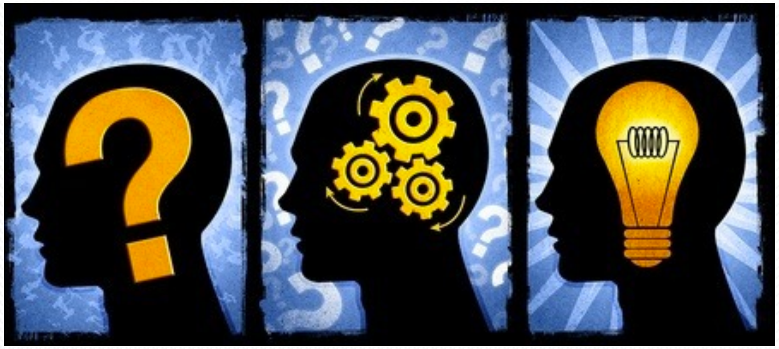
Semantic Web Reasoning



OWL Reasoners (Description Logic-based)

- An OWL DL reasoner can be used to
  - Check consistency of ontologies
  - Compute the classification hierarchy (subsumption)
  - Satisfiability
  - Instantiation
  - And other standard TBox and Abox reasoning tasks
- Some DL reasoners
  - HermiT (Oxford University, comes with Protégé 4+)
  - FaCT++ (Manchester University)
  - Pellet
  - Racer
  - …

DL-based Reasoning

- Reasoner can be used to check
  - Consistency
  - Subsumption
  - Satisfiability
  - Instantiation

Instantiating disjoint classes

Individual: MadCow ❌
  Types: Vegetarian, Carnivore
  DisjointClasses: Vegetarian, Carnivore

- Mostly implemented based on Tableau algorithms

Tableau Algorithm

- Often used to decide concept satisfiability
- An Example of subsumption reasoning
  - Given: Cat⊆Mammal, Mammal⊆ Animal
  - Question: If : Cat⊆ Animal
- Reasoning process
  - Test the satisfiability of concept *C=(CatΠ¬Animal)*, see if there is an individual x that is an instance of Cat but not Animal
  - *C(x) -> Cat(x), ¬Animal(x)*
  - *Cat(x) -> Mammal(x)*
  - *Mammal(x)->Animal(x)*
  - *Animal(x)* contradicts *¬Animal(x)*
  - Therefore C is unsatisfiable, therefore Cat⊆ Animal is true with the given ontology.

Rule-base Reasoning

- Semantic Web Rule language (SWRL)
  - A Rule Language Combing OWL and RuleML
  - Allows users to write rules that can be expressed in terms of OWL concepts
  - Provide more powerful deductive reasoning capabilities than OWL alone
  - https://www.w3.org/Submission/SWRL/
- Jena Inference Support: Reasoners and rule engines
  - Inference API
  - General Purpose Rule Engine
  - https://jena.apache.org/documentation/inference/

Semantic Web Rule Language - SWRL

- In SWRL syntax, a rule has the form:

  Antecedent -> Consequent

  - both antecedent and consequent are conjunctions of atoms written a1 ^ a2 ... ^ an. Variables are indicated using the standard convention of prefixing them with a question mark.
  - e.g. hasParent(?x,Tom) represents a triple statement

    Variable ?x hasParent Tom

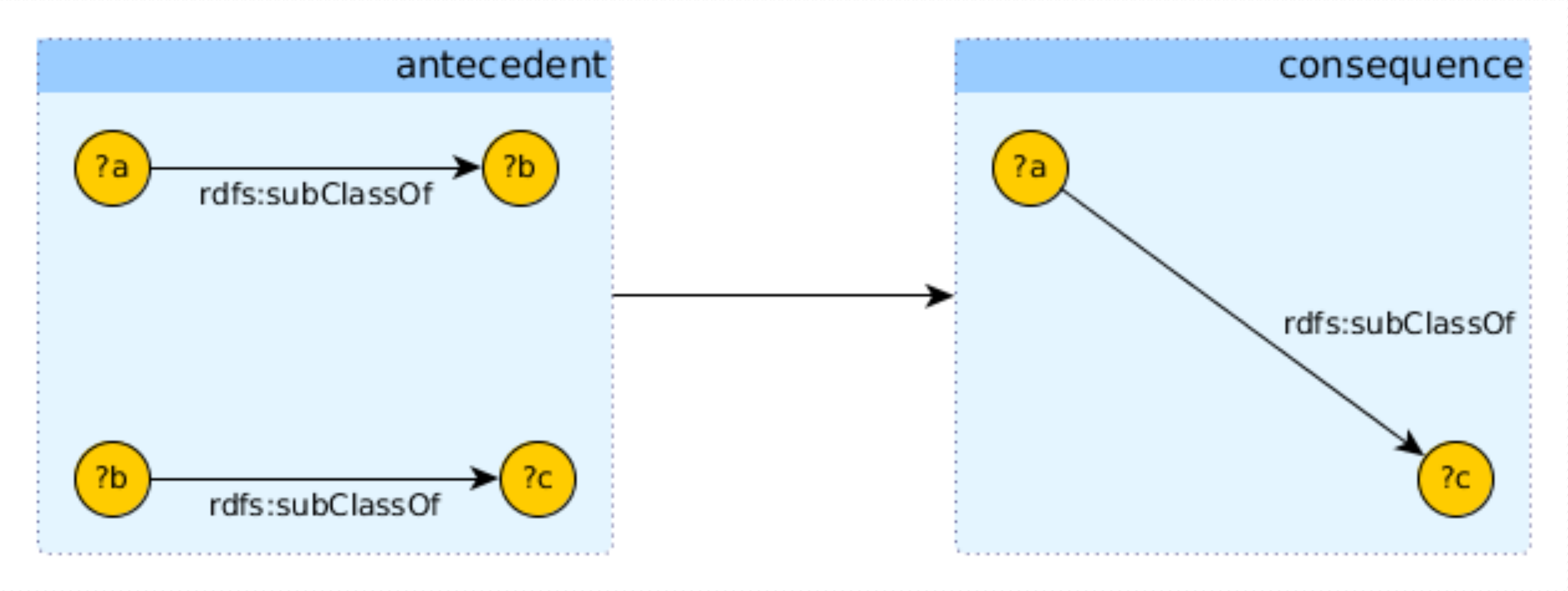
- For example:
  - *hasParent(?x,?y) ^ hasBrother(?y,?z) ->hasUncle(?x,?z)*
  - *hasParent(?x,?y) ^ Man(?y)->Father(?y)*
  - *hasChild(?x,?y) ^ Man(?y)-> hasSon(?x,?y)*
  - *hasParent(?x,?y) ^ hasParent(?z,?y) ^ Man(?z)-> hasBrother(?x,?z)*
  - *hasSibling(?x,?y) ^ Man(?y)->hasBrother(?x,?y)*

  ..

Semantic Web Rule Language - SWRL (cont.)

Example 1: SWRL rule that enables the transitive closure of the class hierarchy

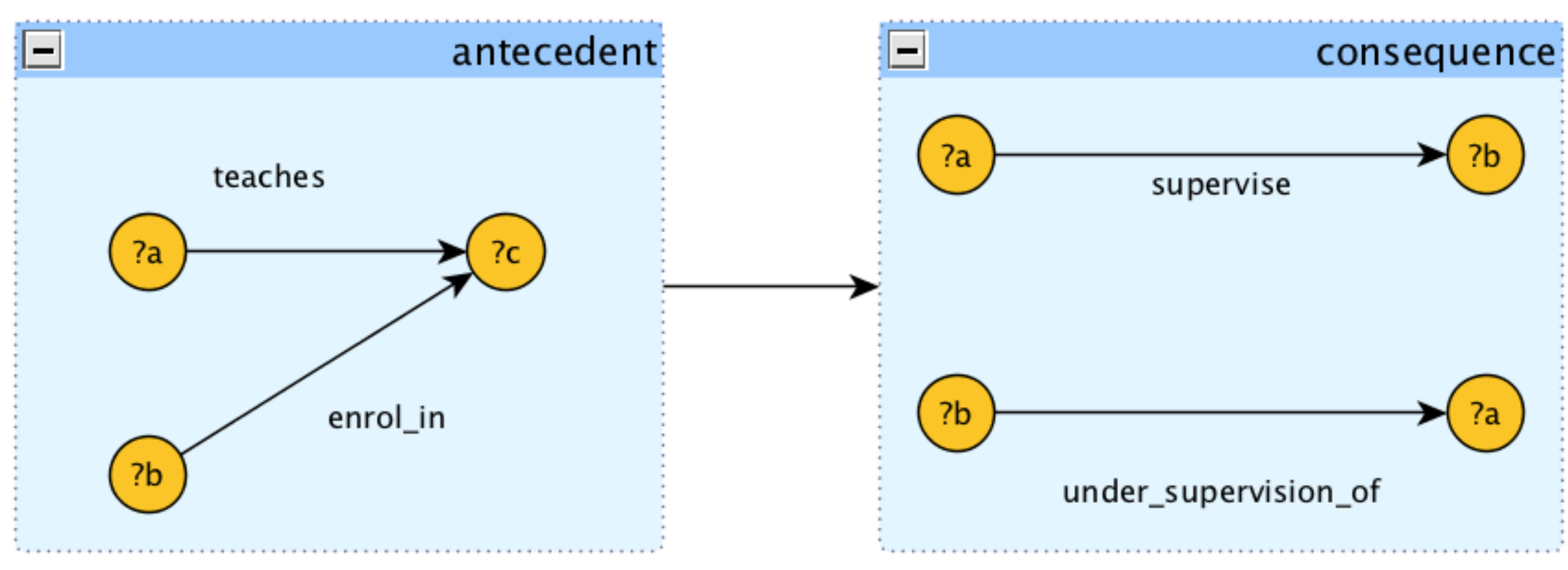
SubClassOf(?a, ?b) ^ SubClassOf(?b,?c) -> SubClassOf(?a, ?c)



Semantic Web Rule Language - SWRL (cont.)

Example 2: custom SWRL rule

teach(?a,?c)^ enrol_in(?b,?c) -> supervise(?a, ?b)



Limitation: predicates cannot be variables!

Jena Reasoning Rules

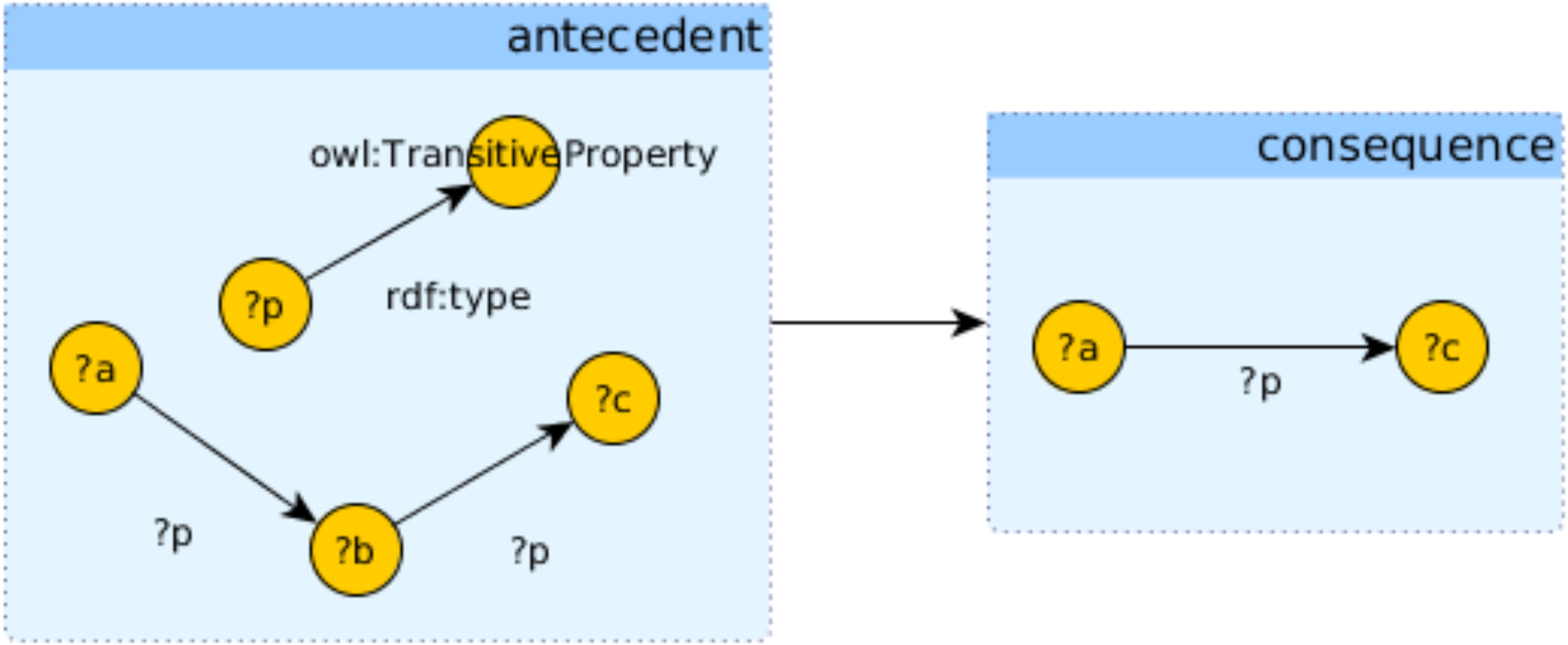
- Jena Reasoning rules are similar to SWRL rules but its predicate can also be a variable.
- Jena Reasoning Rule Syntax:

  Antecedent -> Consequent

  - both antecedent and consequent are conjunctions of triple pattern written t1, t2 ... , tn.
- Examples:
  - [rule1: (?a rdfs:subClassOf ?b), (?b rdfs:subClassOf ?c)->(?a rdfs:subClassOf ?c)]
  - [transitive_rule1: (?p rdf:type owl:TransitiveProperty),(?a ?p ?b), (?b ?p ?c)-> (?a ?p ?c)]

Jena Reasoning Rules (cont.)

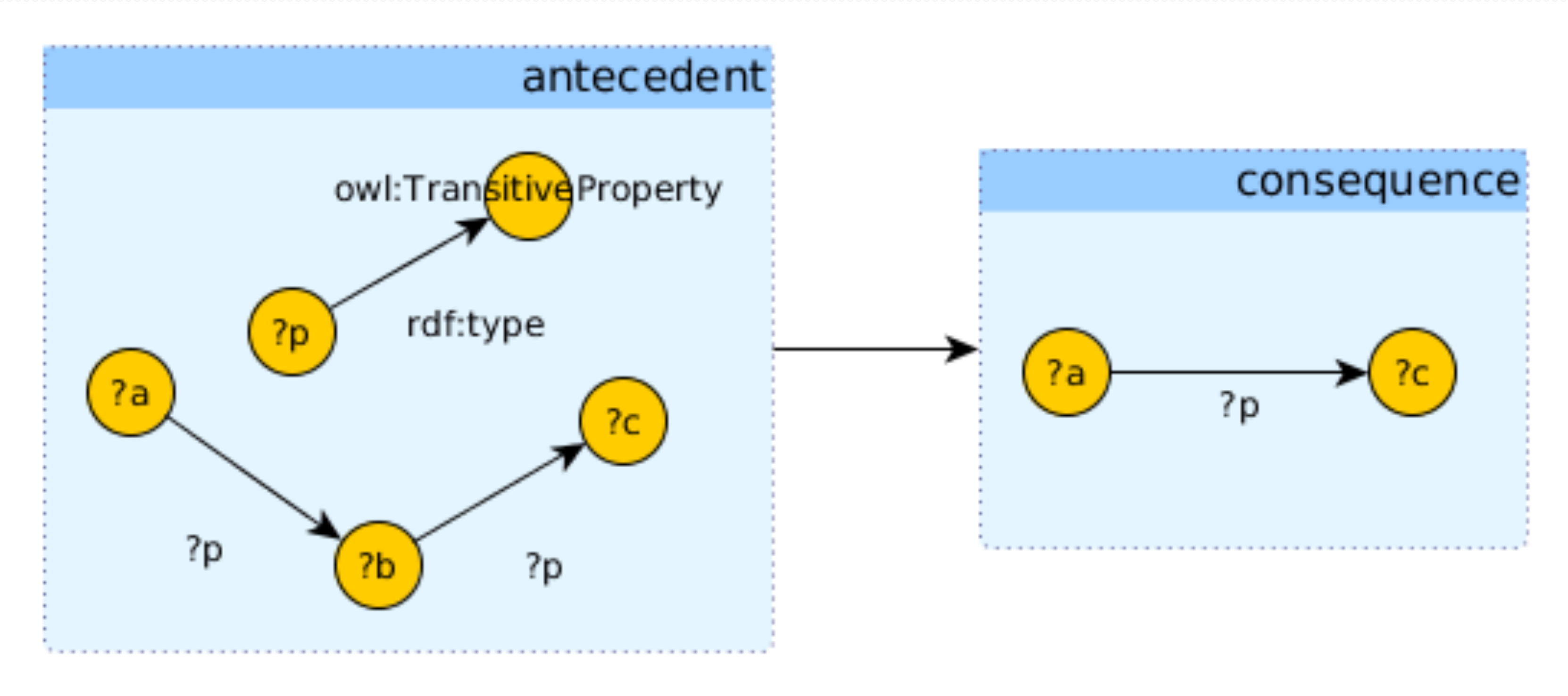
[transitive_rule1:
(?p rdf:type owl:TransitiveProperty),(?a ?p ?b), (?b ?p ?c)-> (?a ?p ?c)]



Jena Reasoning Rules (cont.)

rule name

[transitive_rule1:
(?p rdf:type owl:TransitiveProperty),(?a ?p ?b), (?b ?p ?c)-> (?a ?p ?c)]



Programming the Semantic Web

protégé

Jena

The OWL API

An Overview of the Semantic Web API

- Protégé OWL API
  - Supporting OWL-DL and RDF(S)
  - http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide
- Apache Jena API
  - For building RDF(S)-centric Semantic Web Application
  - https://jena.apache.org/
- OWL API
  - More OWL2-centric

Protégé OWL API

- Open-source Java library for OWL and RDF(S)
  - For the development of components executed inside Protégé-OWL editor (Version 3.X)
  - For development for standalone applications
- Tutorial:
  - http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide
  - Basics: http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Basics
  - Advanced Topics

    http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Advanced_Topics
  - Advanced Class Definition (OWL-DL)

    http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Advanced_Class_Definitions

Apache Jena

- Open source Semantic Web framework for Java. https://jena.apache.org/index.html
  - ARQ: Query your RDF data
  - Ontology API: Work with models, RDFS and OWL 1
  - Jena TDB: RDF triplestore
  - Fuseki: SPARQL end-point accessible over HTTP
- More appropriate for developing RDF-centric applications.(e.g SPARQL)

OWL API

- Java API and reference implementation for creating, manipulating and serialising OWL Ontologies https://github.com/owlcs/owlapi/wiki/Documentation
- More OWL-centric
- Support new OWL2 constructs

The OWL API