

# Τεχνητή Νοημοσύνη

Σημαντικά Δίκτυα – Πλαίσια - Οντολογίες

Δρ. Δημήτριος Κουτσομητρόπουλος

1

Σημαντικά Δίκτυα

2

2

## Σημαντικά Δίκτυα

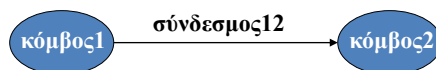
- ▶ Βασίζονται στην αίσθηση ότι ένα χαρακτηριστικό της ανθρώπινης μνήμης είναι η ύπαρξη μεγάλου αριθμού συσχετίσεων μεταξύ των τμημάτων πληροφορίας που βρίσκεται αποθηκευμένη σ' αυτή.
- ▶ Ονομάζονται **σημαντικά ή σημασιολογικά ή συσχετιστικά δίκτυα** (semantic or associative networks). Ο όρος 'σημαντικά' προήλθε από το γεγονός ότι χρησιμοποιήθηκαν πρωταρχικά (από τον Quillian, 1968) για αναπαράσταση της σημασίας (σημαντικής ή σημασιολογίας) λέξεων της Αγγλικής γλώσσας.
- ▶ Υπάρχουν πολλοί τύποι σημαντικών δικτύων. Κοινό γνώρισμα, η χρήση των στοιχείων 'κόμβος' (node) και 'σύνδεσμος' (link) για την αναπαράσταση γνώσης.

▶ 3

3

## Βασικές Έννοιες(1)

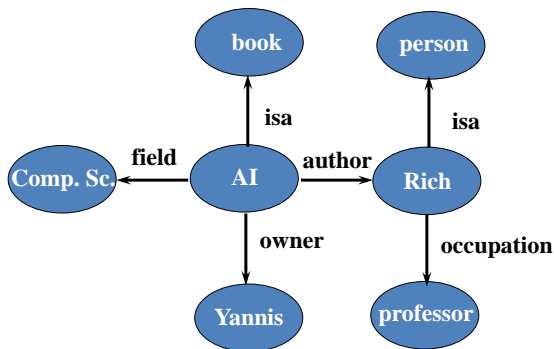
- ▶ Βασικά στοιχεία ενός σημαντικού δικτύου είναι οι έννοιες 'κόμβος' και 'σύνδεσμος'.
- ▶ Οι κόμβοι παριστάνουν **οντότητες** (φυσικά αντικείμενα, έννοιες, γεγονότα, καταστάσεις κλπ). Οι σύνδεσμοι συνδέουν τους κόμβους μεταξύ τους και παριστάνουν **δυναμικές σχέσεις** μεταξύ τους.
- ▶ Η αναπαράσταση μέσω ενός σημαντικού δικτύου αποτυπώνεται καλύτερα γραφικά. Η βασική δομική μονάδα ενός τέτοιου γραφήματος είναι:



▶ 4

4

## Βασικές Έννοιες(2)



Ένα απλό Σημαντικό Δίκτυο

▶ 5

5

## Βασικές Έννοιες(3)

- ▶ Κάθε βασική μονάδα ενός σημαντικού δικτύου μπορεί να θεωρηθεί ότι παριστάνει ένα κατηγορημα 2-θέσεων. Π.χ. στο προηγούμενο παράδειγμα μπορούμε να διακρίνουμε, μεταξύ άλλων, τις παρακάτω ατομικές προτάσεις για τα κατηγορήματα *isa*, *author*, *owner*:

*isa* (AI, book), *author* (Rich, AI), *owner* (Yannis, AI)

- ▶ Οι κόμβοι μπορεί να παριστάνουν είτε **γενικές** (generic) οντότητες είτε **ατομικές** (individual). Π.χ. στο προηγούμενο παράδειγμα οι κόμβοι 'book' και 'person' παριστάνουν γενικές οντότητες, δηλ. κλάσεις αντικειμένων, ενώ οι 'Yannis' και 'AI' ατομικές οντότητες, δηλ. συγκεκριμένα αντικείμενα.

▶ 6

6

## Βασικές Έννοιες(4)

- ▶ Η βάση γνώσης (ΒΓ) ενός σημαντικού δικτύου είναι μια συλλογή αντικειμένων (κάθε είδους) και σχέσεων μεταξύ τους ώστε να αντιπροσωπεύουν το αντίστοιχο δίκτυο. Οι τροποποιήσεις στη ΒΓ, δηλ. εισαγωγή ή διαγραφή αντικειμένων και σχέσεων, απαιτούν τη σχεδίαση ενός ισχυρού μηχανισμού διαχείρισης.

▶ 7

7

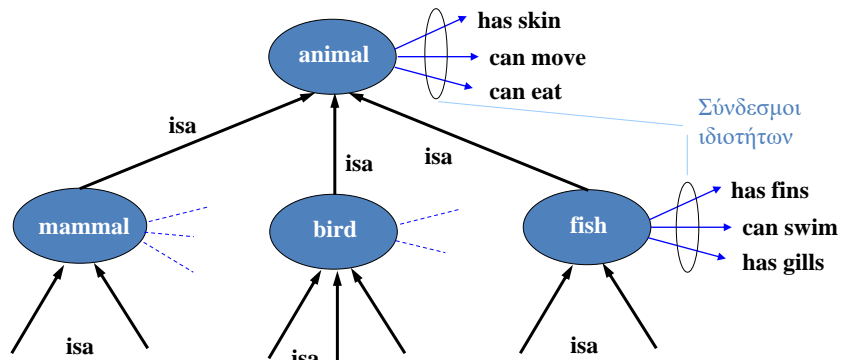
## Ιεραρχικά Δίκτυα (1)

- ▶ Οι κόμβοι είναι οργανωμένοι σε μια ιεραρχία.
- ▶ Οι κόμβοι ενός επιπέδου είναι υπο-περιπτώσεις (υπο-κατηγορίες) των κόμβων του παραπάνω επιπέδου και έχουν σαν υπο-περιπτώσεις (υπο-κατηγορίες) τους κόμβους του παρακάτω επιπέδου.
- ▶ Οι κόμβοι ενός επιπέδου συνδέονται με αυτούς του παραπάνω ή παρακάτω επιπέδου με **συνδέσμους τύπου isa**. Γι' αυτό καλούνται και **ιεραρχίες isa**.
- ▶ Υπάρχουν σύνδεσμοι που συσχετίζουν ένα σύνολο χαρακτηριστικών ή ιδιοτήτων με κάθε κόμβο και λέγονται **σύνδεσμοι ιδιοτήτων** (property links). Οι ιδιότητες αυτές χαρακτηρίζουν (ή ορίζουν) το αντικείμενο. Ένας συνηθής σύνδεσμος ιδιοτήτων είναι ο has\_part ή has\_property ή απλώς has.

▶ 8

8

## Ιεραρχικά Δίκτυα (2)



Παράδειγμα ιεραρχικού σημαντικού δικτύου

▶ 9

9

## Κληρονόμηση Ιδιοτήτων

- ▶ Σ' ένα ιεραρχικό δίκτυο, ιδιότητες που ισχύουν για ένα αντικείμενο-κόμβο στην ιεραρχία, ισχύουν επίσης και για αντικείμενα-κόμβους που βρίσκονται χαμηλότερα στην ιεραρχία και συνδέονται μαζί του με συνδέσμους isa, χωρίς να είναι αναγκαίο να αναπαρασταθούν φανερά και σ' αυτούς.
- ▶ Αυτό εισάγει ένα μηχανισμό εξαγωγής συμπερασμάτων που ονομάζεται **κληρονόμηση ιδιοτήτων** (property inheritance). Σύμφωνα μ' αυτόν, για να εξάγουμε συμπεράσματα, αρκεί να προχωράμε προς τα πάνω στην ιεραρχία, μέχρι να συναντήσουμε τον ζητούμενο ισχυρισμό. Π.χ. από το προηγούμενο δίκτυο, μπορούμε να συνάγουμε τα συμπεράσματα ότι ένα ψάρι (fish), εκτός από το ότι has fins, can swim και has gills, επίσης has skin, can move και can eat, λόγω κληρονομικότητας από το animal.

▶ 10

10

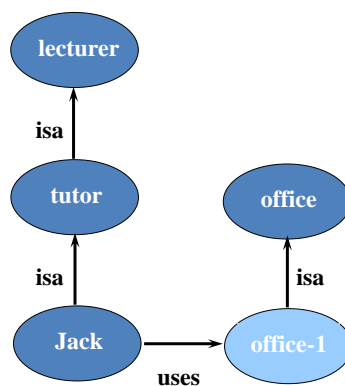
## Ιεραρχικά Δίκτυα με Στιγμιότυπα(1)

- ▶ Μια επέκταση των ιεραρχικών δικτύων ήταν η εισαγωγή της έννοιας της **στιγμιοτυποποίησης** (instantiation) σ' αυτά, δηλ. η χρήση **στιγμιοτύπων** (instances) για την αναπαράσταση συγκεκριμένων αντιπροσώπων μιας κλάσης.
- ▶ Π.χ. αν θέλουμε να αναπαραστήσουμε το γεγονός "Jack uses an office", δεδομένου ότι είναι λέκτορας (lecturer) και δάσκαλος (tutor), θα χρησιμοποιήσουμε το επόμενο δίκτυο, όπου 'office-1' είναι ένα στιγμιότυπο της πιο γενικής έννοιας 'office'.

▶ 11

11

## Ιεραρχικά Δίκτυα με Στιγμιότυπα(2)



▶ 1212

12

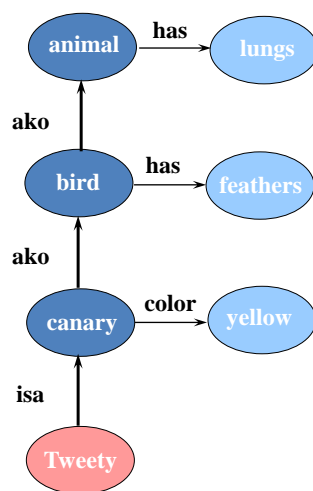
## Ιεραρχικά Δίκτυα με Στιγμιότυπα(3)

- ▶ Γενικότερα, σ' ένα σύγχρονο ιεραρχικό δίκτυο ξεχωρίζουμε δύο τύπους κόμβων: **κόμβος-κλάση** (που παριστάνει μια γενική έννοια, μια κλάση αντικειμένων) και **κόμβος-στιγμιότυπο** (που παριστάνει μια συγκεκριμένη οντότητα, μια ατομική έννοια).
- ▶ Συνήθως χρησιμοποιείται ξεχωριστή σχέση για τη σύνδεση ενός κόμβου-στιγμιότυπο και ενός κόμβου-κλάση απ' αυτή μεταξύ δύο κόμβων-κλάση.
- ▶ Τέτοιες δυάδες συμβολισμού των σχέσεων αυτών είναι: (sub, inst), (ako, isa), (isa, inst)

▶ 13

13

## Ιεραρχικά Δίκτυα με Στιγμιότυπα(4)



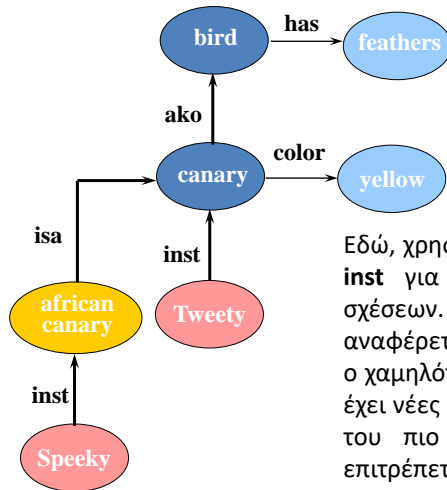
Εδώ, χρησιμοποιείται το ζεύγος **ako** (από το a kind of) και **isa** για αναπαράσταση ιεραρχικών σχέσεων. Θα μπορούσαμε αντ' αυτών να χρησιμοποιήσουμε τα **sub** και **inst** αντίστοιχα.

Η **κληρονόμηση ιδιοτήτων** ισχύει και για τους κόμβους στιγμιότυπα. Κληρονομούν ιδιότητες από την κλάση τους μέσω του **isa** και από τις κλάσεις πάνω απ' αυτήν μέσω των συνδέσμων **ako**. Π.χ. αν ο προς απόδειξη ισχυρισμός είναι 'Tweety has feathers', τότε επειδή η ιδιότητα αυτή δεν υπάρχει στον κόμβο 'Tweety', η διαδικασία προχωρά στον κόμβο 'canary' μέσω του **isa**. Επειδή και κει δεν υπάρχει, προχωρά στον κόμβο 'bird', όπου υπάρχει η ιδιότητα και επομένως επιστρέφει ότι ο ισχυρισμός είναι αληθής.

▶ 14

14

## Ιεραρχικά Δίκτυα με Στιγμιότυπα (5)



Εδώ, χρησιμοποιείται η τριάδα **ako**, **isa** και **inst** για τη αναπαράσταση ιεραρχικών σχέσεων. Στην περίπτωση αυτή η σχέση **isa** αναφέρεται σε δύο κόμβους-κλάσεις, όπου ο χαμηλότερα ευρισκόμενος δεν μπορεί να έχει νέες ιδιότητες, μόνο κληρονομεί αυτές του πιο πάνω κόμβου. Το μόνο που επιτρέπεται είναι η αλλαγή τιμών στις ιδιότητες.

▶ 15

15

## Ιεραρχικά Δίκτυα με Στιγμιότυπα (6)

### ▶ Σχέση **ako**

- ✓ Χρησιμοποιείται για τη δήλωση ιεραρχικής σχέσης μεταξύ δύο κόμβων-κλάσεων.
- ✓ Μπορούν να προστεθούν νέοι σύνδεσμοι ιδιοτήτων στον κόμβο-υποκλάση.

### ▶ Σχέση **isa**

- ✓ Χρησιμοποιείται για τη δήλωση ιεραρχικής σχέσης μεταξύ δύο κόμβων-κλάσεων.
- ✓ **Δεν** μπορούν να προστεθούν νέοι σύνδεσμοι ιδιοτήτων στον κόμβο-υποκλάση.

### ▶ Σχέση **inst**

- ✓ Χρησιμοποιείται για τη δήλωση ιεραρχικής σχέσης μεταξύ ενός κόμβου-στιγμιότυπου και ενός κόμβου-κλάσης.
- ✓ **Δεν** μπορούν να προστεθούν νέοι σύνδεσμοι ιδιοτήτων στον κόμβο-υποκλάση.

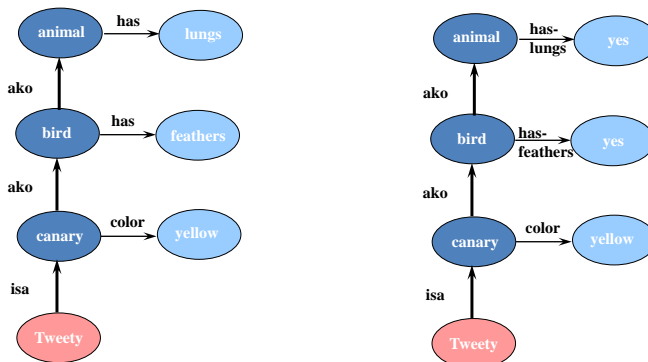
▶ 16

16



## Ιεραρχικά Δίκτυα με Στιγμιότυπα (7)

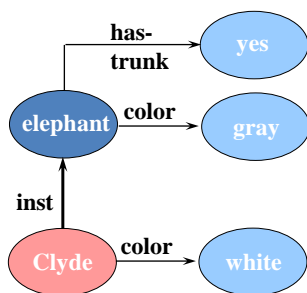
- Για καλύτερη αναπαράσταση και διευκόλυνση της κληρονομικότητας ως μηχανισμού εξαγωγής συμπερασμάτων/απαντήσεων από ένα σύγχρονο ιεραρχικό δίκτυο με στιγμιότυπα, αποφεύγουμε να χρησιμοποιούμε ως ιδιότητες (συνδέσμους ιδιοτήτων) βοηθητικά ρήματα, όπως «έχει» (has), «μπορεί» (can) κλπ. Π.χ. προτιμούμε το κάτω δεξιά από το κάτω αριστερά δίκτυο.



► 17

17

## Εξ ορισμού (default) Συλλογισμός



Εδώ, φαίνεται πόσο φυσικά ένα σημαντικό δίκτυο μπορεί να χειριστεί εξαιρέσεις. Ο Clyde είναι λευκός ελέφαντας, ενώ συνήθως οι ελέφαντες είναι γκριζοί. Όταν ζητήσουμε το χρώμα του Clyde, επειδή συναντάμε πρώτα τον δικό του σύνδεσμο ιδιότητας για το χρώμα, απαντάμε λευκό (white), χωρίς να χρειαστεί να ανέβουμε στον 'elephant'. Δηλ. ένας σύνδεσμος ιδιότητας **επισκιάζει** (ή **καλύπτει**) ένα ίδιο σύνδεσμο που βρίσκεται πιο ψηλά στην ιεραρχία.

► 18

18

## Προβλήματα (1)

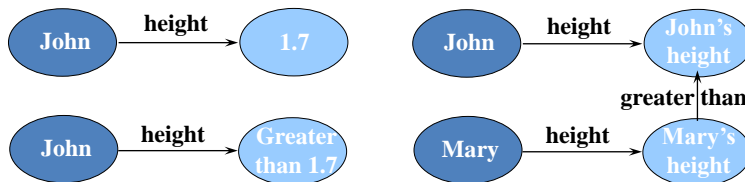
Είναι κυρίως προβλήματα ερμηνείας κόμβων και συνδέσμων.

Ας υποθέσουμε ότι θέλουμε να αναπαραστήσουμε τα παρακάτω γεγονότα:

“John’s height is 1.7”, “John’s height is greater than 1.7m”

“John’s height is greater than Mary’s”.

Τότε κατασκευάζουμε τα παρακάτω σημαντικά δίκτυα.



Προσέξτε ότι οι κόμβοι προορισμού του συνδέσμου 'height' έχουν διαφορετικές έννοιες, παριστάνουν εντελώς διαφορετικές οντότητες.

▶ 19

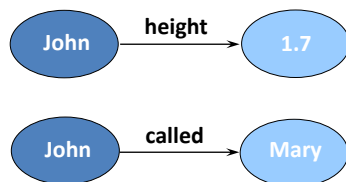
19

## Προβλήματα (2)

Επίσης, αν κάποιος δει ένα κόμβο-κλάση με το όνομα 'τηλέφωνο', δεν είναι καθαρό το τι ακριβώς σημαίνει: την έννοια τηλέφωνο, την κλάση (δηλ. το σύνολο) των τηλεφώνων ή ένα τυπικό τηλέφωνο. Παρομοίως και ένας κόμβος-στιγμιότυπο μπορεί να ερμηνευθεί πολλαπλά: ένα συγκεκριμένο τηλέφωνο ή ένα τυχαίο τηλέφωνο.

Ας υποθέσουμε τώρα ότι θέλουμε να αναπαραστήσουμε τα γεγονότα:

“John’s height is 1.7”, “John called Mary”. Οι αναπαραστάσεις είναι:



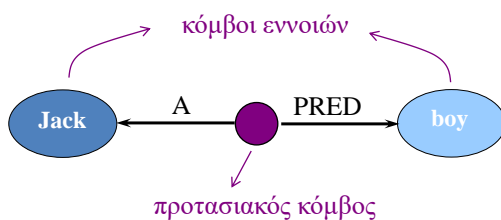
Παρατηρούμε ότι στη μια περίπτωση ο σύνδεσμος αντιπροσωπεύει μια ιδιότητα, ενώ στην άλλη μια ενέργεια, δηλ. δύο διαφορετικά πράγματα.

▶ 20

20

## Προτασιακά Δίκτυα (1)

- ▶ Ένας άλλος τύπος σημαντικών δικτύων είναι τα λεγόμενα **προτασιακά δίκτυα** (proposition nets). Χρησιμοποιούν την έννοια του **προτασιακού κόμβου** (proposition node) και προσπαθούν να μιμηθούν την **κατηγορηματική λογική** (n-ary predicates).
- ▶ Π.χ. το απλό γεγονός (κατηγορηματικό 1ου βαθμού ή μιας θέσης) "Jack is a boy", παριστάνεται ως εξής:

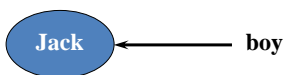


▶ 21

21

## Προτασιακά Δίκτυα(2)

Το οποίο απλοποιείται ως εξής:



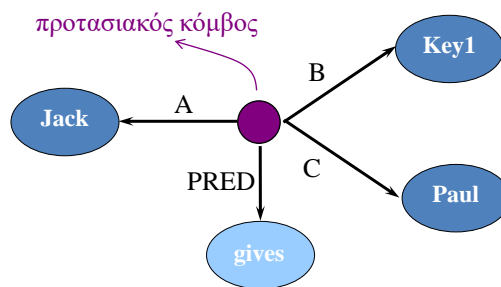
όπου ο προτασιακός κόμβος αντιπροσωπεύεται από το **όνομα** (boy) του κατηγορήματος (ώστε να θυμίζει το boy(Jack)). Το όνομα ενεργεί σαν φύλακας-θέσης για τον προτασιακό κόμβο. Ο σύνδεσμος προς τον κόμβο του κατηγορήματος είναι κρυμμένος/υπονοείται.

▶ 22

22

### Προτασιακά Δίκτυα(3)

Ένα πιο σύνθετο γεγονός, π.χ. “Jack gives the key to Paul”, που περιέχει ένα κατηγορήμα τρίτου βαθμού (ή τριών θέσεων) παριστάνεται με το επόμενο δίκτυο.

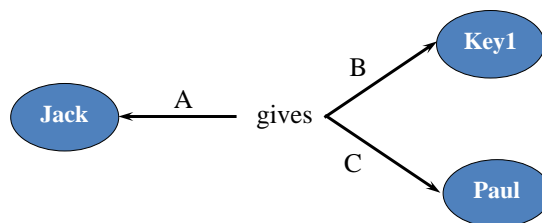


▶ 23

23

### Προτασιακά Δίκτυα(4)

Η απλοποιημένη του παράσταση είναι:



Τα A, B, C δείχνουν τη σειρά των ορισμάτων του κατηγορήματος

▶ 24

24

## Διαδικαστικά Δίκτυα(1)

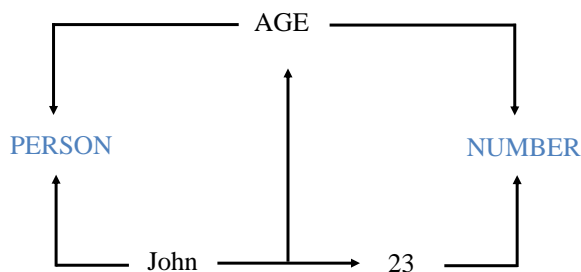
- ▶ Τα **διαδικαστικά σημαντικά δίκτυα** (procedural semantic nets-PSNs) χρησιμοποιούν τους όρους **κλάση** (class) και **σχέση** (relation) για την αναπαράσταση του τύπου μιας έννοιας (κόμβος) και του τύπου μιας σχέσης (σύνδεσμος) στο δίκτυο.
- ▶ Μια κλάση είναι μια συλλογή από αντικείμενα με κοινές ιδιότητες. Τα συγκεκριμένα αντικείμενα μιας κλάσης ονομάζονται **στιγμιότυπα** (instances) της κλάσης και μπορεί να είναι κι' αυτά κλάσεις.
- ▶ Μια σχέση είναι μια απεικόνιση από μια κλάση, που ονομάζεται **πεδίο ορισμού** (domain) σε μια άλλη κλάση, που ονομάζεται **πεδίο τιμών** (range). Οι σχέσεις μπορεί να θεωρηθούν σαν κλάσεις με στιγμιότυπα ισχυρισμούς (assertions).

▶ 25

25

## Διαδικαστικά Δίκτυα(2)

Π.χ. το γεγονός "John is a person whose age is 23",  
παριστάνεται ως εξής:



**PERSON, NUMBER:** κλάσεις  
**AGE:** σχέση (πεδίο ορισμού:  
PERSON, πεδίο τιμών: NUMBER)

**Σύνδεσμοι:** δείχνουν τον τύπο  
(‘John’ και ‘23’ στιγμιότυπα των  
PERSON και NUMBER)

▶ 26

26

## Διαδικαστικά Δίκτυα(3)

- ▶ Η κύρια ιδέα είναι να συσχετίσουμε **προγράμματα** με κάθε τύπο κλάσης και σχέσης ανεξάρτητα μεταξύ τους. Ένα πρόγραμμα θεωρείται σαν μια κλάση της οποίας τα στιγμιότυπα καλούνται διαδικασίες και αντιστοιχούν σε ενεργοποιήσεις προγραμμάτων
- ▶ Υπάρχουν τέσσερις βασικές λειτουργίες σε κάθε κλάση και σχέση: **πρόσθεσε** (add) ένα στιγμιότυπο σε μια κλάση, **διάγραψε** (remove) ένα στιγμιότυπο από μια κλάση, **ανάκτησε** (fetch) όλα τα στιγμιότυπα μιας κλάσης, **έλεγε** (test) αν ένα στιγμιότυπο ανήκει σε μια κλάση. Κάθε λειτουργία αποτελεί ένα πρόγραμμα προσαρτημένο σε κάθε κλάση ή σχέση.
- ▶ Π.χ. για να εκφράσουμε το γεγονός ότι η έννοια της σχέσης PROFIT (κέρδος) είναι η διαφορά της τιμής μείον το κόστος, δημιουργούμε ένα πρόγραμμα για τη λειτουργία fetch που φέρνει την τιμή και το κόστος ενός προϊόντος και υπολογίζει τη διαφορά.

▶ 27

27

## Πλεονεκτήματα - Μειονεκτήματα

### ΠΛΕΟΝΕΚΤΗΜΑΤΑ

- ▶ Η αναπαράσταση με σημαντικά δίκτυα ανταποκρίνεται στη δομή του πραγματικού κόσμου, οπότε δημιουργεί αναπαραστάσεις με μεγάλη φυσικότητα.
- ▶ Οδηγεί σε συμπαγείς αναπαραστάσεις, δηλ. αναπαραστάσεις όπου όλες οι εμπλεκόμενες οντότητες σχετίζονται σε μια ενιαία περιγραφή.
- ▶ Εύκολη αναπαράσταση και χειρισμός εξαιρέσεων (μη μονότονη αναπαράσταση).

### ΜΕΙΟΝΕΚΤΗΜΑΤΑ

- ▶ Προβλήματα ερμηνείας.
- ▶ Πολύπλοκη διαδικασία αναζήτησης και κληρονόμησης πληροφορίας.
- ▶ Πολύπλοκος μηχανισμός διαχείρισης της βάσης γνώσης (έλλειψη τμηματικότητας)

▶ 28

28



Πλαίσια

29

29

## Αναπαράσταση με Πλαίσια

---

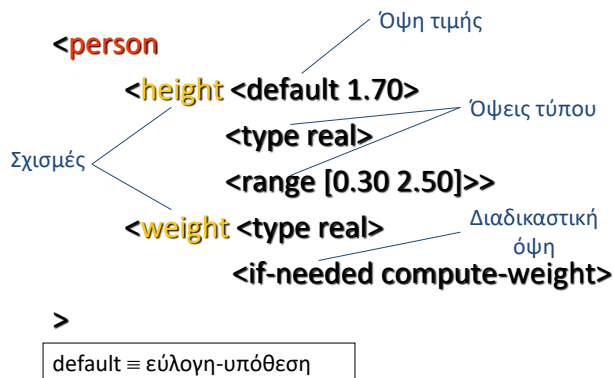
- ▶ Τα πλαίσια έχουν:
  - ▶ όνομα
  - ▶ μία σειρά από σχισμές (slots) που περιγράφουν ιδιότητες
  - ▶ όψεις (facets) → τιμές (fillers)
  - ▶ δηλωτικές
    - ▶ τύπου (type, range)
    - ▶ τιμής (value, default)
  - ▶ διαδικαστικές → διαδικαστική προσάρτηση
    - ▶ (προαιρετικά) προσαρτημένες διαδικασίες (π.χ if-needed, if-added, if-removed)
    - ▶ μπορεί να ενεργοποιούνται όταν τα πλαίσια μεταβάλλονται για κάποιο λόγο

---

▶ 30

30

## Πλαίσια - Παράδειγμα



31

## Ιεραρχίες Πλαισίων

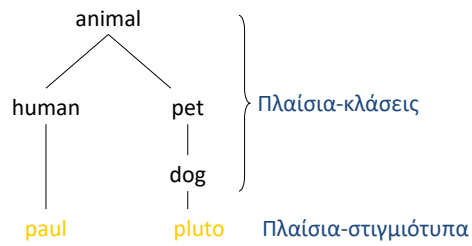
- ▶ Τα πλαίσια είναι οργανωμένα πάντα σε ιεραρχίες
- ▶ Διακρίνουμε **πλαίσια-τάξεις** (περιγραφές γενικών οντοτήτων) και **πλαίσια-στιγμιότυπα** (περιγραφές συγκεκριμένων οντοτήτων)
- ▶ Διακρίνουμε **υποπλαίσια** και **υπερπλαίσια** μεταξύ των πλαισίων-τάξης
- ▶ Κάθε πλαίσιο-τάξη (πλην του κορυφαίου) είναι υποπλαίσιο ενός γενικότερου πλαισίου-τάξης (γενίκευση)
- ▶ Ένα υποπλαίσιο μπορεί να έχει επί πλέον σχισμές από το υπερπλαίσιο του (εξειδίκευση)
- ▶ Κάθε πλαίσιο-στιγμιότυπο ανήκει σ' ένα πλαίσιο-τάξη
- ▶ Ένα πλαίσιο-στιγμιότυπο μπορεί να έχει επί πλέον σχισμές από το πλαίσιο-τάξη στο οποίο ανήκει

32

32



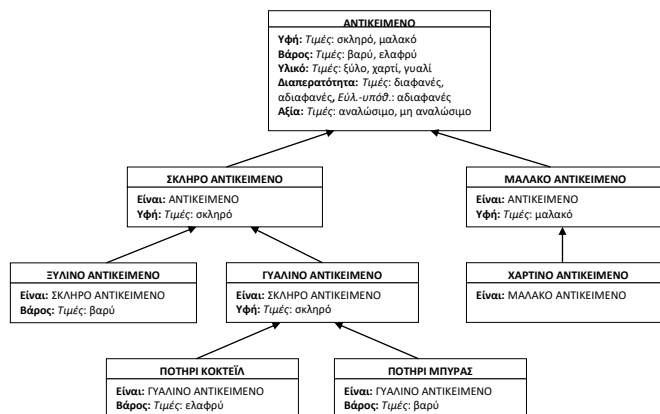
## Ιεραρχίες Πλαισίων - Παράδειγμα



33

33

## Ιεραρχίες Πλαισίων - Παράδειγμα (2)



34

34

## Πλαίσια - Χαρακτηριστικά

---

- ▶ **Τα πλαίσια παρέχουν:**
  - ▶ κληρονομικότητα
  - ▶ προσκόλληση διαδικασιών
  - ▶ προκαθορισμένες (ή εύλογες) τιμές
- ▶ Ένα πλαίσιο περιέχει **όλη** την πληροφορία για τη συγκεκριμένη έννοια που αναπαριστά
- ▶ Παράκαμψη λογικής ανεπάρκειας
  - ▶ Δεν εμφανίζουν τόσο έντονο το φαινόμενο της συνδυαστικής έκρηξης στην αναζήτηση



35

35

## Συλλογισμός με Πλαίσια

---

- ▶ Κάθε πλαίσιο **κληρονομεί χαρακτηριστικά** (σχισμές) από τα υπερπλαίσιά του
- ▶ Η κληρονόμηση είναι δυναμική
- ▶ Είναι ο μοναδικός γενικός μηχανισμός που χρησιμοποιείται στην εξαγωγή συμπερασμάτων
- ▶ **Συλλογισμός με πλαίσια σημαίνει την αναζήτηση της τιμής μιας ιδιότητας/χαρακτηριστικού ενός πλαισίου**



36

36

## Κληρονομικότητα

- ▶ Τύποι κληρονομικότητας
  - ▶ Απλή: κάθε πλαίσιο-τάξη κληρονομεί μόνο από ένα υπερπλαίσιο
  - ▶ Πολλαπλή: κάθε πλαίσιο-τάξη μπορεί να κληρονομεί από περισσότερα από ένα υπερπλαίσια
  
- ▶ Ο τύπος κληρονομικότητας εξαρτάται από την επιτρεπόμενη δομή της ιεραρχίας των πλαισίων



37

37

## Συλλογισμός με Πλαίσια - Αλγόριθμος τύπου N

**Δεδομένα:** Πλαίσιο F, Χαρακτηριστικό S, **Ζητούμενο:** τιμή του S  
**Αλγόριθμος τύπου N (απλή κληρονομικότητα)**

1. Αναζήτησε την τιμή της S στην όψη «value» της σχισμής S στο πλαίσιο F
2. Αν την βρεις, σταμάτα (επιτυχία)
3. Αν δεν υπάρχει, τότε ακολούθησε την ιεραρχία προς τα πάνω (μέχρι την κορυφή) ψάχνοντας σε κάθε υπερπλαίσιο στην όψη «value» της σχισμής S. Αν βρεις τιμή, σταμάτα (επιτυχία)
4. Αν δεν βρεις, τότε επανέλαβε τα βήματα 1-3 εξετάζοντας αυτή τη φορά την όψη «if-needed»
5. Αν το βήμα 4 αποτύχει, τότε επανέλαβε τα βήματα 1-3 εξετάζοντας αυτή τη φορά την όψη «default»
6. Αν δεν βρεθεί τιμή, σταμάτα (αποτυχία)



38

38

## Συλλογισμός με Πλαίσια - Αλγόριθμος τύπου Z

**Δεδομένα:** Πλαίσιο F, Χαρακτηριστικό S, **Ζητούμενο:** τιμή του S  
Αλγόριθμος τύπου Z (απλή κληρονομικότητα)

1. Αναζήτησε την τιμή της S στις όψεις «value», «if-needed» και «default» (μ' αυτή τη σειρά) της σχισμής S στο πλαίσιο F
2. Αν την βρεις, σταμάτα (επιτυχία)
3. Αν δεν υπάρχει, τότε ακολούθησε την ιεραρχία προς τα πάνω (μέχρι την κορυφή) ψάχνοντας σε κάθε υπερπλαίσιο στις όψεις «value», «if-needed» και «default» (μ' αυτή τη σειρά) της σχισμής S. Αν βρεις τιμή, σταμάτα (επιτυχία)
4. Αν δεν βρεις, σταμάτα (αποτυχία)



39

39

## Συλλογισμός με Πλαίσια - Αλγόριθμοι N και Z

### ▶ Αλγόριθμος τύπου N

- ▶ Δίνει προτεραιότητα στην όψη «value» και όχι στην «default», έστω και αν η τιμή βρίσκεται υψηλότερα (δηλ. σε μεγαλύτερη απόσταση) στην ιεραρχία

### ▶ Αλγόριθμος τύπου Z

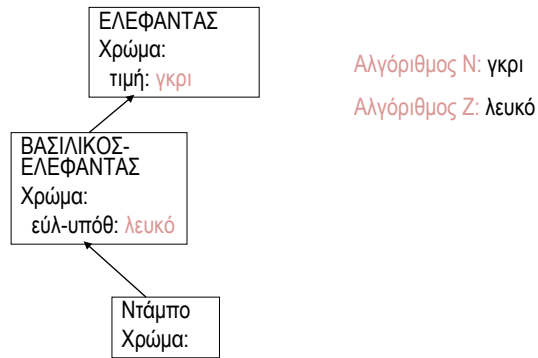
- ▶ Δίνει προτεραιότητα στην πλησιέστερη όψη τιμής, ανεξάρτητα αν είναι value ή default



40

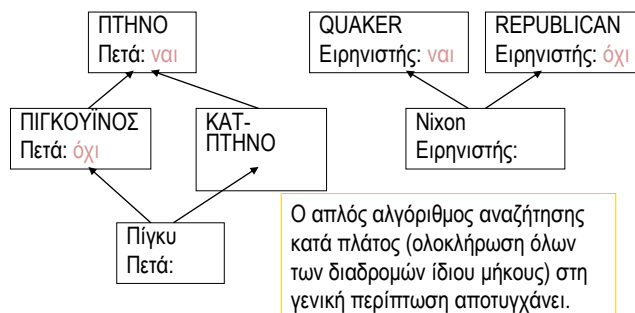
40

## Συλλογισμός με Πλαίσια - Παράδειγμα



41

## Πολλαπλή Κληρονομικότητα



42

## Απόσταση Συλλογισμού

### Απόσταση Συλλογισμού:

- ▶ Η απόσταση ενός πλαισίου F1 από ένα πλαίσιο F2 είναι μικρότερη από την απόστασή του από το πλαίσιο F3 αν και μόνο αν υπάρχει διαδρομή συλλογισμού από το F1 στο F3 δια μέσου του F2



43

43

## Συλλογισμός με Πλαίσια - Συλλογιστική Απόσταση

**Δεδομένα:** Πλαίσιο F, Χαρακτηριστικό S, **Ζητούμενο:** τιμή του S

### Αλγόριθμος Βασισμένος στη Συλλογιστική Απόσταση

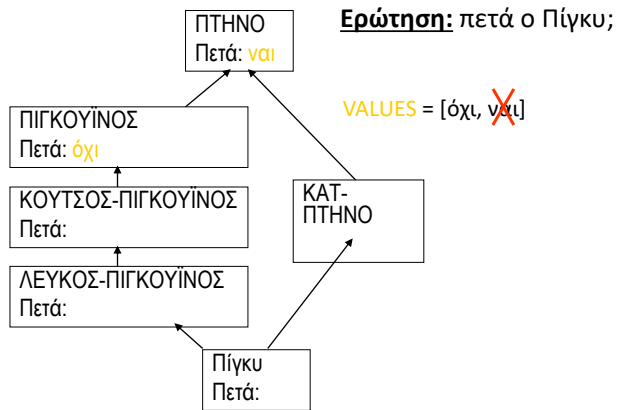
1. Εφάρμοσε αναζήτηση κατά πλάτος (ή βάθος) ακολουθώντας όλες τις δυνατές διαδρομές από το F προς τα πάνω και αποθήκευσε στη λίστα VALUES όλες τις τιμές που θα βρεις για το S
2. Για κάθε τιμή στη VALUES εξέτασε αν υπάρχει άλλη τιμή που προέρχεται από πλαίσιο που βρίσκεται σε μικρότερη συλλογιστική απόσταση από το F. Αν υπάρχει, διάγραψε την τιμή
3. Αν απομείνουν 0 τιμές, τότε δεν υπάρχει απάντηση. Αν απομείνει μια (1) τιμή είναι η απάντηση. Αν απομείνουν περισσότερες από μία, τότε υπάρχει αντίφαση



44

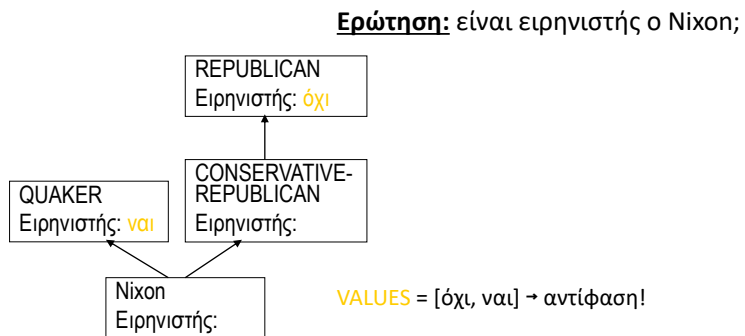
44

## Συλλογισμός με Πλαίσια - Συλλογιστική Απόσταση - Παράδειγμα



45

## Συλλογισμός με Πλαίσια - Συλλογιστική Απόσταση - Παράδειγμα (2)



46

## Κανόνες Σχεδίασης Πλαισίων

1. Περιγράφουμε κάθε τμήμα γνώσης μια φορά, ώστε να μην έχουμε πολλαπλή αναπαράσταση
  - ▶ Αυτό επιτυγχάνεται με το να το καταχωρούμε όσο το δυνατόν υψηλότερα στην ιεραρχία ώστε να κληρονομείται από το δυνατόν περισσότερα υποπλαίσια (υποκλάσεις)
2. Όσο προχωρούμε προς τα κάτω τα πλαίσια περιέχουν μόνο τη γνώση που τα διαφοροποιεί από τα υπερπλαίσιά τους
  - ▶ Αυτό σημαίνει είτε περιγραφές νέων ιδιοτήτων είτε νέες τιμές ήδη περιγραφέντων ιδιοτήτων. Δεν επαναλαμβάνουμε γνώση που έχει καταχωρηθεί παραπάνω και θα κληρονομηθεί
  - ▶ Όταν υπάρχει η λέξη "συνήθως" στην απόδοση τιμής σε ένα χαρακτηριστικό, τότε αυτό τυπικά αποτυπώνεται με μια τιμή "εύλογης υπόθεσης" (default)
3. Οι τοποθετήσεις των διαφόρων τιμών θα πρέπει να γίνονται έτσι ώστε να εξάγονται σωστές απαντήσεις με βάση κάποιον αλγόριθμο



47

47

## Κανόνες Σχεδίασης Πλαισίων (2)

4. Κάθε ιδιότητα/χαρακτηριστικό περιγράφεται ως εξής:
  - ▶ <όνομα>
  - ▶ τιμές: <τύπος ή απαρίθμηση τιμών>
  - ▶ τιμή: <πραγματική τιμή>
  - ▶ εύλ-υπόθ: <τιμή εύλογης υπόθεσης>
  - ▶ Από αυτά το <όνομα> και η όψη 'τιμές' είναι απαραίτητα στην αρχική περιγραφή μιας ιδιότητας. Η όψη 'τιμές' δεν επαναλαμβάνεται πιο κάτω στην ιεραρχία
5. Πιο κάτω στην ιεραρχία υπάρχουν περιγραφές που έχουν είτε το <όνομα> και την όψη 'τιμή' είτε το <όνομα> και την όψη 'εύλ-υπόθ'. Οι όψεις 'τιμή' και 'εύλ-υπόθ' δεν έχει νόημα να υπάρχουν μαζί στην ίδια περιγραφή στο ίδιο πλαίσιο



48

48



## Πλεονεκτήματα/Μειονεκτήματα Πλαισίων

---

### ▶ Πλεονεκτήματα

- ▶ Φυσικότητα αναπαράστασης
- ▶ Υψηλή αποδοτικότητα
- ▶ Εξ' ορισμού (ή εύλογος) συλλογισμός

### ▶ Μειονεκτήματα

- ▶ Έλλειψη σαφούς σημαντικής
- ▶ Περιορισμένη έκφραση



49

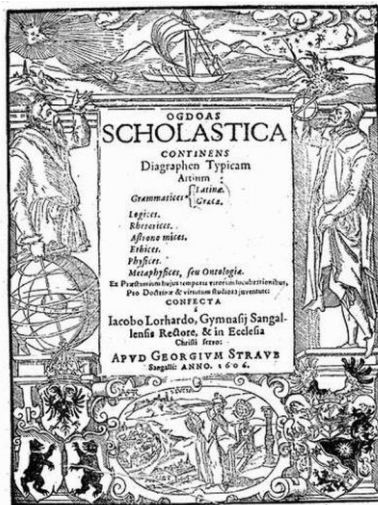
49

Οντολογίες

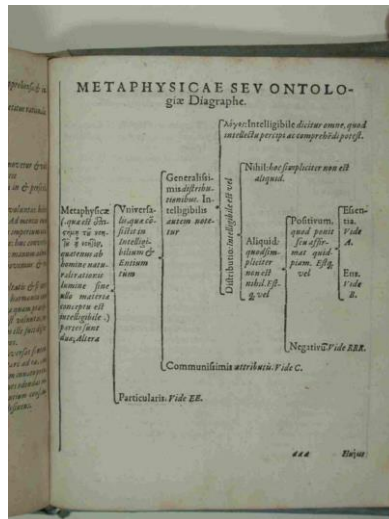
50

50

## Προέλευση



▶ 51



51

## Οντολογίες

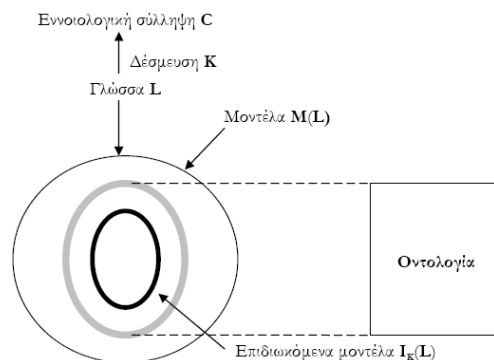
- ▶ Προσδιορισμός της ορολογίας και των σχέσεων για τη γνώση σε ένα πεδίο αναφοράς
  - ▶ Φιλοσοφία: Οριστική και εξαντλητική ταξινόμηση των οντοτήτων σε όλες τις σφαίρες της ύπαρξης
  - ▶ [Gruber '93]: Οντολογία είναι ο **ρητός προσδιορισμός** μιας **ενοσιολογικής σύλληψης** (specification of a conceptualization)
  - ▶ Οντολογική Δέσμευση (Ontological Commitment): Συμφωνία ερμηνείας του σύμπαντος αναφοράς ως προς την συγκεκριμένη ενοσιολογική σύλληψη
    - ▶ Η **ενοσιολογική σύλληψη** αφορά το ενοσιολογικό (αφηρημένο) μοντέλο ενός γνωστικού πεδίου (π.χ. ασθένεια), δηλ. τις έννοιες που το απαρτίζουν.
    - ▶ Ο **ρητός προσδιορισμός** αφορά τις σχέσεις μεταξύ των εννοιών, τον ορισμό των ιδιοτήτων των εννοιών, τους περιορισμούς στις ιδιότητες.

▶ 52

52

## Οντολογία κατά Guarino

- ▶ **Μοντέλα γλώσσας M:** Δομές που αποδίδουν σημασία στις προτάσεις μιας τυπικής γλώσσας
- ▶ **Επιδιωκόμενα μοντέλα:** Τα μοντέλα εκείνα που είναι σύμφωνα με την δέσμευση K
- ▶ Η **οντολογία** προσεγγίζει τα επιδιωκόμενα μοντέλα όσοδηποτε κοντά.



Οι οντολογίες ως εννοιολογικές μορφοποιήσεις (conceptualizations) αποτελούν προϊόντα υποκειμενικής κρίσης, οπότε το ίδιο πεδίο ενδιαφέροντος είναι δυνατόν να περιγραφεί με διαφορετικούς τρόπους.

▶ 53

53

## Δομικά στοιχεία οντολογίας

- ▶ **Κλάσεις (Classes).** Οι κλάσεις αναπαριστούν έννοιες (concepts) του πεδίου. Μια έννοια μπορεί να είναι οτιδήποτε για το οποίο μπορεί να ειπωθεί κάτι και μπορεί να αφορά μια εργασία, μια λειτουργία, μια ενέργεια, μια ιδέα, κλπ.
- ▶ **Θυρίδες (Slots) ή Ιδιότητες (Properties) ή Ρόλοι (Roles).** Αναπαριστούν διάφορα χαρακτηριστικά (features/attributes) των κλάσεων.
- ▶ **Όψεις (Facets) ή Περιορισμοί (Restrictions).** Περιορισμοί που αφορούν τις ιδιότητες/ρόλους.
- ▶ **Στιγμιότυπα (Instances).** Τα στιγμιότυπα αναπαριστούν συγκεκριμένες οντότητες που ανήκουν σε κλάσεις.
- ▶ Οι κλάσεις και τα στιγμιότυπα αποτελούν τη **βάση γνώσης** (knowledge base) μιας εφαρμογής.

▶ 54

54

## Οντολογία στο Σημαντικό Ιστό

- ▶ [W3C]: Μια οντολογία ορίζει τους όρους που χρησιμοποιούνται για την περιγραφή και την αναπαράσταση ενός πεδίου γνώσης. Παρέχει περιγραφές για:
  - ▶ Κλάσεις (γενικές έννοιες) στο πεδίο αναφοράς
  - ▶ Ιδιότητες (ή χαρακτηριστικά) που περιγράφουν και συσχετίζουν τις κλάσεις
  - ▶ Στιγμιότυπα των κλάσεων αυτών

▶ 55

55

## Παράδειγμα Οντολογίας Ιστού

The screenshot displays a web ontology editor interface. On the left, a tree view shows the class hierarchy, including 'owl:Thing', 'adult', 'animal', 'bone', 'brain', 'comp', 'dog', 'fema', 'haul', 'leaf', and 'male'. The main area is divided into two panes: 'Description: Mick' and 'Property assertions: Mick'. The 'Description' pane shows the 'male' class with its subclasses 'dog\_owner' and 'white\_van\_man', and lists 'Same individuals' and 'Different individuals'. The 'Property assertions' pane shows 'drives Q123\_ABC' and 'reads Daily\_Mirror'. A 'Synchronisation' checkbox is visible at the bottom left. A floating window titled 'Individuals By Class' is shown in the foreground, listing classes like 'cow', 'van', 'duck', 'owl:Thing', 'tabloid', 'tiger', 'male', 'Mick', 'cat', and 'femal', with 'Mick' highlighted.

▶ 56

56

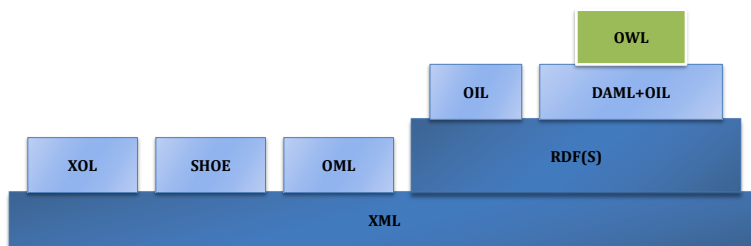
## Κλάσεις

- ▶ Αποτελούν τα βασικότερα στοιχεία μιας οντολογίας και αντιστοιχούν σε έννοιες, που αντιπροσωπεύουν σύνολα οντοτήτων.
  - ▶ Π.χ. η κλάση Wine αντιπροσωπεύει όλα τα κρασιά
- ▶ Αποτελούνται από ιδιότητες, οι οποίες περιγράφονται με περιορισμούς.
  - ▶ Π.χ. η maker είναι μια ιδιότητα της wine και το Winery είναι περιορισμός της τιμής της maker-την υποχρεώνει να παίρνει ως τιμές οντότητες/στιγμιότυπα που ανήκουν στην κλάση Winery
- ▶ Είναι οργανωμένες σε ιεραρχίες και σχετίζονται μεταξύ τους με τη σχέση υποκλάση (subclass).
  - ▶ Μια υποκλάση είναι εξειδίκευση μιας κλάσης, δηλ. αντιπροσωπεύει ένα υποσύνολο οντοτήτων της κλάσης.
  - ▶ Π.χ. RedWine μπορεί να είναι υποκλάση της Wine, αντιπροσωπεύοντας τα κόκκινα κρασιά, ένα υποσύνολο των κρασιών.

▶ 57

57

## Γλώσσες Αναπαράστασης Οντολογιών



▶ 58

58

## OWL: Ontology Web Language

- ▶ OWL: Γλώσσα Οντολογιών Ιστού
- ▶ Επέκταση (μέρους) του RDF(S)
  - ▶ Περισσότερη εκφραστικότητα
  - ▶ Αντιστοιχεί σε συγκεκριμένη Λογική
  - ▶ Δεν έχει τη σημασιολογία του RDF(S)
    - ▶ Δεν είναι πλήρως συμβατή με Horn προτάσεις
- ▶ Χρήση RDF/XML σύνταξης
  - ▶ Όμως **νέο λεξιλόγιο**
    - ▶ owl:Class αντί για rdfs:Class
    - ▶ owl:ObjectProperty & owl:DataProperty
  - ▶ OWL Lite → OWL DL → RDF(S) ← → OWL Full

▶ 59

59

## Κλάσεις

- ▶ Δύο προκαθορισμένες κλάσεις:
  - ▶ owl:Thing: Τα πάντα
  - ▶ owl:Nothing: Το τίποτα
- ▶ Ορισμός (υπο-) κλάσης:

```
<owl:Class rdf:ID="WineGrape"> (αντί για rdfs:Class)
  <rdfs:subClassOf rdf:resource="&food;Grape"/>
</owl:Class>
```
- ▶ Στιγμιότυπα: Άτομα (individuals)
  - ▶ Γνωρίζουμε σε ποια κλάση ανήκουν:

```
<WineGrape rdf:ID="CabernetSauvignonGrape" />
```
  - ▶ **Δεν** γνωρίζουμε σε ποια κλάση ανήκουν:

```
<owl:Thing rdf:ID="CabernetSauvignonGrape" />
```

▶ 60

60

## Ιδιότητες

### ▶ Object Properties

- ▶ Συνδέουν ένα άτομο με ένα άλλο άτομο

```
<owl:ObjectProperty rdf:ID="madeFromGrape">  
  <rdfs:subPropertyOf rdf:resource="#food;madeFromFruit" />  
  <rdfs:domain rdf:resource="#Wine"/>  
  <rdfs:range rdf:resource="#WineGrape"/>  
</owl:ObjectProperty>
```

### ▶ Datatype Properties

- ▶ Συνδέουν ένα άτομο με ένα λεκτικό (literal)

```
<owl:DatatypeProperty rdf:ID="yearValue">  
  <rdfs:domain rdf:resource="#VintageYear" />  
  <rdfs:range rdf:resource="#xsd:positiveInteger"/>  
</owl:DatatypeProperty>
```

### ▶ Τύποι Δεδομένων

- ▶ Από το XML Schema ή rdfs:Literal

### ▶ Ιεραρχία ιδιοτήτων

- ▶ rdfs:subPropertyOf

▶ 61

61

## Παράδειγμα δήλωσης ιδιοτήτων

### ▶ Object Property assertion

```
<wine rdf:ID="myWine">  
  <madeFromGrape rdf:resource="# SauvignonBlancGrape"/>  
</wine>
```

### ▶ Data Property assertion

```
<VintageYear rdf:ID="Year1998">  
  <yearValue rdf:datatype="#xsd:positiveInteger">1998</yearValue>  
</VintageYear>
```

- ▶ Ρητή ανάθεση τύπων στα λεκτικά

▶ 62

62

## Σύνθετες Κλάσεις

### ▶ Τομή

```
<owl:Class rdf:ID="WhiteWine">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Wine" />
    <owl:Class rdf:about="#isWhite"/>
  </owl:intersectionOf>
</owl:Class>
```

### ▶ Ένωση

```
<owl:Class rdf:ID="Fruit">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#SweetFruit" />
    <owl:Class rdf:about="#NonSweetFruit" />
  </owl:unionOf>
</owl:Class>
```

### ▶ Συμπλήρωμα

```
<owl:Class rdf:ID="NonConsumableThing">
  <owl:complementOf rdf:resource="#ConsumableThing" />
</owl:Class>
```

▶ 63

63

## Σύνθετες Κλάσεις (2)

### ▶ Απαρίθμηση

```
<owl:Class rdf:ID="WineColor">
  <rdfs:subClassOf rdf:resource="#WineDescriptor"/>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#White"/>
    <owl:Thing rdf:about="#Rose"/>
    <owl:Thing rdf:about="#Red"/>
  </owl:oneOf>
</owl:Class>
```

### ▶ Ισοδύναμες Κλάσεις

```
<owl:Class rdf:ID="isWhite">
  <owl:equivalentClass rdf:resource="hasWhiteColor" />
</owl:Class>
```

### ▶ Ξένες κλάσεις

```
<owl:Class rdf:ID="NonSweetFruit">
  <owl:disjointWith rdf:resource="#SweetFruit" />
</owl:Class>
```

▶ 64

64



## Περιορισμοί στις Ιδιότητες

- ▶ Ποσοτικοποίηση (quantification)
  - ▶ Υπαρξιακή ποσοτικοποίηση
    - ▶ owl:someValuesFrom
  - ▶ Περιορισμός Τιμής
    - ▶ owl:allValuesFrom
- ▶ Πληθικότητα (cardinality)
  - ▶ owl: (min/max) cardinality
- ▶ Ανάθεση τιμής (hasValue)
  - ▶ owl:hasValue *individual*

**Κάθε περιορισμός ιδιότητας ισοδυναμεί με μια νέα κλάση!**

▶ 65

65

## owl:someValuesFrom

- ▶ Υπαρξιακή ποσοτικοποίηση (existential quantification)

```
<owl:Class rdf:ID="Wine">  
<rdfs:subClassOf>  
  <owl:Restriction>  
    <owl:onProperty rdf:resource="#hasMaker" />  
    <owl:someValuesFrom rdf:resource="#Winery" />  
  </owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

- ▶ Τοπική εμβέλεια πεδίου ορισμού

▶ 66

66

## owl:allValuesFrom

---

### ▶ Περιορισμός Τιμής (value restriction)

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:allValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### ▶ Τοπική εμβέλεια πεδίου τιμών

---

▶ 67

67

## owl:hasValue

---

### ▶ Υπαρξιακή Ποσοτικοποίηση με Άτομο

```
<owl:Class rdf:ID="WhiteWine">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#hasColor" />
    <owl:hasValue rdf:resource="#White" />
  </owl:Restriction>
</owl:Class>
```

---

▶ 68

68

## Πληθικότητα

---

### ▶ Περιορισμοί Αριθμού (Πληθικότητας)

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#madeFromGrape"/>
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

- ▶ owl:maxCardinality, owl:cardinality

---

▶ 69

69

## Χαρακτηριστικά των ιδιοτήτων

---

- ▶ Μεταβατικές
  - ▶ *owl:transitiveProperty*
- ▶ Συμμετρικές
  - ▶ *owl:symmetricProperty*
- ▶ Συναρτησιακές
  - ▶ *owl:functionalProperty*
- ▶ Αντίστροφες
  - ▶ *owl:inverseOf*
- ▶ Αντιστρόφως συναρτησιακές
  - ▶ *owl:inverseFunctionalProperty*

---

▶ 70

70

## Μεταβατικότητα

```
<owl:ObjectProperty rdf:ID="locatedIn">  
  <rdf:type rdf:resource="&owl;TransitiveProperty" />  
</owl:ObjectProperty>
```



▶ 71

71

## Συμμετρία

```
<owl:ObjectProperty rdf:ID="adjacentRegion">  
  <rdf:type rdf:resource="&owl;SymmetricProperty" />  
</owl:ObjectProperty>
```



▶ 72

72

## Συναρτησιακότητα

```
<owl:ObjectProperty rdf:ID="hasVintageYear">  
  <rdf:type rdf:resource="&owl;FunctionalProperty" />  
  <rdfs:domain rdf:resource="#Vintage" />  
  <rdfs:range rdf:resource="#VintageYear" />  
</owl:ObjectProperty>
```

► Ισοδυναμεί με:

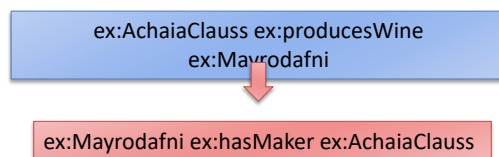
```
<owl:Class rdf:about="#Vintage">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#hasVintageYear">  
        <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
</owl:Class>
```

► 73

73

## Αντίστροφες Ιδιότητες

```
<owl:ObjectProperty rdf:ID="producesWine">  
  <owl:inverseOf rdf:resource="#hasMaker" />  
</owl:ObjectProperty>
```



► 74

74

## Αντίστροφη Συναρτησιακότητα

```
<owl:ObjectProperty rdf:ID="producesWine">  
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty" />  
  <owl:inverseOf rdf:resource="#hasMaker" />  
</owl:ObjectProperty>
```

- ▶ Ισοδυναμεί με:

```
<owl:ObjectProperty rdf:ID="hasMaker">  
  <rdf:type rdf:resource="&owl;FunctionalProperty" />  
</owl:ObjectProperty>
```

- ▶ Έννοια κλειδιού όπως στις ΒΔ
  - ▶ Αλλά όχι για datatype properties!

▶ 75

75

## Ταυτότητα Ατόμων

- ▶ Ταυτότητα μεταξύ ατόμων

```
<Wine rdf:ID="MikesFavoriteWine">  
  <owl:sameAs rdf:resource="#StGenevieveTexasWhite" />  
</Wine>
```

- ▶ Διαφορετικά άτομα

```
<WineSugar rdf:ID="Sweet">  
  <owl:differentFrom rdf:resource="#Dry" />  
</WineSugar>
```

- ▶ Αμοιβαίως αποκλειόμενα άτομα

```
<owl:AllDifferent>  
  <owl:distinctMembers rdf:parseType="Collection">  
    <vin:WineColor rdf:about="#Red" />  
    <vin:WineColor rdf:about="#White" />  
    <vin:WineColor rdf:about="#Rose" />  
  </owl:distinctMembers>  
</owl:AllDifferent>
```

▶ 76

76

## Ισοδυναμίες

- ▶ Ισοδυναμία κλάσεων

```
<owl:Class rdf:ID="TexasThings">  
  <owl:equivalentClass>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#locatedIn" />  
      <owl:someValuesFrom rdf:resource="#TexasRegion" />  
    </owl:Restriction>  
  </owl:equivalentClass>  
</owl:Class>
```

- ▶ *Ισοδυναμεί με (αμφίδρομες) σχέσεις υποκλάσης!*

- ▶ Ισοδυναμία Ιδιοτήτων

- ▶ owl:equivalentProperty

▶ 77

77

## Συλλογισμός με οντολογίες

- ▶ Συλλογισμός:

- ▶ Συνδυασμός εκφρασμένων γεγονότων ώστε:

- ▶ Έλεγχος ορθότητας άλλων γεγονότων

- ▶ Παραγωγή νέων γεγονότων

- ▶ Ικανοποιησιμότητα έννοιας ( $C \neq \emptyset$ )

- ▶ Έλεγχος Υπαγωγής ( $C \sqsubseteq D \Leftrightarrow C \sqcap \neg D$ )

- ▶ Αναγωγή οντολογίας σε:

- ▶ Σύνολο αξιωμάτων και γεγονότων ΛΠ

- ▶ Σύνολο κανόνων Horn (DLP)

- ▶ Σύνολο προτάσεων ΛΠΤ

- ▶ Χρήση του αντίστοιχου αλγορίθμου

- ▶ Ταμπλό (tableau)

- ▶ Αλυσίδωση (chaining), RETE

- ▶ Ανάλυση (resolution)

▶ 78

78

## Λογικές Περιγραφής (Description Logics)

- ▶ Καλώς ορισμένο υποσύνολο ΛΠΤ
  - ▶ Οι τύποι έχουν δύο μεταβλητές
  - ▶ Αντιστοιχία με *Σημαντικά Δίκτυα και Πλαίσια*
- ▶ Παρέχει το λογικό υπόβαθρο για διεξαγωγή συλλογισμού στο Σημαντικό Ιστό
  - ▶ Η OWL έχει σχεδιαστεί βάσει συγκεκριμένων ΛΠ
- ▶ Υπάρχουν υλοποιημένοι αλγόριθμοι και συστήματα
  - ▶ RACER, FaCT++, Pellet...
  - ▶ Μηχανισμοί συμπερασμού (inference engines)

▶ 79

79

## Σύνταξη και Ορολογία

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human $\sqcap$ Male	$C_1(x) \wedge \dots \wedge C_n(x)$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1(x) \vee \dots \vee C_n(x)$
complementOf	$\neg C$	$\neg$ Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x = x_1 \vee \dots \vee x = x_n$
allValuesFrom	$\forall P.C$	$\forall$ hasChild.Doctor	$\forall y.P(x, y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	$\exists$ hasChild.Lawyer	$\exists y.P(x, y) \wedge C(y)$
maxCardinality	$\leq nP$	$\leq 1$ hasChild	$\exists^{\leq n} y.P(x, y)$
minCardinality	$\geq nP$	$\geq 2$ hasChild	$\exists^{\geq n} y.P(x, y)$

- ▶ Έννοιες (Concepts)
  - ▶  $C, D, \dots$  αντιστοιχούν στις **κλάσεις**
- ▶ Ρόλοι (Roles)
  - ▶  $R, P, \dots$  αντιστοιχούν στις **ιδιότητες**
- ▶ Στιγμιότυπα (Instances)
  - ▶  $a, o, \dots$  αντιστοιχούν στα **άτομα**

▶ 80

80



## Χαρακτηριστικά των Λογικών Περιγραφής

- ▶ Έννοιες, στιγμιότυπα, ρόλοι
- ▶ Κατασκευαστές
  - ▶ Τομή, Ένωση, Συμπλήρωμα ( $\text{Άνθρωπος} \sqcap \text{Άρσενικό}$ )
  - ▶ Υπαρξιακή Ποσοτικοποίηση ( $\exists \text{έχειΠαιδί.Άνθρωπος}$ )
  - ▶ Περιορισμός Τιμής ( $\forall \text{έχειΠαιδί.Αγόρι}$ )
  - ▶ Περιορισμοί Αριθμού (Πληθικότητα) ( $\geq 3 \text{έχειΠαιδί}$ )
- ▶ Συμπερασμός
  - ▶ Υπαγωγή ( $\text{Αγόρι} \sqsubseteq \text{Άνθρωπος}$ )
- ▶ Αξιώματα
  - ▶ Σχέσεις υπαγωγής
- ▶ Γεγονότα
  - ▶ Δηλώσεις για τα στιγμιότυπα

▶ 81

81

## Ονοματολογία

ALC	Attribute Language with Complement
(D)	Απτά Πεδία (τύποι δεδομένων)
F	Συναρτησιακοί Ρόλοι (έχουν το πολύ έναν πληρωτή)
H	Ίεραρχία Ρόλων (έχειΠαιδί, έχειΚόρη)
I	Αντίστροφοι Ρόλοι (έχειΠαιδί, έχειΓονέα)
O	Ονοματικά ( $\Sigma/K \equiv \{\Sigma\beta\beta\alpha\tau\sigma\} \sqcup \{\text{Κυριακή}\}$ )
Q	Προσδιορισμένοι Περιορισμοί Αριθμού $\geq 3 \text{έχειΠαιδί.Αγόρι}$
N	Απροσδιόριστοι Περιορισμοί Αριθμού $\leq 1 \text{έχειΠαιδί}$
R+	Μεταβατικοί Ρόλοι (έχειΑδελφό)
S	ALCR+

▶ 82

82

## Αντιστοιχία με OWL

- ▶ Η OWL (DL/Lite) αντιστοιχεί σε ΛΠ
  - ▶ OWL Lite  $\rightarrow$  SHIF(D)
  - ▶ OWL DL  $\rightarrow$  SHOIN (D)
  - ▶ Οντολογία  $\equiv$  Βάση Γνώσης ΛΠ

### Αξιώματα

OWL Syntax	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
transitiveProperty	$P^+ \sqsubseteq P$	ancestor <sup>+</sup> $\sqsubseteq$ ancestor

### Γεγονότα

OWL Syntax	DL Syntax	Example
type	$a : C$	John : Happy-Father
property	$\langle a, b \rangle : R$	$\langle$ John, Mary $\rangle$ : has-child

▶ 83

83

## Γιατί ΛΠ;

- ▶ Πλεονεκτήματα
  - ▶ Υπάρχουν συνεπείς και πλήρεις αλγόριθμοι (ταμπλό)
    - ▶ Αποφασίζουν το σύνολο της OWL (και OWL 2)
  - ▶ Υλοποιημένα συστήματα ΛΠ
    - ▶ Βελτιστοποιήσεις με δραματική βελτίωση της απόδοσης
  - ▶ Ευθεία αντιστοιχία της OWL προς τις ΛΠ
    - ▶ Σχεδιαστικός στόχος
- ▶ Μειονεκτήματα
  - ▶ Υψηλή πολυπλοκότητα
  - ▶ Καλύτερες επιδόσεις των άλλων προσεγγίσεων
    - ▶ Για λιγότερο εκφραστικές υπο-γλώσσες (π.χ. OWL Lite)
    - ▶ Για Μικρά TBox και μεγάλα ABox.

▶ 84

84

## Πλεονεκτήματα Συλλογισμού με Οντολογίες

---

- ▶ Σχεδιασμός και συντήρηση ποιοτικών οντολογιών
  - ▶ Συνεπείς — όλες οι κλάσεις μπορούν να έχουν στιγμιότυπα
  - ▶ Ορθές — αντανακλούν τη διαίσθηση των domain experts
  - ▶ Χωρίς περιττούς πλεονασμούς— ανίχνευση συνώνυμων
- ▶ Απάντηση ερωτημάτων στις κλάσεις και τα στιγμιότυπα της οντολογίας
  - ▶ Εύρεση γενικότερων/ειδικότερων κλάσεων
  - ▶ Ανάκτηση ατόμων/πλειάδων που ταιριάζουν στο συγκεκριμένο ερώτημα
- ▶ Ολοκλήρωση και ευθυγράμμιση (alignment) οντολογιών