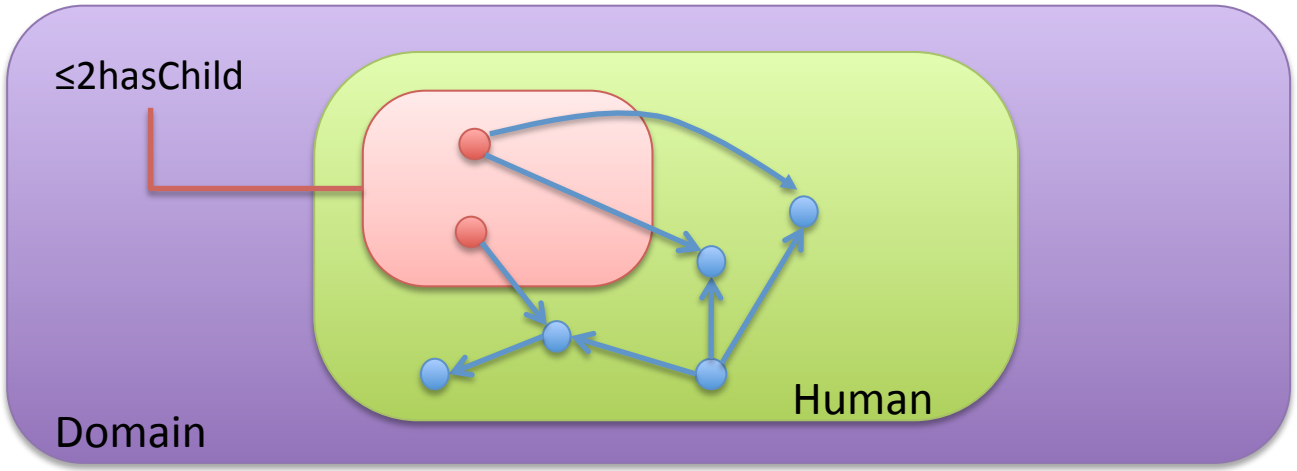
Constructing Concepts

- Universal Restriction
  - ∀R.C: $\{a \in \Delta^{\mathcal{I}} \mid \forall b\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
    - Those that have a relation R only with *b* that is in *C*
  - ∀hasChild.Man
    - Those who have children who are all sons



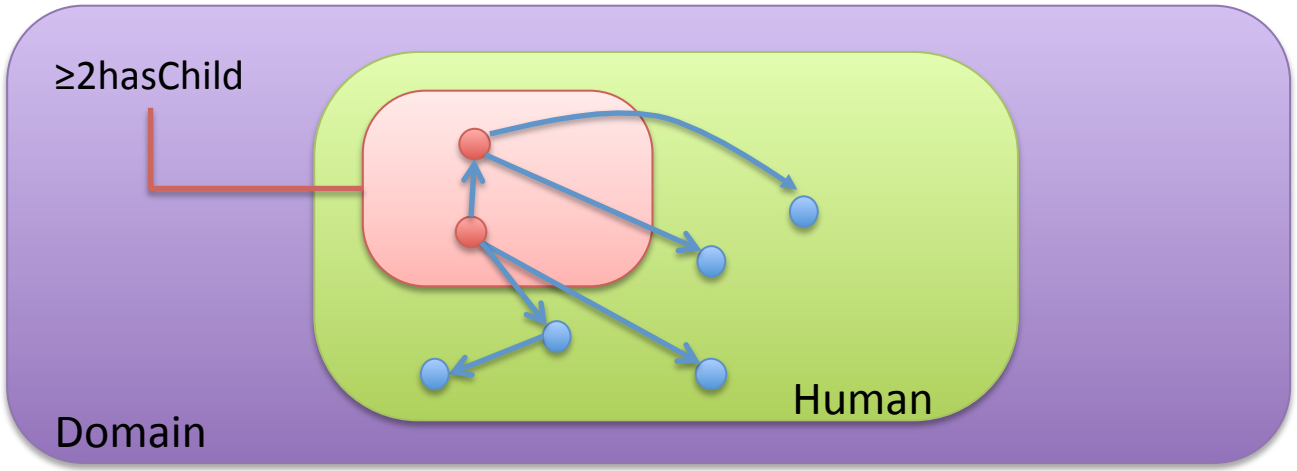
Constructing Concepts

- At Most Restriction
  - ≤nR: $\{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| \leq n\}$
    - Those that have relation *R* with at most *n* many *b*'s
  - ≤2hasChild
    - Those who have at most two children



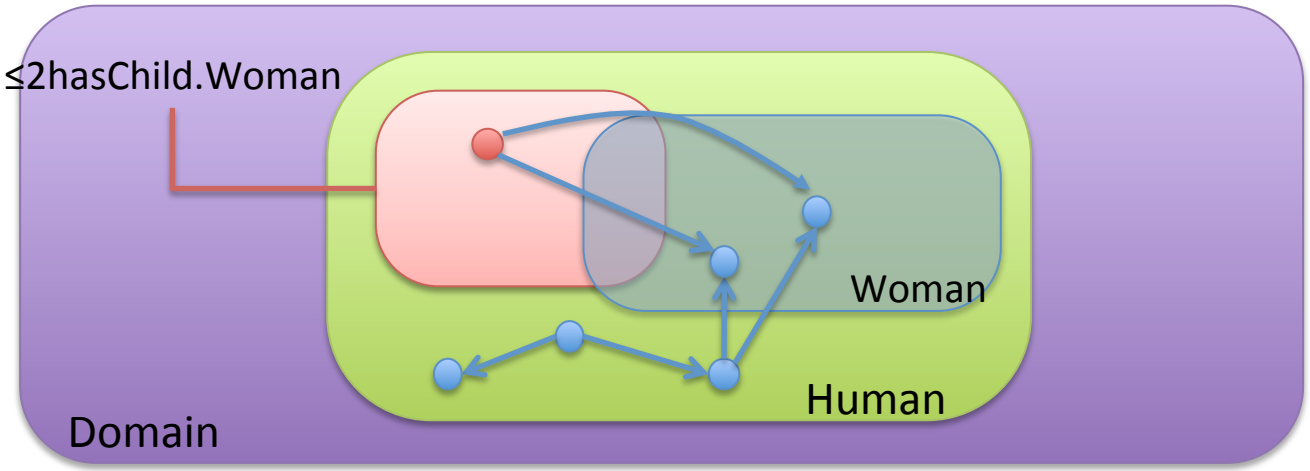
Constructing Concepts

- At Least Restriction
  - ≥nR: $\{a \in \Delta^{\mathcal{I}} \mid |\{b \in \Delta^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| \geq n\}$
    - Those that have relation *R* with at least *n* many *b*'s
  - ≥2hasChild
    - Those who have at least two children



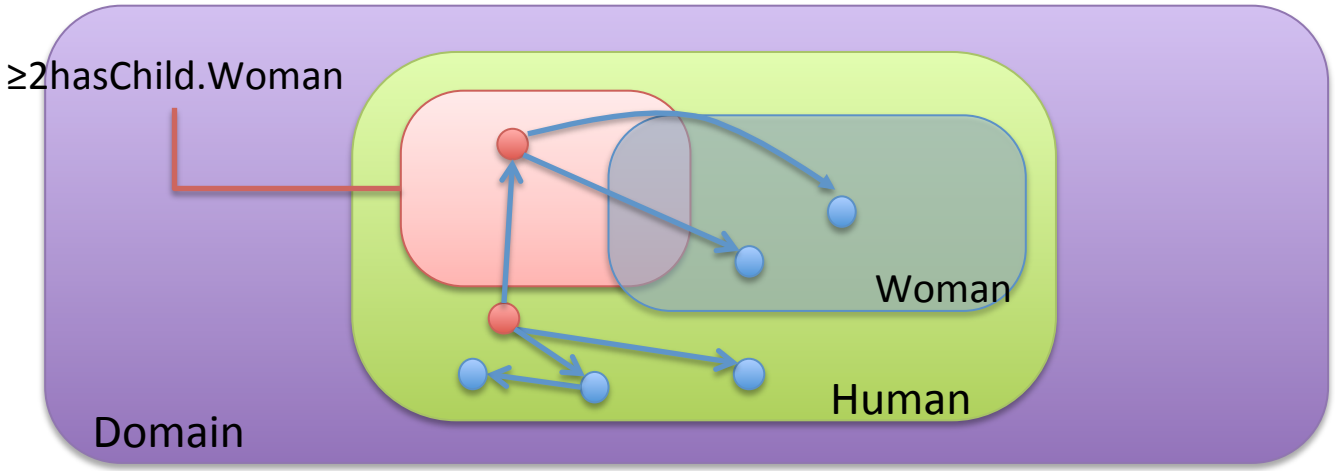
Constructing Concepts

- Qualified At Most Restriction
  - ≤nR.C: $\{a \in \Delta^{\mathcal{I}} \mid |\{b \in C^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| \leq n\}$
    - Those that have relation *R* with at most *n* many *b*'s in C
  - ≤2hasChild.Woman
    - Those who have at most two daughters



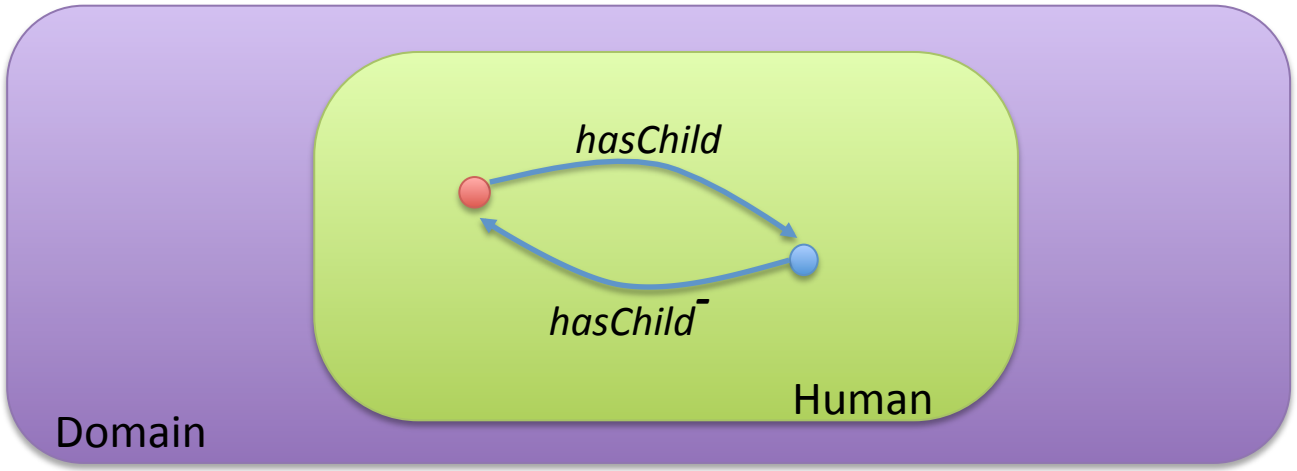
Constructing Concepts

- Qualified At Least Restriction
  - ≥nR.C : $\{a \in \Delta^{\mathcal{I}} \mid |\{b \in C^{\mathcal{I}} | (a,b) \in R^{\mathcal{I}}\}| \geq n\}$
    - Those that have relation *R* with at least *n* many *b*'s
  - ≥2hasChild.Woman
    - Those who have at least two daughters



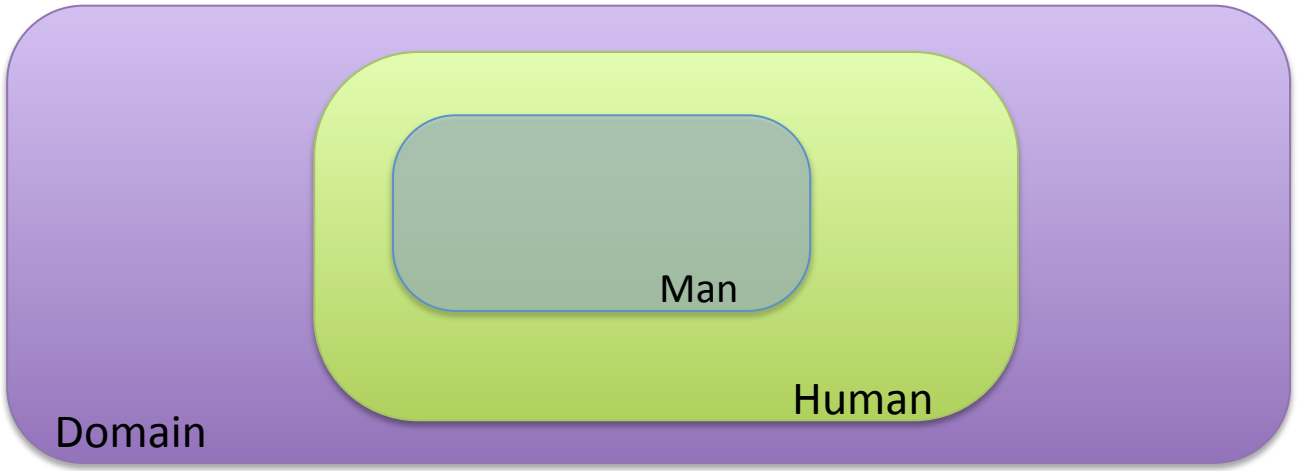
Constructing Concepts

- Inverse Role of
  - $R^{-} : \{(a, b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (b, a) \in R^{\mathcal{I}}\}$
    - Relation that is the inverse of relation *R*
  - $hasChild^{-}$
    - has a parent relation



Constructing Concepts

- Inclusion
  - C ⊆ D: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
  - C ⊇ D: $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$
  - Human ⊇ Man



TBox

- Knowledge base $\mathcal{K} = <\mathcal{T}, \mathcal{A}>$
- TBox $\mathcal{T}$
  - a set of concept/role definitions
    - C ≡ D (interpreted $C^{\mathcal{I}} = D^{\mathcal{I}}$)

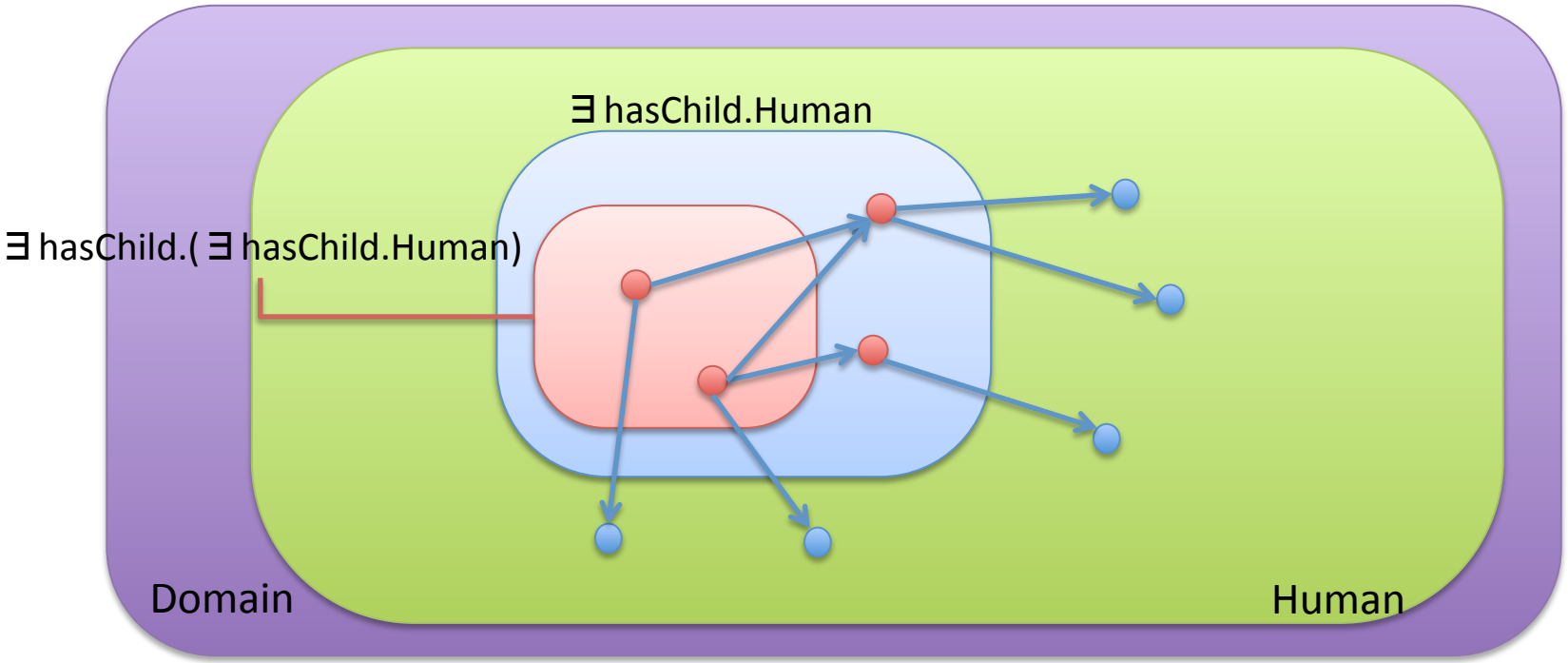
TBox: Example

- TBox $\mathcal{T}$ {
  - Female ≡ ¬Male
  - Woman ≡ Human ∏ Female
  - Man ≡ Human ∏ Male
  - Parent ≡ Human ∏ ∃HasChild.Human
  - Mother ≡ Woman ∏ Parent
  - Father ≡ Male ∏ Parent
  - Child ≡ Human ∏ ∃HasParent.Parent
  - GrandParent ≡ Parent ∏ ∃HasChild.(∃HasChild.Human)

  }

Constructing Concepts

- Nested Full Existence Restriction
  - ∃ hasChild.(∃ hasChild.Human)
    - Those who have a child that also has a child that is human



ABox

- Knowledge base $\mathcal{K} = <\mathcal{T}, \mathcal{A}>$
- ABox $\mathcal{A}$
  - a set of extensional assertions
    - *C(a)* means $a^{\mathcal{I}} \in C^{\mathcal{I}}$
    - *R(a, b)* means $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$

ABox: Example

- Abox $\mathcal{A}$ {
  - Woman(Claudia)
  - Father(Leonardo), Mother(Giovanna)
  - Child(Valentine)
  - HasParent(Claudia, Giovanna)
  - HasChild (Leonardo, Claudia)
  - HasSibling(Leonardo, Vito)
  - HasUncle(Valentine, Giovanna)

  }

Formal Language: OWL

- OWL (Ontology Web Language)
  - developed as a follow-on from RDF and RDFS and OIL, DAML, DAML+OIL.
- OWL is intended to be used over the World Wide Web,
- Supported by well-founded semantics of DLs
- together with a series of available automated reasoning services allowing to derive logical consequences from an ontology

DL in OWL

```
<owl:Class rdf:ID=”Father”>
        <owl:equivalentClass>
                <owl:Class>
                        <owl:intersectionOf rdf:parseType=”Collection”>
                                <owl:Class rdf:ID=”Man”/>
                                <owl:Class rdf:ID=”Parent”/>
                        </owl:intersectionOf>
                </owl:Class>
        </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID=”Female”>
        <owl:disjointWith>
                <owl:Class rdf:ID=”Male”/>
        </owl:disjointWith>
</owl:Class>
```

$\text{Father} \equiv \text{Male} \sqcap \text{Parent}$

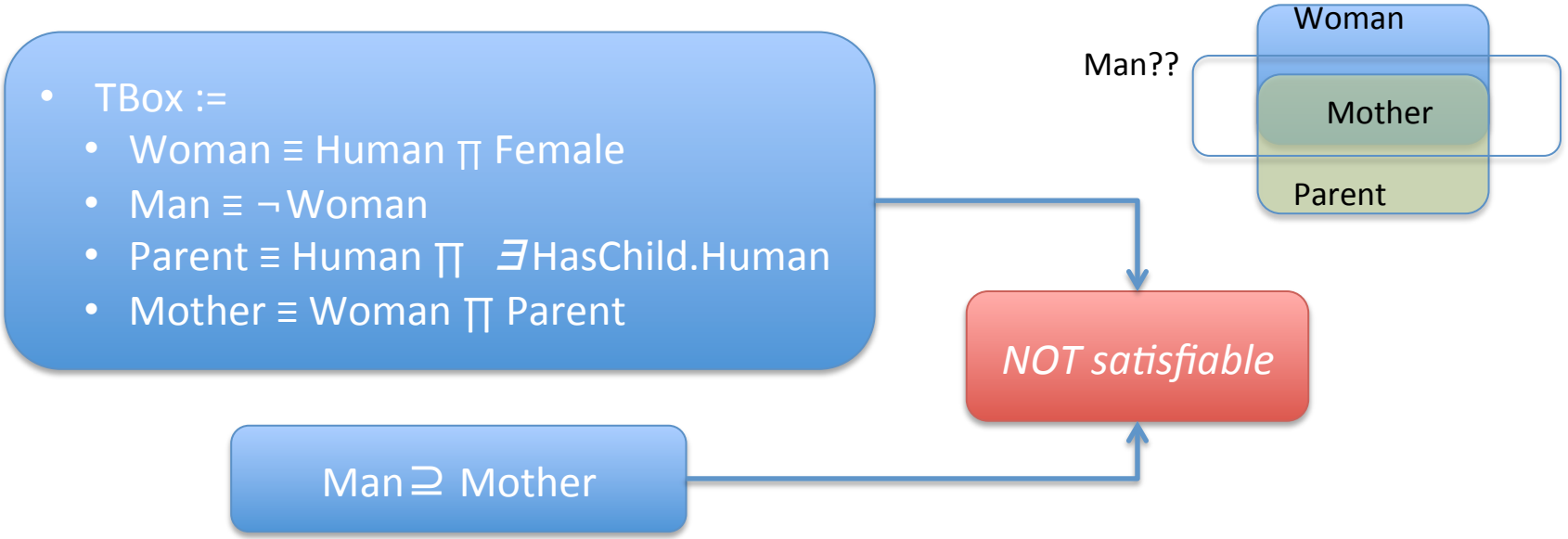
DisjointWith(Female, Male)

Inferences

- Tbox Standard Inference Services
  - Concept Satisfiability
  - Subsumption
- ABox Standard Inference Services
  - ABox Consistency w.r.t. the TBox
  - Instance Checking
  - Retrieval

TBox Inferences

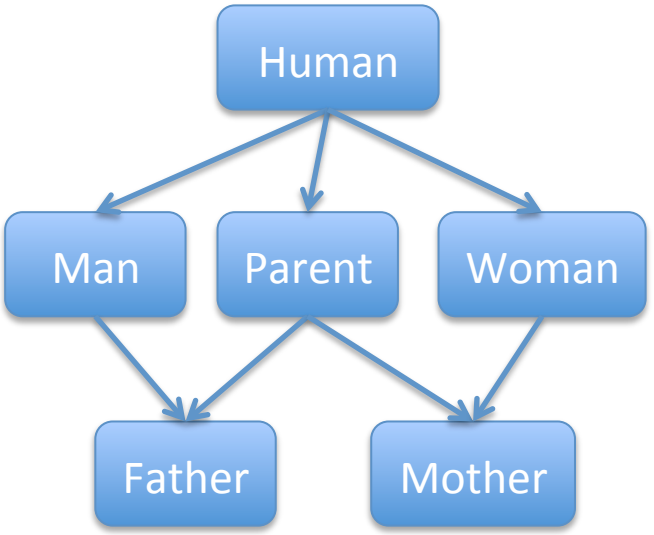
- Concept Satisfiability
  - Checks if a newly defined concept makes sense with respect to (w.r.t.) existing TBox, or contradictory



TBox Inferences

- Subsumption
  - Checks if concept *C* is more general than concep *D*

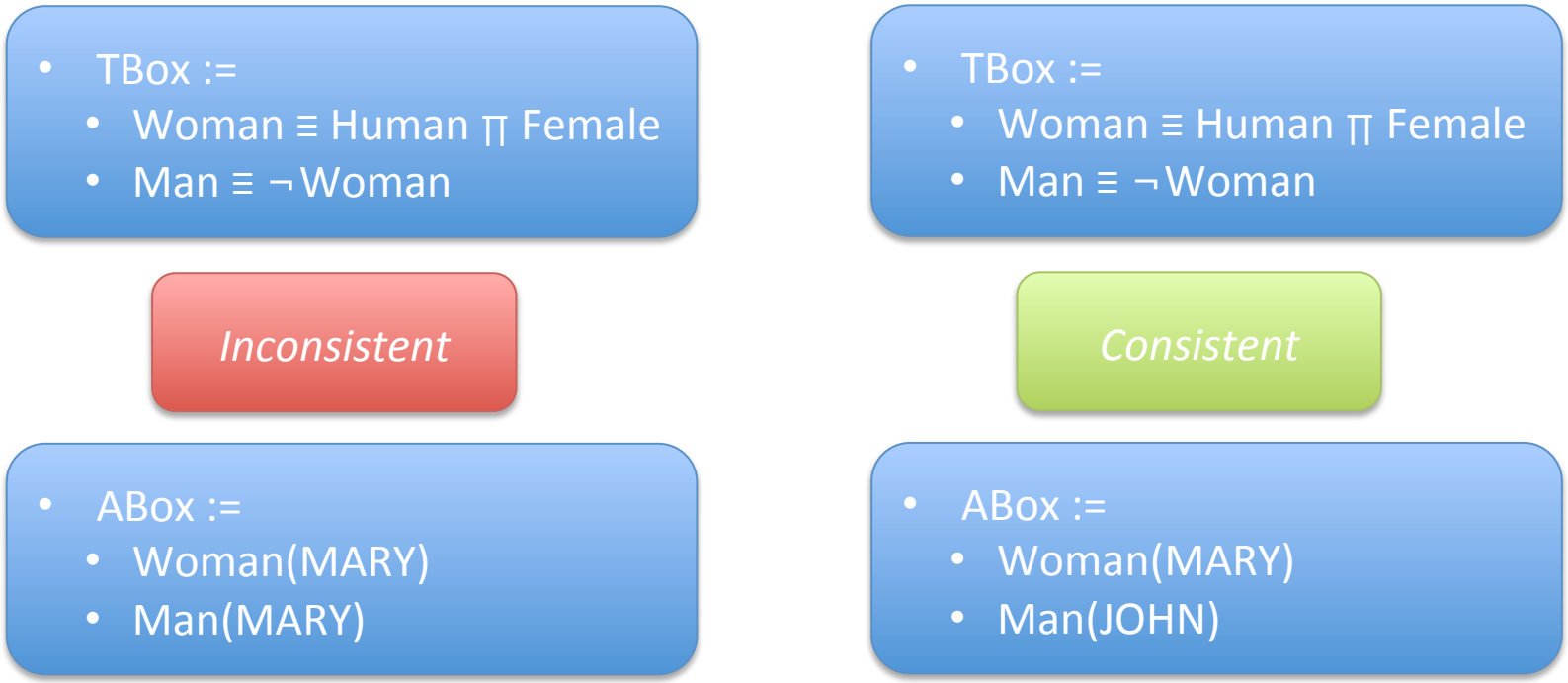
- TBox :=
  - Man ≡ Human ∏ Male
  - Woman ≡ ¬Man
  - Parent ≡ Human ∏ ∃HasChild.Human
  - Mother ≡ Woman ∏ Parent
  - Father ≡ Man ∏ Parent



Concept Hierarchy

ABox Inferences

- ABox Consistency w.r.t. TBox
  - Checks if a new assertion makes Abox inconsistent or not



ABox Inferences

- Instance Checking
  - Decide whether an individual is an instance of a concept or not

- TBox :=
  - Man ≡ Human ∏ Male
  - Woman ≡ ¬Man
  - Parent ≡ Human ∏ ∃HasChild.Human
  - Mother ≡ Woman ∏ Parent
  - Father ≡ Man ∏ Parent

- ABox :=
  - Woman(MARY)
  - Human(JOHN)
  - HasChile(MARY, JOHN)

Mother(MARY)?

YES

Father(MARY)?

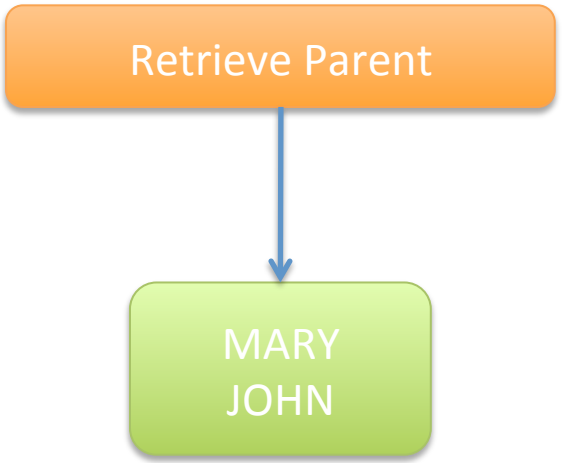
NO

ABox Inferences

- Retrieval
  - Finds all individual instances of a concept

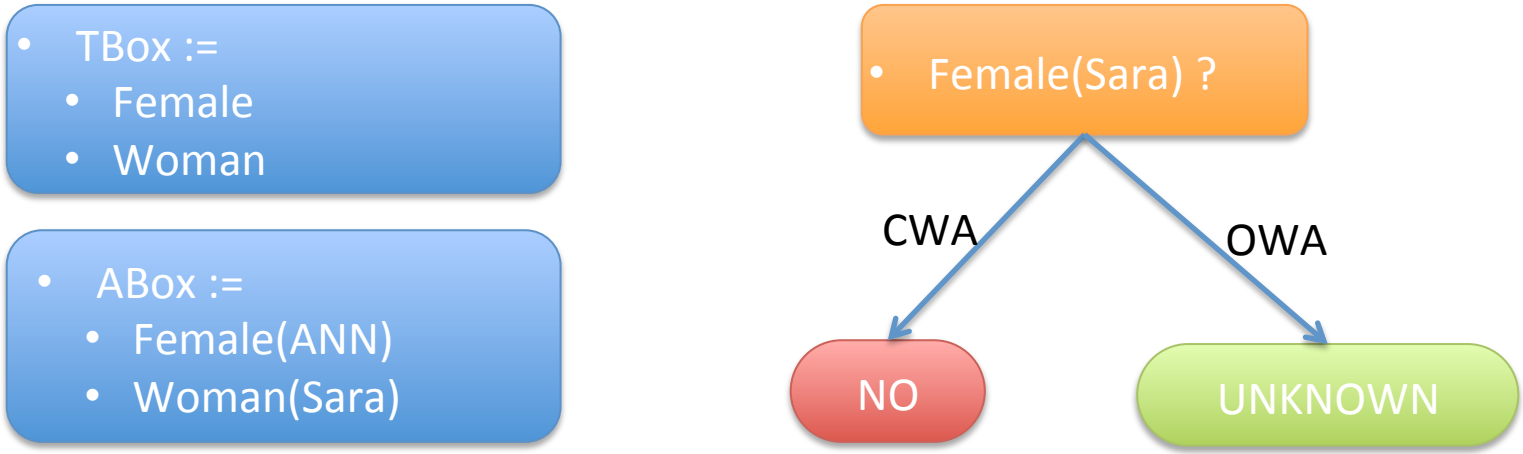
- TBox :=
  - Man ≡ Human ∏ Male
  - Woman ≡ ¬Man
  - Parent ≡ Human ∏ ∃HasChild.Human
  - Mother ≡ Woman ∏ Parent
  - Father ≡ Man ∏ Parent

- ABox :=
  - Woman(MARY)
  - Human(JOHN)
  - HasChile(MARY, JOHN)



OWA vs. CWA

- Closed World Assumption
  - Absence of information is interpreted as negative information
  - Traditional Database concept
- Open World Assumption
  - Absence of information is interpreted as unknown inofrmation
  - Knowledge



Tools for building and managing ontologies

- **Protege**: free, open source ontology editor and knowledge-base framework
- **Chimaera**: system for creating and maintaining distributed ontologies on the web. Major supported functions: merging multiple ontologies; diagnosing of one or more ontologies.
- **Ontolingua**: distributed collaborative environment to browse, create, edit, modify, and use ontologies.
- **OntoEdit**: Engineering Environment for the development and maintenance of ontologies using graphical means.
- **WebOnto**: Java applet coupled with a customised web server allowing to browse and edit knowledge models over the web.
- **KAON**: open-source infrastructure for ontology creation and management, and providing a framework for building ontology-based applications.

Conclusions

- Summarizing
  - An ontology is a formal conceptualization of a domain that is shared and reused across domains, tasks and group of people
  - Ontologies are used in different fields for:
    - Constituting a community reference
    - Sharing consistent understanding of what information means
    - Making possible interoperability between systems
    - Making the Web machine-readable and processable besides of human-readable (Semantic Web)
- Line of research:
  - Ontology construction is the result of a complex process of knowledge acquisition ⇒ (semi)-automatic tools for building ontologies are necessary
  - Machine learning methods can be useful for accomplishing such a goal