DR. MIGUEL CERIANI

PROF. ALEJANDRO VAISMAN

ONTOLOGÍAS EN LA WEB
6. RDFS Y INFERENCIA

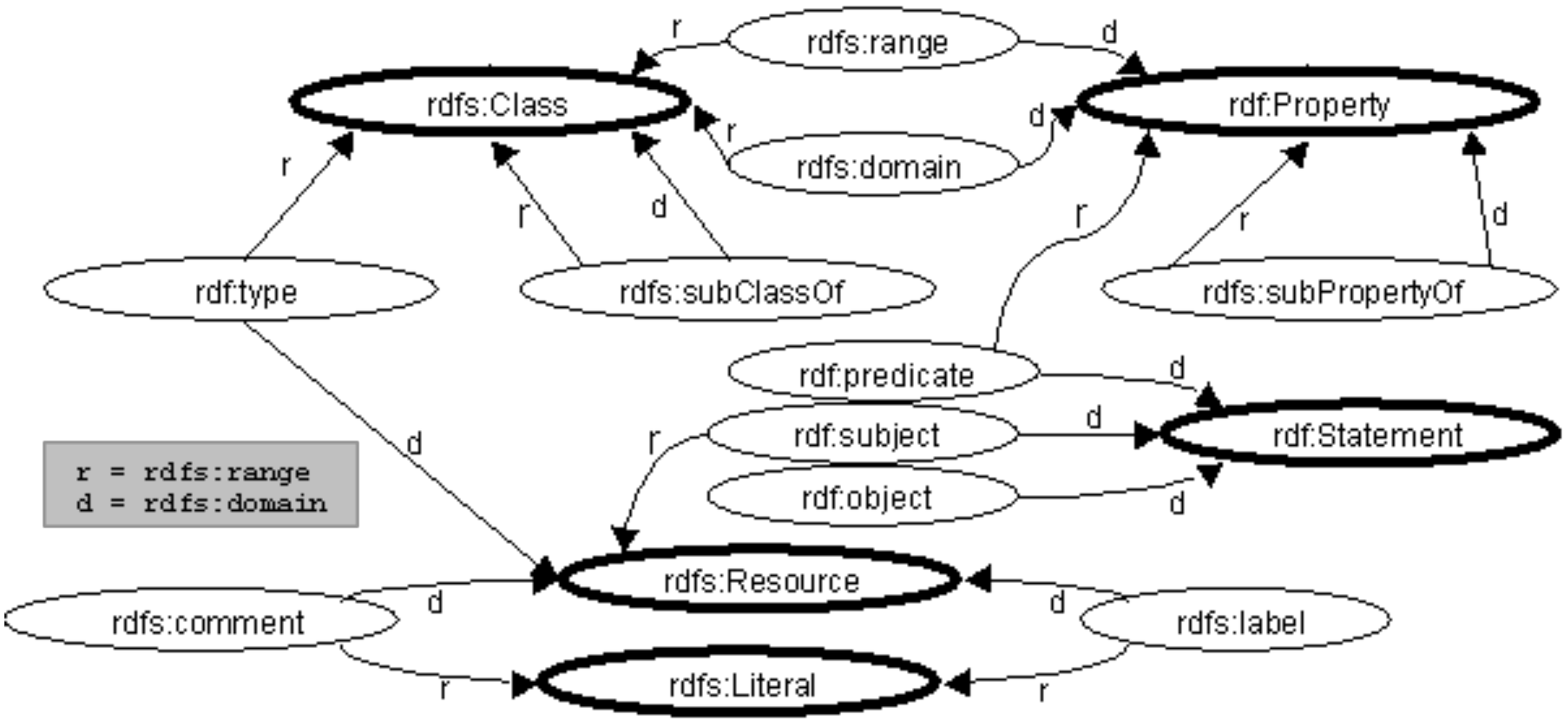
REFERENCIAS PARA ESTA CLASE (RDFS)

- Caps. 6 y 7 de “Semantic for the Working Ontologist”
- W3C RDF 1.1 Schema
- Semantic University: RDFS Introduction

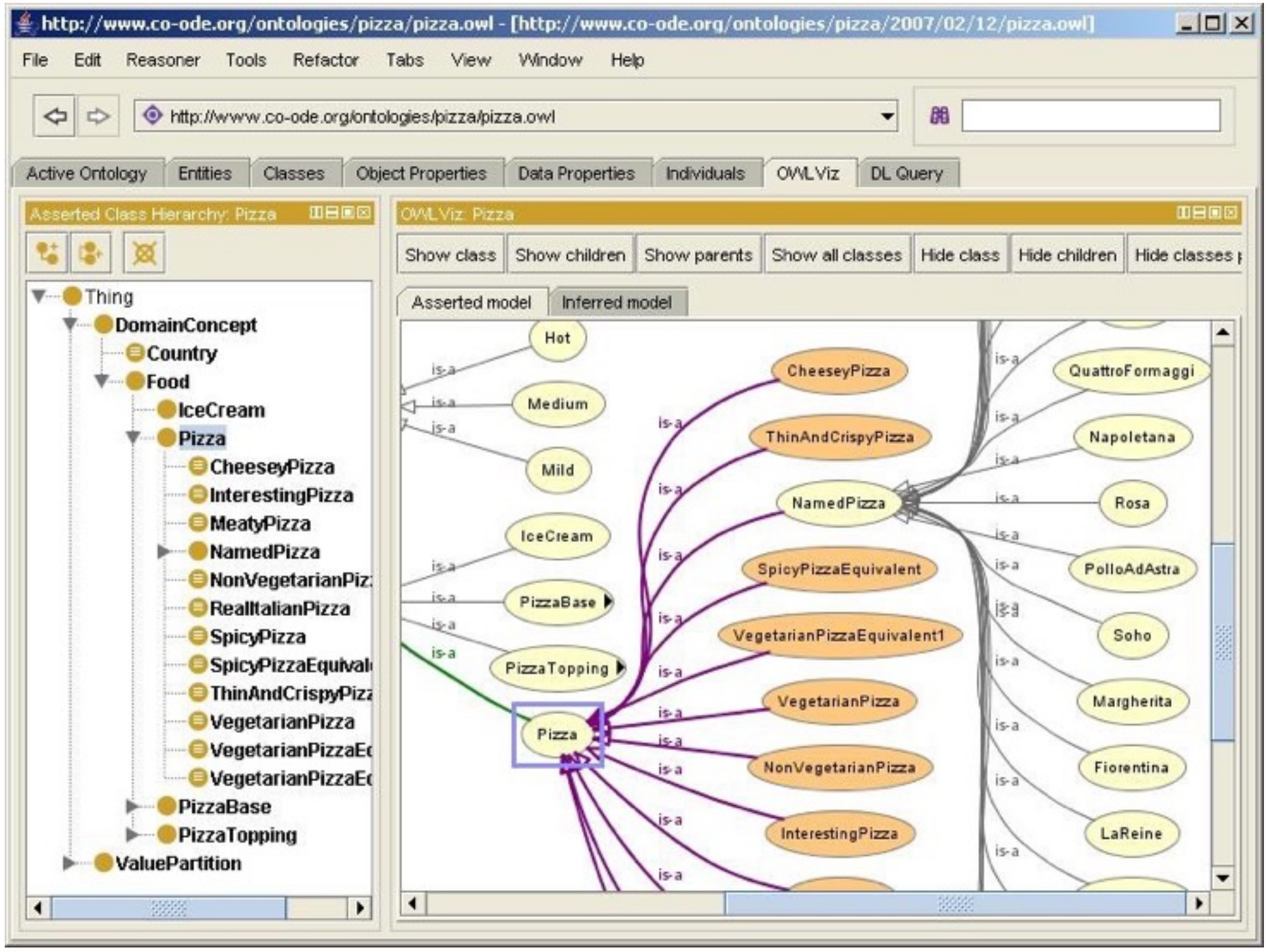
REPASO RDFS

- **Clases**: rdfs:Resource, rdfs:Class, rdfs:Literal, rdf:Property
- **Propiedades:** rdf:type, rdfs:subClassOf, rdfs:domain, rdfs:range, rdfs:subPropertyOf
- **Más Propiedades**: rdfs:label, rdfs:comment

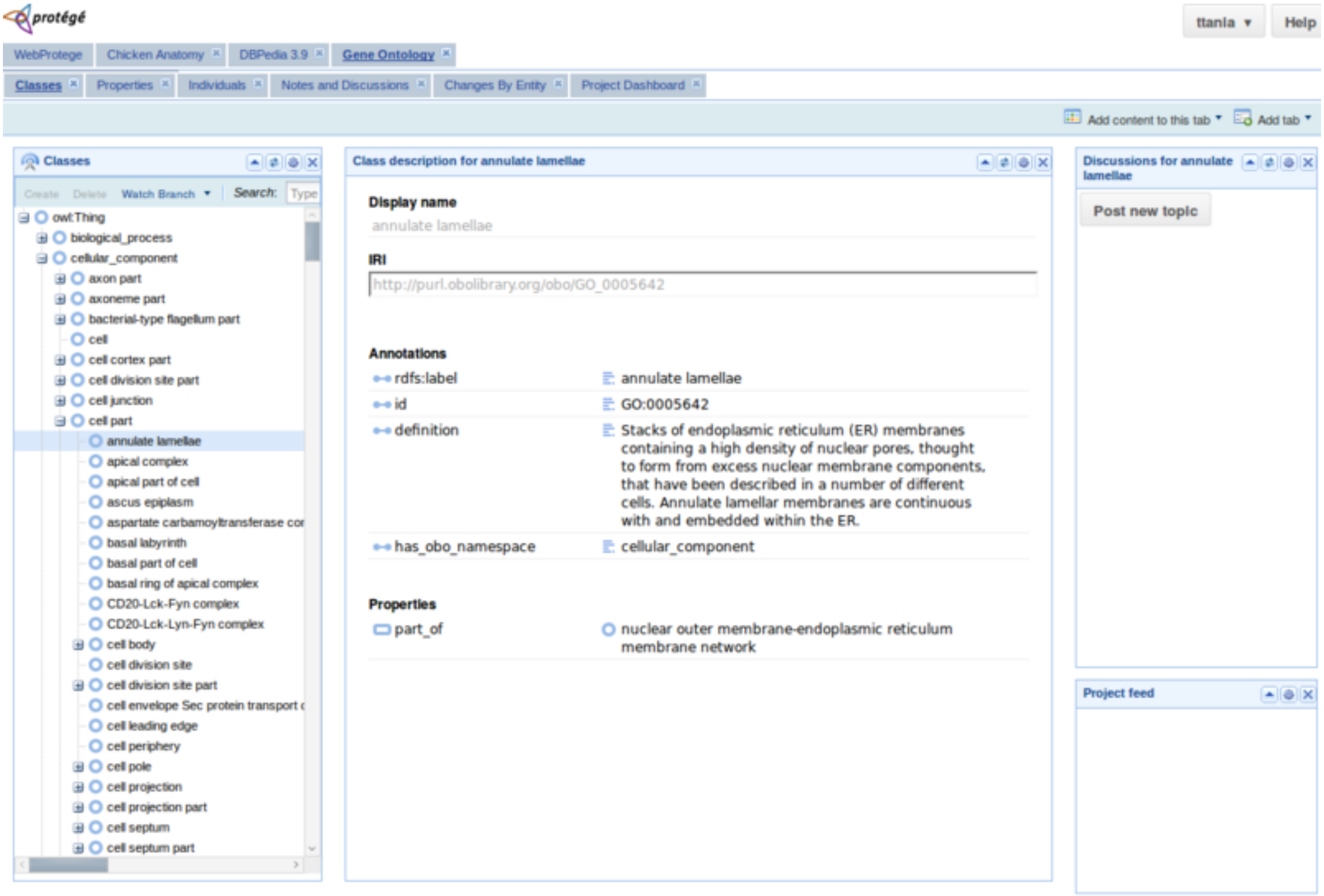
RELACIONES ENTRE CLASES Y PROPIEDADES RDFS



PROTÉGÉ: EDITOR DE ONTOLOGÍAS



WEB PROTÉGÉ: EDITOR DE ONTOLOGÍAS COLABORATIVO



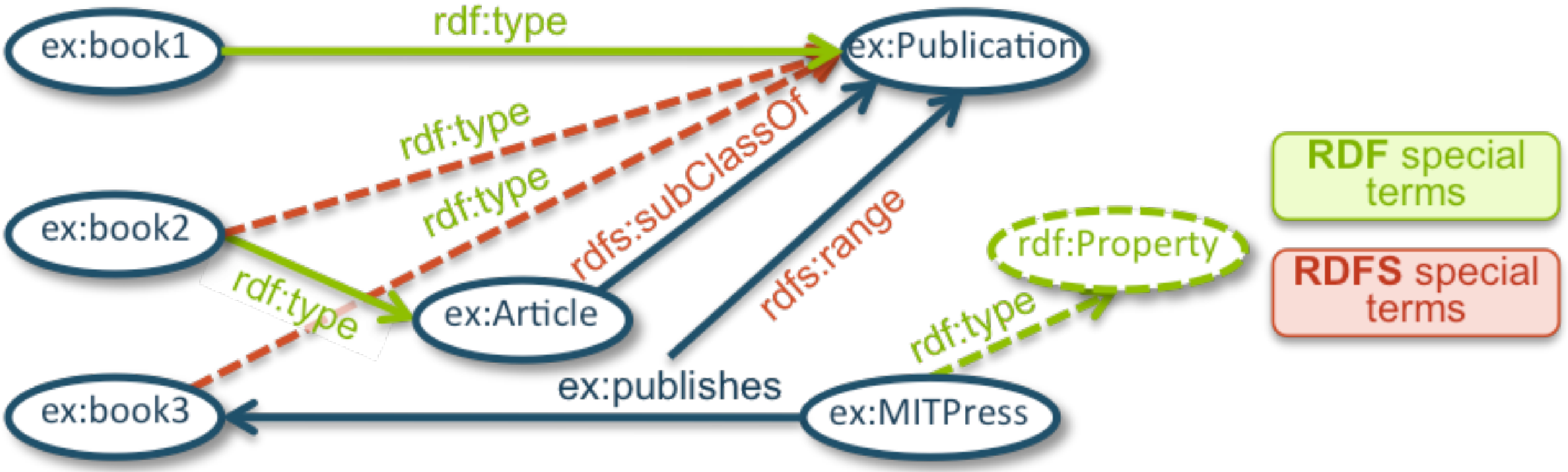
ACTIVIDAD: RDFS

- Instalar Protégé Desktop
- Crear una nueva ontología
- Definir una pequeña jerarquía de clases
- Definir algunas propiedades y caracterizarlas con domain y range
- Definir algunas entidades
- Salvar/Exportar en RDF/XML y Turtle

¿QUÉ PODEMOS HACER CON RDFS/OWL?

- evaluar las **consistencia** de un modelo (satisfacibilidad lógica)
- **deducir** nuevos hechos (implicación lógica)

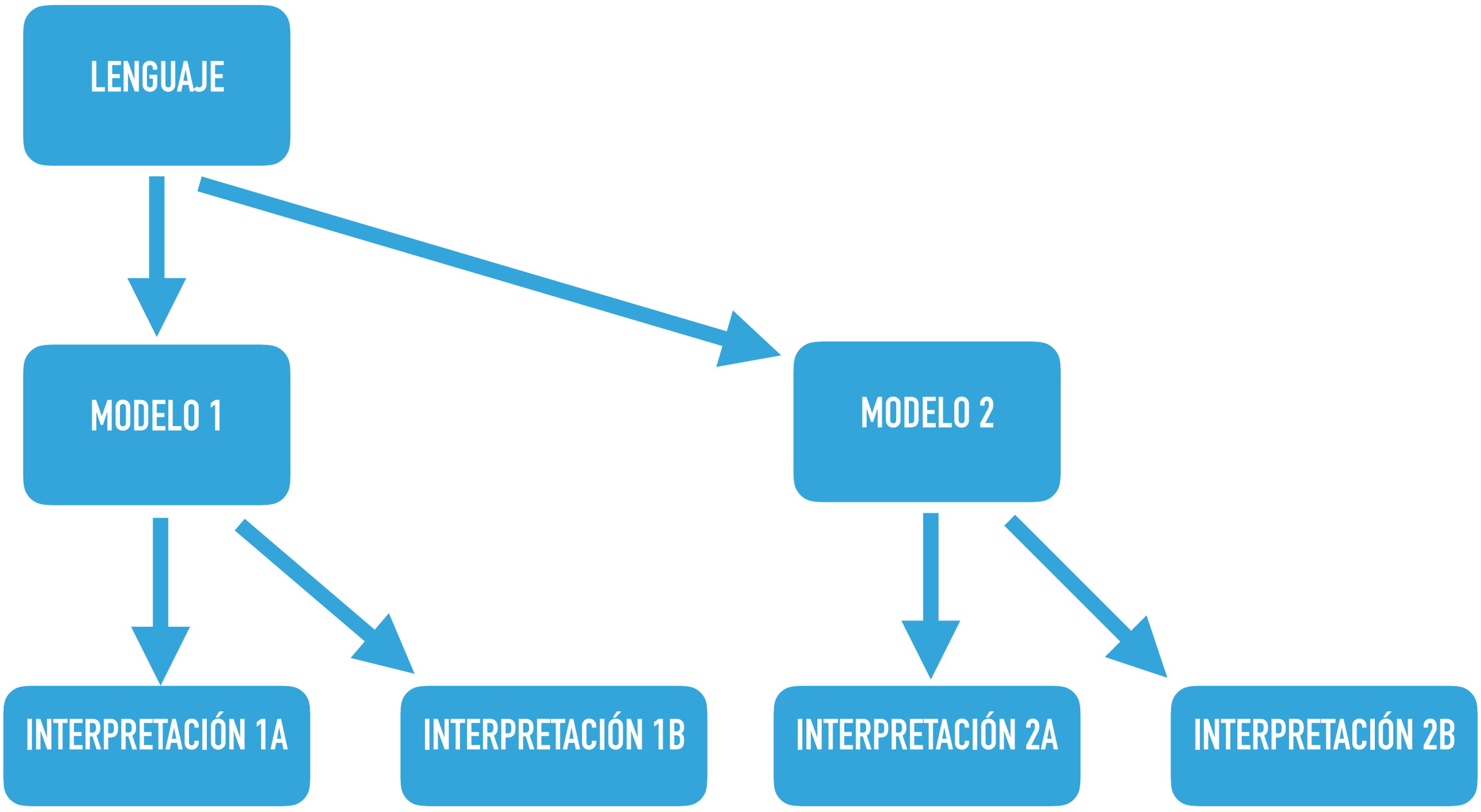
INFERENCIA


ex:book1
rdf:type
ex:Publication
rdf:type
rdf:type
rdfs:subClassOf
ex:book2
rdf:type
ex:Article
rdfs:range
rdf:Property
rdf:type
ex:publishes
ex:book3
ex:MITPress
RDF special terms
RDFS special terms

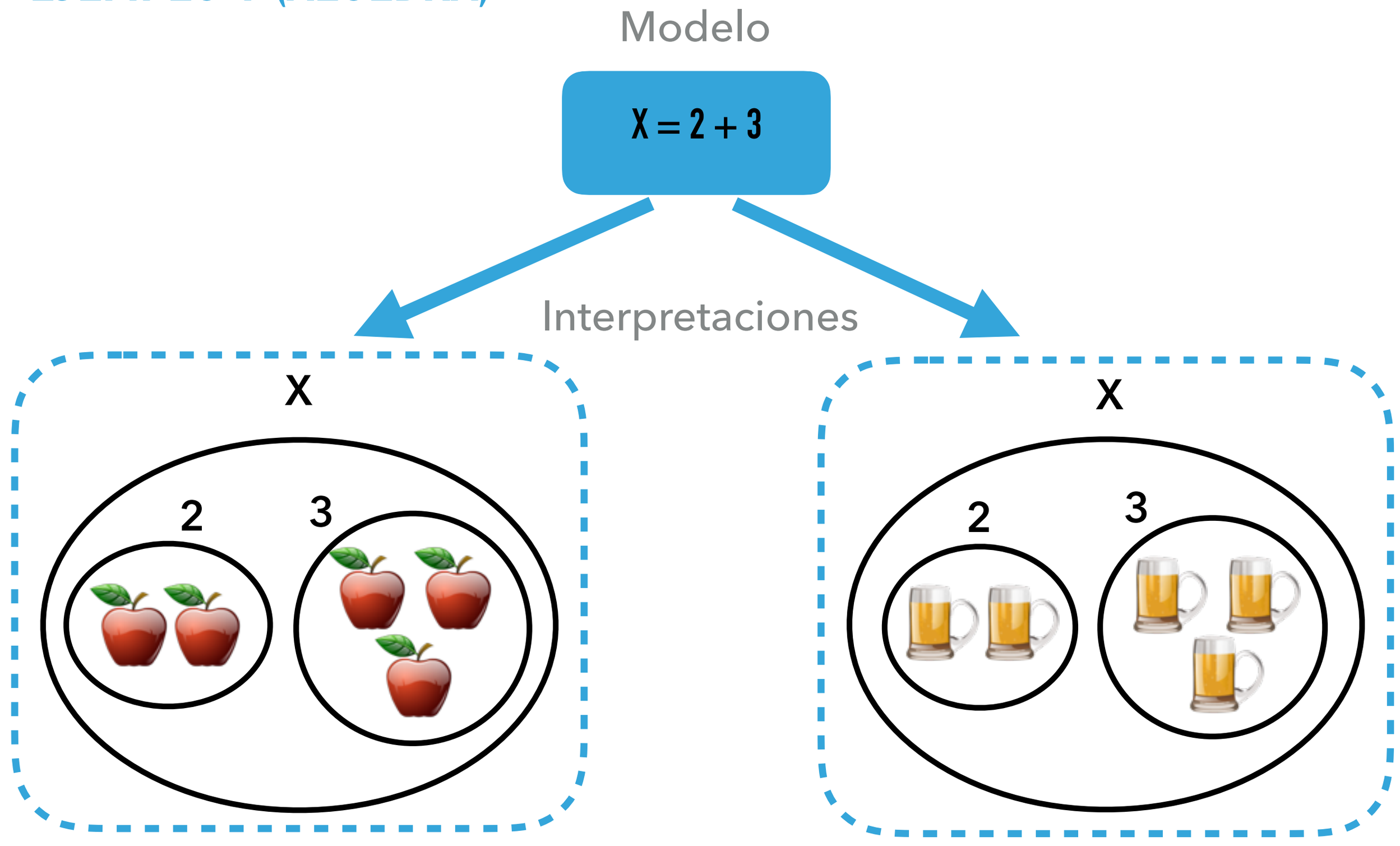
WEB SEMÁNTICA Y LÓGICA

- RDF, RDFS, OWL son **lenguajes formales** basados sobre IRIs, triplas y vocabularios estándar (rdf:, rdfs:, owl:)
- con estos lenguajes formales podemos definir **modelos** (RDF, RDFS, OWL)
- un posible significado de un modelo es una **interpretación**

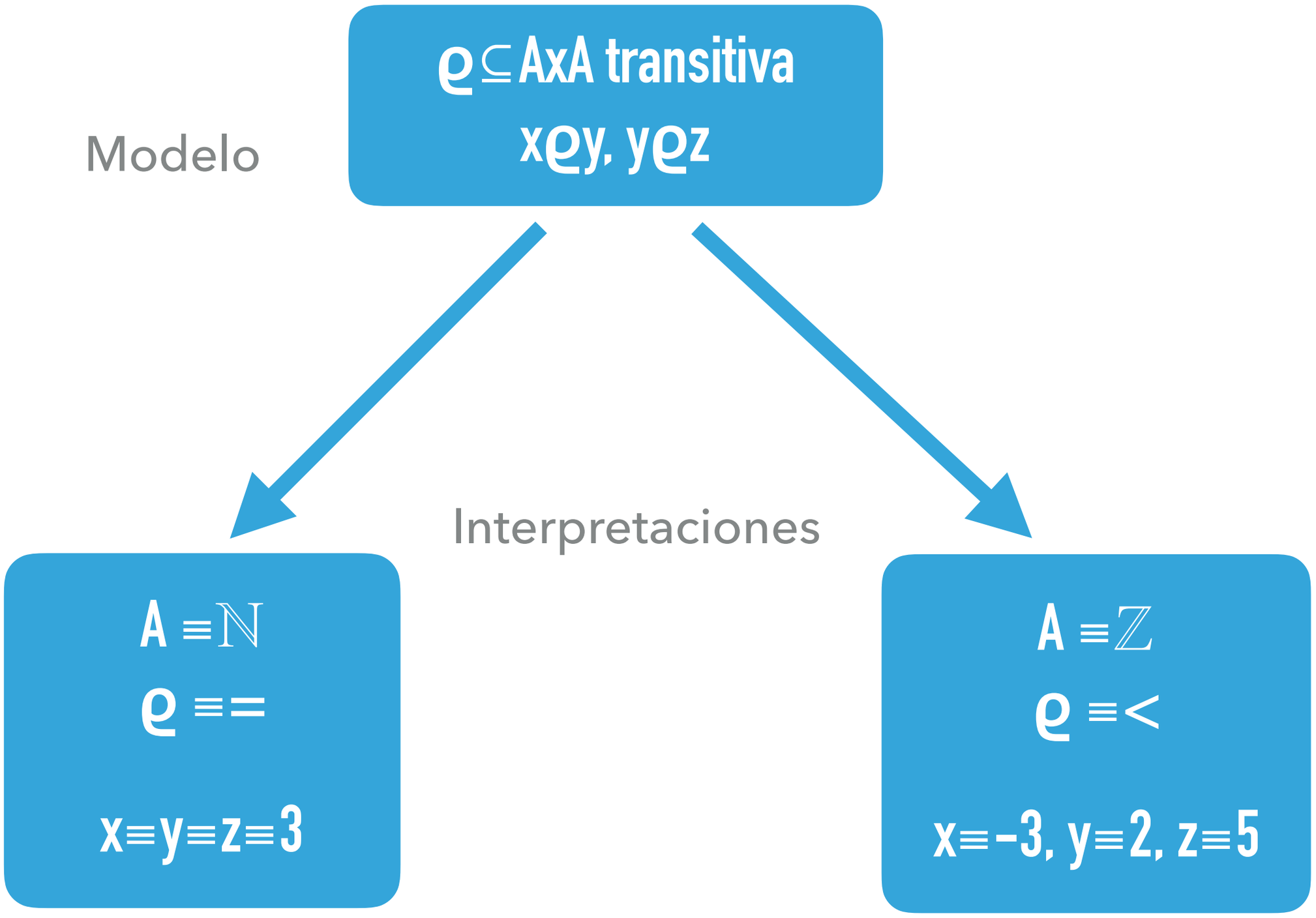
MODELOS Y INTERPRETACIONES



EJEMPLO 1 (ÁLGEBRA)



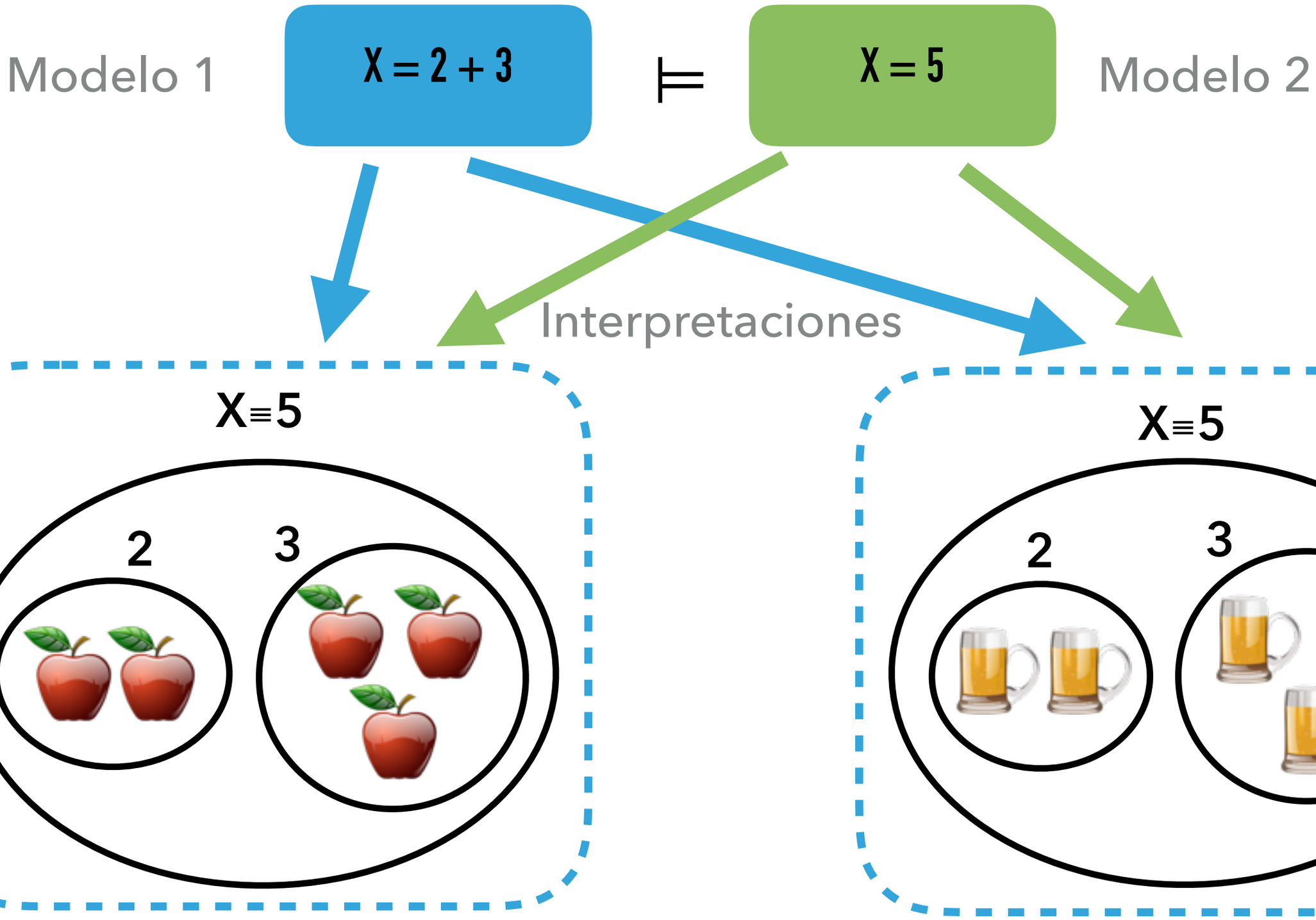
EJEMPLO 2 (CONJUNTOS Y RELACIONES)



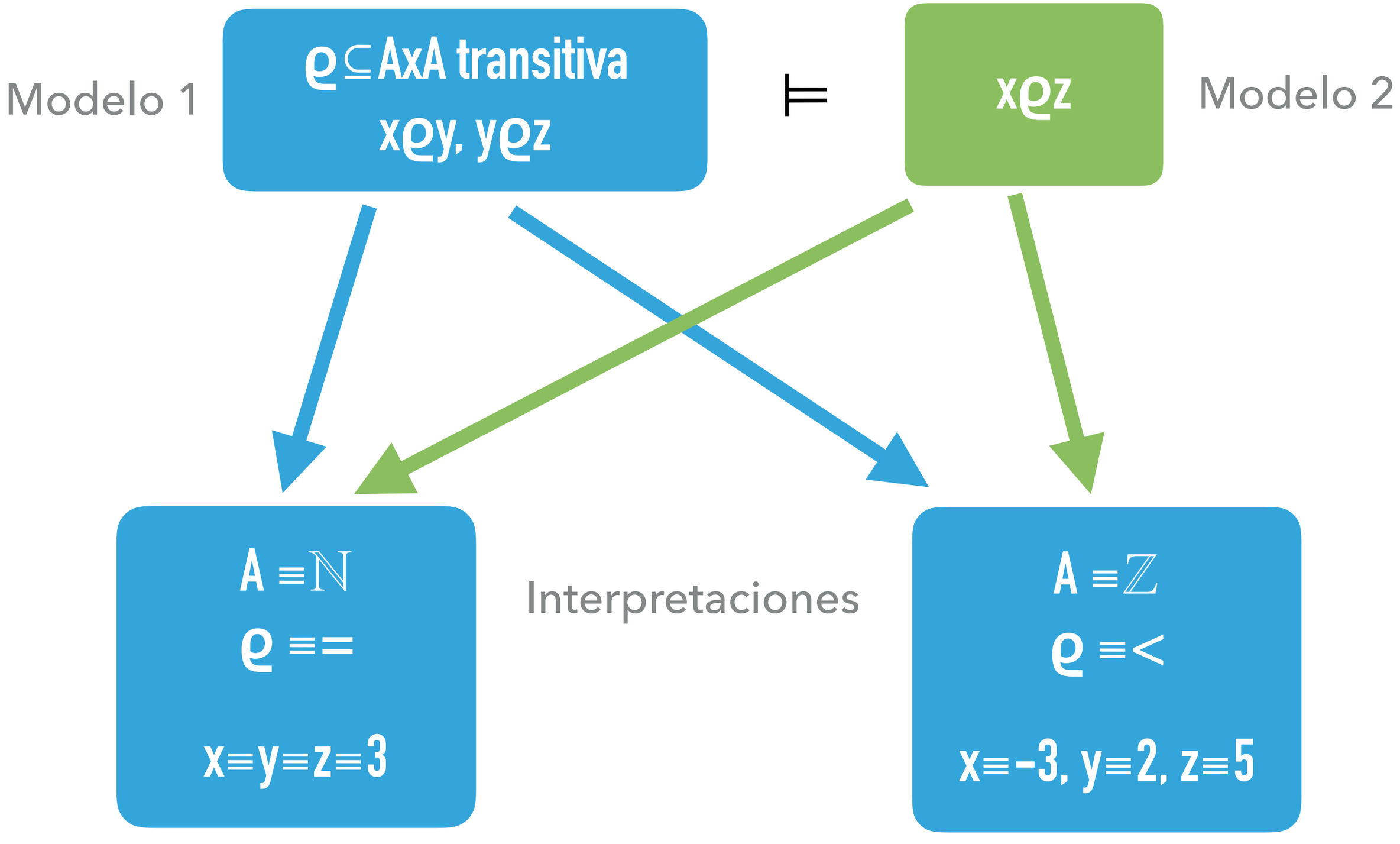
INTERPRETACIONES, IMPLICACIÓN, CONSISTENCIA

- las interpretaciones tienen que ser coherentes con los vínculos que elijamos: “**simples**”, **RDF**, **RDFS**, conjunto de tipos de dato reconocidos…
- interpretaciones con vínculos X, se llaman “**interpretaciones-X**”: interpretaciones-simples, interpretaciones-RDF, interpretaciones-RDFS,…
- un grafo RDF A “**X-implica**” otro grafo RDF B si cada interpretación-X de A es una interpretación-X para B
- un grafo RDF A es “**X-satisfacible**” si tiene al menos una interpretación-X.

IMPLICACIÓN: EJEMPLO 1 (ÁLGEBRA)



IMPLICACIÓN: EJEMPLO 2 (CONJUNTOS Y RELACIONES)



INTERPRETACIONES

- Interpretación Simple
- Interpretación con Conjunto de Datatypes D
- Interpretación RDF (con D)
- Interpretación RDFS (con D)

INTERPRETACIÓN SIMPLE

- cada IRI es asociada a un elemento en el conjunto del los recursos **IR**
- cada propiedad es asociada además a un subconjunto de **IR x IR**
- los nodos en blanco no son asociados a elementos de IR (interpretación existencial)

INTERPRETACIÓN SIMPLE: EFECTOS

- todos los grafos RDF son **satisfacibles**
- un grafo RDF A **implica** un grafo B si y solo si un sub-grafo de A es igual a B o una **instancia** de B (substitución de nodos en blanco)
- prácticamente no muy util de por sí solo

INTERPRETACIÓN CON CONJUNTO DE DATATYPES D

- valen los vínculos de una interpretación simple
- D es el conjunto de datatypes (xsd:integer, xsd:string, etc.) reconocidos
- cada datatype tiene un función de interpretación universal
- según el datatype, distintos literales pueden confluir en la misma interpretación
  `("25.0"^^xsd:decimal y "25"^^xsd:decimal)`

INTERPRETACIÓN CON CONJUNTO DE DATATYPES D: EFECTOS

- los grafos RDF pueden ser **no satisfacibles** solo en el caso de literales que no respetan los vínculos del datatype

```
ex:a ex:p "25.0"^^xsd:boolean .
```

- un grafo RDF A **implica** un grafo B en los casos previos o dependiendo de los datatype específicos

```
          ex:a ex:p "25.0"^^xsd:decimal .
implica   ex:a ex:p "25"^^xsd:decimal .
```

INTERPRETACIÓN RDF (CON D)

- valen los vínculos de una interpretación con un conjunto D de datatypes
- por cada tripla <s p o>, tiene que ser <p rdf:type rdf:Property>
- por cada datatype d en D y cada literal “..”^^d, el literal es interpretado con tipo d
- por cada propiedad p en el namespace rdf: (rdf:type, rdf:nil, ...), tiene que ser <p rdf:type rdf:Property>

INTERPRETACIÓN RDF (CON D): EFECTOS

- **triplas axiomáticas**

```
rdf:type rdf:type rdf:Property .
rdf:subject rdf:type rdf:Property .
rdf:predicate rdf:type rdf:Property .
rdf:object rdf:type rdf:Property .
rdf:first rdf:type rdf:Property .
rdf:rest rdf:type rdf:Property .
rdf:value rdf:type rdf:Property .
rdf:nil rdf:type rdf:List .
rdf:_1 rdf:type rdf:Property .
rdf:_2 rdf:type rdf:Property .
…
```

INTERPRETACIÓN RDF (CON D): EFECTOS

- reglas **deductivas** para la implicación

	si	entonces
rdfD1	xxx aaa "sss"^^ddd . *con* ddd *en* D	xxx aaa _:nnn . _:nnn **rdf:type** ddd .
rdfD2	xxx aaa yyy .	aaa **rdf:type** rdf:Property .

INTERPRETACIÓN RDFS (CON D)

- todos los nodos son de tipo rdfs:Resource
- todos los literales de texto con idioma son de tipo rdf:langString
- cada tipo datatype d en D es de tipo rdfs:Datatype
- si <x rdfs:domain y>, <u x v> ——> u es de tipo y
- si <x rdfs:range y>, <u x v> ——> v es de tipo y
- rdfs:subPropertyOf y rdfs:subClassOf son transitivas, reflexivas y se respeta el significado como “subconjunto de”
- cada clase es rdfs:subClassOf de rdfs:Resource
- cada rdfs:ContainerMembershipProperty es rdf:subPropertyOf de rdfs:member
- cada rdfs:Datatype es rdf:subClassOf de rdfs:Literal

INTERPRETACIÓN RDFS (CON D): EFECTOS

- **triplas axiomáticas 1/3**

```
rdf:type rdfs:domain rdfs:Resource .
rdfs:domain rdfs:domain rdf:Property .
rdfs:range rdfs:domain rdf:Property .
rdfs:subPropertyOf rdfs:domain rdf:Property .
rdfs:subClassOf rdfs:domain rdfs:Class .
rdf:subject rdfs:domain rdf:Statement .
rdf:predicate rdfs:domain rdf:Statement .
rdf:object rdfs:domain rdf:Statement .
rdfs:member rdfs:domain rdfs:Resource .
rdf:first rdfs:domain rdf:List .
rdf:rest rdfs:domain rdf:List .
rdfs:seeAlso rdfs:domain rdfs:Resource .
rdfs:isDefinedBy rdfs:domain rdfs:Resource .
rdfs:comment rdfs:domain rdfs:Resource .
rdfs:label rdfs:domain rdfs:Resource .
rdf:value rdfs:domain rdfs:Resource .
```

INTERPRETACIÓN RDFS (CON D): EFECTOS

- **triplas axiomáticas 2/3**

```
rdf:type rdfs:range rdfs:Class .
rdfs:domain rdfs:range rdfs:Class .
rdfs:range rdfs:range rdfs:Class .
rdfs:subPropertyOf rdfs:range rdf:Property .
rdfs:subClassOf rdfs:range rdfs:Class .
rdf:subject rdfs:range rdfs:Resource .
rdf:predicate rdfs:range rdfs:Resource .
rdf:object rdfs:range rdfs:Resource .
rdfs:member rdfs:range rdfs:Resource .
rdf:first rdfs:range rdfs:Resource .
rdf:rest rdfs:range rdf:List .
rdfs:seeAlso rdfs:range rdfs:Resource .
rdfs:isDefinedBy rdfs:range rdfs:Resource .
rdfs:comment rdfs:range rdfs:Literal .
rdfs:label rdfs:range rdfs:Literal .
rdf:value rdfs:range rdfs:Resource .
```

INTERPRETACIÓN RDFS (CON D): EFECTOS

- **triplas axiomáticas 3/3**

```
rdf:Alt rdfs:subClassOf rdfs:Container .
rdf:Bag rdfs:subClassOf rdfs:Container .
rdf:Seq rdfs:subClassOf rdfs:Container .
rdfs:ContainerMembershipProperty rdfs:subClassOf rdf:Property .

rdfs:isDefinedBy rdfs:subPropertyOf rdfs:seeAlso .

rdfs:Datatype rdfs:subClassOf rdfs:Class .

rdf:_1 rdf:type rdfs:ContainerMembershipProperty .
rdf:_1 rdfs:domain rdfs:Resource .
rdf:_1 rdfs:range rdfs:Resource .
rdf:_2 rdf:type rdfs:ContainerMembershipProperty .
rdf:_2 rdfs:domain rdfs:Resource .
rdf:_2 rdfs:range rdfs:Resource .
…
```

INTERPRETACIÓN RDFS (CON D): EFECTOS

- **reglas deductivas** para la implicación 1/4

	si	entonces
rdfs1	*cada* IRI aaa *en* D	aaa **rdf:type** rdfs:Datatype .
rdfs2	aaa **rdfs:domain** xxx . yyy aaa zzz .	yyy **rdf:type** xxx .
rdfs3	aaa **rdfs:range** xxx . yyy aaa zzz .	zzz **rdf:type** xxx .
rdfs4a	xxx aaa yyy .	xxx **rdf:type rdfs:Resource** .
rdfs4b	xxx aaa yyy.	yyy **rdf:type rdfs:Resource** .

INTERPRETACIÓN RDFS (CON D): EFECTOS

- reglas **deductivas** para la implicación 2/4

	si	entonces
rdfs 5	xxx **rdfs:subPropertyOf** yyy yyy **rdfs:subPropertyOf** zzz	xxx **rdfs:subPropertyOf** zzz
rdfs 6	xxx **rdf:type** rdf:Property	xxx **rdfs:subPropertyOf** xxx
rdfs 7	aaa **rdfs:subPropertyOf** bbb xxx aaa yyy .	xxx bbb yyy

INTERPRETACIÓN RDFS (CON D): EFECTOS

- reglas **deductivas** para la implicación 3/4

	si	entonces
rdfs8	xxx rdf:type rdfs:Class .	xxx rdfs:subClassOf rdfs:Resource .
rdfs9	xxx rdfs:subClassOf yyy . zzz rdf:type xxx .	zzz rdf:type yyy .
rdfs10	xxx rdf:type rdfs:Class .	xxx rdfs:subClassOf xxx .
rdfs11	xxx rdfs:subClassOf yyy . yyy rdfs:subClassOf zzz .	xxx rdfs:subClassOf zzz .

INTERPRETACIÓN RDFS (CON D): EFECTOS

- reglas **deductivas** para la implicación 4/4

	si	entonces
rdfs 12	xxx rdf:type rdfs:ContainerMembershipProperty	xxx rdfs:subPropertyOf rdfs:member
rdfs 13	xxx rdf:type rdfs:Datatype	xxx rdfs:subClassOf rdfs:Literal

RAZONADOR (REASONER)

- De un dataset genera nuevas triplas utilizando un **conjunto de reglas** (RDF, RDFS, OWL, datalog, …)
- Las nuevas triplas pueden ser:
  - agregadas cada vez que el dataset es modificado (**forward chaining**); o
  - generadas para responder a una query solo cuando necesario (**backward chaining**)
- Sea Protégé que los principales Graph Store permiten elegir entre varios razonadores incorporados o externos

ACTIVIDAD: INFERENCIA RDFS

- Utilizar en Protégé un reasoner
- Crear en un Graph Store (Sesame, OpenLink Virtuoso, Apache Fuseki) un repository con inferencia RDFS
- Exportar Ontología realizada con Protégé
- Importarla en el Graph Store
- Realizar queries para probar las inferencias

EMAIL

mceriani@itba.edu.ar