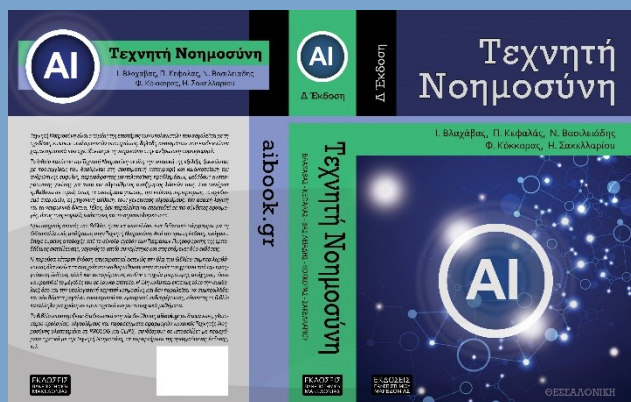




Κεφάλαιο 11

Συστήματα Κανόνων



Τεχνητή Νοημοσύνη - Δ' Έκδοση-Εκδόσεις Πανεπιστημίου Μακεδονίας

ISBN: 978-618-5196-44-8 - <https://aibook.gr/>

Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου.

Ι. Βλαχάβας, καθηγητής

Τμήμα Πληροφορικής, ΑΠΘ



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
ΟΝ 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 κατηγορήμα.
- ❖ 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*):
 - ❑ Οι κανόνες εκφράζουν μια (δηλωτική) γνώση που δηλώνει μία αλήθεια για τον κόσμο του προβλήματος, αλλά δεν αναφέρει ρητά πότε και πώς εφαρμόζεται.
 - ❑ Οι κανόνες αυτοί ονομάζονται *συμπερασματικοί* ή *εξαγωγής συμπερασμάτων* (*deductive rules*).
 - ❑ Παραδείγματα τέτοιων συστημάτων αποτελούν η γλώσσα λογικού προγραμματισμού PROLOG, οι συνεπαγωγικές βάσεις δεδομένων (*deductive databases*) και η γλώσσα ερωτήσεων DATALOG, καθώς και οι γλώσσες κανόνων σε οντολογίες του Σημασιολογικού Ιστού (*semantic web*) SWRL και SPIN (a [SPARQL](#)-based rule and constraint language for the Semantic Web).
- ❖ Συστήματα παραγωγής (*production systems*): π.χ. [CLIPS](#), [Flex](#)
 - ❑ Οι κανόνες (*παραγωγής* – production rules) εκφράζουν διαδικαστική 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.
 - ❑ Παραδείγματα τέτοιων συστημάτων αποτελούν τα *CLIPS*, *JESS*, *JBOSS DROOLS*, *IBM BUSINESS RULES*, καθώς και η γλώσσα κανόνων του Σημασιολογικού Ιστού *SPIN*.
 - ❑ Αν και αναπαριστούν διαδικαστική γνώση, εντούτοις δεν είναι σαφώς ορισμένο πότε ακριβώς εκτελούνται οι ενέργειές τους, παρά μόνον αναφέρεται με ασάφεια πως οι κανόνες εκτελούνται όταν η συνθήκη είναι αληθής.
 - ✓ Αν και εκφράζουν διαδικαστική γνώση, η συνθήκη τους περιέχει δηλωτική γνώση



Ενεργοί Κανόνες

- ❖ Κανόνες οδηγούμενοι από συμβάντα ή γεγονότα (*event-driven rules*)
 - ❑ Εκφράζουν καθαρά διαδικαστική γνώση
- ❖ Ενεργοποιούνται όταν συμβεί ένα συγκεκριμένο συμβάν
 - ❑ 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.
- ❖ Εμφανίζονται με διάφορες μορφές σε διάφορα είδη λογισμικού,
 - ❑ π.χ. ως *δαίμονες* στα πλαίσια ή ως *σκανδαλιστές* (triggers) στις ΒΔ.
 - ❑ Αποτελούν τη γέφυρα μεταξύ της ΤΝ και των συμβατικών πληροφοριακών συστημάτων.
 - ✓ π.χ. τα συστήματα ενεργών κανόνων ORACLE TRIGGERS, RULECORE, IBM AMIT, DRUPAL RULES MODULE
 - ❑ Πρόσφατα, με την ανάπτυξη διαδικτυακών υπηρεσιών (web services) του Web 2.0, του mobile computing) καθώς και του Internet of Things (IoT), αναπτύχθηκαν εφαρμογές, όπως:
 - ✓ Η υπηρεσία *IFTTT* (If This Then That) η οποία βασίζεται στο μοντέλο των ενεργών κανόνων.
 - ✓ Τα συμβάντα αποτελούν οι διάφορες αλλαγές όπως η λήψη ενός email, το ανέβασμα μια φωτογραφίας σε κοινωνικό δίκτυο, ή η ενεργοποίηση του συναγερμού στο σπίτι μας.



Rule-based system



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

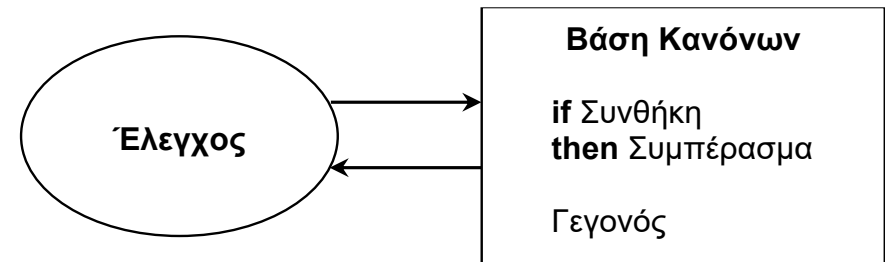
ON εκτύπωση
IF τα χρώματα δε τυπώνονται σωστά
THEN αλλάξτε την κεφαλή με το έγχρωμο μελάνι



Α) Συστήματα Εξαγωγής Συμπερασμάτων (deduction systems)

❖ Αποτελούνται από δύο μέρη:

- ❑ Τη **βάση κανόνων** (*rule base*), η οποία περιέχει ένα σύνολο από κανόνες και γεγονότα.
 - ✓ τα γεγονότα θεωρούνται ως τα δεδομένα του προγράμματος και μπορούν να αναπαρασταθούν και ως κανόνες των οποίων οι συνθήκες είναι πάντα αληθείς: *if TRUE then Γεγονός*.
- ❑ 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.





Παράδειγμα ελέγχου στην γλώσσα PROLOG

Επιλογή μηχανισμού ελέγχου	Γλώσσα PROLOG
Αλγόριθμος ελέγχου	SLDNF resolution
Ποιοι κανόνες είναι υποψήφιοι για να επιλύσουν το πρόβλημα;	Οι κανόνες των οποίων η κεφαλή ενοποιείται με την τρέχουσα ερώτηση.
Με ποιόν τρόπο θα γίνει η επιλογή;	Βάσει της θέσης του κανόνα στο λογικό πρόγραμμα.
Ποιος από τους κανόνες/γεγονότα τελικά θα επιλεγεί;	Ο πρώτος που ταιριάζει.
Τι θα γίνει με τους υπόλοιπους κανόνες;	Θα παραμείνουν "διαθέσιμοι" ως σημεία οπισθοδρόμησης και θα χρησιμοποιηθούν σε περίπτωση αποτυχίας



Εξαγωγή Συμπερασμάτων

- ❖ Χρησιμοποιείται η συνεπαγωγική συλλογιστική (deductive reasoning).
 - ❑ Η εξαγωγή συμπερασμάτων (inference) εξαρτάται όχι μόνο από τη συλλογιστική, αλλά και από τη στρατηγική αναζήτησης στη γνώση ενός προβλήματος (δες κεφ.8) η οποία υλοποιείται από την ακολουθία εκτέλεσης κανόνων (chaining)
- ❖ Υπάρχουν δύο είδη εκτέλεσης κανόνων:
 - ❑ **Ορθή ακολουθία εκτέλεσης** (forward chaining):
 - ✓ Η διαδικασία εξαγωγής συμπερασμάτων εξετάζει αν οι προϋποθέσεις στο αριστερό μέρος του κανόνα είναι αληθείς ώστε το συμπέρασμα που υπάρχει 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.
 - ❑ Π.χ. έστω ότι υπάρχουν οι ακόλουθοι κανόνες και το γεγονός F1 στη βάση κανόνων
 - ✓ Με ορθή ακολουθία εκτέλεσης θα εκτελεστούν οι κανόνες R3 και R4 και θα προκύψουν τα συμπεράσματα B και W
 - ✓ Ο μηχανισμός εκτέλεσης δεν θα ασχοληθεί με τους κανόνες που δεν δίνουν αποτελέσματα, όπως οι R1 και R2.

```
R1: if A then B
R2: if C then B
R3: if D then B
R4: if D then W
F1: D
```



Εξαγωγή Συμπερασμάτων

- ❑ **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).
- ❑ Παράδειγμα. Έστω ότι υπάρχουν οι ακόλουθοι κανόνες και το γεγονός F1 στη βάση κανόνων
 - ✓ Αν το ζητούμενο είναι να αποδειχθεί αν ισχύει το B, θα εξεταστούν και 3 κανόνες R1, R2, R3, ενώ μόνο ο R3 μπορεί να αποδείξει το συμπέρασμα.
 - ✓ Δεν ασχολείται με τον κανόνα R4, παρόλο που μπορεί να "εκτελεστεί"

```
R1: if A then B
R2: if C then B
R3: if D then B
R4: if D then W
F1: D
```

οι



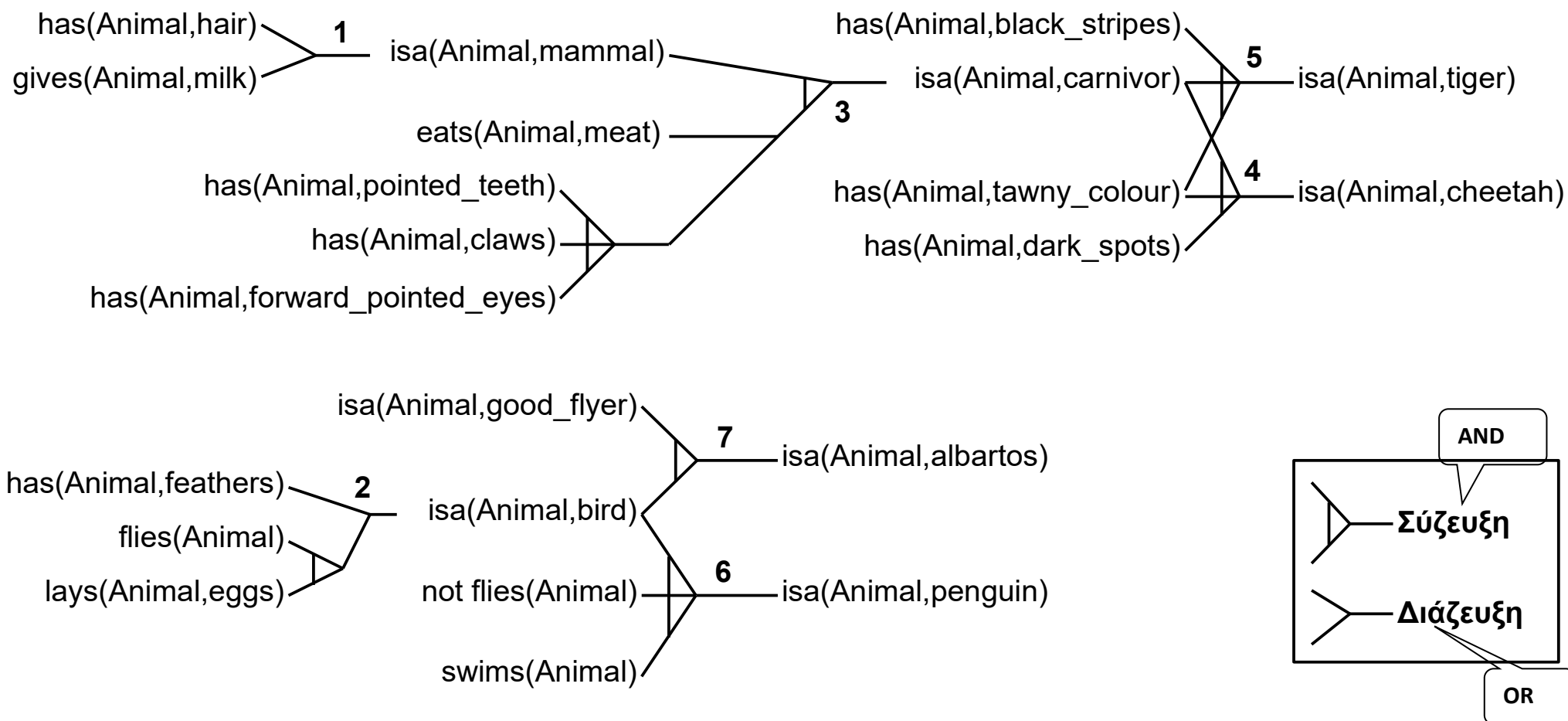
Αναπαράσταση με Κανόνες

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



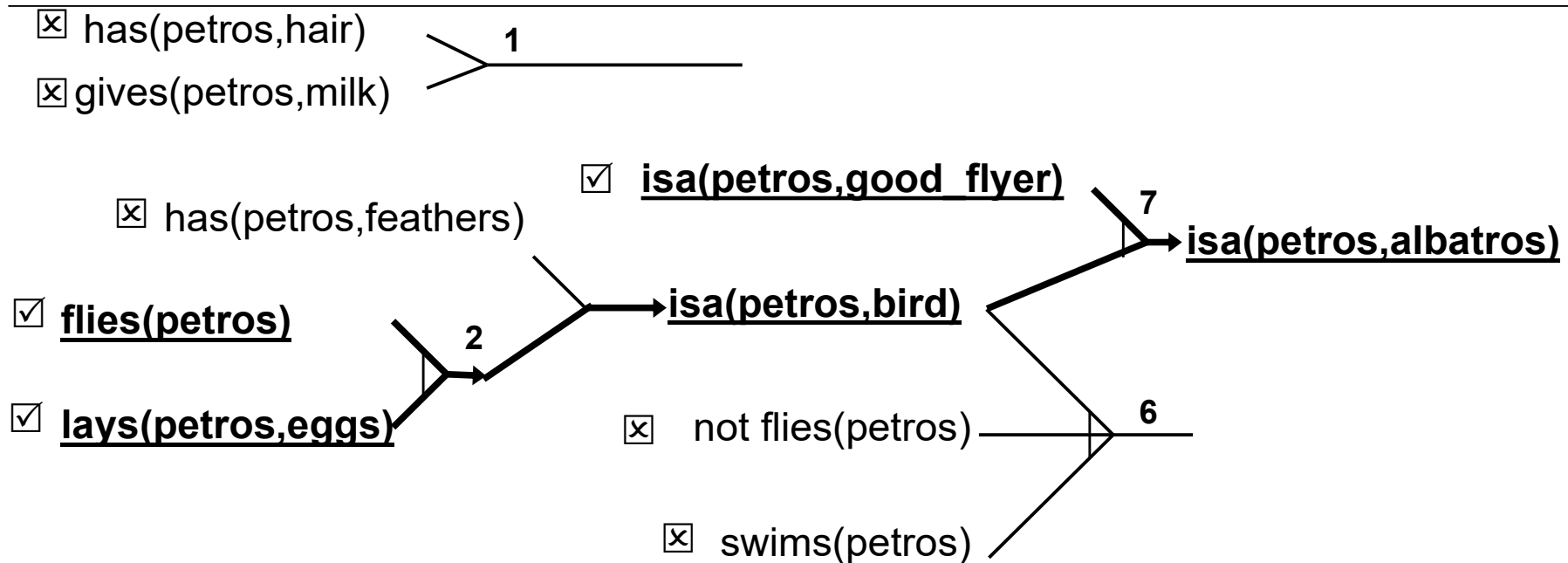
❖ Στο σύνολο κανόνων του παραδείγματος μπορεί να χρησιμοποιηθεί ορθή ή ανάστροφη ακολουθία εκτέλεσης.



Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

Ορθή Ακολουθία Εκτέλεσης

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



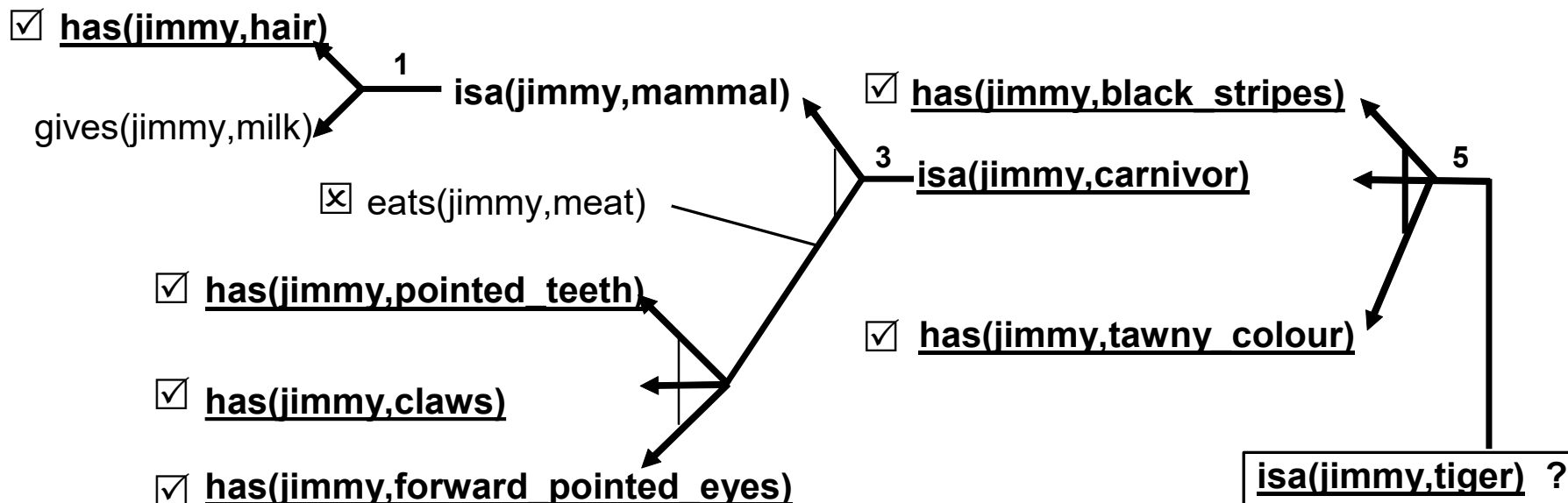
- ❖ Παράγονται τα συμπεράσματα: `isa(petros, albatros)` και `isa(petros, bird)`
- ❑ Είθισται στα συστήματα ορθής ακολουθίας εκτέλεσης να μην γίνονται ερωτήσεις, αλλά να χρησιμοποιείται η υπόθεση του κλειστού κόσμου, καθώς θα ήταν πολύ με-γάλο το πλήθος των πιθανών ερωτήσεων.



Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

Ανάστροφη Ακολουθία Εκτέλεσης

❖ Ερώτηση: **isa(jimmy, tiger)**



- ❖ Η ανάστροφη ακολουθία εκτέλεσης ξεκινά με την ερώτηση του αποδεικτέου `isa(jimmy, tiger)`.
- ❑ Στη συνέχεια, το σύστημα εξαγωγής συμπερασμάτων προσπαθεί να απαντήσει με true ή false στην ερώτηση χρησιμοποιώντας τους κανόνες που έχουν ως συμπέρασμα την παραπάνω ερώτηση.
 - ❑ είτε χρησιμοποιώντας με τον ίδιο τρόπο τα συμπεράσματα άλλων κανόνων, είτε στηριζόμενος στην αλήθεια ή όχι των γεγονότων

❖ **Απάντηση: Yes**



Παράδειγμα

- ❖ 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))  
)
```



Σειρά Εκτέλεσης Κανόνων (1/2)

- ❖ Η σειρά εκτέλεσης των κανόνων στα συστήματα εξαγωγής συμπερασμάτων παίζει ρόλο κυρίως όταν χρησιμοποιείται η ορθή ακολουθία εκτέλεσης
- ❖ Στην ορθή ακολουθία, όταν δεν υπάρχει άρνηση στις συνθήκες, η σειρά εκτέλεσης, συνήθως παίζει ρόλο μόνο στην σειρά με την οποία επιστρέφονται τα αποτελέσματα.
 - ❑ Στο παράδειγμα Α, αν η σειρά εκτέλεσης των κανόνων είναι η R1; R2; R3 ή R2; R1; R3, θα παραχθούν τα ίδια συμπεράσματα C, D, E, απλά με διαφορετική σειρά.
 - ❑ Στο παράδειγμα Β, αν η σειρά εκτέλεσης των κανόνων είναι η R1; R2, τότε ο κανόνας R3 δεν θα εκτελεστεί λόγω του ότι θα έχει εξαχθεί το συμπέρασμα D, και τελικά τα συμπεράσματα που θα εξαχθούν θα είναι τα C και D.
 - ❑ Αν όμως η σειρά εκτέλεσης των κανόνων είναι η R1; R3; R2, τότε ο R3 εκτελείται πριν ο R2 βγάλει το συμπέρασμα D, οπότε τα συμπεράσματα που εξάγονται είναι τα C, E και D.
 - ❑ Επίλυση του προβλήματος με την τεχνική της διαστρωμάτωσης (**stratification**).
 - ✓ Χρησιμοποιείται στη γλώσσα *Datalog* (Λεπτομέρειες στο βιβλίο)
 - ✓ Διαχωρίζει τη βάση γνώσης σε στρώματα (strata), δίνοντας προτεραιότητα στην εκτέλεση όλων των κανόνων που παράγουν συμπεράσματα που αφορούν σε κάποιο κατηγορημα, το οποίο εμφανίζεται με άρνηση στην συνθήκη κάποιου άλλου κανόνα.

Παράδειγμα Α

```
⌈
R1: ·if ·A ·then ·C⌋
→ → R2: ·if ·B ·then ·D⌋
→ → R3: ·if ·C ·and ·D ·then ·E⌋
→ → F1: ·A⌋
→ → F2: ·B⌋
⌋
```

Παράδειγμα Β

```
⌈
R1: ·if ·A ·then ·C⌋
→ → R2: ·if ·B ·then ·D⌋
→ → R3: ·if ·C ·and ·not (D) ·then ·E⌋
→ → F1: ·A⌋
→ → F2: ·B⌋
⌋
```



Σειρά Εκτέλεσης Κανόνων (2/2)

❖ Στην ανάστροφη ακολουθία εκτέλεσης:

- ❑ Η σειρά εκτέλεσης των κανόνων δεν παίζει ρόλο στα συμπεράσματα που επιστρέφονται (ούτε και στην περίπτωση ύπαρξης άρνησης)
- ❑ Συνήθως η αναζήτηση στον χώρο των κανόνων και των γεγονότων της βάσης γνώσης είναι εξαντλητική, από πάνω-προς-τα-κάτω,
 - ✓ όπως για παράδειγμα στην PROLOG
- ❑ Αυτό που αλλάζει είναι η σειρά με την οποία θα εξεταστούν τα δεδομένα του προβλήματος οπότε και η σειρά με την οποία θα γίνουν ερωτήσεις στον χρήστη.
 - ✓ Και συνεπώς το πόσο γρήγορα θα αποδειχθεί ένα συμπέρασμα
- ❑ Στο παράδειγμα Γ, εάν το ζητούμενο είναι η απόδειξη του B:
 - ✓ Αυτό θα αποδειχθεί ανεξάρτητα από την σειρά εκτέλεσης των κανόνων R1, R2, R3
 - ✓ Αν όμως ο κανόνας R3 εξεταστεί πριν από τους R1, R2, τότε η απόδειξη θα δοθεί σε ένα βήμα, ενώ
 - ✓ Αν εξεταστεί μετά από αυτούς, τότε η απόδειξη θα δοθεί σε 3 βήματα

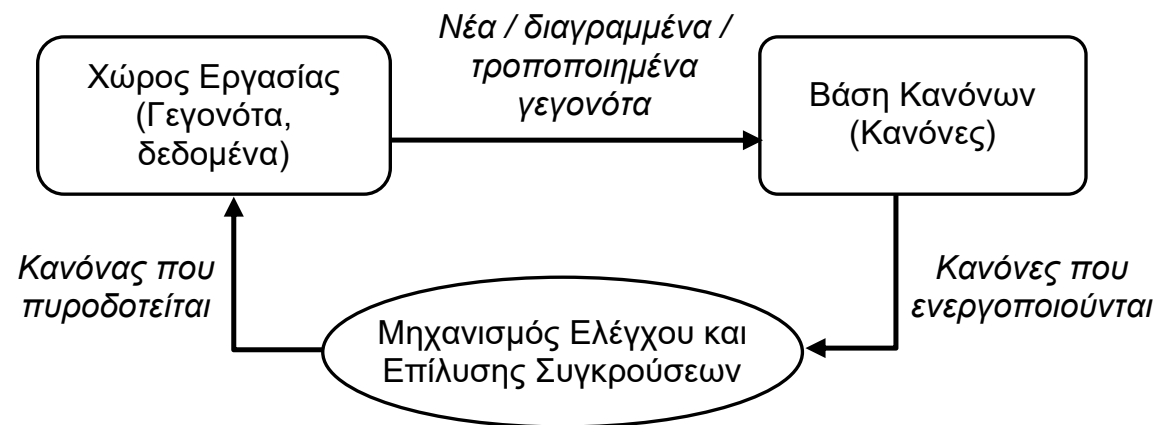
Παράδειγμα Γ

```
R1: if A then B
R2: if C then B
R3: if D then B
R4: if D then W
F1: D
```



Β) Συστήματα Παραγωγής

- ❖ Στα συστήματα παραγωγής οι κανόνες στο δεξιό μέρος τους εκφράζουν ενέργειες που πρέπει να εκτελεστούν αν ισχύουν οι προϋποθέσεις
- ❖ 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).
 - ❑ Ο χώρος εργασίας είναι δυναμικός με την έννοια ότι τα περιεχόμενά του είναι διαφορετικά σε κάθε κύκλο λειτουργίας του συστήματος.
 - ❑ Οι κανόνες παραγωγής είναι αυτοί που καθορίζουν τα περιεχόμενα του, προσθέτοντας ή αφαιρώντας γεγονότα, σύμφωνα με τις ενέργειες του κάθε κανόνα.



[https://en.wikipedia.org/wiki/Production_system_\(computer_science\)](https://en.wikipedia.org/wiki/Production_system_(computer_science))



Ενεργοποίηση Κανόνων

- ❖ Όταν ικανοποιούνται οι συνθήκες του κανόνα, αυτός *ενεργοποιείται* ή *οπλίζει* (*triggers*)
 - ☐ Το σύνολο των κανόνων που ενεργοποιούνται σχηματίζουν το *σύνολο σύγκρουσης* (*conflict set*)
- ❖ Η εκτέλεση ή εφαρμογή των ενεργειών ενός κανόνα ονομάζεται *πυροδότηση* (*fire*) του
 - ☐ Ο μηχανισμός ελέγχου είναι υπεύθυνος να καθορίσει ποιος τελικά κανόνας από το σύνολο συγκρούσεων θα πυροδοτηθεί.
- ❖ Μερικές από τις πιο γνωστές στρατηγικές επίλυσης συγκρούσεων είναι οι εξής:
 - ☐ Τυχαία (*random*)
 - ☐ Διάταξης (*ordering*)
 - ✓ Επιλέγεται ο κανόνας που είναι πρώτος στη σειρά εμφάνισης
 - ☐ Επιλογή του πρόσφατου (*recency*)
 - ✓ Επιλέγεται ο κανόνας που ενεργοποιείται από τα πιο πρόσφατα δεδομένα του χώρου εργασίας
 - ☐ Επιλογή του πιο ειδικού (*specificity*)
 - ✓ Επιλέγεται ο κανόνας του οποίου η συνθήκη εκφράζεται με αναλυτικότερο τρόπο
 - ☐ Αποφυγή επανάληψης (*refractoriness*)
 - ✓ Δεν επιλέγεται ο ίδιος κανόνας με τα ίδια δεδομένα για δεύτερη συνεχόμενη φορά
 - ☐ Ανάλυση μέσων-σκοπών (*means-ends analysis*)
 - ✓ Προτεραιότητα έχουν οι κανόνες των οποίων η πρώτη συνθήκη, η οποία υποδηλώνει τον στόχο με τον οποίο σχετίζεται ο κανόνας, είναι πιο πρόσφατη
 - ☐ Μετα-έλεγχος (μετα-κανόνες)



Κύκλος Λειτουργίας Συστήματος Παραγωγής

Έως ότου δε μπορεί να εκτελεστεί κανένας κανόνας επανέλαβε:

1. Βρες όλους του κανόνες που ενεργοποιούνται και σχημάτισε το σύνολο σύγκρουσης.
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

- ❖ Έστω ότι υπάρχουν στη Μνήμη Εργασίας τα γεγονότα F1, F2 και στη Βάση Γνώσης οι κανόνες R1-R3
- ❖ Με την στρατηγική της αποφυγής επανάληψης
 - ☐ Αν εκτελεστεί πρώτα (στον 1^ο κύκλο) ο κανόνας R1, μετά (στον 2^ο κύκλο) θα εκτελεστεί ο κανόνας R2 και όχι ο κανόνας R1 αν και εξακολουθεί να είναι ενεργοποιημένος
 - ☐ Για να σταματήσει να είναι ενεργοποιημένος πρέπει κάποιος κανόνας να σβήσει το A από τη Μνήμη Εργασίας.
- ❖ Με την στρατηγική της επιλογής του πιο πρόσφατου
 - ☐ Αν εκτελεστεί (για κάποιο λόγο) πρώτα ο R1 στη συνέχεια, θα εκτελεστεί ο κανόνας R3 και όχι ο R2, γιατί το C είναι πιο πρόσφατο από το B.

F1: A

F2: B

R1: if A then addwm(C)

R2: if B then addwm(D)

R3: if C then addwm(E)



Παράδειγμα

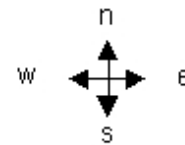
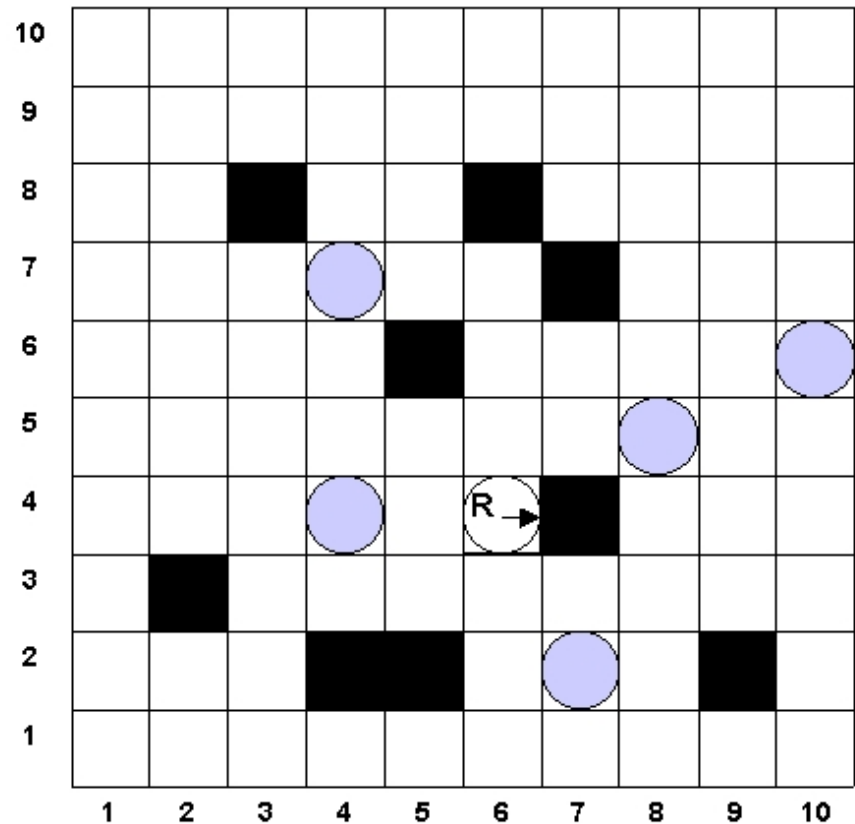
IF: Company is customer, and
Company has good Credit, and
Company is over limit by small amount, or
Company is very large customer

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

Σταθερά

}
- ❖ Οι **ενέργειες** των κανόνων εμπεριέχουν τέσσερις λειτουργίες:
 - ☐ `addwm`: βάλει 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



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 (ΑΕ),
- ☐ επιλογή του πιο ειδικού (ΕΕ), και
- ☐ τυχαία επιλογή (ΤΕ).

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

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



Συνδυασμός Κανόνων με Πλαίσια

- ❖ Ένα σύστημα γνώσης μπορεί να χρησιμοποιεί πολλαπλές μεθόδους αναπαράστασης γνώσης, οι οποίες συνδυάζονται για την καλύτερη αναπαράσταση ενός προβλήματος και της διαδικασίας επίλυσής του.
- ❖ Ένας τέτοιος συνηθισμένος συνδυασμός είναι οι κανόνες με τα πλαίσια
 - ☐ που χρησιμεύουν στην στατική αναπαράσταση των αντικειμένων ενός προβλήματος και των συσχετίσεών τους, ενώ
 - ☐ οι κανόνες αναπαριστούν την συλλογιστική για την επίλυση του προβλήματος.
- ❖ Έτσι, οι συνθήκες των κανόνων αναφέρονται σε συγκεκριμένα πλαίσια ή σε τιμές των σχισμών των πλαισίων ενώ τα συμπεράσματα που εξάγονται,
 - ☐ στους συμπερασματικούς κανόνες, αφορούν σε τιμές των σχισμών, ενώ
 - ☐ οι ενέργειες των κανόνων παραγωγής αφορούν είτε σε αλλαγές (προσθήκες/διαγραφές/μεταβολές) των τιμών των σχισμών ή σε δημιουργία νέων /διαγραφή πλαισίων.
- ❖ Παράδειγμα συμπερασματικού R1 και κανόνα παραγωγής R'1
 - ☐ R1: if S isa student[attends:C] and C isa course[teacher:P] then S[knows:P]
 - ✓ ο οποίος ορίζει ότι κάποιος φοιτητής S ξέρει έναν καθηγητή P όταν ο τελευταίος διδάσκει το μάθημα C που παρακολουθεί ο φοιτητής
 - ☐ R'1: if S isa student[attends:C] and C isa course[teacher:P] then addwm(S[knows:P])



Ερωτήσεις

- ❖ Όταν ένας κανόνας σε ένα Σύστημα Γνώσης πυροδοτείται (fires) τότε
- ❖ Οι κανόνες της μορφής if συνθήκες then ενέργειες, εκφράζουν γνώση
- ❖ Η επιλογή των κανόνων προς πυροδότηση γίνεται από την στρατηγική
- ❖ Ένας κανόνας σε ένα Σύστημα Γνώσης οπλίζει (triggers) όταν
- ❖ Αναφέρετε τέσσερις στρατηγικές επίλυσης συγκρούσεων
- ❖ Τι είναι η ανάστροφη ακολουθία εκτέλεσης κανόνων;
- ❖ Τι είναι η ορθή ακολουθία εκτέλεσης κανόνων;
- ❖ Περιγράψτε τα βήματα που ακολουθούνται στον κύκλο λειτουργίας ενός συστήματος παραγωγής
- ❖ Σε ένα σύνολο κανόνων η γνώση μπορεί να είναι οργανωμένη με ανάστροφη συλλογιστική (backward reasoning), όταν.....
 - ❑ Τα συστήματα που αποτελούνται από τέτοιους κανόνες ονομάζονται και συστήματα
- ❖ Το σύνολο των κανόνων που οπλίζουν (σε μία δεδομένη χρονική στιγμή) σχηματίζουν το
- ❖ Ένας κανόνας οπλίζει (triggers) όταν
- ❖ Όταν ένας κανόνας πυροδοτείται (fires) τότε



Άσκηση 1

❖ Έστω το παρακάτω πρόγραμμα σε CLIPS:

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

```
(defrule rule2  
  (declare (salience 0))  
  (data ?x)  
=>  
  (printout t (* 2 ?x) crlf)  
)  
(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)
```



Άσκηση 2

❖ 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

❖ Ερωτήσεις

- ☐ 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.



Άσκηση 3

❖ 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

❖ Ερωτήσεις

- ☐ 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;



Άσκηση 4

❖ 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;



Άσκηση 5

❖ Έστω το ακόλουθο πρόγραμμα CLIPS.

```
(deffacts elevetor-move  
  (elevator on ground)  
  (elevator ON))
```

```
(defrule initial-rule  
  =>  
  (assert (goto floorA)))
```

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

```
(defrule empty-elevetor  
  (elevator on floorA)  
  =>  
  (assert (empty elevetor))  
  (assert (goto ground)))
```

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