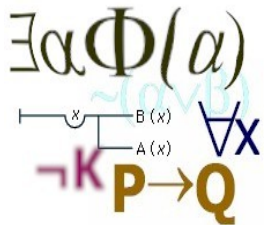


Εκτέλεση Prolog Προγραμμάτων

Ηλίας Σακελλαρίου



Δομή

- Πως απαντά η Prolog στις ερωτήσεις;
 - Διαδικασία
 - Ενοποίηση και απόδοση τιμών σε μεταβλητές
 - Σύνολο απάντησης
 - Οπισθοδρόμηση (backtracking)
- Ισχυρά θεωρητικά θεμέλια

Εκτέλεση Προγραμμάτων

- Η εκτέλεση ξεκινά με μια ερώτηση που υποβάλλει ο χρήστης
 - Φτάνουμε σε λύση δεν υπάρχει τίποτε να αποδειχθεί = κενή πρόταση.
 - Η απάντηση (**yes/no + αναθέσεις μεταβλητών**) είναι το αποτέλεσμα του προγράμματος.
- Η ερώτηση μπορεί να είναι:
 - **απλή**, περιέχει ένα στόχο (**goal**), ή
 - **σύνθετη**, αποτελείται από σύζευξη πολλών στόχων (υποστόχων).

Βήματα Εκτέλεσης

- **Υποστόχοι** προς απόδειξη επιλέγονται από αριστερά προς τα δεξιά.
- Για ένα επιλεγμένο **υποστόχο**, ο μηχανισμός ελέγχου αναλαμβάνει να βρει μια **πρόταση του προγράμματος που ενοποιείται**.
 - Ο **στόχος αντικαθίσταται** κατάλληλα
 - Εφόσον υποστόχοι περιέχουν κοινές μεταβλητές, θα πρέπει αυτές να πάρουν την ίδια τιμή.
- Επανάληψη διαδικασίας, μέχρι
 - Εξάντληση (**ικανοποίηση**) όλων των στόχων, ή
 - Αποτυχία

Ενοποιήσεις και αντικαταστάσεις με προτάσεις του προγράμματος

- Όταν ο τρέχων στόχος ενοποιείται με ένα **γεγονός** του προγράμματος τότε ικανοποιείται και απομακρύνεται από την ερώτηση.
- Αν ο τρέχων στόχος ενοποιείται με κάποιον **κανόνα**, τότε αντικαθίσταται από το **σώμα του κανόνα**.
 - Νέοι στόχοι
- Ύπαρξη **πολλών γεγονότων ή/και κανόνων** που μπορούν να ενοποιηθούν.
 - Επιλογή του **πρώτου** στο πρόγραμμα
 - Δημιουργία σημείου επιλογής (**choice point**)

Οπισθοδρόμηση (backtracking)

- Σε περίπτωση αποτυχίας ο μηχανισμός οπισθοδρομεί στο πιο πρόσφατο σημείο επιλογής (**choice point**).
 - Χρονολογική Οπισθοδρόμηση (**Chronological Backtracking**)
- Εξερεύνηση όλων των εναλλακτικών λύσεων με συστηματικό τρόπο.
- Κατά την οπισθοδρόμηση οι αναθέσεις τιμών που έγιναν μετά από το σημείο επιλογής αναιρούνται.
 - Νέες αναθέσεις τιμών

Παράδειγμα

?- follows(ilias,X), follows(petros,Y) .

[follows(ilias,X), follows(petros,Y)]

[follows(petros,Y)] {X/petros}

[] {X/petros, Y/ilias}

Κώδικας Prolog

follows(ilias, petros).

follows(petros, ilias).

follows(petros, demos). ●

follows(petros, sofia). ●


```
?- follows(ilias,X) , follows(petros,Y) .
```

```
X = petros
```

```
Y = ilias ;
```

Κώδικας Prolog

follows(ilias, petros).

follows(petros,ilias).

follows(petros,demos).

follows(petros,sofia).

?- follows(ilias,X), follows(petros,Y) .

[follows(ilias,X), follows(petros,Y)]

[follows(petros,Y)] {X/petros}

[] {X/petros, Y/demos}

Κώδικας Prolog

follows(ilias, petros).

● *follows(petros, ilias).*

➤ *follows(petros, demos).* ●

follows(petros, sofia). ●

?- follows(ilias,X) , follows(petros,Y) .

X = petros
Y = ilias ;

X = petros
Y = demos ;

Κώδικας Prolog

follows(ilias, petros).
follows(petros,ilias).
follows(petros,demos).
follows(petros,sofia).

?- follows(ilias,X), follows(petros,Y) .

[follows(ilias,X), follows(petros,Y)]

[follows(petros,Y)] {X/petros}

[] {X/petros, Y/sofia}

Κώδικας Prolog

follows(ilias, petros).

follows(petros, ilias).

● *follows(petros, demos).*

→ *follows(petros, sofia).* ●

```
?- follows(ilias,X) , follows(petros,Y) .
```

```
X = petros
```

```
Y = ilias
```

```
X = petros
```

```
Y = demos
```

```
X = petros
```

```
Y = sofia
```

```
true.
```

Κώδικας Prolog

follows(ilias, petros).

follows(petros,ilias).

follows(petros,demos).

follows(petros,sofia).

Παράδειγμα με Κανόνες

?- friends(X,Y) .

[friends(X,Y)]

[follows(X,Y), follows(Y,X)]

[follows(petros,ilias)] {X/ilias,Y/petros}

[] {X/ilias, Y/petros}

Κώδικας Prolog

friends(X,Y):-

*follows(X,Y),
follows(Y,X).*

follows(ilias, petros).

follows(petros,ilias).

follows(petros,demos).

follows(petros,sofia).

```
?- friends(X,Y) .
```

```
X = ilias
```

```
Y = petros
```

```
true.
```

Κώδικας Prolog

friends(X, Y):-

follows(X, Y),

follows(Y, X).

follows(ilias, petros).

follows(petros, ilias).

follows(petros, demos).

follows(petros, sofia).

Παράδειγμα (Εναλλακτικό)

Παράδειγμα Εκτέλεσης

- Ερώτηση

?-slow_down

- Αρχική στοίβα στόχων
(**goal stack**).

slow_down.

- Κάθε φορά
αντικαθιστούμε την
κορυφή της στοίβας.

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Παράδειγμα Εκτέλεσης

slow_down

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

night

danger

slow_down *

Παράδειγμα Εκτέλεσης

night
danger
slow_down*

Πρόγραμμα

fog.
night.
rain(heavy).
road(good).

rain:-rain(heavy).
rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.
slow_down:- night, rain(heavy).
slow_down:-road(bad).

danger
slow_down*

Παράδειγμα Εκτέλεσης

danger
slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

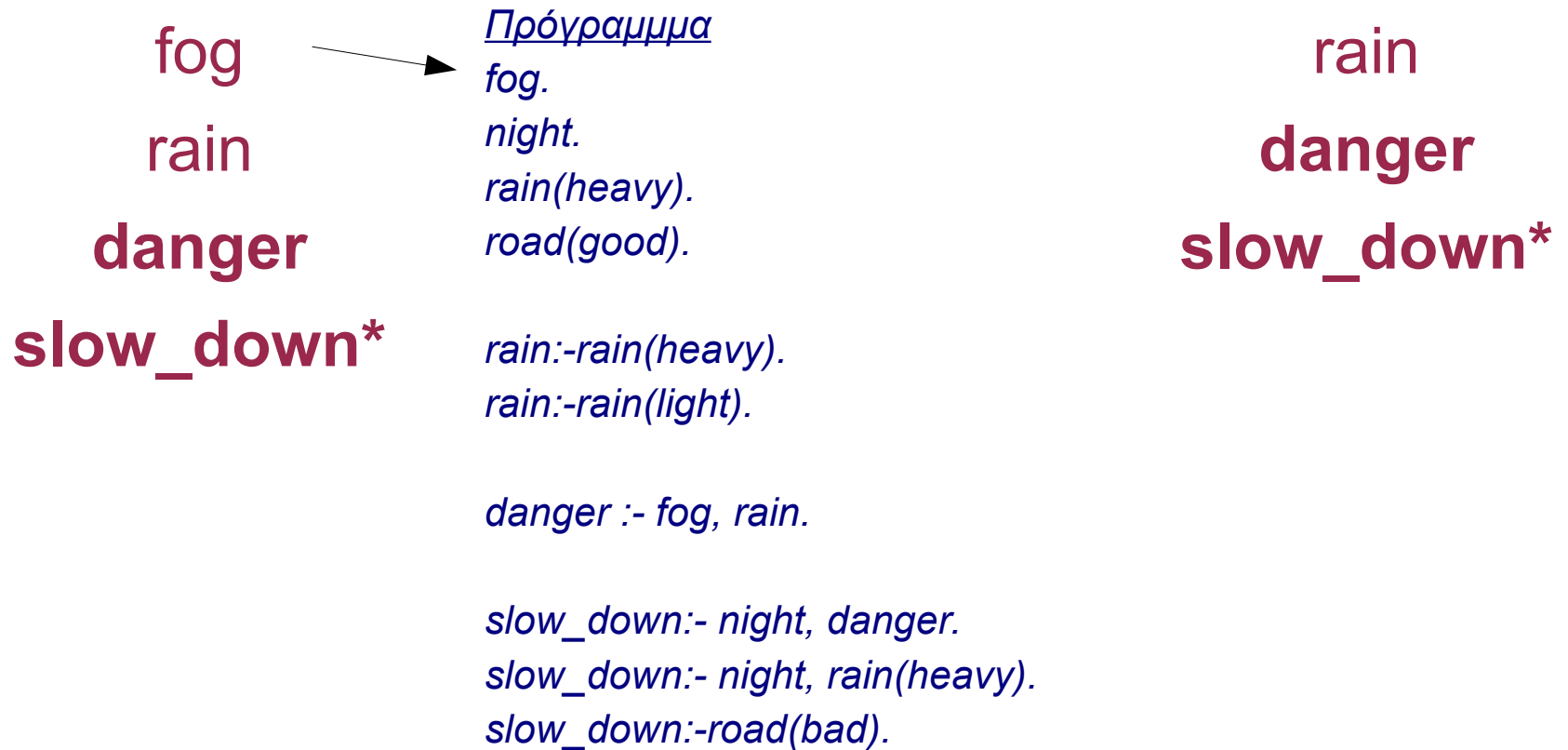
slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

fog
rain
danger
slow_down*

Παράδειγμα Εκτέλεσης



Παράδειγμα Εκτέλεσης

rain
danger
slow_down*

Πρόγραμμα

fog.
night.
rain(heavy).
road(good).

rain:-rain(heavy).
rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.
slow_down:- night, rain(heavy).
slow_down:-road(bad).

rain(heavy)
rain*
danger
slow_down*

Παράδειγμα Εκτέλεσης

rain(heavy)
rain*
danger
slow_down*

Πρόγραμμα

fog.
night.
rain(heavy).
road(good).

rain:-rain(heavy).
rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.
slow_down:- night, rain(heavy).
slow_down:-road(bad).

rain*
danger
slow_down*

Παράδειγμα Εκτέλεσης

rain*

danger

slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

-yes

(δεν μένει να
αποδείξω τίποτε
άλλο-επιτυχία).

Παράδειγμα Εκτέλεσης – Επόμενη Λύση

rain*
danger
slow_down*

Πρόγραμμα

fog.
night.
rain(heavy).
road(good).

rain:-rain(heavy).
rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.
slow_down:- night, rain(heavy).
slow_down:-road(bad).

rain(light)
rain
danger
slow_down*

redo: στο τελευταίο
σημείο επιλογής
(choice point)

Παράδειγμα Εκτέλεσης – Επόμενη Λύση

fails

rain(light)

rain*

danger

slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

~~rain(light)~~

~~rain~~

~~danger~~

~~slow_down*~~

Παράδειγμα Εκτέλεσης – Επόμενη Λύση

slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

night

rain(heavy)

slow_down*

Παράδειγμα Εκτέλεσης – Επόμενη Λύση

night
rain(heavy)
slow_down*

Πρόγραμμα

fog.
night.
rain(heavy).
road(good).

rain:-rain(heavy).
rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.
slow_down:- night, rain(heavy).
slow_down:-road(bad).

rain(heavy)
slow_down*

Παράδειγμα Εκτέλεσης – Επόμενη Λύση

rain(heavy)
slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

slow_down*

Παράδειγμα Εκτέλεσης – Επόμενη Λύση

slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

-yes

(Δεν υπάρχει
τίποτε άλλο
προς απόδειξη
- επιτυχία)

Παράδειγμα Εκτέλεσης – Τρίτη Λύση

slow_down*

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

road(bad)

slow_down

Παράδειγμα Εκτέλεσης – Τρίτη Λύση

fails

road(bad)
slow_down*

Πρόγραμμα
fog.
night.
rain(heavy).
road(good).

rain:-rain(heavy).
rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.
slow_down:- night, rain(heavy).
slow_down:-road(bad).

~~road(bad)~~
slow_down

-no

Εναλλακτική Αναπαράσταση

?-slow_down*

?-night,danger

?-danger

?-fog,rain.

?-rain*

?-rain(heavy)

?- <> (κενή πρόταση)

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Εναλλακτική Αναπαράσταση- (Β απάντηση)

?-slow_down*

?-night,danger

?-danger

?-fog,rain

?-rain*

?-rain(light)

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Εναλλακτική Αναπαράσταση- (Β απάντηση)

?-slow_down*

?-night,rain(heavy)

?-rain(heavy)

?-<> (κενή πρόταση επιτυχία)

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Εναλλακτική Αναπαράσταση- (Γαπάντηση)

?-slow_down*

?-road(bad)

(αποτυχία)

Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

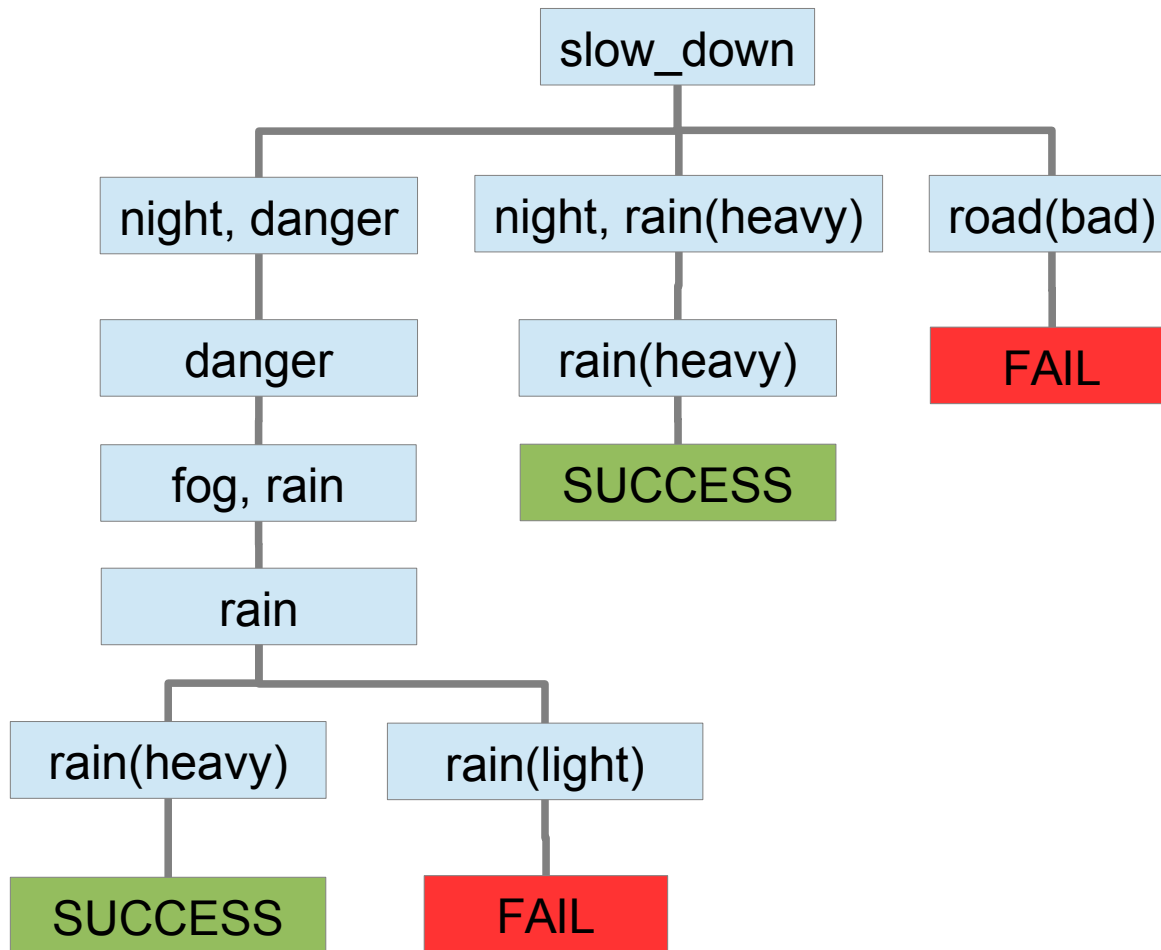
danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Δένδρο Εκτέλεσης



Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

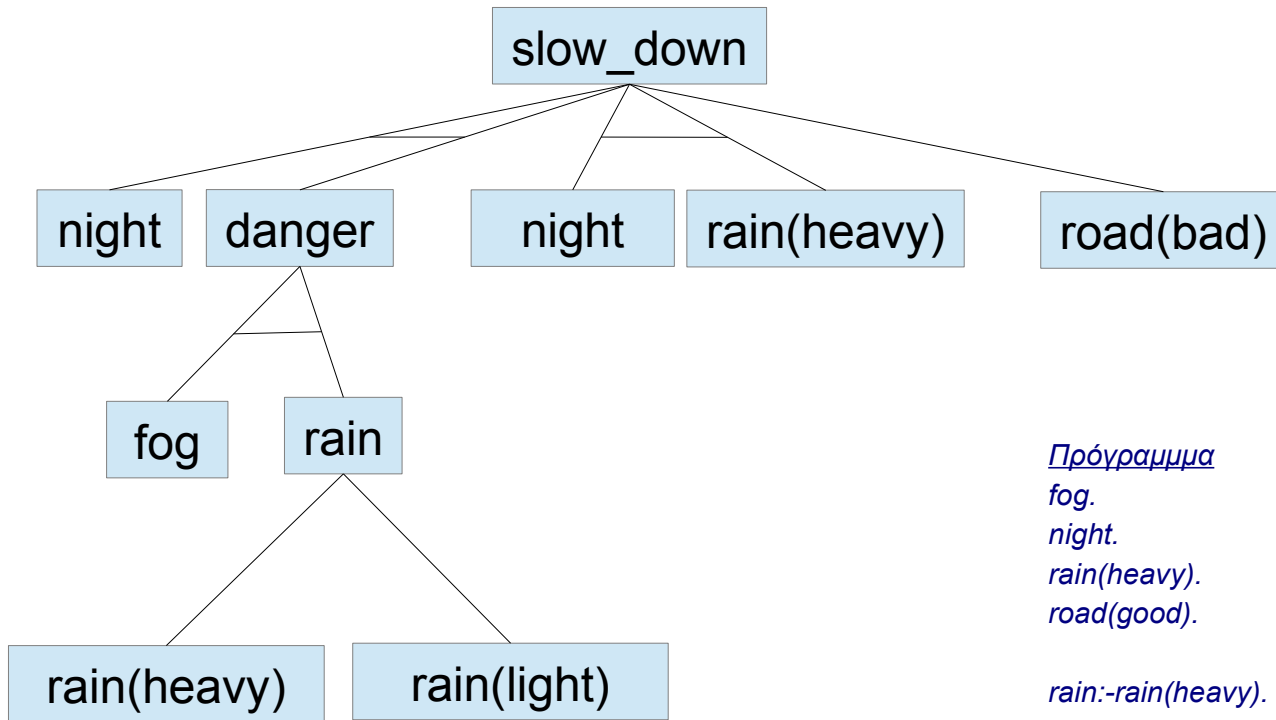
danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Δένδρο AND/OR



Πρόγραμμα

fog.

night.

rain(heavy).

road(good).

rain:-rain(heavy).

rain:-rain(light).

danger :- fog, rain.

slow_down:- night, danger.

slow_down:- night, rain(heavy).

slow_down:-road(bad).

Παράδειγμα με μεταβλητές

Πρόγραμμα

task(a,analysis).

task(b,development).

task(c,analysis).

task(d,testing).

cando(teamA,analysis).

cando(teamA,development).

cando(teamB,analysis).

cando(teamB,development).

cando(teamC,analysis).

assign(task_to(Task,Team)):-

task(Task,Req),

cando(Team,Req).

Εύρεση Λύσης – Α Λύση

?-assign(task_to(W,teamC).

{Task/W, Team/teamC}

?-task(W,Req), cando(teamC,Req)

*{W/a, Req/analysis} **

?-cando(teamC,analysis)

?- <>.

Answer W = a

yes

task(a,analysis).

task(b,developement).

task(c,analysis).

task(d,testing).

cando(teamA,analysis).

cando(teamA,developement).

cando(teamB,analysis).

cando(teamB,developement).

cando(teamC,analysis).

assign(task_to(Task,Team)):-

task(Task,Req),

cando(Team,Req).

Εύρεση Λύσης – Β Λύση

?-assign(task_to(W,teamC).

{Task/W, Team/teamC}

?-task(W,Req), cando(teamC,Req)

*{W/b, Req/development) **

?-cando(teamC,development)

BACKTRACKING!

task(a,analysis).

task(b,developement).

task(c,analysis).

task(d,testing).

cando(teamA,analysis).

cando(teamA,developement).

cando(teamB,analysis).

cando(teamB,developement).

cando(teamC,analysis).

assign(task_to(Task,Team)):-

task(Task,Req),

cando(Team,Req).

Εύρεση Λύσης – Β Λύση

?-assign(task_to(W,teamC).

{Task/W, Team/teamC}

?-task(W,Req), cando(teamC,Req)

*{W/c, Req/analysis} **

?-cando(teamC,analysis)

?- <>.

Answer W = c

yes

task(a,analysis).

task(b,developement).

task(c,analysis).

task(d,testing).

cando(teamA,analysis).

cando(teamA,developement).

cando(teamB,analysis).

cando(teamB,developement).

cando(teamC,analysis).

assign(task_to(Task,Team)):-

task(Task,Req),

cando(Team,Req).

Εύρεση Λύσης – Τέλος Διαδικασίας

?-assign(task_to(W,teamC).

{Task/W, Team/teamC}

?-task(W,Req), cando(teamC,Req)

{W/d, Req/testing}

?-cando(teamC,testing)

BACKTRACKING

task(a,analysis).

task(b,developement).

task(c,analysis).

task(d,testing).

cando(teamA,analysis).

cando(teamA,developement).

cando(teamB,analysis).

cando(teamB,developement).

cando(teamC,analysis).

assign(task_to(Task,Team)):-

task(Task,Req),

cando(Team,Req).

Εύρεση Λύσης – Τέλος Διαδικασίας

?-assign(task_to(W,teamC).

{Task/W, Team/teamC}

?-task(W,Req), cando(teamC,Req)

FAILURE!

Answer

no

task(a,analysis).

task(b,developement).

task(c,analysis).

task(d,testing).

cando(teamA,analysis).

cando(teamA,developement).

cando(teamB,analysis).

cando(teamB,developement).

cando(teamC,analysis).

assign(task_to(Task,Team)):-

task(Task,Req),

cando(Team,Req).

Θεωρητικά Θεμέλια

- Είναι η διαδικασία σωστή και πλήρης;

Διαδικασίες “εις άτοπο απαγωγή”

- “Εις άτοπο απαγωγή” (**Proof by Contradiction**)
 - Προσθέτουμε την άρνηση της προς απόδειξη πρότασης στα αξιώματα και αποδεικνύουμε ότι οδηγούμαστε σε αντίφαση.
- Πλεονέκτημα
 - Ευκολότερη εύρεση απόδειξης καθώς η αναζήτηση είναι καθοδηγούμενη.

Αρχή της Ανάλυσης

- Εισήχθηκε από τον Robinson (1965)
- Κανόνας συμπερασμού
 - Εξαγωγή νέων προτάσεων από ήδη υπάρχουσες.
 - Σε ένα σύστημα “εις άτοπο απαγωγής” η αναλύση είναι **πλήρης**.
- Αρχή της ανάλυσης:
 - $\{ (P \vee Q), (\neg Q \vee Z) \}$ μπορούμε να εξάγουμε το $P \vee Z$
 - Παράδειγμα

$\text{loves}(\text{john}, \text{coffee}) \vee \text{loves}(\text{john}, \text{tea})$

$\neg \text{loves}(\text{john}, \text{coffee}) \vee \text{loves}(\text{john}, \text{lemonade})$

κατά συνέπεια *$\text{loves}(\text{john}, \text{tea}) \vee \text{loves}(\text{john}, \text{lemonade})$*

Αρχή της Ανάλυσης (συν)

- $P \vee \neg Q \Leftrightarrow P \leftarrow Q$ (Ισοδυναμία)
 - $P \Leftrightarrow P \leftarrow$ (γεγονός)
 - $\neg P \Leftrightarrow \leftarrow P$ (άρνηση)
- $P \vee \neg Q \vee \neg Z \Leftrightarrow P \leftarrow Q \wedge Z$ (γιατί)
- Αρχή ανάλυσης
 - $\leftarrow P, P \leftarrow Q \Rightarrow \leftarrow Q$
 - $\leftarrow P, P \leftarrow \Rightarrow \varepsilon$ (κενή πρόταση – άτοπο)
- Λογικό Πρόγραμμα
 - $Q \leftarrow$ **γεγονός**
 - $Z \leftarrow$ **γεγονός**
 - $P \leftarrow Q \wedge Z$ **κανόνας**
 - $\leftarrow P$ **ερώτηση**

Αρχή της Ανάλυσης (συν)

- Μια αυτοματοποιημένη διαδικασία οδηγεί σε εύκολη κωδικοποίηση σε Η/Υ.
- Η σημασία του κανόνα της αρχής της ανάλυσης:

**ΕΝΑΣ ΚΑΝΟΝΑΣ ΕΙΝΑΙ ΑΡΚΕΤΟΣ
ΓΙΑ ΝΑ ΑΠΟΔΕΙΞΕΙ ΤΑ ΠΑΝΤΑ**

Ανάλυση στο Λογικό Προγραμματισμό

- SLD – Resolution

- **S**election Function **L**inear Resolution for **D**efinite Clauses.

- Αναλυθέν (resolvent)

- Σύζευξη υποστόχων (subgoals)

- **Άρνηση** των πραγμάτων που θέλουμε να αποδείξουμε.

$\leftarrow A1 \wedge A2 \wedge A3 \dots A_n$

$\leftarrow \varepsilon$ (κενή πρόταση)

Βήμα της διαδικασίας απόδειξης

- Υπάρχουν πολλές εναλλακτικές ως προς το ποια πρόταση υποστόχος θα επιλεγεί σαν αναλυθέν και με ποιος από τους εναλλακτικούς κανόνες θα χρησιμοποιηθεί.
 - **Κανόνας Επιλογής** (Computation / Selection Rule)
 - **Κανόνας Αναζήτησης** (Search Rule)

Διαδικασία απόδειξης (1/2)

- **Κανόνας Επιλογής:** Ποιος υποστόχος A_i από την σύζευξη θα επιλεγεί για “απόδειξη”.
 - $\leftarrow A1 \wedge A2 \wedge \dots \wedge \mathbf{A_i} \wedge \dots \wedge A_n$
- **Κανόνας Αναζήτησης:** Εύρεση k προτάσεων από το πρόγραμμα
 - Για κάθε πρόταση η κεφαλή πρέπει να είναι ενοποιήσιμη με τον υποστόχο που επιλέχθηκε.
 $H1 \leftarrow B11, B12 \dots B1n \quad A_i\theta1 = H1\theta1$
...
 $Hk \leftarrow Bk1, Bk2 \dots Bkn \quad A_i\theta k = Hk\theta k$

Διαδικασία Απόδειξης (2/2)

- Δημιουργία k νέων αναλυθέντων (resolvents) με αντικατάσταση από τους αντίστοιχους κανόνες

$$\leftarrow A1 \wedge A2 \wedge \dots \wedge \mathbf{Ai} \wedge \dots \wedge An$$

γίνεται

$$\leftarrow (A1 \wedge A2 \wedge \dots \wedge \mathbf{B11, B12...B1n} \wedge \dots \wedge An)\theta1$$

($\theta1$ σε όλη τη σύζευξη)

...

$$\leftarrow (A1 \wedge A2 \wedge \dots \wedge \mathbf{Bk1, Bk2...Bkn} \wedge \dots \wedge An)\theta k$$

- Επιλογή ενός αναλυθέντος από τα k.
- Επανάληψη της διαδικασίας μέχρι να **προκύψει η κενή πρόταση**.

Εκτέλεση Προγραμμάτων Λογικού Προγραμματισμού

- **LUSH Resolution**

- Leftmost Uppermost Selection Heuristic

- **Leftmost Computation Rule**

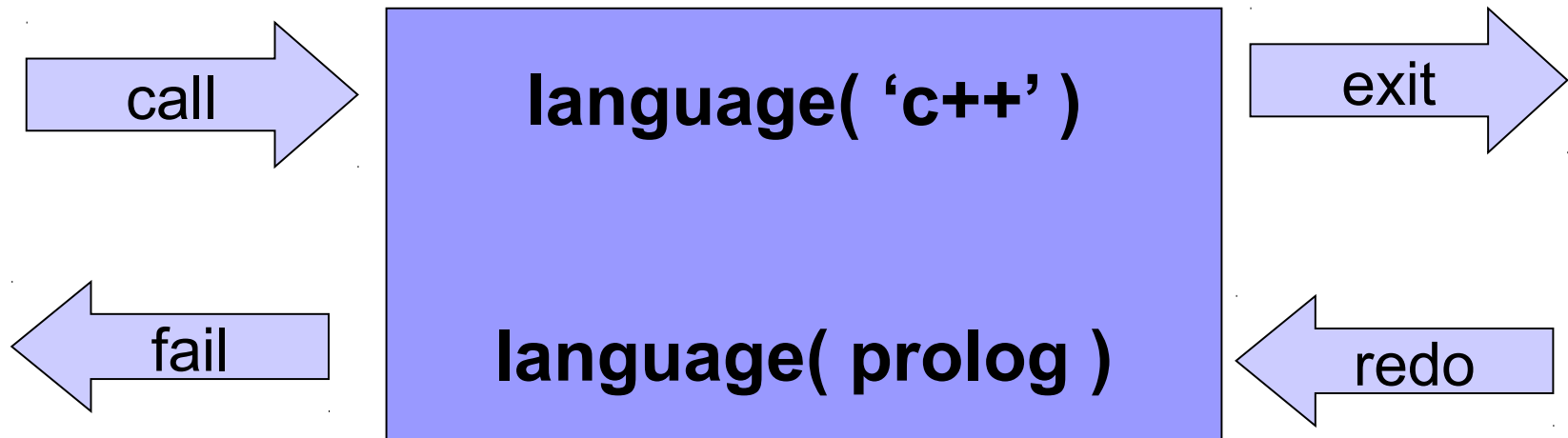
- Επιλογή του αριστερότερου υποστόχου στο αναλυθέν

- **Uppermost Search Rule**

- Επιλογή της πρώτης (upper) πρότασης που ενοποιείται.

Debugging σε Prolog

- Trace command
- Box model



Δομή

- Πως απαντά η Prolog στις ερωτήσεις;
 - Διαδικασία
 - Ενοποίηση και απόδοση τιμών σε μεταβλητές
 - Σύνολο απάντησης
 - Οπισθοδρόμηση (backtracking)
- Ισχυρά θεωρητικά θεμέλια