OWL: Web Ontology Language and "RDFS-Plus"

Unit 6 Preliminary

Most of this first section of content slides on OWL are for your general awareness. We are not formally studying or using these aspects of OWL in this course.
The two exceptions are (1) the concept of *transitivity* and *transitive properties*, and the use of *owl:sameAs*, because these comes up in SKOS

Slides 21 and following are completely optional. They show a few additional aspects of OWL for those who might happen to be interested. There's a lot more to OWL beyond what's covered in these slides, but this is the extent I'm showing in this course.

OWL goes beyond RDFS

- **RDFS** is a base-level ontology language and deals primarily with:
  - Classes, subclasses, properties, subproperties, domain and range
  - *This is primarily what we've covered so far in this course*
  - These are the most important and widely used ontology constructs in most cultural heritage ontologies
- **OWL** is a full-fledged ontology language
  - Not a direct extension of RDFS, but does builds on it
  - *Since we've been using Protégé in this course, we've already been working with some OWL elements*
  - Allows for much fuller reasoning and inferencing by enabling specifications for:
    - Richer typing of properties (object vs. datatype, specific datatypes)
    - Characteristics of properties / special properties (transitive, symmetric, functional, inverse functional)
    - Relations between classes (disjointness, equivalence, union, intersection)
    - Cardinality (minimum, maximum, exact number)
    - Equality and difference (same as, different from)
    - Enumerated classes

OWL elements (1)

- **RDF and RDFS elements used in OWL**
  - rdf:type
  - rdf:Property
  - rdfs:subClassOf
  - rdfs:subPropertyOf
  - rdfs:domain
  - rdfs:range
- **Basic OWL elements**
  - owl:Thing
  - owl:Class
  - owl:NamedIndividual
  - owl:ObjectProperty
  - owl:DatatypeProperty
- **Property characteristics**
  - **owl:inverseOf**
  - **owl:TransitiveProperty**
  - **owl:SymmetricProperty**
  - owl:FunctionalProperty
  - owl:InverseFunctionalProperty
- **Cardinality restrictions**
  - owl:minCardinality
  - owl:maxCardinality
  - owl:cardinality
- **Datatype specification**
  - xsd:datatypes

OWL elements (2)

- **Equality/inequality**
  - owl:equivalentClass
  - owl:equivalentProperty
  - owl:sameAs
  - owl:differentFrom
  - owl:AllDifferent
  - owl:distinctMembers
- **Property restrictions**
  - owl:Restriction
  - owl:onProperty
  - owl:allValuesFrom
  - owl:someValuesFrom
- **Class intersection**
  - owl:intersectionOf

OWL DL & OWL Full:

- **Class axioms**
  - owl:one of, dataRange
  - owl:disjointWith
  - owl:equivalentClass
    - (applied to class expressions)
  - rdfs:subClassOf
    - (applied to class expressions)
- **Boolean combinations of class expressions**
  - owl:unionOf
  - owl:complementOf
  - owl:intersectionOf
- **Property information**
  - owl:hasValue

OWL elements (3)

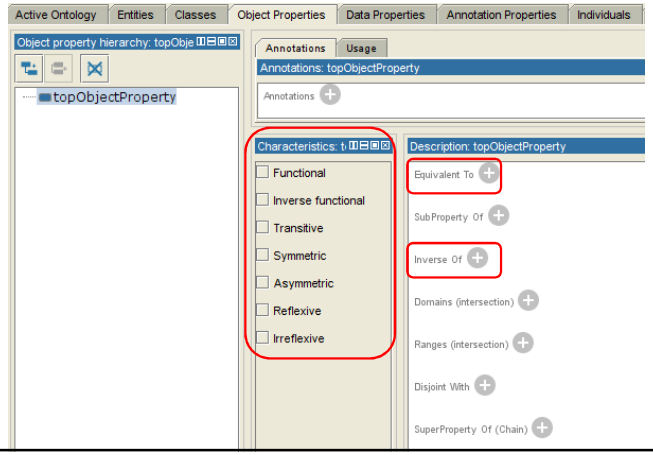
- **Ontology header information**
  - owl:Ontology
  - owl:imports
- **Ontology versioning information**
  - owl:versionInfo
  - owl:priorVersion
  - owl:backwardCompatibleWith
  - owl:incompatibleWith
  - owl:DeprecatedClass
  - owl:DeprecatedProperty

- OWL versions
  - OWL 1
  - OWL 2
- OWL 1 sublanguages
  - OWL Lite
  - OWL DL
  - OWL Full

"RDFS-Plus"

- **A selected subset of OWL elements & constructs**
- Selected to satisfy certain goals:
  - Provide a gentle introduction to OWL by adding selected constructs to RDFS without full complexity of OWL
  - Give background for real-world applications that rarely use RDFS by itself, but often use these selected OWL elements with RDFS but without full OWL (*SKOS is a prime example*)
  - Learn this subset of OWL elements which can be implemented using a wide variety of inferencing technologies
- Focus on: (a) special property characteristics and (b) equivalence of classes, properties, and individuals

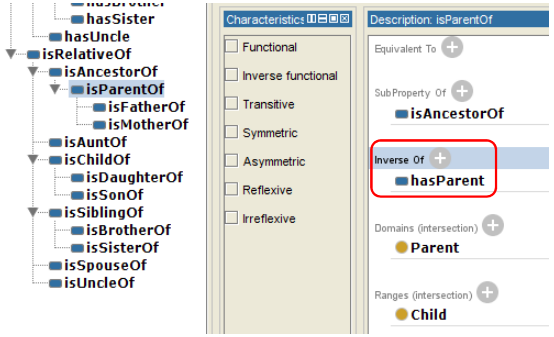
Property characteristics in Protégé



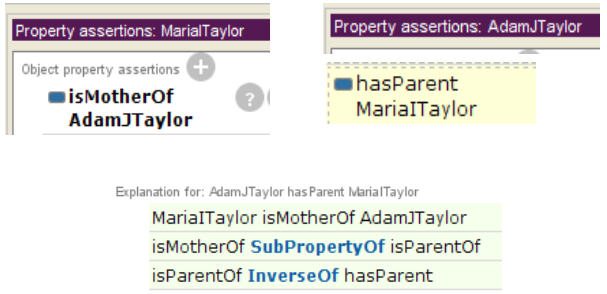
Inverse properties (owl:inverseOf)

- **Statements:**
  - :isParentOf owl:inverseOf :hasParent
  - :MariaITaylor :isParentOf :AdamJTaylor
  - :isParentOf rdfs:subPropertyOf :hasAncestor
- **Inferences:**
  - :hasParent owl:inverseOf :isParentOf
  - :AdamJTaylor :hasParent :MariaITaylor
  - :AdamJTaylor :hasAncestor :MariaITaylor

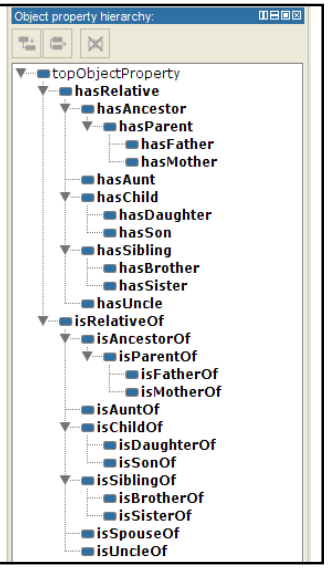
Inverse properties in Protégé



Inferences in Protégé



Option: establish / assert all possible inverse properties

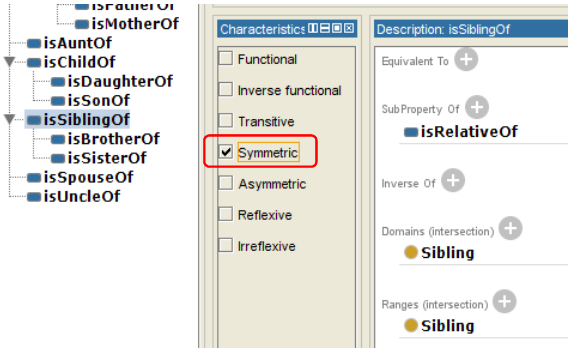


Symmetric properties
(owl:SymmetricProperty)

- If a property is symmetric, then if the pair (x,y) is an instance of the symmetric property P, then the pair (y,x) is also an instance of P.
  - For example, friend may be stated to be a symmetric property. Then a reasoner that is given that Frank is a friend of Deborah can deduce that Deborah is a friend of Frank.

  (Source: W3C OWL Web Ontology Language Overview: http://www.w3.org/TR/owl-features/)
- **Statements:**
  - :hasSibling a :owl:SymmetricProperty
  - :SofiaMTaylor :hasSibling :AdamJTaylor
- **Inferences:**
  - :AdamJTaylor :hasSibling :SofiaMTaylor

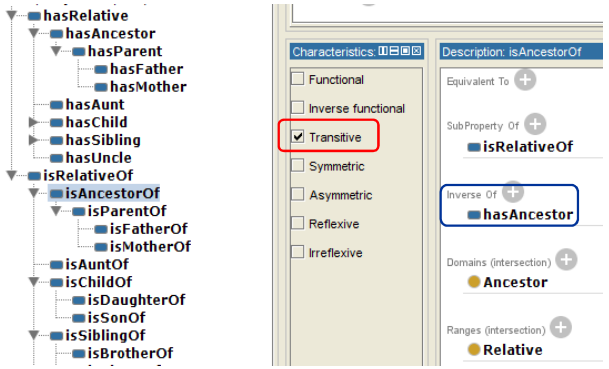
Symmetric properties in Protégé



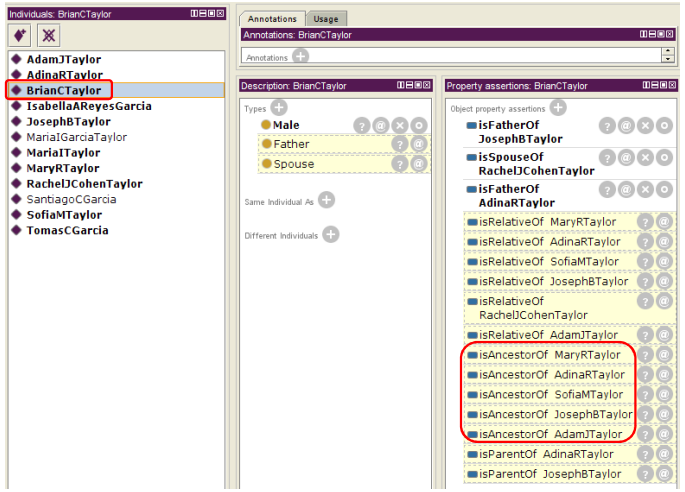
Transitive properties
(owl:TransitiveProperty)

- If a property is transitive, then if the pair (x,y) is an instance of the transitive property P, and the pair (y,z) is an instance of P, then the pair (x,z) is also an instance of P.
  - For example, if ancestor is stated to be transitive, and if Sara is an ancestor of Louise (i.e., (Sara,Louise) is an instance of the property ancestor) and Louise is an ancestor of Deborah (i.e., (Louise,Deborah) is an instance of the property ancestor), then a reasoner can deduce that Sara is an ancestor of Deborah (i.e., (Sara,Deborah) is an instance of the property ancestor).
- **Statements:**
  - :isAncestorOf a owl:TransitiveProperty
  - :BrianCTaylor :isAncestorOf :JosephBTaylor
  - :JosephBTaylor :isAncestorOf :SofiaMTaylor
- **Inferences:**
  - :BrianCTaylor :isAncestorOf :SofiaMTaylor

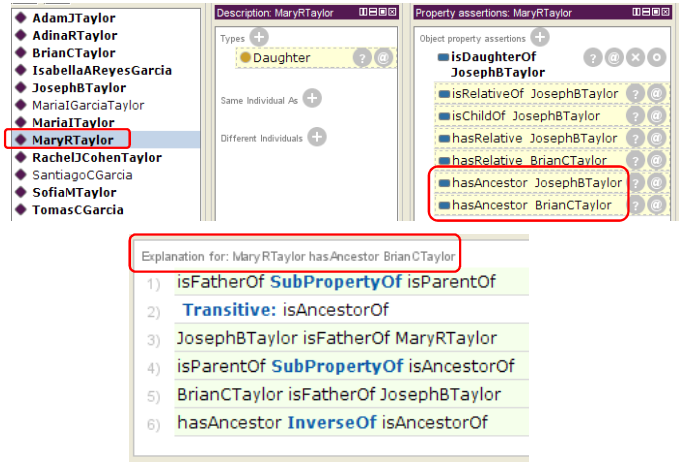
Transitive properties in Protégé



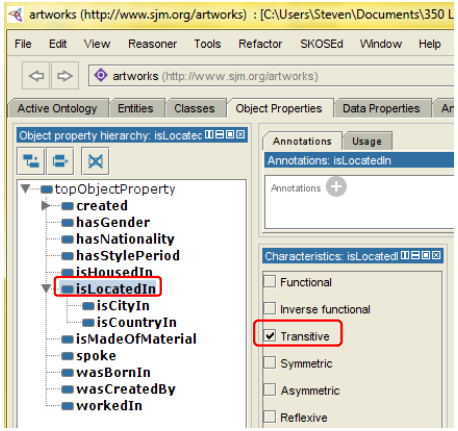
Inferences in Protégé



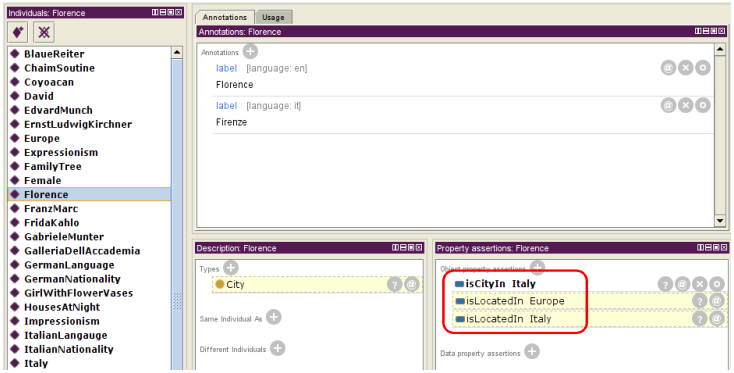
Inferences in Protégé



Transitive property in artworks ontology



Resulting inferences



Same individual (*owl:sameAs*)

- **Two different URIs identify the same entity** (person, corporate body, place, event, object, concept, etc.)
- *Example: t*wo URIs in my knowledgebase identify the same individual:
  - :MariaITaylor **owl:sameAs** :MariaIGarciaTaylor
- Same individual in two different vocabularies:
  - Frida Kahlo in LC NAF and VIAF linked data authorities:
    - http://id.loc.gov/authorities/names/n82031966 **owl:sameAs** http://viaf.org/viaf/110981647/#skos:Concept

OPTIONAL:

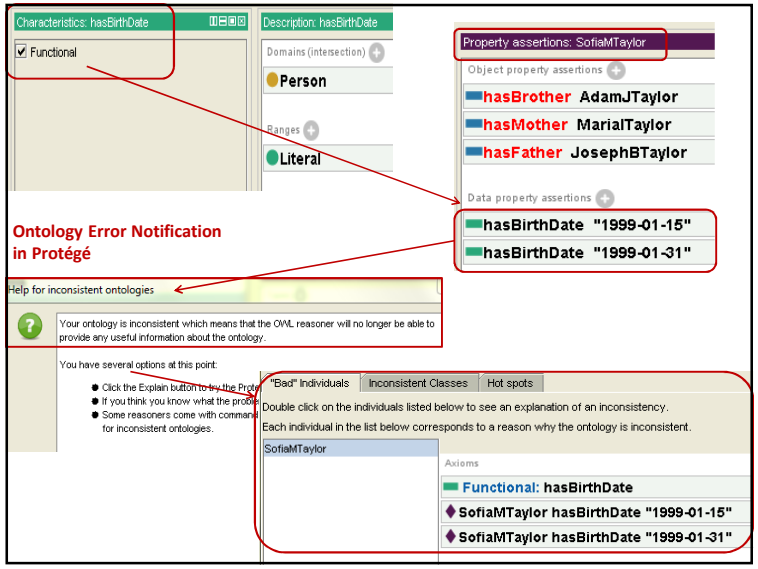
More on OWL / "RDFS-Plus"

Unit 6 Optional Supplement*:*

for those interested in being exposed to yet a bit more about OWL beyond what we'll actually work with in this course

Functional properties
(owl:FunctionalProperty)

- If a property is a FunctionalProperty, then it has no more than one value for each individual (it may have no values for an individual). This characteristic has been referred to as having a unique property. FunctionalProperty is shorthand for stating that the property's minimum cardinality is zero and its maximum cardinality is 1.
  - For example, *hasPrimaryEmployer* may be stated to be a FunctionalProperty. From this a reasoner may deduce that no individual may have more than one primary employer. This does not imply that every Person must have at least one primary employer however.
- **Statements:**
  - :hasBirthdate a owl:FunctionalProperty
  - :SofiaMTaylor :hasBirthDate "1999-01-15"
  - :SofiaMTaylor :hasBirthDate "1999-01-05"
- **Inferences:**
  - *Error in ontology:* an individual may have only one unique value for the hasBirthDate property



Functional properties
(owl:FunctionalProperty)

- **Statements:**
  - :hasMother a owl:FunctionalProperty
  - :SofiaMTaylor :hasMother :MarialTaylor[URI]
  - :SofiaMTaylor :hasMother :MarialGarciaTaylor[URI]
- **Inferences:**
  - SofiaMTaylor may have only one individual who is her mother
  - In this case, these two URIs refer to the same individual
  - Because of the non-unique names assumption, there can be no inference that :MarialTaylor and :MarialGarciaTaylor are different individuals
  - In this hypothetical case, they are in fact the same individual with two different URIs

Semantic Web and OWL assumptions

- **AAA slogan**
  - Anyone can say Anything about Any topic
- **Open World Assumption**
  - **Closed world:** databases with tightly controlled content; all relevant information about an entity is included; inferences can be made accordingly
  - **Open world:** uncontrolled open data; someone can always contribute something new about an entity
    - Machine inferencing must take this into account: "we may draw no conclusions that rely on assuming that the information available at any one point is all the information available"
- **Nonunique Naming Assumption**
  - **Unique names:** may hold in controlled databases or triple stores
  - **Nonunique names:** in an open world context, different Web authors will use different URIs for the same individual entity
    - Machine inferencing cannot assume that two entities with different URIs are different individuals

Source: Allemang and Hendler p. 6-10

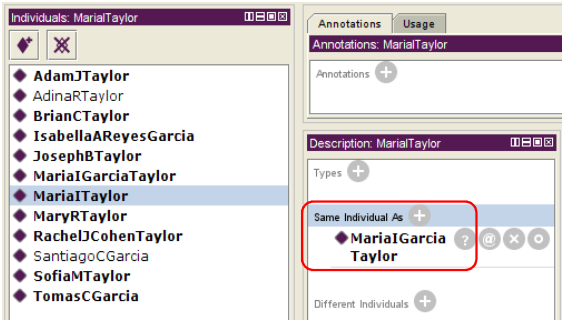
"Closing the world" in OWL

- Ways to put a limitation on the AAA slogan, Open World assumption, and Nonunique names assumption, to assert that certain parts of the world are closed
  - owl:sameAs, owl:one of, owl:differentFrom, owl:AllDifferent, owl:distinctMembers, owl:disjointWith

Same individual (*owl:sameAs*)

- Two different URIs identify the same entity (person, corporate body, place, event, object, concept, etc.)
- *Example: t*wo URIs in my knowledgebase identify the same individual:
  - :MarialTaylor **owl:sameAs** :MarialGarciaTaylor

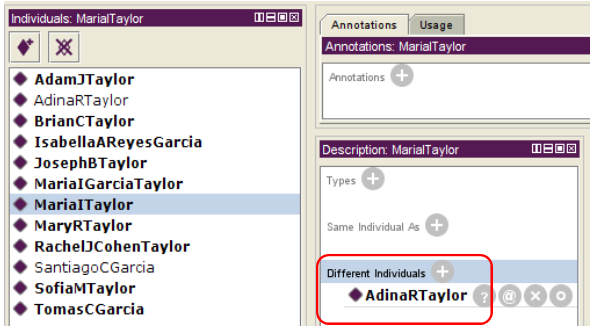
Same individuals in Protégé



Different individuals (*owl:differentFrom*)

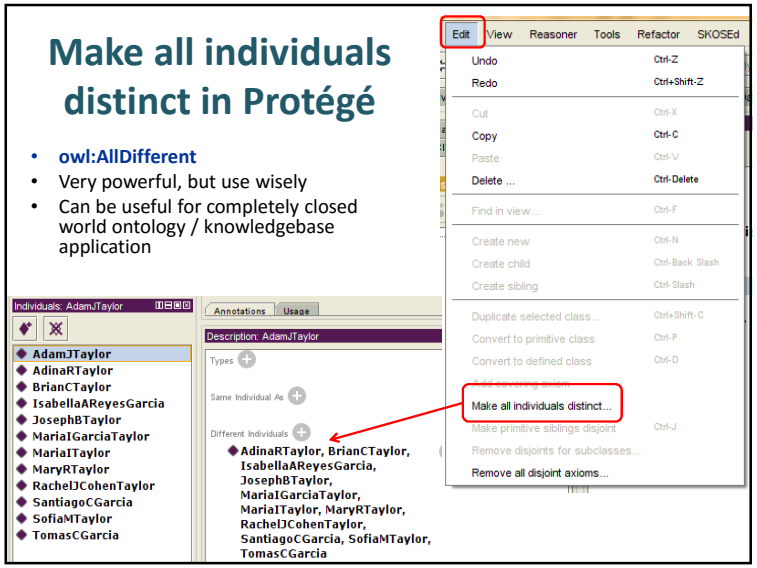
- **Statements:**
  - :hasMother a owl:FunctionalProperty
  - :SofiaMTaylor :hasMother :MarialTaylor
  - :SofiaMTaylor :hasMother :AdinaRTaylor
  - :MarialTaylor owl:differentFrom :AdinaRTaylor
- **Inferences:**
  - *Error in ontology:* one individual may have only one other individual as the value of the :hasMother property, and these two have now been asserted to be different individuals

Different individuals in Protégé

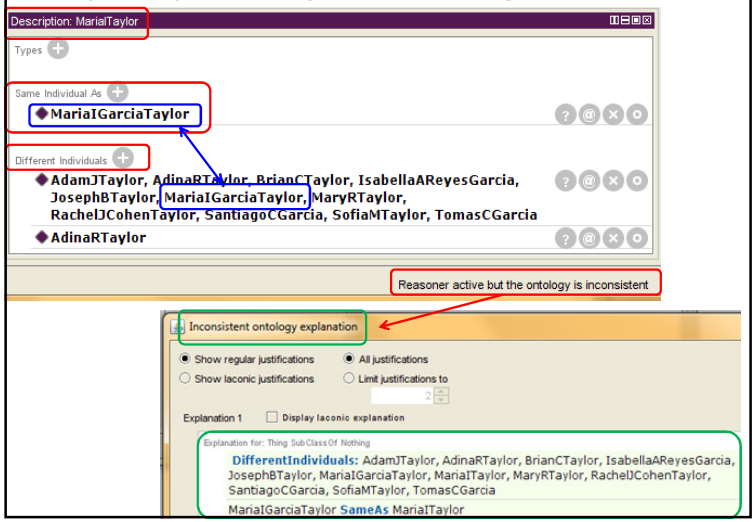


Make all individuals distinct in Protégé

- **owl:AllDifferent**
- Very powerful, but use wisely
- Can be useful for completely closed world ontology / knowledgebase application



Example of potential problems using owl:AllDifferent



Turtle output

```
###  http://www.mydomain.org/familyrelations#MariaIGarciaTaylor

:MariaIGarciaTaylor rdf:type owl:NamedIndividual ;
                    owl:sameAs :MariaITaylor .

#################################################
#
#    General axioms
#
#################################################


[ rdf:type owl:AllDifferent ;
  owl:distinctMembers ( :AdamJTaylor
                        :AdinaRTaylor
                        :BrianCTaylor
                        :IsabellaAReyesGarcia
                        :JosephBTaylor
                        :MariaIGarciaTaylor
                        :MariaITaylor
                        :MaryRTaylor
                        :RachelJCohenTaylor
                        :SantiagoCGarcia
                        :SofiaMTaylor
                        :TomasCGarcia
                      )
] .
```

Turtle output

```
:hasAncestor rdf:type owl:ObjectProperty ,
                      owl:TransitiveProperty ;
             rdfs:subPropertyOf :hasRelative ;
             owl:inverseOf :isAncestorOf .

:isSiblingOf rdf:type owl:ObjectProperty ,
                      owl:SymmetricProperty ;
             rdfs:range :Sibling ;
             rdfs:domain :Sibling ;
             rdfs:subPropertyOf :isRelativeOf .

:hasBirthDate rdf:type owl:DatatypeProperty ,
                       owl:FunctionalProperty ;
              rdfs:domain :Person ;
              rdfs:range xsd:string .
```

Inverse Functional properties
(owl:InverseFunctionalProperty)

- If a property is inverse functional then the inverse of the property is functional. Thus the inverse of the property has at most one value for each individual. This characteristic has also been referred to as an unambiguous property.
  - For example, hasUSSocialSecurityNumber (a unique identifier for United States residents) may be stated to be inverse functional (or unambiguous). The inverse of this property (which may be referred to as isTheSocialSecurityNumberFor) has at most one value for any individual in the class of social security numbers. Thus any one person's social security number is the only value for their isTheSocialSecurityNumberFor property.
  - From this a reasoner can deduce that no two different individual instances of Person have the identical US Social Security Number.
  - Also, a reasoner can deduce that if two instances of Person have the same social security number, then those two instances refer to the same individual.

Inverse functional properties
(owl:InverseFunctionalProperty)

- **Statements:**
  - :hasUSPassportNumber a owl:InverseFunctionalProperty
  - :MariaITaylor :hasUSPassportNumber "12345678"
  - :JosephBTaylor :hasUSPassportNumber "12345678"
- **Inferences:**
  - Only one individual may have any given passport number (for a specific country)
  - *No error in the inferencing*, however, because the non-unique names assumption does not allow the conclusion that the URIs for :MariaITaylor and :JosephBTaylor are for different individuals

Different individuals (owl:differentFrom)

- **Statements:**
  - :hasUSPassportNumber a owl:InverseFunctionalProperty
  - :MarialTaylor :hasUSPassportNumber "12345678"
  - :JosephBTaylor :hasUSPassportNumber "12345678"
  - :MarialTaylor owl:differentFrom :JosephBTaylor
- **Inferences:**
  - *Error in ontology:* different individuals cannot have the same value for the hasUSPassportNumber property

Another inverse functional property example

- A health care network assigns unique patient IDs
- Different doctors offices, clinics, and hospitals in the network are beginning to share medical records within new semantic environment
- **Statements:**
  - **:hasPatientID a owl:InverseFunctionalProperty**
  - *Doctor Lee's office:*
    - :Linda-A-Brown-URI :hadAppointmentDate "2013-02-06"
    - :Linda-A-Brown-URI :treatedFor :xyzDisease
    - **:Linda-A-Brown-URI :hasPatientID "987654"**
  - Hilltop Hospital
    - :Linda-A-Porter-URI :admittedOn "2013-05-10"
    - :Linda-A-Porter-URI :discharedOn "2013-05-12"
    - :Linda-A-Porter-URI :treatedFor :abcDisease
    - **:Linda-A-Porter-URI :hasPatientID "987654"**
- **Inferences:**
  - :Linda-A-Brown-URI and :Linda-A-Porter-URI are the same individual
  - This individual has been treated for :xyzDisease and :abcDisease
  - This individual had an appointment with Dr. Lee on February 6, 2013 and was in Hilltop Hospital May 10-12, 2013
  - etc.

Functional & Inverse Functional properties: another way to think about them

- **Functional:**
  - **Any** specific individual subject of the property may have **only one value** as the object of that property
    - *Example:* any specific person may have only one value for the **:hasBirthdate** property
    - :MariaGarciaTaylor can have only one birthdate, but many other individuals can also have the same birthdate
    - If it is asserted that :MariaGarciaTaylor :hasBirthdate "1976-12-03" and :hasBirthdate "1976-02-12" there is an error
- **Inverse Functional:**
  - **Only one** specific individual subject of the property may have **the same specific value** of that property
    - *Example:* only one specific person may have the same value for the **:hasSocialSecurityNumber** property
    - Only one individual person may have the social security number 111-22-3333
    - If both :MariaGarcia and :MariaTaylor are asserted to have the same SS number, they must be the same individual

Equivalence and Difference

- Especially useful in a distributed knowledge environment like the open web and for linking or merging RDF individual and/or ontology data
- Equivalent classes
  - owl:EquivalentClass
- Equivalent properties
  - owl:equivalentProperty
- Same individual
  - owl:sameAs
- Different individuals
  - owl:differentFrom

Equivalent classes and properties

- The Person class in my family relations ontology is equivalent to (used in exactly the same way as) the Person class in the Friend of a Friend (FOAF) ontology:
  - **:Person owl:EquivalentClass foaf:Person**
    - FOAF Person class URI: http://xmlns.com/foaf/spec/#term_Person
- The hasFamilyName property in my ontology is equivalent to (used in the same way as) the lastName property in the FOAF ontology:
  - **:hasFamilyName owl:equivalentProperty foaf:lastName**
    - FOAF lastName property URI: http://xmlns.com/foaf/spec/#term_lastName

Same individuals

- The Virtual International Authority File URI for Michelangelo identifies the same individual as the Library of Congress Linked Data Service Name Authority File URI for Michelangelo
  - **viaf:24585191 owl:sameAs lcnaf:n8015236**

    = http://viaf.org/viaf/24585191/#Michelangelo_Buonarroti,_1475-1564 **owl:sameAs** http://id.loc.gov/authorities/names/n80152368
- National Agricultural Library URI for concept Climatic Zones represents exactly the same concept as that from the Library of Congress Subject Headings:
  - **nal:12169 owl:sameAs lcsh:n8502703**

    = http://lod.nal.usda.gov/nalt/12169 **owl:sameAs** http://id.loc.gov/authorities/subjects/sh85027043