

# Pronto:Reasoner

Ngoc-Tung Nguyen

Week Six

CS-498

# Ontology File Structure

- Classical Part
  - OWL ontology
    - Example: [Classical part URI](#)
- Probabilistic Part
  - Statistical knowledge
    - Example: [Probabilistic part URI](#)

# Pronto - P-SHIQ(D)

- *Generic probabilistic knowledge*
  - knowledge doesn't apply to any specific individual but rather to a randomly chosen one
    - $(D \mid C)[l,u]$ , where  $C$  and  $D$  are DL concepts and  $[l,u]$  is a closed subinterval of  $[0,1]$
- *Concrete probabilistic knowledge*
  - knowledge applies to a specific individual
    - $\text{Tweety}:(\text{FlyingObject} \mid \text{owl:Thing})[0.0;0.05]$

# How Reasoning Works

- Pronto computes all minimally sufficient (*tightest*) subsets of conditional constraints for a particular entailment (*implies*), both for generic and concrete
- Entailment-  $A \models B$ 
  - T - antecedent, B – consequent
    - *Ex:* Let the set  $A$  of sentences include 'All horses are animals' and 'All stallions are horses', and the set  $B$  of sentences include 'All stallions are animals'

<http://www.kr.tuwien.ac.at/research/reports/rr0605.pdf>

<http://en.wikipedia.org/wiki/Entailment>

# Generic and Concrete Entailments

- Generic Entailment
  - given a probabilistic KB and a pair of concepts, compute the tightest interval  $(D | C)[l,u]$
- Concrete Entailment
  - given a probabilistic KB, an individual “a”, and a concept “D”, compute the tightest interval  $(D | owl:Thing)[l,u]$  for “a”.
- What does this allow for?
  - Pronto can infer the probability of a statement like Tweety being a flying object based on other statements rather than asserting the conditional constraint
  - Ts are the concepts (Classical) , B is the probabilistic KB

# Override Principle

- Ex: Nixon Diamond
  - Handles conflicts between condition constraints
    - T concepts
  - Allowing more specific constraints (SC) to *override* more generic ones
  - Ex: Tweety is a Penguin (SC), Penguin is a subclass of Bird (Classical Ontology)
    - (FlyingObject | Bird)[0.9;1.0] (Overridden)
    - (FlyingObject | Penguin)[0.0;0.05] (Taken)
    - => entail Tweety:(FlyingObject | owl:Thing)[0.0;0.05]