### Syntax vs. Semantics

#### Syntax

* Merupakan Proportitional Variables
* Syntax merupakan koleksi simbol dan term bersamaan dengan rules, untuk membentuk suatu sentence yang dianggap valid.
* Koleksi dari terms itu kita sebut dengan vocabulary.

#### Statement

* Setiap proporsitional variable itu adalah sebuah statement.
* Jika misalnya a dan b adalah statement, maka:
  + $a \and b$ merupakan statement
  + $a \or b$ merupakan statement
  + merupakan statement
  + merupakan statement

#### Semantics (Basis logic)

* Merupakan himpunan semesta.
* mapping statement to himpunan semesta
* Meaning dari syntax.
* Didefinisikan dengan himpunan,
* Interpretasi yang menspesifikan setiap simbol dan statement itu artinya apa dan kapan sebuah statement disebut true atau false.

**Syntax is the grammatical structure of the text, whereas semantics is the meaning being conveyed**

### Entailment

Hubungan antara statement yang true saat diketahui suatu himpunan statement itu true.

Suatu himpunan disebut **entails** sebuah himpunan lain , bila setiap interpretasi dari yang membuat benar juga membuat benar.

$$G: \{p \implies q, p\}\\
H: \{q\}\qquad\text{(1)}$$

Jadi cara untuk mengecek entailment adalah looping setiap interpretasi yang membuat benar, terus apabila ada satu saja yang membuat salah, maka itu tidak entail.

### Reasoning/inference:

Diberikan dua buah himpunan dan , cek apakah entails , intinya ini proses inferensi.

### Semantic dari RDF Graph

* Graf RDF itu hanya himpunan triple
  + Tidak ada deskripsi tentang struktur dan bentuknya
  + Tidak ada makna spesifik yang diberikan kepada setiap IRI

Sebuah RDF graph true bila semua triplenya true, dan definisi dari triple true jika dan hanya jika ada sebuah relasi binary yang diidentifikasi dengan yang menhubungkan entity dan entity lain .

* Kebanyakan triple hampir selalu true
* Jika literal, maka bentuk lexicalnya mesti cocok dengan tipe datanya, "test"^^xsd:integer itu ill typed dan bermasalah.

RDF triple itu dinyatakan dalam bentuk .

* , subject, bisaa IRI atau blank node.
* , mesti IRI
* , bisa IRI, blank, atau literal.

Kalau RDF graph itu himpunan triple. Dari DELG, kita tinggal menentukan node mana ayng literal dan tipenya apa, terus sisanya bisa kita IRI dan jadikan dia object entitas sendiri.

### Entailment of RDF Graph

xxx ppp yyy .

entails

xxx ppp yyy .

xxx ppp yyy .

entails

xxx ppp \_:nnn .

xxx ppp yyy .

entails

\_:nnn ppp yyy .

xxx ppp "aaa"^^ddd .

entails

xxx ppp \_:nnn .  
\_:nnn rdf:type ddd .

xxx ppp yyy .

entails

ppp rdf:type rdf:Property .

entails

rdf:type rdf:type rdf:Property .

### RDFS Features

Pada RDF Schema, ada beberapa komponen penting: Model classes, Instances, Property constraint, Class, Property Hierarchy

### rdf:type dan rdfs:Class

:Surabaya a o:City .  
:Jakarta a o:City .  
o:City a rdfs:Class .

rdf:type merupakan relasi **instance-of** yang merupakan relasi yang menyatakan keanggotaan himpunan.

Saat kita menggunakan relasi **instance-of**, itu sisi kanannya otomatis diinfer sebagai himpunan sebagai class.

Secara semantik, o:City merupakan instance dari rdfs:Class, jadi sebenarnya baris ke tiga bisa diderive dari baris 1.

SPARQL tidak memproses RDFS entailment. Namun, jika RDFS Entailment Regime di enable

### rdfs:subclassOf

:UI rdf:type o:University .  
:Pertamina a o:OilCompany .  
o:University rdfs:subClassOf o:EducationInstitution .  
o:OilCompany rdfs:subClassOf o:ForProfitOrganization .  
o:EducationInstitution rdf:subClassOf rdfs:Organization .  
o:ForProfitOrganization rdf:subClassOf rdfs:Organization .

* rdfs:subClassOf merupakan sebuah relasi subhimpunan
* Line ke-3 berarti setiap instance o:University juga merupakan instance dari o:EducationInstitution, implicationnya adalah (x, rdf:type, o:University) (x, rdf:type, o:EducationInstitution)
* Sebuah class bisa memiliki beberapa subclasses dan multiple superclass.
* rdfs:subClassOf bentuknya transitif, jadi bisa membentuk hierarki kelas.

Negasi doesn't exist in RDF. Tapi di (OWL Web Ontology Language) bisa. Semakin ekspresif sebuah bahasa, semakin mahal.

### rdf:property

Pada dasarnya merupakan class setiap property. Untuk setiap predikat dari sebuah triple, itu disebut sebuah property.

Property memiliki domain dan range, himpunan subjek dari sebuah properti disebut domain, himpunan objeknya bisa disebut range.

### rdfs:domain dan rdfs:range

o:taughtBy rdfs:domain o:Course .  
o:taughtBy rdfs:range o:FacultyMember .

* Line 1 implies
  + o:taughtBy instance dari rdf:Property.
  + o:Course instance dari rdf:Class.
  + (x, o:taughtBy, y) x instance dari o:Course.
* Line 2 implies
  + (x, o:taughtBy, y) y instance dari o:FacultyMember.

#### rdfs:subPropertyOf

o:taughtBy rdfs:subPropertyOf o:involves

* (x, o:taughtBy, y) (x, o:involves, y)