

Κεφάλαιο 11

Συστήματα Κανόνων

Τεχνητή Νοημοσύνη - Β' Έκδοση

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



Αναπαράσταση με Κανόνες

- ❖ Πολύ πρακτικός τρόπος αναπαράστασης για την εξαγωγή συμπερασμάτων
- ❖ Αποτελούν τη βάση πολλών συστημάτων γνώσης (knowledge systems)
- ❖ Γενικά Πλεονεκτήματα:
 - ❑ Κοντά στην ανθρώπινη γνώση
 - ❑ Επάρκεια συνεπαγωγών
- ❖ Συγκεκριμένα Πλεονεκτήματα:
 - ❑ *Modularity*: Κάθε κανόνας ορίζει ένα μικρό, (σχεδόν) ανεξάρτητο τμήμα της γνώσης
 - ❑ *Incrementability*: Μπορούν να προστεθούν νέοι κανόνες (σχεδόν) ανεξάρτητα από τους υπάρχοντες
 - ❑ *Modifiability*: Οι υπάρχοντες κανόνες μπορούν να αλλάξουν (σχεδόν) ανεξάρτητα από τους υπόλοιπους

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 (συμβάν) Αν οι συνθήκες αληθεύουν τότε εκτέλεσε τις ενέργειες

- ❖ **Συνθήκες** (conditions): ακολουθία από κατηγορήματα που συνδέονται με λογικούς τελεστές AND, OR
 - Προϋποθέσεις (premises) ή αριστερό μέρος του κανόνα (left hand side - LHS)
 - ❖ **Συμπέρασμα** (conclusion): κατηγορημα
 - ❖ **Ενέργειες** (actions): μία σειρά από εντολές που πρέπει να εκτελεστούν
- Επακόλουθα (consequent) ή δεξιό μέρος του κανόνα (right hand side -



Συστήματα Κανόνων

- ❖ Συστήματα εξαγωγής συμπερασμάτων (deduction systems)
 - ❑ π.χ. Prolog, Datalog
 - ❑ Γνώση που δηλώνει μία αλήθεια για τον κόσμο, 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
 - ❑ Οι εκτελούμενες ενέργειες επιφέρουν μη-αναστρέψιμα αποτελέσματα
- ❖ Ενεργά Συστήματα ((re-)active systems, active database systems)
 - ❑ π.χ. Oracle Triggers, Δαίμονες πλαισίων Flex

Ενεργοί Κανόνες

- ❖ Οι κανόνες παραγωγής δηλώνουν διαδικαστική γνώση
 - ❑ Οι κανόνες εκτελούνται "όταν η συνθήκη είναι αληθής"
 - ❑ Δεν είναι σαφώς ορισμένο πότε ακριβώς εκτελούνται οι ενέργειες
 - ❑ 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)
- ❖ 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

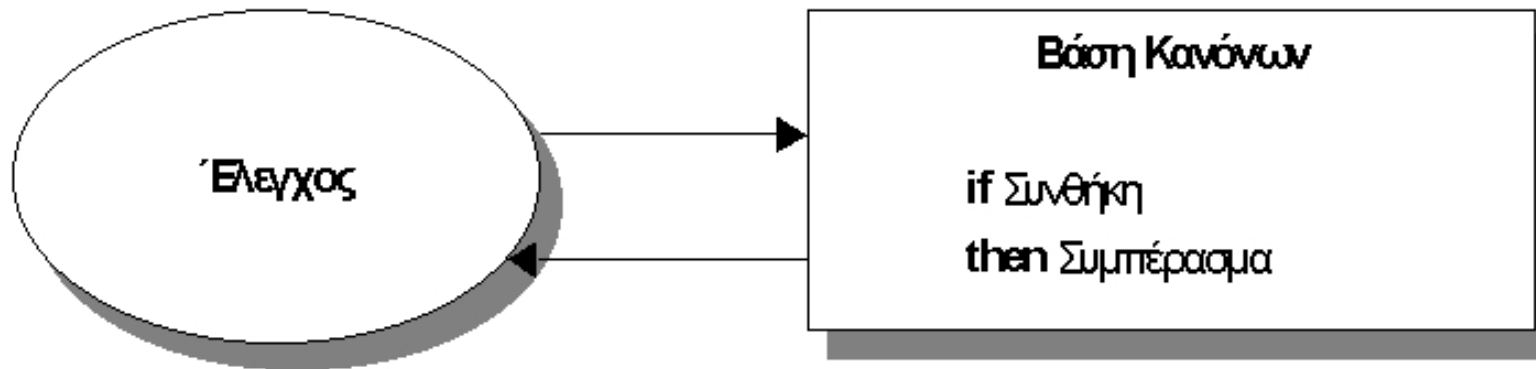
IF ο εκτυπωτής τυπώνει σωστά **AND**
τα χρώματα δε τυπώνονται σωστά
THEN αλλάξτε την κεφαλή με το έγχρωμο μελάνι

❖ Ενεργός Κανόνας

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

Συστήματα Εξαγωγής Συμπερασμάτων

Deduction Systems



- ❖ **Βάση κανόνων** (rule base): περιέχει ένα σύνολο από κανόνες
- ❖ **Έλεγχος** (control): καθορίζει τον τρόπο εκτέλεσης των κανόνων για την εξαγωγή συμπερασμάτων
 - ❑ Ποιοι κανόνες είναι υποψήφιοι για να επιλύσουν το πρόβλημα;
 - ❑ Με ποιόν τρόπο θα γίνει η επιλογή;
 - ❑ Ποιος από τους κανόνες τελικά θα επιλεγεί;
 - ❑ Τι θα γίνει με τους υπόλοιπους κανόνες;

Εξαγωγή Συμπερασμάτων

- ❖ Χρησιμοποιείται η συνεπαγωγική συλλογιστική.
 - ❑ Υλοποιείται από την *ακολουθία εκτέλεσης κανόνων (chaining)*
 - ❑ Αλγόριθμος που συνδυάζει τα δεδομένα, τους κανόνες και τα ενδιάμεσα συμπεράσματα
- ❖ *Ορθή ακολουθία εκτέλεσης (forward chaining)*
 - ❑ Εξετάζονται 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
 - ❑ Μπορεί να συμπεράνει περισσότερα συμπεράσματα από τα επιθυμητά
 - ❑ Ενδείκνυται όταν υπάρχουν λίγα δεδομένα που οδηγούν σε πολλά συμπεράσματα
 - ❑ **Εφαρμογές:** Συστήματα Διάγνωσης, Συστήματα Παραγωγής
- ❖ *Ανάστροφη ακολουθία εκτέλεσης (backward chaining)*
 - ❑ 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 οι μη-αληθείς) έως ότου αποδειχθεί η αλήθεια του συμπεράσματος (π.χ. Prolog).
 - ❑ Ενδείκνυται όταν υπάρχουν λίγα συμπεράσματα και πολλά δεδομένα
 - ❑ Το σύστημα ζητά τα δεδομένα με λογική σειρά και μόνο 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) .

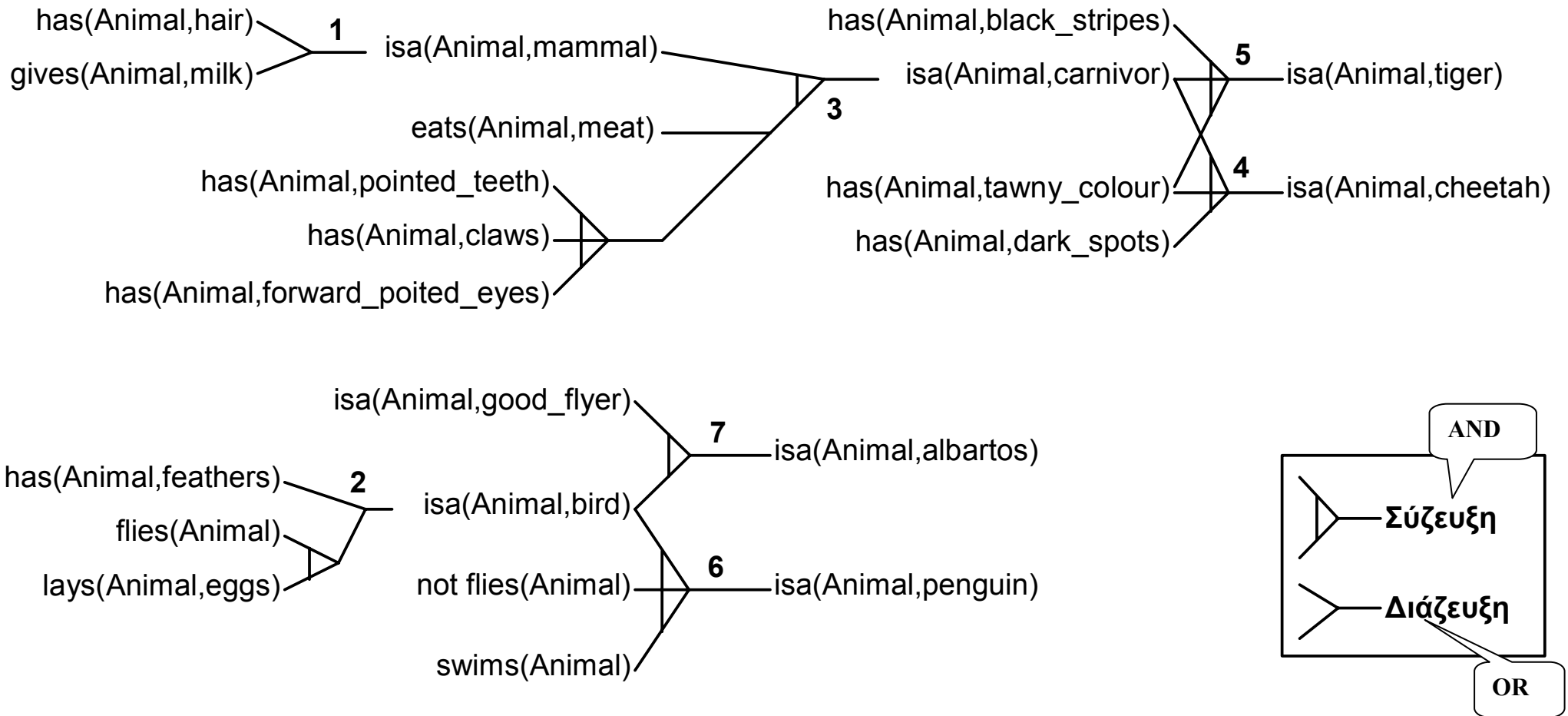
4: if isa(Animal, carnivore) and
has(Animal, tawny_colour) and
has(Animal, dark_spots)
then isa(Animal, cheetah) .

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

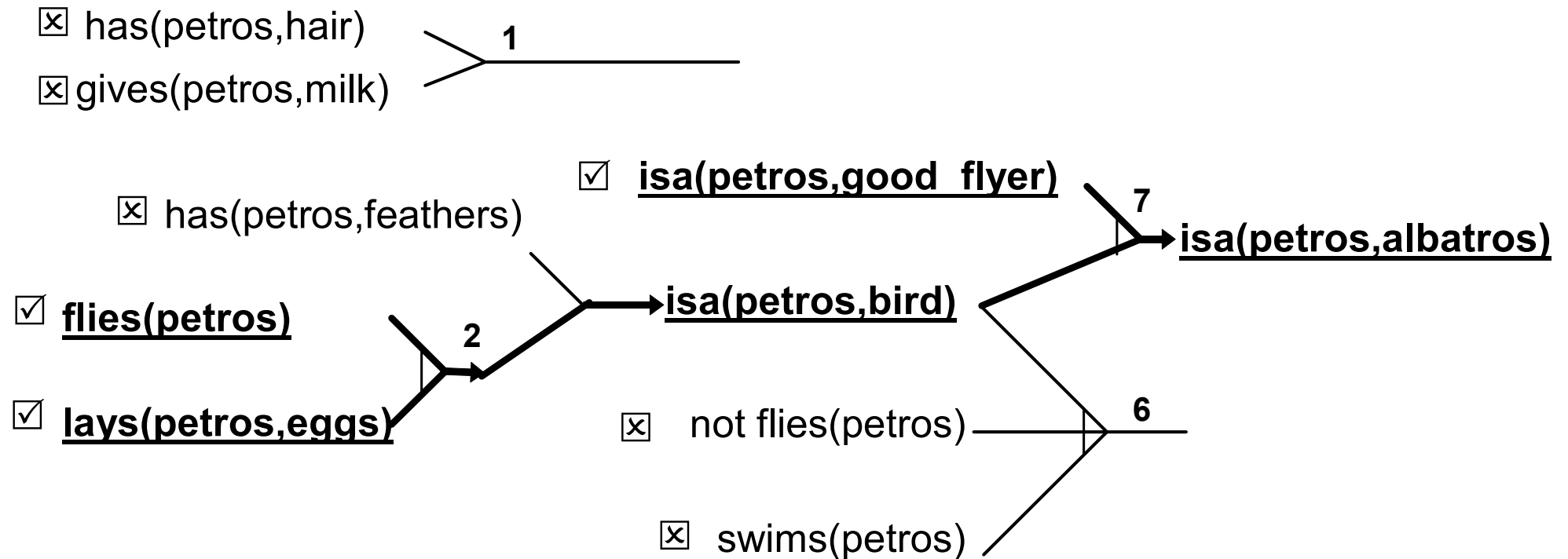


❖ Μπορεί να χρησιμοποιηθεί ορθή ή ανάστροφη ακολουθία εκτέλεσης

Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

Ορθή Ακολουθία Εκτέλεσης

❖ Αρχικά δεδομένα: **flies(petros)**, **lays(petros, eggs)**, **isa(petros, goodflyer)**.

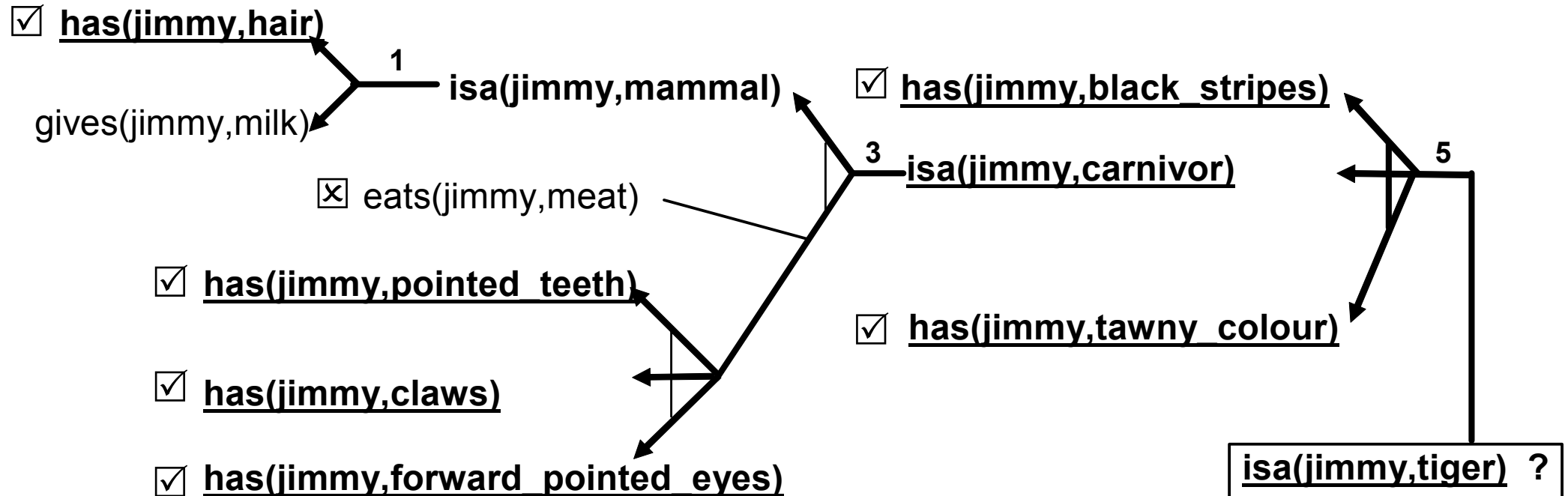


❖ Παράγονται τα συμπεράσματα: **isa(petros, albatros)** και **isa(petros, bird)**

Γραφική Αναπαράσταση Εξαγωγής Συμπεράσματος

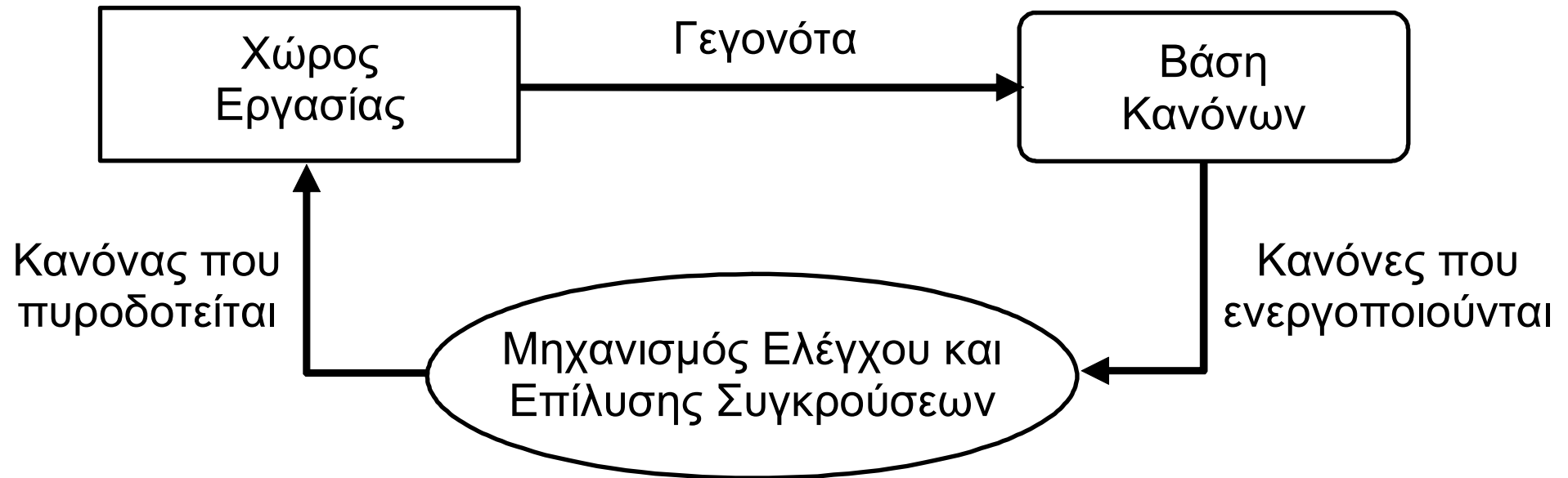
Ανάστροφη Ακολουθία Εκτέλεσης

❖ Ερώτηση: `isa(jimmy, tiger)`



❖ Απάντηση: Yes

Συστήματα Παραγωγής



- ❖ **Βάση κανόνων:** περιέχει τους κανόνες παραγωγής
- ❖ **Χώρος εργασίας (working memory):** περιέχει γεγονότα
 - ❑ Αρχικά δεδομένα (data) ή ενδιάμεσα συμπεράσματα (partial conclusions)
 - ❑ Στοιχεία της μνήμης εργασίας (working memory elements)
- ❖ **Μηχανισμός ελέγχου (control ή scheduler) και επίλυσης συγκρούσεων (conflict resolution):** εκτέλεση των κανόνων βάσει στρατηγικής επίλυσης συγκρούσεων (conflict resolution strategy)

Επίλυση Συγκρούσεων

- ❖ Ο κανόνας **ενεργοποιείται** (triggers) όταν ικανοποιούνται οι συνθήκες του
- ❖ Όταν **πυροδοτείται** (fires) ο κανόνας, τότε εφαρμόζονται/εκτελούνται οι ενέργειές του
- ❖ **Σύνολο σύγκρουσης** (conflict set): Το σύνολο των κανόνων που ενεργοποιούνται
- ❖ Ο μηχανισμός ελέγχου καθορίζει ποιος κανόνας θα πυροδοτηθεί
- ❖ Στρατηγικές επίλυσης συγκρούσεων:
 - ❑ 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)

Στρατηγικές Επίλυσης Συγκρούσεων (1/4)

Τυχαία (random)

- ❖ Επιλέγεται ένας κανόνας στην τύχη.
- ❖ Συνήθως δεν υπάρχει στα συστήματα κανόνων παραγωγής
 - ❑ Αν υπάρχει δεν είναι πραγματικά τυχαία η επιλογή
 - ❑ Είναι απροσδιόριστη - εξαρτάται από τις λεπτομέρειες της υλοποίησης

Διάταξη (ordering)

- ❖ Επιλέγεται ο κανόνας που είναι πρώτος στη σειρά ή έχει μεγαλύτερη προτεραιότητα βάσει κάποιου αριθμητικού μεγέθους

Αποφυγή επανάληψης (refractoriness)

- ❖ Δεν επιλέγεται ο ίδιος κανόνας με τα ίδια δεδομένα για δεύτερη συνεχόμενη φορά
- ❖ Αποφεύγονται άσκοπες ή ατέρμονες επαναλήψεις
- ❖ *Παράδειγμα:*
 - ❑ Υπάρχουν τα γεγονότα A, B και οι κανόνες:
1: **if A then C** 2: **if B then D**
 - ❑ Αν εκτελεστεί πρώτα ο 1, μετά θα εκτελεστεί ο 2
 - ❑ Ο 1 δε θα εκτελεστεί ξανά, αν και εξακολουθεί να είναι ενεργοποιημένος

Στρατηγικές Επίλυσης Συγκρούσεων (3/4)

Επιλογή του πιο ειδικού (specificity)

- ❖ Επιλέγεται ο κανόνας που είναι πιο ειδικός από τους άλλους
 - ❑ Η συνθήκη του εκφράζεται με αναλυτικότερο τρόπο
- ❖ Εξετάζονται πρώτα τα πιο συγκεκριμένα θέματα τα οποία οδηγούν πιθανότατα σε λύση πιο γρήγορα και στη συνέχεια τα πιο γενικά.
- ❖ *Παράδειγμα:*
 - ❑ Υπάρχουν τα γεγονότα A, B, C και οι κανόνες
 - 1: if A and B and C then D
 - 2: if A and B then E
 - ❑ Θα εκτελεστεί πρώτα ο 1 γιατί έχει πιο πολλές συνθήκες από τον 2.

Μετα-έλεγχος

- ❖ Τα συστήματα παραγωγής εφαρμόζουν μία ή περισσότερες στρατηγικές επίλυσης συγκρούσεων
- ❖ Όταν υπάρχουν πολλές στρατηγικές, πρέπει να υπάρχει προτεραιότητα μεταξύ τους
- ❖ *Μετα-έλεγχος (meta-control)*: καθορίζει ποια στρατηγική θα εφαρμοστεί, πού και πότε
- ❖ *Απλά συστήματα*: σταθερή προτεραιότητα
 - ❑ Χαμηλότερη τιμή στην τυχαία επιλογή
- ❖ *Σύνθετα συστήματα*: η προτεραιότητα αλλάζει δυναμικά (at run-time)
 - ❑ *Μετα-κανόνες (meta-rules)*: κανόνες που καθορίζουν τη σειρά εκτέλεσης άλλων κανόνων

Κύκλος Λειτουργίας Συστήματος Παραγωγής

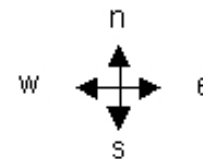
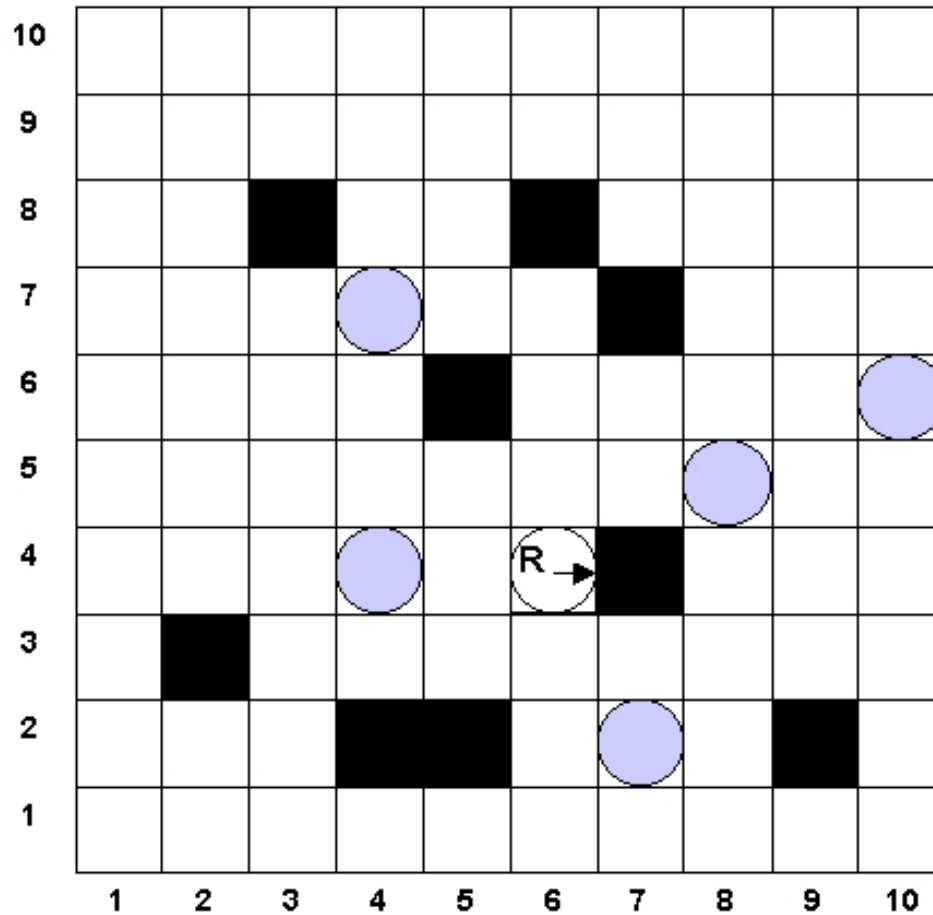
Έως ότου δε μπορεί να εκτελεστεί κανένας κανόνας επανέλαβε:

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.

- ❖ Στα συστήματα παραγωγής υπάρχει ορθή ακολουθία εκτέλεσης κανόνων
- ❖ Εξαγωγή συμπερασμάτων:
 - Δεν έχει νόημα
 - Οι κανόνες παραγωγής αναφέρονται σε ενέργειες που εκτελούνται και όχι σε συμπεράσματα
- ❖ Η λειτουργία των κανόνων παραγωγής "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 Κίνησης Ρομπότ

- ❖ Ρομπότ κινείται σε χώρο με εμπόδια
- ❖ Στόχος: 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

❖ Μνήμη εργασίας:

- ❑ θέση ρομπότ: **robot_at**(X, Y)
- ❑ κατεύθυνση προς την οποία κινείται: **direction**(D), $D \in \{e, w, n, s\}$
- ❑ θέση εμποδίων: **obstacle_at**(X, Y)
- ❑ θέση αντικειμένων: **object_at**(X, Y)
- ❑ επιλογή κατεύθυνσης: **choice**(D), $D \in \{e, w, n, s\}$

Σταθερά

❖ *Ενέργειες* κανόνων:

- ❑ **addwm**: βάλε κάτι στη μνήμη εργασίας
- ❑ **delwm**: σβήσε κάτι από τη μνήμη εργασίας
- ❑ **output**: εκτύπωσε ένα μήνυμα στην οθόνη
- ❑ αριθμητικές εκφράσεις.

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

- ❖ Αποφυγή Επανάληψης (ΑΕ)
 - ❑ Βοηθά να μην κολλήσει το ρομπότ σε εμπόδιο, επιλέγοντας συνεχώς την κατεύθυνση προς την οποία βρίσκεται το εμπόδιο
- ❖ Επιλογή του πιο Ειδικού (ΕΕ)
 - ❑ Δίνει προτεραιότητα στην αποφυγή εμποδίων (κανόνες 6 - 9)
- ❖ Τυχαία Επιλογή (ΤΕ)
 - ❑ Το ρομπότ επιλέγει τυχαία μία από τις 4 κατευθύνσεις

Παράδειγμα Κίνησης Ρομπότ

Παρακολούθηση Εκτέλεσης (1/3)

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>object_at(4,7)</code> . . .	<code>{3,</code> <code>9 (ND=w) ,</code> <code>9 (ND=n) ,</code> <code>9 (ND=s) ,</code> <code>9 (ND=e) }</code>	EE TE	<code>9:avoid_obstacle_east</code> <code>(ND=n)</code>
2	<code>robot_at(6,4)</code> <code>direction(n)</code> . . .	<code>{4}</code>	-	<code>4: move_north</code>
3	<code>robot_at(6,5)</code> <code>direction(n)</code> . . .	<code>{4}</code>	-	<code>4: move_north</code>

Κανόνες που σπλίζουν 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)`

Παράδειγμα Κίνησης Ρομπότ

Παρακολούθηση Εκτέλεσης (2/3)

Κύκλος	Μνήμη Εργασίας	Σύνολο Συγκρούσεων	Στρατηγική	Κανόνας που πυροδοτεί
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)

Παράδειγμα Κίνησης Ρομπότ

Παρακολούθηση Εκτέλεσης (3/3)

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