



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών

# Τίτλος Μαθήματος

Ενότητα 5: Λογικός προγραμματισμός για  
αναπαράσταση γνώσης

Ιζαμπώ Καράλη

Σχολή Θετικών Επιστημών

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

# Περιγραφή ενότητας

Μεθοδολογίες αναπαράστασης γνώσης και υλοποιήσεις τους με χρήση της Prolog.  
Συστήματα βασισμένα στη γνώση, η γλώσσα Datalog και σύνδεση του λογικού προγραμματισμού με τον παγκόσμιο ιστό.



Λογικός προγραμματισμός για  
αναπαράσταση γνώσης

# Αναπαράσταση γνώσης και συλλογιστική

- Στην πράξη, πολλά προβλήματα εμπλέκουν ένα σύνολο από αντικείμενα τα οποία συσχετίζονται και αλληλεπιδρούν μεταξύ τους.
- Ακόμα και τα ίδια τα αντικείμενα μπορεί να εμπεριέχουν πληθώρα χαρακτηριστικών/ιδιοτήτων ή οι αλληλεπιδράσεις που έχουν μεταξύ τους να είναι πολύπλοκες.
- Για μεγάλο όγκο πληροφορίας, γίνεται αναγκαία μια πιο δομημένη οργάνωση της γνώσης.



# Αναπαράσταση γνώσης μέσω σημασιολογικών δικτύων (1/5)

- Ένα δίκτυο επιτρέπει να γίνεται εμφανής η οργάνωση της γνώσης.
- Κομμάτια γνώσης που σχετίζονται μεταξύ τους συγκεντρώνονται σε συστάδες μέσα στο δίκτυο.
- Η αναπαράσταση γνώσης γίνεται μέσα από μια εικόνα.
- Semantic Networks (Associative Networks): αναδείχθηκαν από τον Quillian (1968) για την απεικόνιση της σημασίας των προτάσεων και λέξεων της αγγλικής γλώσσας.



# Αναπαράσταση γνώσης μέσω σημασιολογικών δικτύων (2/5)

- Γράφος με κόμβους
- οντότητες (entities) και
- ιδιότητες οντοτήτων.
- Οι ακμές παριστάνουν σχέσεις (relations).
- Δεν υπάρχει καθολικά αποδεκτή σύνταξη.
- Προσπάθειες για την καθιέρωση προτύπων, π.χ. Sowa (1974): Conceptual Graphs.
- Έχουν υιοθετηθεί διάφορα σύνολα ιδιοτήτων από διάφορες κοινότητες χρηστών, π.χ. Γλωσσολογία.

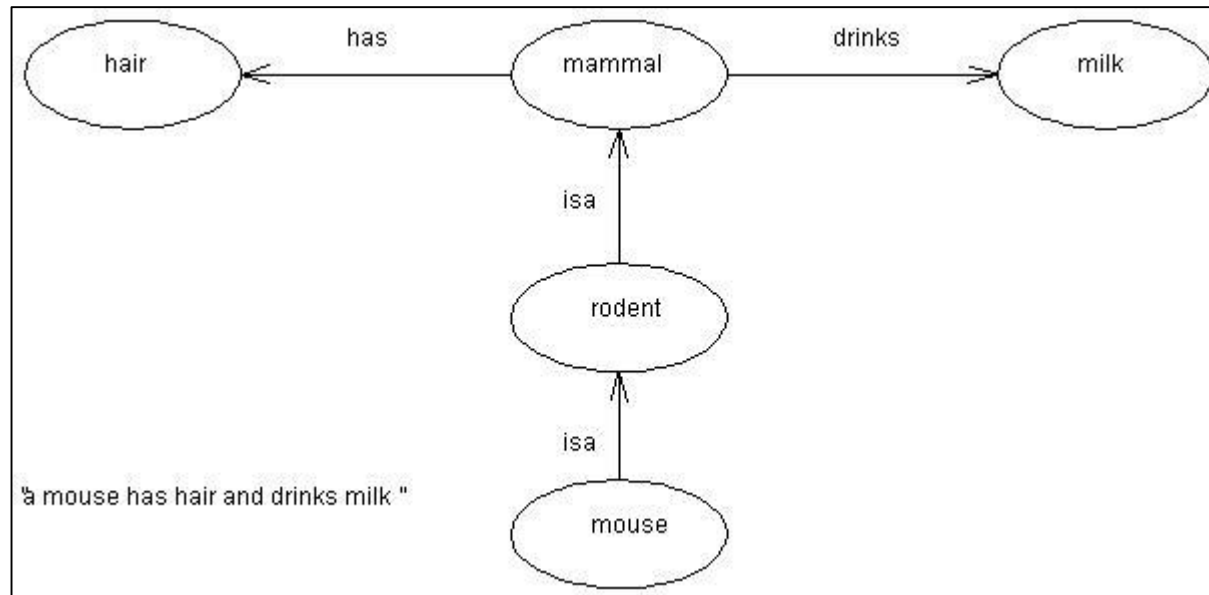


# Αναπαράσταση γνώσης μέσω σημασιολογικών δικτύων (3/5)

- Χρησιμοποιούνται ευρέως για να αναπαραστήσουν ιεραρχίες οντοτήτων (ή ταξονομίες).
- Μια ειδική σχέση είναι η “isa” που συνδέει ειδικότερες με γενικότερες κλάσεις αντικειμένων καθώς και συγκεκριμένα αντικείμενα/στιγμιότυπα (instances) με κλάσεις αντικειμένων.
- Κληρονομικότητα (inheritance)
- Σχέση «ανήκειν» και υποσυνόλου για σύνολα: Οι κόμβοι που αντιστοιχούν σε οντότητες που «ανήκουν» σε οντότητες άλλων κόμβων ή οι οντότητες που είναι υποσύνολα οντοτήτων άλλων κόμβων κληρονομούν ιδιότητες από προγονικούς κόμβους.
- Εκτός κι αν έχει διευκρινιστεί το αντίθετο, μια οντότητα κληρονομεί χαρακτηριστικά και ιδιότητες από προγονικές οντότητες.
- Εύκολη υλοποίηση σε prolog.

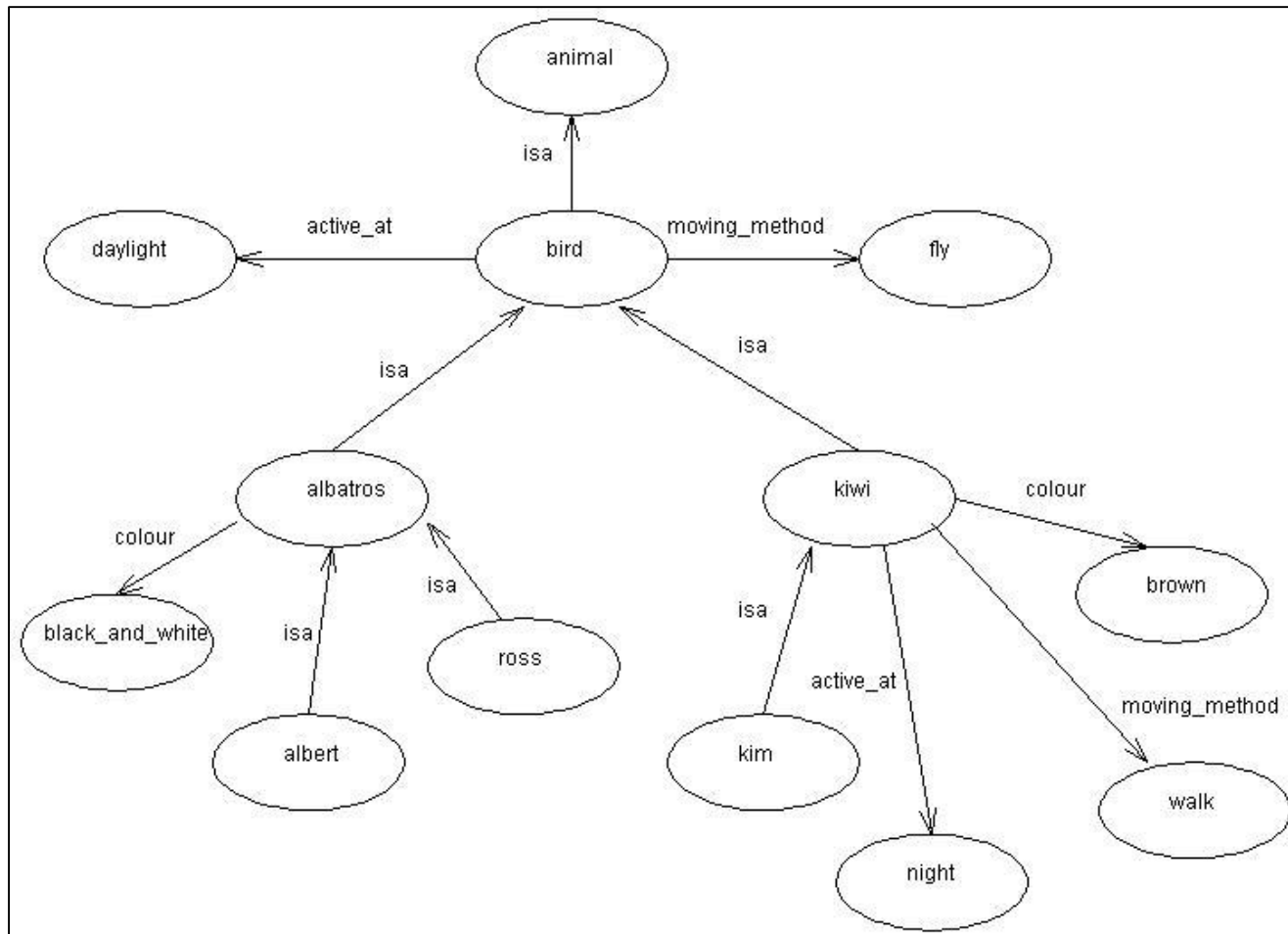


# Αναπαράσταση γνώσης μέσω σημασιολογικών δικτύων (4/5)





# Αναπαράσταση γνώσης μέσω σημασιολογικών δικτύων (5/5)



# Υλοποίηση σε Prolog

```
isa (bird, animal) .  
isa (albatross, bird) .  
isa (albert, albatross) .  
isa (ross, albatross) .  
isa (kiwi, bird) .  
isa (kim, kiwi) .
```

```
active_at (bird, daylight) .  
active_at (kiwi, night) .  
moving_method (bird, fly) .  
moving_method (kiwi, walk) .  
colour (albatross, black_and  
white) .  
colour (kiwi, brown) .
```



# Κληρονόμηση ιδιοτήτων σε σημασιολογικά δίκτυα: στοιχειώδης προσέγγιση

- Ειδικός ορισμός για κάθε ιδιότητα:
- `moving_method(X, Method) :-  
    isa(X, SuperX),  
    moving_method(SuperX, Method) .`
- `colour(...`
- `active_at(...`
- ...



# Κληρονόμηση ιδιοτήτων σε σημασιολογικά δίκτυα: γενικευμένη προσέγγιση (1/2)

- Κώδικας:

```
fact (Fact) :-  
    call (Fact), !.
```

```
fact (Fact) :-  
    Fact =.. [Rel, Arg1, Arg2],  
    isa (Arg1, SuperArg),  
    SuperFact =..  
[Rel, SuperArg, Arg2],  
    fact (SuperFact ).
```



# Κληρονόμηση ιδιοτήτων σε σημασιολογικά δίκτυα: γενικευμένη προσέγγιση (2/2)

- Παραδείγματα ερωτημάτων:

```
?- fact (moving_method (kim, Method) ) .
```

```
Method = walk
```

```
?- fact (moving_method (albert, Method) ) .
```

```
Method = fly
```



# Πλαίσια (1/3)

- Συγγενής μεθοδολογία με αυτή των σημασιολογικών δικτύων.
- Εισήχθησαν από τον Minsky (1974) για να αναπαραστήσουν ένα διανοητικό μοντέλο για μια κατάσταση, π.χ. οδήγηση αυτοκινήτου, γεύμα σε εστιατόριο.
- Η γνώση που σχετίζεται με μια οντότητα συσσωρεύεται στη μνήμη και αποτελεί μια μονάδα γνώσης.
- Οργάνωση της γνώσης σε πλαίσια που περιγράφουν αντικείμενα (συγκεκριμένα στιγμιότυπα ή γενικότερες κλάσεις).
- Ένα πλαίσιο είναι μια δομή δεδομένων που περιλαμβάνει σχισμές (slots) οι οποίες έχουν ονόματα και τιμές.



## Πλαίσια (2/3)

- Η τιμή μιας σχισμής μπορεί να είναι απλή, αναφορά σε άλλο πλαίσιο, διαδικασία υπολογισμού της πραγματικής τιμής ή απροσδιόριστη.
- Αντίστοιχα με τα σημασιολογικά δίκτυα, τα πλαίσια μπορούν να συνδεθούν μεταξύ τους μέσω ειδικών σχισμών (π.χ. `ako`) και να δομήσουν ιεραρχίες πλαισίων.
- Ειδικές σχισμές είναι οι “`a_kind_of`” για τη σχέση κλάσεων-υπερκλάσεων και “`instance_of`” για τη σχέση στιγμιοτύπων-κλάσεων.
- Η μεθοδολογία των πλαισίων παρουσιάζει πολλά κοινά στοιχεία με την αντικειμενοστραφή προσέγγιση.



# Πλαίσια (3/3)

```
FRAME: bird  
a_kind_of: animal  
moving_method: fly  
active_at: daylight
```

```
FRAME: albatross  
a_kind_of: bird  
colour: black_and_white  
size: 115
```

```
FRAME: kiwi  
a_kind_of: bird  
moving_method: walk  
active_at: night  
colour: brown  
size: 40
```

```
FRAME: albert  
instance_of: albatross  
size: 120
```





# Υλοποίηση σε Prolog (1/2)

```
bird(a_kind_of, animal).  
bird(moving_method, fly).  
bird(active_at, daylight).
```

```
albatross(a_kind_of, bird).  
albatross(colour, black_and_white).  
albatross(size, 115).
```

```
kiwi(a_kind_of, bird).  
kiwi(moving_method, walk).  
kiwi(active_at, night).  
kiwi(size, 40).  
kiwi(colour, brown).
```



# Υλοποίηση σε Prolog (2/2)

```
albert(instance_of, albatross) .  
albert(size, 120) .
```

```
ross(instance_of, albatross) .  
ross(size, 40) .
```

```
animal(relative_size,  
        execute(relative_size(Object, Value),  
                Object,  
                Value)) .
```



# Απλός υπολογισμός των τιμών των σχισμών (1/2)

- Δεν καλύπτει τον υπολογισμό τιμών με χρήση διαδικασιών

- `value(Frame, Slot, Value) :-`

```
    Query =.. [Frame, Slot, Value],  
    call(Query), !.
```

```
value(Frame, Slot, Value) :-
```

```
    parent(Frame, ParentFrame),
```

```
    value(ParentFrame, Slot, Value).
```

```
parent(Frame, ParentFrame) :-
```

```
    (Query =.. [Frame, a_kind_of, ParentFrame]);
```

```
    Query =..
```

```
[Frame, instance_of, ParentFrame]),
```

```
    call(Query).
```



# Απλός υπολογισμός των τιμών των σχισμών (2/2)

```
?-value(albert,active_at,AlbertTime).
```

```
AlbertTime = daylight
```

```
?-value(kiwi,active_at,KiwiTime).
```

```
KiwiTime = night
```



# Έστω ότι έχουμε ακόμα τη διαδικασία

```
relative_size(Object, RelativeSize) :-  
    value(Object, size, ObjSize),  
    value(Object, instance_of, ObjClass),  
    value(ObjClass, size, ClassSize),  
    RelativeSize is ObjSize/ClassSize *  
    100.
```

- Θέλουμε να μπορούμε να απαντάμε και ερωτήματα όπως:

```
?-value(ross, relative_size, R).
```

R = 34.78



# Υπολογισμός των τιμών των σχισμών καλύπτοντας και την περίπτωση των διαδικασιών (1/2)

```
value (Frame, Slot, Value) :-
```

```
    value (Frame, Frame, Slot, Value) .
```

```
value (Frame, SuperFrame, Slot, Value) :-
```

```
    Query =.. [SuperFrame, Slot, Information],  
    call (Query) ,
```

```
    process (Information, Frame, Value) , ! .
```

```
value (Frame, SuperFrame, Slot, Value) :-
```

```
    parent (SuperFrame, ParentSuperFrame) ,
```

```
    value (Frame, ParentSuperFrame, Slot, Value) .
```



## Υπολογισμός των τιμών των σχισμών καλύπτοντας και την περίπτωση των διαδικασιών (2/2)

```
process (execute (Goal, Frame, Value) , Frame, Value) :- !,  
    call (Goal) .
```

```
process (Value, _, Value) .
```

```
parent (Frame, ParentFrame) :-  
    (Query =.. [Frame, a_kind_of, ParentFrame];  
     Query =.. [Frame, instance_of, ParentFrame]),  
    call (Query) .
```



# Επαγωγικές (συμπερασματικές) βάσεις δεδομένων – datalog (1/3)

- Δεδομένα (απλά ή σύνθετα) VS γνώση
- Συστήματα βασισμένα σε γνώση – συστήματα βάσης γνώσης VS συστήματα διαχείρισης βάσης γνώσης (τα τελευταία χρειάζονται, π.χ., αποδοτική προσπέλαση, χειρισμό δοσοληψιών)
- Δηλωτική γλώσσα
  - Για χειρισμό δεδομένων
  - Σαν φιλόξενη γλώσσα (host language)
- Οι δηλωτικές γλώσσες συχνά βασίζονται σε κάποια μορφή λογικής, π.χ. SQL βασίζεται στο σχεσιακό λογισμό (relational calculus)





# Επαγωγικές (συμπερασματικές) βάσεις δεδομένων – datalog (2/3)

- ΓΕΓΟΝΟΣ:
  - θετική η ανάμιξη της λογικής και των βάσεων δεδομένων
  - θετική η ανάμιξη της τεχνητής νοημοσύνης και των τεχνολογιών των βάσεων δεδομένων για την ανάπτυξη βάσεων γνώσης
- Η προσπάθεια στην ενοποίηση εστιάζει σε θέματα:
  - Εκφραστικότητα (expressiveness) και σημασιολογίας (semantics)
  - Βελτιστοποίηση των ερωτήσεων που εκφράζονται σαν λογικά προγράμματα
  - Το πρώτο είναι πολύ κρίσιμο για τις επαγωγικές βάσεις δεδομένων



# Επαγωγικές (συμπερασματικές) βάσεις δεδομένων – datalog (3/3)

- Απευθείας πρόσβαση στα δεδομένα από το σύστημα του λογικού προγραμματισμού: η περίπτωση της datalog
- Λογικοί κανόνες με συναρτησιακά σύμβολα είναι εξίσου ισχυροί με μια μηχανή Turing
- Ακόμα και χωρίς συναρτησιακά σύμβολα, οι λογικοί κανόνες έχουν τη δυνατότητα να εκφράσουν υπολογισμούς που δεν εκφράζονται στις συμβατικές γλώσσες χειρισμού δεδομένων, π.χ. τη μεταβατική κλειστότητα μιας σχέσης («Ποιες είναι οι ιεραρχίες managers;»)
- Η λογική πρώτης τάξης (first order logic) μπορεί να αποτελέσει τόσο ένα μέσο αναπαράστασης της γνώσης όσο και μια γλώσσα για την έκφραση πράξεων πάνω σε σχέσεις
- Υπάρχει μια ιεραρχία μοντέλων δεδομένων καθένα από τα οποία υποστηρίζει δεδομένα παρόμοια με εκείνα του σχεσιακού μοντέλου αλλά με σταδιακά πιο ισχυρές λογικές γλώσσες στις οποίες εκφράζουμε πράξεις στα δεδομένα: *το πιο απλό μοντέλο είναι η datalog.*



# Σύνταξη προγραμμάτων datalog (1/2)

- $I_0 :- I_1 \ \& \ \dots \ \& \ I_n$ 
  - $I_0$ : κεφαλή (head)
  - $I_1 \ \& \ \dots \ \& \ I_n$ : σώμα (body)
  - $I_i$ : λεκτικό (literal) της μορφής  $p_i(t_1, \dots, t_{k_i})$   
(**ατομικός τύπος**, atomic formula)
  - $p_i$ : σύμβολο κατηγορήματος (predicate symbol)
  - $t_j$ : όρος (term) (**σταθερά ή μεταβλητή**)



# Σύνταξη προγραμμάτων datalog (2/2)

- Προτάσεις datalog
  - Πλήρως αποτιμημένα γεγονότα που φυλάσσονται σε σχεσιακή βάση (άμεση βάση – extensional database, EDB)
  - Λογικοί κανόνες (έμμεση βάση – intensional database, IDB)
- Κατηγορήματα IDB – κατηγορήματα EDB
- Παράδειγμα:
  - $\text{grandparent}(Z,X) :- \text{par}(Y,X) \ \& \ \text{par}(Z,Y).$
  - $\text{father}(\text{bob},\text{john}).$



# Συντακτικοί Περιορισμοί

- Κανόνες Ασφάλειας:
  - Κάθε γεγονός πρέπει να είναι πλήρως αποτιμημένο (ground)
  - Κάθε μεταβλητή που εμφανίζεται στην κεφαλή ενός κανόνα πρέπει να εμφανίζεται και στο σώμα του
- Απαγορεύεται ένα κατηγορημα EDB να εμφανίζεται σε κεφαλή κανόνα.



# Σημασιολογία των προγραμμάτων datalog

- Μοντελοθεωρητική Σημασιολογία (model theoretic semantics)
  - Βασίζεται στις *ερμηνείες* (interpretations) και τα *μοντέλα* (models) ενός συστήματος
  - Μια ερμηνεία κάνει άλλους ατομικούς τύπους *αληθείς* και άλλους *ψευδείς* (ως προς την ερμηνεία αυτή)
  - Συνήθως, ταυτίζουμε μια ερμηνεία με το σύνολο των πλήρως αποτιμημένων τύπων που κάνει αληθείς (*ερμηνεία Herbrand*)
- Μια ερμηνεία είναι μοντέλο για ένα σύνολο από κανόνες αν κάθε κανόνας είναι αληθής ως προς την ερμηνεία αυτή
- Θεωρούμε ότι στιγμιότυπο (instance) ενός ατομικού τύπου με κατηγορήμα EDB ισχύει εάν και μόνο εάν η αντίστοιχη σχέση περιέχει αυτό το στιγμιότυπο σαν πλειάδα



# Παραδείγματα (1/6)

- $p(X) :- q(X)$ .
  - $I_1 = \{q(a)\}$ , δεν κάνει τον κανόνα αληθή
  - $I_2 = \{q(a), p(a)\}$ , κάνει τον κανόνα αληθή
- $p(X) :- q(X)$ .  
 $v(X) :- r(X)$ .
  - $I_1 = \{q(a), p(a)\}$  είναι μοντέλο για τους κανόνες
  - $I_2 = \{q(a), p(a), r(b), v(b)\}$  είναι μοντέλο για τους κανόνες
  - $I_3 = \{r(b)\}$  δεν είναι μοντέλο για τους κανόνες



# Παραδείγματα (2/6)

- $p(X) :- q(X).$

$q(X) :- r(X).$

και έστω  $r/1$  EDB τέτοιο ώστε  $r(1)$  ισχύει από τη βάση

- $I_1 = \{r(1), q(1), p(1)\}$ ,  $I_2 = \{r(1), q(1), p(1), q(2), p(2)\}$ ,  
 $I_3 = \{r(1), q(1), p(1), q(2), p(2), p(3)\}$

- $I_1, I_2, I_3$  είναι μοντέλα για το παραπάνω

- Το  $I_1$  είναι ελάχιστο μοντέλο (minimal model)

- Το  $I_1$  είναι το ελάχιστο μοντέλο Herbrand (least Herbrand model)





# Παραδείγματα (3/6)

- Ένα γεγονός  $F$  έπεται λογικά από ένα σύνολο προτάσεων  $S$  εάν και μόνο εάν κάθε ερμηνεία που κάνει αληθές το  $S$  κάνει αληθές και το  $F$ . Τότε λέμε ότι το  $F$  είναι λογικό επακόλουθο του  $S$  και γράφουμε

$$S \models F$$

- $\text{cons}(S) = \cap \{I \mid I \text{ είναι μοντέλο του } S\}$ , όπου  $\text{cons}(S) = \{F \in \text{HB} \mid S \models F\}$
- Πώς μπορεί να υπολογιστεί το  $\text{cons}(S)$ ;



# Παραδείγματα (4/6)

- Σημασιολογία Απόδειξης (proof theoretic semantics)
  - Οι κανόνες ερμηνεύονται ως αξιώματα προς χρήση σε αποδείξεις
  - Αντικαθιστούμε αποδεδειγμένα ή δεδομένα γεγονότα στο δεξί τους μέρος και συμπεραίνουμε το γεγονός που προκύπτει από την κεφαλή
  - Χρησιμοποιούνται κατά την εμπρόσθια (forward) φορά



# Παραδείγματα (5/6)

- $L_0$  :-  $L_1$  & ...  $L_n$  και
- $F_1, \dots, F_n$ : πλήρως αποτιμημένα γεγονότα και
- $\theta$ : αντικατάσταση (substitution):  $L_i\theta = F_i$
- Τότε, με ένα βήμα, συμπεραίνουμε:  $L_0\theta$
- Κανόνας συμπερασμού (inference rule)



# Παραδείγματα (6/6)

- Έστω  $S$  ένα σύνολο από προτάσεις datalog. Ένα πλήρως αποτιμημένο γεγονός  $F$  συμπεραίνεται (is inferred) από το  $S$  ( $S \vdash F$ ) εάν και μόνο εάν είτε  $F \in S$  είτε μπορεί να παραχθεί από πεπερασμένο αριθμό εφαρμογών του κανόνα συμπερασμού.
- Η ακολουθία των εφαρμογών του κανόνα προκειμένου να συμπεράνουμε ένα πλήρως αποτιμημένο γεγονός  $F$  από το  $S$  καλείται απόδειξη (proof) του  $F$  από το  $S$
- Ισχύει:  $S \vDash F \Leftrightarrow S \vdash F$
- Το σύνολο  $\text{cons}(S)$  μπορεί να προσδιοριστεί ως το ελάχιστο σταθερό σημείο ενός μετασχηματισμού



# Επεκτάσεις της datalog (I)

- **Ενσωματωμένα κατηγορήματα:** κατηγορήματα αριθμητικών συγκρίσεων ( $=, \leq, \dots$ )
- Πρόβλημα: τα ενσωματωμένα κατηγορήματα δεν αναπαριστούν απαραίτητα πεπερασμένες σχέσεις
- Κατά τον υπολογισμό πρέπει να έχουν ικανοποιητική αποτίμηση για να μην οδηγήσουν σε μη πεπερασμένες σχέσεις (ένα πρόγραμμα datalog πρέπει να έχει πεπερασμένη έξοδο)
- Λύση: Οποτεδήποτε ένας κανόνας περιέχει στο σώμα του έναν ατομικό τύπο με ενσωματωμένο κατηγορήμα, το εύρος των τιμών των μεταβλητών του πρέπει να περιορίζεται από κάποιον άλλο ατομικό τύπο στον κανόνα
- Τα ενσωματωμένα κατηγορήματα μπορούν να θεωρηθούν σαν κατηγορήματα EDB
- Με βάση τη σχεσιακή άλγεβρα μπορούν να διερμηνευτούν ως επιλογές (selections) πάνω σε μια σχέση ή συνενώσεις (joins) σχέσεων



# Γράφος εξαρτήσεων (1/4)

```
sibling(X,Y) :- parent(X,Z)
  & parent(Y,Z) & X ≠ Y.
```

```
cousin(X,Y) :- parent(X,Xp)
  & parent(Y,Yp) &
  sibling(Xp,Yp).
```

```
cousin(X,Y) :- parent(X,Xp)
  & parent(Y,Yp) &
  cousin(Xp,Yp).
```

- Κύκλοι στο γράφο δηλώνουν αναδρομή
- Ένα κατηγορήμα είναι μη αναδρομικό αν δεν εμφανίζεται μέσα σε κάποιο κύκλο
- Στα κατηγορήματα EDB δε καταλήγει κανένα βέλος



# Γράφος εξαρτήσεων (2/4)

- Υπολογισμός μη αναδρομικών κανόνων μέσω μετασχηματισμού τους σε εκφράσεις της σχεσιακής άλγεβρας
- Οι σχέσεις που προκύπτουν για τα κατηγορήματα IDB ταυτίζονται τόσο με το ελάχιστο μοντέλο των κανόνων όσο και με το σύνολο των γεγονότων IDB που συμπεραίνονται από τους κανόνες και τη βάση
- Για το μετασχηματισμό εκμεταλλευόμαστε τη διάταξη που προκύπτει από το γράφο εξαρτήσεων (έναν υπάρχει ένα βέλος  $r_i \rightarrow r_j$  για δυο κατηγορήματα  $r_i$  και  $r_j$  τότε  $r_i < r_j$ ). Οπότε, όταν πάμε να υπολογίσουμε τη σχέση που αντιστοιχεί στο σώμα ενός κανόνα, έχουμε υπολογίσει όλες τις σχέσεις που αντιστοιχούν στους διάφορους υποστόχους του κανόνα.



# Γράφος εξαρτήσεων (3/4)

- Παράδειγμα:

$$p(X,Y) :- q(a,X) \& r(X,Z,X) \& s(Y,Z)$$

Σχέσεις:

$$q \rightarrow Q, r \rightarrow R, s \rightarrow S$$

$$T(X) = \pi_2 (\sigma_{\$1=a} (Q))$$

$$U(X,Z) = \pi_{1,2}(\sigma_{\$1=\$3} (R) )$$

$$B(X,Y,Z) = T(X) \bowtie U(X,Z) \bowtie S(Y,Z)$$





# Γράφος εξαρτήσεων (4/4)

- Για τον υπολογισμό αναδρομικών κανόνων, ο γράφος εξαρτήσεων δεν μας προσφέρει καμιά διάταξη
- Οπότε, κάνουμε αλληπάλληλους υπολογισμούς

$P_i := \text{EVAL}(p_i, R_1, \dots, R_k, P_1, \dots, P_m)$ , όπου

$R_1, \dots, R_k$ : δεδομένες EDB σχέσεις και

$P_1, \dots, P_m$ : IDB σχέσεις προς υπολογισμό

με τα  $P_j$  αρχικά κενά, έως ότου να μη παράγονται άλλες πλειάδες, τότε

$$P_i = \text{EVAL}(p_i, R_1, \dots, R_k, P_1, \dots, P_m)$$

και πλέον έχουμε τη λύση για τις σχέσεις που αντιστοιχούν στα κατηγορήματα IDB αυτών των εξισώσεων (σταθερό σημείο – fixpoint)



# Ιδιότητες των προγραμμάτων datalog

- Έχουν ένα μοναδικό ελάχιστο σταθερό σημείο
- Έχουν ένα μοναδικό ελάχιστο μοντέλο
- Τα παραπάνω ταυτίζονται με το σύνολο των γεγονότων που μπορούμε να παράγουμε χρησιμοποιώντας τους κανόνες για μια δεδομένη βάση
- Η παραπάνω διαδικασία είναι μονότονη. Αρκεί σε κάθε βήμα να λαμβάνουμε υπόψη μας όχι ολόκληρες τις σχέσεις αλλά τις επαυξήσεις (διαφορές) που προέκυψαν από το προηγούμενο.



# Επεκτάσεις της datalog (II) (1/8)

- **Άρνηση** (negation):
- Κανόνες με άρνηση στα σώματά τους δεν είναι προτάσεις Horn (χάνουμε πολλά από τα ωραία συμπεράσματα που ισχύουν για τις προτάσεις Horn)
- $\text{trueCousin}(X,Y) :- \text{cousin}(X,Y) \ \& \ \neg \text{sibling}(X,Y)$
- Διαισθητικά μπορούμε να πούμε ότι η άρνηση δίνει το «συμπλήρωμα» μιας σχέσης



# Επεκτάσεις της datalog (II) (2/8)

- Προβλήματα:
  - «Συμπλήρωμα» ως προς ποιο πεδίο ορισμού;
  - Το συμπλήρωμα μπορεί να οδηγεί σε μη πεπερασμένη σχέση
  - Δεν υπάρχει απαραίτητα ένα ελάχιστο σταθερό σημείο για το λογικό πρόγραμμα
  - Υπάρχουν πολλά ελάχιστα μοντέλα για το λογικό πρόγραμμα
- $C(X,Y) \bowtie \text{compl } S(X,Y)$ , όπου  $\text{compl } S$  είναι το συμπλήρωμα του  $S$  σε σχέση με κάποιο σύμπαν  $U$  το οποίο περιέχει τουλάχιστον τις πλειάδες του  $C$



# Επεκτάσεις της datalog (II) (3/8)

- Πρόβλημα: μεταβλητές που εμφανίζονται ΜΟΝΟ σε υποστόχους με άρνηση
- $\text{bachelor}(X) :- \text{male}(X) \ \& \ \neg\text{married}(X,Y)$
- $\text{MRD} = \{ \langle 1,a \rangle, \langle 2,b \rangle \}$
- $M = \{1,2,3,4\}$
- $\text{compl MRD} = \{ \langle 1,b \rangle, \langle 2,a \rangle, \langle 3,a \rangle, \dots \}$ : δεν είναι το επιθυμητό
- Λύση: απαγορεύεται η χρήση μιας μεταβλητής σ' έναν υποστόχο με άρνηση, εάν αυτή δεν εμφανίζεται σε κάποιον άλλο υποστόχο ο οποίος δεν πρέπει να περιέχει άρνηση ή άλλο ενσωματωμένο κατηγορημα
- $\text{husband}(X) :- \text{married}(X,Y).$
- $\text{bachelor}(X) :- \text{male}(X) \ \& \ \neg\text{husband}(X).$
- $H(X) = \pi_x(\text{MRD}(X,Y)), B(X) = M(X) - H(X)$



# Επεκτάσεις της datalog (II) (4/8)

- Πρόβλημα: δεν υπάρχει ένα μοναδικό ελάχιστο σταθερό σημείο
- $p(X) :- r(X) \ \& \ \neg q(X).$   
 $q(X) :- r(X) \ \& \ \neg p(X).$

$R\{1\}$

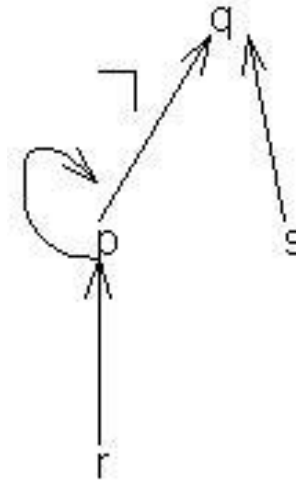
S1:  $P = \{1\}, Q = \text{empty\_set}$

S2:  $P = \text{empty\_set}, Q = \{1\}$



# Επεκτάσεις της datalog (II) (5/8)

- Στρωματοποιημένη άρνηση (stratified negation)
- $p(X) :- r(X).$
- $p(X) :- p(X).$
- $q(X) :- s(X) \ \& \ \neg p(X).$
  
- Δεν υπάρχει μονοπάτι από το  $q$  στο  $p$ : στρωματοποιημένο
- $R\{1\}, S\{1,2\}$
- $S1: P = \{1\}, Q = \{2\}$
- $S2: P = \{1,2\}, Q = \text{empty\_set}$
- Και τα δυο είναι ελάχιστα, το  $S1$ , όμως, δείχνει πιο «φυσικό»



# Επεκτάσεις της datalog (II) (6/8)

- **Στρωματοποίηση:** Καθορισμός στρωμάτων (strata)
  - Εάν ένα κατηγορημα  $p$  έχει έναν κανόνα που περιέχει υποστόχο με άρνηση με κατηγορημα  $q$ , τότε το  $q$  είναι σε χαμηλότερο στρώμα από το  $p$ .
  - Εάν ένα κατηγορημα  $p$  έχει έναν κανόνα που περιέχει υποστόχο χωρίς άρνηση που περιέχει το  $q$  τότε το στρώμα του  $p$  είναι το στρώμα του  $q$  ή ψηλότερο.
- Τα στρώματα μας δίνουν μια σειρά με την οποία μπορούν να υπολογιστούν οι σχέσεις που αντιστοιχούν στα κατηγορήματα IDB
- Επεξεργαζόμαστε κάθε φορά ένα στρώμα ξεκινώντας από το χαμηλότερο
- Έτσι, μεταχειριζόμαστε τους υποστόχους με άρνηση σαν να ήταν σχέσεις EDB





# Επεκτάσεις της datalog (II) (7/8)

- Η σχέση για έναν υποστόχο με άρνηση  
 $\neg q(X_1, \dots, X_n)$

είναι:

$$\underbrace{\text{DOM} \times \text{DOM} \times \dots \times \text{DOM}}_{n \text{ φορές}} - Q,$$

όπου DOM είναι η ένωση όλων των σταθερών που εμφανίζονται στην IDB και στην EDB



# Επεκτάσεις της datalog (II) (8/8)

- Κατασκευή στρωμάτων
- Κατασκευή DOM
- Υπολογισμός σχέσεων ανά στρώμα, ξεκινώντας από το χαμηλότερο
  1. Για τους υποστόχους που δεν περιέχουν άρνηση και τα κατηγορήματά τους περιέχονται σε χαμηλότερα στρώματα, θεωρούμε τις σχέσεις όπως έχουν υπολογιστεί
  2. Για τους υποστόχους που περιέχουν άρνηση, θεωρούμε τα συμπληρώματα των σχέσεων – οι σχέσεις έχουν ήδη υπολογιστεί
  3. Εφαρμόζουμε τον αλγόριθμο κατασκευής των σχέσεων μέσα στο στρώμα, θεωρώντας τις σχέσεις που αναφέρονται στα δυο πρώτα βήματα σαν να ήταν EDB
- Ο αλγόριθμος αυτός παράγει ένα ελάχιστο σταθερό σημείο (που είναι και ελάχιστο μοντέλο: το τέλειο – perfect – )



# Άλλα θέματα που σχετίζονται με τη datalog

- Βελτιστοποίηση υπολογισμού ερωτήσεων datalog
- Επεκτάσεις της datalog (III): σύνθετα αντικείμενα (complex objects)
- Επεκτάσεις της datalog (IV): αντικειμενοστραφείς επεκτάσεις (object oriented extensions)



# Σύγχρονες ανάγκες για αναπαράσταση γνώσης

- World wide web:
  - Πραγματικότητα: “Web of documents”
- Μεταδεδομένα
- Σημασιολογική πληροφορία
- Semantic web
  - Vision: “Web of data”



# World Wide Web (1/2)

- Web of documents
- URLs
- HTTP
- HTML σελίδες (HTML εμπνευσμένη από την SGML – Standard Generalized Markup Language)



# World Wide Web (2/2)

- Παράδειγμα:  
<H1> Λογικός Προγραμματισμός </H1>  
<H2> Εξάμηνο ΣΤ' </H2>  
<H3> Περιεχόμενο </H3>  
<UL>  
<LI> Prolog  
<LI> Θεωρία Λογικού Προγραμματισμού  
<LI> ...  
....  
</UL>



# XML και Ημιδομημένα Δεδομένα (1/2)

- Tags που ορίζονται από το χρήστη σύμφωνα με τις ανάγκες μιας εφαρμογής
- Θεωρούμε ότι απεικονίζουν κάποιας μορφής «νόημα» για το περιεχόμενό τους, μια που θα το χειριστεί κατάλληλα η αντίστοιχη εφαρμογή
- XML (Extensible Markup Language )
- W3 activity
- Recommendation (5η έκδοση, Feb 2008)
- Βασίζεται στην SGML (υποσύνολό της)
- Ο ορισμός του συντακτικού ενός εγγράφου XML μπορεί να γίνει από κάποιο φορμαλισμό όπως το DTD (Document Type Definition) ή το XML Schema
- Γλώσσες και μεθοδολογίες για τον υπολογισμό απαντήσεων για έγγραφα XML, π.χ. XQuery και XPath, αντίστοιχα



# XML και Ημιδομημένα Δεδομένα (2/2)

- Παράδειγμα:

```
<book>
```

```
<title> Gone with the wind </title>
```

```
<author> Margaret Mitchell </author>
```

```
<year> 1936 </year>
```

```
</book>
```





# Ευελιξία στο σχήμα των δεδομένων – RDF (1/2)

- RDF (Resource Description Framework)
- Αρχικά W3 activity, μετά semantic web activity
- Specification recommendation (2004)
- Μοντέλο δεδομένων
- Γενική μορφή: «τριπλέτες»  $(x, P, y)$ , όπου:
  - P: property
  - $x, y$ : objects
  - Αντίστοιχος λογικός τύπος:  $P(x, y)$  – P binary predicate



# Ευελιξία στο σχήμα των δεδομένων – RDF (2/2)

- URIs (Uniform Resource Identifiers) μπορούν να χρησιμοποιηθούν για να ονομάσουν τόσο τα  $P$  και  $x$  αλλά και τα  $y$  (αλλιώς το  $y$  μπορεί να είναι μια σταθερά)
- Γλώσσες και μεθοδολογίες για τον υπολογισμό απαντήσεων για αρχεία RDF, π.χ. SPARQL
- RDF και ER
- RDF και Conceptual Graphs



# Ορισμός σχήματος – «ελαφρά» μεταδεδομένα

- RDF Schema (RDF Vocabulary Description Language)
- W3 activity
- Specification recommendation (2004)
- Classes-Instances-Properties/Resources
- Subclass/belongs/subproperty
- Κληρονομικότητα



# Ανταλλαγή πληροφορίας – περιγραφή γνωστικού πεδίου (1/2)

- Μοντέλο δεδομένων και μεταδεδομένων
- Για περιγραφή πολύπλοκων γνωστικών πεδίων, π.χ.
  - δυνατότητα περιγραφής κλάσεων, όχι απλά δυνατότητα ονομασίας
  - περιορισμούς στα πεδία τιμών
  - απόδοση ιδιοτήτων μεταξύ κλάσεων (π.χ. «ξένες» μεταξύ τους)



# Ανταλλαγή πληροφορίας – περιγραφή γνωστικού πεδίου (2/2)

- OWL (Web Ontology Language)
- W3 activity
- Specification recommendation (2004)
- Συμβιβασμός μεταξύ εκφραστικότητας και υλοποιησιμότητας
- OWL Full (first order logic) (υπερσύνολο της RDFS)
- OWL-DL (description logic)
- OWL-Lite (απλή λειτουργικότητα στον ορισμό κλάσεων)



# Λογικές Περιγραφών (1/2)

- Βασικές έννοιες:
  - Concepts (unary predicates), π.χ. άνθρωπος
  - Roles (binary predicates), π.χ. hasChild
  - Individual names (σταθερές), π.χ. Mary
  - Τελεστές για να περιγράψουμε concepts και roles
  - Προσοχή: πρέπει να είναι περιορισμένοι ώστε
- Satisfiability/subsumption is decidable and, *if possible*, of low complexity



# Λογικές Περιγραφών (2/2)

– TBox: Το σύνολο των περιγραφών, π.χ.

$\text{Doctor} \subseteq \text{Person},$

$\text{HappyParent} = \text{Person} \cap \forall \text{ hasChild}. (\text{Doctor} \cup \exists \text{ hasChild}. \text{Doctor})$

– ABox: Το σύνολο των δεδομένων. π.χ.

$\text{John}:\text{HappyParent},$

$\text{John hasChild Mary}$

–  $\text{KB} = \text{TBox} \cup \text{Abox}$

• Σημασιολογία βασισμένη στις ερμηνείες και τα μοντέλα



# Συλλογιστική

- SWRL (Semantic Web Rule Language) (W3 member submission)
- RuleML:
  - Κανόνες για το WWW
  - RuleML consortium collaborates with W3C
- Κανόνες πάνω από την OWL VS Horn (datalog) κανόνες (Λογικός Προγραμματισμός)
- “open” semantics VS “closed” semantics για την άρνηση





# Λογικός Προγραμματισμός

- Μπορεί να χρησιμοποιηθεί για:
  - Μοντέλο δεδομένων,
  - περιγραφή μεταδεδομένων και
  - κανόνων για χρήση σε συλλογιστική
- Απλότητα
- Θεωρία
- Υλοποίηση (prolog)
- Επεκτάσεις (constraints, modules)



# Βιβλιογραφία

- Ivan Bratko, “Prolog Programming for Artificial Intelligence”, Addison Wesley, 2000
- Jeffrey D. Ullman, “Principles of database and knowledge-base systems, Vol. I”, Computer Science Press, Inc , 1988
- W3 Consortium (<http://www.w3.org/>)
- Ivan Herman, “Introduction to the Semantic Web” WWW2006, Edinburgh, UK, 2006-05-24 (<http://www.w3.org/2006/Talks/0524-Edinburgh-IH>)
- Ian Horrocks, "Description Logics in Ontology Applications“, KI/Tableaux 2005 (<http://www.cs.man.ac.uk/~horrocks/Slides/index.html>)
- Carsten Lutz and Ulrike Sattler, “Description Logics Course” ESSLLI 2002 (<http://lat.inf.tu-dresden.de/~clu/esslli.html>)



Τέλος Ενότητας

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

# Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0



# Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Παναγιώτης Σταματόπουλος, Ιζαμπώ Καράλη. «Λογικός Προγραμματισμός, Λογικός προγραμματισμός για αναπαράσταση γνώσης». Έκδοση: 1.0. Αθήνα 2015.  
Διαθέσιμο από τη δικτυακή διεύθυνση:  
<http://opencourses.uoa.gr/courses/DI117/>.



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



# Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

**Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες**

