



Αριστοτέλειο  
Πανεπιστήμιο  
Θεσσαλονίκης

# Τεχνητή Νοημοσύνη

Αναπαράσταση Γνώσης - Λογική - Συστήματα Κανόνων

Ιώαννης Βλαχάβας

Τμήμα Πληροφορικής ΑΠΘ



## Άδειες Χρήσης

Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons. Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



## Χρηματοδότηση

Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα. Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης**» έχει χρηματοδοτήσει μόνο τη 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.



# Αναπαράσταση Γνώσης και Συλλογιστικές

# Αναπαράσταση Γνώσης και Συλλογιστικές (Knowledge Representation - KR & Reasoning)

Τεχνητή Νοημοσύνη = Περιγραφή Προβλήματος + **Αναπαράσταση Γνώσης** + Αναζήτηση

- ❖ Η περιγραφή των προβλημάτων περιέχει γνώση
  - ❑ Περιγραφή αντικειμένων, ιδιοτήτων, σχέσεων και τιμών.
  - ❑ Τελεστές μετάβασης.
- ❖ Χρειάζονται τυποποιημένες μέθοδοι:
  - ❑ Για την περιγραφή της γνώσης (αποθήκευση σε H/Y).
  - ❑ Διευκόλυνση επίλυσης προβλημάτων ή εξαγωγής συμπερασμάτων.

# Μέθοδοι Αναπαράσταση Γνώσης

- ❖ Διαδικαστική αναπαράσταση (*procedural attachment*).
- ❖ Λογική
  - ❑ Προτασιακή λογική (*propositional logic*)
  - ❑ Κατηγορηματική λογική (*predicate logic*)
  - ❑ Μη-μονότονη λογική (*non-monotonic logic*)
- ❖ Δομημένες αναπαραστάσεις γνώσης
  - ❑ Σημασιολογικά Δίκτυα (*semantic networks*)
  - ❑ Πλαίσια (*frames*)
  - ❑ Αντικείμενα (*objects*)
  - ❑ Εννοιολογική εξάρτηση (*conceptual dependency*)
  - ❑ Σενάρια (*scripts*)
  - ❑ Εννοιολογικοί γράφοι (*conceptual graphs*)
- ❖ Κανόνες (*if-then rules*).

# **Βασικές Αρχές Αναπαράστασης Γνώσης και Συλλογιστικής**

# Αναπαράσταση Γνώσης

**Αναπαράσταση γνώσης** είναι ένα σύνολο συντακτικών και σημασιολογικών παραδοχών, οι οποίες καθιστούν δυνατή την περιγραφή ενός κόσμου.

- ❖ Η φυσική γλώσσα είναι ακατάλληλη για αναπαράσταση γνώσης λόγω της πολυσημαντικότητας (ambiguity) και της ερμηνείας με βάση τα συμφραζόμενα (context).
  - ❑ Υπάρχει ειδικός τομέας της ΤΝ που ασχολείται με την επεξεργασία φυσικής γλώσσας (natural language processing, NLP).
- ❖ **Για τα συστήματα ΤΝ** πρέπει να χρησιμοποιηθεί ένας μονοσήμαντος και τυποποιημένος συμβολισμός ο οποίος,
  - ❑ εκτός της δυνατότητας για ακριβή αναπαράσταση της γνώσης,
  - ❑ θα πρέπει να μπορεί να συνδυαστεί κατάλληλα με μηχανισμούς εξαγωγής συμπερασμάτων (**inference mechanisms**).
    - ✓ Μηχανισμός που χρησιμοποιείται για εξαγωγή συμπερασμάτων από υπάρχουσα γνώση.
- ❖ Μία μέθοδος αναπαράστασης γνώσης έχει:
  - ❑ **Συντακτικό (syntax)**: ο ορισμός των συμβόλων που χρησιμοποιεί και οι κανόνες με τους οποίους τα σύμβολα αυτά μπορούν να συνδυαστούν, και
  - ❑ **Σημασιολογία (semantics)**: ο καθορισμός των εννοιών που αποδίδονται στα σύμβολα και συνδυασμούς συμβόλων που επιτρέπει το συντακτικό.

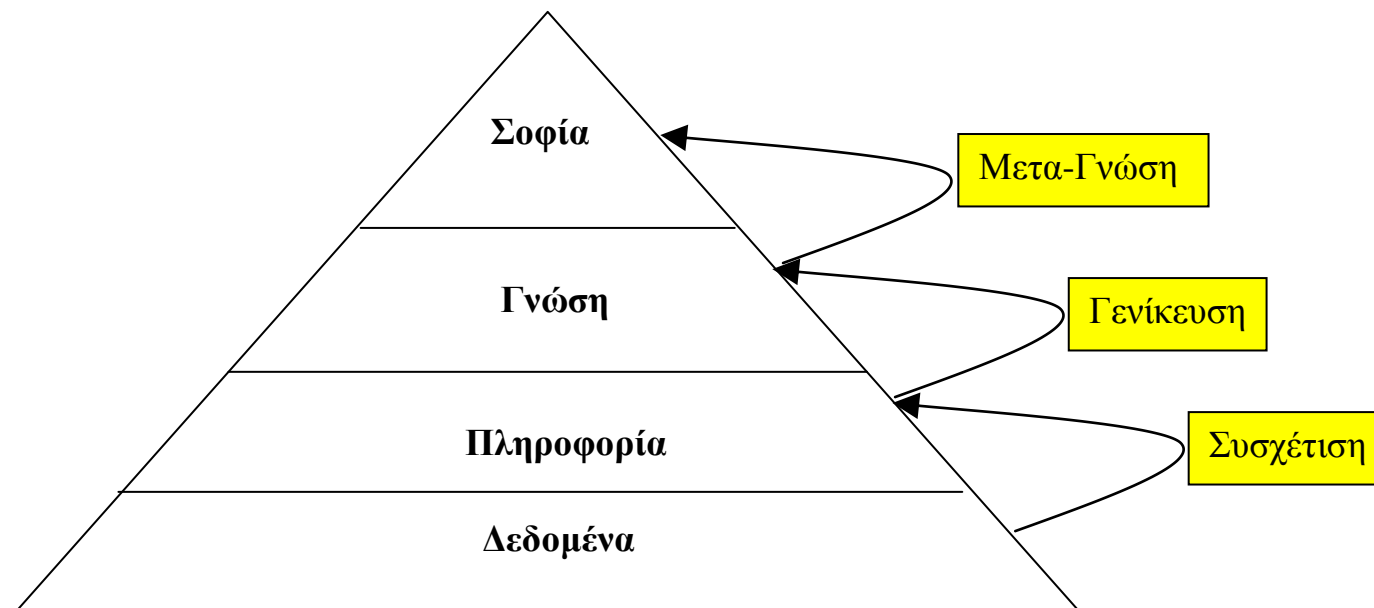
# Δεδομένα, Πληροφορία, Γνώση και Σοφία

**Δεδομένο** (*data*) είναι μια μετρήσιμη ή υπολογίσιμη τιμή μίας ιδιότητας.

**Πληροφορία** (*information*) αποτελείται από δεδομένα τα οποία όμως έχουν φιλτραριστεί και μορφοποιηθεί κατάλληλα.

**Γνώση** (*knowledge*) είναι πληροφορία η οποία έχει υποστεί μία σειρά ειδικών ελέγχων για την πιστοποίησή της.

**Σοφία** (*wisdom*) είναι η ικανότητα να χρησιμοποιεί κάποιος τη γνώση όσο το δυνατόν αποδοτικότερα (αναθεώρηση, μάθηση, διορατικότητα, πρόβλεψη).





# Είδη Γνώσης

- ❖ **Αντικείμενα (objects):** Αναπαράσταση των αντικειμένων ενός κόσμου καθώς και της σχέσης μεταξύ των.
  - ❑ Σημασιολογική γνώση (semantic knowledge), ιεραρχικά δομημένη.
- ❖ **Γεγονότα (events):** Αναπαράσταση των ενεργειών και της χρονικής ακολουθίας με την οποία συμβαίνουν, καθώς και τις σχέσεις αίτιου-αποτελέσματος.
  - ❑ Επεισοδιακή γνώση (episodical knowledge).
  - ❑ Προσωπικές εμπειρίες ενός ατόμου, οργανωμένες χρονικά και χωρικά σε επεισόδια και όχι σε έννοιες ή σχέσεις.
- ❖ **Εκτέλεση (performance):** Αναπαράσταση των δεξιοτήτων για το πώς κάποιος κάνει πράγματα (εκτελεί εργασία ή διεκπεραιώνει διαδικασία)
  - ❑ Διαδικαστική γνώση (procedural knowledge).
- ❖ **Μετα-γνώση (meta-knowledge):** Αναπαράσταση της γνώσης για το τι γνωρίζει κάποιος και πότε πρέπει να το εφαρμόσει.
  - ❑ Συνώνυμη της σοφίας.

# Κριτήρια Αξιολόγησης Μεθόδων Αναπαράστασης Γνώσης

- ❖ Επάρκεια αναπαράστασης (representational adequacy).
  - ❑ Η ικανότητα να αναπαριστά όλα τα είδη της γνώσης.
- ❖ Επάρκεια συνεπαγωγής (inferential adequacy).
  - ❑ Η ικανότητα να συνεργάζεται με μηχανισμούς που επεξεργάζονται υπάρχουσες δομές γνώσης με σκοπό τη δημιουργία νέων δομών (δηλ. νέας γνώσης) ή συμπερασμάτων.
- ❖ Αποδοτικότητα συνεπαγωγής (inferential efficiency).
  - ❑ Η ικανότητα να μπορεί να εισάγει επιπλέον πληροφορίες στις δομές γνώσης, έτσι ώστε να κατευθύνει τους μηχανισμούς εξαγωγής συμπερασμάτων γρηγορότερα προς τη λύση.
- ❖ Αποδοτικότητα απόκτησης (acquisitional efficiency).
  - ❑ Η ικανότητα να επιτρέπει την απόκτηση νέας γνώσης εύκολα και γρήγορα.

# Διαδικαστική Αναπαράσταση

- ❖ Η γνώση αναπαρίσταται μέσω εξειδικευμένων διαδικασιών, οι οποίες κωδικοποιούνται κατάλληλα σε κάποια γλώσσα προγραμματισμού.
  - ❑ Οι διαδικασίες αυτές αποσκοπούν στην επίλυση συγκεκριμένων υποπροβλημάτων. Ενεργοποιούνται είτε από άλλες διαδικασίες ή από δεδομένα.
  - ❑ Υπάρχει μια ειδική κατηγορία διαδικασιών που ονομάζονται *δαίμονες* (demons), οι οποίες ενεργοποιούνται μόλις ικανοποιηθεί μια συνθήκη.
- ❖ **Χαρακτηριστικό** της είναι η ανάμιξη της γνώσης ελέγχου για την επίλυση του προβλήματος με την ίδια την γνώση για το πρόβλημα, το οποίο προκαλεί.
  - ❑ Γρηγορότερη επεξεργασία της κωδικοποιημένης γνώσης σε σύγκριση με τις δηλωτικές αναπαραστάσεις (το σημαντικότερο πλεονέκτημα της μεθόδου).
  - ❑ Αυξημένη δυσκολία αναθεώρησης ή προσθήκης γνώσης λόγω της αλληλεπίδρασης μεταξύ των διαδικασιών (μειονέκτημα)
  - ❑ Μικρότερη διαφάνεια του προγράμματος, σε σύγκριση με τις δηλωτικές μεθόδους αναπαράστασης (μειονέκτημα).
- ❖ Ο αριθμός συστημάτων τα οποία χρησιμοποιούν αυτή την προσέγγιση σαν μοναδικό τρόπο αναπαράστασης γνώσης είναι περιορισμένος.
  - ❑ Συνήθως για την αντιμετώπιση των περισσότερων προβλημάτων απαιτείται συνδυασμός διαδικαστικών και δηλωτικών μεθόδων.

# Συλλογιστική (reasoning)

- ❖ Ένα υπολογιστικό σύστημα επιδεικνύει νοημοσύνη όταν δεδομένης της γνώσης για ένα πρόβλημα, μπορεί να κάνει συνεπαγωγές, δηλαδή να βγάλει συμπεράσματα.
- ❖ **Συλλογιστική:** Η μέθοδος με την οποία τμήματα υπάρχουσας γνώσης συνδυάζονται μεταξύ τους ώστε να παράγουν νέα γνώση ή να εξάγουν συμπεράσματα.
- ❖ Πιο γνωστές συλλογιστικές:
  - ❑ Συνεπαγωγή (*deduction*)
  - ❑ Επαγωγή (*induction*)
  - ❑ Απαγωγή (*abduction*)
- ❖ Εξελιγμένες συλλογιστικές:
  - ❑ Συλλογιστική των Μοντέλων (*model-based reasoning*)
  - ❑ Ποιοτική Συλλογιστική (*qualitative reasoning*)
  - ❑ Συλλογιστική των Περιπτώσεων (*case-based reasoning*)
    - Συλλογιστική με αναλογίες (*analogical reasoning*)

# Είδη Συλλογιστικής

**Η συνεπαγωγική συλλογιστική (*deductive reasoning*) εξάγει συμπεράσματα βασισμένη στους κλασικούς μηχανισμούς εξαγωγής συμπερασμάτων της λογικής.**

Δεδομένου του κανόνα: Όλα τα σκυλιά του Κώστα είναι καφέ  
και του γεγονότος: Αυτά τα σκυλιά είναι του Κώστα  
Συμπέρασμα που εξάγεται: Αυτά τα σκυλιά είναι καφέ

**Η επαγωγική συλλογιστική (*inductive reasoning*) αφορά την εξαγωγή γενικών συμπερασμάτων από ένα σύνολο παραδειγμάτων.**

Δεδομένων των γεγονότων: Το σκυλί Α είναι του Κώστα και είναι καφέ.  
Το σκυλί Β είναι του Κώστα και είναι καφέ.  
...  
Κανόνας που εξάγεται: Όλα τα σκυλιά του Κώστα είναι καφέ.

**Η απαγωγική συλλογιστική (*abductive reasoning*) αφορά την εξαγωγή συμπερασμάτων κατά την οποία, με δεδομένα μία βάση γνώσης και μερικές παρατηρήσεις (*observations*) επιχειρείται η εύρεση υποθέσεων οι οποίες μαζί με τη βάση γνώσης εξηγούν τις παρατηρήσεις.**

Δεδομένου του κανόνα: Όλα τα σκυλιά του Κώστα είναι καφέ  
και του αποτελέσματος: Τα σκυλιά είναι καφέ  
Υπόθεση που γίνεται: Αυτά τα σκυλιά είναι του Κώστα

# Μηχανισμός Εξαγωγής Συμπερασμάτων

- ❖ Οι **συλλογιστικές** υλοποιούνται από έναν ή περισσότερους εναλλακτικούς μηχανισμούς εξαγωγής συμπερασμάτων (*inference mechanisms*)

**Μηχανισμός Εξαγωγής Συμπερασμάτων:** Αλγόριθμος που συνδυάζει τις δομές που αναπαριστούν τα διάφορα τμήματα της γνώσης που βρίσκονται αποθηκευμένα στη μνήμη του υπολογιστή, σύμφωνα με το αφαιρετικό μοντέλο της συλλογιστικής, και παράγει νέες δομές (γνώση) που επίσης αποθηκεύει στον υπολογιστή.

## Στρατηγική Αναζήτησης

- ❖ Η εξαγωγή συμπερασμάτων (*inference*) εξαρτάται όχι μόνο από τη συλλογιστική, αλλά και από τη στρατηγική αναζήτησης στη γνώση ενός προβλήματος.
- ❖ **Στρατηγική αναζήτησης** είναι ο τρόπος με τον οποίο έχει δομηθεί και κωδικοποιηθεί η γνώση προκειμένου να δώσει λύση σε ένα πρόβλημα.
- ❖ Υλοποιείται με διάφορους τρόπους:
  - ❑ **Οδηγούμενη από τους στόχους** (goal driven ή top-down): Ξεκινάμε από πιθανά συμπεράσματα και φτάνουμε στις αιτίες που τα στηρίζουν.
  - ❑ **Οδηγούμενη από τα δεδομένα** (data driven ή bottom-up): Ξεκινάμε από τα δεδομένα του προβλήματος και φτάνουμε σε συμπεράσματα.
- ❖ Μηχανισμός Εξαγωγής συμπερασμάτων = Συλλογιστική + Στρατηγική Αναζήτησης

# Ασκήσεις

- ❖ 7.1. Δώστε ένα παράδειγμα για δεδομένα, πληροφορίες, γνώση, σοφία, για το ίδιο πεδίο γνώσης:
  - ❑ Πεδίο γνώσης: Θερμοκρασία περιβάλλοντος
- ❖ 7.2. Δώστε παραδείγματα των διαφόρων ειδών γνώσης για το ίδιο πεδίο γνώσης:
  - ❑ Πεδίο γνώσης: Θέρμανση κλειστών χώρων
- ❖ 7.3. Δώστε παραδείγματα διαφόρων ειδών συλλογιστικής για το ίδιο πεδίο γνώσης:
  - ❑ Πεδίο γνώσης: Θέρμανση κλειστών χώρων



# Λογική

Περίληψη



# Λογική

- ❖ Η λογική παρέχει έναν τρόπο για την αποσαφήνιση και την τυποποίηση της διαδικασίας της ανθρώπινης σκέψης
- ❖ Προσφέρει μια σημαντική και εύχρηστη μεθοδολογία για την αναπαράσταση και επίλυση προβλημάτων.
  - ❑ Σαφής, ακριβής και απλή στη σύνταξη γλώσσα
  - ❑ Δυνατότητα παραγωγής νέας γνώσης από την ήδη υπάρχουσα.
- ❖ Είδη Λογικής:
  - ❑ *Προτασιακή λογική (propositional logic)*
  - ❑ *Κατηγορηματική λογική (predicate logic)*
  - ❑ *Μη-μονότονη λογική (non-monotonic logic)*
    - Αναιρέσιμη Λογική (defeasible logic)

# Προτασιακή Λογική

- ❖ Η απλούστερη μορφή μαθηματικής λογικής.
- ❖ Κάθε γεγονός του πραγματικού κόσμου αναπαριστάται με μια λογική πρόταση, η οποία χαρακτηρίζεται είτε ως *αληθής* (*t-true*) ή ως *ψευδής* (*f-false*).
- ❖ Οι λογικές προτάσεις αναπαριστώνται συνήθως από λατινικούς χαρακτήρες *P*, *Q*, *R*, κτλ., και ονομάζονται *άτομα* (*atoms*).
- ❖ Τα άτομα μπορούν να συνδυαστούν με τη χρήση *λογικών συμβόλων* ή *συνδετικών* (*connectives*) (π.χ. AND ( $\wedge$ ), OR ( $\vee$ ), IF-THEN ( $\rightarrow$ ) κλπ.)
- ❖ Παράδειγμα:
  - ❑ *P*: "επιδιώκω την ειρήνη"
  - ❑ *Q*: "αποφεύγω τον πόλεμο"
  - ❑  $P \rightarrow Q$  "εάν επιδιώκω την ειρήνη, τότε αποφεύγω τον πόλεμο"
- ❖ Πλεονεκτήματα:
  - ❑ Απλότητα στη σύνταξη.
  - ❑ Μπορεί να καταλήξει πάντα σε συμπέρασμα.
- ❖ Μειονεκτήματα:
  - ❑ Έλλειψη γενικότητας που οδηγεί σε ογκώδεις αναπαραστάσεις γνώσης.
  - ❑ Κάθε γεγονός πρέπει να αναπαριστάται με μια χωριστή λογική πρόταση.

# Κατηγορηματική Λογική

- ❖ Η κατηγορηματική λογική (*predicate logic*) επεκτείνει την προτασιακή λογική εισάγοντας όρους (*terms*), κατηγορήματα (*predicates*) και ποσοδείκτες (*quantifiers*).
- ❖ Ο κόσμος περιγράφεται σαν ένα σύνολο αντικειμένων, ιδιοτήτων και σχέσεων
  - ❑ Αναπαραστάσεις που είναι περισσότερο κοντά στην ανθρώπινη εμπειρία.
- ❖ Η κατηγορηματική λογική αντιμετωπίζει το πρόβλημα της μη προσπελασιμότητας των στοιχείων των γεγονότων της προτασιακής λογικής.
  - ❑ Π.χ. η πρόταση " ο τζίμης είναι τίγρης " αναπαριστάται με *τίγρης(τζίμης)*, επιτρέποντας την προσπέλαση των στοιχείων του συγκεκριμένου αντικειμένου (*τζίμης*) από τους κανόνες εξαγωγής συμπερασμάτων για τη δημιουργία νέων προτάσεων.
- ❖ Οι μεταβλητές επιτρέπουν την αναπαράσταση "γενικής" γνώσης
  - ❑ Π.χ. "όλοι οι άνθρωποι είναι θνητοί".
- ❖ Πλεονεκτήματα:
  - ❑ Αντιστοιχία με τη φυσική γλώσσα, ικανοποιητική έκφραση ποσοτικοποίησης των εννοιών με τους κατάλληλους ποσοδείκτες, ικανότητα να συλλάβει τη γενικότητα.
- ❖ Μειονεκτήματα
  - ❑ *Αδυναμία έκφρασης ασάφειας*: Κάθε πρόταση μπορεί να είναι μόνο αληθής ή ψευδής.
  - ❑ *Αθροιστικότητα των αποτελεσμάτων*: Ένα συμπέρασμα προστίθεται στη γνώση χωρίς να δίνεται η δυνατότητα αναθεώρησής του αν αργότερα κριθεί ότι είναι εσφαλμένο.

# Παράδειγμα

*Κάθε ζώο το οποίο έχει τρίχωμα ή παράγει γάλα είναι θηλαστικό.*

*Κάθε ζώο που έχει φτερά και γεννάει αυγά είναι πουλί.*

*Κάθε θηλαστικό που τρέφεται με κρέας ή έχει κοφτερά δόντια είναι σαρκοβόρο.*

*Κάθε σαρκοβόρο με χρώμα καφέ-πορτοκαλί που έχει μαύρες ρίγες είναι τίγρης.*

*Κάθε σαρκοβόρο με χρώμα καφέ-πορτοκαλί και μαύρες βούλες είναι τσιτάχ.*

*Κάθε πουλί το οποίο δεν πετά και κολυμπά είναι πιγκουΐνος.*

$\forall X (\text{έχει}(X, \text{τρίχωμα}) \vee \text{παράγει}(X, \text{γάλα})) \rightarrow \text{είναι}(X, \text{θηλαστικό})$

$\forall X (\text{έχει}(X, \text{φτερά}) \wedge \text{γεννάει}(X, \text{αυγά})) \rightarrow \text{είναι}(X, \text{πουλί})$

$\forall X (\text{είδος}(X, \text{θηλαστικό}) \wedge ((\text{τρέφεται}(X, \text{κρέας}) \vee \text{έχει}(X, \text{δόντια}(\text{κοφτερά}))))$   
 $\rightarrow \text{είναι}(X, \text{σαρκοβόρο})$

$\forall X (\text{είναι}(X, \text{σαρκοβόρο}) \wedge \text{χρώμα}(X, \text{καφέ-πορτοκαλί}) \wedge \text{έχει}(X, \text{ρίγες}(\text{μαύρες})))$   
 $\rightarrow \text{είναι}(X, \text{τίγρης})$

$\forall X (\text{είναι}(X, \text{σαρκοβόρο}) \wedge \text{χρώμα}(X, \text{καφέ-πορτοκαλί}) \wedge \text{έχει}(X, \text{βούλες}(\text{μαύρες})))$   
 $\rightarrow \text{είναι}(X, \text{τσιτάχ})$

$\forall X (\text{είναι}(X, \text{πουλί}) \wedge (\neg \text{πετά}(X)) \wedge \text{κολυμπά}(X)) \rightarrow \text{είναι}(X, \text{πιγκουΐνος})$

# Συζευκτική Μορφή της Λογικής

- ❖ Είναι μια ευρέως διαδεδομένη μορφή της κατηγορηματικής λογικής που προκύπτει με απαλοιφή των υπαρξιακών και καθολικών ποσοδεικτών και των συμβόλων απλής και διπλής συνεπαγωγής.
- ❖ Οι προτάσεις (clauses) αυτής της λογικής είναι της μορφής:
  - ❑  $R \leftarrow Q_1, Q_2, \dots, Q_n$  (ισχύει  $R$  εάν  $Q_1$  και  $Q_2 \dots$  και  $Q_n$ ),
  - ❑  $\leftarrow Q_1, Q_2, \dots, Q_n$  (δηλαδή Δεν ισχύει  $Q_1$  και  $Q_2$  και  $\dots Q_n$ )
  - ❑  $R \leftarrow$  (δηλαδή το  $R$  ισχύει πάντα)
  - ❑ Κενή πρόταση (δηλώνει πρόταση πάντα αναληθή).

## Μηχανισμός εξαγωγής συμπερασμάτων

- ❖ Εφαρμόζεται η "Αρχή της ανάλυσης" (**resolution principle**) η οποία σύμφωνα με την αναπαράσταση της Συζευκτικής Λογικής γίνεται:

$$(R_1 \leftarrow Q_1) \wedge (\leftarrow R_1) \vdash \leftarrow Q_1$$

- ❖ Αυτή η μέθοδος αναπαράστασης χρησιμοποιείται από τη γλώσσα προγραμματισμού **Prolog**.

## Παράδειγμα

```
father(george,mary) .      father(george,nick) .  
father(peter,marina) .    father(nick,jim) .  
father(gus,james) .  
mother(helen,mary) .      mother(helen,nick) .  
mother(ann,marina) .      mother(mary,james) .  
mother(katie,jim) .
```

```
parent(X,Y) :- father(X,Y) .  
parent(X,Y) :- mother(X,Y) .  
gfather(X, Y) :- parent(X, Z) , parent(Z, Y) .
```

```
?gfather(george, james) .
```

Απάντηση: Yes

```
?gfather(george, X) .
```

Απάντηση: X=james, X=jim

# Μονότονη λογική

- ❖ Σε μια *μονότονη* λογική, υπάρχει ένα σύστημα αξιωμάτων  $S$  (η αρχική βάση γνώσης) και ένα σύνολο τύπων  $F$  που αποδεικνύονται (συνάγονται) από το  $S$ .
- ❖ Η προσθήκη ενός ή περισσότερων αξιωμάτων στο  $S$  (απόκτηση νέας γνώσης), αυξάνει *μονότονα* το σύνολο  $F$ .
  
- ❖ Πλεονεκτήματα:
  - Κάθε φορά που προστίθεται ένα νέο γεγονός στο  $S$ , δε χρειάζονται νέοι έλεγχοι για τη συνέπεια της γνώσης του συστήματος.
  - Για κάθε νέο γεγονός που αποδεικνύεται δεν είναι απαραίτητη η καταγραφή των γεγονότων πάνω στα οποία βασίζεται η αλήθεια του, αφού δεν υπάρχει κίνδυνος απομάκρυνσης παλαιότερων γεγονότων.
  
- ❖ Μειονεκτήματα:
  - Η προσθήκη νέων αξιωμάτων είναι δυνατό να μειώσει το σύνολο των δυνατών συμπερασμάτων, αφαιρώντας κάποια που αποδεικνύονται εσφαλμένα μετά την προσθήκη.

# Συλλογιστική εύλογων υποθέσεων

- ❖ Η συλλογιστική εύλογων υποθέσεων (default reasoning) χρησιμοποιείται σε περιπτώσεις κατά τις οποίες ένα γεγονός συνάγεται από ένα δοσμένο γεγονός, γιατί έτσι συμβαίνει συνήθως και γιατί δεν υπάρχει ένδειξη για το αντίθετο.
- ❖ Το πρόβλημα της μονοτονίας αντιμετωπίζεται με την εισαγωγή κατάλληλων μηχανισμών εξαγωγής συμπερασμάτων οι οποίοι καταγράφουν ποια γεγονότα χρησιμοποιήθηκαν για την εξαγωγή ενός νέου συμπεράσματος.
- ❖ Τα συστήματα που χρησιμοποιούν αυτούς τους μηχανισμούς ονομάζονται *συστήματα συντήρησης αλήθειας (truth maintenance systems)*.
  - ❑ TMS (McAllester 1980): Διατηρεί συνεχώς τη συνέπεια ενός συνόλου λογικών ισχυρισμών, ώστε να βρεθεί κάποια λύση σε ένα πρόβλημα ικανοποίησης περιορισμών.
  - ❑ ATMS (De Kleer 1986): Δίνει τη δυνατότητα εύρεσης περισσότερων εναλλακτικών λύσεων μέσω της συλλογιστικής σε παράλληλους κόσμους
    - ✓ Οι κόσμοι είναι εσωτερικά συνεπείς, αλλά μεταξύ τους μπορεί να είναι ασυνεπείς.



# Μη-μονότονη λογική

- ❖ Οι μη-μονότονες συλλογιστικές είναι κατάλληλες για την αντιμετώπιση κάποιων καταστάσεων που εμφανίζονται συχνά στον πραγματικό κόσμο:
  - ❑ Καταστάσεις για τις οποίες δεν έχουμε πλήρη γνώση ή η γνώση δημιουργείται κατά τη διάρκεια της εκτέλεσης ενεργειών, για τις οποίες δεν είμαστε βέβαιοι για την αναγκαιότητα ή ορθότητά τους.
  - ❑ Καταστάσεις στις οποίες η γνώση μεταβάλλεται, λόγω μεταβολών που συμβαίνουν στον κόσμο.
  - ❑ Καταστάσεις στις οποίες το σύστημα χρησιμοποιεί υποθέσεις (assumptions) στα πλαίσια της στρατηγικής επίλυσης προβλημάτων.
  
- ❖ Στη μη-μονότονη τροπική λογική (*non-monotonic modal logic*) εισάγεται ένας νέος τροπικός τελεστής ο οποίος δηλώνει ότι ένα γεγονός "είναι συνεπές με τις τρέχουσες πεποιθήσεις".

# Αναιρέσιμη Λογική

- ❖ Η αναιρέσιμη λογική (*defeasible logic*) είναι απλή αλλά αποδοτική προσέγγιση στη μη-μονότονη λογική
  - ❑ Έχει πολλές εφαρμογές στο σημασιολογικό διαδίκτυο και ηλεκτρονικό εμπόριο
  - ❑ Π.χ. μοντελοποίηση *εμπορικών κανόνων (business rules)* και *κανονισμών (regulations)*, μοντελοποίηση *συμβολαίων (contracts)*, συλλογιστική σε *νομικά θέματα (legal reasoning)*, στρατηγικές *διαπραγμάτευσης πρακτόρων (agent negotiation)*, *ενοποίηση ετερογενών πηγών γνώσης και οντολογιών (ontology integration)*.
- ❖ Η αναιρέσιμη λογική αναπαριστά και διαχειρίζεται *αντιφάσεις (conflicts)* μεταξύ των κανόνων ενός προγράμματος.
  - ❑ Οι αντιφάσεις εκφράζονται ως αντικρουόμενα συμπεράσματα.
  - ❑ Η απλούστερη μορφή μιας αντίφασης είναι όταν το συμπέρασμα ενός κανόνα αποτελεί την άρνηση του συμπεράσματος του άλλου κανόνα.
- ❖ Παράδειγμα:

$r_1: \text{πιγκουϊνος}(X) \rightarrow \text{πουλί}(X)$	Ισχυρός κανόνας
$r_2: \text{πουλί}(X) \Rightarrow \text{πετάει}(X)$	Αναιρέσιμος κανόνας
$r_3: \text{πιγκουϊνος}(X) \Rightarrow \neg \text{πετάει}(X)$	Αναιρέσιμος κανόνας
$r_3 > r_4$	Σχέση Υπεροχής
$r_4: \text{βαρύ}(X) \sim > \neg \text{πετάει}(X)$	Αναιρετής

# Ασκήσεις

- ❖ 7.4. Έστω οι προτάσεις (1)  $P \rightarrow Q$ , (2)  $Q \rightarrow R$ , (3)  $P \wedge W$ . Να αποδείξετε την πρόταση R:
- με τη χρήση των κανόνων συμπερασμού modus ponens και απαλοιφή συζεύξεων
  - με τη χρήση μόνο του κανόνα συμπερασμού της αρχής της ανάλυσης
  - με τη χρήση του κανόνα συμπερασμού της αρχής της ανάλυσης και την απαγωγή σε άτοπο

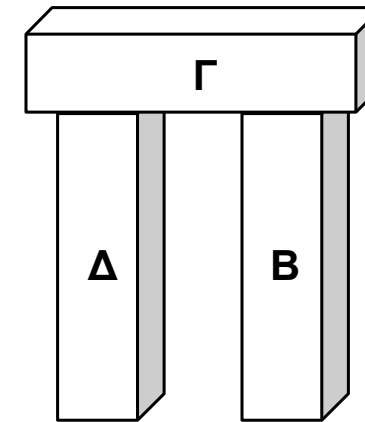
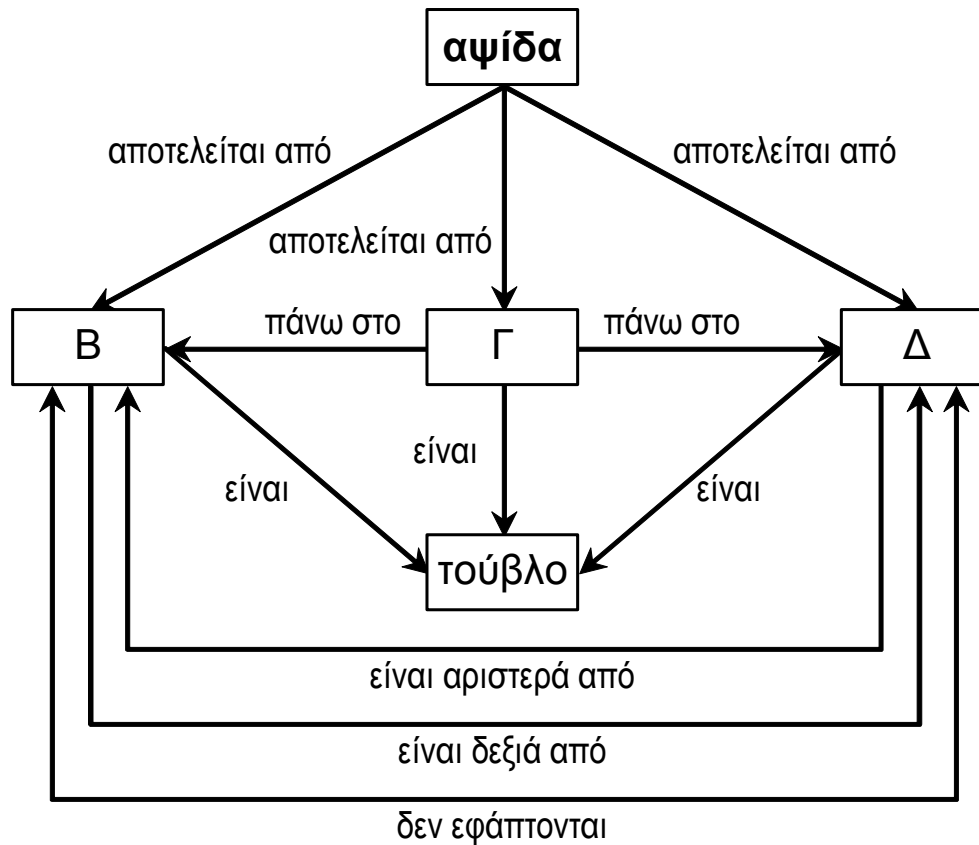
# **Δομημένες Αναπαραστάσεις Γνώσης**

# Δομημένες Αναπαραστάσεις Γνώσης

- ❖ Η κλασική λογική χαρακτηρίζεται από αυστηρότητα στην αναπαράσταση της γνώσης που απαιτείται για την επίλυση ενός προβλήματος.
- ❖ Στην πράξη όμως:
  - ❑ απαιτείται μια λιγότερο αυστηρή και περισσότερο διαισθητική προσέγγιση
  - ❑ είναι επιθυμητή η μείωση του αριθμού των συμβόλων και εκφράσεων που απαιτούνται για την περιγραφή ενός προβλήματος (μείωση όγκου γνώσης)
    - Σημαντική μείωση όγκου επιφέρει η χρήση σύνθετων δομών αναπαράστασης, όπως π.χ. τα αντικείμενα και οι κλάσεις τους (που δεν υποστηρίζονται από την κλασική λογική).
- ❖ Δομημένες Αναπαραστάσεις Γνώσης
  - ❑ Σημασιολογικά Δίκτυα
  - ❑ Πλαίσια
  - ❑ Αντικείμενα
  - ❑ Εννοιολογική Εξάρτηση
  - ❑ Σενάρια
  - ❑ Εννοιολογικοί Γράφοι

# 1. Σημασιολογικά Δίκτυα (semantic networks)

- ❖ Αποτελούνται από κόμβους (nodes) και δεσμούς (links) ανάμεσά τους.
  - ❑ **κόμβοι:** υποδηλώνουν κλάσεις αντικειμένων (classes), αντικείμενα (objects), έννοιες (concepts), τιμές ιδιοτήτων (values)
  - ❑ **δεσμοί:** υποδηλώνουν σχέσεις (relations) μεταξύ αντικειμένων ή ιδιότητες που συνδέουν αντικείμενα με τιμές.

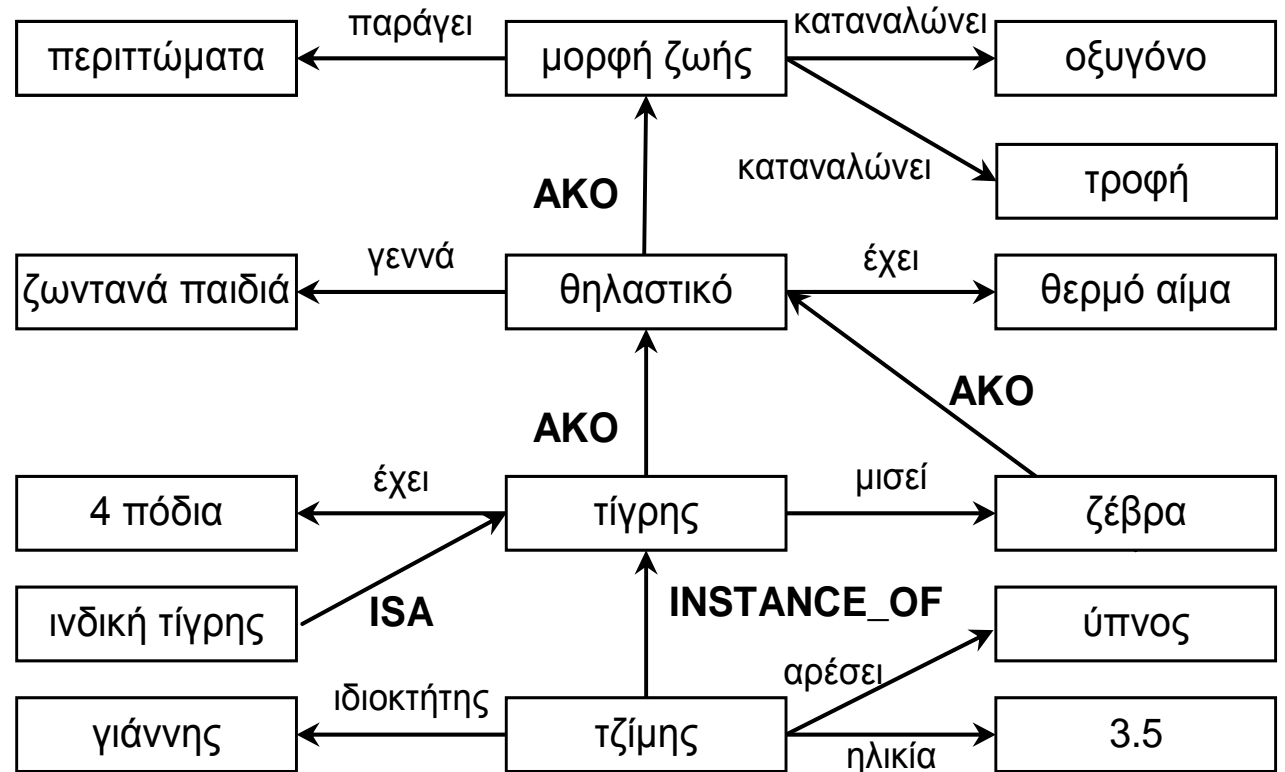


# Η Ιεραρχική Δομή των Σημασιολογικών Δικτύων

- ❖ Υπάρχουν διάφορα είδη δεσμών ή σχέσεων. Σημαντικότεροι είναι οι ΑΚΟ, ISA και INSTANCE\_OF.
- ❖ Η σχέση **ΑΚΟ** υπάρχει **μεταξύ κλάσεων** αντικειμένων. Σε κόμβο που συνδέεται με σχέση ΑΚΟ με κάποιον άλλον μπορούν να προστεθούν νέοι δεσμοί που προσδίδουν νέες ιδιότητες.
  - ❑ η κλάση "τίγρης" είναι ΑΚΟ της κλάσης "θηλαστικό"
- ❖ Η σχέση **ISA** είναι **παρόμοια με τη σχέση ΑΚΟ**, με τη διαφορά ότι δε μπορεί να προστεθούν νέες ιδιότητες παρά μόνον να κληρονομηθούν οι ήδη υπάρχουσες ιδιότητες από κόμβους ψηλότερα στην ιεραρχία ή οι ιδιότητες αυτές να αλλάξουν τιμές.
  - ❑ η κλάση "ινδική τίγρης" είναι ISA της κλάσης "τίγρης"
- ❖ Η σχέση **INSTANCE\_OF** υπάρχει μόνο **μεταξύ κόμβων αντικειμένων και κόμβων γενικότερων κλάσεων**.
  - ❑ Δεν μπορεί να αποτελεί υπερκλάση άλλης κλάσης.
  - ❑ ο συγκεκριμένος τίγρης "τζίμης" είναι INSTANCE\_OF της κλάσης "τίγρης"

# Κληρονομικότητα στα Σημασιολογικά Δίκτυα

- ❖ Χάρη στην ιεραρχία ένα αντικείμενο κληρονομεί ιδιότητες από μία υψηλότερη ιεραρχικά κλάση από αυτή στην οποία ανήκει.
- ❖ Για το συγκεκριμένο τίγρη ("τζίμη") δε χρειάζεται να δηλωθούν παρά μόνον τα χαρακτηριστικά αυτά που είναι αποκλειστικά δικά του ("ιδιοκτήτης", "ηλικία", "αρέσει").
- ❖ Τα υπόλοιπα χαρακτηριστικά-ιδιότητες κληρονομούνται από την ιεραρχία των κλάσεων στις οποίες υπάγεται ο τίγρης

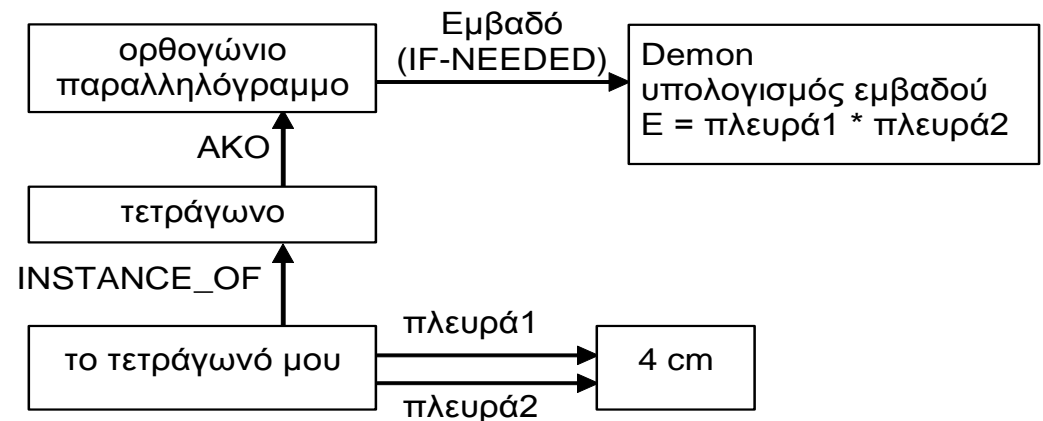




# Προσκόλληση Διαδικασιών

- ❖ Αντί για την τιμή της ιδιότητας μπορεί να οριστεί μια διαδικασία που θα καλείται για να δώσει κάποιο αποτέλεσμα, *μόνον εάν χρειάζεται (IF-NEEDED)*.

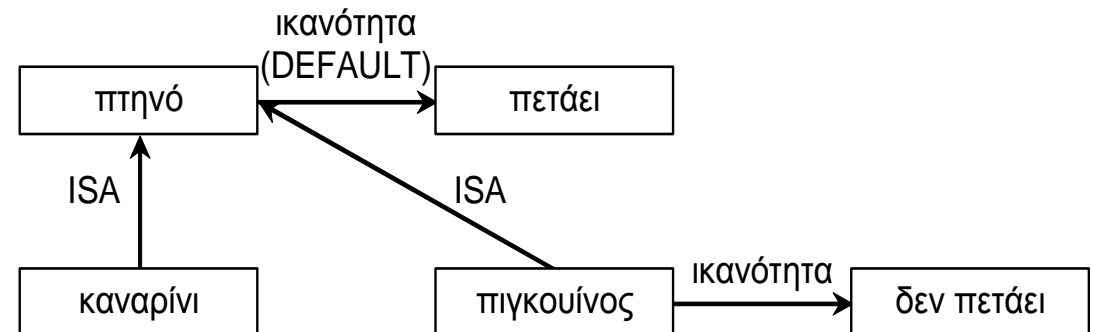
- ❑ Οι διαδικασίες αυτές ονομάζονται και *δαίμονες (daemons)*.



## Προκαθορισμένες Τιμές και Εξαιρέσεις

- ❖ Η συνήθης τιμή μιας ιδιότητας σε ένα κόμβο που βρίσκεται ψηλά στην ιεραρχία μπορεί να προκαθοριστεί και ονομάζεται *προκαθορισμένη τιμή (DEFAULT)*

- ❑ Αν δεν υπάρχουν πληροφορίες για την τιμή μιας ιδιότητας, εύλογες υποθέσεις αποτελούν οι προκαθορισμένες τιμές.
- ❑ Όταν και αν χρειαστεί, οι προκαθορισμένες τιμές αυτές μπορεί να αλλάξουν σε άλλες κλάσεις ή αντικείμενα, χαμηλότερα στην ιεραρχία



- ❑ Είναι ένας τρόπος για να υλοποιηθεί η συλλογιστική των εύλογων υποθέσεων.

## Πλεονεκτήματα

- ❖ Συμπαγής τρόπος αναπαράστασης λόγω της κληρονομικότητας και των προσκολλημένων διαδικασιών
- ❖ Δυνατότητα εξαγωγής συμπερασμάτων λόγω της ιεραρχικής δομής.

## Μειονεκτήματα

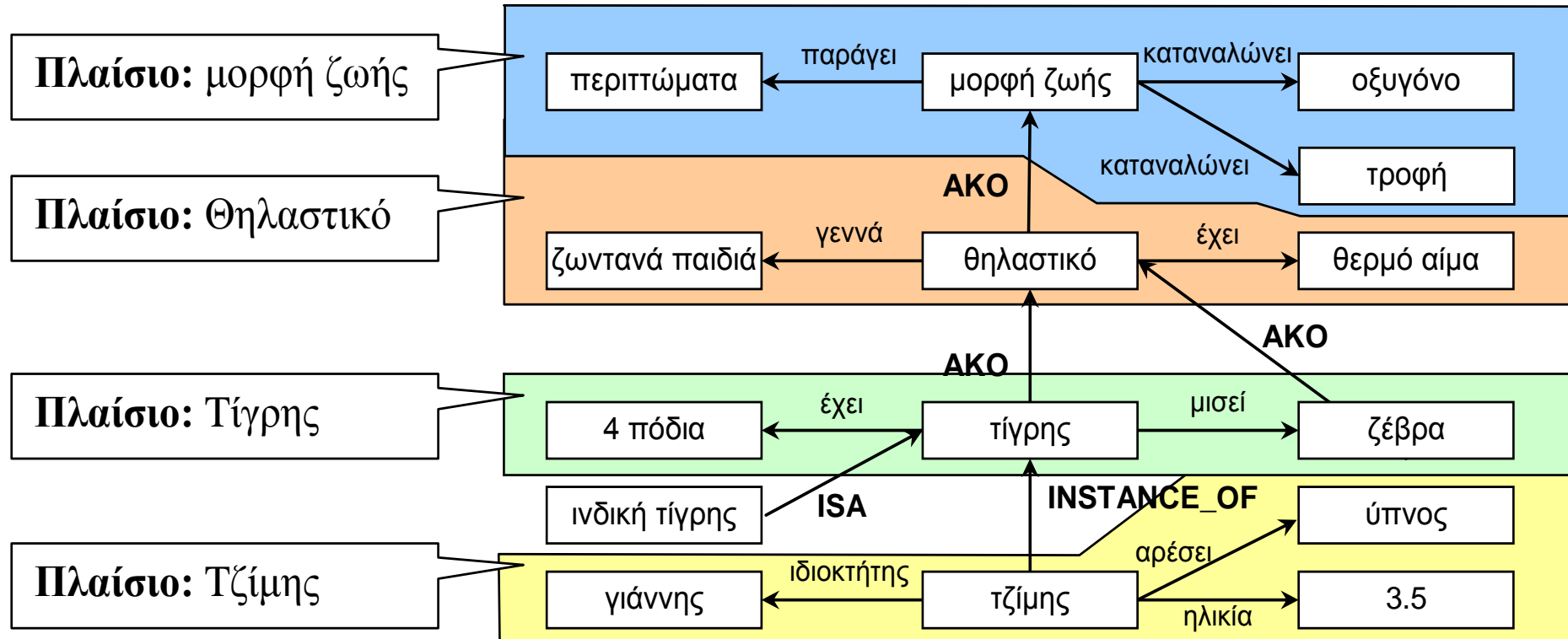
- ❖ Η γνώση που περιγράφεται είναι πολλές φορές διάσπαρτη μέσα στα δίκτυα:
  - ❑ η αναζήτηση παίρνει πολύ χρόνο
  - ❑ η παραμικρή αλλαγή επιφέρει σημασιολογικές αλλαγές στον κόσμο που αναπαριστάται
- ❖ **Ευριστική ανεπάρκεια** (heuristic inadequacy)
  - ❑ Λόγω της πολυπλοκότητας της δομής, κάθε φορά που γίνεται αναζήτηση συγκεκριμένης πληροφορίας μπορεί να εμφανιστεί το φαινόμενο της *συνδυαστικής έκρηξης* (*combinatorial explosion*).
- ❖ **Λογική ανεπάρκεια** (logical inadequacy)
  - ❑ Η σημασιολογία των αντικειμένων δεν είναι ξεκάθαρη
  - ❑ ένα "αυτοκίνητο" στο σημασιολογικό δίκτυο μπορεί να εκληφθεί ως οποιοδήποτε αυτοκίνητο (κλάση) ή ένα συγκεκριμένο αυτοκίνητο (αντικείμενο)
- ❖ **Βασικότερο μειονέκτημα:** δεν υπάρχουν προκαθορισμένες δομικά σχέσεις πάνω στις οποίες μπορεί να στηριχτεί ο σχεδιασμός τους.

## 2α. Πλαίσια (frames)

- ❖ Ορίστηκαν από τον Minsky σαν "δομές δεδομένων για την αναπαράσταση στερεότυπων καταστάσεων". Ονομάζονται και *σχήματα (schemata)*.
  - ❑ **Βασική ιδέα:** το ανθρώπινο μυαλό συγκρατεί μόνο σημαντικές πληροφορίες για αντικείμενα που ανήκουν στην ίδια κατηγορία και είναι όλα τυπικά παραδείγματα της κατηγορίας αυτής.
- ❖ Τα πλαίσια έχουν:
  - ❑ όνομα,
  - ❑ μία σειρά από **ιδιότητες (slots)** που συνδέονται άμεσα με τις **τιμές τους (fillers)**,
    - Οι ιδιότητες μπορεί να έχουν **προκαθορισμένες τιμές** που χρησιμοποιούνται όταν δεν υπάρχει άλλη διαθέσιμη πληροφορία.
    - Μία ειδική κατηγορία ιδιοτήτων είναι οι **όψεις (facets)**, που περιγράφουν τα είδη ή το εύρος τιμών που μπορεί να πάρουν οι ιδιότητες.
  - ❑ **προσκολλημένες διαδικασίες** (όχι υποχρεωτικά) που ονομάζονται **δαίμονες (demons)** και που μπορεί να ενεργοποιούνται όταν τα πλαίσια μεταβάλλονται για κάποιο λόγο.
- ❖ Παρόλο που η συγκέντρωση της γνώσης και η οργάνωσή της σε πολύπλοκες δομές απαιτούν επιδεξιότητα και επίπονη εργασία, τα πλαίσια εξελίχθηκαν σε έναν σημαντικό τρόπο αναπαράστασης γνώσης σε προβλήματα της ΤΝ.

# Ιεραρχία στα Πλαίσια

- ❖ Τα πλαίσια μπορούν να αποτελέσουν αντικείμενα-κόμβους ενός σημασιολογικού δικτύου και να συνδεθούν με μία ιεραρχία (όπως στα σημασιολογικά δίκτυα).



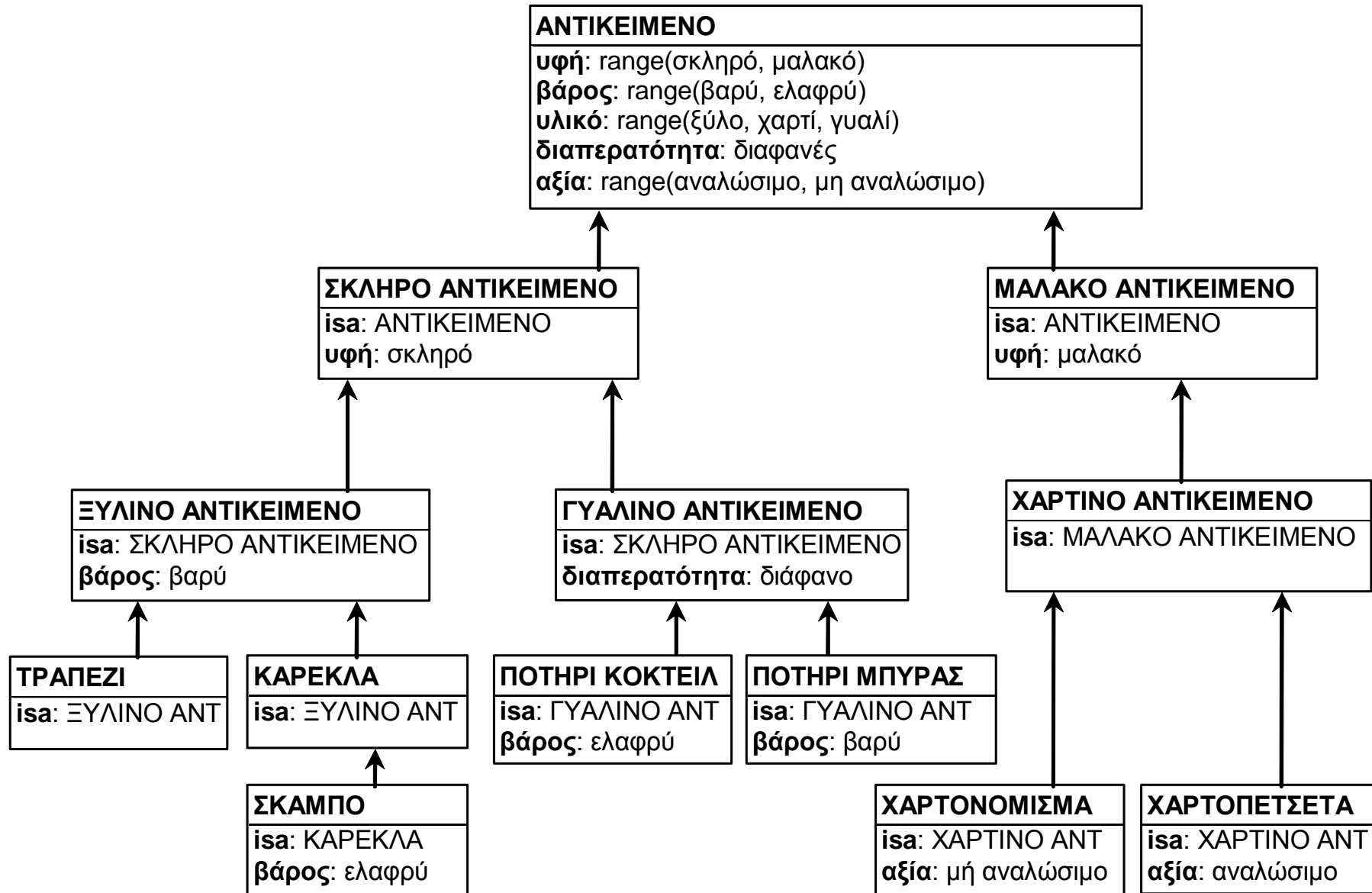
# Πλαίσια και Σημασιολογικά Δίκτυα

- ❖ Τα πλαίσια (όπως και τα σημασιολογικά δίκτυα) παρέχουν:
  - ❑ κληρονομικότητα,
  - ❑ προσκόλληση διαδικασιών και
  - ❑ προκαθορισμένες τιμές
  
- ❖ Τα πλαίσια υπερτερούν έναντι των σημασιολογικών δικτύων στο ότι:
  - ❑ Η ιεραρχία των εννοιών στα πλαίσια είναι πιο ξεκάθαρη σε σχέση με τα σημασιολογικά δίκτυα (**βασικότερη διαφορά**):
    - Ένα πλαίσιο περιέχει **όλη** την πληροφορία για τη συγκεκριμένη έννοια που αναπαριστά.
    - Ένας κόμβος σε ένα σημασιολογικό δίκτυο αναπαριστά **μόνο** την έννοια, ενώ οι ιδιότητές της περιγράφονται σε **άλλους** κόμβους που συνδέονται με αυτόν.
  - ❑ Τα πλαίσια παρακάμπτουν το μειονέκτημα των σημασιολογικών δικτύων που αναφέρεται ως *λογική ανεπάρκεια*, καθώς δεν εμφανίζουν τόσο έντονο το φαινόμενο της συνδυαστικής έκρηξης στην αναζήτηση πληροφορίας.

# Πλαίσια και Εγγραφές

- ❖ Τα πλαίσια θυμίζουν ίσως τις εγγραφές (*records*) των γλωσσών προγραμματισμού, γιατί ομαδοποιούν πληροφορίες που σχετίζονται μεταξύ τους.
- ❖ Όμως τα πλαίσια:
  - ❑ δεν είναι κατ' ανάγκη όμοια μεταξύ τους, όπως οι εγγραφές,
  - ❑ δεν περιέχουν ίδιου τύπου πληροφορίες, ούτε μόνον απλά δεδομένα,
  - ❑ οργανώνονται σε ιεραρχικές δομές υποστηρίζοντας ταυτόχρονα κληρονομικότητα, και
  - ❑ μπορεί να έχουν προσκολλημένες διαδικασίες.
- ❖ Τελικά τα πλαίσια μοιάζουν περισσότερο με τα αντικείμενα (*objects*) του αντικειμενοστραφούς προγραμματισμού και λιγότερο με τις εγγραφές.

# Παράδειγμα Ιεραρχίας Πλαισίων



# Εξαγωγή Συμπερασμάτων με Πλαίσια

- ❖ Η συλλογιστική στα πλαίσια μπορεί να υλοποιηθεί με οποιονδήποτε από τους μηχανισμούς συμπερασμάτων της λογικής.
- ❖ Επιπλέον χρειάζεται ένας τρόπος για την ανάκληση της τιμής μίας ιδιότητας:

Διαδικασία `find(Frame,Attribute,Value)`

Αν η ιδιότητα `Attribute` υπάρχει στο πλαίσιο `Frame`, τότε επέστρεψε την τιμή της `Value`

Αλλιώς,

ακολούθησε την ιεραρχία δεσμών `ISA` ή `AKO` ή `INSTANCE_OF` και επανέλαβε τη διαδικασία με νέο πλαίσιο `NewFrame` το αμέσως παραπάνω πλαίσιο του `Frame` στην ιεραρχίας, δηλ. `find(NewFrame,Attribute,Value)`.

- ❖ Λόγω δυνατότητας ύπαρξης **πολλαπλής κληρονομικότητας** (*multiple inheritance*), όπου μία ιδιότητα υπάρχει σε δύο υπερκλάσεις ενός πλαισίου, ο αλγόριθμος αναζήτησης πρέπει να καθορίζει από ποιο πλαίσιο θα κληρονομηθεί η ιδιότητα.
  - ❑ π.χ. η αναζήτηση πρώτα σε πλάτος θα επιστρέψει την τιμή της ιδιότητας που βρίσκεται στο πιο κοντινό πλαίσιο.



## 2β. Πλαίσια (frames) ..... (συνοπτική παρουσίαση)

Τα πλαίσια (*frames*) ή σχήματα (*schemata*) είναι "δομές δεδομένων για την αναπαράσταση στερεότυπων καταστάσεων".

- ❖ Είναι ο πιο προσφιλής τρόπος δομημένης αναπαράστασης γνώσης και αποτελούν την *αντικειμενοστραφή* (*object oriented*) του προσέγγιση
- ❖ Πρώτος τα όρισε ο Minsky. Η ιδέα του ήταν ότι το ανθρώπινο μυαλό συγκρατεί μόνο σημαντικές πληροφορίες για αντικείμενα που ανήκουν στην ίδια κατηγορία.
- ❖ Τα πλαίσια έχουν:
  - ❑ Όνομα
  - ❑ Μία σειρά από *ιδιότητες* (*slots*) που συνδέονται άμεσα με τις *τιμές* τους (*fillers*).
  - ❑ Ένα πλαίσιο περιέχει όλη την πληροφορία για τη συγκεκριμένη έννοια που αναπαριστά.

# Η ιεραρχική δομή των πλαισίων

Υπάρχουν διάφορα είδη δεσμών ή σχέσεων, AKO, ISA, INSTANCE\_OF.

- ❖ Η σχέση **AKO** υπάρχει **μεταξύ κλάσεων αντικειμένων**. Σε κόμβο που συνδέεται με σχέση AKO με κάποιον άλλον μπορούν να προστεθούν νέοι δεσμοί που προσδίδουν νέες ιδιότητες.
- ❖ Η σχέση **ISA** είναι **παρόμοια με τη σχέση AKO, με τη διαφορά ότι** δε μπορούν να προστεθούν νέες ιδιότητες παρά μόνον να κληρονομηθούν οι ήδη υπάρχουσες ιδιότητες από κόμβους ψηλότερα στην ιεραρχία ή οι ιδιότητες αυτές να αλλάξουν τιμές.
  - ❑ Μπορεί να αποτελεί υπερκλάση άλλης κλάσης.
- ❖ Η σχέση **INSTANCE\_OF** είναι παρόμοια με τη σχέση ISA, αλλά υπάρχει μόνο **μεταξύ κόμβων αντικειμένων και των κόμβων κλάσεων** στις οποίες ανήκουν.  
Δεν μπορεί να αποτελεί υπερκλάση άλλης κλάσης.

❖ Επίσης, στα πλαίσια υπάρχει και μία ειδική κατηγορία ιδιοτήτων, οι **όψεις (facets)**, που περιγράφουν τα είδη ή το εύρος τιμών που μπορούν να πάρουν οι ιδιότητες.

### ❖ Πλεονεκτήματα:

- ❑ κληρονομικότητα,
- ❑ προσκόλληση διαδικασιών και
- ❑ προκαθορισμένες τιμές

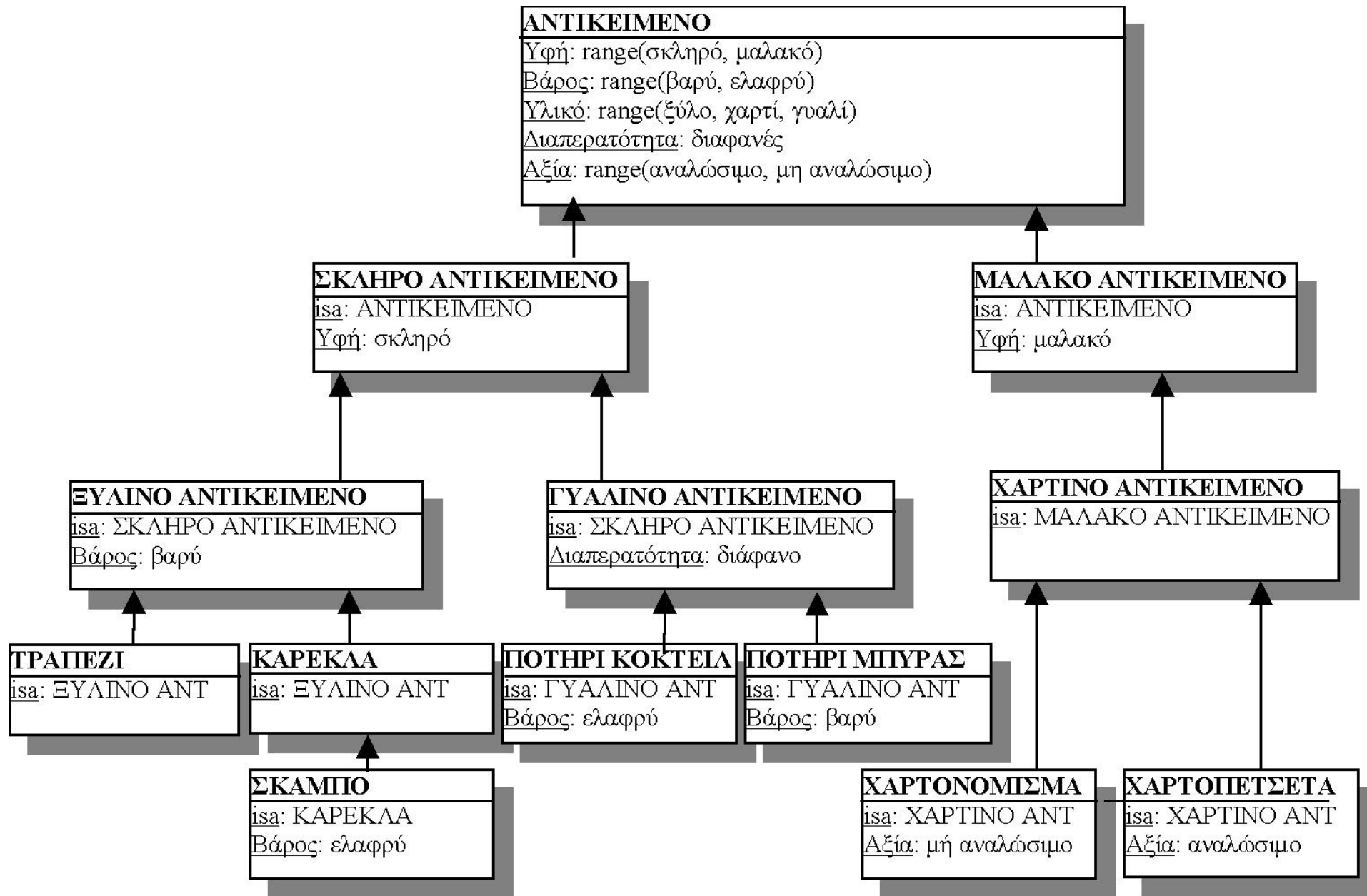
❖ Τα πλαίσια μοιάζουν πολύ με τις **εγγραφές (records)**. Οι κύριες διαφορές τους είναι:

- ❑ Τα πλαίσια δεν είναι κατ' ανάγκη όμοια μεταξύ τους
- ❑ Δεν περιέχουν ίδιου τύπου πληροφορίες, ούτε μόνον απλά δεδομένα
- ❑ Τα πλαίσια οργανώνονται σε ιεραρχικές δομές και όχι σειριακές

## Κληρονομικότητα

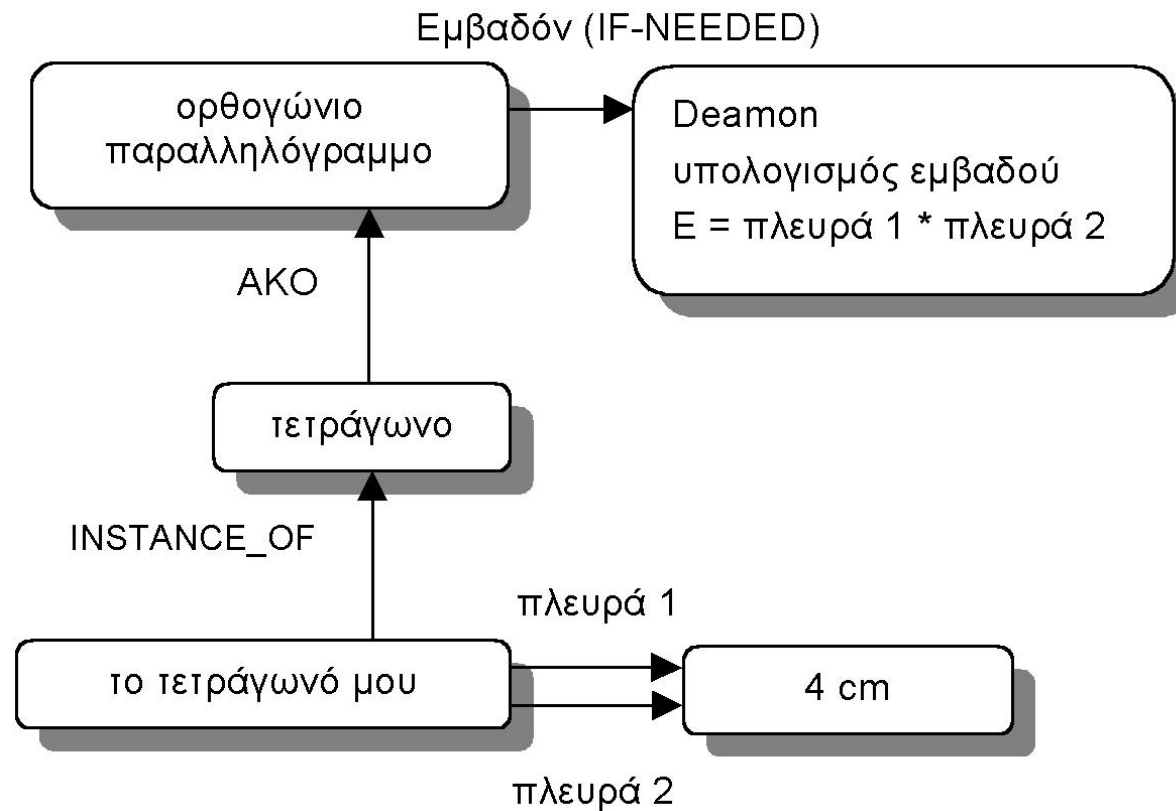
❖ Χάρη στην ιεραρχία ένα αντικείμενο κληρονομεί ιδιότητες από μία γενικότερη κλάση στην οποία ανήκει.

# Παράδειγμα 1



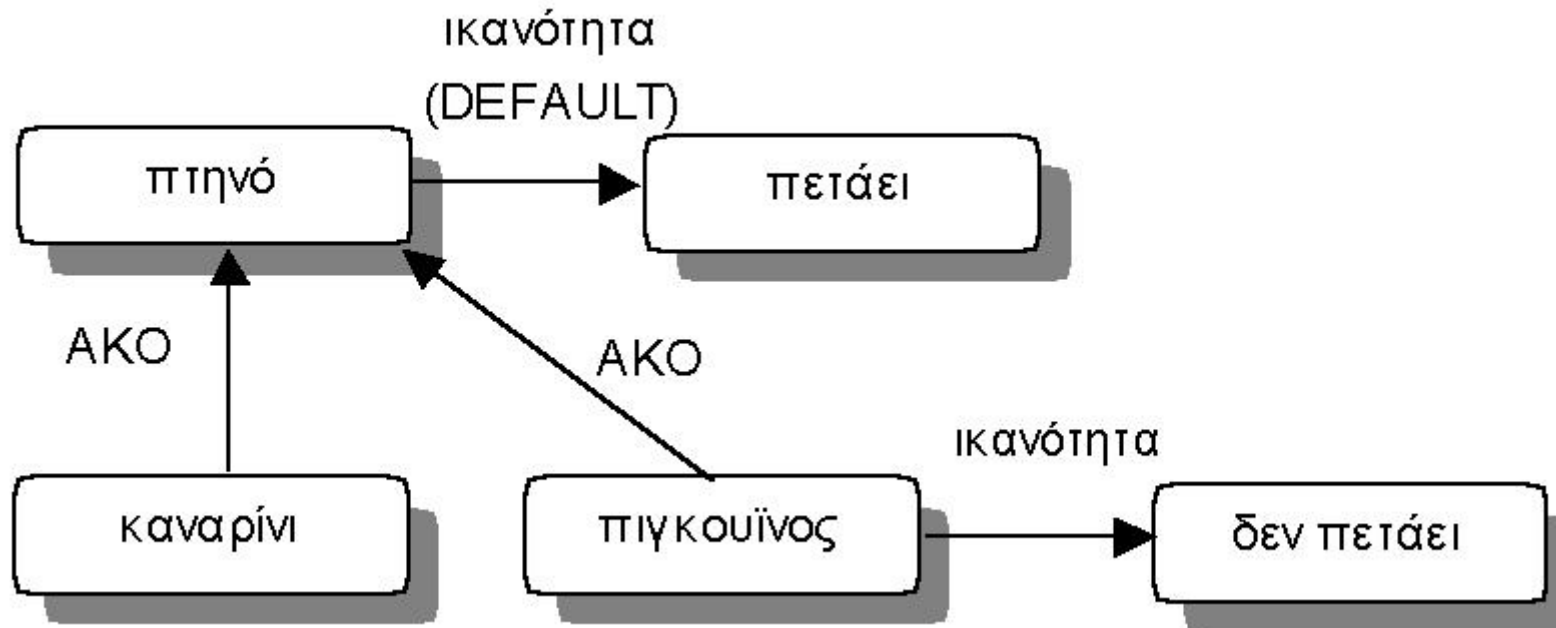
# Προσκόλληση διαδικασιών

- ❖ Αντί για την τιμή της ιδιότητας μπορεί να οριστεί μια διαδικασία η οποία θα καλείται *μόνον εάν χρειάζεται (IF-NEEDED)* για να δώσει κάποιο αποτέλεσμα.
- ❖ Οι διαδικασίες αυτές ονομάζονται και *δαίμονες (daemons)*.



# Ιδιότητες με προκαθορισμένες τιμές και εξαιρέσεις τους

- ❖ Η συνήθης τιμή μιας ιδιότητας που εμφανίζεται σε ένα κόμβο που βρίσκεται ψηλά στην ιεραρχία μπορεί να προκαθοριστεί και ονομάζεται *προκαθορισμένη τιμή (DEFAULT)*.



Οι προκαθορισμένες τιμές είναι ένας τρόπος για να υλοποιηθεί η συλλογιστική των εύλογων υποθέσεων που γίνονται για κλάσεις αντικειμένων στα σημασιολογικά δίκτυα.

## Παράδειγμα 2 (σε FLEX. Υποστηρίζει μόνο ISA και INSTANCE)

Mammal:

is\_a: Animal  
has\_head: yes  
warm\_blooded: yes  
eats: everything

Carnivore:

is\_a: Animal  
eats: meat  
pointed\_Teath: yes

Elephant:

is\_a: Mammal  
colour: grey  
size: large

Human:

is\_a: Mammal  
birthday: {date} *{Τύπος δεδομένων}*  
age: [birthday-current\_date()]/365  
*{Δαίμονας - Διαδικασία υπολογισμού}*

Feline:

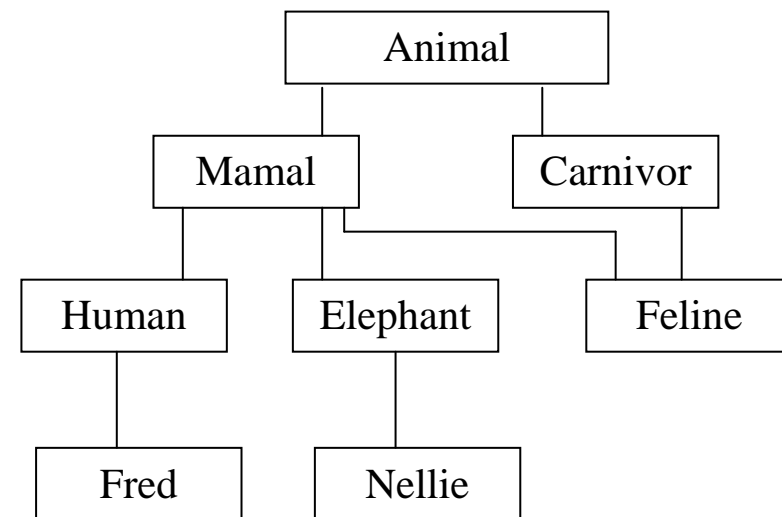
*{Πολλαπλή κληρονομικότητα}*  
is\_a: Mammal, Carnivore  
sharp\_claws: yes

Fred:

is\_a: Human  
birthday: 05/10/1972  
owns: Nellie *{Όνομα άλλου πλαισίου}*  
buys: {string}

Nellie:

instance\_of: Elephant  
size : small





Animal

Mammal:  
 is\_a: Animal  
 has\_head: yes  
 warm\_blooded: yes  
 eats: everything

Carnivore:  
 is\_a: Animal  
 eats: meat  
 pointed\_Teath: yes

Human:  
 is\_a: Mammal  
 birthday: {date} *{Τύπος δεδομένων}*  
 age: [birthday-current\_date()]/365  
*{Δαίμονας - Διαδικασία υπολογισμού}*

Feline:  
 is\_a: Mammal, Carnivore  
*{Πολλαπλή κληρονομικότητα}*  
 sharp\_claws: yes

Fred:  
 is\_a: Human  
 birthday: 05/10/1972  
 owns: Nellie *{Όνομα άλλου  
πλασιίου}*  
 buys: {string}

Elephant:  
 is\_a: Mammal  
 colour: grey  
 size: large

Nellie:  
 instance\_of: Elephant  
 size : small

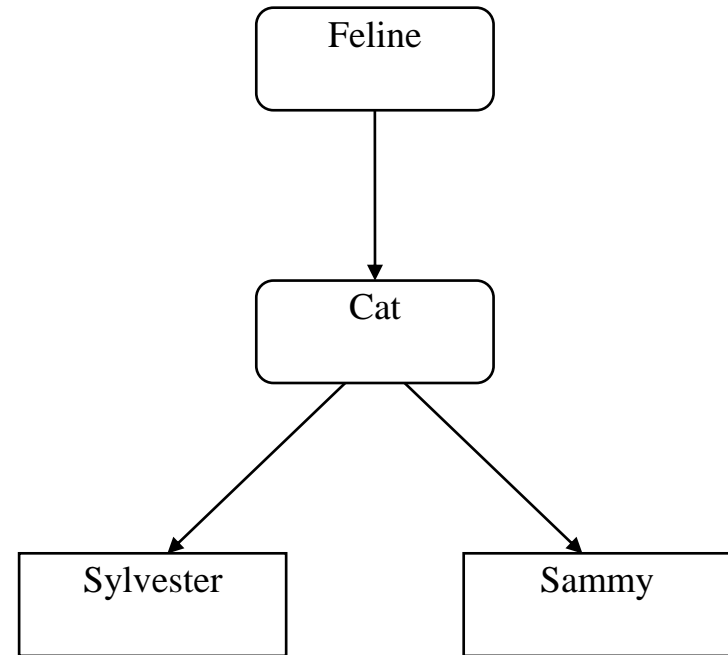


# Παράδειγμα (σε Flex)

**frame** feline **is a** mammal, carnivore ;  
**default** legs **are** 4 .

**frame** cat **is a** feline  
**default** habitat **is** house **and**  
**default** meal **is** kit\_e\_kat .

**instance** sylvester **is a** kind **of** cat .  
**instance** sammy **is an** instance **of** cat .



- ❖ Σ' αυτό το παράδειγμα εξ' ορισμού τόσο ο sylvester όσο και η sammy κατοικούν σε σπίτι, τρώνε kit\_e\_kat και έχουν τέσσερα πόδια.
- ❖ Στο διάγραμμα τα πλαίσια παριστάνονται με σχήματα που έχουν στρογγυλεμένες γωνίες, ενώ τα στιγμιότυπα με ορθογώνια.

# Πλαίσια

## Μηχανισμός εξαγωγής συμπερασμάτων

- ❖ Διαδικασία βρες(Frame, Attribute, Value)
  - ❑ Αν η ιδιότητα Attribute υπάρχει στο πλαίσιο Frame, τότε επέστρεψε την τιμή της Value
  - ❑ Αλλιώς, ακολούθησε την ιεραρχία δεσμών ISA ή ΑΚΟ ή INSTANCE\_OF και επανέλαβε τη διαδικασία με νέο πλαίσιο NewFrame το αμέσως παραπάνω πλαίσιο του Frame στην ιεραρχίας, δηλ. βρες(NewFrame, Attribute, Value).

## Μειονεκτήματα

- ❖ Πολλαπλή κληρονομικότητα (multiple inheritance).

### 3. Αντικείμενα (objects)

- ❖ Τρόπος αναπαράστασης δομημένης γνώσης που προήλθε από την έρευνα για γλώσσες προσομοίωσης.
  - ❑ υπήρχε η ανάγκη για τη δημιουργία μοντέλων των υπό εξέταση προβλημάτων με βάση τον τρόπο που αυτά γίνονται αντιληπτά από τον άνθρωπο
  - ❑ αποδόμηση (decomposition) ενός προβλήματος σε αλληλεπιδρώντα αντικείμενα που το κάθε ένα αντιπροσωπεύει συνήθως μία οντότητα του φυσικού κόσμου

#### Βασικά χαρακτηριστικά αντικειμένων:

- ❑ έχουν κάποια κατάσταση που περιγράφεται από ένα σύνολο **ιδιοτήτων** (*properties-fields*)
- ❑ έχουν ένα σύνολο από **μεθόδους** (*methods*) που ορίζουν τη συμπεριφορά του αντικειμένου
  - Οι μέθοδοι μπορεί να υλοποιούν από μία απλή επιστροφή της τιμής κάποιου χαρακτηριστικού έως κάποιον αρκετά πολύπλοκο αλγόριθμο.
  - Ο τρόπος με τον οποίο υλοποιείται μία μέθοδος δεν είναι "ορατός" στον υπόλοιπο κόσμο. Μόνο το αποτέλεσμα της εκτέλεσής της γίνεται γνωστό.
- ❑ αντιδρούν σε προκαθορισμένα **μηνύματα** ή **γεγονότα** (*messages ή events*) που λαμβάνουν από τον εξωτερικό κόσμο
  - για κάθε μήνυμα υπάρχει και η "αρμόδια" μέθοδος του αντικειμένου που θα το χειριστεί
  - πρακτικά, τα μηνύματα προκαλούν την κλήση της αντίστοιχης μεθόδου.

# Οργάνωση Αντικειμένων

- ❖ Το σύνολο των ιδιοτήτων των μεθόδων και των μηνυμάτων στα οποία αποκρίνεται ένα αντικείμενο ορίζουν την *κλάση* (*class*) του αντικειμένου.
- ❖ Οι κλάσεις
  - ❑ αποκρύπτουν την εσωτερική πολυπλοκότητα των αντικειμένων (*εγκλεισμός* – *encapsulation*) και αποτελούν πρότυπα για την παραγωγή *στιγμιότυπων* (*instances*) της κλάσης
  - ❑ μπορούν να κρύβουν τμήμα της εσωτερικής πληροφορίας των αντικειμένων από τον έξω κόσμο και μάλιστα με διαβαθμισμένο τρόπο
  - ❑ είναι συνήθως οργανωμένες σε *ιεραρχίες* με τις γενικότερες κλάσεις να βρίσκονται ψηλότερα στην ιεραρχία (π.χ. "*όχημα*" γενικότερη από "*αυτοκίνητο*")
  - ❑ υποστηρίζουν *κληρονομικότητα* (*inheritance*): η δομή και η συμπεριφορά μιας γενικότερης κλάσης κληρονομείται στις περισσότερο συγκεκριμένες
    - αποφυγή της επανάληψης ορισμού κοινών χαρακτηριστικών και μεθόδων
  - ❑ υποστηρίζουν *πολυμορφισμό* (*polymorphism*): μια μέθοδος μπορεί απλά να είναι δηλωμένη στη μητρική κλάση (*virtual method*) και να υλοποιείται κατά βούληση στις κλάσεις-παιδιά που την κληρονομούν
  - ❑ υποστηρίζουν *υπερφόρτωση* (*overloading*): μια πλήρως ορισμένη μέθοδος (όχι virtual) που κληρονομήθηκε μπορεί να οριστεί εκ νέου.
  - ❑ υποστηρίζουν *πολλαπλή κληρονομικότητα* (*multiple inheritance*): μία κλάση μπορεί να κληρονομεί χαρακτηριστικά και μεθόδους από πολλές γενικότερες κλάσεις

# Αντικείμενα και Πλαίσια

- ❖ **Ομοιότητες:** δομημένη περιγραφή, κληρονομικότητα, ιδιότητες-slots, μέθοδοι-δαίμονες.
- ❖ **Βασικές Διαφορές**
  - ❑ Τα αντικείμενα δεν είναι υποχρεωτικό να ακολουθήσουν αυστηρά τις προδιαγραφές που ορίζει η κλάση τους (περισσότερη ευελιξία και εκφραστικότητα στον προγραμματιστή).
  - ❑ Η αυστηρά καθορισμένη δομή των αντικειμένων, διευκολύνει τον έλεγχο ορθότητας των προγραμμάτων και επιτρέπει την παραγωγή αποδοτικότερου κώδικα.
  - ❑ Η πρόσβαση στις ιδιοτήτων και τις μεθόδους των αντικειμένων είναι *διαβαθμισμένη* ενώ στα πλαίσια οι τιμές των *ιδιοτήτων (slots)* είναι πάντα προσβάσιμες για ανάγνωση και εγγραφή από τη γλώσσα προγραμματισμού.
  - ❑ Τα αντικείμενα εμπεριέχουν τον κώδικα ελέγχου μέσα τους, με τη μορφή μεθόδων. Στα πλαίσια ο υπόλοιπος κώδικας είναι αποθηκευμένος εκτός των πλαισίων.
  - ❑ Στα πλαίσια, οι δαίμονες ενεργοποιούνται αυτόματα όταν γίνει πρόσβαση στις ιδιότητες. Στα αντικείμενα οι μέθοδοι, κατά κύριο λόγο, ενεργοποιούνται με την εκούσια αποστολή μηνυμάτων από τους χρήστες ή από μεθόδους άλλων αντικειμένων.
- ❖ Λόγω των παραπάνω, η χρήση αντικειμένων αντί πλαισίων στην TN είναι πλέον περισσότερο συχνή.

## 4. Εννοιολογική Εξάρτηση (conceptual dependency)

- ❖ Προέκυψε (Schank, '74) ως αποτέλεσμα των προσπαθειών να ενσωματωθούν οι βασικές σημασιολογικές σχέσεις της φυσικής γλώσσας στον ίδιο το φορμαλισμό, παρά να αποτελούν τμήμα του πεδίου γνώσης (*domain knowledge*). Σαν αποτέλεσμα:
  - ❑ η υλοποίηση συστημάτων βασισμένων σε γνώση θα απαιτούσε λιγότερη προσπάθεια,
  - ❑ θα υπήρχε το όφελος της γενικότητας και της συνέπειας του τρόπου αναπαράστασης
- ❖ Υπάρχουν τέσσερις **αρχέγονες εννοιολογικές μορφές** (*primitive conceptualizations*) πάνω στις οποίες μπορεί να βασιστεί η ερμηνεία.
  - ❑ **ACTs** (ενέργειες), **PPs** (αντικείμενα), **AAs** (Προσδιορισμοί Ενεργειών), **PAs** (Προσδιορισμοί Αντικειμένων)
  - ❑ αυτές αναλύονται περαιτέρω ώστε να καλύψουν περισσότερο εξειδικευμένες περιπτώσεις
- ❖ Υπάρχουν **σύμβολα** για τον ακριβέστερο καθορισμό χρόνου και τρόπου.
  - ❑ π.χ. **p** (παρελθόν), **f** (μέλλον), **c** (υπό προϋποθέσεις), κτλ.
- ❖ Τα αρχέγονα στοιχεία και σύμβολα χρησιμοποιούνται για να οριστούν σταθερές και με καλά ορισμένη σημασιολογία, σχέσεις μεταξύ των εννοιών (**σχέσεις εννοιολογικής εξάρτησης** - *conceptual dependency relationships*)

❖ Οι σχέσεις εννοιολογικής εξάρτησης είναι *εννοιολογικοί συντακτικοί κανόνες* και μπορεί να χρησιμοποιηθούν για την αναπαράσταση του νοήματος προτάσεων φυσικής γλώσσας με τρόπο τέτοιο ώστε:

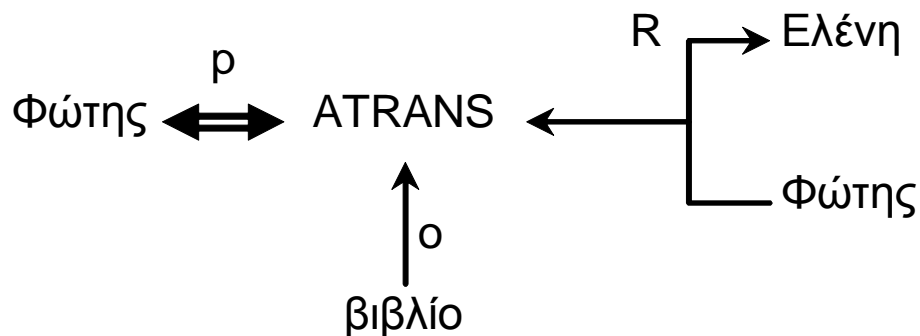
- ❑ να είναι εύκολη η εξαγωγή συμπερασμάτων από προτάσεις
- ❑ η αναπαράσταση να είναι ανεξάρτητη από τη γλώσσα διατύπωσης της πρότασης

## Παράδειγμα Σχέσεων Εννοιολογικής Εξάρτησης

Σχέση	Ερμηνεία
$PP \Leftrightarrow ACT$	Κάποιος ενεργεί.
$ACT \xleftarrow{O} PP$	Το αντικείμενο κάποιας ενέργειας.
$ACT \xleftarrow{R} \begin{array}{l} \rightarrow PP \\ \rightarrow PP \end{array}$	Ο δότης και ο παραλήπτης ενός αντικειμένου σε μία ενέργεια.

# Γράφοι Εννοιολογικής Εξάρτησης

- ❖ Γράφοι που προκύπτουν κατά την αναπαράσταση προτάσεων με χρήση στοιχείων της θεωρίας της εννοιολογικής εξάρτησης. (*conceptual dependency graphs*).
- ❖ **Παράδειγμα:** Έστω η πρόταση: "Ο Φώτης έδωσε το βιβλίο στην Ελένη".



- ❑ τα βέλη δείχνουν την κατεύθυνση της εξάρτησης,
- ❑ το διπλό βέλος " $\Leftrightarrow$ " σημαίνει ότι υπάρχει αμφίδρομη σχέση μεταξύ του δράστη και της πράξης,
- ❑ το "p" σημαίνει παρελθόν (past),
- ❑ το "ΑΤΡΑΝΣ" σημαίνει μεταφορά κτήσης ενός πράγματος (abstract transitive),
- ❑ το "o" δηλώνει το αντικείμενο (object) και
- ❑ το "R" δηλώνει τον παραλήπτη (recipient).



# Πλεονεκτήματα - Μειονεκτήματα

- ❖ Αυστηρά ορισμένη μεθοδολογία για την χωρίς αμφιβολίες κωδικοποίηση της σημασίας της φυσικής γλώσσας. Εξασφαλίζει ότι προτάσεις που έχουν το ίδιο νόημα θα αναπαρίστανται και συντακτικά με παρόμοιο τρόπο.
  - ❑ Αυτό έχει σαν αποτέλεσμα να αρκεί ένα ταίριασμα δύο γράφων εννοιολογικής εξάρτησης για να αποδειχθεί ότι δύο προτάσεις έχουν το ίδιο νόημα.
- ❖ Πλήρως δοκιμασμένο μοντέλο που μπορεί να εφαρμοστεί σε πολλές, όχι όμως γενικές, περιπτώσεις.
  - ❑ Η πολυπλοκότητα της φυσικής γλώσσας είναι τέτοια που είναι αδύνατο να κατασκευαστεί ένα πρόγραμμα που θα τη μετατρέπει αυτόματα σε μικρά κομμάτια αρχέγονης γνώσης.
- ❖ Συχνά, από μικρές φράσεις παράγονται τεράστιοι γράφοι που απαιτούν ανάλογους υπολογιστικούς πόρους για το χειρισμό τους.
- ❖ Υπάρχουν ορισμένες πολύ κοινές έννοιες της φυσικής γλώσσας που όμως είναι δύσκολο να κωδικοποιηθούν με αρχέγονα στοιχεία, όπως π.χ. οι έννοιες "λίγο", "χαμηλή", "βαρύς", κτλ.
  - ❑ Τέτοιες έννοιες είναι από τη φύση τους *σημασιολογικά ασαφείς* και ο χειρισμός τους γίνεται καλύτερα με άλλες μεθόδους όπως η *ασαφής λογική*.

## 5. Σενάρια (scripts)

- ❖ Ο ανθρώπινος νους οργανώνει τη γνώση σε δομές που αντιπροσωπεύουν στερεότυπες καταστάσεις (*stereotypical situations*).
  - ❑ π.χ. η φράση "πήγα στο εστιατόριο" έχει ως έμμεση συνέπεια το συμπέρασμα ορισμένων δραστηριοτήτων, χωρίς απαραίτητα αυτές να ειπωθούν ρητά, όπως: "κάθισα σε ένα τραπέζι, παρήγγειλα, έφαγα, πλήρωσα, κτλ"
- ❖ Τα σενάρια (προτάθηκαν από τον Schank) είναι μία στερεότυπη ακολουθία γεγονότων που περιγράφουν τυπικές καταστάσεις σε συγκεκριμένα πλαίσια δραστηριότητας
- ❖ Ένα σενάριο έχει τα εξής βασικά μέρη:
  - ❑ *Συνθήκες εισόδου (entry conditions)*: πρέπει να ικανοποιηθούν προτού συμβούν τα γεγονότα που περιέχει το σενάριο.
  - ❑ *Αποτελέσματα (results)*: πρέπει να προκύψουν μετά τα γεγονότα του σεναρίου.
  - ❑ *Σκηνικά (props)*: είναι αντικείμενα που συμμετέχουν στο σενάριο.
  - ❑ *Ρόλους (roles)*: είναι οντότητες που διαδραματίζουν κάποιο ρόλο στα γεγονότα του σεναρίου.
  - ❑ *Παραπομπές (track)*: άλλες τυπικές καταστάσεις που μοιράζονται πολλά από τα γεγονότα του συγκεκριμένου σεναρίου.
  - ❑ *Σκηνές (scenes)*: είναι η ακολουθία των γεγονότων (αναπαριστάνονται με βάση τη μέθοδο της εννοιολογικής εξάρτησης).

# Παράδειγμα Σεναρίου (Εστιατόριο)

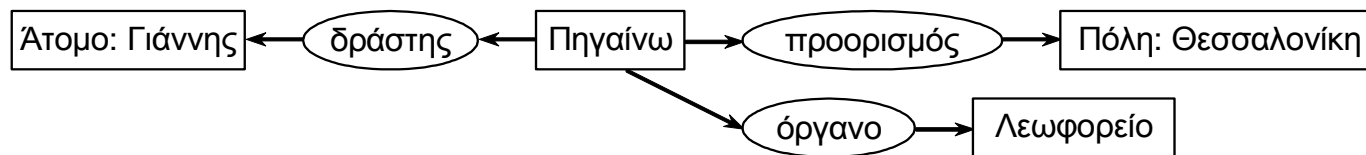
<p><b>Σενάριο:</b> Εστιατόριο  <b>Παραπομπή:</b> Καφετερία</p>	<p><b>ΣΚΗΝΗ 1: Είσοδος</b>          Πελάτης <b>PTRANS</b> Πελάτη στο εστιατόριο          Πελάτης <b>ATTENDS</b> μάτια στα τραπέζια          Πελάτης <b>MBUILD</b> που θα καθίσει          Πελάτης <b>PTRANS</b> στο τραπέζι</p>
<p><b>Ιδιότητες/Αντικείμενα:</b>          Τραπέζι          Μενού          Φαγητό          Λογαριασμός          Χρήματα</p>	<p><b>ΣΚΗΝΗ 2: Παραγγελία</b>          (Μενού στο Τραπέζι) (Σερβιτόρος φέρνει Μενού)          Πελάτης <b>PTRANS</b> Μενού στον Πελάτη          Πελάτης <b>MBUILD</b> επιλογή Φαγητού          Πελάτης <b>MTRANS</b> σήμα στο Σερβιτόρο          Σερβιτόρος <b>PTRANS</b> Σερβιτόρο στο Τραπέζι          Πελάτης <b>MTRANS</b> "Θέλω Φαγητό Χ" στο Σερβιτόρο          Σερβιτόρος <b>PTRANS</b> Σερβιτόρο στο Μάγειρα          Σερβιτόρος <b>ATRANS</b> Χ στο Μάγειρα          Μάγειρας <b>DO</b> (Σενάριο ετοιμασίας φαγητού Χ)</p>
<p><b>Ρόλοι:</b>          Πελάτης          Σερβιτόρος          Μάγειρας          Ταμίας          Ιδιοκτήτης</p>	<p><b>ΣΚΗΝΗ 3: Φαγητό</b>          Μάγειρας <b>ATRANS</b> Φαγητό στο Σερβιτόρο          Σερβιτόρος <b>ATRANS</b> Φαγητό στον Πελάτη          Πελάτης <b>INGEST</b> Φαγητό</p>
<p><b>Συνθήκες Εισόδου:</b>          Πελάτης πεινάει          Πελάτης έχει χρήματα</p>	<p>(Επιλογή: Σκηνή 2 για επόμενη παραγγελία ή αλλιώς Σκηνή 4)</p>
<p><b>Αποτελέσματα:</b>          Πελάτης έχει λιγότερα χρήματα          Ιδιοκτήτης έχει περισσότερα χρήματα          Πελάτης δεν πεινάει</p>	<p><b>ΣΚΗΝΗ 4: Έξοδος</b>          Σερβιτόρος <b>MOVE</b> (ετοιμάζει Λογαριασμό)          Σερβιτόρος <b>PTRANS</b> Σερβιτόρο στον Πελάτη          Σερβιτόρος <b>ATRANS</b> Λογαριασμό στον Πελάτη          Πελάτης <b>ATRANS</b> Φιλοδώρημα στον Σερβιτόρο          Πελάτης <b>PTRANS</b> Πελάτη στον Ταμία          Πελάτης <b>ATRANS</b> Χρήματα στον Ταμία          Πελάτης <b>PTRANS</b> Πελάτη έξω από το εστιατόριο</p>

- ❖ Υπάρχουν αντικείμενα και ρόλοι.
- ❖ Αποτελείται από τέσσερις σκηνές:
  - την είσοδο στο εστιατόριο
  - την παραγγελία
  - την κατανάλωση του φαγητού
  - την έξοδο από το εστιατόριο.
- ❖ Υπάρχουν οι συνθήκες εισόδου και τα αποτελέσματα του σεναρίου
- ❖ Σε διάλογο μεταξύ ενός υπολογιστή και ενός ανθρώπου, η αναφορά στο εστιατόριο από μέρος του ανθρώπου ενεργοποιεί το σενάριο του εστιατορίου και συνεπώς ο υπολογιστής μπορεί να κάνει εύλογες υποθέσεις.

- ❖ Επειδή το σενάριο εκφράζει τυπική κατάσταση, είναι πιθανό η πραγματικότητα να αποκλίνει από αυτό και να γίνουν λάθος εκτιμήσεις από τον υπολογιστή.

## 6. Εννοιολογικοί Γράφοι (conceptual graphs)

- ❖ Προτάθηκαν από τον Sowa ('76) και αποτελούν μία γλώσσα αναπαράστασης γνώσης με ρίζες στη μοντέρνα γλωσσολογία, την ψυχολογία και τη φιλοσοφία
- ❖ Συνοδεύονται από κατάλληλες δομές δεδομένων και τεχνικές για το χειρισμό τους.
- ❖ Ένας εννοιολογικός γράφος είναι ένας πεπερασμένος γράφος που αποτελείται από διασυνδεδεμένους **κόμβους εννοιών και σχέσεων**, που **εναλλάσσονται**.



- ❖ Στο γραφικό τρόπο απεικόνισης οι έννοιες απεικονίζονται με ορθογώνια, ενώ οι σχέσεις με ελλείψεις.
- ❖ Τα τόξα που συνδέουν τους κόμβους του γράφου έχουν κατεύθυνση, και λέμε ότι **ανήκουν στις σχέσεις και είναι προσκολλημένα στις έννοιες**.
- ❖ Ειδικές περιπτώσεις εννοιολογικών γράφων:
  - ❑ ο **κενός** (*blank*) εννοιολογικός γράφος: που δεν περιέχει έννοιες, σχέσεις και τόξα και πρακτικά δεν εκφράζει κάτι αλλά είναι πάντα αληθής
  - ❑ ο **μοναδιαίος** (*singleton*) εννοιολογικός γράφος που αποτελείται από μία μόνο έννοια
    - π.χ. [Αυτοκίνητο]: δηλώνει ότι υπάρχει κάποιο αυτοκίνητο

# Έννοιες

- ❖ Οι **έννοιες** (*concepts*) αποτελούν στιγμιότυπα των **τύπων-εννοιών** (*concept types*) και αποτελούνται από έναν **τύπο-έννοιας** (πρακτικά μία ετικέτα) συνοδευόμενο συνήθως από μία **αναφορά** (*referent*) σε συγκεκριμένη οντότητα. π.χ. [Άτομο: Γιάννης]

## Τύποι Εννοιών

- ❖ Ένας **τύπος-έννοιας** μπορεί να θεωρηθεί ότι είναι μία κλάση ή κατηγορία οντοτήτων, ενεργειών, ιδιοτήτων, αφηρημένων εννοιών, κτλ.
- ❖ Οι τύποι-εννοιών μπορεί να οργανωθούν σε μία **ιεραρχία** (*concept type hierarchy*) που αποτελεί μία **σχέση μερικής διάταξης** (*partial order*) στο σύνολο των τύπων-εννοιών.
  - Οντότητα  $\geq$  Ενέργεια, Άτομο, Δένδρο
  - Άτομο  $\geq$  Φοιτητής, Υπάλληλος
- ❖ Η ιεραρχία στους τύπους των εννοιών:
  - ❑ υποδηλώνει την ύπαρξη **κληρονομικότητας** στις ιδιότητές τους
  - ❑ ορίζει σχέσεις γενίκευσης και εξειδίκευσης
    - π.χ. η έννοια [Φοιτητής] είναι εξειδίκευση της έννοιας [Άτομο]
    - ισοδύναμα: η έννοια [Άτομο] είναι γενίκευση της έννοιας [Φοιτητής]

# Αναφορά

- ❖ Μια έννοια χωρίς αναφορά (π.χ. [Βιβλίο] ή [Βιβλίο: \*]) υποδηλώνει μία υπαρκτή αλλά απροσδιόριστη οντότητα (υπονοείται ο υπαρξιακός ποσοδείκτης "∃").
- ❖ Η αναφορά μπορεί να πάρει πολλές μορφές. Ενδεικτικά:
  - ❑ [Βιβλίο: ∇] – κάθε βιβλίο
  - ❑ [Βιβλίο: {"AI", "I Robot"}] – τα βιβλία "AI" και "I Robot"
  - ❑ [Βιβλίο: {\*}@3] – τρία βιβλία
  - ❑ [Ακέραιος:5] – ο αριθμός 5
  - ❑ [Πόλη:Αθήνα]
  - ❑ [Ηλικία:@<35,έτη>] – παρατίθεται η τιμή και η μονάδα μέτρησης
  - ❑ [Δένδρο:#9143] – συγκεκριμένο δένδρο που είναι καταγεγραμμένο με τον κωδικό 9143 σε κάποιο κατάλογο οντοτήτων
- ❖ Έννοιες συμφραζομένων: ειδικές περιπτώσεις εννοιών (*contextual concepts*) στις οποίες η αναφορά περιγράφεται με έναν ή και περισσότερους, μη-κενούς, εννοιολογικούς γράφους. Συνηθέστερες περιπτώσεις: Πρόταση και Κατάσταση
  - ❑ [ **Πρόταση**: [Άνθρωπος: ∇]→(χαρακτηριστικό)→[Θνητός] ] – Υπάρχει μια πρόταση, η 'Κάθε άνθρωπος είναι θνητός'
  - ❑ [ **Κατάσταση**: [Κρατούμενος:#17]→(μέσα)→[Κελί:#33] ] – Υπάρχει μία κατάσταση σύμφωνα με την οποία 'ο κρατούμενος #17 βρίσκεται στο κελί #33'.

# Συναναφορά

- ❖ Υπάρχουν περιπτώσεις στις οποίες δύο ή περισσότερες έννοιες αναφέρονται στην ίδια οντότητα. "Ένας πίθηκος τρώει ένα καρύδι με ένα κουτάλι φτιαγμένο από το κέλυφος του καρυδιού".

[Τρώω]→

(δράστης)→[Πίθηκος]

(αντικείμενο)→[Καρύδι: \*x]

(όργανο)→[Κουτάλι]→(υλικό)→[Κέλυφος]←(τμήμα)←[Καρύδι: ?x]

- ❖ Οι δύο έννοιες [Καρύδι] αναφέρονται στην ίδια (αλλά ακόμη απροσδιόριστη) οντότητα. Αυτή η υπονοούμενη σχέση ονομάζεται **συναναφορά** – *coreferent link*.
  - ❑ Η *συναναφορά* μπορεί να αναπαρασταθεί με μία μεταβλητή, στην πρώτη καταγραφή της οποίας μπαίνει το πρόθεμα "\*" για να οριστεί η μεταβλητή (*defining referent*) ενώ στις επόμενες μπαίνει το πρόθεμα "?" για να γίνει αναφορά σε συγκεκριμένη μεταβλητή (*bound referent*).
    - [ Κατάσταση: [Τίγρης: \*x]→(δράστης)→[Τρώει] [Τίγρης: ?x]→(ιδιότητα)→[Λαίμαργος] ]  
"Υπάρχει ένας τίγρης που τρώει και ο ίδιος τίγρης είναι λαίμαργος"
  - ❑ Δύο έννοιες μπορεί να συνδέονται με συναναφορά μόνο αν είναι **συμβατές**, δηλαδή:
    - είναι του ίδιου τύπου, ή
    - έχουν κάποια κοινή γενίκευση στην ιεραρχία των τύπων-εννοιών.

# Ορισμός νέων τύπων-εννοιών

❖ Οι τύποι-εννοιών μπορεί να είναι:

- ❑ **αρχέγονοι** (*primitive*): αξιωματικά ορισμένοι μέσω μιας ετικέτας π.χ. [Τίγρης]
- ❑ **ορισμένοι** (*defined*): μέσω μονοπαραμετρικών εκφράσεων-λ (*lambda expressions*)

```
type Τίγρης_Τσίρκου(*λ) is
  [Τίγρης:?λ]←(δράστης)←[Εμφανίζεται]→(τοποθεσία)→[Τσίρκο]
"Ο Τίγρης_Τσίρκου είναι ένας τίγρης που εμφανίζεται σε τσίρκο"
```

❖ Η ερμηνεία του νέου τύπου επηρεάζεται καταλυτικά από το ποια έννοια στον εννοιολογικό γράφο αποτελεί την παράμετρο της έκφρασης-λ.

```
type Τσίρκο_με_Τίγρη(*λ) is
  [Τίγρης]←(δράστης)←[Εμφανίζεται]→(τοποθεσία)→[Τσίρκο:?λ]
"Τσίρκο_με_Τίγρη είναι ένα τσίρκο στο οποίο εμφανίζεται ένας τίγρης"
```

❖ Ένας νέος τύπος-έννοιας θα πρέπει επιπλέον να εισαχθεί σε κατάλληλη θέση στην ιεραρχία τύπων-εννοιών.



# Εννοιολογικές Σχέσεις

- ❖ Οι *εννοιολογικές σχέσεις* (*conceptual relations*) συσχετίζουν τις έννοιες μεταξύ τους. Κάθε σχέση ορίζεται από έναν *τύπο-σχέσης* (*relation type*) ο οποίος περιλαμβάνει:
  - ❑ μία ετικέτα (label), που αποτελεί και το όνομα της σχέσης,
  - ❑ το σθένος (valence), έναν σταθερό θετικό αριθμό που υποδηλώνει τον αριθμό των εννοιών τις οποίες αυτή συσχετίζει (άρα και τον αριθμό των τόξων που ανήκουν στη σχέση)
  - ❑ την υπογραφή (signature), που ορίζει τους τύπους-εννοιών που συνδέει η σχέση
    - για μια n-αδική σχέση η υπογραφή συμβολίζεται με  $\langle t_1, t_2, \dots, t_n \rangle$ .
- ❖ Οι σχέσεις (όπως και οι τύποι-εννοιών) μπορεί να οργανωθούν σε ιεραρχία.
- ❖ Η υπογραφή μιας σχέσης θέτει περιορισμούς στο είδος των εννοιών που αυτή μπορεί να συσχετίσει.
- ❖ Παράδειγμα υπογραφής σχέσης:
  - ❑ ο τύπος-σχέσης **δράστης** έχει **σθένος 2** και υπογραφή **<Ενέργεια, Έμψυχο>**
  - ❑ το πρώτο τόξο της σχέσης πρέπει να προέρχεται από μία έννοια τύπου Ενέργεια ή οποιουδήποτε υποτύπου αυτής
  - ❑ το δεύτερο τόξο, ως τελευταίο, πρέπει να απομακρύνεται από τη σχέση προς μία έννοια τύπου Έμψυχο (ζωντανός οργανισμός) ή οποιουδήποτε υποτύπου αυτής.



# Ορισμός νέων Εννοιολογικών Σχέσεων

- ❖ Οι εννοιολογικές σχέσεις (όπως και οι τύποι-εννοιών) μπορεί να είναι:
  - ❑ αρχέγονες (*primitive*): αξιωματικά ορισμένες
  - ❑ ορισμένες (*defined*): μέσω εκφράσεων-λ (*lambda expressions*)
- ❖ Παράδειγμα ορισμού της σχέσης "πηγαίνω\_σε":

```
relation πηγαίνω_σε(*λ1,*λ2) is
  [Άτομο: ?λ1] ← (δράστης) ← [Πηγαίνω] → (προορισμός) → [Πόλη: ?λ2]
```

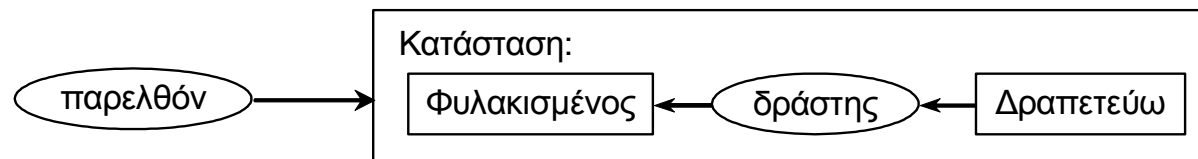
- ❑ οι μεταβλητές λ<sub>1</sub> και λ<sub>2</sub> υλοποιούν την παραμετροποίηση
  - ❑ η υπογραφή της σχέσης είναι <Εμψυχο, Τοποθεσία>.
  - ❑ Παράδειγμα χρήσης: [Άτομο: Φώτης] → (πηγαίνω\_σε) → [Πόλη: Φάρσαλα]
- ❖ Μία ορισμένη σχέση (*defined relation*) μέσα σε έναν σημασιολογικό γράφο μπορεί να αντικατασταθεί με τον ορισμό της (έκφραση-λ) και αντίστροφα.

```
[Άτομο: Ελένη] → (πηγαίνω_σε) → [Πόλη: Φάρσαλα]
[Άτομο: Ελένη] ← (δράστης) ← [Πηγαίνω] → (προορισμός) → [Πόλη: Φάρσαλα]
```

- ❖ Πώς προέκυψαν οι κατευθύνσεις των τόξων στην σχέση "πηγαίνω\_σε";
  - ❑ σε μία *n*-αδική σχέση τα πρώτα *n-1* τόξα βάσει της υπογραφής έχουν κατεύθυνση προς τη σχέση, ενώ το τελευταίο έχει αντίστροφη κατεύθυνση (απομακρύνεται από τη σχέση)

# Τόξα

- ❖ Τα **τόξα** (*arcs*) στους εννοιολογικούς γράφους συνδέουν τις σχέσεις με τις έννοιες.
- ❖ Η φορά των τόξων καθορίζεται από την υπογραφή της σχέσης στην οποία ανήκουν.
  - ❑ σε μία  $n$ -αδική σχέση τα πρώτα  $n-1$  τόξα βάσει της υπογραφής έχουν κατεύθυνση προς τη σχέση, ενώ το τελευταίο έχει αντίστροφη κατεύθυνση (απομακρύνεται από τη σχέση)
  - ❑ ο παραπάνω κανόνας ισχύει και για τις μοναδιαίες σχέσεις, αυτές δηλαδή που στην υπογραφή τους περιλαμβάνουν μία μόνο έννοια
    - η σχέση "παρελθόν" στο σχήμα είναι μοναδική (monadic) και υποδηλώνει ότι η δεδομένη κατάσταση συνέβη στο παρελθόν ("Ένας φυλακισμένος δραπέτευσε")



- ❖ Αρίθμηση Τόξων
  - ❑ Για εποπτικούς λόγους, τα τόξα που αντιστοιχούν στις πρώτες  $n-1$  έννοιες ( $n > 2$ ) της υπογραφής της σχέσης, μπορεί να αριθμηθούν με τους αριθμούς 1, 2, κτλ., βάσει της σειράς των εννοιών στην υπογραφή της σχέσης.
  - ❑ Το τόξο που αντιστοιχεί στην τελευταία έννοια, αυτό δηλαδή που απομακρύνεται από τη σχέση, δε χρειάζεται να αριθμηθεί.

# Εννοιολογικοί Γράφοι και Λογική

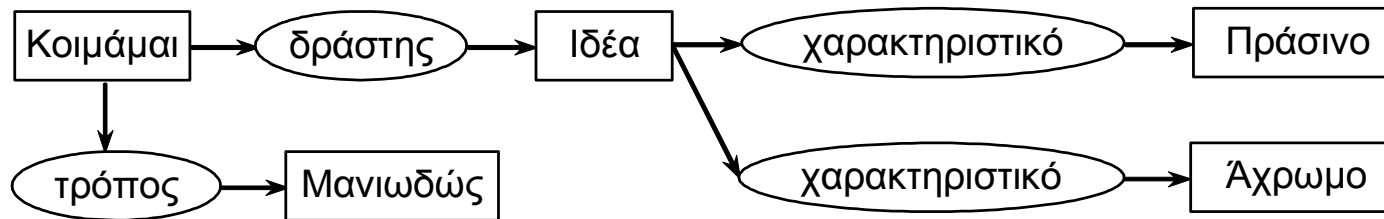
- ❖ Η **άρνηση** (NOT) ενός γράφου  $u$  μπορεί να κωδικοποιηθεί με τη χρήση της μοναδιαίας σχέσης "άρνηση" (negation) εφαρμοσμένης σε μία έννοια συμφραζομένων της οποίας αναφορά είναι ο γράφος  $u$ , δηλαδή: (άρνηση)  $\rightarrow$  [Κατάσταση:  $u$ ]
  - ❑ Πιο απλά:  $\neg [u]$
  - ❑ Παράδειγμα:  $\neg [ [Βάρκα:\#17] \rightarrow (\mu\acute{\epsilon}\sigma\alpha) \rightarrow [Λιμάνι] ]$
- ❖ Η **σύζευξη** (AND) δύο εννοιολογικών γράφων γίνεται με απλή παράθεσή τους στο ίδιο πλαίσιο συμφραζομένων:

[Πρόταση: [Άτομο: *x] $\rightarrow$ (ιδιότητα) $\rightarrow$ [Οπλισμένο] [Άτομο: ?x] $\rightarrow$ (ιδιότητα) $\rightarrow$ [Επικίνδυνο] ]
--

- ❑ Πιο απλά: η σύζευξη δύο γράφων  $u$  και  $v$  μπορεί να γραφεί και ως:  $[u \ v]$
- ❖ Η **διάζευξη** (OR) δύο εννοιολογικών γράφων γίνεται στη βάση του ότι στη λογική η έκφραση  $p \vee q$  είναι ισοδύναμη με την έκφραση  $\neg(\neg p \wedge \neg q)$ .
  - ❑ Βάσει των εκφράσεων για άρνηση και σύζευξη, η έκφραση  $u \vee v$  κωδικοποιείται ως:  
 $\neg[ \neg[u] \neg[v] ]$  (απλουστευμένη αναπαράσταση)
- ❖ **Συνεπαγωγή**: η έκφραση  $p \rightarrow q$  είναι ισοδύναμη με την  $\neg(p \wedge \neg q)$ . Οπότε για του γράφους  $u$  και  $v$  θα είναι:  $\neg[ u \neg[v] ]$  (απλουστευμένη αναπαράσταση)
  - ❑ Σε μορφή κανόνα: if  $p$  and  $q$  then  $w$   $\neg[ p \ q \ \neg[w] ]$  ή  $[If: p \ q \ [Then: w] ]$

# Πώς συνδυάζονται όλα τα προηγούμενα;

- ❖ Δεν μπορεί όλοι οι αυθαίρετοι συνδυασμοί εννοιών και σχέσεων να παράγουν ορθά δομημένους (well-formed) εννοιολογικούς γράφους. Παράδειγμα:



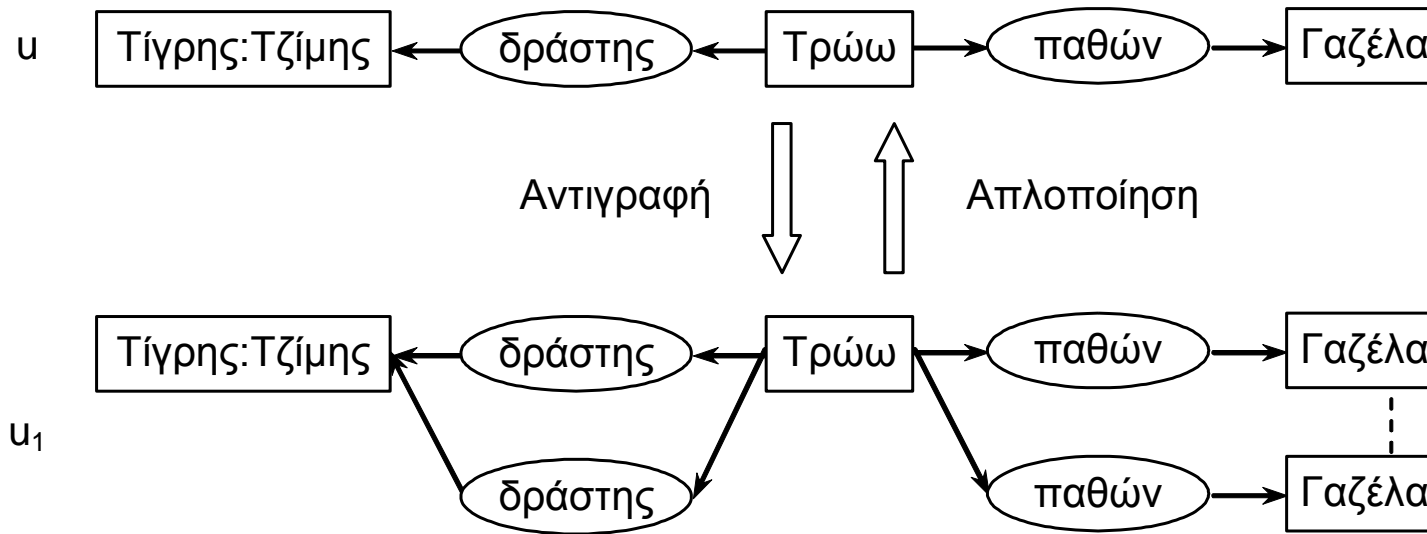
"Άχρωμες πράσινες ιδέες κοιμούνται μανιωδώς" (Chomsky)

- ❑ η έννοια [Ιδέα] δεν είναι υποτύπος της έννοιας [Εμψυχο] που απαιτεί η σχέση (δράστης)
- ❑ η έννοια [Ιδέα] δεν είναι υποτύπος της έννοιας [Αντικείμενο] που απαιτεί η σχέση (χαρακτηριστικό)
- ❑ η ταυτόχρονη παρουσία των εννοιών [Πράσινο] και [Άχρωμο] παραβιάζει τους κανόνες της κοινής λογικής
- ❖ Απαιτούνται μηχανισμοί που θα:
  - ❑ εξασφαλίζουν τη δημιουργία ορθά δομημένων εννοιολογικών γράφων
    - Κανόνες Ορθής Διαμόρφωσης
  - ❑ επιτρέπουν τη συλλογιστική με αυτούς
    - Κανόνες Εξαγωγής Συμπερασμάτων

# Κανόνες Ορθής Διαμόρφωσης (1/2)

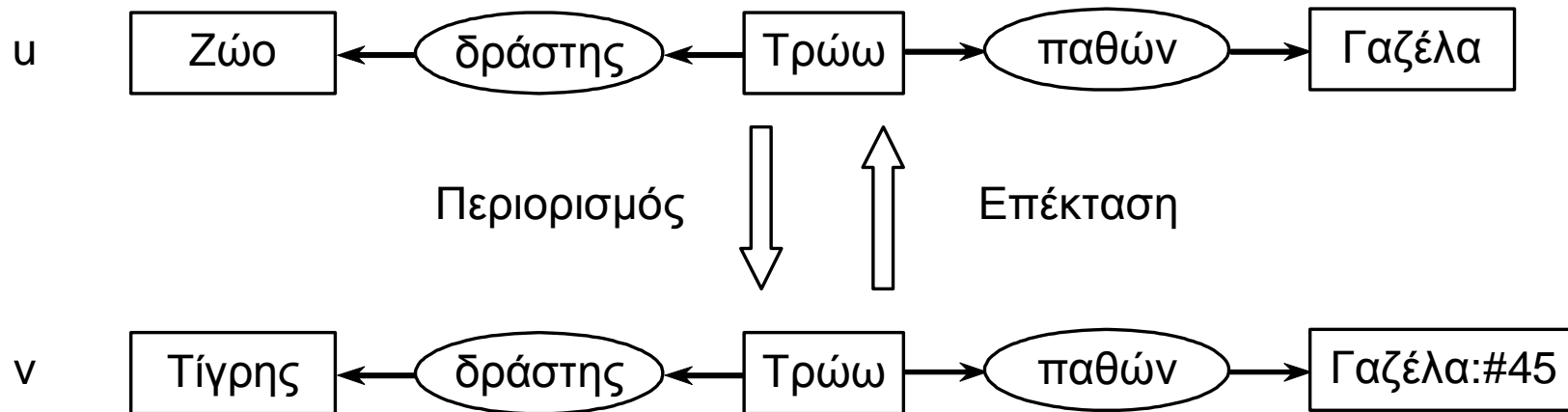
- ❖ Οι κανόνες ορθής διαμόρφωσης (*canonical formation rules*) που αποτελούν ένα είδος γραμματικής για ορθά δομημένους γράφους.
- ❖ Έξι κανόνες που παράγουν έναν ορθά δομημένο γράφο  $w$  επιδρώντας στους επίσης ορθά δομημένους γράφους  $u$  και  $v$  (μπορεί βέβαια να είναι  $u=v$ ).
- ❖ Μπορεί να καταταχτούν στις ακόλουθες 3 κατηγορίες:
  - ❑ **Κανόνες εξειδίκευσης (περιορισμός και συνένωση):** εξειδικεύουν το γράφο στον οποίο δρουν, δηλαδή μειώνουν τις περιπτώσεις στις οποίες ο γράφος είναι αληθής.
  - ❑ **Κανόνες γενίκευσης (επέκταση και διαχωρισμός):** γενικεύουν το γράφο στον οποίο δρουν, δηλαδή αυξάνουν τις περιπτώσεις στις οποίες ο γράφος είναι αληθής.
  - ❑ **Κανόνες ισοδυναμίας (αντιγραφή και απλοποίηση):** μεταβάλλουν τη δομή του γράφου στον οποίο δρουν, αφήνοντάς τον όμως λογικά ισοδύναμο.
- ❖ Για κάθε κανόνα υπάρχει ο αντίστροφός του (που επιφέρει δηλαδή το αντίστροφο αποτέλεσμα)
  - ❑ Αντιγραφή - Απλοποίηση
  - ❑ Περιορισμός - Επέκταση
  - ❑ Συνένωση - Διαχωρισμός

# Αντιγραφή - Απλοποίηση



- ❖ Η έκφραση *κατηγορηματικού λογισμού με τύπους* που αντιστοιχεί στο γράφο  $u$  είναι:  
 $(\exists x:Τίγρης) (\exists y:Τρώω) (\exists z:Γαζέλα)$   
 $( name(x, 'Τζίμης') \wedge δράστης(y,x) \wedge τρόπος(y,z) )$
- ❖ Η έκφραση που αντιστοιχεί στο γράφο  $v$  είναι η:  
 $(\exists x: Τίγρης) (\exists y:Τρώω) (\exists z:Γαζέλα) (\exists w:Γαζέλα) ( name(x, 'Τζίμης') \wedge$   
 $δράστης(y,x) \wedge δράστης(y,x) \wedge τρόπος(y,z) \wedge τρόπος(y,w) \wedge z=w )$
- ❖ Πρόκειται για **κανόνες ισοδυναμίας**: οι δύο εκφράσεις είναι αληθείς ή ψευδείς υπό τις ίδιες ακριβώς συνθήκες.
- ❖ Η διακεκομμένη γραμμή αναπαριστά την **συναναφορά** (coreferent link).

# Περιορισμός και Επέκταση



## ❖ Περιορισμός μέσω του τύπου της έννοιας (*restriction by type*)

- ❑ Η έννοια [Ζώο] στο γράφο  $u$  μπορεί να αντικατασταθεί με την έννοια [Τίγρης] καθώς  $\text{Τίγρης} \leq \text{Ζώο}$ .

## ❖ Περιορισμός μέσω της αναφοράς (*restriction by referent*)

- ❑ Όταν μία έννοια  $c$  δεν αναφέρεται σε συγκεκριμένη οντότητα (όπως η έννοια [Γαζέλα] του γράφου  $u$ ) τότε αυτή μπορεί να αποκτήσει συγκεκριμένη αναφορά (π.χ. [Γαζέλα:#45])

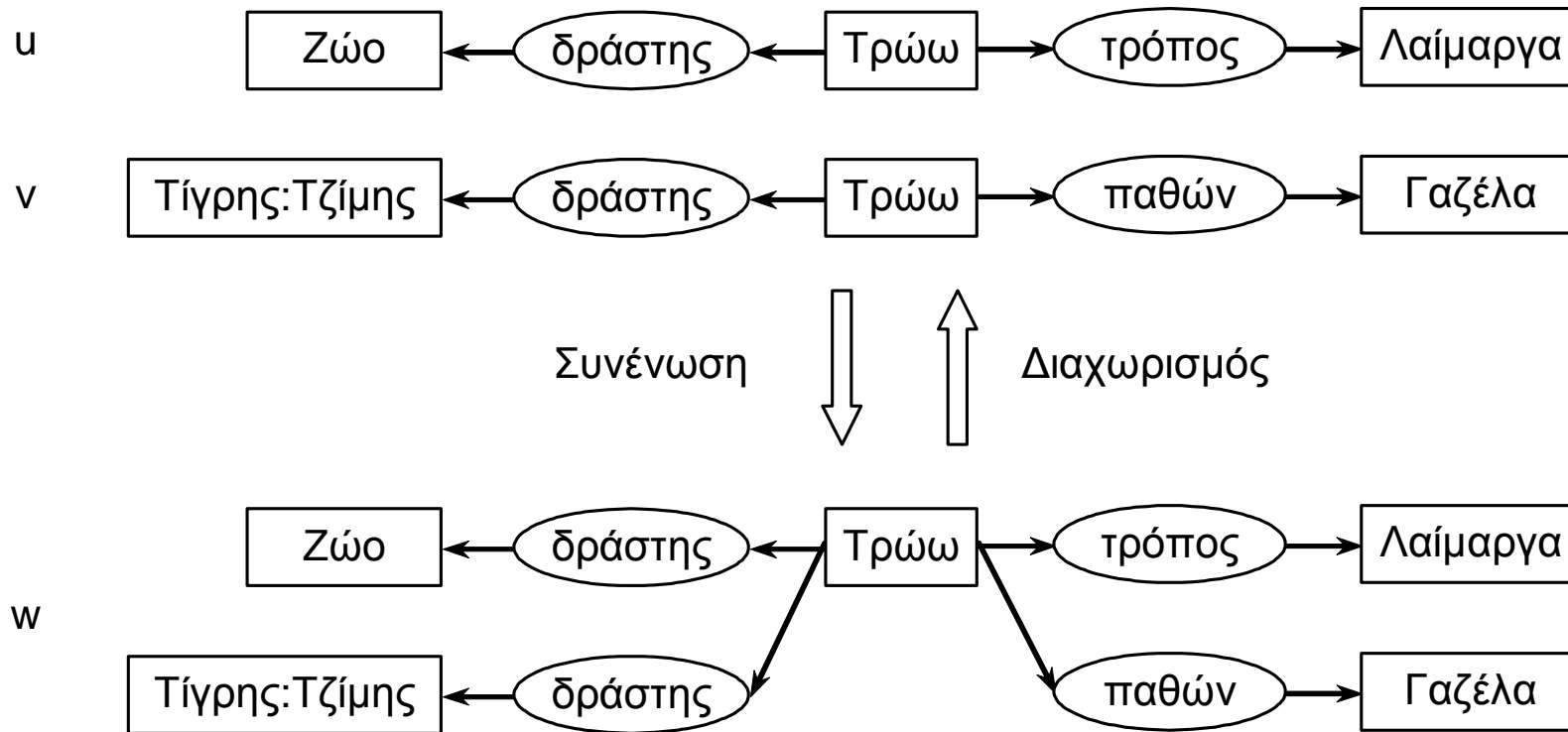
## ❖ Η επέκταση είναι ένας κανόνας γενίκευσης (*generalization rule*).

- ❑ αν "ένας τίγρης τρώει τη γαζέλα #45" (γράφος  $v$ ) τότε οπωσδήποτε "κάποιο ζώο τρώει κάποια γαζέλα" (γράφος  $u$ ).

## ❖ Ο περιορισμός είναι κανόνας εξειδίκευσης (*specialization rule*).



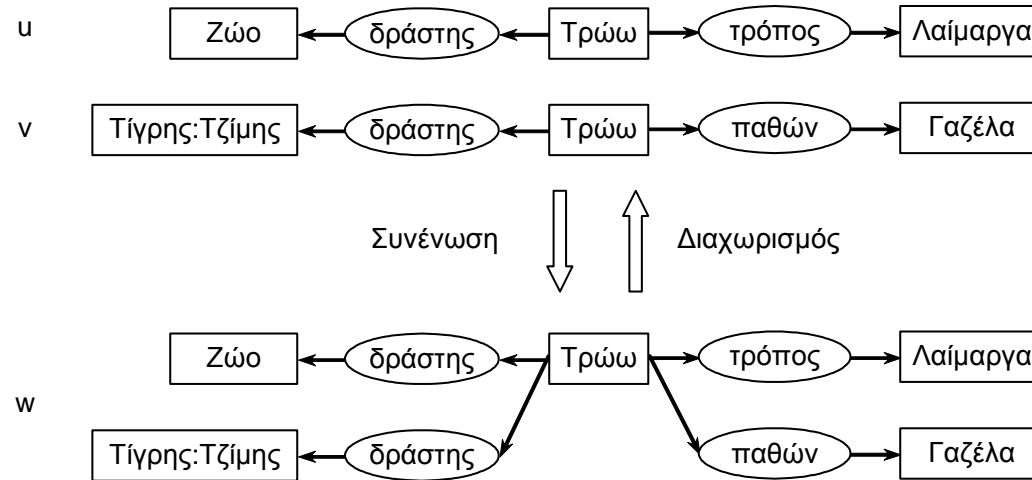
# Συνένωση και Διαχωρισμός



- ❖ Εάν μία έννοια  $c$  ενός γράφου  $u$  είναι όμοια με μία άλλη έννοια  $d$  ενός γράφου  $v$  τότε μπορεί να προκύψει ένας νέος γράφος  $w$  με βάση τον  $u$ , συνενώνοντας στην έννοια  $c$  όλα τα τόξα του  $v$  που ήταν προσκολλημένα στην έννοια  $d$  (η οποία απαλείφεται).
- ❖ Η συνένωση είναι κανόνας εξειδίκευσης (*specialization rule*).
- ❖ Ο διαχωρισμός είναι κανόνας γενίκευσης (*generalization rule*).

## Κανόνες Ορθής Διαμόρφωσης (2/2)

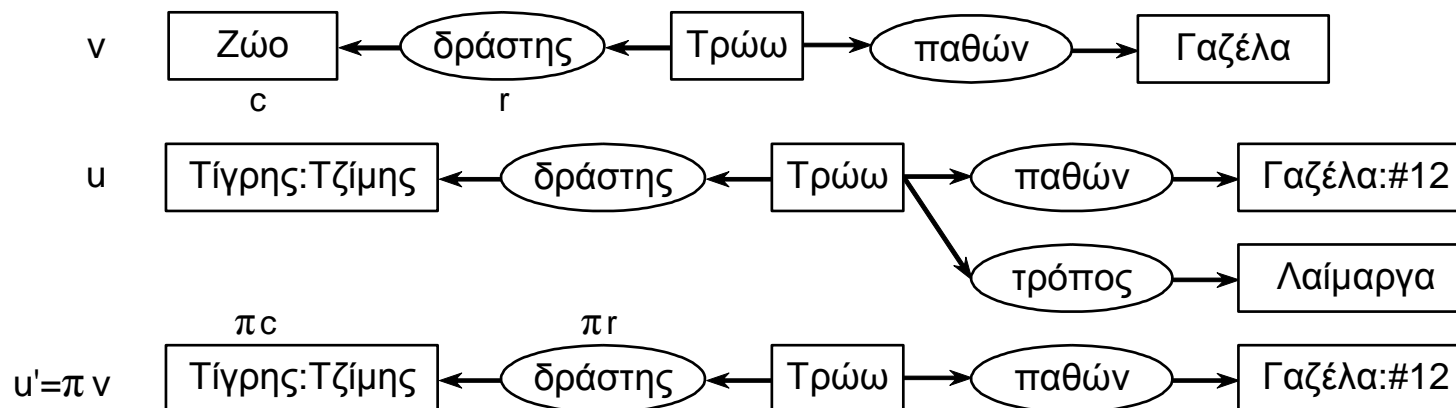
- ❖ Οι κανόνες ορθής διαμόρφωσης δεν αποτελούν κανόνες εξαγωγής συμπερασμάτων.
- ❖ Παράδειγμα:



- ❑ Στη συνένωση δύο γράφων  $u$  και  $v$  που ο καθένας τους είναι αληθής, ο προκύπτων γράφος  $w$  είναι εξασφαλισμένα ορθός δομικά.
- ❑ Κανείς όμως δεν εγγυάται για το αληθές ή ψευδές του νοήματος που κωδικοποιεί:
  - Η συνένωση είναι κανόνας εξειδίκευσης, άρα το νόημα του παραγόμενου γράφου μπορεί και να μην είναι αληθές.
  - Οι κανόνες εξειδίκευσης δε διατηρούν την τιμή αληθείας των προτάσεων στις οποίες δρουν.
  - Δεν είναι σίγουρο ότι ο τίγρης 'Τζίμης' είναι το ζώο που τρώει λαίμαργα!
- ❖ Άρα οι κανόνες ορθής διαμόρφωσης δεν επαρκούν για συλλογιστική με ενν. γράφους.

# Προβολή

- ❖ Η χρήση των κανόνων ορθής διαμόρφωσης σε κάποιο εννοιολογικό γράφο  $g$  παράγει, εν γένει, γενικεύσεις (generalizations) και εξειδικεύσεις (specializations) του  $g$ .
  - ❑ η γενίκευση ενός γράφου που είναι αληθής οδηγεί σε γράφο που επίσης είναι αληθής
  - ❑ η εξειδίκευση ενός γράφου που είναι ψευδής οδηγεί σε γράφο που επίσης είναι ψευδής
- ❖ Αν ένας γράφος  $u$  είναι εξειδίκευση ενός γράφου  $v$ , τότε θα πρέπει να υπάρχει κάποιος γράφος  $u'$  ενσωματωμένος στον  $u$  ο οποίος αντιστοιχεί στον γράφο  $v$ . Ο γράφος  $u'$  ονομάζεται **προβολή** του  $v$  στον  $u$  και συμβολίζεται με  $\pi v$ , όπου  $\pi$  είναι ο τελεστής της προβολής. Η όλη διαδικασία ονομάζεται επίσης "προβολή".



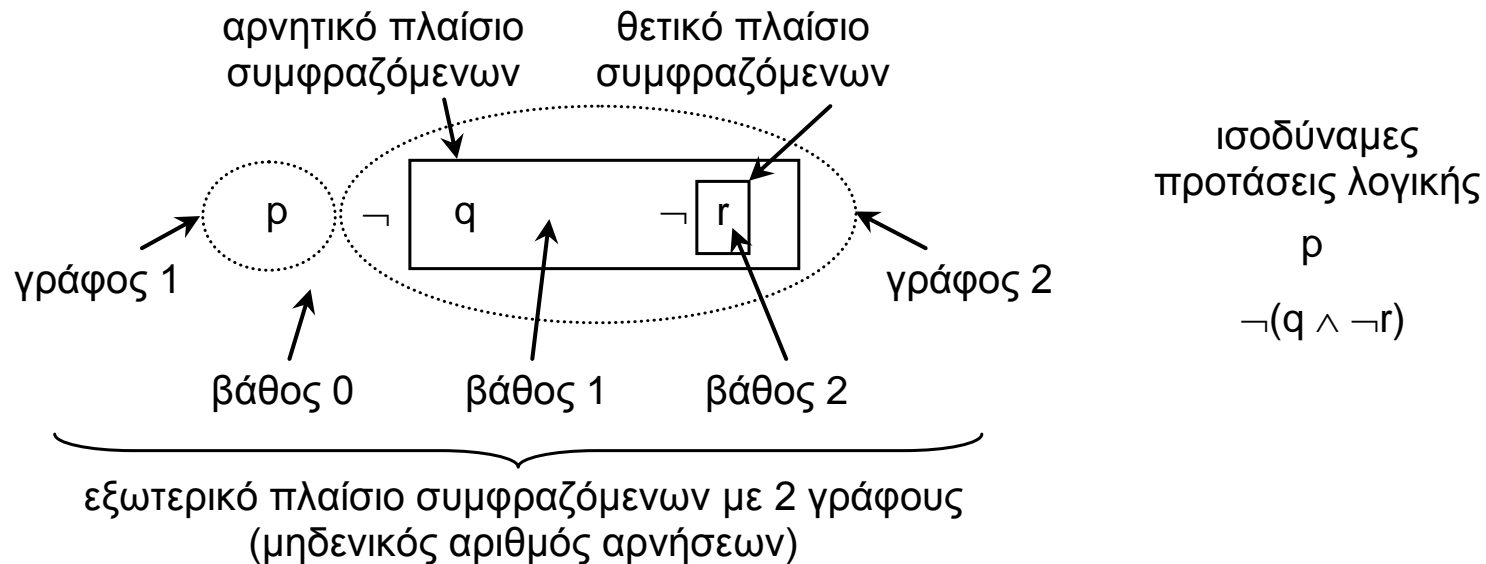
- ❖ Οι έννοιες και οι σχέσεις που συνθέτουν την προβολή είναι εξειδικεύσεις αντίστοιχων στοιχείων του γράφου που "προβάλλεται" (δηλαδή του  $v$ ) π.χ.  $\pi c \leq c$  και  $\pi r \leq r$

# Παραγωγή Γνώσης

- ❖ Η διαδικασία **προβολής** αποτελεί μηχανισμό ενοποίησης (unification) σε εννοιολογικούς γράφους.
- ❖ Έστω μια βάση γνώσης που περιέχει:
  - ❑ κανόνες if-then που έχουν εννοιολογικούς γράφους ως προϋπόθεση και συμπέρασμα
  - ❑ στατική γνώση κωδικοποιημένη σε εννοιολογικούς γράφους
- ❖ Είναι δυνατό να γίνει ενοποίηση κάποιου γράφου με την προϋπόθεση του κανόνα, οπότε συμπεραίνεται ο γράφος-συμπέρασμα αυτού.
  - ❑ παράγεται δηλαδή νέα γνώση
- ❖ Σε απλούς εννοιολογικούς γράφους (χωρίς έννοιες συμφραζομένων, αρνήσεις και συναναφορές) ο μηχανισμός της προβολής είναι sound και complete.

# Κανόνες Εξαγωγής Συμπερασμάτων (1/3)

- ❖ Αποτελούν προσαρμογή (Sowa) αντίστοιχου συστήματος κανόνων κατηγορηματικού λογισμού που είχε προταθεί από τον Peirce στις αρχές του 19<sup>ου</sup> αιώνα.
  - ❑ Οι κανόνες του Peirce αφορούσαν τους *υπαρξιακούς γράφους* (*existential graphs*), των οποίων οι εννοιολογικοί γράφοι του Sowa αποτελούν μια εμπλουτισμένη εκδοχή.
- ❖ **Θετικό** και **αρνητικό** πλαίσιο συμφραζόμενων.



- ❑ Ένα πλαίσιο συμφραζόμενων είναι **θετικό** (*positive context*) εάν βρίσκεται φωλιασμένο σε ζυγό αριθμό αρνήσεων (πιθανώς και σε μηδενικό αριθμό αρνήσεων).
- ❑ Ένα πλαίσιο συμφραζόμενων είναι **αρνητικό** (*negative context*) εάν βρίσκεται φωλιασμένο σε μονό αριθμό αρνήσεων.

## Κανόνες Εξαγωγής Συμπερασμάτων (2/3)

- ❖ Τα πλαίσια συμφραζόμενων επηρεάζουν τους διάφορους κανόνες:
  - ❑ Οι κανόνες ισοδυναμίας παραμένουν ως έχουν σε οποιοδήποτε πλαίσιο συμφραζόμενων.
  - ❑ Οι κανόνες εξειδίκευσης σε αρνητικό πλαίσιο συμφραζόμενων μετατρέπονται σε κανόνες γενίκευσης, ενώ σε θετικό πλαίσιο συμφραζόμενων παραμένουν κανόνες εξειδίκευσης.
  - ❑ Οι κανόνες γενίκευσης σε αρνητικό πλαίσιο συμφραζόμενων μετατρέπονται σε κανόνες εξειδίκευσης, ενώ σε θετικό πλαίσιο συμφραζόμενων παραμένουν κανόνες γενίκευσης.
  - ❑ Η διπλή άρνηση (δύο διαδοχικά αρνητικά πλαίσια συμφραζόμενων μεταξύ των οποίων δεν παρεμβάλλεται κάτι άλλο), αποτελεί κανόνα ισοδυναμίας.
    - Μπορεί να προστεθεί ή να αφαιρεθεί γύρω από οποιοδήποτε γράφο (ή τμήμα γράφου).

## Κανόνες Εξαγωγής Συμπερασμάτων (3/3)

- ❖ Οι κανόνες συλλογιστικής του Pierce όπως υιοθετήθηκαν από τον Sowa ώστε να υλοποιούν κατηγορηματικό λογισμό 1<sup>ης</sup> τάξης για εννοιολογικούς γράφους είναι:
  - ❑ *Διαγραφή (erasure)*: Σε ένα θετικό πλαίσιο συμφραζόμενων, οποιοσδήποτε γράφος (ή τμήμα γράφου)  $u$  μπορεί να αντικατασταθεί με μία γενίκευση του  $u$ . Ειδικότερα, ο γράφος  $u$  μπορεί να διαγραφεί καθώς αυτό ισοδυναμεί με αντικατάστασή του με τον κενό γράφο.
  - ❑ *Εισαγωγή (insertion)*: Σε ένα αρνητικό πλαίσιο συμφραζόμενων, οποιοσδήποτε γράφος (ή τμήμα γράφου)  $u$  μπορεί να αντικατασταθεί με μία εξειδίκευση του  $u$ . Ειδικότερα, μπορεί να εισαχθεί οποιοσδήποτε γράφος  $u$  καθώς αυτός μπορεί να θεωρηθεί εξειδίκευση του κενού γράφου.
  - ❑ *Επανάληψη (iteration)*: Εάν ένας γράφος (ή τμήμα γράφου)  $u$  βρίσκεται εντός ενός πλαισίου συμφραζόμενων  $C$ , τότε ένα αντίγραφο του  $u$  μπορεί να εισαχθεί εντός του  $C$  ή εντός οποιουδήποτε άλλου πλαισίου συμφραζόμενων βρίσκεται εντός του  $C$ .
  - ❑ *Απαλοιφή επανάληψης (deiteration)*: Οποιοσδήποτε γράφος (ή τμήμα γράφου)  $u$  που μπορεί να θεωρηθεί ότι έχει προέλθει με εφαρμογή του κανόνα της επανάληψης, μπορεί να διαγραφεί.
  - ❑ *Ισοδυναμία (equivalence)*: Οποιοσδήποτε κανόνας ισοδυναμίας (αντιγραφή, απλοποίηση, διπλή άρνηση) μπορεί να εφαρμοστεί πάνω σε οποιοδήποτε γράφο (ή τμήμα γράφου)  $u$ , σε οποιοδήποτε πλαίσιο συμφραζόμενων.
- ❖ Όλοι οι παραπάνω κανόνες μπορεί να αναπαρασταθούν γραφικά.

# Παράδειγμα Συλλογιστικής

❖ Έστω η ακόλουθη γνώση:

□ *Κάποιος είναι Έλληνας πολίτης αν και μόνο αν είναι αληθής οποιαδήποτε από τις ακόλουθες συνθήκες:*

1) *Γεννήθηκε στην Ελλάδα.*

2) *Ένας από τους γονείς του είναι Έλληνας πολίτης.*

3) *Έχει πολιτογραφηθεί ως Έλληνας.*

❖ Επειδή η διατύπωση είναι σε μορφή "αν και μόνο αν" κάθε έκφραση θα εμφανιστεί ως προϋπόθεση κάποιας συνεπαγωγής αλλά και ως επακόλουθο κάποιας άλλης.

1. "Εάν γεννήθηκε στη Ελλάδα, τότε είναι Έλληνας πολίτης."

$\neg[ [\text{Άτομο: } *x] \leftarrow (\text{αντικείμενο}) \leftarrow [\text{Γεννήθηκε}] \rightarrow (\text{τόπος}) \rightarrow [\text{Χώρα: 'Ελλάδα'} ]$

$\neg[ [\text{Πολίτης: } ?x] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'} ] ]$

]

2. "Εάν είναι παιδί Έλληνα πολίτη, τότε είναι Έλληνας πολίτης"

$\neg[ [\text{Άτομο: } *x] \leftarrow (\text{παιδί}) \leftarrow [\text{Πολίτης}] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'} ]$

$\neg[ [\text{Πολίτης: } ?x] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'} ] ]$

]

3. "Εάν πολιτογραφήθηκε στην Ελλάδα, τότε είναι Έλληνας πολίτης"

$\neg[ [\text{Άτομο: } *x] \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα: 'Ελλάδα'} ]$

$\neg[ [\text{Πολίτης: } ?x] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'} ] ]$

]



4. "Είναι Έλληνας πολίτης εάν: είτε γεννήθηκε στην Ελλάδα, ή είναι παιδί Έλληνα πολίτη, ή πολιτογραφήθηκε στην Ελλάδα"

$\neg[ \text{[Πολίτης: *x]} \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'}] ]$

$\neg[ \text{[Άτομο: ?x]} \leftarrow (\text{αντικείμενο}) \leftarrow [\text{Γεννήθηκε}] \rightarrow (\text{τόπος}) \rightarrow [\text{Χώρα: 'Ελλάδα'}] ] ]$

$\neg[ \text{[Άτομο: ?x]} \leftarrow (\text{παιδί}) \leftarrow [\text{Πολίτης}] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'}] ] ]$

$\neg[ \text{[Άτομο: ?x]} \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα: 'Ελλάδα'}] ] ]$

]

❖ Έστω ότι στους παραπάνω τέσσερις γράφους προστίθεται και ο γράφος:

5.  $\text{[Άτομο: 'John']} \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα: 'Ελλάδα'}]$

❖ Μέσω του κανόνα της επανάληψης (iteration) ένα αντίγραφο του γράφου (5) μπορεί να εισαχθεί στο εξωτερικό αρνητικό πλαίσιο συμφραζόμενων του γράφου (3). Προκύπτει έτσι ο γράφος (6):

6.  $\neg[ \text{[Άτομο: 'John']} \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα: 'Ελλάδα'}]$

$\text{[Άτομο: *x]} \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα: 'Ελλάδα'}]$

$\neg[ \text{[Πολίτης: ?x]} \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'}] ] ]$

]

❖ Στο γράφο (6), οι δύο πρώτοι εσωτερικοί γράφοι (βάθος 1), μπορεί να συνενωθούν

7.  $\neg[ \text{[Άτομο: 'John']} \leftarrow (\text{αποδέκτης}) \leftarrow [\text{Πολιτογράφηση}] \rightarrow (\text{τοποθεσία}) \rightarrow [\text{Χώρα: 'Ελλάδα'}]$

$\neg[ \text{[Πολίτης: 'John']} \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'}] ] ]$

]



- ❖ Μέσω του κανόνα της απαλοιφής επανάληψης (deiteration) ο πρώτος εσωτερικός γράφος στο γράφο (7), μπορεί να απαλειφθεί καθώς είναι ακριβές αντίγραφο του γράφου (5) που εξακολουθεί να υπάρχει στο αρχικό σύνολο γράφων.

8.  $\neg[ \neg[ [\text{Πολίτης: 'John'}] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'}] ] ]$

- ❖ Η διπλή άρνηση που ορίζουν τα δύο διαδοχικά φωλιασμένα αρνητικά πλαίσια συμφραζόμενων στο γράφο (8), μπορεί να απαλειφθεί, οπότε προκύπτει το συμπέρασμα ότι "Ο John είναι Έλληνας πολίτης" (γράφος (9)).

9.  $[\text{Πολίτης: 'John'}] \leftarrow (\text{μέλος}) \leftarrow [\text{Χώρα: 'Ελλάδα'}]$

# Ασκήσεις

- ❖ 7.5 Έχω τους ορισμούς των παρακάτω 3 πλαισίων-κλάσεων και 1 πλαισίου-στιγμιότυπου:

Πλαίσιο <b>Mammal</b> ako: <b>Animal</b> has_head: <b>yes</b> warm_blooded: <b>yes</b> eats: <b>everything</b>	Πλαίσιο <b>Carnivore</b> ako: <b>Animal</b> eats: <b>meat</b>	Πλαίσιο <b>Feline</b> ako: <b>Carnivore, Mammal</b> sharp_claws: <b>yes</b>
--	---	---

Στιγμιότυπο <b>Silvestre</b> isa: <b>Carnivore</b> likes: <b>birds</b> eats: <b>birds</b>
--

- ❑ Ποια είναι η τιμή της σχισμής **eats** για το στιγμιότυπο **Silvestre**; Γιατί;



❖ 7.6. Έχω τους ορισμούς των παρακάτω 2 πλαισίων-κλάσεων και 1 πλαισίου-στιγμιότυπου:

Πλαίσιο <b>Mammal</b> ako: <b>Animal</b> has_head: <b>yes</b> warm_blooded: <b>yes</b> eats: <b>everything</b>	Πλαίσιο <b>Carnivore</b> ako: <b>Mammal</b> eats: <b>meat</b>	Στιγμιότυπο <b>Silvestre</b> isa: <b>Carnivore</b> likes: <b>birds</b>
--	---	--

- ❑ Ποια είναι η τιμή της σχισμής **eats** για το στιγμιότυπο **Silvestre**;



- ❖ 7.7 Έχω τους ορισμούς των παρακάτω 2 πλαισίων-κλάσεων και 1 πλαισίου-στιγμιότυπου:

Πλαίσιο Mammal ako: Animal has_head: yes warm_blooded: yes eats: everything	Πλαίσιο Cat ako: Mammal has_claws: yes	Στιγμιότυπο Silvestre isa: Carnivore likes: birds
---	--	--

- Ποια είναι η τιμή της σχισμής **eats** για το στιγμιότυπο **Silvestre**;

# Συστήματα Κανόνων

# Αναπαράσταση με Κανόνες

- ❖ Οι κανόνες είναι από τις πιο προσφιλείς μεθόδους αναπαράστασης γνώσης
  - ❑ Ο πρακτικότερος τρόπος αναπαράστασης για την εξαγωγή συμπερασμάτων.
- ❖ Συστήματα κανόνων αποτελούν τη βάση πολλών έμπειρων συστημάτων (expert systems) και συστημάτων γνώσης (knowledge systems), γενικότερα.
- ❖ Γενικά Πλεονεκτήματα:
  - ❑ Η γνώση αναπαριστάται με τρόπο που πλησιάζει την ανθρώπινη γνώση για τις περισσότερες δραστηριότητες που απαιτούν νοημοσύνη
  - ❑ Η εξαγωγή συμπερασμάτων γίνεται με εύκολο τρόπο (επάρκεια συνεπαγωγών).
- ❖ Συγκεκριμένα Πλεονεκτήματα:
  - ❑ Κάθε κανόνας ορίζει ένα μικρό και (σχεδόν) ανεξάρτητο τμήμα της γνώσης για ένα πρόβλημα (*modularity*).
  - ❑ Νέοι κανόνες μπορούν να προστεθούν σε ένα σύνολο κανόνων (σχεδόν) ανεξάρτητα από άλλους υπάρχοντες κανόνες (*incrementability*).
  - ❑ Κανόνες που ήδη υπάρχουν σε ένα σύνολο κανόνων μπορούν να αλλάξουν (σχεδόν) ανεξάρτητα από άλλους κανόνες (*modifiability*).

# Είδη Κανόνων

Μορφές Κανόνων	Εκφράζει	Επεξήγηση
IF συνθήκες THEN συμπέρασμα Συνεπαγωγικός (Deductive) κανόνας	Δηλωτική γνώση	Αν οι συνθήκες αληθεύουν τότε αληθεύει και το συμπέρασμα
IF συνθήκες THEN ενέργειες Κανόνας Παραγωγής (Production)	Διαδικαστική γνώση	Αν οι συνθήκες αληθεύουν τότε εκτέλεσε τις ενέργειες
ON συμβάν IF συνθήκες THEN ενέργειες Ενεργός (active) κανόνας	Διαδικαστική γνώση	Όταν συμβεί το γεγονός Αν οι συνθήκες αληθεύουν τότε εκτέλεσε τις ενέργειες

- ❖ Οι **συνθήκες** (*conditions*) είναι μία ακολουθία από κατηγορήματα τα οποία συνδέονται μεταξύ τους με τους λογικούς τελεστές AND και/ή OR.
  - ❑ Αναφέρονται και ως προϋποθέσεις (premises) ή αριστερό μέρος του κανόνα (left hand side - LHS).
- ❖ Το **συμπέρασμα** (*conclusion*) είναι ένα κατηγορημα.
- ❖ Οι **ενέργειες** (*actions*) είναι μία σειρά από εντολές που πρέπει να εκτελεστούν.
  - ❑ Οι ενέργειες ή το συμπέρασμα αναφέρονται και ως επακόλουθα (consequent) ή δεξιό μέρος του κανόνα (right hand side - RHS).



# Συστήματα Κανόνων

- ❖ Συστήματα εξαγωγής συμπερασμάτων (deduction systems): π.χ. Prolog
  - ❑ Γνώση που δηλώνει μία αλήθεια για τον κόσμο του προβλήματος, αλλά δεν αναφέρει ρητά πότε και πώς εφαρμόζεται.
- ❖ Συστήματα παραγωγής (production systems): π.χ. CLIPS, Flex
  - ❑ Γνώση για το ποιες συγκεκριμένες ενέργειες πρέπει να εκτελεστούν δεδομένης μιας κατάστασης.
  - ❑ Μία ενέργεια που εκτελείται επιφέρει αποτελέσματα που δεν είναι αναστρέψιμα μέσω οπισθοδρόμησης, παρά μόνο μέσω ανάστροφων ενεργειών.
- ❖ Ενεργά Συστήματα ((re-)active systems, active database systems): π.χ. Oracle Triggers, Δαίμονες πλαισίων Flex

# Ενεργοί Κανόνες

- ❖ Οι κανόνες παραγωγής δηλώνουν διαδικαστική γνώση
  - ❑ Δεν είναι σαφώς ορισμένο πότε ακριβώς εκτελούνται οι ενέργειές τους
  - ❑ Αναφέρεται με ασάφεια πώς οι κανόνες εκτελούνται "όταν η συνθήκη είναι αληθής".
  - ❑ Αν και εκφράζουν διαδικαστική γνώση, η συνθήκη τους περιέχει δηλωτική γνώση.
- ❖ Οι ενεργοί κανόνες (active rules) εκφράζουν *καθαρά διαδικαστική γνώση*
  - ❑ Κανόνες οδηγούμενοι από συμβάντα ή γεγονότα (event-driven rules)
- ❖ Οι ενεργοί κανόνες εκφράζουν με σαφήνεια το πότε ακριβώς ενεργοποιούνται:
  - ❑ Όταν συμβεί ένα συγκεκριμένο συμβάν.
  - ❑ Τότε και μόνο τότε εξετάζεται η συνθήκη τους και αν ικανοποιείται, τότε εκτελούνται οι ενέργειές τους.
- ❖ Παραδείγματα συμβάντων:
  - ❑ Μία συγκεκριμένη χρονική στιγμή του ρολογιού του συστήματος
  - ❑ Ένα πάτημα του αριστερού πλήκτρου του ποντικού ή ενός πλήκτρου του πληκτρολογίου
  - ❑ Η επιλογή κάποιου μενού από το χρήστη
  - ❑ Η προσπάθεια προσπέλασης ή αλλαγής κάποιων "ευαίσθητων" δεδομένων, κλπ.

# Παράδειγμα Αναπαράστασης με Κανόνες

Σύμπτωμα	Πιθανή Βλάβη	Επιδιόρθωση
Ο εκτυπωτής τυπώνει σωστά αλλά τα χρώματα δε τυπώνονται σωστά	Έχει τελειώσει το έγχρωμο μελάνι	Αλλάξτε την κεφαλή με το έγχρωμο μελάνι

## ❖ Συνεπαγωγικός Κανόνας

**IF** ο εκτυπωτής τυπώνει σωστά **AND**  
τα χρώματα δε τυπώνονται σωστά  
**THEN** έχει τελειώσει το έγχρωμο μελάνι

## ❖ Κανόνας Παραγωγής

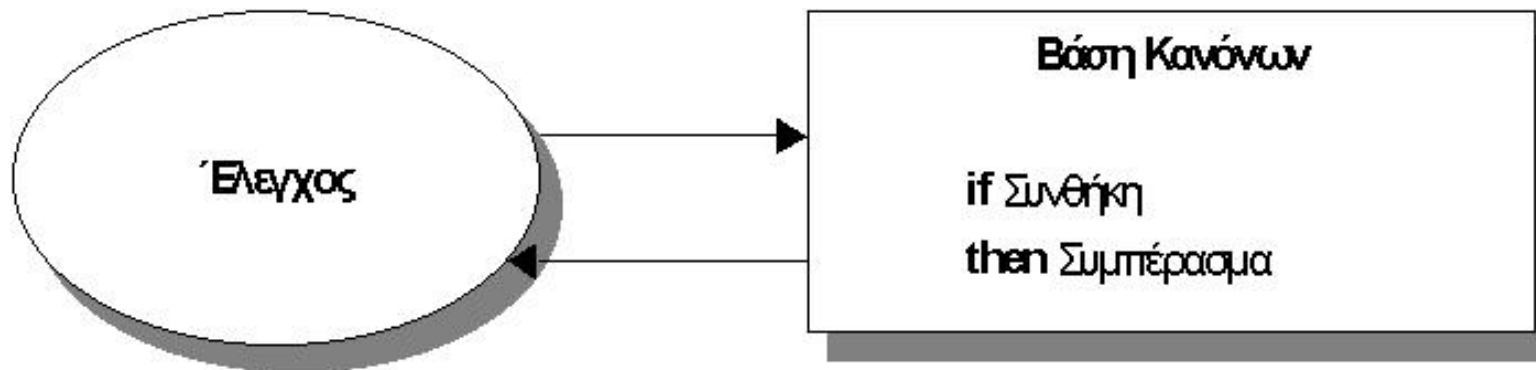
**IF** ο εκτυπωτής τυπώνει σωστά **AND**  
τα χρώματα δε τυπώνονται σωστά  
**THEN** αλλάξτε την κεφαλή με το έγχρωμο μελάνι

## ❖ Ενεργός Κανόνας

**ON** εκτύπωση  
**IF** τα χρώματα δε τυπώνονται σωστά  
**THEN** αλλάξτε την κεφαλή με το έγχρωμο μελάνι

# A) Συστήματα Εξαγωγής Συμπερασμάτων (deduction systems)

- ❖ Αποτελούνται από δύο μέρη:
- ❖ Τη **βάση κανόνων** (rule base), η οποία περιέχει ένα σύνολο από κανόνες.
- ❖ Τον **έλεγχο** (control), ο οποίος καθορίζει τον τρόπο με τον οποίο θα εκτελεστούν οι κανόνες για να εξαχθούν τα συμπεράσματα. Είναι ένας αλγόριθμος που αποφασίζει
  - ❑ Ποιοι από τους κανόνες είναι υποψήφιοι για να επιλύσουν το πρόβλημα,
  - ❑ Με ποιόν τρόπο θα γίνει η επιλογή,
  - ❑ Ποιος από τους κανόνες αυτούς τελικά θα επιλεγεί και
  - ❑ Τι θα γίνει με τους υπόλοιπους κανόνες.
- ❑ Όλα αυτά τα χαρακτηριστικά ανάγουν την επίλυση του προβλήματος σε πρόβλημα αναζήτησης της λύσης, δηλαδή εύρεση της ακολουθίας των κανόνων.



# Εξαγωγή Συμπερασμάτων

- ❖ Χρησιμοποιείται η συνεπαγωγική συλλογιστική.
  - Η εξαγωγή συμπερασμάτων (inference) εξαρτάται όχι μόνο από τη συλλογιστική, αλλά και από τη στρατηγική αναζήτησης στη γνώση ενός προβλήματος. (δες κεφ.8)
- ❖ Υπάρχουν δύο είδη εκτέλεσης κανόνων (στρατηγική αναζήτησης):
  - *Ορθή ακολουθία εκτέλεσης (forward chaining)*: Η διαδικασία εξαγωγής συμπερασμάτων εξετάζει πρώτα αν οι προϋποθέσεις στο αριστερό μέρος του κανόνα είναι αληθείς έτσι ώστε το συμπέρασμα που αναφέρεται στο δεξιό μέρος να είναι αληθές.
    - Εξετάζονται μόνο οι αληθείς τρόποι απόδειξης, αλλά το σύστημα μπορεί να συμπεράνει περισσότερα συμπεράσματα από τα επιθυμητά .
    - **Ενδείκνυται όταν** υπάρχουν λίγα δεδομένα (δίδονται στο σύστημα όλα μαζί στην αρχή) και μπορούν να οδηγήσουν σε πολλά συμπεράσματα.
    - **Εφαρμογές:** Συστήματα Διάγνωσης, Συστήματα Παραγωγής.
  - *Ανάστροφη ακολουθία εκτέλεσης (backward chaining)*: Η διαδικασία εξαγωγής συμπερασμάτων ξεκινά από το δεξιό μέρος του κανόνα και προσπαθεί να βρει αν οι προϋποθέσεις είναι αληθείς.
    - Εξετάζονται όλοι οι εναλλακτικοί τρόποι απόδειξης του συμπεράσματος (ακόμα και αυτοί που δεν είναι αληθείς) έως ότου αποδειχθεί η αλήθεια του συμπεράσματος (όπως στην Prolog).
    - **Ενδείκνυται όταν** υπάρχουν λίγα συμπεράσματα και πολλά δεδομένα, για τα οποία το σύστημα μας καθοδηγεί ζητώντας τα με μια λογική σειρά και όσα χρειάζονται.
    - **Εφαρμογές:** Συστήματα Ελέγχου Λειτουργίας (Monitoring).

# Αναπαράσταση με Κανόνες

## Παράδειγμα

1: if has(Animal, hair) or gives(Animal, milk)  
then isa(Animal, mammal) .

2: if has(Animal, feathers) or  
(flies(Animal) and lays(Animal, eggs))  
then isa(Animal, bird) .

3: if isa(Animal, mammal) and  
(eats(Animal, meat) or  
(has(Animal, pointed\_teeth) and  
has(Animal, claws) and  
has(Animal, forward\_pointing\_eyes)))  
then isa(Animal, carnivore) .

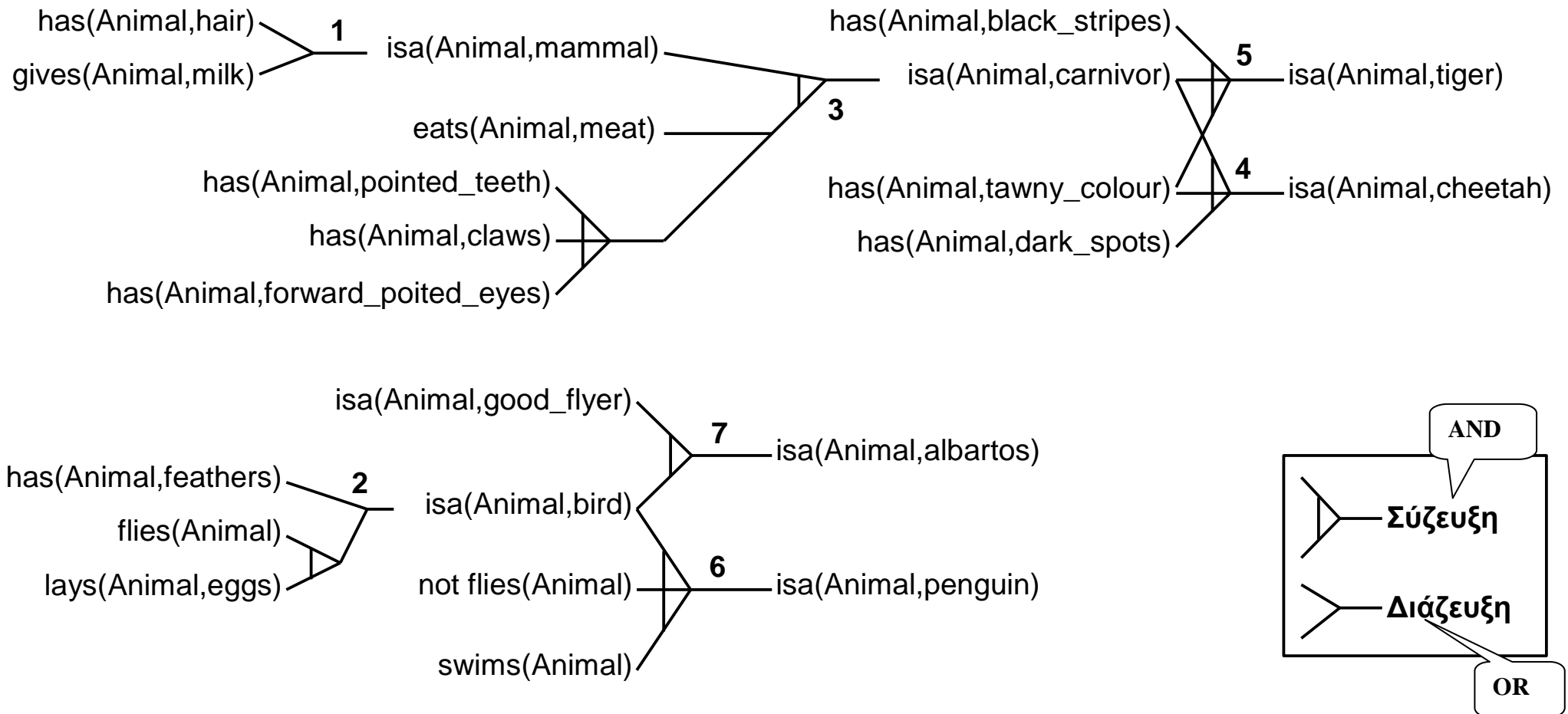
4: if isa(Animal, carnivore) and  
has(Animal, tawny\_colour) and  
has(Animal, dark\_spots)  
then isa(Animal, cheetah) .

5: if isa(Animal, carnivore) and  
has(Animal, tawny\_colour) and  
has(Animal, black\_stripes)  
then isa(Animal, tiger) .

6: if isa(Animal, bird) and  
not flies(Animal) and  
swims(Animal)  
then isa(Animal, penguin) .

7: if isa(Animal, bird) and  
isa(Animal, good\_flyer)  
then isa(Animal, albatros) .

# Γραφική Αναπαράσταση Κανόνων

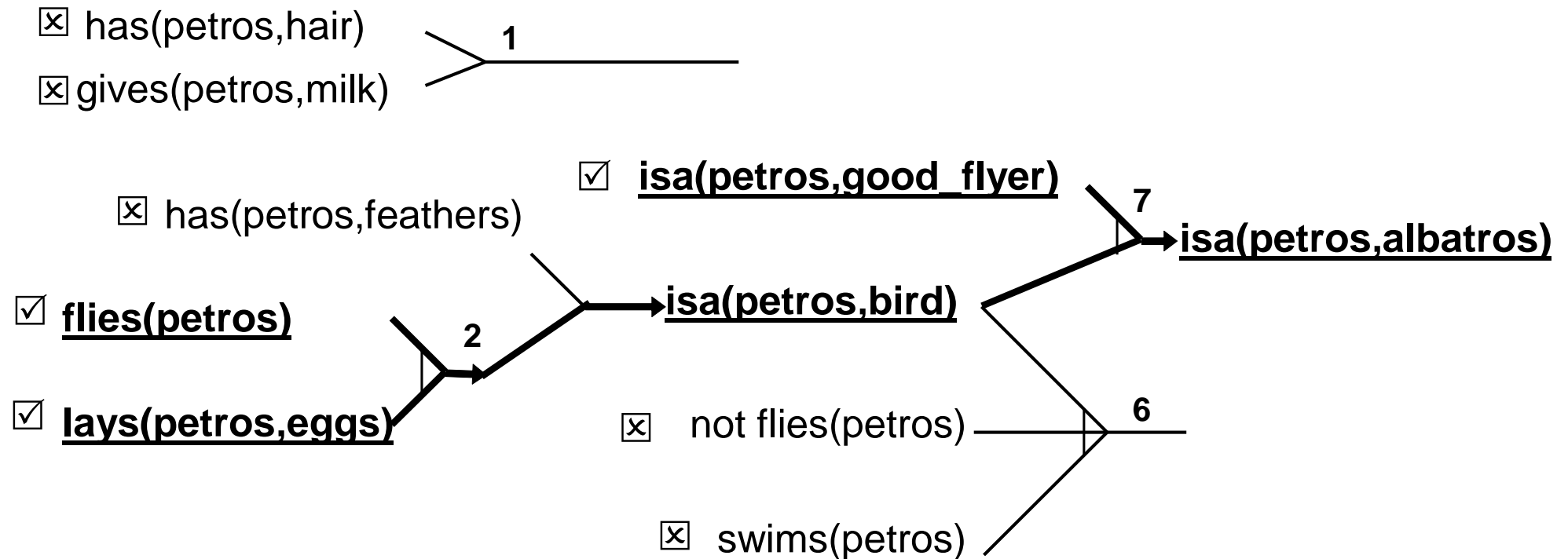


❖ Στο σύνολο κανόνων του παραδείγματος μπορεί να χρησιμοποιηθεί ορθή ή ανάστροφη ακολουθία εκτέλεσης.

# Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

## Ορθή Ακολουθία Εκτέλεσης

- ❖ Αρχικά δεδομένα: `flies(petros)`, `lays(petros, eggs)`, `isa(petros, goodflyer)`.
- ❖ Δίπλα σε κάθε γεγονός φαίνεται με σύμβολα  και  αν οι προτάσεις είναι αληθείς και συμβάλουν στην απόδειξη.



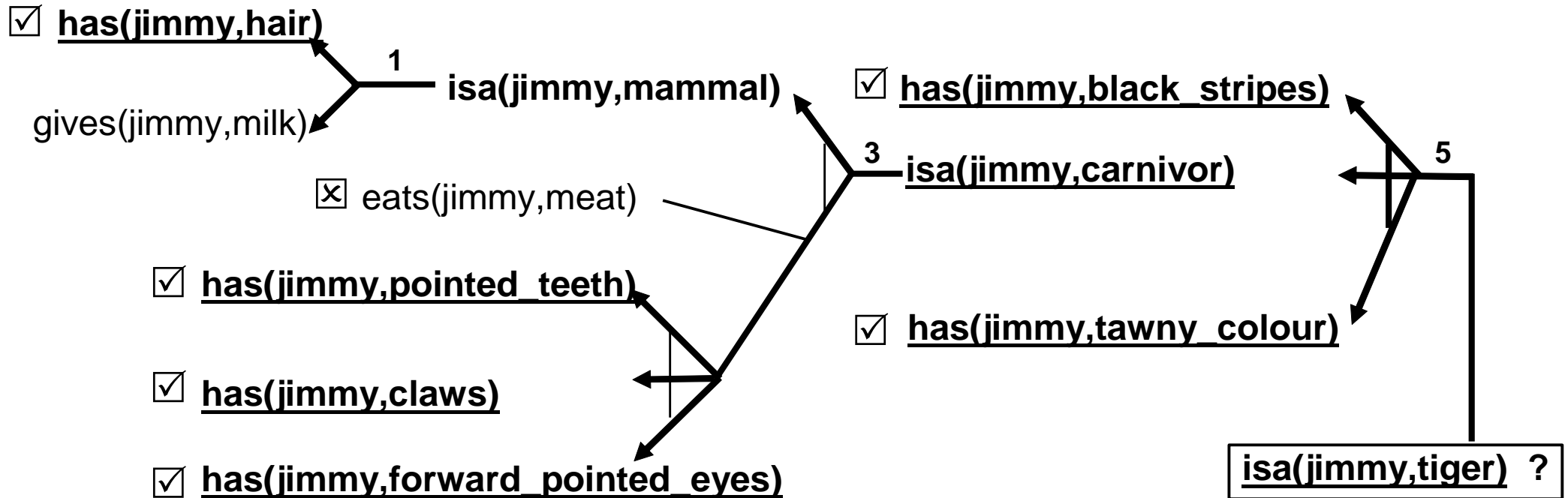
- ❖ Παράγονται τα συμπεράσματα: `isa(petros, albatros)` και `isa(petros, bird)`



# Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

## Ανάστροφη Ακολουθία Εκτέλεσης

❖ Ερώτηση: **isa(jimmy, tiger)**



❖ **Απάντηση: Yes**

# Παράδειγμα

- ❖ **Πρόβλημα:**
- ❖ Καθορισμός του μέσου (Mode) μετακίνησης από ένα τόπο σε έναν άλλο και
- ❖ Καθορισμός του δευτερεύοντος μέσου μετακίνησης (SUBMODE) όταν φτάσεις στον προορισμό.

1. IF DISTANCE > 500 THEN MODE = FLY
2. IF DISTANCE <= 500 THAN MODE = DRIVE
3. IF MODE = DRIVE THEN SUBMODE = DRIVE
4. IF MODE = FLY THEN SUBMODE = RENTCAR CF 50  
AND SUBMODE = TAXI CF 50
5. IF MODE = FLY AND (DISCOUNT OR INDEPENDENT)  
THEN SUBMODE = RENTCAR
6. IF MODE = FLY AND ! FAMILIAR THEN  
SUBMODE = TAXI CF 40

## Δήλωση κανόνων στο FLEX:

### Γενική Μορφή:

**RULE** όνομα\_κανόνα

**IF** συνθήκες

**THEN** ενέργειες ;

**BECAUSE** επεξήγηση ;

**SCORE** βαθμός .

### Παραδείγματα:

**RULE** buy\_food

**IF**

Y is a cat and

Y's owner is X and

Y's meal is Z and

Z's availability is low.

**THEN**

buy (X,Z) ;

**BECAUSE**

“There is not enough food for the cat” ;

**SCORE** 5 .

**RULE** visit\_veterinarian

**IF**

Y is a cat and

Y's health is bad

Y's owner is X

**THEN**

go\_to\_veterinarian (X,Z)

**BECAUSE**

“Your cat is ill”;

**SCORE** 15 .

## Δήλωση κανόνων στο CLIPS:

### Γενική Μορφή:

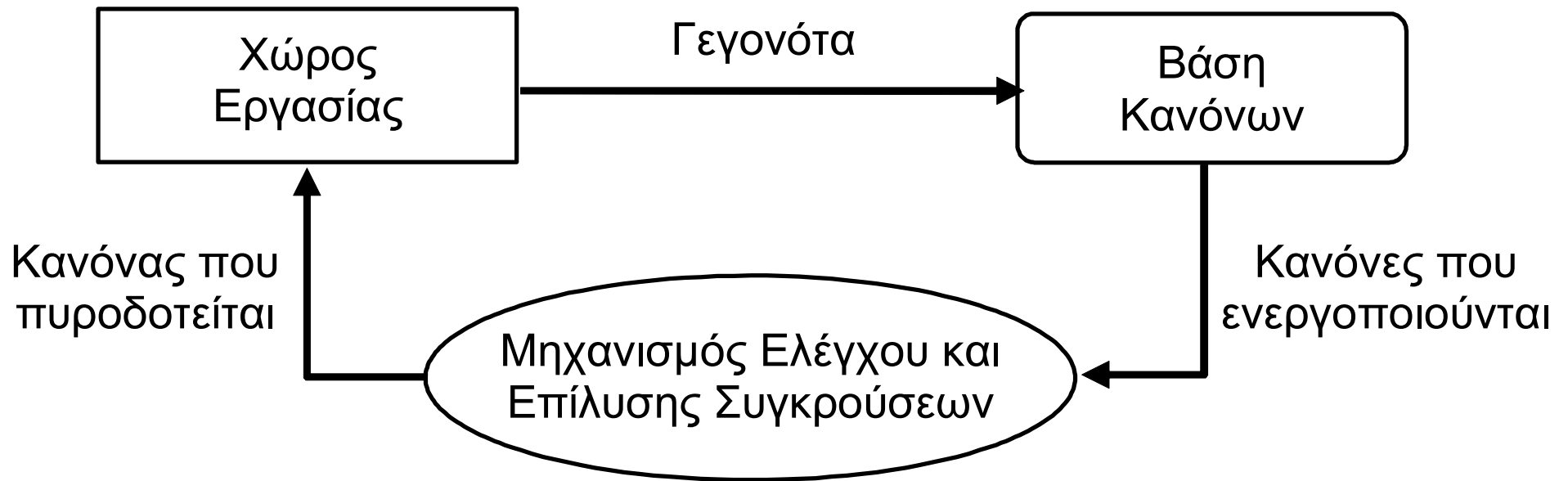
```
(defrule <όνομα κανόνα>
  [”<σχόλια>”]
  [(declare (salience S))]
  (<συνθήκη 1>)
  ...
  ...
  (<συνθήκη n>)
=>
  (<εντολή 1>)
  ...
  (<εντολή m>)
)
```

### Παραδείγματα:

```
(defrule visit_doctor
  “Rule for taking child to doctor”
  (declare (salience 10))
  (child ?Y)
  (has_fever ?Y ?F)
  (test (> ?F 39))
  (parent ?X ?Y)
=>
  (assert (go_to_doctor ?X ?Y))
)
```

## B) Συστήματα Παραγωγής

- ❖ Ένα σύστημα παραγωγής (production system) αποτελείται από 3 μέρη:
  - ❑ Τη **βάση κανόνων** που περιέχει τους κανόνες παραγωγής.
  - ❑ Το **χώρο εργασίας** (working memory), που περιέχει γεγονότα τα οποία είναι αρχικά δεδομένα (data) ή ενδιάμεσα συμπεράσματα (partial conclusions).
    - Στοιχεία της μνήμης εργασίας (working memory elements).
  - ❑ Το **μηχανισμό ελέγχου** (control ή scheduler) και **επίλυσης συγκρούσεων** (conflict resolution), ο οποίος είναι υπεύθυνος για την εκτέλεση των κανόνων, βάσει μιας στρατηγικής επίλυσης συγκρούσεων (conflict resolution strategy).



# Επίλυση Συγκρούσεων

- ❖ Όταν ικανοποιούνται οι συνθήκες του κανόνα, αυτός **ενεργοποιείται** (triggers)
- ❖ Το σύνολο των κανόνων που ενεργοποιούνται σχηματίζουν το **σύνολο σύγκρουσης** (*conflict set*)
- ❖ Ο μηχανισμός ελέγχου είναι υπεύθυνος να καθορίσει ποιος τελικά κανόνας από το σύνολο συγκρούσεων θα πυροδοτηθεί.
- ❖ Όταν ένας κανόνας **πυροδοτείται** (*fires*) τότε οι ενέργειές του εφαρμόζονται ή εκτελούνται.
- ❖ Μερικές από τις πιο γνωστές στρατηγικές επίλυσης συγκρούσεων είναι οι εξής:
  - Τυχαία (random).
  - Διάταξης (ordering).
  - Επιλογή του πρόσφατου (recency).
  - Επιλογή του πιο ειδικού (specificity).
  - Αποφυγή επανάληψης (refractoriness).
  - Ανάλυση μέσων-σκοπών (means-ends analysis)
  - Μετα-έλεγχος (μετα-κανόνες)

# Κύκλος Λειτουργίας Συστήματος Παραγωγής

Έως ότου δε μπορεί να εκτελεστεί κανένας κανόνας επανέλαβε:

1. Βρες όλους του κανόνες που οπλίζουν και σχημάτισε το σύνολο συγκρούσεων.
2. Σύμφωνα με το μηχανισμό επίλυσης συγκρούσεων, διάλεξε ένα κανόνα.
3. Πυροδότησε τον κανόνα που διάλεξες στο βήμα 2.

- ❖ Κατά κανόνα τα συστήματα παραγωγής τα χαρακτηρίζει η ορθή ακολουθία εκτέλεσης κανόνων.
- ❖ Δεν έχει νόημα ο όρος εξαγωγή συμπερασμάτων, γιατί οι κανόνες παραγωγής αναφέρονται σε ενέργειες που εκτελούνται και όχι σε συμπεράσματα.
  - Παρόλα αυτά ο τρόπος λειτουργίας τους παραπέμπει στη συνεπαγωγική συλλογιστική
    - Υιοθέτηση μιας ειδικής ενέργειας από κάτι που ισχύει γενικά.
    - Ταίριασμα των κανόνων που περιέχουν μεταβλητές με δεδομένα στη μνήμη εργασίας που περιέχουν σταθερές.

# Παράδειγμα

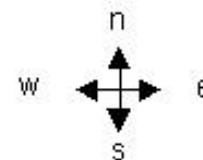
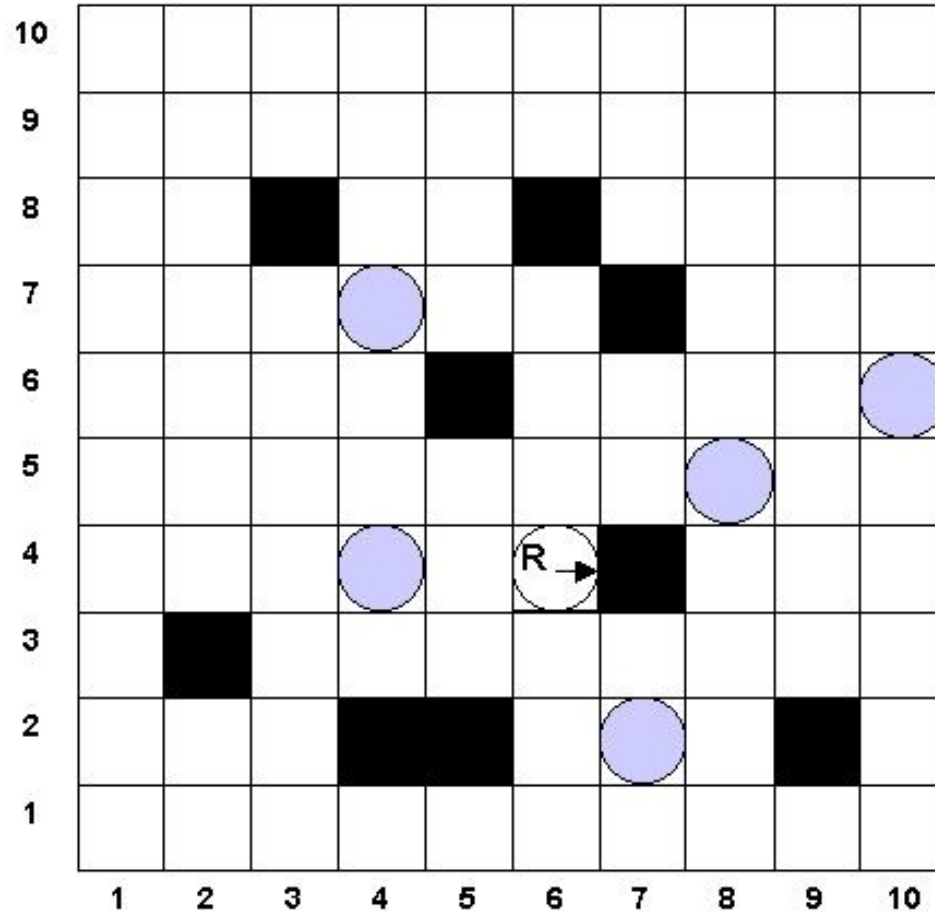
**IF:** Company is customer, and  
Company has good Credit, and  
Company is over limit by small amount, or  
Company is very large customer

**THEN:** Extend Credit for current month, and  
add Company to exception list.



## Παράδειγμα Κίνησης Ρομπότ

- Ένα ρομπότ κινείται σε κάποιο χώρο με εμπόδια. Στόχος του είναι να αποφύγει τα εμπόδια και όταν βρει κάποιο αντικείμενο, να στείλει ένα μήνυμα και να σταματήσει.



```
robot_at(6,4)
direction(e)
choice(w)
choice(s)
choice(n)
choice(e)
obstacle_at(7,4)
obstacle_at(6,8)
obstacle_at(7,7)
...
object_at(4,7)
...
```

## Παρατηρήσεις

- ❖ Η **μνήμη εργασίας** περιέχει: (όπου  $X, Y$  είναι ακέραιοι αριθμοί)
  - ❑ τη θέση του ρομπότ: `robot_at(X, Y)`
  - ❑ την κατεύθυνση προς την οποία κινείται: `direction(D)`, όπου  $D$  είναι μία από τις 4 κατευθύνσεις  $e, w, n, s$ .
  - ❑ τη θέση των εμποδίων: `obstacle_at(X, Y)`
  - ❑ τη θέση των αντικειμένων: `object_at(X, Y)`
  - ❑ Την επιλογή της κατεύθυνσης: `choice(D)`, όπου  $D$  είναι οι τέσσερις επιλογές αλλαγής κατεύθυνσης  $e, w, n, s$
- ❖ Οι **ενέργειες** των κανόνων εμπεριέχουν τέσσερις λειτουργίες:
  - ❑ `addwm`: βάλε κάτι στη μνήμη εργασίας
  - ❑ `delwm`: σβήσε κάτι από τη μνήμη εργασίας
  - ❑ `output`: εκτύπωσε ένα μήνυμα στην οθόνη, και
  - ❑ αριθμητικές εκφράσεις.
- ❖ Οι κανόνες παραγωγής εκφράζουν τη συμπεριφορά του ρομπότ

Σταθερά

# Κανόνες Κίνησης Ρομπότ

- 1: `detect_object`: `if robot_at(X,Y) and object_at(X,Y)`  
    `then output('object is found')`.
- 2: `move_west`: `if robot_at(X,Y) and direction(w)`  
    `then delwm(robot_at(X,Y)) and NX=X-1 and addwm(robot_at(NX,Y))`.
- 3: `move_east`: `if robot_at(X,Y) and direction(e)`  
    `then delwm(robot_at(X,Y)) and NX=X+1 and addwm(robot_at(NX,Y))`.
- 4: `move_north`: `if robot_at(X,Y) and direction(n)`  
    `then delwm(robot_at(X,Y)) and NY=Y+1 and addwm(robot_at(X,NY))`.
- 5: `move_south`: `if robot_at(X,Y) and direction(s)`  
    `then delwm(robot_at(X,Y)) and NY=Y-1 and addwm(robot_at(X,NY))`.
- 6: `avoid_obstacle_south`:  
    `if robot_at(X,Y) and NY=Y-1 and obstacle_at(X,NY) and direction(s) and choice(ND)`  
    `then delwm(direction(s)) and addwm(direction(ND))`.
- 7: `avoid_obstacle_west`:  
    `if robot_at(X,Y) and NX=X-1 and obstacle_at(NX,Y) and direction(w) and choice(ND)`  
    `then delwm(direction(w)) and addwm(direction(ND))`.
- 8: `avoid_obstacle_north`:  
    `if robot_at(X,Y) and NY=Y+1 and obstacle_at(X,NY) and direction(n) and choice(ND)`  
    `then delwm(direction(n)) and addwm(direction(ND))`.
- 9: `avoid_obstacle_east`:  
    `if robot_at(X,Y) and NX=X+1 and obstacle_at(NX,Y) and direction(e) and choice(ND)`  
    `then delwm(direction(e)) and addwm(direction(ND))`.

MOVE

Change  
Direction

# Στρατηγική Επίλυσης Κίνησης Ρομπότ

- ❖ Οι στρατηγικές επίλυσης συγκρούσεων είναι με τη σειρά:
  - ❑ αποφυγή επανάληψης (AE),
  - ❑ επιλογή του πιο ειδικού (EE), και
  - ❑ τυχαία επιλογή (TE).

Κύκλος	Μνήμη Εργασίας	Σύνολο Συγκρούσεων	Στρατηγική	Κανόνας που πυροδοτεί
1	robot_at(6,4) direction(e) choice(w) choice(n) choice(s) choice(e) obstacle_at(7,4) obstacle_at(6,8) . . . object_at(4,7) . . .	{3, 9 (ND=w) , 9 (ND=n) , 9 (ND=s) , 9 (ND=e) }	EE TE	9:avoid_obstacle_east (ND=n)
2	robot_at(6,4) direction(n) . . .	{4}	-	4: move_north

Κανόνες που οπλίζουν απο τα δεδομένα της Μνήμης Εργασίας.  
 Ο (3) από το direction(e) και ο (9) από το direction(e) και από το obstacle(7,4).

Επιλέγεται ένας (9), με choice(n)

Κύκλος	Μνήμη Εργασίας	Σύνολο Συγκρούσεων	Στρατηγική	Κανόνας που πυροδοτεί
3	robot_at(6,5) direction(n) . . .	{4}	-	4: move_north
4	robot_at(6,6) direction(n) . . .	{4}	-	4: move_north
5	robot_at(6,7) direction(n) . . . obstacle_at(6,8) . . .	{4, 8 (ND=w) , 8 (ND=n) , 8 (ND=s) , 8 (ND=e) }	EE TE	8:avoid_obstacle_north (ND=n)
6	robot_at(6,7) direction(n) . . . obstacle_at(6,8) . . .	{4, 8 (ND=w) , 8 (ND=n) , 8 (ND=s) , 8 (ND=e) }	AE EE TE	8:avoid_obstacle_north (ND=e)
7	robot_at(6,7) direction(e) . . . obstacle_at(7,7) . . .	{3, 9 (ND=w) , 9 (ND=n) , 9 (ND=s) , 9 (ND=e) }	EE TE	9: avoid_obstacle_east (ND=w)

Κύκλος	Μνήμη Εργασίας	Σύνολο Συγκρούσεων	Στρατηγική	Κανόνας που πυροδοτεί
8	robot_at(6,7) direction(w) . . .	{2}	-	2: move_west
9	robot_at(5,7) direction(w) . . .	{2}	-	2: move_west
10	robot_at(4,7) direction(w) object_at(4,7) . . .	{1,2}	EE TE	1: detect_object

# Ασκήσεις

❖ 7.8 Έστω το παρακάτω πρόγραμμα σε CLIPS:

```
(deffacts sample-data (data 3)) (defrule rule2
(defrule rule1          (declare (salience 0))
  (declare (salience 0)) (data ?x)
  (data ?x)             =>
=>                       (printout t (* 2 ?x) crlf)
  (printout t (* 1 ?x) crlf) )
  (assert (extra 5))    (defrule rule3
)                          (declare (salience 100))
                          (extra ?x)
                          =>
                          (printout t (* 3 ?x) crlf)
                          )
```

- Τι θα τυπώσει το παραπάνω πρόγραμμα στην οθόνη αν εκτελεστούν οι παρακάτω εντολές:

```
(set-strategy breadth)
(reset)
(run)
```



## ❖ 7.9. Δίνεται το εξής σύστημα κανόνων:

1. IF has\_fever AND has\_throat\_ache THEN virus\_infected
2. IF has\_fever AND NOT has\_throat\_ache THEN has\_cold
3. IF has\_throat\_ache AND NOT has\_fever THEN has\_pharyngitis
4. IF angina AND NOT has\_cold then pneumonia
5. IF angina AND has\_pharyngitis then bronchitis
6. IF has\_pharyngitis AND NOT angina then get\_syrup
7. IF virus\_infected THEN get\_antibiosis
8. IF has\_cold THEN get\_aspirin
9. IF has\_pharyngitis THEN visit\_doctor
10. IF pneumonia THEN visit\_doctor
11. IF bronchitis THEN visit\_doctor

- ❑ Θεωρήστε ότι αρχικά στη μνήμη εργασίας βρίσκονται τα γεγονότα: *angina*. Να βρεθούν όλα τα συμπεράσματα που προκύπτουν από το παραπάνω σύστημα κανόνων με ορθή ακολουθία εκτέλεσης.
- ❑ Θεωρήστε ότι αρχικά στη μνήμη εργασίας βρίσκονται το γεγονός: *has\_fever*. Να βρεθούν όλα τα συμπεράσματα που προκύπτουν από το παραπάνω σύστημα κανόνων με ορθή ακολουθία εκτέλεσης.





## ❖ 7.10. Δίνεται το εξής σύστημα κανόνων:

1. IF Destination="Athens" AND has\_a\_car THEN go\_by\_car
2. IF Destination="Athens" AND NOT has\_a\_car AND has\_money THEN go\_by\_plane
3. IF Destination="Heraklion" AND has\_money THEN go\_by\_plane
4. IF Destination="Heraklion" AND NOT has\_money THEN go\_by\_boat
5. IF go\_by\_plane THEN buy\_plane\_tickets
6. IF go\_by\_boat THEN buy\_boat\_tickets
7. IF go\_by\_car THEN perform\_car\_service

- ❑ Θεωρείστε ότι αρχικά στη μνήμη εργασίας βρίσκονται τα εξής γεγονότα Destination="Athens" και has\_money.
- ❑ Να βρεθούν όλα τα συμπεράσματα που προκύπτουν από το παραπάνω σύστημα κανόνων με ορθή ακολουθία εκτέλεσης και να δοθεί διαγραμματικά η εκτέλεση.
- ❑ Πόσοι κύκλοι λειτουργίας θα εκτελεστούν;



❖ 7.11. Έστω ότι στην λίστα γεγονότων υπάρχουν σε κάποια χρονική στιγμή τα παρακάτω γεγονότα

(first A) (first B) (second D) (second E)

❑ και σε κάποιο πρόγραμμα που είναι φορτωμένο στην μνήμη ο κανόνας  
(defrule combine

(first ?x)

(second ?y)

=>

(printout t " comb " ?x ?y crlf))

❑ Ποιό θα είναι το αποτέλεσμα του παραπάνω κανόνα;



## ❖ 7.12. Έστω το ακόλουθο πρόγραμμα CLIPS.

```
(deffacts elevetor-move          (defrule initial-rule
  (elevator on ground)           =>
  (elevator ON))                 (assert (goto floorA)))

(defrule moving-elevetor        (defrule empty-elevetor
  (elevator ON)                  (elevator on floorA)
  ?x <- (elevator on ?floor)     =>
  ?y <- (goto ?floor2)           (assert (empty elevetor))
=>                                (assert (goto ground)))
  (retract ?x ?y)
  (assert (elevator on ?floor2)))
```

- ❑ Έστω ότι το πρόγραμμα φορτώνεται στην μνήμη και εκτελούνται οι εντολές (reset) και (run).
- ❑ Να γραφεί αναλυτικά ποιό κανόνες θα πυροδοτηθούν και ποιό θα είναι το αποτέλεσμα τους στην λίστα γεγονότων (ποιά γεγονότα θα προστεθούν ποιά θα διαγραφούν) σε κάθε κύκλο.