ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

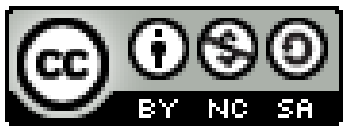
ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Συστήματα Γνώσης

Θεωρητικό Κομμάτι Μαθήματος

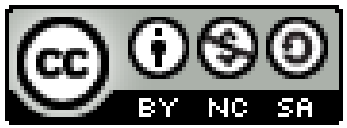
Ενότητα 4: Αναπαράστασης Γνώσης και Συλλογιστικής – Συστήματα Κανόνων

Νίκος Βασιλειάδης, Αναπλ. Καθηγητής
Τμήμα Πληροφορικής



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.

Αναπαράστασης Γνώσης και Συλλογιστικής

Συστήματα Κανόνων

Αναπαράσταση με Κανόνες

- Οι κανόνες είναι από τις πιο προσφιλείς μεθόδους αναπαράστασης γνώσης
  - Ο πρακτικότερος τρόπος αναπαράστασης για την εξαγωγή συμπερασμάτων.
- Τα συστήματα κανόνων αποτελούν τη βάση πολλών έμπειρων συστημάτων και συστημάτων γνώσης
- Γενικά Πλεονεκτήματα:
  - Η γνώση αναπαριστάται με τρόπο που πλησιάζει την ανθρώπινη για τις περισσότερες δραστηριότητες που απαιτούν νοημοσύνη
  - Η εξαγωγή συμπερασμάτων γίνεται με εύκολο τρόπο (επάρκεια συνεπαγωγών).

Αναπαράσταση με Κανόνες

- Οι κανόνες είναι από τις πιο προσφιλείς μεθόδους αναπαράστασης γνώσης
  - Ο πρακτικότερος τρόπος αναπαράστασης για την εξαγωγή συμπερασμάτων.
- Τα συστήματα κανόνων αποτελούν τη βάση πολλών έμπειρων συστημάτων και συστημάτων γνώσης
- Γενικά Πλεονεκτήματα:
  - Η γνώση αναπαριστάται με τρόπο που πλησιάζει την ανθρώπινη για τις περισσότερες δραστηριότητες που απαιτούν νοημοσύνη
  - Η εξαγωγή συμπερασμάτων γίνεται με εύκολο τρόπο (επάρκεια συνεπαγωγών).

Αναπαράσταση με Κανόνες
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

Είδος Κανόνα	Μορφή Κανόνα	Εκφράζει	Επεξήγηση
Συνεπαγωγικός κανόνας Deductive rule	IF συνθήκες THEN συμπέρασμα	Δηλωτική γνώση	**Αν** οι συνθήκες αληθεύουν **τότε** αληθεύει και το συμπέρασμα
Κανόνας Παραγωγής Production rule	IF συνθήκες THEN ενέργειες	Διαδικαστική γνώση	**Αν** οι συνθήκες αληθεύουν **τότε** εκτέλεσε τις ενέργειες
Ενεργός κανόνας Active rule ECA rule (*Event-Condition-Action*)	ON συμβάν IF συνθήκες THEN ενέργειες	Διαδικαστική γνώση	**Όταν** συμβεί το γεγονός (συμβάν) **Αν** οι συνθήκες αληθεύουν **τότε** εκτέλεσε τις ενέργειες

Τμήματα του κανόνα

- Οι ***συνθήκες*** (***conditions***) είναι μία ακολουθία από ***κατηγορήματα*** (***predicates***) τα οποία συνδέονται μεταξύ τους με τους λογικούς τελεστές AND/OR.
  - Αναφέρονται και ως ***προϋποθέσεις*** (***premises***) ή αριστερό μέρος του κανόνα (left hand side - LHS).
  - Ερμηνεύονται και ως ερωτήματα (queries) προς την τρέχουσα κατάσταση της βάσης δεδομένων ή γνώσης.
- Το ***συμπέρασμα*** (***conclusion***) είναι ένα κατηγόρημα.
  - Ερμηνεύεται και ως ένα νέο γεγονός (fact) που πρέπει να προστεθεί στη βάση γνώσης, γιατί αληθεύει.
- 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, Datalog, OOJDrew
  - 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, Jess, Drools, 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 databases):
  - π.χ. Oracle Triggers, Rulecore (Open Source), IBM Amit, Drupal Rules Module, Δαίμονες πλαισίων Flex

Ενεργοί Κανόνες vs. Κανόνες Παραγωγής

- 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)

Ενεργοί Κανόνες

- 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 δεν τυπώνονται σωστά	Έχει τελειώσει το έγχρωμο μελάνι	Αλλάξτε την κεφαλή με το έγχρωμο μελάνι

Συνεπαγωγικός Κανόνας	Κανόνας Παραγωγής	Ενεργός Κανόνας
IF ο εκτυπωτής τυπώνει σωστά **AND** τα χρώματα δεν τυπώνονται σωστά **THEN** έχει τελειώσει το έγχρωμο μελάνι	**IF** ο εκτυπωτής τυπώνει σωστά **AND** τα χρώματα δεν τυπώνονται σωστά **THEN** αλλάξτε την κεφαλή με το έγχρωμο μελάνι	**ON** εκτύπωση **IF** τα χρώματα δεν τυπώνονται σωστά **THEN** αλλάξτε την κεφαλή με το έγχρωμο μελάνι

Παράδειγμα Ενεργών Κανόνων

Η διαδικτυακή υπηρεσία IFTTT

What is IFTTT? IFTTT is a service that lets you create powerful connections with one simple statement:

Recipe

if this then that

Trigger Action

IFTTT is pronounced like “gift” without the “g.”

Channels Channels are the basic building blocks of IFTTT. Each Channel has its own Triggers and Actions. Some example Channels are:

Παράδειγμα Ενεργών Κανόνων

Η διαδικτυακή υπηρεσία IFTTT

Triggers	The **this** part of a Recipe is a Trigger. Some example Triggers are “I’m tagged in a photo on Facebook” or “I check in on Foursquare.”
Actions	The **that** part of a Recipe is an Action. Some example Actions are “send me a text message” or “create a status message on Facebook.”
Ingredients	Pieces of data from a Trigger are called Ingredients. For example, the Ingredients of an Email Trigger could be: subject, body, attachment, received date, and the sender’s address.

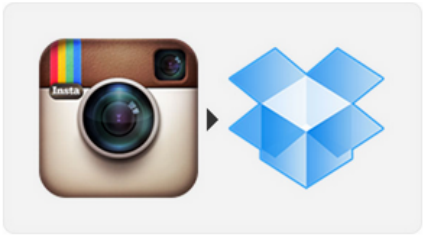
Παράδειγμα Ενεργών Κανόνων

Η διαδικτυακή υπηρεσία IFTTT

Recipes

Personal Recipes are a combination of a Trigger and an Action from your active Channels. Personal Recipes look like this:

Shared Recipes are useful templates shared by the IFTTT community. Shared Recipes look like this:



Autosave all your Instagram photos to Dropbox

by Linden on 22 Nov 2011
used 9183 times

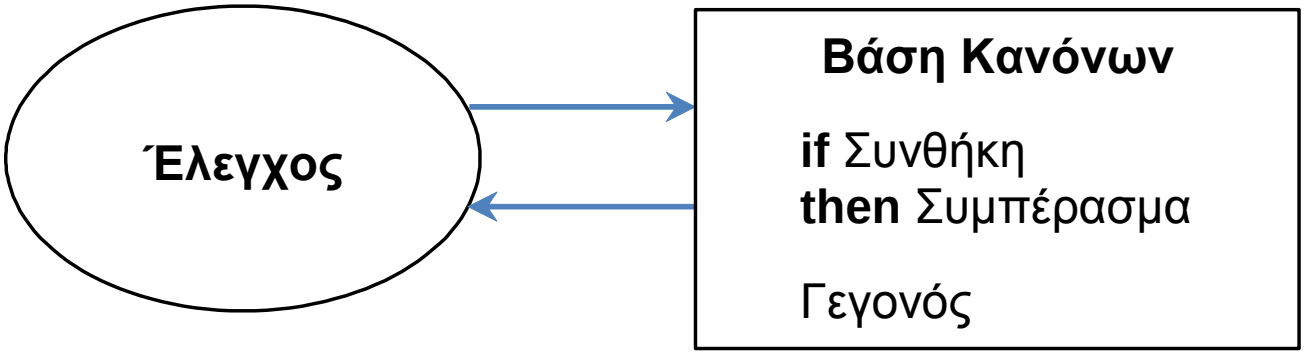
Παράδειγμα Ενεργών Κανόνων
Η διαδικτυακή υπηρεσία IFTTT

On / Off	Personal Recipes can be turned on and off. When turned back on, they pick up as if you had just created them.
Polling Period	Most Personal Recipes check for new Trigger data every 15 minutes, some are even faster.

https://ifttt.com

Συστήματα Εξαγωγής Συμπερασμάτων
Deduction Systems

- Αποτελούνται από:
  - Τη βάση κανόνων (rule base)
  - Τον έλεγχο (control)



Βάση Κανόνων

- Περιέχει ένα σύνολο από κανόνες.
- Περιέχονται και τα δεδομένα του προγράμματος υπό τη μορφή γεγονότων (facts), που μπορούν να θεωρηθούν κανόνες χωρίς συνθήκη (πάντα αληθείς).

Έλεγχος (control)

- Ο έλεγχος καθορίζει τον τρόπο με τον οποίο θα εκτελεστούν οι κανόνες για να εξαχθούν τα συμπεράσματα.
  - Π.χ. στην Prolog ο έλεγχος είναι ο depth-first μηχανισμός ταυτοποίησης.
- Είναι ένας αλγόριθμος που αποφασίζει
  - Ποιοι από τους κανόνες/γεγονότα είναι υποψήφιοι για να επιλύσουν το πρόβλημα;
  - Με ποιόν τρόπο θα γίνει η επιλογή;
  - Ποιος από τους κανόνες/γεγονότα τελικά θα επιλεγεί;
  - Τι θα γίνει με τους υπόλοιπους κανόνες/γεγονότα;

Έλεγχος στην Prolog

- Ποιοι από τους κανόνες/γεγονότα είναι υποψήφιοι για να επιλύσουν το πρόβλημα;
  - Οι κανόνες των οποίων η κεφαλή ταυτοποιείται 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/γεγονότα;
  - Θα παραμείνουν «διαθέσιμοι» ως σημεία οπισθοδρόμησης και θα χρησιμοποιηθούν σε περίπτωση αποτυχίας

Έλεγχος

- Η επίλυση του προβλήματος ανάγεται σε πρόβλημα αναζήτησης της λύσης
  - Εύρεση της **ακολουθίας** κανόνων/γεγονότων που λύνουν το πρόβλημα
- Ο έλεγχος ουσιαστικά υλοποιεί τη συλλογιστική.
  - Στα Συστήματα Εξαγωγής Συμπερασμάτων χρησιμοποιείται η ***Συνεπαγωγική συλλογιστική***
  - Υλοποιείται με 2 τρόπους ή ***ακολουθίες εκτέλεσης (chaining)***
    - Αλγόριθμοι με τους οποίους συνδυάζονται τα δεδομένα, οι κανόνες και τα ενδιάμεσα συμπεράσματα

Ακολουθία Εκτέλεσης (Chaining)

- ***Ανάστροφη ακολουθία εκτέλεσης***
  - *Backward chaining*
  - IF A THEN B
    - Ισχύει το Β? Πρέπει να αποδείξω το Α.
    - Αν ισχύει το Α, τότε ισχύει και το Β.
    - Αν όχι, τότε πρέπει να ψάξω και άλλο.
  - Από τα δεξιά προς τα αριστερά
- ***Ορθή ακολουθία εκτέλεσης***
  - *Forward chaining*
  - IF A THEN B
    - Ισχύει το Α. Άρα ισχύει το Β.
  - Από τα αριστερά προς τα δεξιά

Ανάστροφη ακολουθία εκτέλεσης

- Η εξαγωγή 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

Ανάστροφη ακολουθία εκτέλεσης

R1: IF A THEN B

R2: IF C THEN B

R3: IF D THEN B

R4: IF D THEN W

} ***Κανόνες***

D ***Γεγονός***

- Ισχύει το B?
- Θα εξεταστούν και οι 3 κανόνες R1, R2, R3
  - Μόνο ο R3 δίνει θετικό αποτέλεσμα.
- Ασχολείται μόνο με τον προς απόδειξη στόχο και τους αντίστοιχους κανόνες
  - Δεν ασχολείται με τον R4 κανόνα, παρόλο που λογικά είναι ορθός

Ανάστροφη ακολουθία εκτέλεσης

- Ενδείκνυται όταν υπάρχουν λίγα συμπεράσματα και πολλά δεδομένα, για τα οποία το σύστημα μας καθοδηγεί ζητώντας τα με μια λογική σειρά και όσα χρειάζονται.
  - Ισχύει το A?, το C?, το D?
- Εφαρμογές: Συστήματα Ελέγχου Λειτουργίας (Monitoring).

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).

Παράδειγμα

4: **if** isa(Animal,carnivore) **and**

has(Animal,tawny_colour) **and**

has(Animal,dark_spots)

then isa(Animal,cheetah).

5: **if** has(Animal,tawny_colour) **and**

isa(Animal,carnivore) **and**

has(Animal,black_stripes)

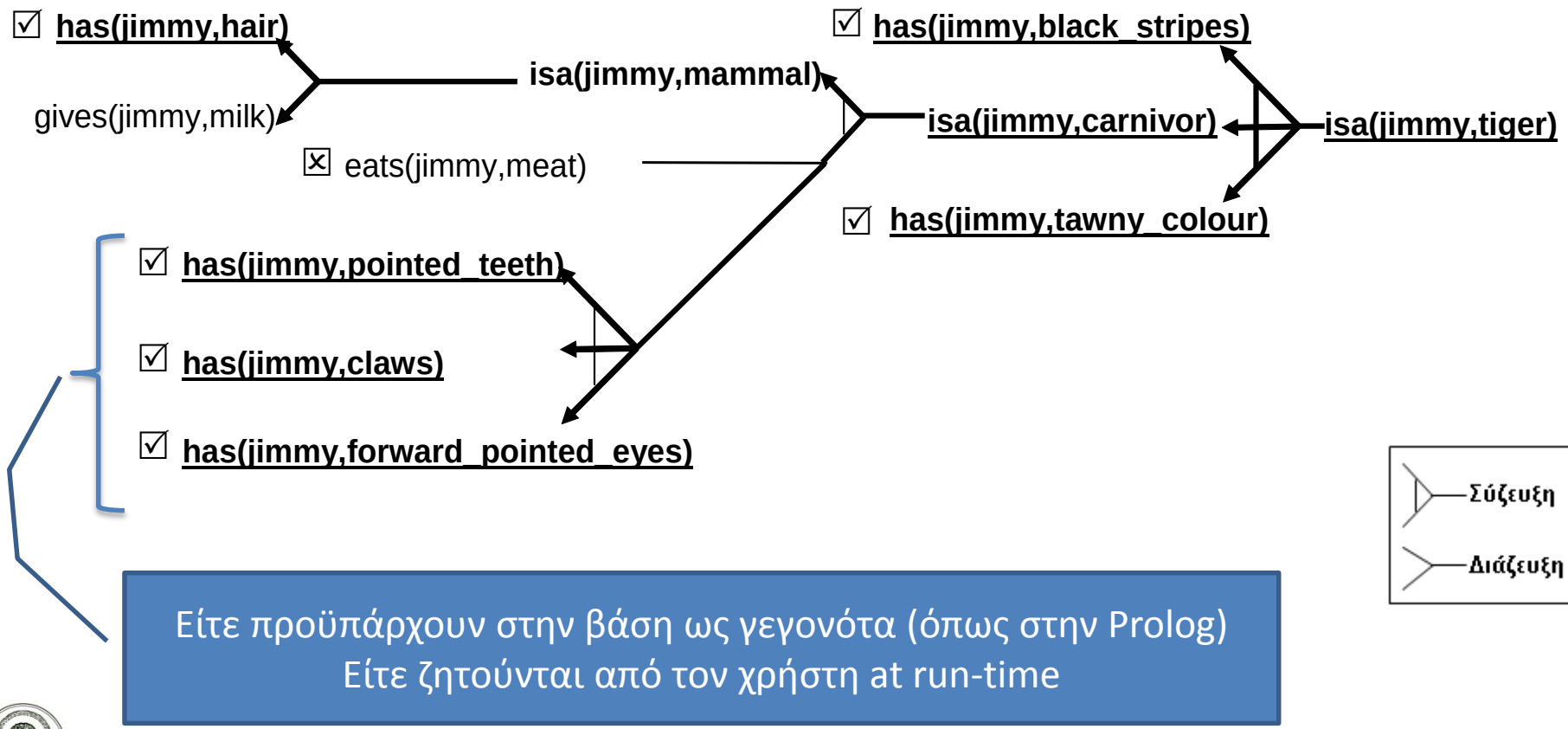
then isa(Animal,tiger).

Παράδειγμα

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).

Απόδειξη του isa(jimmy,tiger) με ανάστροφη ακολουθία εκτέλεσης



Ορθή ακολουθία εκτέλεσης

- Η εξαγωγή 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.

Ορθή ακολουθία εκτέλεσης

R1: IF A THEN B

R2: IF C THEN B

R3: IF D THEN B

R4: IF D THEN W

} *Κανόνες*

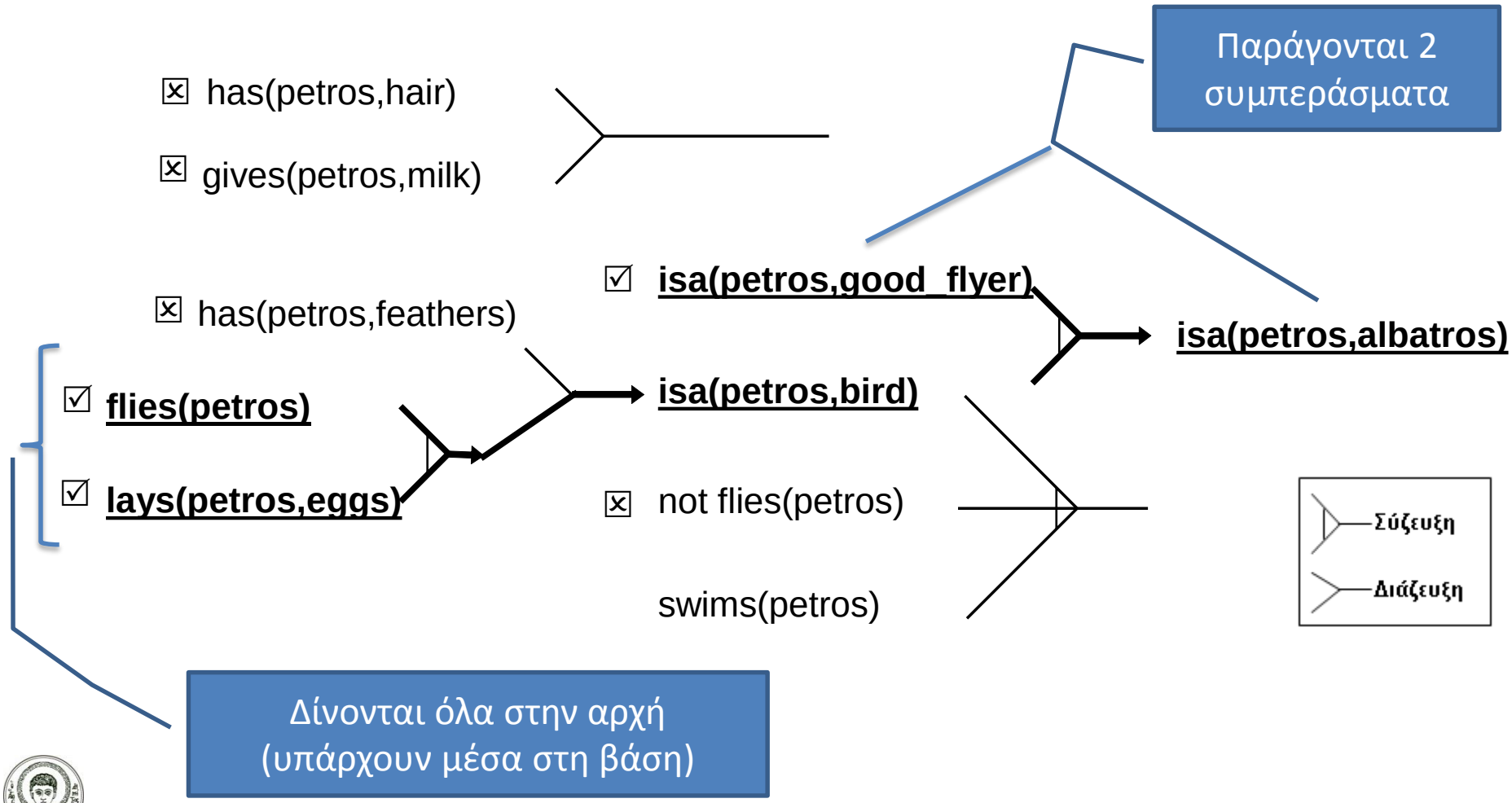
D *Γεγονός*

- Θα εκτελεστούν οι κανόνες R3 και R4
  - Δεν θα ασχοληθεί με κανόνες που δεν δίνουν θετικά αποτελέσματα (R1, R2)
  - Εκτός από το ζητούμενο αποτέλεσμα θα δώσει και «αχρείαστα» αποτελέσματα.
    - Έστω ότι μας ενδιαφέρει μόνο το **B**
    - Από το **D** προκύπτουν τα **B** και **W**

Ορθή ακολουθία εκτέλεσης

- Ενδείκνυται όταν υπάρχουν λίγα δεδομένα (δίδονται στο σύστημα όλα μαζί στην αρχή) και μπορούν να οδηγήσουν σε πολλά συμπεράσματα.
  - Από το D προκύπτουν τα B και W
- Εφαρμογές: Συστήματα Διάγνωσης.

Εξαγωγή συμπερασμάτων με ορθή ακολουθία εκτέλεσης



Σειρά Εκτέλεσης Κανόνων

Ανάστροφη Ακολουθία Εκτέλεσης

- Συνήθως χρησιμοποιείται SLDNF resolution, όπως στην Prolog.
- Αναλύεται ο πρώτος από αριστερά στόχος χρησιμοποιώντας την πρώτη από πάνω-προς-τα-κάτω πρόταση που μπορεί να ενοποιηθεί.
  - Σε περίπτωση αποτυχίας υπάρχει χρονολογική οπισθοδρόμηση (backtracking).
- Χρησιμοποιείται η άρνηση-ως-αποτυχία (negation-as-failure ή default negation) χωρίς ελεύθερες μεταβλητές
  - Δημιουργούνται αποδείξεις, οι οποίες όταν αποτυγχάνουν τότε επιτυγχάνεται η άρνησή τους (και το αντίστροφο)

Σειρά Εκτέλεσης Κανόνων

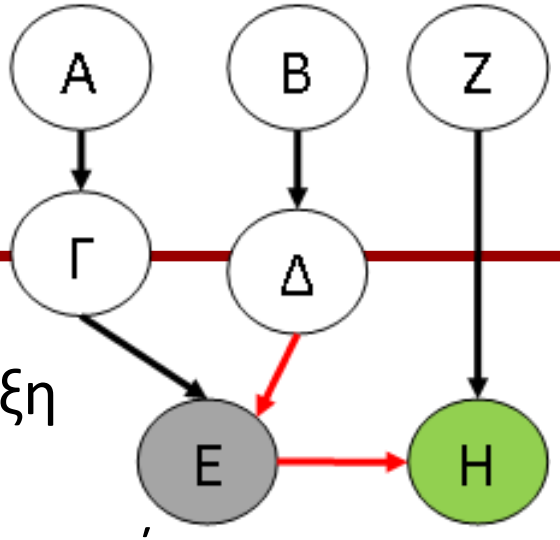
Ορθή Ακολουθία Εκτέλεσης

- Αν δεν υπάρχει άρνηση, τότε δεν παίζει ρόλο η σειρά εκτέλεσης των κανόνων.
  - **Α → Γ, Β → Δ, Γ & Δ → Ε, Α, Β**
  - Με οποιαδήποτε σειρά αν εκτελεστούν οι κανόνες, βγαίνει συμπέρασμα **Ε**.
- Αν υπάρχει άρνηση, η σειρά εκτέλεσης έχει σημασία
  - **Α → Γ, Β → Δ, Γ & not(Δ) → Ε, Α, Β**
  - Αν η σειρά εκτέλεσης είναι **Α → Γ, Β → Δ**, τότε ο κανόνας **Γ & not(Δ) → Ε** δεν εκτελείται και δεν βγαίνει συμπέρασμα **Ε**
  - Αν η σειρά εκτέλεσης είναι **Α → Γ, Γ & not(Δ) → Ε, Β → Δ,** τότε βγαίνει συμπέρασμα **Ε**
- Για να μην δημιουργείται πρόβλημα, η εκτέλεση των κανόνων γίνεται σε «στρώματα» (strata)

Διαστρωμάτωση (Stratification)

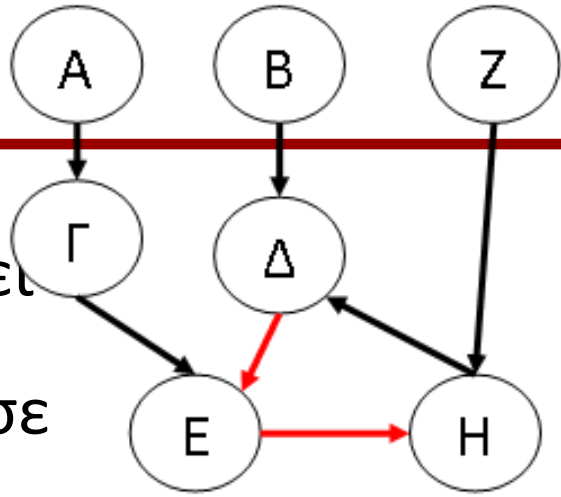
- Συντακτικός περιορισμός που επιβάλλεται στην σειρά εκτέλεσης κανόνων σε ένα σύνολο συμπερασματικών κανόνων με άρνηση.
- Δίνεται προτεραιότητα στην εκτέλεση όλων των κανόνων που παράγουν συμπεράσματα που αφορούν κάποιο κατηγόρημα, το οποίο εμφανίζεται με άρνηση στην συνθήκη κάποιου άλλου κανόνα.
- Π.χ. στο σύνολο κανόνων **Α → Γ, Β → Δ, Γ & not(Δ) → Ε, Α, Β,** επιβάλλεται ο κανόνας **Γ & not(Δ) → Ε** να εκτελεστεί μετά από τον κανόνα **Β → Δ**
  - Έτσι είναι «γνωστά όλα τα Δ» πριν αποφανθεί το σύστημα ότι «δεν υπάρχει Δ» για να εκτελέσει τον κανόνα

Stratification



- Το σύνολο των κανόνων χωρίζεται σε «στρώματα» (strata) ανάλογα με την ύπαρξη άρνησης στη συνθήκη τους.
  - Η εκτέλεση ξεκινάει από τα χαμηλά στρώματα κανόνων και προχωράει σε υψηλότερα στρώματα (αυξανόμενο κατά 1) όταν το προηγούμενο στρώμα δεν έχει άλλα συμπεράσματα να δώσει
  - Τα γεγονότα ανήκουν στο στρώμα 1.
- Π.χ. στο σύνολο κανόνων **A → Γ, B → Δ, Γ & not(Δ) → E, A, B**
  - Στρώμα 1: **A → Γ, B → Δ, A, B**
  - Στρώμα 2: **Γ & not(Δ) → E**
  - Αν υπήρχε κανόνας **Z & not(E) → H**, τότε θα ήταν στο στρώμα 3

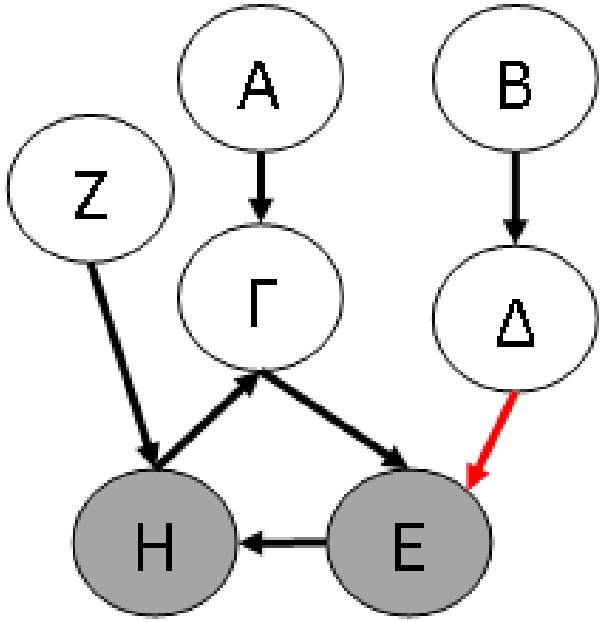
Stratification



- Όταν κάποιο «λογικό πρόγραμμα» δεν έχει αναδρομή (άμεση ή έμμεση), τότε είναι πάντα δυνατός ο χωρισμός των κανόνων σε «στρώματα».
- Όταν υπάρχει αναδρομή (κυκλική αλληλεξάρτηση κανόνων), τότε ενδέχεται να μην μπορεί να εκτελεστεί διαστρωματωμένα το λογικό πρόγραμμα
  - Όταν υπάρχει κυκλική αλληλεξάρτηση κανόνων και μέσα στο κύκλο εμπλέκεται κάποιος κανόνας με άρνηση στη συνθήκη.
  - Π.χ. **A → Γ, B → Δ, Γ & not(Δ) → E, Z & not(E) → H, H → Δ**

Stratification

- Δεν υπάρχει πρόβλημα όταν ο «κύκλος» δεν έχει άρνηση
  - Π.χ. **Α → Γ, Β → Δ,**
    Γ & not(Δ) → Ε,
    Ζ & Ε → Η, Η → Γ

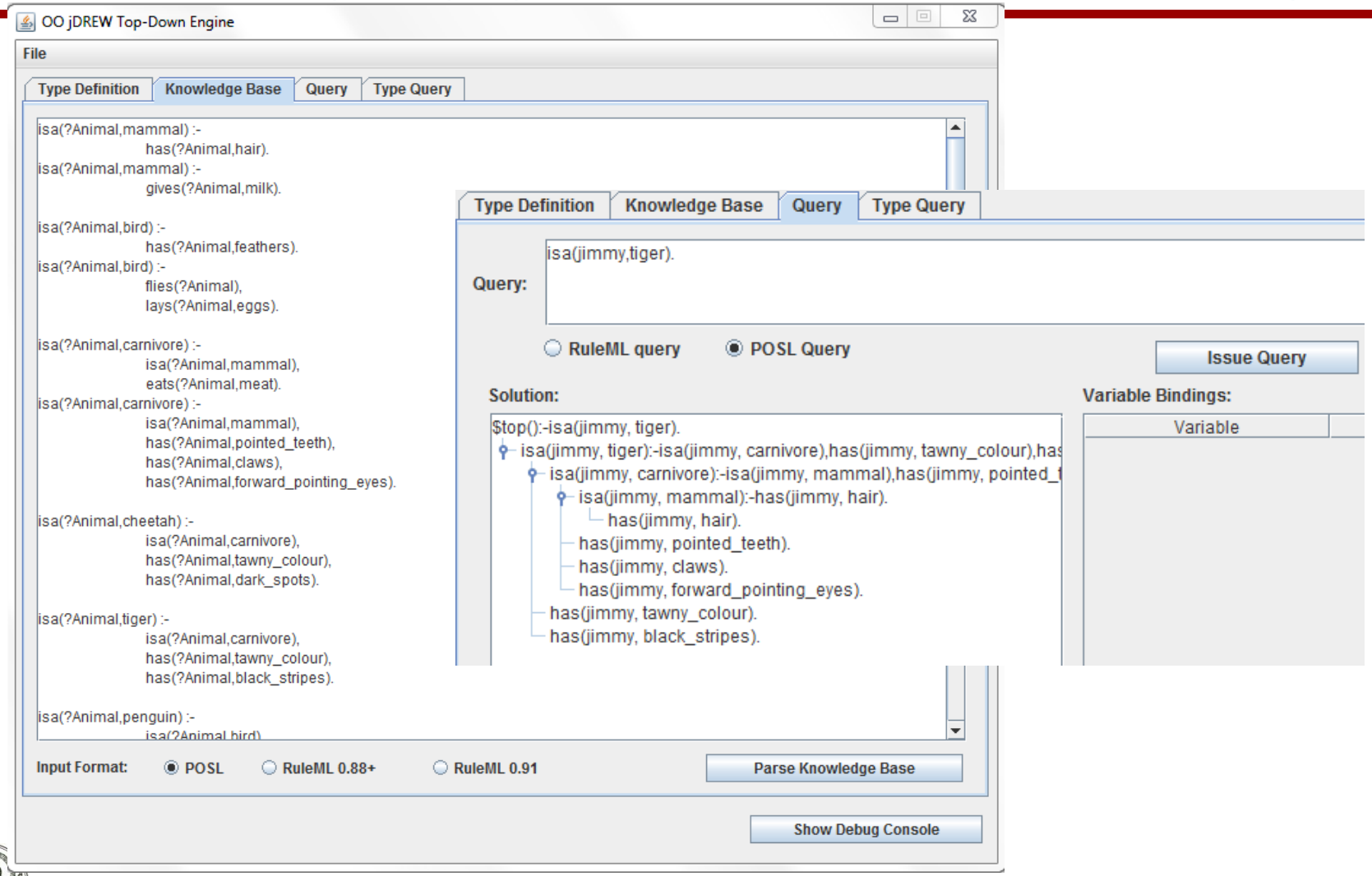


Σύστημα OO jDREW

- http://www.jdrew.org/oojdrew/
- Είναι ένα σύστημα εξαγωγής συμπερασμάτων για την γλώσσα RuleML
  - XML γλώσσα ανταλλαγής κανόνων στο web
- Εκτός από την XML σύνταξη της RuleML υποστηρίζει και μια σύνταξη που μοιάζει πολύ με την Prolog
  - Ονομάζεται POSL
- Υποστηρίζει και ορθή και ανάστροφη ακολουθία εκτέλεσης κανόνων.
- Στην ορθή ακολουθία ελέγχει και αν το σύνολο κανόνων είναι διαστρωματωμένο.

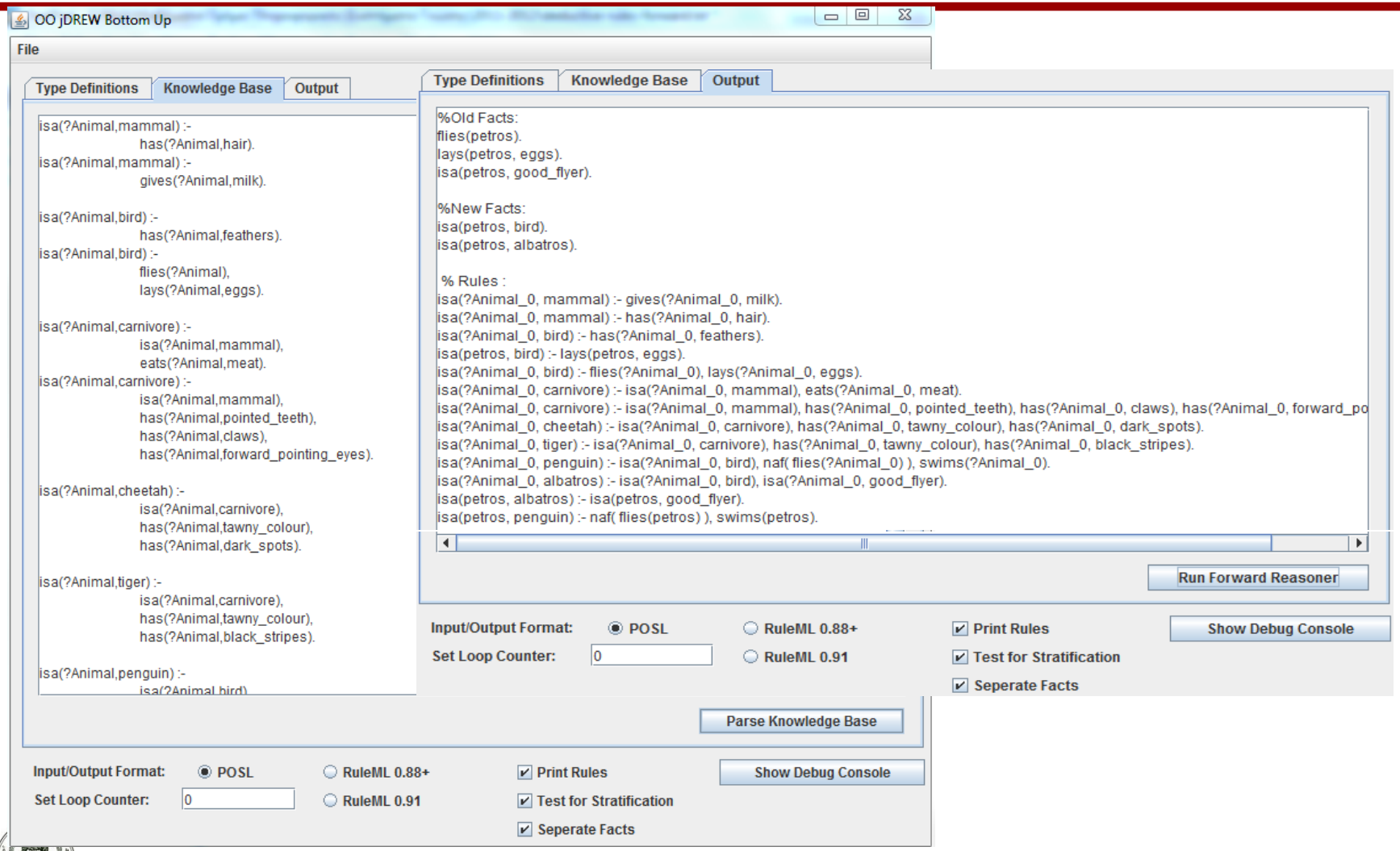
Σύστημα OO jDREW

Ανάστροφη Ακολουθία Εκτέλεσης



Σύστημα OO jDREW

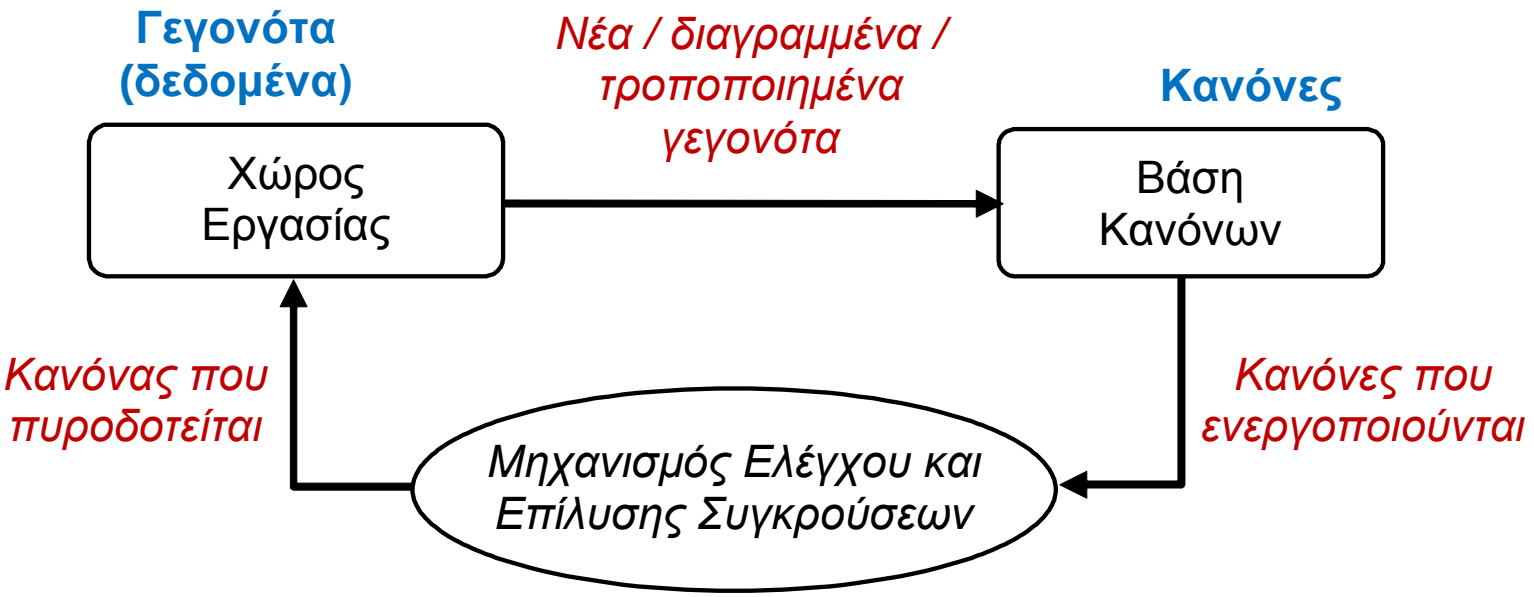
Ορθή Ακολουθία Εκτέλεσης



Συστήματα Παραγωγής (ΣΠ)

- Εκτελούν Κανόνες Παραγωγής (production rules)
  - **IF** Συνθήκη **THEN** Ενέργειες
- Αποτελείται από:
  - Τη *βάση κανόνων*.
  - Το *χώρο εργασίας* (*working memory*), που περιέχει στοιχεία της μνήμης εργασίας (*working memory elements*).
    - Αρχικά δεδομένα, ενδιάμεσα ή τελικά συμπεράσματα
  - 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*).

Δομή και Λειτουργία ΣΠ



Χώρος Εργασίας

- Ο χώρος εργασίας είναι δυναμικός
  - Τα περιεχόμενά του είναι διαφορετικά σε κάθε κύκλο λειτουργίας του συστήματος.
- Οι κανόνες παραγωγής είναι αυτοί που καθορίζουν τα περιεχόμενα του χώρου εργασίας, προσθέτοντας ή αφαιρώντας γεγονότα από αυτόν, σύμφωνα με τις ενέργειες του κάθε κανόνα.

Κύκλος Λειτουργίας ΣΠ

1. Έως ότου δε μπορεί να εκτελεστεί κανένας κανόνας επανέλαβε:
2. Βρες όλους του κανόνες που ενεργοποιούνται και σχημάτισε το σύνολο συγκρούσεων.
3. Σύμφωνα με το μηχανισμό επίλυσης συγκρούσεων, διάλεξε ένα κανόνα.
4. Πυροδότησε τον κανόνα που διάλεξες στο βήμα 2.

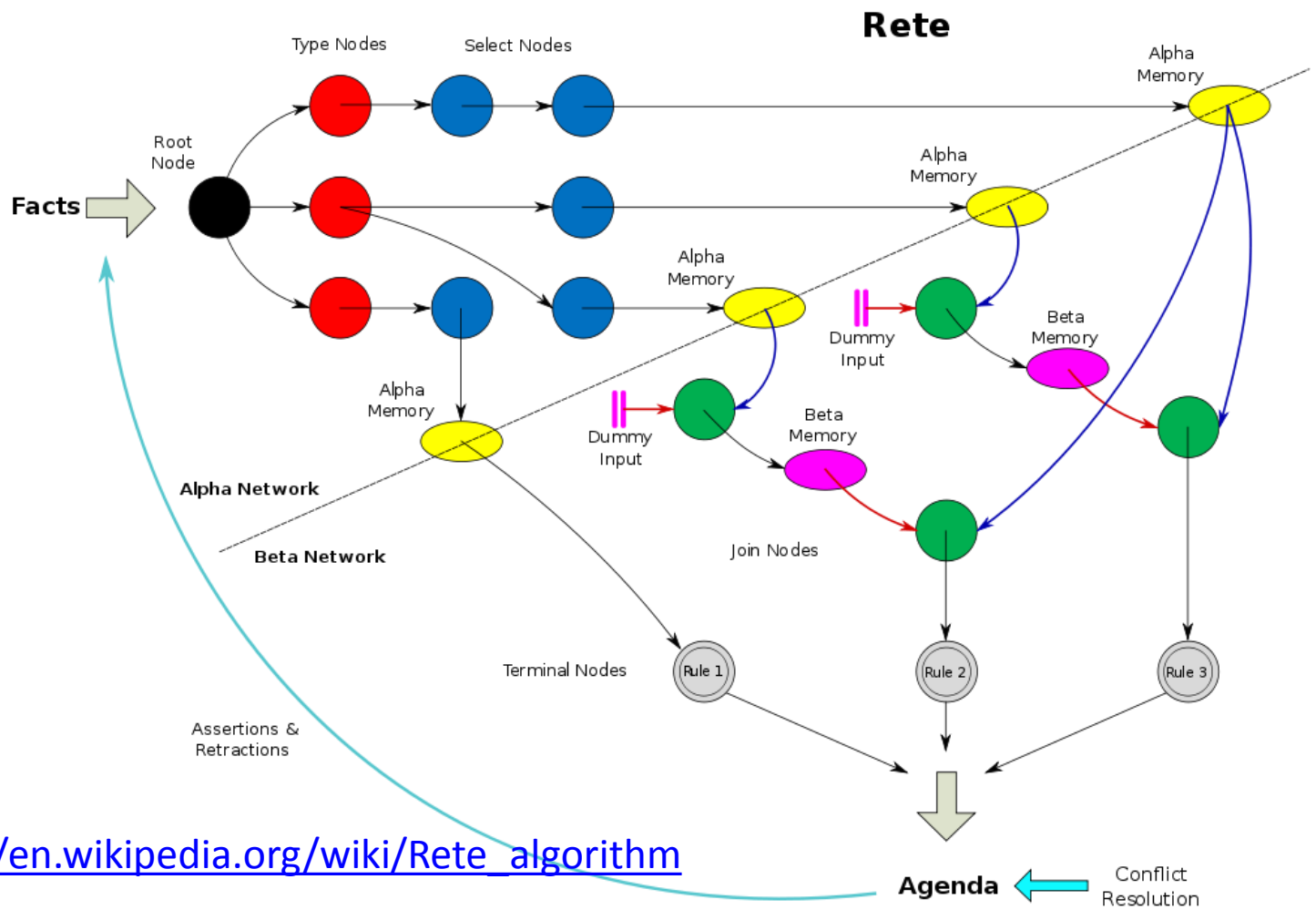
Ενεργοποίηση Κανόνων

- Πώς εντοπίζονται οι κανόνες που ενεργοποιούνται;
- **Αφελής λύση**: Ταίριασμα ΟΛΩΝ των γεγονότων με τις συνθήκες ΟΛΩΝ των κανόνων, σε ΚΑΘΕ κύκλο λειτουργίας
  - Μεγάλη πολυπλοκότητα – μεγάλος χρόνος εκτέλεσης (90%)
- **Έξυπνη λύση**: Αυξητικός (incremental) αλγόριθμος ταυτοποίησης κανόνων
  - Υπάρχει δεικτοδότηση μεταξύ των γεγονότων και των κανόνων που ενεργοποιούν
  - Υπάρχει μια δομή δεδομένων που κρατάει στοιχεία για την μερική ενεργοποίηση των συνθηκών των κανόνων

Αλγόριθμος / Δίκτυο RETE

- Αλγόριθμος αυξητικής ταυτοποίησης κανόνων
- Βασίζεται στην δομή δεδομένων «Δίκτυο RETE»
  - Ένας κόμβος **ρίζα**
  - **Κόμβοι «άλφα»** → ελέγχουν τις απλές συνθήκες του κανόνα
    - Υπάρχει ένας α-κόμβος για κάθε απλή συνθήκη σε κάθε κανόνα
    - **IF A & B & C THEN …** → α-κόμβοι: **A**, **B**, **C**
    - Όταν εντοπίσουν την «ύπαρξη» απλής συνθήκης αποθηκεύουν τα αντίστοιχα γεγονότα στην α-μνήμη
  - **Κόμβοι «βήτα»** → κάνουν JOIN μεταξύ των α-μνημών
    - Όταν υπάρχουν Ν απλές συνθήκες στον κανόνα, υπάρχουν Ν-1 β-κόμβοι
    - **IF A & B & C THEN …** → β-κόμβοι: **A** & **B**, **(A &B)** & **C**
    - Όταν εντοπίσουν την «ύπαρξη» κάποιας σύζευξης (JOIN) αποθηκεύουν τα αντίστοιχα γεγονότα στην β-μνήμη
  - Κάθε κανόνας παραγωγής αποτελεί **τερματικό** κόμβο

Αλγόριθμος / Δίκτυο RETE



Πηγή:
http://en.wikipedia.org/wiki/Rete_algorithm

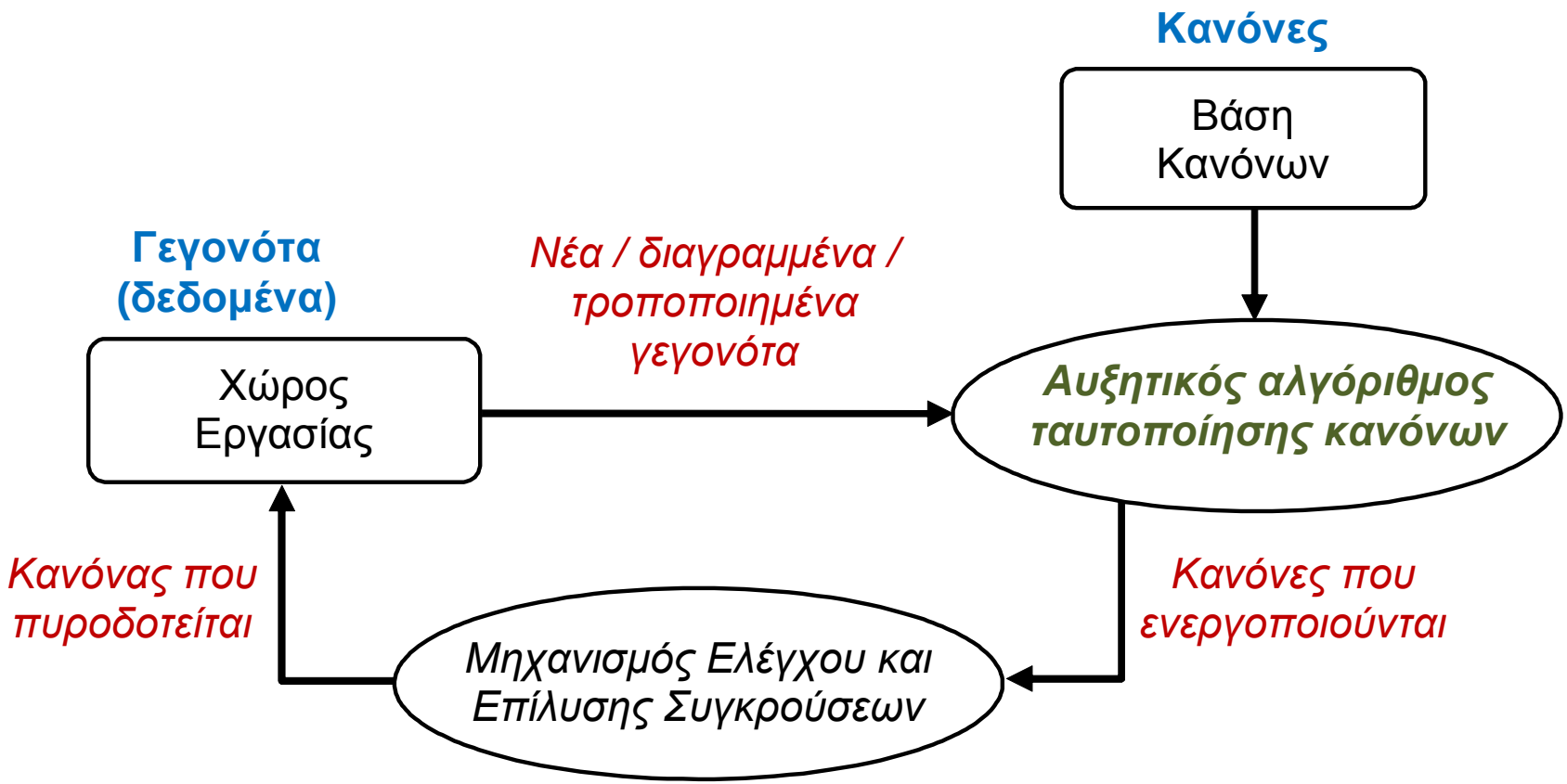
Γιατί ο αλγόριθμος RETE είναι γρηγορότερος;

- **IF a(1,X) & b(X,2) THEN …**
- Έστω ότι υπάρχουν 1000 γεγονότα **a(A1,A2)**
  - Το 10% είναι της μορφής **a(1,X)**
- Έστω ότι υπάρχουν 1000 γεγονότα **b(B1,B2)**
  - Το 10% είναι της μορφής **b(X,2)**
- Η αφελής λύση θα κάνει 2 loops (**1000x1000**) σε κάθε κύκλο
- Ο αλγόριθμος RETE σε έναν κύκλο:
  - Έστω ότι προστίθεται ένα γεγονός τύπου **a(1,X)**
  - Περνάει από τον α-κόμβο και μπαίνει στην α-μνήμη
  - Πάει στον β-κόμβο και γίνεται JOIN με την άλλη α-μνήμη που περιέχει 100 εγγραφές (πολυπλοκότητα **1x100**)
  - Όσα ταιριάζουν προωθούνται στον τερματικό κόμβο

Πλεονεκτήματα / Μειονέκτημα RETE

- Ο αριθμός των κανόνων ΔΕΝ επηρεάζει την αποδοτικότητα και την κατανάλωση μνήμης του RETE
- Οι α- και β-μνήμες καταναλώνουν χώρο στην κύρια μνήμη
  - Επιπλέον της μνήμης που καταλαμβάνουν τα γεγονότα στο χώρο εργασίας
  - Π.χ. στο προηγούμενο παράδειγμα στις α-μνήμες αποθηκεύονται 100+100=200 γεγονότα και στην β-μνήμη ένα ποσοστό από τα 200 αυτά γεγονότα
- Όταν υπάρχουν πάρα πολλά γεγονότα, και μάλιστα χωρίς υψηλή επιλεκτικότητα (selectivity) από τους α-κόμβους μπορεί να εξαντληθεί η κύρια μνήμη
  - Δεν ενδείκνυται για εφαρμογές πολύ μεγάλου όγκου δεδομένων (big data)
- Υπάρχουν αρκετές παραλλαγές του RETE (αλλά και νέες προσεγγίσεις) που ξεπερνούν τα παραπάνω προβλήματ

Δομή και Λειτουργία ΣΠ (2)



Συλλογιστική & Ακολουθία Εκτέλεσης ΣΠ

- Ακολουθείται η *ορθή ακολουθία εκτέλεσης κανόνων*.
- Δεν έχει νόημα ο όρος εξαγωγή συμπερασμάτων, γιατί οι κανόνες παραγωγής αναφέρονται σε ενέργειες που εκτελούνται
- 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.

Επίλυση Συγκρούσεων

- Ένας κανόνας *ενεργοποιείται* (*triggers*) όταν οι συνθήκες του κανόνα ικανοποιούνται
- Όταν ένας κανόνας *πυροδοτείται* (*fires*) τότε οι ενέργειές του *εφαρμόζονται* ή *εκτελούνται*.
- Το σύνολο των κανόνων που ενεργοποιούνται σχηματίζουν το *σύνολο σύγκρουσης* (*conflict set*).

Στρατηγικές Επίλυσης Συγκρούσεων

- *Τυχαία* (*random*).
  - Επιλέγεται ένας κανόνας στην τύχη.
- *Διάταξης* (*ordering*).
  - Επιλέγεται ο κανόνας που είναι πρώτος στη σειρά, ή
  - Έχει μεγαλύτερη προτεραιότητα βάσει κάποιου αριθμητικού μεγέθους.

Στρατηγικές Επίλυσης Συγκρούσεων

- *Αποφυγή επανάληψης* (*refractoriness*).
  - Δεν επιλέγεται ο ίδιος κανόνας με τα ίδια δεδομένα για δεύτερη συνεχόμενη φορά.
  - Αποφεύγονται άσκοπες ή ατέρμονες επαναλήψεις
  - Π.χ. Γεγονότα: **Α, Β**
  - Κανόνες: **1: Α → Γ, 2: Β → Δ**
  - Αν εκτελεστεί πρώτα ο 1, μετά θα εκτελεστεί ο 2
    - Ο 1 δε θα εκτελεστεί ξανά

Στρατηγικές Επίλυσης Συγκρούσεων

- *Επιλογή του πιο πρόσφατου* (*recency*)
  - Επιλέγεται ο κανόνας που ενεργοποιείται από τα πιο πρόσφατα δεδομένα που προστέθηκαν στο χώρο εργασίας.
  - Ακολουθείται μία χρονικά συνεπής πορεία σκέψης.
  - Π.χ. Γεγονότα: **Α, Β**
  - Κανόνες: **1: Α → Γ, 2: Β → Δ, 3: Γ → Ε**
  - Αν εκτελεστεί πρώτα ο 1, μετά θα εκτελεστεί ο 3
    - Γιατί το Γ είναι πιο πρόσφατο από το Β

Στρατηγικές Επίλυσης Συγκρούσεων

Επιλογή του πιο πρόσφατου (recency)

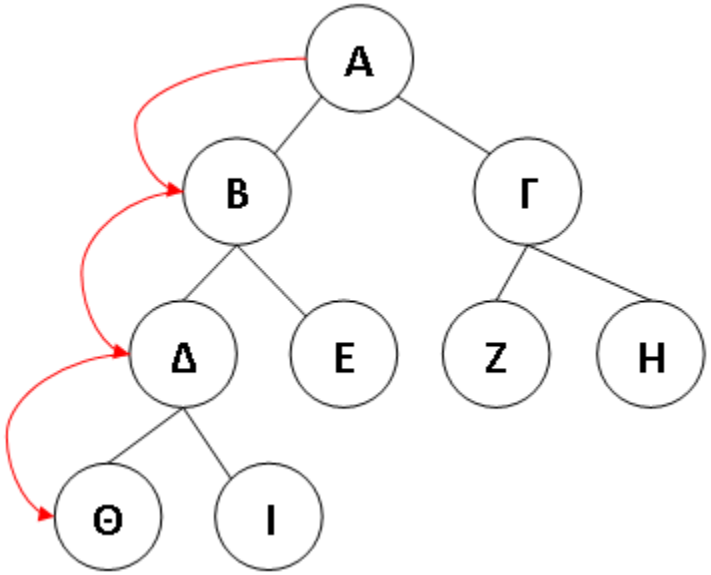
- Μοιάζει με την Αναζήτηση πρώτα-σε-βάθος (depth-first search)
- Παράδειγμα:
  - Π.χ. στη μνήμη εργασίας το γεγονός **A**
  - Στη βάση γνώσης οι κανόνες:

1: if A then B **2: if A then Γ**

3: if B then Δ **4: if B then E**

5: if Γ then Z **6: if Γ then H**

7: if Δ then Θ **8: if Δ then I**



  - Έστω ότι μεταξύ κανόνα 1 και 2, εκτελείται πρώτα ο 1.
  - Στη συνέχεια, θα εκτελεστεί ο 3 ή ο 4 και όχι ο 2, γιατί το B είναι πιο πρόσφατο από το A
  - Αν εκτελεστεί ο 3, στη συνέχεια προτεραιότητα έχουν οι 7 και 8 (όχι οι 2, 4)

Στρατηγικές Επίλυσης Συγκρούσεων

- *Επιλογή του πιο ειδικού ή συγκεκριμένου* (*specificity*).
  - Επιλέγεται ο κανόνας που είναι πιο ειδικός ή πιο συγκεκριμένος από τους άλλους, δηλαδή η συνθήκη του εκφράζεται με αναλυτικότερο τρόπο.
  - Εξετάζονται πρώτα τα πιο συγκεκριμένα θέματα τα οποία οδηγούν πιθανότατα σε λύση πιο γρήγορα.
  - Π.χ. Γεγονότα: **Α, Β, Γ**
  - Κανόνες: **1: Α & Β & Γ → Δ, 2: Α & Β → Ε**
  - Θα εκτελεστεί πρώτα ο 1, γιατί έχει πιο πολλές συνθήκες

Στρατηγικές Επίλυσης Συγκρούσεων

Επιλογή του πιο ειδικού (specificity)

- Συνήθως ο πιο ειδικός κανόνας εκφράζει μια εξαίρεση σε κάποιον πιο γενικό κανόνα.
  - Π.χ. **IF πουλί THEN πετάει**
  - **IF πουλί & πιγκουίνος THEN δεν_πετάει**

Στρατηγικές Επίλυσης Συγκρούσεων

Ανάλυση μέσων-σκοπών (means-ends analysis)

- Το συνολικό πρόβλημα που επιλύει το σύστημα κανόνων επιμερίζεται σε απλούστερες διεργασίες (tasks)
  - Κάθε διεργασία υλοποιείται από μία ομάδα κανόνων (cluster), υποσύνολο της συνολικής βάσης γνώσης.
- Όταν εκτελείται κάποια διεργασία, τότε οι κανόνες που ανήκουν σε άλλη ομάδα (διεργασία) δεν προτιμούνται, παρά μόνο αν δεν υπάρχουν άλλοι κανόνες της ίδιας ομάδας.
- Η αποδεικτική διαδικασία είναι επικεντρωμένη στους τρέχοντες **στόχους** της.

Παράδειγμα means-ends analysis

- Υπάρχουν στη μνήμη εργασίας τα γεγονότα **A, B, C, G1, G2**
- Στη βάση γνώσης υπάρχουν οι κανόνες

1: if G1 and A and B and C then D

2: if G2 and A and B then E

- Τα **G1**, **G2** υποδηλώνουν τη διεργασία στην οποία ανήκει ο κάθε κανόνας
  - Το **G2** είναι πιο πρόσφατο από το **G1**
- Θα εκτελεστεί πρώτα ο 2, γιατί ασχολείται με τον πιο τρέχοντα στόχο
  - Ο 1 μπορούσε να έχει προτεραιότητα λόγω άλλης στρατηγικής, (π.χ. specificity)

Μετα-έλεγχος

- Τα συστήματα παραγωγής εφαρμόζουν μία ή περισσότερες στρατηγικές επίλυσης συγκρούσεων
- Όταν υποστηρίζονται περισσότερες από μία στρατηγικές, πρέπει να υπάρχει προτεραιότητα μεταξύ αυτών
- Αυτό οδηγεί στην υλοποίηση ενός νέου επιπέδου ελέγχου, του *μετα-ελέγχου* (*meta-control*), που καθορίζει ποια στρατηγική θα εφαρμοστεί, πού και πότε
- Απλά συστήματα: σταθερή προτεραιότητα
  - Χαμηλότερη τιμή στην τυχαία επιλογή
- Σύνθετα συστήματα: η προτεραιότητα αλλάζει δυναμικά κατά τη διάρκεια της εκτέλεσης των κανόνων (at run-time).
  - *Μετα-κανόνες* (*meta-rules*): κανόνες που χρησιμοποιούνται για να καθορίσουν τη σειρά εκτέλεσης άλλων κανόνων

Στιγμιότυπα Κανόνων
Rule instantiations (1/2)

- Όταν οι κανόνες παραγωγής έχουν στη συνθήκη τους μεταβλητές, τότε οι συνθήκες των κανόνων «ταιριάζουν» (matching) με τα στοιχεία της μνήμης εργασίας και οι μεταβλητές παίρνουν συγκεκριμένες τιμές (σταθερές)
- Θυμίζει την ενοποίηση (unification) της Prolog
  - Διαφορά: είναι προς μία κατεύθυνση μόνο
  - Στην Prolog μεταβλητές μπορούν να υπάρχουν και στις 2 πλευρές των όρων που ενοποιούνται
  - Στους κανόνες παραγωγής η μια πλευρά μόνο (του κανόνα) μπορεί να περιέχει μεταβλητές, ενώ η άλλη πλευρά (μνήμη εργασίας) περιέχει μόνο σταθερές

Στιγμιότυπα Κανόνων
Rule instantiations (2/2)

- Παράδειγμα:
  - Στη μνήμη εργασίας υπάρχουν τα γεγονότα: **a(1,2), a(2,3)**
  - Έστω ο κανόνας **IF a(1,X) THEN …**
  - Η συνθήκη **a(1,X)** ταιριάζει με το γεγονός **a(1,2)** και η μεταβλητή **X** παίρνει την τιμή **2**
- Στο σύνολο συγκρούσεων δεν μπαίνουν οι κανόνες με μεταβλητές, αλλά με συγκεκριμένες τιμές
  - Ονομάζονται *στιγμιότυπα κανόνων* (*rule instantiations*)
  - Π.χ. **IF a(1,2) THEN …**

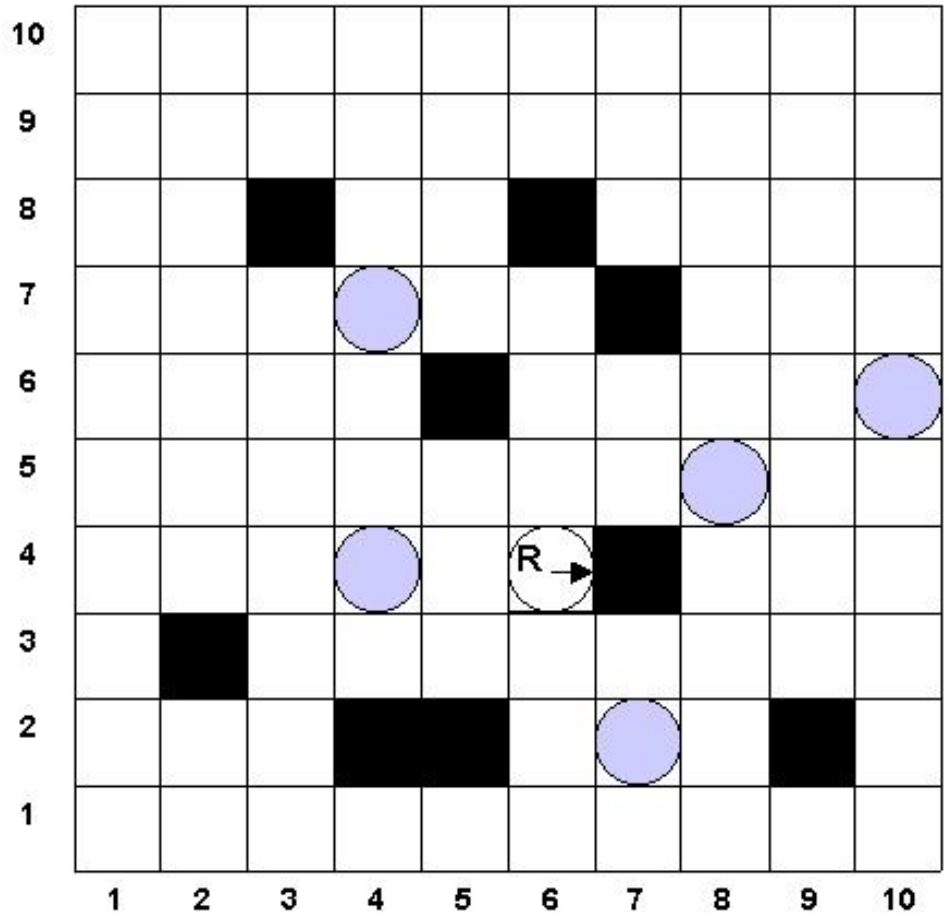
Στιγμιότυπα Κανόνων και Σύνολο Συγκρούσεων (1/2)

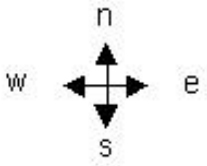
- Τι συμβαίνει όταν στην φάση του ταιριάσματος υπάρχουν πολλά γεγονότα που ταιριάζουν με την συνθήκη ενός κανόνα?
  - Π.χ. στη μνήμη εργασίας υπάρχουν τα γεγονότα: **a(1,2), a(1,4), a(2,3)**
  - Έστω ο κανόνας **IF a(1,X) THEN …**
  - Η συνθήκη **a(1,X)** ταιριάζει με τα γεγονότα **a(1,2)** και **a(1,4)**
- Στο σύνολο συγκρούσεων θα μπουν ΟΛΑ τα στιγμιότυπα κανόνων
  - Π.χ. **IF a(1,2) THEN …** και **IF a(1,4) THEN …**
- Η ύπαρξη μεταβλητών στους κανόνες υπονοεί τον καθολικό ποσοδείκτη ∀
  - **∀X, IF a(1,X) THEN …**

Στιγμιότυπα Κανόνων και Σύνολο Συγκρούσεων (2/2)

- Δηλαδή, οι μεταβλητές υπονοούν την ύπαρξη ενός βρόχου επανάληψης, ανάλογου με το βρόχο for στις διαδικαστικές γλώσσες προγραμματισμού
  - Π.χ. **IF a(X) THEN print(X)**
  - Ερμηνεία: Τύπωσε όλα τα X
- Οι στρατηγικές επίλυσης συγκρούσεων ισχύουν και για τα στιγμιότυπα κανόνων
  - Π.χ. ισχύει η στρατηγική επίλυσης συγκρούσεων «*επιλογή του πιο πρόσφατου*»
  - το γεγονός **a(1,4)** είναι πιο πρόσφατο από το γεγονός **a(1,2)**
  - το στιγμιότυπο **IF a(1,4) THEN …** έχει προτεραιότητα στην εκτέλεση από το **IF a(1,2) THEN …**

Παράδειγμα Κίνησης Ρομπότ






```
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

- 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

Παρατηρήσεις
Ενέργειες Των Κανόνων

- Οι ενέργειες των κανόνων εμπεριέχουν 4 λειτουργίες:
  - addwm: βάλε κάτι στη μνήμη εργασίας
  - delwm: σβήσε κάτι από τη μνήμη εργασίας
  - output: εκτύπωσε ένα μήνυμα στην οθόνη, και
  - 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: ...

5: move_south: ...

Κανόνες Κίνησης Ρομπότ

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: ...

9: avoid_obstacle_east: ...

Στρατηγικές Επίλυσης Κίνησης Ρομπότ

- Οι στρατηγικές επίλυσης συγκρούσεων είναι με τη σειρά:
  - αποφυγή επανάληψης (ΑΕ),
  - επιλογή του πιο ειδικού (ΕΕ), και
  - τυχαία επιλογή (ΤΕ).

Παρακολούθηση Εκτέλεσης

Κύκ-λος	Μνήμη Εργασίας	Σύνολο Συγκρού-σεων	Στρατη-γική	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)

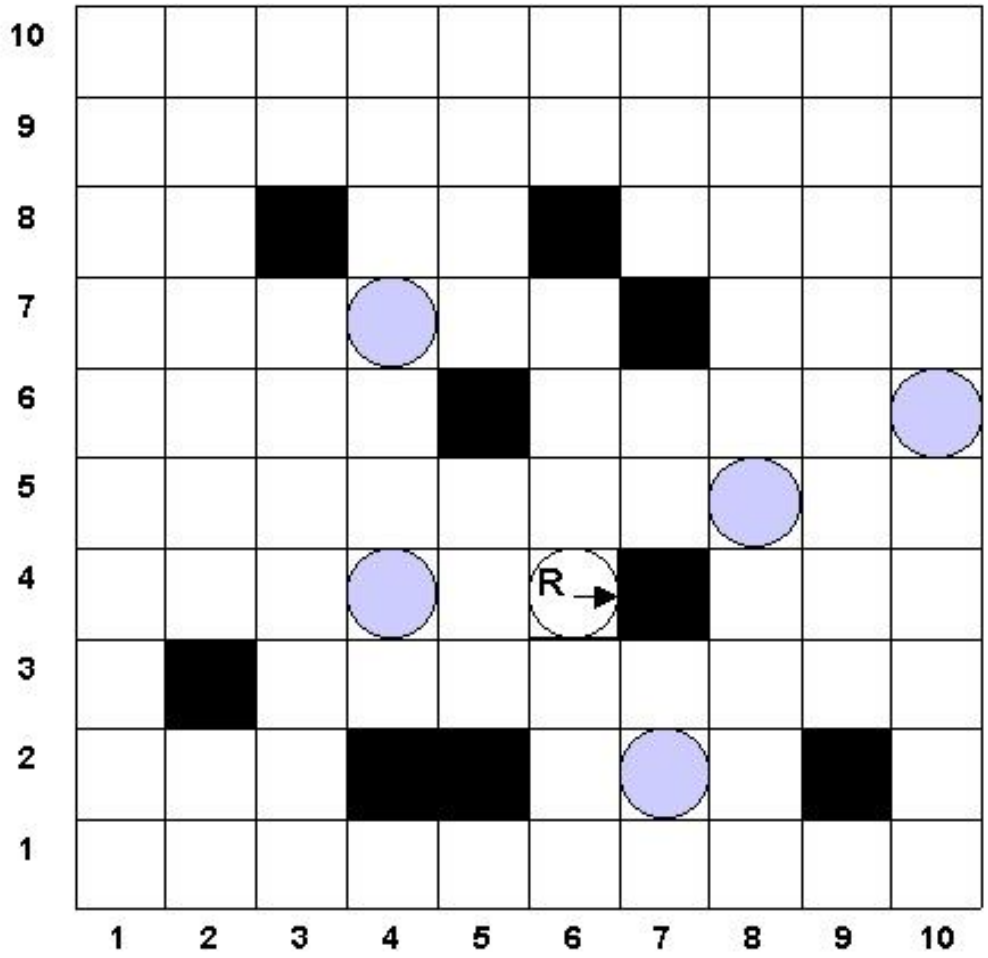
Επιλέγεται ένας (9), με choice(n) – τυχαία

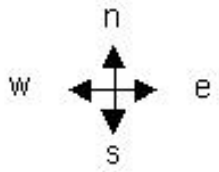
Ο (3) αποκλείεται γιατί είναι πιο γενικός

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)

Παράδειγμα Κίνησης Ρομπότ





Παρακολούθηση Εκτέλεσης

Κύκ-λος	Μνήμη Εργασίας	Σύνολο Συγκρού-σεων	Στρατη-γική	Κανόνας που πυροδοτεί

2	`robot_at(6,4)`
	`direction(n)`
	`. . .`
		`{4}`	-	`4: move_north`

3	`robot_at(6,5)`
	`direction(n)`
	`. . .`
		`{4}`	-	`4: move_north`

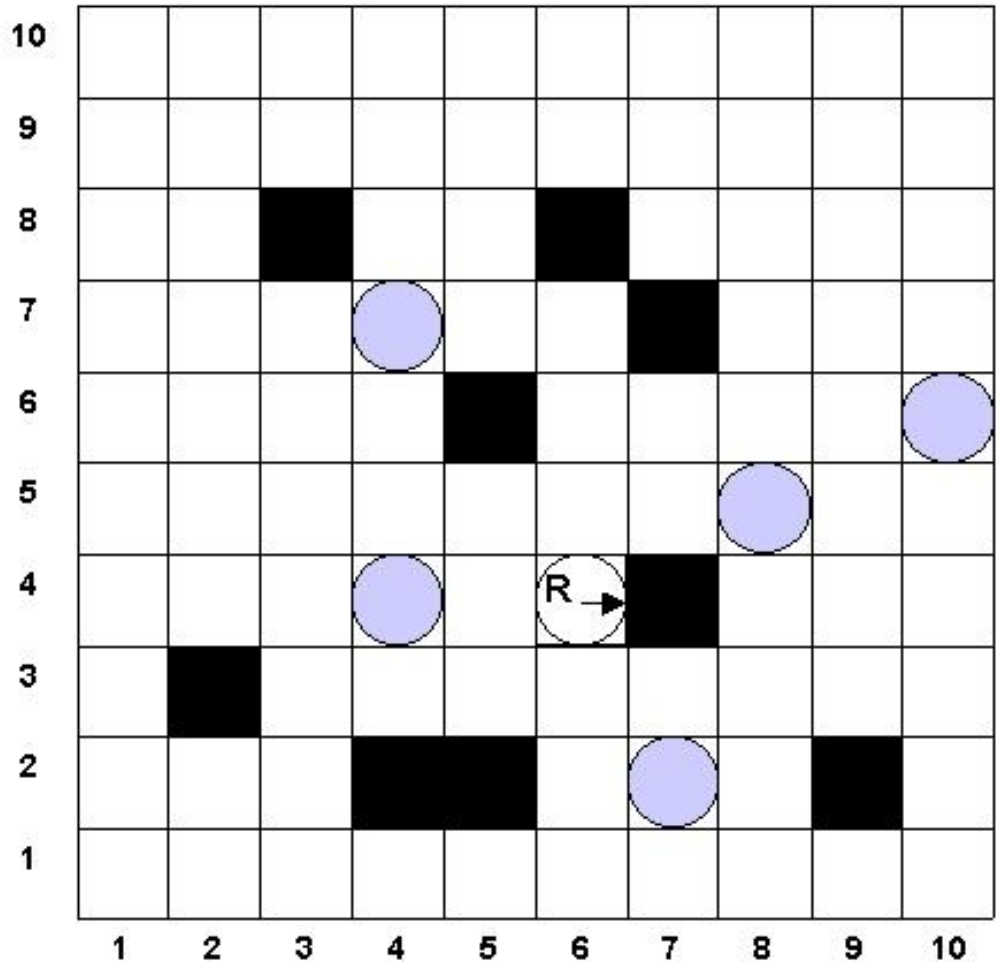
4	`robot_at(6,6)`
	`direction(n)`
	`. . .`
		`{4}`	-	`4: move_north`

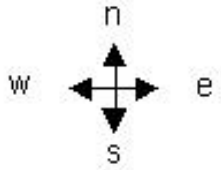
5	`robot_at(6,7)`
	`direction(n)`
	`. . .`
	`obstacle_at(6,8)`
	`. . .`
		`{4,`
		`8 (ND=w), 8 (ND=n), 8 (ND=s), 8 (ND=e)}`
			EE
			TE
				`8:avoid_obstacle_north (ND=n)`

Παρακολούθηση Εκτέλεσης

Κύκ-λος	Μνήμη Εργασίας	Σύνολο Συγκρού-σεων	Στρατη-γική	Κανόνας που πυροδοτεί
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_obs tacle_nor th (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_obs tacle_eas t (ND=w)
8	robot_at(6,7) direction(w) . . .	{2}	-	2: move_west

Παράδειγμα Κίνησης Ρομπότ





Παρακολούθηση Εκτέλεσης

Κύκ-λος	Μνήμη Εργασίας	Σύνολο Συγκρού-σεων	Στρατη-γική	Κανόνας που πυροδοτεί

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`

Σχέση Κανόνων Παραγωγής και Συνεπαγωγικών Κανόνων

- Οι κανόνες παραγωγής μοιάζουν πολύ με τους συνεπαγωγικούς κανόνες που εκτελούνται με ορθή ακολουθία εκτέλεσης
  - Μπορούν να χρησιμοποιηθούν για να προσομοιώσουν την διαδικασία εξαγωγής συμπερασμάτων ως προσθήκη των συμπερασμάτων στην μνήμη εργασίας (entailment)
- Παράδειγμα συνεπαγωγικών κανόνων:

IF A THEN B. **IF B THEN C.** **A.**

  - Αν εκτελεστούν με ορθή ακολουθία εκτέλεσης προκύπτουν τα συμπεράσματα **B** και **C**
- Προσομοίωση με κανόνες παραγωγής:

IF A THEN addwm(B). **IF B THEN addwm(C).** **A.**

  - Αν εκτελεστούν οι κανόνες παραγωγής, στη μνήμη εργασίας θα προστεθούν τα γεγονότα **B** και **C**

Συνδυασμός Πλαισίων και Κανόνων

- Η συνθήκη του κανόνα είναι μια σειρά από στοιχειώδεις κλήσεις σε τιμές πλαισίων που συνδέονται με λογικούς τελεστές.
- Οι ενέργειες του κανόνα είναι αναθέσεις τιμών σε τιμές πλαισίων.

```
IF      X is animal AND
        Y is human AND
        Y owns X AND
        X likes Z
THEN Y buys Z
```

Παράδειγμα Πλαισίων

- Στιγμιότυπο Fred:
  - is_a: Human
  - buys: {string}
  - owns: Nellie
- Στιγμιότυπο Nellie
  - is_a: Elephant
  - likes: apples
  - Size: small
- Ο προηγούμενος κανόνας ενεργοποιείται από τα 2 παραπάνω πλαίσια και θέτει apples ως τιμή του buys στον Fred

Παράδειγμα Πλαισίων

- Στιγμιότυπο John:
  - is_a: Human
  - buys: {string}
  - owns: Fido
- Στιγμιότυπο Fido:
  - is_a: Dog
  - likes:
- Ο προηγούμενος κανόνας δεν ενεργοποιείται από τα 2 παραπάνω πλαίσια γιατί ο Fido δεν έχει τιμή στη σχισμή likes

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ

Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας

Θεσσαλονίκη, 17/3/2014

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης