



# *Γλώσσα Οντολογιών Ιστού:* **OWL**

Εισαγωγή στο Σημασιολογικό Ιστό

# Εισαγωγή

- Η εκφραστικότητα των γλωσσών RDF και RDF Schema είναι σκόπιμα πολύ περιορισμένη
  - Η RDF περιορίζεται σε δυαδικά βασικά κατηγορήματα
  - Η RDF Schema σε μία ιεραρχία υποκλάσεων και μία ιεραρχία ιδιοτήτων, με ορισμούς του πεδίου ορισμού και του συνόλου τιμών για τις ιδιότητες αυτές
- Παρόλα αυτά, η Ομάδα Εργασίας Οντολογιών Ιστού του W3C προσδιόρισε έναν αριθμό χαρακτηριστικών περιπτώσεων χρήσης για το ΣΙ, οι οποίες θα απαιτούσαν πολύ περισσότερη εκφραστικότητα από αυτήν που προσφέρουν οι RDF και RDF Schema
  - Αρκετές ερευνητικές ομάδες έχουν ήδη αναγνωρίσει την ανάγκη για μία ισχυρότερη γλώσσα μοντελοποίησης οντολογιών
- Αυτό οδήγησε σε μια κοινή πρωτοβουλία ορισμού μιας πλουσιότερης γλώσσας, που ονομάζεται DAML+OIL
- Η Ομάδα Εργασίας Οντολογιών Ιστού χρησιμοποίησε την DAML+OIL ως αφετηρία για τον ορισμό της OWL
  - Γλώσσα που αποσκοπεί να γίνει η πρότυπη και ευρέως αποδεκτή γλώσσα οντολογιών για το ΣΙ



# **OWL KAI RDF/RDFS**

# Απαιτήσεις των Γλωσσών Οντολογιών (1/5)

- Οι γλώσσες οντολογιών επιτρέπουν στους χρήστες να γράφουν, ρητές, τυπικές επίνοιες (conceptualizations) για μοντέλα πεδίων
- Οι κύριες απαιτήσεις είναι
  - Καλά ορισμένη σύνταξη
  - Αποδοτική υποστήριξη συλλογισμών
  - Τυπική σημασιολογία
  - Επαρκής εκφραστική ισχύς
  - Ευκολία στην έκφραση
- Η σπουδαιότητα της *καλά ορισμένης σύνταξης* είναι ξεκάθαρη και γνωστή από το χώρο των γλωσσών προγραμματισμού
  - Αποτελεί αναγκαία συνθήκη για την επεξεργασία πληροφοριών από υπολογιστές
  - Όλες οι γλώσσες που παρουσιάσαμε έχουν καλά ορισμένη σύνταξη
  - Οι γλώσσες DAML+OIL και OWL βασίζονται στις γλώσσες RDF και RDFS και έχουν το ίδιο είδος σύνταξης

# Απαιτήσεις των Γλωσσών Οντολογιών (2/5)

- Βέβαια, υπάρχουν αμφιβολίες αν η σύνταξη της RDF που βασίζεται σε XML είναι πολύ φιλική προς το χρήστη
  - Υπάρχουν εναλλακτικές επιλογές που ταιριάζουν περισσότερο στους ανθρώπους-χρήστες
  - Ωστόσο, το μειονέκτημα αυτό δεν είναι ιδιαίτερα σημαντικό, επειδή οι χρήστες θα αναπτύσσουν τελικά τις δικές τους οντολογίες χρησιμοποιώντας εργαλεία συγγραφής, ή γενικότερα, *εργαλεία ανάπτυξης οντολογιών*
- Η τυπική σημασιολογία περιγράφει το νόημα της γνώσης με ακρίβεια
  - «Με ακρίβεια» σημαίνει ότι η σημασιολογία δεν αναφέρεται σε υποκειμενικές αντιλήψεις, ούτε και επιδέχεται διαφορετικές ερμηνείες από διαφορετικά άτομα

# Απαιτήσεις των Γλωσσών Οντολογιών (3/5)

- Μία χρήση της τυπικής σημασιολογίας είναι να επιτρέπει στους ανθρώπους να κάνουν συλλογισμούς σχετικά με τη γνώση
- Για οντολογική γνώση, μπορούν να γίνουν συλλογισμοί σχετικά με τα εξής:
  - **Συμμετοχή σε κλάσεις:** Αν το  $x$  είναι στιγμιότυπο μιας κλάσης  $C$ , και η  $C$  είναι υποκλάση της  $D$ , τότε μπορούμε να συμπεράνουμε ότι το  $x$  είναι στιγμιότυπο της  $D$
  - **Ισοδυναμία κλάσεων:** Αν η κλάση  $A$  είναι ισοδύναμη με την κλάση  $B$ , και η κλάση  $B$  είναι ισοδύναμη με την κλάση  $C$ , τότε και η  $A$  είναι επίσης ισοδύναμη με τη  $C$
  - **Συνέπεια:** Έστω ότι έχουμε δηλώσει ότι το  $x$  είναι στιγμιότυπο της κλάσης  $A$  και ότι η  $A$  είναι υποκλάση της τομής  $B \cap C$ , επίσης ότι η  $A$  είναι υποκλάση της  $D$ , και οι  $B$  και  $D$  είναι ξένες μεταξύ τους. Τότε θα έχουμε ασυνέπεια, επειδή η  $A$  θα έπρεπε να είναι κενή, αλλά περιέχει το στιγμιότυπο  $x$ . Αυτό αποτελεί ένδειξη σφάλματος στην οντολογία
  - **Ταξινόμηση:** Αν έχουμε δηλώσει ότι συγκεκριμένα ζεύγη ιδιότητας-τιμής αποτελούν ικανή συνθήκη για τη συμμετοχή σε μία κλάση  $A$ , τότε αν ένα μεμονωμένο στοιχείο  $x$  ικανοποιεί τέτοιες συνθήκες, μπορούμε να συμπεράνουμε ότι το  $x$  πρέπει να είναι στιγμιότυπο της  $A$

# Απαιτήσεις των Γλωσσών Οντολογιών (4/5)

- Η σημασιολογία αποτελεί προϋπόθεση για την υποστήριξη συλλογισμών
  - Τέτοια συμπεράσματα μπορούν να εξαχθούν αυτόματα παρά με μη αυτόματο τρόπο
- Η υποστήριξη συλλογισμών είναι σημαντική επειδή επιτρέπει
  - Έλεγχο της συνέπειας της οντολογίας και της γνώσης
  - Έλεγχο ανεπιθύμητων σχέσεων ανάμεσα σε κλάσεις
  - Αυτόματη ταξινόμηση των στιγμιοτύπων σε κλάσεις
- Η αυτοματοποιημένη υποστήριξη συλλογισμών παρέχει δυνατότητα ελέγχου πολύ περισσότερων περιπτώσεων από αυτές που θα μπορούσαν να ελεγχθούν με μη αυτόματο τρόπο
  - Τέτοιοι έλεγχοι είναι πολύτιμοι για το σχεδιασμό μεγάλων οντολογιών όπου συμμετέχουν πολλοί συγγραφείς, καθώς και για την ενοποίηση και το διαμοιρασμό οντολογιών από διάφορες πηγές

# Απαιτήσεις των Γλωσσών Οντολογιών (5/5)

- Η τυπική σημασιολογία και η υποστήριξη συλλογισμών παρέχονται συνήθως με
  - Την αντιστοίχιση μίας γλώσσας οντολογιών σε μία γνωστή λογική τυπολογία
  - Τη χρήση προγραμμάτων αυτοματοποιημένης συλλογιστικής που ήδη υπάρχουν για τις τυπολογίες αυτές
- Η γλώσσα **OWL** αντιστοιχίζεται (μερικώς) σε μία περιγραφική λογική και χρησιμοποιεί υπάρχοντα προγράμματα συλλογιστικής (π.χ. **FaCT** και **RACER**)
- Οι περιγραφικές λογικές αποτελούν ένα υποσύνολο της κατηγορηματικής λογικής, για το οποίο είναι δυνατό να υπάρξει αποδοτική υποστήριξη συλλογισμών



# Περιορισμοί της εκφραστικής ισχύος της RDF Schema (1/4)

- Οι γλώσσες RDF και RDFS επιτρέπουν την αναπαράσταση ενός μέρους της οντολογικής γνώσης
- Τα κύρια θεμελιώδη στοιχεία μοντελοποίησης των RDF/RDFS αφορούν την οργάνωση των λεξιλογίων σε τυποποιημένες ιεραρχίες:
  - Σχέσεις υποκλάσης και υποϊδιότητας
  - Περιορισμούς πεδίου ορισμού και συνόλου τιμών
  - Στιγμιότυπα κλάσεων
- Όμως, λείπουν αρκετές άλλες δυνατότητες:
  - **Τοπική εμβέλεια ιδιοτήτων**
    - Το `rdfs:range` ορίζει το σύνολο τιμών μιας ιδιότητας (π.χ. `eats`) για όλες τις κλάσεις
    - Επομένως, δεν μπορούμε στην RDF Schema να δηλώσουμε περιορισμούς στο σύνολο τιμών, που θα ισχύουν μόνο για μερικές κλάσεις
    - Π.χ. δεν μπορούμε να πούμε ότι οι αγελάδες τρώνε μόνο φυτά, ενώ άλλα ζώα μπορεί να τρώνε και κρέας

# Περιορισμοί της εκφραστικής ισχύος της RDF Schema (2/4)

- **Μη επικάλυψη κλάσεων**

- Μερικές φορές, θέλουμε να δηλώσουμε ότι ορισμένες κλάσεις είναι ξένες μεταξύ τους (π.χ. *male* και *female*)
- Αλλά στην RDF Schema μπορούμε να δηλώσουμε μόνο σχέσεις υποκλάσεων (π.χ. η *female* είναι υποκλάση της *person*)

- **Λογικοί συνδυασμοί κλάσεων**

- Μερικές φορές, θέλουμε να δημιουργήσουμε νέες κλάσεις, συνδυάζοντας άλλες κλάσεις με τη χρήση της ένωσης, της τομής και του συμπληρώματος
- Π.χ. έστω ότι θέλουμε να ορίσουμε ότι η κλάση *person* είναι η μη συμβιβαστή ένωση των κλάσεων *male* και *female*
- Τέτοιοι ορισμοί δεν επιτρέπονται στην RDF Schema

# Περιορισμοί της εκφραστικής ισχύος της RDF Schema (3/4)

- **Περιορισμοί πληθικότητας**
  - Μερικές φορές επιθυμούμε να επιβάλλουμε περιορισμούς στο πλήθος των διακριτών τιμών που μπορεί ή πρέπει να πάρει μια ιδιότητα
  - Π.χ. έστω ότι θέλουμε να δηλώσουμε ότι ένα άτομο έχει ακριβώς δύο γονείς ή ότι ένα μάθημα διδάσκεται από τουλάχιστον ένα διδάσκοντα
  - Και πάλι, τέτοιοι περιορισμοί δεν είναι δυνατό να εκφραστούν σε RDF Schema
- **Ειδικά χαρακτηριστικά ιδιοτήτων**
  - Μερικές φορές είναι χρήσιμο να δηλώσουμε ότι μία ιδιότητα είναι *μεταβατική* (π.χ. “greater than”), *μοναδική* (π.χ. “is mother of”) ή *αντίστροφη* (π.χ. “eats” και “is eaten by”) μιας άλλης ιδιότητας

# Περιορισμοί της εκφραστικής ισχύος της RDF Schema (4/4)

- Άρα χρειαζόμαστε μία γλώσσα οντολογιών που θα είναι πλουσιότερη από την RDF Schema
- Κατά το σχεδιασμό μιας τέτοιας γλώσσας, θα πρέπει να έχουμε υπόψη μας το αντίτιμο που έχει η εκφραστική ισχύς στην αποδοτική υποστήριξη συλλογισμών
  - Όσο πλουσιότερη είναι η γλώσσα, τόσο λιγότερο αποδοτική γίνεται η υποστήριξη συλλογισμών, ξεπερνώντας συχνά το όριο της μη υπολογισσιμότητας
- Επομένως, χρειάζεται ένας συμβιβασμός, δηλαδή μία γλώσσα που θα υποστηρίζεται από σχετικά αποδοτικά προγράμματα συλλογιστικής
  - Ενώ θα είναι και επαρκώς εκφραστική για να μπορεί να εκφράζει μεγάλες κλάσεις οντολογιών και γνώσης

# Συμβατότητα των OWL και RDF/RDFS

- Ιδανικά, η OWL θα ήταν επέκταση της RDF Schema
  - Με την έννοια ότι η OWL θα χρησιμοποιούσε τη σημασία της RDF για τις κλάσεις και τις ιδιότητες (*rdfs:Class*, *rdfs:subClassOf*, κλπ)
  - Και θα προσέθετε γλωσσικά θεμελιώδη στοιχεία για να υποστηρίξει τη μεγαλύτερη εκφραστικότητα που απαιτείται
- Μία τέτοια επέκταση της RDF Schema θα ήταν συνεπής και με τη διαστρωματωμένη αρχιτεκτονική του ΣI
- Δυστυχώς, η απλή επέκταση της RDF Schema θα λειτουργούσε ενάντια στην εκφραστική ισχύ και την αποδοτική συλλογιστική
- Η RDF Schema έχει ορισμένα πολύ ισχυρά θεμελιώδη στοιχεία μοντελοποίησης
  - Δομές όπως οι *rdfs:Class* (η κλάση όλων των κλάσεων) και *rdf:Property* (η κλάση όλων των ιδιοτήτων) είναι πολύ εκφραστικές και θα οδηγούσαν σε ανεξέλεγκτες υπολογιστικές ιδιότητες
    - Αν η επέκταση της λογικής γινόταν με τέτοια εκφραστικά θεμελιώδη στοιχεία



° ΤΡΕΙΣ ΥΠΟΓΛΩΣΣΕΣ  
ΤΗΣ OWL

# Τρεις Υπογλώσσες της OWL

- Το πλήρες σύνολο των απαιτήσεων για μία γλώσσα οντολογιών μοιάζει ανέφικτο:
  - Αποδοτική υποστήριξη συλλογισμών και ευκολία στην έκφραση για μία γλώσσα τόσο ισχυρή όσο είναι ο συνδυασμός της RDF Schema με μία πλήρη λογική
- Οι απαιτήσεις αυτές ώθησαν την Ομάδα Εργασίας Οντολογιών Ιστού του W3C να ορίσει την OWL ως τρεις διαφορετικές υπογλώσσες
  - Κάθε μία από τις οποίες έχει προσαρμοστεί να ανταποκρίνεται σε διαφορετικές πτυχές του πλήρους συνόλου των απαιτήσεων

# OWL Full (1/2)

- Η πλήρης γλώσσα ονομάζεται **OWL Full** και χρησιμοποιεί όλα τα γλωσσικά θεμελιώδη στοιχεία της **OWL**
- Επιτρέπει επίσης το συνδυασμό αυτών των θεμελιωδών στοιχείων με αυθαίρετους τρόπους, χρησιμοποιώντας τις γλώσσες **RDF** και **RDF Schema**
  - Αυτό περιλαμβάνει τη δυνατότητα (που υπάρχει και στην **RDF**) αλλαγής του νοήματος των προκαθορισμένων θεμελιωδών στοιχείων (της **RDF** ή της **OWL**) με την εφαρμογή των γλωσσικών θεμελιωδών στοιχείων μεταξύ τους
  - Π.χ. θα μπορούσαμε στην **OWL Full** να επιβάλλουμε έναν περιορισμό πληθικότητας στην κλάση όλων των κλάσεων, περιορίζοντας ουσιαστικά τον αριθμό των κλάσεων που μπορούν να περιγραφούν σε οποιαδήποτε οντολογία



# OWL Full (2/2)

- Το πλεονέκτημα της OWL Full είναι ότι είναι πλήρως συμβατή προς τα επάνω με την RDF, τόσο συντακτικά όσο και σημασιολογικά
  - Κάθε έγκυρο έγγραφο RDF είναι ταυτόχρονα και έγκυρο έγγραφο OWL Full
  - Και κάθε έγκυρο συμπέρασμα σε RDF/RDF Schema είναι επίσης έγκυρο συμπέρασμα σε OWL Full
- Το μειονέκτημα είναι ότι η γλώσσα έχει γίνει τόσο ισχυρή
  - ώστε να δημιουργεί πλέον διλήμματα στις αποφάσεις, εκμηδενίζοντας κάθε ελπίδα πλήρους (ή αποδοτικής) υποστήριξης συλλογισμών

# OWL DL

- Προκειμένου να ανακτηθεί η υπολογιστική αποδοτικότητα, η OWL DL (συντομογραφία του Description Logic) είναι μία υπογλώσσα της OWL Full
  - Που περιορίζει τον τρόπο χρήσης των constructors των OWL και RDF
- Ουσιαστικά δεν επιτρέπεται η εφαρμογή ενός constructor της OWL σε έναν άλλον
  - Διασφαλίζοντας έτσι ότι η γλώσσα αντιστοιχεί σε μία καλά ορισμένη περιγραφική λογική
- Το πλεονέκτημα είναι ότι επιτρέπεται η αποδοτική υποστήριξη συλλογισμών
- Το μειονέκτημα είναι ότι χάνεται η πλήρης συμβατότητα με την RDF
  - Ένα έγγραφο RDF θα χρειαστεί γενικά να επεκταθεί με ορισμένους τρόπους και να περιοριστεί με άλλους για να θεωρηθεί έγκυρο έγγραφο OWL DL
  - Κάθε έγκυρο έγγραφο OWL DL είναι και έγκυρο έγγραφο RDF

# OWL Lite (1/4)

- Ένας πρόσθετος περιορισμός περιορίζει την OWL DL σε ένα υποσύνολο των constructors της γλώσσας
  - Π.χ. η OWL Lite αποκλείει τις απαριθμητές κλάσεις, τις προτάσεις μη επικάλυψης και την αυθαίρετη πληθικότητα
- Το πλεονέκτημα είναι ότι η γλώσσα είναι ευκολότερη στην κατανόηση (για τους χρήστες) αλλά και στην υλοποίηση (για τους κατασκευαστές εργαλείων)
- Το μειονέκτημα είναι η περιορισμένη εκφραστικότητα

# OWL Lite (2/4)

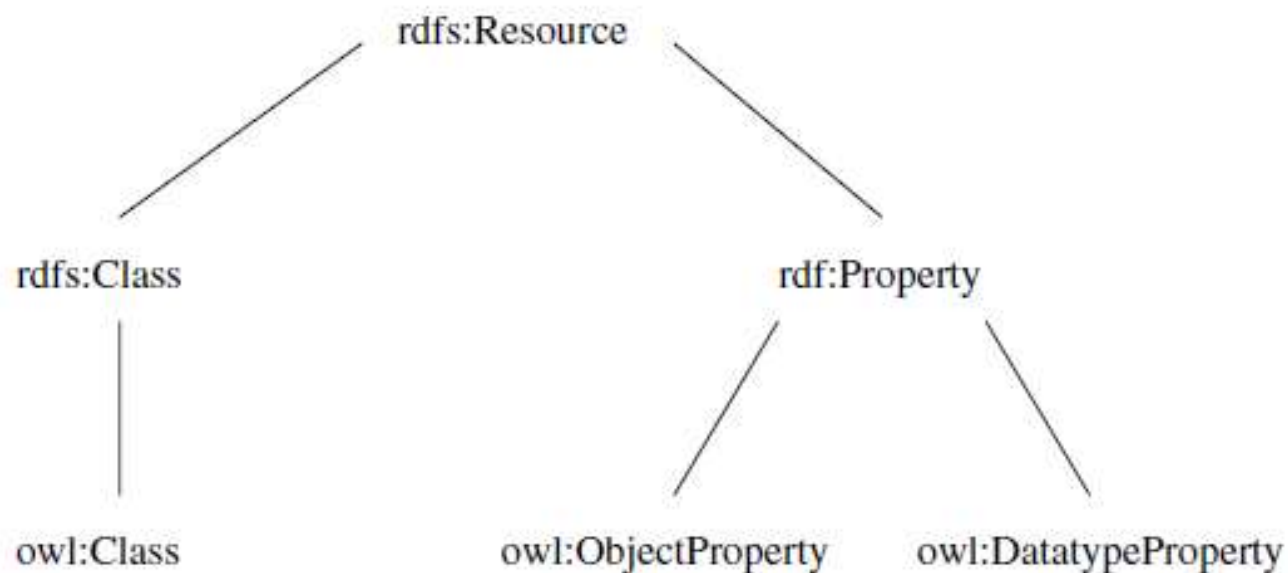
- Οι προγραμματιστές οντολογιών που υιοθετούν την OWL θα πρέπει να αναλογιστούν ποια υπογλώσσα ταιριάζει καλύτερα στις ανάγκες τους
- Η επιλογή ανάμεσα στην OWL Lite και την OWL DL εξαρτάται από το βαθμό απαίτησης των περισσότερο εκφραστικών δομών της OWL DL από τους χρήστες
- Η επιλογή ανάμεσα στην OWL DL και OWL Full εξαρτάται κυρίως από το βαθμό στον οποίο οι χρήστες απαιτούν τις λειτουργίες μεταμοντελοποίησης της RDF Schema (π.χ. ορισμός κλάσεων από κλάσεις ή προσάρτηση ιδιοτήτων στις κλάσεις)
- Συγκριτικά με την OWL DL, η υποστήριξη συλλογισμών κατά τη χρήση της OWL Full είναι λιγότερο προβλέψιμη, επειδή οι πλήρεις υλοποιήσεις σε OWL Full δεν είναι εφικτές

# OWL Lite (3/4)

- Υπάρχουν αυστηρά πλαίσια συμβατότητας προς τα επάνω ανάμεσα στις τρεις αυτές υπογλώσσες:
  - Κάθε έγκυρη οντολογία OWL Lite είναι και μία έγκυρη οντολογία OWL DL
  - Κάθε έγκυρη οντολογία OWL DL είναι και μία έγκυρη οντολογία OWL Full
  - Κάθε έγκυρο συμπέρασμα σε OWL Lite είναι και έγκυρο συμπέρασμα σε OWL DL
  - Κάθε έγκυρο συμπέρασμα σε OWL DL είναι και έγκυρο συμπέρασμα σε OWL Full
- Η OWL εξακολουθεί να χρησιμοποιεί σε μεγάλο βαθμό τις RDF και RDF Schema:
  - Όλες οι παραλλαγές της OWL χρησιμοποιούν την RDF για τη σύνταξή τους
  - Τα στιγμιότυπα δηλώνονται όπως και στην RDF, με τη χρήση περιγραφών RDF και πληροφοριών τυποποίησης
  - Οι constructors της OWL, όπως οι `owl:Class`, `owl:DatatypeProperty`, και `owl:ObjectProperty` είναι εξειδικεύσεις των αντίστοιχων κατασκευαστών της RDF

# OWL Lite (4/4)

- Ένα από τα κύρια κίνητρα πίσω από τη διαστρωματωμένη αρχιτεκτονική του ΣΙ είναι η ελπίδα για συμβατότητα προς τα κάτω, μαζί με την αντίστοιχη επαναχρησιμοποίηση του λογισμικού στα διάφορα επίπεδα
- Το πλεονέκτημα της πλήρους συμβατότητας προς τα κάτω για την OWL επιτυγχάνεται μόνο για την OWL Full, με τίμημα την υπολογιστική δυσεπιλυσιμότητα



Σχέσεις υποκλάσεων μεταξύ OWL και RDF/RDFS



° ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΓΛΩΣΣΑΣ  
**OWL**

# Σύνταξη

- Η OWL βασίζεται στις γλώσσες RDF και RDF Schema και χρησιμοποιεί τη σύνταξη της RDF που βασίζεται στην XML
- Άλλες συντακτικές μορφές που έχουν οριστεί για την OWL:
  - Μία σύνταξη που βασίζεται στην XML, η οποία δεν ακολουθεί τις συμβάσεις της RDF και είναι συνεπώς πιο ευανάγνωστη από τους ανθρώπους-χρήστες
  - Μία αφηρημένη σύνταξη που χρησιμοποιείται στο έγγραφο προδιαγραφών της γλώσσας, η οποία είναι πολύ πιο συμπαγής και ευανάγνωστη από τη σύνταξη XML αλλά και από τη σύνταξη RDF/XML
  - Μία γραφική σύνταξη που βασίζεται στις συμβάσεις της UML (Unified Modeling Language), η οποία χρησιμοποιείται ευρέως και αποτελεί έτσι έναν εύκολο τρόπο εξοικείωσης των ανθρώπων με την OWL



# Κεφαλίδα (1/2)

- Τα έγγραφα OWL αποκαλούνται συνήθως *οντολογίες OWL* και είναι έγγραφα RDF
- Το στοιχείο-ρίζα μιας οντολογίας OWL είναι ένα στοιχείο *rdf:RDF*
  - Που καθορίζει επίσης έναν αριθμό από χώρους ονομάτων:
- Μία οντολογία OWL μπορεί να ξεκινάει με μία συλλογή ισχυρισμών για λόγους «νοικοκυρέματος»
- Οι ισχυρισμοί αυτοί ομαδοποιούνται σε ένα στοιχείο *owl:Ontology*
  - Που περιέχει σχόλια, έλεγχο εκδόσεων, και προσθήκη άλλων οντολογιών

```
<rdf:RDF
  xmlns:owl ="http://www.w3.org/2002/07/owl#"
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd ="http://www.w3.org/2001/XMLSchema#">
```

```
<owl:Ontology rdf:about="">
  <rdfs:comment>An example OWL
  ontology</rdfs:comment>
  <owl:priorVersion
  rdf:resource="http://www.mydomain.org/uni-ns-old"/>
  <owl:imports
  rdf:resource="http://www.mydomain.org/persons"/>
  <rdfs:label>University Ontology</rdfs:label>
</owl:Ontology>
```

# Κεφαλίδα(2/2)

- Μόνο ένας από τους ισχυρισμούς αυτούς έχει συνέπειες για το λογικό νόημα της οντολογίας: *owl:imports*
  - Που απαριθμεί άλλες οντολογίες, το περιεχόμενο των οποίων αποτελεί μέρος της τρέχουσας οντολογίας
- Ενώ οι χώροι ονομάτων χρησιμοποιούνται για την εξάλειψη της αμφισημίας, οι εισαγόμενες οντολογίες παρέχουν ορισμούς που μπορούν όντως να χρησιμοποιηθούν
- Συνήθως, θα υπάρχει ένα σημείο εισαγωγής για κάθε χώρο ονομάτων που χρησιμοποιείται, αλλά είναι εφικτό να εισαχθούν πρόσθετες οντολογίες
  - Π.χ. οντολογίες που παρέχουν ορισμούς χωρίς να εισάγουν καινούρια ονόματα
- *owl:imports* : μεταβατική ιδιότητα
  - Αν η οντολογία *A* εισάγει την οντολογία *B*, και η οντολογία *B* εισάγει την οντολογία *C*, τότε η *A* εισάγει επίσης τη *C*

# Στοιχεία Κλάσεων (1/2)

- Οι κλάσεις ορίζονται με τη χρήση ενός στοιχείου *owl:Class*
  - Π.χ. μία κλάση *associateProfessor* μπορεί να οριστεί ως εξής:

```
<owl:Class rdf:ID="associateProfessor">  
<rdfs:subClassOf rdf:resource="#academicStaffMember"/>  
</owl:Class>
```

- Μπορούμε επίσης να δηλώσουμε ότι η κλάση αυτή είναι ξένη ως προς τις κλάσεις *assistantProfessor* και *professor* με τη χρήση στοιχείων *owl:disjointWith*
- Τα στοιχεία αυτά μπορούν να περιέχονται στον προηγούμενο ορισμό ή να προστεθούν με αναφορά στο ID χρησιμοποιώντας το *rdf:about*
- Ο μηχανισμός αυτός κληρονομείται από την RDF

```
<owl:Class rdf:about="#associateProfessor">  
<owl:disjointWith rdf:resource="#professor"/>  
<owl:disjointWith rdf:resource="#assistantProfessor"/>  
</owl:Class>
```

# Στοιχεία Κλάσεων (2/2)

- Η ισοδυναμία κλάσεων μπορεί να οριστεί μέσω του στοιχείου *owl:equivalentClass*:

```
<owl:Class rdf:ID="faculty">  
<owl:equivalentClass rdf:resource="#academicStaffMember"/>  
</owl:Class>
```

- Τέλος, υπάρχουν δύο προκαθορισμένες κλάσεις
  - *owl:Thing* : η πιο γενική κλάση που περιέχει τα πάντα
  - *owl:Nothing* : η κενή κλάση
- Έτσι, κάθε κλάση είναι υποκλάση της *owl:Thing* και υπερκλάση της *owl:Nothing*

# Στοιχεία Ιδιοτήτων (1/3)

- Στην OWL υπάρχουν δύο είδη ιδιοτήτων:
  - Ιδιότητες αντικειμένου, που συσχετίζουν αντικείμενα με άλλα αντικείμενα
    - Για παράδειγμα οι ιδιότητες *isTaughtBy* και *supervises*
  - Ιδιότητες τύπου δεδομένων, που συσχετίζουν αντικείμενα με τιμές ενός τύπου δεδομένων
    - Για παράδειγμα οι ιδιότητες *phone*, *title*, και *age*
    - Η OWL δεν έχει προκαθορισμένους τύπους δεδομένων, ούτε και παρέχει ειδικές λειτουργίες ορισμού
    - Αντιθέτως, επιτρέπει τη χρήση των τύπων δεδομένων της γλώσσας XML Schema, αξιοποιώντας έτσι τη διαστρωματωμένη αρχιτεκτονική του ΣΙ
    - Ακολουθεί ένα παράδειγμα ιδιότητας τύπου δεδομένων:

```
<owl:DatatypeProperty rdf:ID="age">  
<rdfs:range  
rdf:resource="http://www.w3.org/2001/XMLSchema  
#nonNegativeInteger"/>  
</owl:DatatypeProperty>
```

# Στοιχεία Ιδιοτήτων (2/3)

- Οι τύποι δεδομένων που ορίζονται από το χρήστη θα συλλέγονται συνήθως σε ένα σχήμα XML και θα χρησιμοποιούνται στη συνέχεια σε μία οντολογία OWL
- Π.χ. μιας ιδιότητας αντικειμένου:

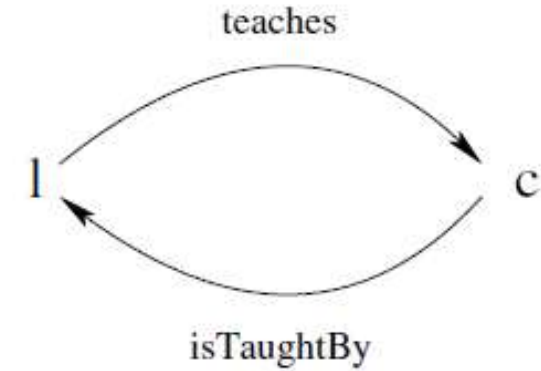
```
<owl:ObjectProperty rdf:ID="isTaughtBy">  
<rdfs:domain rdf:resource="#course"/>  
<rdfs:range rdf:resource="#academicStaffMember"/>  
<rdfs:subPropertyOf rdf:resource="#involves"/>  
</owl:ObjectProperty>
```

- Μπορούν να δηλωθούν περισσότερα από ένα πεδία ορισμού και σύνολα τιμών
  - Παίρνοντας την τομή των πεδίων ορισμού και των συνόλων τιμών, αντίστοιχα
- Η OWL επιτρέπει τη συσχέτιση αντίστροφων ιδιοτήτων
  - Π.χ. το ζεύγος *isTaughtBy* και *teaches*:

```
<owl:ObjectProperty rdf:ID="teaches">  
<rdfs:range rdf:resource="#course"/>  
<rdfs:domain rdf:resource="#academicStaffMember"/>  
<owl:inverseOf rdf:resource="#isTaughtBy"/>  
</owl:ObjectProperty>
```

# Στοιχεία Ιδιοτήτων (3/3)

- Η εικόνα απεικονίζει τη σχέση ανάμεσα σε μία ιδιότητα και την αντίστροφή της
  - Το πεδίο ορισμού και το σύνολο τιμών μπορούν να κληρονομηθούν από την αντίστροφη ιδιότητα (εναλλαγή του πεδίου ορισμού με το σύνολο τιμών)
- Η ισοδυναμία των ιδιοτήτων μπορεί να οριστεί με τη χρήση του στοιχείου *owl:equivalentProperty*



```
<owl:ObjectProperty rdf:ID="lecturesIn">  
<owl:equivalentProperty rdf:resource="#teaches"/>  
</owl:ObjectProperty>
```

# Περιορισμοί Ιδιοτήτων (1/7)

- Με την ιδιότητα *rdfs:subClassOf* μπορούμε να δηλώσουμε ότι η κλάση *C* είναι υποκλάση μιας άλλης κλάσης *C'*
  - Τότε κάθε στιγμιότυπο της *C* είναι και στιγμιότυπο της *C'*
- Έστω ότι θέλουμε να δηλώσουμε ότι η κλάση *C* ικανοποιεί συγκεκριμένες συνθήκες
  - Δηλαδή ότι όλα τα στιγμιότυπα της *C* ικανοποιούν τις συνθήκες
- Αυτό είναι ισοδύναμο με τη δήλωση ότι η *C* είναι υποκλάση μιας κλάσης *C'*
  - Όπου η *C'* συγκεντρώνει όλα τα αντικείμενα που ικανοποιούν τις συνθήκες
  - Το ίδιο ακριβώς συμβαίνει στην OWL
  - Γενικά, η *C'* μπορεί να παραμείνει ανώνυμη



# Περιορισμοί Ιδιοτήτων (2/7)

- Το ακόλουθο στοιχείο απαιτεί τα μαθήματα του 1<sup>ου</sup> έτους να διδάσκονται μόνο από καθηγητές :

```
<owl:Class rdf:about="#firstYearCourse">
  <rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#isTaughtBy"/>
    <owl:allValuesFrom rdf:resource="#Professor"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

- Το στοιχείο *owl:allValuesFrom* χρησιμοποιείται για τον ορισμό της κλάσης των δυνατών τιμών που μπορεί να πάρει η ιδιότητα που καθορίζεται από το στοιχείο *owl:onProperty*
  - Με άλλα λόγια, όλες οι τιμές της ιδιότητας πρέπει να προέρχονται από αυτή την κλάση
  - Στο παράδειγμά μας, μόνο οι καθηγητές μπορούν να είναι τιμές της ιδιότητας *isTaughtBy*

# Περιορισμοί Ιδιοτήτων (3/7)

- Μπορούμε να δηλώσουμε ότι τα μαθήματα των μαθηματικών διδάσκονται από τον David Billington ως εξής:
  - Το στοιχείο *owl:hasValue* δηλώνει μία συγκεκριμένη τιμή, την οποία πρέπει να έχει η ιδιότητα που καθορίζεται από το *owl:onProperty*
- Μπορούμε επίσης να δηλώσουμε ότι όλα τα μέλη του ακαδημαϊκού προσωπικού πρέπει να διδάσκουν τουλάχιστον ένα προπτυχιακό μάθημα:

```
<owl:Class rdf:about="#mathCourse">  
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#isTaughtBy"/>  
<owl:hasValue rdf:resource="#949318"/>  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

```
<owl:Class rdf:about="#academicStaffMember">  
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#teaches"/>  
<owl:someValuesFrom  
rdf:resource="#undergraduateCourse"/>  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

# Περιορισμοί Ιδιοτήτων (4/7)

- Ας συγκρίνουμε τα *owl:allValuesFrom* και *owl:someValuesFrom*
  - Το παράδειγμα που χρησιμοποιεί το πρώτο στοιχείο απαιτεί κάθε άτομο που διδάσκει ένα στιγμιότυπο της κλάσης (ένα μάθημα 1<sup>ου</sup> έτους) να είναι καθηγητής
    - Με όρους λογικής έχουμε μία καθολική ποσοτικοποίηση
  - Το παράδειγμα που χρησιμοποιεί το δεύτερο στοιχείο απαιτεί να υπάρχει ένα προπτυχιακό μάθημα, το οποίο θα διδάσκεται από από ένα στιγμιότυπο της κλάσης (ένα μέλος του ακαδημαϊκού προσωπικού)
    - Είναι πιθανό ο ίδιος ακαδημαϊκός να διδάσκει επιπρόσθετα και μεταπτυχιακά μαθήματα
    - Με όρους λογικής έχουμε μία υπαρξιακή ποσοτικοποίηση
- Γενικά, ένα στοιχείο *owl:Restriction* περιέχει ένα στοιχείο *owl:onProperty* και μία ή περισσότερες δηλώσεις περιορισμών
- Ένας τύπος δήλωσης περιορισμών ορίζει τους περιορισμούς στα είδη των τιμών που μπορεί να πάρει η ιδιότητα
  - *owl:allValuesFrom*
  - *owl:hasValue*
  - *owl:someValuesFrom*

# Περιορισμοί Ιδιοτήτων (5/7)

- Ένας άλλος τύπος καθορίζει περιορισμούς πληθικότητας
  - Π.χ. μπορούμε να απαιτήσουμε ότι κάθε μάθημα θα διδάσκεται από τουλάχιστον ένα άτομο:

```
<owl:Class rdf:about="#course">  
<rdfs:subClassOf>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#isTaughtBy"/>  
<owl:minCardinality  
rdf:datatype="&xsd;nonNegativeInteger">  
1  
</owl:minCardinality>  
</owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

- Σημειώστε ότι θα έπρεπε να καθορίσουμε ότι το λεκτικό “1” πρέπει να ερμηνευτεί ως *nonNegativeInteger* και ότι χρησιμοποιήσαμε τη δήλωση του χώρου ονομάτων *xsd* στο στοιχείο της κεφαλίδας για να αναφερθούμε στο έγγραφο XML Schema

# Περιορισμοί Ιδιοτήτων (6/7)

- Επίσης θα μπορούσαμε να δηλώσουμε ότι ένα τμήμα (department) πρέπει να έχει τουλάχιστον δέκα και το πολύ τριάντα μέλη:
- Είναι δυνατό να καθορίσουμε ένα ακριβή αριθμό
  - Π.χ. Ένας υποψήφιος διδάκτορας πρέπει να έχει ακριβώς δύο επιβλέποντες
  - Αυτό μπορεί να επιτευχθεί, χρησιμοποιώντας τον ίδιο αριθμό στα *owl:minCardinality* και *owl:maxCardinality*
  - Για ευκολία, η OWL παρέχει και το *owl:cardinality*

```
<owl:Class rdf:about="#department">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#hasMember"/>
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">
        10
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="#hasMember"/>
      <owl:maxCardinality
        rdf:datatype="&xsd;nonNegativeInteger">
        30
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

# Περιορισμοί Ιδιοτήτων (7/7)

- Το *owl:Restriction* ορίζει μία ανώνυμη κλάση, η οποία δεν έχει ID, δεν ορίζεται από το στοιχείο *owl:Class*, και έχει μόνο τοπική εμβέλεια:
  - Μπορεί να χρησιμοποιηθεί στο μοναδικό σημείο που εμφανίζεται ο περιορισμός
- Όταν αναφερόμαστε σε κλάσεις, πρέπει να έχουμε υπόψη το διττό τους νόημα:
  - Είναι κλάσεις που ορίζονται από το *owl:Class* και έχουν ID
  - Είναι τοπικές ανώνυμες κλάσεις ως συλλογές αντικειμένων που ικανοποιούν ορισμένες συνθήκες περιορισμών ή ως συνδυασμοί άλλων κλάσεων
    - Οι τελευταίες αποκαλούνται μερικές φορές παραστάσεις κλάσης

# Ειδικές Ιδιότητες

- Μερικές ιδιότητες των στοιχείων ιδιοτήτων μπορούν να οριστούν απευθείας:
  - owl:TransitiveProperty: ορίζει μία μεταβατική ιδιότητα, όπως οι “has better grade than”, “is taller than”, ή “is ancestor of”
  - owl:SymmetricProperty: : ορίζει μία συμμετρική ιδιότητα, όπως οι “has same grade as” ή “is sibling of”
  - owl:FunctionalProperty : ορίζει μία ιδιότητα που έχει το πολύ μία τιμή για κάθε αντικείμενο, όπως οι “age”, “height”, ή “directSupervisor”
  - owl:InverseFunctionalProperty : ορίζει μία ιδιότητα για την οποία δύο διαφορετικά αντικείμενα δεν μπορούν να έχουν την ίδια τιμή
    - Π.χ. η ιδιότητα “isTheSocialSecurityNumberfor” (ο αριθμός κοινωνικής ασφάλισής του, που αντιστοιχίζεται σε ένα μόνο άτομο)
- Ένα παράδειγμα των συντακτικών μορφών των παραπάνω:

```
<owl:ObjectProperty rdf:ID="hasSameGradeAs">  
<rdf:type rdf:resource="&owl;TransitiveProperty" />  
<rdf:type rdf:resource="&owl;SymmetricProperty" />  
<rdfs:domain rdf:resource="#student" />  
<rdfs:range rdf:resource="#student" />  
</owl:ObjectProperty>
```

# Λογικοί Συνδυασμοί (1/4)

- Είναι δυνατό να αναφερθούμε σε λογικούς συνδυασμούς (ένωση, τομή, συμπλήρωμα) των κλάσεων
  - Που ορίζονται είτε με το στοιχείο `owl:Class` ή με παραστάσεις κλάσεων
  - Π.χ. μπορούμε να αναφέρουμε ότι τα μαθήματα και τα μέλη του προσωπικού είναι ξένα μεταξύ τους:

```
<owl:Class rdf:about="#course">  
<rdfs:subClassOf>  
<owl:Class> <owl:complementOf rdf:resource="#staffMember"/> </owl:Class>  
</rdfs:subClassOf>  
</owl:Class>
```

- Αυτό αναφέρει ότι κάθε μάθημα είναι στιγμιότυπο του συμπληρώματος των μελών του προσωπικού, δηλαδή κανένα μάθημα δεν είναι μέλος του προσωπικού
- Η πρόταση αυτή θα μπορούσε επίσης να εκφραστεί με χρήση του στοιχείου `owl:disjointWith`



# Λογικοί Συνδυασμοί (2/4)

- Η ένωση των κλάσεων δημιουργείται χρησιμοποιώντας το στοιχείο `owl:unionOf`:

```
<owl:Class rdf:ID="peopleAtUni">  
<owl:unionOf rdf:parseType="Collection">  
<owl:Class rdf:about="#staffMember"/>  
<owl:Class rdf:about="#student"/>  
</owl:unionOf>  
</owl:Class>
```

- Αυτό δεν δηλώνει ότι η νέα κλάση είναι υποκλάση της ένωσης, αλλά ότι η νέα κλάση είναι *ίση* με την ένωση
  - Με άλλα λόγια, έχουμε δηλώσει μία *ισοδυναμία κλάσεων*
- Επίσης, δεν καθορίσαμε ότι οι δύο κλάσεις πρέπει να είναι ξένες:
  - Ένα μέλος του προσωπικού μπορεί να είναι και φοιτητής

# Λογικοί Συνδυασμοί (3/4)

- Η τομή δηλώνεται με το στοιχείο *owl:intersectionOf*:

```
<owl:Class rdf:ID="facultyInCS">  
<owl:intersectionOf rdf:parseType="Collection">  
<owl:Class rdf:about="#faculty"/>  
<owl:Restriction>  
<owl:onProperty rdf:resource="#belongsTo"/>  
<owl:hasValue rdf:resource="#CSDepartment"/>  
</owl:Restriction>  
</owl:intersectionOf>  
</owl:Class>
```

- Προσέξτε ότι έχουμε δημιουργήσει την τομή δύο κλάσεων, η μία από τις οποίες ορίστηκε ανώνυμα:
  - Η κλάση όλων των αντικειμένων που ανήκουν στο τμήμα Επιστήμης Υπολογιστών
  - Η κλάση αυτή τέμνεται με την κλάση *faculty* για να μας δώσει τελικά το διδακτικό προσωπικό του τμήματος Επιστήμης Υπολογιστών

# Λογικοί Συνδυασμοί (4/4)

- Η ένθεση των λογικών συνδυασμών μπορεί να γίνει αυθαίρετα
  - Το παράδειγμα ορίζει ότι το διοικητικό προσωπικό αποτελείται από εκείνα τα μέλη του προσωπικού που δεν είναι ούτε διδακτικό προσωπικό ούτε προσωπικό τεχνικής υποστήριξης:

```
<owl:Class rdf:ID="adminStaff">  
<owl:intersectionOf rdf:parseType="Collection">  
<owl:Class rdf:about="#staffMember"/>  
<owl:Class>  
<owl:complementOf>  
<owl:Class>  
<owl:unionOf rdf:parseType="Collection">  
<owl:Class rdf:about="#faculty"/>  
<owl:Class rdf:about="#techSupportStaff"/>  
</owl:unionOf>  
</owl:Class>  
</owl:complementOf>  
</owl:Class>  
</owl:intersectionOf>  
</owl:Class>
```

# Απαριθμήσεις

- Η απαρίθμηση είναι ένα στοιχείο *owl:oneOf* που χρησιμοποιείται για τον ορισμό μιας κλάσης με παράθεση όλων των στοιχείων της:

```
<owl:Class rdf:ID="weekdays">  
<owl:oneOf rdf:parseType="Collection">  
<owl:Thing rdf:about="#Monday"/>  
<owl:Thing rdf:about="#Tuesday"/>  
<owl:Thing rdf:about="#Wednesday"/>  
<owl:Thing rdf:about="#Thursday"/>  
<owl:Thing rdf:about="#Friday"/>  
<owl:Thing rdf:about="#Saturday"/>  
<owl:Thing rdf:about="#Sunday"/>  
</owl:oneOf>  
</owl:Class>
```

# Στιγμιότυπα (1/3)

- Τα στιγμιότυπα των κλάσεων δηλώνονται όπως και στην RDF:

```
<rdf:Description rdf:ID="949352">  
<rdf:type rdf:resource="#academicStaffMember"/>  
</rdf:Description>
```
- Ή ισοδύναμα:

```
<academicStaffMember rdf:ID="949352"/>
```
- Μπορούμε επίσης να παρέχουμε περισσότερες λεπτομέρειες:

```
<academicStaffMember rdf:ID="949352">  
<uni:age rdf:datatype="&xsd;integer">39</uni:age>  
</academicStaffMember>
```
- Σε αντίθεση με τα τυπικά συστήματα βάσεων δεδομένων, η OWL δεν υιοθετεί την υπόθεση μοναδικών ονομάτων (*unique-names assumption*)
  - Επειδή δύο στιγμιότυπα έχουν διαφορετικό όνομα ή ID δεν συνεπάγεται απαραίτητα ότι είναι δύο διαφορετικά μεμονωμένα στοιχεία

# Στιγμιότυπα (2/3)

- Π.χ. αν δηλώσουμε ότι κάθε μάθημα διδάσκεται από τουλάχιστον ένα μέλος του προσωπικού:

```
<owl:ObjectProperty rdf:ID="isTaughtBy">  
<rdf:type rdf:resource="#owl:FunctionalProperty" />  
</owl:ObjectProperty>
```

- Και δηλώσουμε στη συνέχεια ότι κάποιο συγκεκριμένο μάθημα διδάσκεται από 2 μέλη του προσωπικού:

```
<course rdf:ID="CIT1111">  
<isTaughtBy rdf:resource="#949318"/>  
<isTaughtBy rdf:resource="#949352"/>  
</course>
```

- Αυτό δεν θα προκαλέσει την επισήμανση σφάλματος από το πρόγραμμα συλλογιστικής OWL
- Εξάλλου, Το σύστημα θα μπορούσε να συμπεράνει έγκυρα ότι οι πόροι "949318" και "949352" είναι κατά τα φαινόμενα ίσοι
- Για να διασφαλίσουμε ότι τα διαφορετικά μεμονωμένα στοιχεία αναγνωρίζονται όντως ως διαφορετικά, πρέπει να διατυπώσουμε ρητά την ανισότητά τους:

```
<lecturer rdf:ID="949318"> <owl:differentFrom rdf:resource="#949352"/> </lecturer>
```

# Στιγμιότυπα (3/3)

- Επειδή τέτοιες προτάσεις ανισότητας προκύπτουν συχνά, και ο απαιτούμενος αριθμός αυτών των προτάσεων θα γινόταν τεράστιος αν θέλαμε να διατυπώσουμε την ανισότητα ενός μεγάλου αριθμού μεμονωμένων στοιχείων
  - Η OWL παρέχει μία σύντομη αναπαράσταση για τη δήλωση της ανισότητας ανά ζεύγη για όλα τα μεμονωμένα στοιχεία σε μια δεδομένη λίστα:

```
<owl:AllDifferent>
<owl:distinctMembers
rdf:parseType="Collection">
<lecturer rdf:about="#949318"/>
<lecturer rdf:about="#949352"/>
<lecturer rdf:about="#949111"/>
</owl:distinctMembers>
</owl:AllDifferent>
```

- Παρατηρήστε ότι το στοιχείο *owl:distinctMembers* μπορεί να χρησιμοποιηθεί μόνο σε συνδυασμό με το *owl:allDifferent*

# Τύποι Δεδομένων

- Παρόλο που η γλώσσα XML Schema παρέχει ένα μηχανισμό δημιουργίας τύπων δεδομένων που ορίζονται από το χρήστη
  - Π.χ. ο τύπος `adultAge` περιλαμβάνει όλους τους ακέραιους που είναι μεγαλύτεροι του 18
- Τέτοιοι παράγωγοι τύποι δεδομένων δεν μπορούν να χρησιμοποιηθούν στην OWL
- Στην πραγματικότητα, το ίδιο ισχύει και για τους πολλούς ενσωματωμένους τύπους δεδομένων της XML Schema
- Το έγγραφο αναφοράς της OWL αναφέρει όλους τους τύπους δεδομένων της XML Schema που επιτρέπονται
  - όπως `string`, `integer`, `Boolean`, `time`, και `date`



# Πληροφορίες Έκδοσης (1/2)

- Εκτός από την *owl:priorVersion* (χρησιμοποιείται ως μέρος των πληροφοριών της κεφαλίδας για την ένδειξη παλαιότερων εκδόσεων της τρέχουσας οντολογίας), η OWL διαθέτει τρεις ακόμα προτάσεις για την ένδειξη περισσότερων ανεπίσημων πληροφοριών έκδοσης
- Καμία απ' αυτές δεν έχει κάποιο τυπικό νόημα
  - ***owl:versionInfo***: γενικά περιέχει ένα αλφαριθμητικό με πληροφορίες σχετικά με την τρέχουσα έκδοση (όπως λέξεις-κλειδιά RCS/CVS)

# Πληροφορίες Έκδοσης (2/2)

• ...

- **owl:backwardCompatibleWith:** περιέχει αναφορά σε κάποια άλλη οντολογία
  - Αυτό προσδιορίζει την καθορισμένη οντολογία ως μία προγενέστερη έκδοση της οντολογίας στην οποία περιέχεται και υποδεικνύει επιπλέον ότι είναι συμβατή προς τα πίσω (backward-compatible) με αυτή
  - Συγκεκριμένα, αυτό υποδηλώνει ότι όλα τα αναγνωριστικά από την προηγούμενη έκδοση έχουν τις ίδιες ερμηνείες και στη νέα έκδοση
  - Έτσι, είναι μία υπόδειξη προς τους συγγραφείς των εγγράφων ότι μπορούν να τροποποιήσουν τα έγγραφά τους με ασφάλεια για να υπακούν στην καινούρια έκδοση
- **owl:incompatibleWith:** υποδηλώνει ότι η αναφερόμενη οντολογία αποτελεί προγενέστερη έκδοση της οντολογίας στην οποία περιέχεται αλλά δεν είναι συμβατή προς τα πίσω με αυτή
  - Αυτό χρησιμεύει στους συγγραφείς οντολογιών, οι οποίοι θέλουν να δηλώσουν ρητά ότι τα έγγραφα δεν μπορούν να αναβαθμιστούν για να χρησιμοποιούν τη νέα έκδοση, χωρίς προηγουμένως να γίνει έλεγχος αν απαιτούνται αλλαγές



# ΔΙΑΣΤΡΩΜΑΤΩΣΗ ΤΗΣ OWL

Τώρα μπορούμε να καθορίσουμε πλήρως ποιες δυνατότητες της γλώσσας μπορούν να χρησιμοποιηθούν σε ποια υπογλώσσα

# OWL Full

- Στην OWL Full, όλοι οι «κατασκευαστές» γλώσσας μπορούν να χρησιμοποιηθούν με οποιονδήποτε συνδυασμό, με την προϋπόθεση ότι το αποτέλεσμα είναι έγκυρο σε RDF

# OWL DL (1/2)

- Για να αξιοποιηθούν οι τυπικές βάσεις και η υπολογιστική επιλυσιμότητα της περιγραφικής λογικής, πρέπει να τηρούνται οι εξής περιορισμοί σε μία οντολογία OWL DL :
  - **Διαμέριση λεξιλογίου (Vocabulary partitioning)**: Κάθε πόρος επιτρέπεται να είναι μόνο ένα από τα εξής: κλάση, τύπος δεδομένων, ιδιότητα τύπου δεδομένων, ιδιότητα αντικειμένου, μεμονωμένο στοιχείο, τιμή δεδομένων, ή τμήμα του ενσωματωμένου λεξιλογίου
    - Που σημαίνει ότι, για παράδειγμα, μία κλάση δεν μπορεί να είναι ταυτόχρονα και μεμονωμένο στοιχείο, ή ότι μία ιδιότητα δεν μπορεί να έχει μερικές τιμές από έναν τύπο δεδομένων και μερικές από μία κλάση
  - **Ρητή Τυποποίηση (Explicit typing)**: όχι μόνο πρέπει όλοι οι πόροι να έχουν διαμεριστεί, αλλά η διαμέριση πρέπει να δηλώνεται ρητά
    - Π.χ. αν μία οντολογία περιέχει τα εξής:

```
<owl:Class rdf:ID="C1"> <rdfs:subClassOf rdf:about="#C2" /> </owl:Class>
```

- Αυτό συνεπάγεται ήδη ότι η κλάση C2 είναι κλάση (λόγω του καθορισμού του συνόλου τιμών της ιδιότητας rdfs:subClassOf)
- Όμως, μία οντολογία OWL DL πρέπει να περιέχει ρητά την εξής πληροφορία: 

```
<owl:Class rdf:ID="C2"/>
```

# OWL DL (2/2)

- ...
  - **Διαχωρισμός Ιδιοτήτων:** λόγω του πρώτου περιορισμού, τα σύνολα των ιδιοτήτων αντικειμένου και των ιδιοτήτων τύπου δεδομένων είναι ξένα μεταξύ τους
    - Αυτό συνεπάγεται ότι τα παρακάτω δεν μπορούν ποτέ να οριστούν για ιδιότητες τύπου δεδομένων:
      - `owl:inverseOf`
      - `owl:FunctionalProperty`
      - `owl:InverseFunctionalProperty`
      - `owl:SymmetricProperty`
  - **Απουσία μεταβατικών περιορισμών πληθικότητας:** Οι περιορισμοί πληθικότητας δεν μπορούν να εφαρμοστούν σε μεταβατικές ιδιότητες
    - Ή στις υπεριδιότητές τους, οι οποίες είναι κι αυτές μεταβατικές λόγω συνεπαγωγής
  - **Περιορισμένες ανώνυμες κλάσεις:** Οι ανώνυμες κλάσεις επιτρέπεται να εμφανίζονται μόνο ως πεδίο ορισμού και σύνολο τιμών είτε του στοιχείου `owl:equivalentClass` ή του `owl:disjointWith`, καθώς και ως σύνολο τιμών της ιδιότητας `rdfs:subClassOf`

# OWL Lite

- Μία οντολογία OWL Lite πρέπει να είναι οντολογία OWL DL και πρέπει να ικανοποιεί επιπλέον και τους εξής περιορισμούς:
  - Οι «κατασκευαστές» *owl:oneOf*, *owl:disjointWith*, *owl:unionOf*, *owl:complementOf*, και *owl:hasValue* δεν επιτρέπονται
  - Οι προτάσεις πληθικότητας (ελάχιστο, μέγιστο και ακριβές πλήθος) μπορούν να διατυπώνονται μόνο για τις τιμές 0 ή 1 και όχι πλέον για αυθαίρετους μη αρνητικούς ακεραίους
  - Οι προτάσεις *owl:equivalentClass* δεν μπορούν πλέον να διατυπώνονται μεταξύ ανωνύμων κλάσεων αλλά μόνο ανάμεσα σε αναγνωριστικά κλάσεων

# OWL DLP (1/5)

- Η OWL, όπως είδαμε, βασίζεται στην περιγραφική λογική
- Αφού η τελευταία αποτελεί τμήμα της λογικής πρώτης τάξης, κληρονομεί πολλές από τις ιδιότητες της συγκεκριμένης λογικής
  - Συγκεκριμένα, κληρονομεί την υπόθεση του ανοιχτού κόσμου και την υπόθεση του μη μοναδικού ονόματος
- Τα παραπάνω έχουν οδηγήσει στον ορισμό μιας ενδιαφέρουσας υπογλώσσας της OWL DL με την ονομασία OWL DLP
  - Η OWL DLP δεν ανήκει στα επίσημα είδη OWL του W3C
- Σύμφωνα με την *υπόθεση του ανοιχτού κόσμου* δεν μπορούμε να συμπεράνουμε ότι μία πρόταση  $x$  είναι ψευδής, απλώς επειδή δεν μπορούμε να αποδείξουμε ότι η  $x$  είναι αληθής
  - Εξάλλου, τα αξιώματά μας ενδέχεται να μην είναι δεσμευτική, όσον αφορά την κατάσταση αλήθειας της πρότασης  $x$
  - Με άλλα λόγια, δεν μπορούμε να συμπεράνουμε το ψεύδος από την απουσία της αλήθειας
- Η αντίθετη υπόθεση (υπόθεση του κλειστού κόσμου) θα μας επέτρεπε να συνάγουμε το ψεύδος εξαιτίας της αδυναμίας να συνάγουμε την αλήθεια



# OWL DLP (2/5)

- Ακολουθεί ένα παράδειγμα, όπου αυτό παίζει σημαντικό ρόλο:

**Question:** “Did it rain in Tokyo yesterday?”

**Answer:** “I don’t know that it rained, but that’s not enough reason to conclude that it didn’t rain.”

- Είναι σαφές ότι η υπόθεση του ανοιχτού κόσμου είναι κατάλληλη για την ερώτηση αυτή
- Έστω, τα εξής:

**Question:** “Was there a big earthquake disaster in Tokyo yesterday?”

**Answer:** “I don’t know that there was, but if there had been such a disaster, I’d have heard about it. Therefore I conclude that there wasn’t such a disaster.”

- Η υπόθεση του κλειστού κόσμου είναι κατάλληλη γι’ αυτήν την ερώτηση και μας επιτρέπει να εξάγουμε το λογικό συμπέρασμα που θα είχε μπλοκαριστεί από την υπόθεση του ανοιχτού κόσμου
  - Άρα, το γεγονός ότι η OWL είναι αυστηρά δεσμευμένη από την υπόθεση του ανοιχτού κόσμου δεν είναι η σωστή επιλογή σε ορισμένες εφαρμογές

# OWL DLP (3/5)

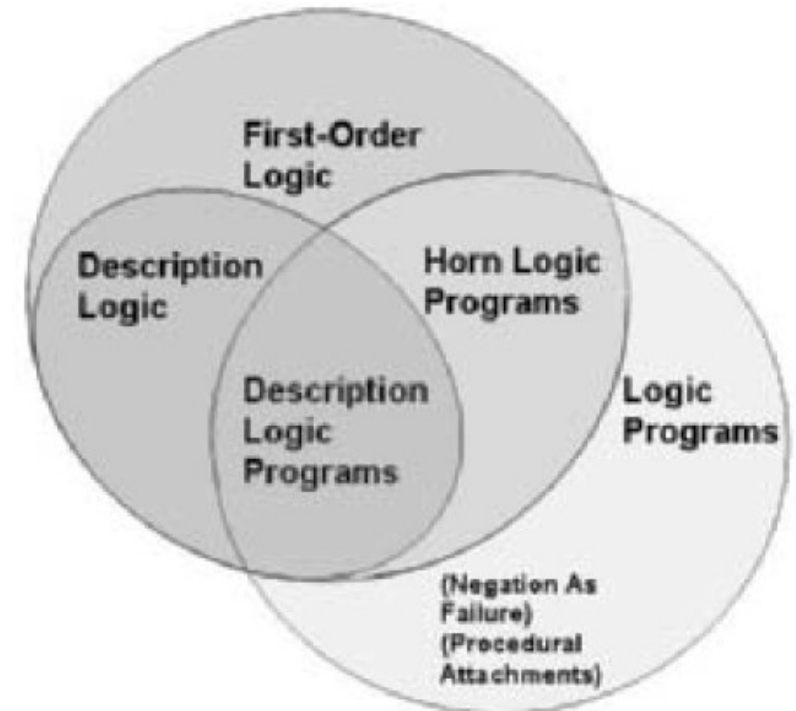
- Σύμφωνα με την υπόθεση του μοναδικού ονόματος, όταν δύο μεμονωμένα στοιχεία είναι γνωστά με διαφορετικά ονόματα, τότε αποτελούν στην πραγματικότητα διαφορετικά μεμονωμένα στοιχεία
- Και πάλι, αυτή είναι μία υπόθεση που άλλες φορές λειτουργεί σωστά και άλλες όχι
  - Όταν αναφερόμαστε σε κωδικούς προϊόντων ενός καταλόγου, είναι λογικό να υποθέσουμε ότι, όταν δύο προϊόντα είναι γνωστά με διαφορετικούς κωδικούς, τότε αποτελούν όντως διαφορετικά προϊόντα
  - Αλλά όταν αναφερόμαστε σε ανθρώπους του κοινωνικού περιβάλλοντός μας, τυχάνει συχνά να ανακαλύπτουμε αργότερα ότι δύο άτομα με διαφορετικά αναγνωριστικά είναι στην πραγματικότητα το ίδιο άτομο
    - Π.χ. “Καθ. van Harmelen” και “Frank”
- Η **OWL** δεν χρησιμοποιεί την υπόθεση του μοναδικού ονόματος, παρόλο που είναι δυνατό να επιβάλλουμε ρητά ένα σύνολο αναγνωριστικών, τα οποία είναι όλα μοναδικά
  - Με χρήση του στοιχείου *owl:allDifferent*

# OWL DLP (4/5)

- Συστήματα όπως οι βάσεις δεδομένων και τα συστήματα λογικού προγραμματισμού έχουν την τάση να υποστηρίζουν τους κλειστούς κόσμους και τα μοναδικά ονόματα
  - Ενώ τα συστήματα αναπαράστασης γνώσης και απόδειξης θεωρημάτων υποστηρίζουν τους ανοιχτούς κόσμους και τα μη μοναδικά ονόματα
- Οι οντολογίες χρειάζονται μερικές φορές το πρώτο ζευγάρι και άλλες φορές το δεύτερο
- Η αντιπαράθεση αυτή λύθηκε με τον καλύτερο τρόπο από τους Volz και Horrocks που προσδιόρισαν ένα τμήμα της OWL που ονομάζεται **DLP (Description Logic Programming)**
  - Το τμήμα αυτό είναι το μεγαλύτερο τμήμα στο οποίο δεν έχει σημασία η επιλογή για τις υποθέσεις του κλειστού κόσμου και του μοναδικού ονόματος
    - Δηλαδή η OWL DLP είναι αρκετά αδύναμη ώστε να μην γίνονται εμφανής οι διαφορές ανάμεσα στις επιλογές
  - Το πλεονέκτημα είναι ότι οι άνθρωποι ή οι εφαρμογές που επιθυμούν να κάνουν διαφορετικές επιλογές σχετικά με τις υποθέσεις αυτές, μπορούν να ανταλλάσσουν οντολογίες σε OWL DLP χωρίς πρόβλημα
  - Φυσικά, μόλις ξεφύγουν απ' τα πλαίσια της OWL DLP, θα παρατηρήσουν ότι εξάγουν διαφορετικά συμπεράσματα από τις ίδιες προτάσεις
    - Δηλαδή, ότι διαφωνούν στη σημασιολογία

# OWL DLP (5/5)

- Ευτυχώς, η DLP εξακολουθεί να είναι αρκετά μεγάλη για να επιτρέπει χρήσιμες λειτουργίες αναπαράστασης και συλλογιστικής
- Επιτρέπει τη χρήση «κατασκευαστών» της OWL
  - Ισοδυναμία κλάσεων και ιδιοτήτων
  - Ισότητα και ανισότητα μεταξύ μεμονωμένων στοιχείων
  - Αντίστροφες, μεταβατικές, συμμετρικές και συναρτησιακές ιδιότητες
  - .
- Όμως, αποκλείει «κατασκευαστές» όπως είναι η τομή και οι αυθαίρετοι περιορισμοί πληθικότητας
  - Οι «κατασκευαστές» αυτοί παρέχουν τη δυνατότητα για χρήσιμη εκφραστικότητα σε πολλές πρακτικές περιπτώσεις, ενώ εγγυώνται και τη σωστή ανταλλαγή ανάμεσα στα προγράμματα συλλογιστικής της OWL
  - Επιπλέον, επιτρέπουν τη μετάφραση σε τεχνικές συλλογιστικής που μπορούν να υλοποιηθούν αποδοτικά και βασίζονται σε βάσεις δεδομένων και λογικά προγράμματα

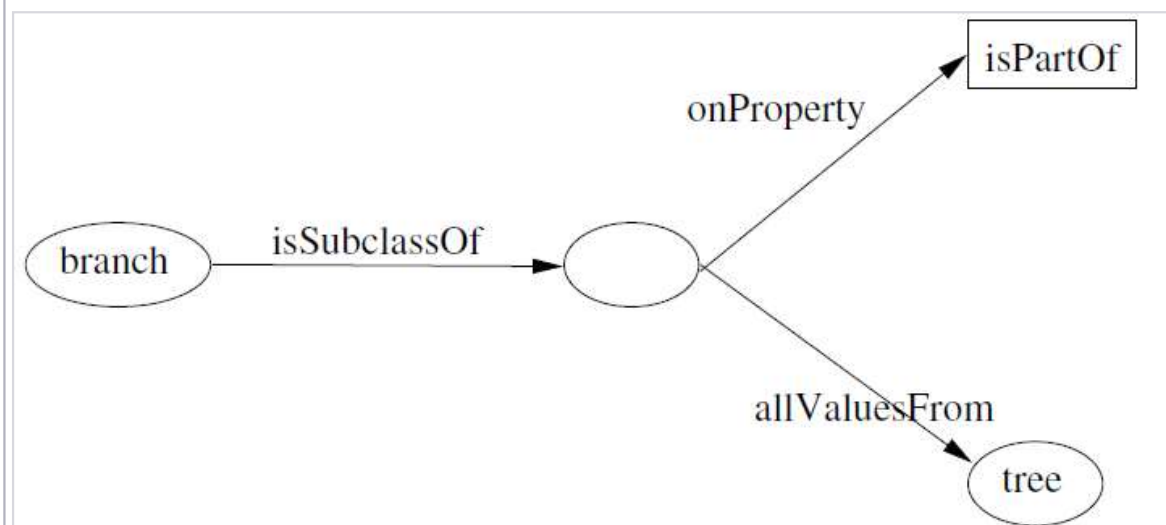
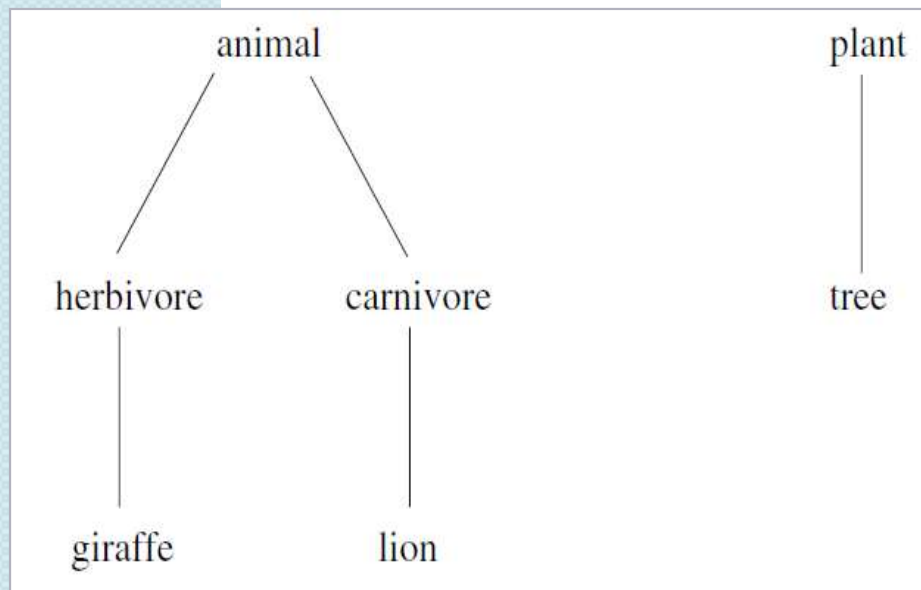




# ΠΑΡΑΔΕΙΓΜΑΤΑ

# Μια οντολογία για την αφρικανική άγρια πανίδα (1/7)

- Η ακόλουθη οντολογία περιγράφει την αφρικανική άγρια πανίδα
- Στην εικόνα φαίνονται οι βασικές κλάσεις και οι σχέσεις υποκλάσεων αυτών
- Οι πληροφορίες των υποκλάσεων αποτελούν μέρος μόνο των πληροφοριών που περιλαμβάνονται στην οντολογία
- Ο ολοκληρωμένος γράφος είναι πολύ μεγαλύτερος
  - Στην εικόνα βλέπουμε την γραφική αναπαράσταση της πρότασης ότι τα κλαδιά αποτελούν τμήμα των δένδρων
- Η οντολογία περιέχει σχόλια, με τη χρήση της *rdfs:comment*



# Μια οντολογία για την αφρικανική άγρια πανίδα (2/7)

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#">

<owl:Ontology rdf:about="xml:base"/>

<owl:Class rdf:ID="animal">
<rdfs:comment>Animals form a class.</rdfs:comment>
</owl:Class>

<owl:Class rdf:ID="plant">
<rdfs:comment>
Plants form a class disjoint from animals.
</rdfs:comment>
<owl:disjointWith rdf:resource="#animal"/>
</owl:Class>

<owl:Class rdf:ID="tree">
<rdfs:comment>Trees are a type of plant.</rdfs:comment>
<rdfs:subClassOf rdf:resource="#plant"/>
</owl:Class>
```

# Μια οντολογία για την αφρικανική άγρια πανίδα (3/7)

- ...

```
<owl:Class rdf:ID="branch">
<rdfs:comment>Branches are parts of trees.</rdfs:comment>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#is_part_of"/>
<owl:allValuesFrom rdf:resource="#tree"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="leaf">
<rdfs:comment>Leaves are parts of branches.</rdfs:comment>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#is_part_of"/>
<owl:allValuesFrom rdf:resource="#branch"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```



# Μια οντολογία για την αφρικανική άγρια πανίδα (4/7)

• ...

```
<owl:Class rdf:ID="herbivore">
<rdfs:comment> Herbivores are exactly those animals that eat only plants or parts of plants.
</rdfs:comment>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#animal"/>
<owl:Restriction>
<owl:onProperty rdf:resource="#eats"/>
<owl:allValuesFrom>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#plant"/>
<owl:Restriction> <owl:onProperty rdf:resource="#is_part_of"/>
<owl:allValuesFrom rdf:resource="#plant"/>
</owl:Restriction>
</owl:unionOf>
</owl:Class>
</owl:allValuesFrom>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>
```

# Μια οντολογία για την αφρικανική άγρια πανίδα (5/7)

• ...

```
<owl:Class rdf:ID="carnivore">
<rdfs:comment> Carnivores are exactly those animals that eat animals. </rdfs:comment>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#animal"/>
<owl:Restriction>
<owl:onProperty rdf:resource="#eats"/>
<owl:someValuesFrom rdf:resource="#animal"/>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="giraffe">
<rdfs:comment> Giraffes are herbivores, and they eat only leaves. </rdfs:comment>
<rdfs:subClassOf rdf:resource="#herbivore"/>
<rdfs:subClassOf>
<owl:Restriction> <owl:onProperty rdf:resource="#eats"/>
<owl:allValuesFrom rdf:resource="#leaf"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

# Μια οντολογία για την αφρικανική άγρια πανίδα (6/7)

- ...

```
<owl:Class rdf:ID="lion">
<rdfs:comment> Lions are animals that eat only herbivores. </rdfs:comment>
<rdfs:subClassOf rdf:resource="#carnivore"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#eats"/>
<owl:allValuesFrom rdf:resource="#herbivore"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="tasty_plant">
<rdfs:comment> Tasty plants are plants that are eaten both by herbivores and carnivores.
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#plant"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#eaten_by"/>
<owl:someValuesFrom>
```

# Μια οντολογία για την αφρικανική άγρια πανίδα (7/7)

• ...

```
<owl:Class rdf:about="#herbivore"/>
</owl:someValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction> <owl:onProperty rdf:resource="#eaten_by"/>
<owl:someValuesFrom> <owl:Class rdf:about="#carnivore"/> </owl:someValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:TransitiveProperty rdf:ID="is_part_of"/>

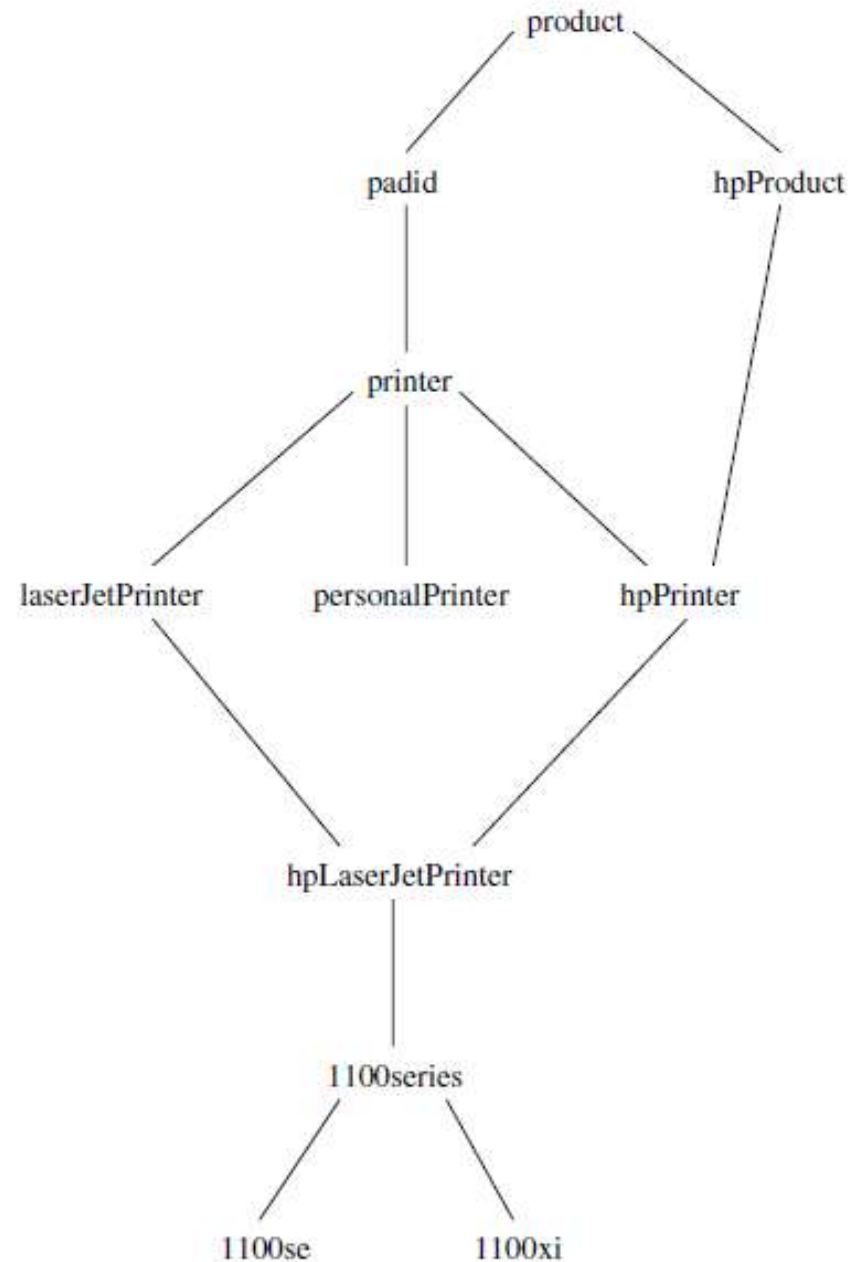
<owl:ObjectProperty rdf:ID="eats">
<rdfs:domain rdf:resource="#animal"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="eaten_by">
<owl:inverseOf rdf:resource="#eats"/>
</owl:ObjectProperty>

</rdf:RDF>
```

# Μία οντολογία εκτυπωτών (1/9)

- Κλάσεις και σχέσεις υποκλάσεων της οντολογίας εκτυπωτών



# Μία οντολογία εκτυπωτών (2/9)

```
<!DOCTYPE owl [  
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >  
>  
  
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"  
  xmlns:owl="http://www.w3.org/2002/07/owl#"  
  xmlns="http://www.cs.vu.nl/~frankh/spool/printer.owl#">  
  
  <owl:Ontology rdf:about="">  
    <owl:versionInfo>  
      My example version 1.2, 17 October 2002  
    </owl:versionInfo>  
  </owl:Ontology>  
  
  <owl:Class rdf:ID="product">  
    <rdfs:comment>Products form a class.</rdfs:comment>  
  </owl:Class>
```

# Μία οντολογία εκτυπωτών (3/9)

• ...

```
<owl:Class rdf:ID="padid">
```

```
<rdfs:comment>
```

```
Printing and digital imaging devices form a subclass of products.
```

```
</rdfs:comment>
```

```
<rdfs:label>Device</rdfs:label>
```

```
<rdfs:subClassOf rdf:resource="#product"/>
```

```
</owl:Class>
```

```
<owl:Class rdf:ID="hpProduct">
```

```
<rdfs:comment>
```

```
HP products are exactly those products that are manufactured by Hewlett Packard.
```

```
</rdfs:comment>
```

```
<owl:intersectionOf rdf:parseType="Collection">
```

```
<owl:Class rdf:about="#product"/>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="#manufactured_by"/>
```

```
<owl:hasValue rdf:datatype="xsd:string"> Hewlett Packard </owl:hasValue>
```

```
</owl:Restriction>
```

```
</owl:intersectionOf>
```

```
</owl:Class>
```

# Μία οντολογία εκτυπωτών (4/9)

• ...

```
<owl:Class rdf:ID="printer">
<rdfs:comment>
Printers are printing and digital imaging devices.
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#padid"/>
</owl:Class>

<owl:Class rdf:ID="personalPrinter">
<rdfs:comment>
Printers for personal use form a subclass of printers.
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#printer"/>
</owl:Class>

<owl:Class rdf:ID="hpPrinter">
<rdfs:comment>
HP printers are HP products and printers.
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#printer"/>
<rdfs:subClassOf rdf:resource="#hpProduct"/>
</owl:Class>
```



# Μία οντολογία εκτυπωτών (5/9)

• ...

```
<owl:Class rdf:ID="laserJetPrinter">
<rdfs:comment>
Laser jet printers are exactly those printers that use laser jet printing technology.
</rdfs:comment>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#printer"/>
<owl:Restriction>
<owl:onProperty rdf:resource="#printingTechnology"/>
<owl:hasValue rdf:datatype="xsd:string">
laser jet
</owl:hasValue>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="hpLaserJetPrinter">
<rdfs:comment> HP laser jet printers are HP products and laser jet printers. </rdfs:comment>
<rdfs:subClassOf rdf:resource="#laserJetPrinter"/>
<rdfs:subClassOf rdf:resource="#hpPrinter"/>
</owl:Class>
```

# Μία οντολογία εκτυπωτών (6/9)

• ...

```
<owl:Class rdf:ID="I100series">
  <rdfs:comment>
    I100series printers are HP laser jet printers with
    8ppm printing speed and 600dpi printing resolution.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#hpLaserJetPrinter"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#printingSpeed"/>
      <owl:hasValue rdf:datatype="&xsd:string"> 8ppm </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#printingResolution"/>
      <owl:hasValue rdf:datatype="&xsd:string"> 600dpi </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

# Μία οντολογία εκτυπωτών (7/9)

• ...

```
<owl:Class rdf:ID="I100se">
<rdfs:comment> I100se printers belong to the I100 series and cost $450. </rdfs:comment>
<rdfs:subClassOf rdf:resource="#I100series"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#price"/>
<owl:hasValue rdf:datatype="&xsd;integer"> 450 </owl:hasValue>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="I100xi">
<rdfs:comment> I100xi printers belong to the I100 series and cost $350. </rdfs:comment>
<rdfs:subClassOf rdf:resource="#I100series"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#price"/>
<owl:hasValue rdf:datatype="&xsd;integer"> 350 </owl:hasValue>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

# Μία οντολογία εκτυπωτών (8/9)

• ...

```
<owl:DatatypeProperty rdf:ID="manufactured_by">  
<rdfs:domain rdf:resource="#product"/>  
<rdfs:range rdf:resource="&xsd:string"/>  
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="price">  
<rdfs:domain rdf:resource="#product"/>  
<rdfs:range rdf:resource="&xsd;nonNegativeInteger"/>  
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="printingTechnology">  
<rdfs:domain rdf:resource="#printer"/>  
<rdfs:range rdf:resource="&xsd:string"/>  
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="printingResolution">  
<rdfs:domain rdf:resource="#printer"/>  
<rdfs:range rdf:resource="&xsd:string"/>  
</owl:DatatypeProperty>
```

# Μία οντολογία εκτυπωτών (9/9)

• ...

```
<owl:DatatypeProperty rdf:ID="printingSpeed">  
<rdfs:domain rdf:resource="#printer"/>  
<rdfs:range rdf:resource="&xsd:string"/>  
</owl:DatatypeProperty>  
  
</rdf:RDF>
```

- Αυτή η οντολογία δείχνει ότι τα «αδέρφια» σε ένα δένδρο ιεραρχίας δεν χρειάζεται να είναι ξένα μεταξύ τους
  - Π.χ. ένας προσωπικός εκτυπωτής μπορεί να είναι HP ή LaserJet, παρόλο που οι τρεις εμπλεκόμενες κλάσεις είναι υποκλάσεις της κλάσης όλων των εκτυπωτών



# ΟΡΙΣΜΟΣ ΤΗΣ ΓΛΩΣΣΑΣ OWL ΜΕ ΧΡΗΣΗ ΤΗΣ OWL

Εδώ παρουσιάζουμε ένα μέρος του ορισμού της OWL, χρησιμοποιώντας την ίδια

# Χώροι Ονομάτων

```
<?xml version="1.0"?>
<!DOCTYPE owl [
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY owl "http://www.w3.org/2002/07/owl#"> ]>
<rdf:RDF
xml:base ="http://www.w3.org/2002/07/owl"
xmlns ="&owl;" xmlns:owl ="&owl;" xmlns:rdf ="&rdf;" xmlns:rdfs="&rdfs;"
xmlns:dc ="http://purl.org/dc/elements/1.1/">
```

- Η διεύθυνση URI του τρέχοντος εγγράφου (του ορισμού της OWL) ορίζεται ως ο προεπιλεγμένος χώρος ονομάτων
  - Επομένως, το πρόθεμα *owl:* δε θα χρησιμοποιηθεί παρακάτω
- Επίσης, η χρήση των ορισμών των οντοτήτων XML, μας επιτρέπουν να γράφουμε συντμήσεις των διευθύνσεων URL που εμφανίζονται σε τιμές χαρακτηριστικών

# Κλάσεις κλάσεων (μετακλάσεις)

- Η κλάση όλων των κλάσεων της OWL είναι και η ίδια υποκλάση όλων των κλάσεων της γλώσσας RDF Schema:

```
<rdfs:Class rdf:ID="Class">  
<rdfs:label>Class</rdfs:label>  
<rdfs:comment>The class of all OWL classes</rdfs:comment>  
<rdfs:subClassOf rdf:resource="&rdfs;Class"/>  
</rdfs:Class>
```

- Η κλάση *Thing* είναι η γενικότερη κλάση αντικειμένων στην OWL, και η *Nothing* είναι η ειδικότερη (δηλ. η κενή)

- Ισχύουν οι σχέσεις:

```
<Class rdf:ID="Thing">  
<rdfs:label>Thing</rdfs:label>  
<unionOf rdf:parseType="Collection">  
<Class rdf:about="#Nothing"/>  
<Class> <complementOf rdf:resource="#Nothing"/> </Class>  
</unionOf>  
</Class>
```

```
<Class rdf:ID="Nothing"> <rdfs:label>Nothing</rdfs:label>  
<complementOf rdf:resource="#Thing"/>  
</Class>
```

$$\begin{aligned} \textit{Thing} &= \textit{Nothing} \cup \textit{Nothing}^c \\ \textit{Nothing} &= \textit{Thing}^c = \textit{Nothing}^c \cap \textit{Nothing}^{cc} = \emptyset \end{aligned}$$



# Ισοδυναμία κλάσεων (1/3)

- Η ισοδυναμία κλάσεων, που εκφράζεται με το στοιχείο *owl:EquivalentClass*, υποδηλώνει μία σχέση υποκλάσεων και ορίζεται πάντα μεταξύ δύο κλάσεων
  - Αυτό είναι ανάλογο με το *owl:EquivalentProperty*
- Οι προτάσεις μη επικάλυψης μπορούν να διατυπωθούν μόνο μεταξύ κλάσεων

```
<rdf:Property rdf:ID="EquivalentClass">
<rdfs:label>EquivalentClass</rdfs:label>
<rdfs:subPropertyOf rdf:resource="&rdfs;subClassOf"/>
<rdfs:domain rdf:resource="#Class"/>
<rdfs:range rdf:resource="#Class"/>
</rdf:Property>

<rdf:Property rdf:ID="EquivalentProperty">
<rdfs:label>EquivalentProperty</rdfs:label>
<rdfs:subPropertyOf rdf:resource="&rdfs;subPropertyOf"/>
</rdf:Property>

<rdf:Property rdf:ID="disjointWith">
<rdfs:label>disjointWith</rdfs:label>
<rdfs:domain rdf:resource="#Class"/>
<rdfs:range rdf:resource="#Class"/>
</rdf:Property>
```

# Ισοδυναμία κλάσεων (2/3)

- Η ισότητα και η ανισότητα μπορούν να δηλωθούν μεταξύ αυθαίρετων αντικειμένων
  - Στην OWL Full αυτό μπορεί να ισχύει και για κλάσεις
  - Το στοιχείο `owl:sameAs` είναι απλώς συνώνυμο του `owl:sameIndividualAs`

```
<rdf:Property rdf:ID="sameIndividualAs">  
<rdfs:label>sameIndividualAs</rdfs:label>  
<rdfs:domain rdf:resource="#Thing"/>  
<rdfs:range rdf:resource="#Thing"/>  
</rdf:Property>
```

```
<rdf:Property rdf:ID="differentFrom">  
<rdfs:label>differentFrom</rdfs:label>  
<rdfs:domain rdf:resource="#Thing"/>  
<rdfs:range rdf:resource="#Thing"/>  
</rdf:Property>
```

```
<rdf:Property rdf:ID="sameAs">  
<rdfs:label>sameAs</rdfs:label>  
<EquivalentProperty rdf:resource="#sameIndividualAs"/>  
</rdf:Property>
```

# Ισοδυναμία κλάσεων (3/3)

- Το *owl:distinctMembers* μπορεί να χρησιμοποιηθεί μόνο για το *owl:AllDifferent*:

```
<rdfs:Class rdf:ID="AllDifferent">  
<rdfs:label>AllDifferent</rdfs:label>  
</rdfs:Class>  
  
<rdf:Property rdf:ID="distinctMembers">  
<rdfs:label>distinctMembers</rdfs:label>  
<rdfs:domain rdf:resource="#AllDifferent"/>  
<rdfs:range rdf:resource="&rdf;List"/>  
</rdf:Property>
```

# Δημιουργία κλάσεων από άλλες κλάσεις

- Το στοιχείο *owl:unionOf* δημιουργεί μία κλάση από μία λίστα
  - Που υποτίθεται ότι είναι λίστα με παραστάσεις άλλων κλάσεων

```
<rdf:Property rdf:ID="unionOf">  
<rdfs:label>unionOf</rdfs:label>  
<rdfs:domain  
  rdf:resource="#Class"/>  
<rdfs:range rdf:resource="#&rdf;List"/>  
</rdf:Property>
```

- Τα στοιχεία *owl:intersectionOf* και *owl:oneOf* δημιουργούν επίσης κλάσεις από μία λίστα, παρόλο που για τις δομές αυτές η λίστα υποτίθεται ότι είναι λίστα μεμονωμένων στοιχείων
- Το *owl:complementOf* ορίζει μία κλάση σε σχέση με μία μόνο άλλη κλάση

```
<rdf:Property rdf:ID="complementOf">  
<rdfs:label>complementOf</rdfs:label>  
<rdfs:domain rdf:resource="#Class"/>  
<rdfs:range rdf:resource="#Class"/>  
</rdf:Property>
```

# Περιορισμοί ιδιοτήτων των κλάσεων (1/2)

- Οι περιορισμοί στην OWL ορίζουν την κλάση εκείνων των αντικειμένων που ικανοποιούν ορισμένες προσαρτημένες συνθήκες

```
<rdfs:Class rdf:ID="Restriction">  
<rdfs:label>Restriction</rdfs:label>  
<rdfs:subClassOf rdf:resource="#Class"/>  
</rdfs:Class>
```

- Όλες οι παρακάτω ιδιότητες επιτρέπεται να εμφανίζονται μόνο μέσα σε έναν ορισμό περιορισμού
  - Δηλαδή το πεδίο ορισμού τους είναι το στοιχείο owl:Restriction, αλλά διαφέρουν όσον αφορά το σύνολο τιμών τους:

```
<rdf:Property rdf:ID="onProperty">  
<rdfs:label>onProperty</rdfs:label>  
<rdfs:domain rdf:resource="#Restriction"/>  
<rdfs:range rdf:resource="&rdf;Property"/>  
</rdf:Property>
```

# Περιορισμοί ιδιοτήτων των κλάσεων (2/2)

- ...
- Τα στοιχεία *owl:maxCardinality* και *owl:cardinality* ορίζονται ανάλογα με το *owl:minCardinality*, και το *owl:someValuesFrom* ορίζεται ανάλογο με το *owl:allValuesFrom*
- Το *owl:onProperty* επιτρέπει τον περιορισμό μιας ιδιότητας αντικειμένου ή τύπου δεδομένων. Επομένως, το σύνολο τιμών των ιδιοτήτων περιορισμού, όπως το *owl:allValuesFrom* δεν είναι η κλάση *owl:Class* αλλά η πιο γενική κλάση *rdfs:Class*

```
<rdf:Property rdf:ID="allValuesFrom">  
<rdfs:label>allValuesFrom</rdfs:label>  
<rdfs:domain rdf:resource="#Restriction"/>  
<rdfs:range rdf:resource="#&rdfs;Class"/>  
</rdf:Property>
```

```
<rdf:Property rdf:ID="hasValue">  
<rdfs:label>hasValue</rdfs:label>  
<rdfs:domain rdf:resource="#Restriction"/>  
</rdf:Property>
```

```
<rdf:Property rdf:ID="minCardinality">  
<rdfs:label>minCardinality</rdfs:label>  
<rdfs:domain rdf:resource="#Restriction"/>  
<rdfs:range  
rdf:resource="#&xsd;nonNegativeInteger"/>  
</rdf:Property>
```

# Ιδιότητες

- Το στοιχείο *owl:ObjectProperty* είναι ειδική περίπτωση του *rdf:Property*
- Ομοίως και το *owl:DatatypeProperty*

```
<rdfs:Class rdf:ID="ObjectProperty">  
<rdfs:label>ObjectProperty</rdfs:label> <rdfs:subClassOf rdf:resource="&rdf;Property"/>  
</rdfs:Class>
```

- Το *owl:TransitiveProperty* μπορεί να εφαρμοστεί μόνο σε ιδιότητες αντικειμένου
- Ομοίως για τις συμμετρικές, συναρτησιακές και αντίστροφες συναρτησιακές ιδιότητες
- Το *owl:inverseOf* συσχετίζει δύο ιδιότητες ενός αντικειμένου:

```
<rdfs:Class rdf:ID="TransitiveProperty">  
<rdfs:label>TransitiveProperty</rdfs:label>  
<rdfs:subClassOf rdf:resource="#ObjectProperty"/>  
</rdfs:Class>
```

```
<rdf:Property rdf:ID="inverseOf">  
<rdfs:label>inverseOf</rdfs:label>  
<rdfs:domain rdf:resource="#ObjectProperty"/>  
<rdfs:range rdf:resource="#ObjectProperty"/>  
</rdf:Property>
```



# ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ



# Υπομονάδες και εισαγωγές

- Η εισαγωγή οντολογιών που ορίζονται από άλλους θα αποτελέσει τον κανόνα στο ΣΙ
- Όμως, η λειτουργία εισαγωγής της OWL είναι πολύ στοιχειώδης
  - Επιτρέποντας μόνο την εισαγωγή μιας *ολόκληρης οντολογίας*, η οποία καθορίζεται από τη θέση της
- Ακόμα κι αν κάποιος επιθυμούσε να χρησιμοποιήσει μόνο ένα μικρό τμήμα μιας άλλης οντολογίας, θα ήταν αναγκασμένος να εισάγει ολόκληρη την οντολογία
- Η κατασκευή υπομονάδων στις γλώσσες προγραμματισμού βασίζεται στην έννοια της απόκρυψης πληροφοριών
- Η υπομονάδα παρέχει μία λειτουργικότητα προς τον εξωτερικό κόσμο, αλλά η υπομονάδα εισαγωγής δεν χρειάζεται να ασχοληθεί με τον τρόπο που επιτεύχθηκε αυτή η λειτουργικότητα
- Η αντίστοιχη έννοια της απόκρυψης πληροφοριών για τις οντολογίες, καθώς επίσης και η χρήση της ως βάση για μια καλή δομή εισαγωγής, παραμένει ανοικτό ερευνητικό ζήτημα

# Προεπιλεγμένες τιμές και υπόθεση κλειστού κόσμου

- Πολλά πρακτικά συστήματα αναπαράστασης γνώσης επιτρέπουν στις πιο συγκεκριμένες κλάσεις της ιεραρχίας να παρακάμπτουν τις κληρονομημένες τιμές
  - αντιμετωπίζοντας τις τελευταίες ως προεπιλεγμένες τιμές
- Παρόλο που η πρακτική αυτή ακολουθείται ευρέως
  - δεν έχει επιτευχθεί ακόμα συνέναιση σχετικά με τη σωστή τυποποίηση
- **Υπόθεση κλειστού κόσμου**
- Η σημασιολογία της OWL υιοθετεί προς το παρόν το κλασσικό λογικό μοντέλο της υπόθεσης του ανοιχτού κόσμου
  - Μια πρόταση δεν μπορεί να θεωρηθεί αληθής λόγω αποτυχίας της απόδειξής της
- Όμως, η αντίθετη προσέγγιση (υπόθεση του κλειστού κόσμου: μία πρόταση είναι αληθής, όταν δεν μπορεί να αποδειχτεί η άρνησή της) είναι επίσης χρήσιμη σε ορισμένες εφαρμογές
- Η υπόθεση του κλειστού κόσμου είναι στενά συνδεδεμένη με την έννοια των προεπιλεγμένων τιμών και οδηγεί στην ίδια μη μονοτονική συμπεριφορά
  - λόγος για να μη συμπεριληφθεί στην OWL

# Υπόθεση μοναδικού ονόματος

- Οι συνηθισμένες εφαρμογές βάσεων δεδομένων υποθέτουν ότι τα μεμονωμένα στοιχεία με διαφορετικά ονόματα είναι όντως διαφορετικά μεμονωμένα στοιχεία
- Η OWL ακολουθεί το σύνηθες λογικό πρότυπο στο οποίο αυτό δεν ισχύει
  - Αν δύο μεμονωμένα στοιχεία (ή κλάσεις ή ιδιότητες) έχουν διαφορετικά ονόματα, εξακολουθούμε να είμαστε σε θέση να εξάγουμε μέσω συμπερασμού ότι αντιστοιχούν στο ίδιο μεμονωμένο στοιχείο
- Παρόλα αυτά υπάρχουν καταστάσεις όπου η υπόθεση του μοναδικού ονόματος είναι χρήσιμη
- Επίσης, κάποιος μπορεί να επιθυμεί να δηλώσει τα μέρη της οντολογίας για τα οποία η υπόθεση ισχύει ή όχι

# Διαδικαστική Προσάρτηση

- Μία κοινή έννοια στην αναπαράσταση γνώσης είναι ο ορισμός του νοήματος ενός όρου όχι μέσω ρητών ορισμών μέσα στη γλώσσα
  - Αλλά με την προσάρτηση ενός κομματιού κώδικα, που θα εκτελεστεί για να υπολογίσει το νόημα του όρου
- Παρόλο που η έννοια αυτή χρησιμοποιείται ευρέως, δεν είναι ιδιαίτερα κατάλληλη για ενσωμάτωση σε ένα σύστημα με τυπική σημασιολογία και δεν έχει συμπεριληφθεί στην OWL

# Κανόνες Αλυσιδωτής Σύνδεσης Ιδιοτήτων

- Η OWL δεν επιτρέπει τη σύνθεση ιδιοτήτων για λόγους λήψης αποφάσεων
  - Αλλά αυτή είναι μια χρήσιμη λειτουργία σε πολλές εφαρμογές
- Γενικότερα, θα θέλαμε να ορίζουμε ιδιότητες ως γενικούς κανόνες πάνω από άλλες ιδιότητες
  - Αυτή η ενοποίηση της αναπαράστασης γνώσης που βασίζεται σε κανόνες και της αναπαράστασης γνώσης με στυλ περιγραφικής λογικής παραμένει ενεργό πεδίο έρευνας