

3. Ο ΜΗΧΑΝΙΣΜΟΣ ΕΚΤΕΛΕΣΗΣ ΤΗΣ PROLOG

Η διαδικασία εκτέλεσης ενός προγράμματος της PROLOG βασίζεται σ' ένα μηχανισμό **ενοποίησης** και σε ένα μηχανισμό **επαγωγής** συμπερασμάτων.

3.1 Ενοποίηση

Η **ενοποίηση** (*unification*) αποτελεί την κεντρική διαδικασία υπολογισμού στην PROLOG. Κατά τη διάρκεια της εκτέλεσης ενός προγράμματος, οποιαδήποτε επεξεργασία δεδομένων γίνεται με τη βοήθεια της διαδικασίας αυτής. Για να ορίσουμε τη διαδικασία της ενοποίησης, είναι απαραίτητο να ορίσουμε πρώτα την έννοια της **αντικατάστασης** (*substitution*) και την έννοια του **στιγμιότυπου** (*instance*).

Ορισμός: **Αντικατάσταση** είναι ένα πεπερασμένο σύνολο (πιθανά κενό) από ζευγάρια της μορφής $X_i = t_i$, όπου X_i είναι μια μεταβλητή και t_i είναι ένας όρος. Πρέπει $X_i \neq X_j$ για κάθε $i \neq j$ και η μεταβλητή X_i να μην εμφανίζεται στον όρο t_j για κάθε i και j .

Οι αντικαταστάσεις μπορούν να εφαρμοστούν πάνω σε όρους. Το αποτέλεσμα της εφαρμογής της αντικατάστασης $\theta = \{X_1=t_1, \dots, X_n=t_n\}$ πάνω στον όρο t , συμβολίζεται σαν $t\theta$ ή $(t)\theta$ και είναι ο όρος που προκύπτει από την αλλαγή κάθε εμφάνισης της μεταβλητής X_i του όρου t με t_i , $1 \leq i \leq n$.

Η αλλαγή της εμφάνισης του X_i με t_i , μπορεί επίσης να αναφέρεται και σαν:

- ♦ **τοποθέτηση** του X_i με t_i ή
- ♦ **εντίμηση** του X_i με t_i

Ορισμός: Το A είναι ένα **στιγμιότυπο** του B αν υπάρχει αντικατάσταση θ τέτοια ώστε $A = B\theta$

Παραδείγματα:

Ο όρος $t(a,b)$ είναι ένα στιγμιότυπο του $t(X,b)$ διότι υπάρχει αντικατάσταση

$$\theta = \{X=a\},$$

τέτοια ώστε να ισχύει:

$$t(a,b) = (t(X,b))\theta$$

Ο όρος $parent(jim, john)$ είναι ένα στιγμιότυπο του $parent(X,Y)$ διότι υπάρχει αντικατάσταση

$$\theta = \{X=jim, Y=john\}$$

τέτοια ώστε να ισχύει:

$$parent(jim, john) = (parent(X,Y))\theta$$

Επίσης οι όροι $f(a,a)$ και $f(t_1(a), t_1(a))$ είναι στιγμιότυπα του $f(X,X)$ με αντικαταστάσεις $\{X=a\}$ και $\{X=t_1(a)\}$ αντίστοιχα.

Αντίθετα ο $f(a,b)$ δεν είναι στιγμιότυπο του $f(X,X)$ διότι δεν μπορούμε να αντικαταστήσουμε στον ίδιο όρο, την πρώτη εμφάνιση της μεταβλητής X με a και με τον διαφορετικό όρο b την δεύτερη εμφάνιση της μεταβλητής X .

Η διαδικασία της ενοποίησης δέχεται σαν είσοδο δύο όρους και αποφασίζει εάν οι δύο αυτοί όροι μπορούν να ενοποιηθούν. Στην περίπτωση που οι δύο όροι μπορούν να ενοποιηθούν, η διαδικασία της ενοποίησης επιστρέφει την τιμή **true**, ενώ στην αντίθετη περίπτωση επιστρέφει την τιμή **false**.

Δύο όροι t_1 και t_2 μπορούν να ενοποιηθούν εάν:

- ☛ ο t_1 είναι μεταβλητή, ο t_2 οτιδήποτε
- ☛ ο t_1 είναι σταθερά, ο t_2 η ίδια σταθερά

- ο t_1 είναι σύνθετος όρος με συναρτησιακό σύμβολο f και n ορίσματα, ο t_2 είναι επίσης σύνθετος όρος με συναρτησιακό σύμβολο f και n ορίσματα και τα n ορίσματα του t_1 μπορούν να ενοποιηθούν με τα αντίστοιχα n ορίσματα του t_2 .

Όταν η ενοποίηση των δύο όρων πετύχει, παράγεται ένας τρίτος όρος, ο οποίος έχει όσο το δυνατόν περισσότερες από τις μεταβλητές του τοποθετημένες. Ο όρος αυτός ονομάζεται **ο πιο γενικός ενοποιητής (the most general unifier)**. Πιο αυστηρά οι έννοιες της ενοποίησης και του πιο γενικού ενοποιητή δίνονται από τον παρακάτω ορισμό:

Ορισμός: Μια αντικατάσταση θ ονομάζεται **ενοποιητής** των όρων t_1 και t_2 αν $t_1\theta = t_2\theta$, η δε εφαρμογή της θ πάνω στους t_1 και t_2 ονομάζεται **ενοποίηση**. Μια τέτοια αντικατάσταση ονομάζεται **ο πιο γενικός ενοποιητής** των t_1 και t_2 , αν για κάθε άλλο ενοποιητή θ_1 των t_1 και t_2 το $t_1\theta_1$ είναι ένα στιγμιότυπο του $t_1\theta$.

Παραδείγματα ενοποίησης που πετυχαίνουν, μαζί με τον πιο γενικό ενοποιητή που παράγεται (θ) και τις αντίστοιχες αντικαταστάσεις των μεταβλητών δίνονται παρακάτω:

t_1	t_2	θ	Αντικατάσταση
X	red	red	{ X=red }
cheese	Y	cheese	{ Y=cheese }
size(X,big)	size(lion,Y)	size(lion,big)	{ X=lion,Y=big }
book(prolog)	book(prolog)	book(prolog)	{ }
same(dog,X)	same(S,S)	same(dog,dog)	{ S=dog,X=dog }
g(h(X),[])	g(Y,Z)	g(h(X),[])	{ Y=h(X),Z=[] }
t(s(p,q),P)	t(s(L,M),G)	t(s(p,q),G)	{ L=p,M=q,P=G }

Στα παρακάτω παραδείγματα η ενοποίηση αποτυχαίνει:

t_1	t_2
time	hour
s(X)	t(d)
s(X,4)	s(X,4,Y)
at(rome,bob)	at(X,X)
s(1,X,X)	s(Y,Y,2)
plus(2,5)	7

Παρατήρηση: Οτι αναφέρθηκε μέχρι τώρα για την ενοποίηση δύο όρων ισχύει με τον ίδιο ακριβώς τρόπο για την διαδικασία ενοποίησης δύο ατομικών τύπων: Δύο ατομικοί τύποι μπορούν να ενοποιηθούν εάν έχουν το ίδιο όνομα κατηγορήματος και τα αντίστοιχα ορίσματά τους μπορούν να ενοποιηθούν.

3.2 Ο Μηχανισμός Επαγωγής Συμπερασμάτων

Ο μηχανισμός επαγωγής συμπερασμάτων που χρησιμοποιεί η PROLOG βασίζεται σε ένα μοναδικό κανόνα συμπερασμού που ονομάζεται **ανάλυση (resolution)**. Σύμφωνα με τον κανόνα της ανάλυσης:

Δοθέντος ενός στόχου (ερώτησης) G

$$?- A_1, \dots, A_{i-1}, A_i, A_{i+1}, \dots, A_n. \quad (I)$$

και μιας φράσης C

$$A :- B_1, \dots, B_m.$$

τέτοιας ώστε η κεφαλή της A και ο υποστόχος A_i να μπορούν να ενοποιηθούν και αν θ είναι ο πιο γενικός ενοποιητής των A και A_i , η ανάλυση παράγει το νέο στόχο G' :

$$?- (A_1, \dots, A_{i-1}, B_1, \dots, B_m, A_{i+1}, \dots, A_n)\theta. \quad (\text{II})$$

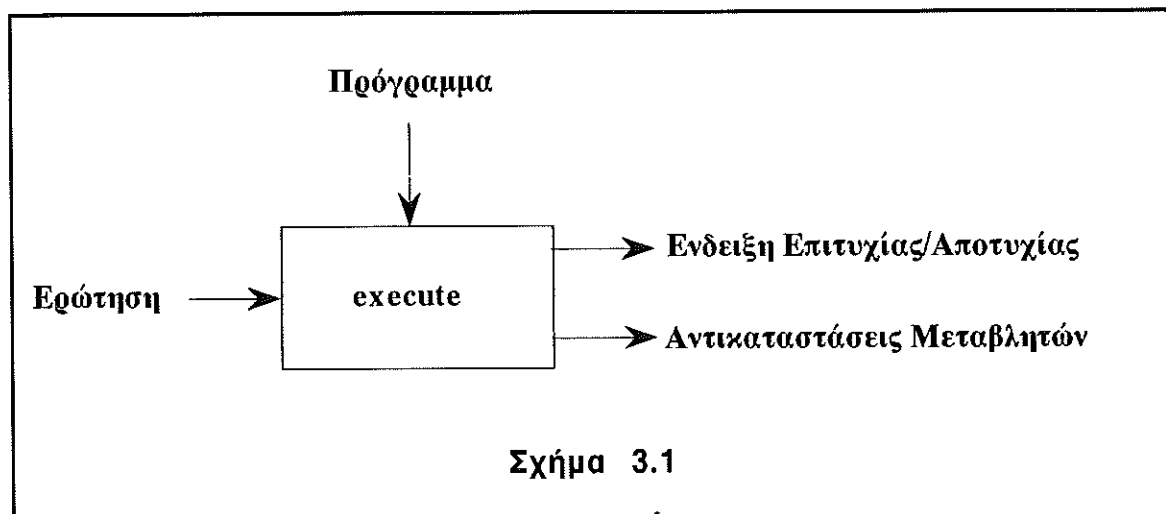
Η διαδικασία εκτέλεσης ενός στόχου βασίζεται στην επαναλαμβανόμενη εφαρμογή του παραπάνω κανόνα ανάλυσης, για την παραγωγή νέων στόχων από παλιότερους για κάθε υποστόχο A_i .

Η εκτέλεση ενός στόχου G είναι επιτυχής όταν ο στόχος αυτός μπορεί, μετά από ένα πεπερασμένο αριθμό βημάτων εφαρμογής του κανόνα ανάλυσης, να μετασχηματιστεί στον κενό στόχο. Ένας υποστόχος μετασχηματίζεται στον κενό υποστόχο εάν η φράση με την οποία μπορεί να ενοποιηθεί είναι ένα γεγονός.

Αν θεωρήσουμε το γενικό βήμα εφαρμογής του κανόνα ανάλυσης, στο στόχο (I) παρατηρούμε ότι εάν η φράση με την οποία ενοποιείται ο υποστόχος A_i είναι γεγονός τότε ο νέος στόχος G' που παράγεται είναι αντί του (II) ο εξής:

$$?- (A_1, \dots, A_{i-1}, A_{i+1}, \dots, A_n)\theta. \quad (\text{III})$$

Ο στόχος (III) είναι μειωμένος σε σχέση με τον (I) κατά ένα υποστόχο. Αυτού του είδους οι μειώσεις μπορούν τελικά να μετασχηματίσουν τον αρχικό στόχο στον κενό.



Στην περίπτωση που η εκτέλεση τερματίζεται με επιτυχία, αν ο αρχικός στόχος G , περιείχε μεταβλητές, τότε οι τιμές των μεταβλητών αυτών που προέκυψαν από τις αντίστοιχες ενοποιήσεις κατά τη διάρκεια της εκτέλεσης επιστρέφονται σαν απάντηση. Μια σχηματική αναπαράσταση της διαδικασίας εκτέλεσης φαίνεται στο σχήμα 3.1

Για την υλοποίηση ενός διερμηνέα (*interpreter*) της PROLOG που θα αναλαμβάνει την εκτέλεση των στόχων υπάρχουν δύο βασικά ζητήματα που πρέπει να αντιμετωπιστούν, σε σχέση με τον παραπάνω κανόνα ανάλυσης. Το πρώτο αφορά στον τρόπο επιλογής του υποστόχου A_i της ερώτησης G που θα αναλυθεί σε κάθε βήμα της εκτέλεσης. Το δεύτερο αφορά στον τρόπο επιλογής της φράσης C με την οποία θα ενοποιηθεί ο υποστόχος A_i . Ετσι σε κάθε βήμα της παραπάνω διαδικασίας η PROLOG:

- (α) διαλέγει πάντοτε τον υποστόχο που βρίσκεται πιο αριστερά στην ερώτηση. (Αρχικά τον A_1 στη συνέχεια τον A_2 και τελικά τον A_n).
- (β) καθορίζει τη σειρά των φράσεων C που δοκιμάζονται για να ενοποιηθούν με τον υπό εκτέλεση υποστόχο (αν υπάρχουν περισσότερες από μία), να είναι εκείνη που αυτές εμφανίζονται στο κείμενο του προγράμματος (ή ισοδύναμα στο χώρο εργασίας).

Αν σε οποιαδήποτε στιγμή η διαδικασία εκτέλεσης της PROLOG αποτύχει να βρεί ενοποιητή για έναν υποστόχο A_{i+1} **οπισθοδρομεί (*backtracks*)**. Κατά τη διαδικασία της οπισθοδρόμησης απορρίπτει τη φράση C με την οποία ενοποιηθηκε ο υποστόχος A_i και ακυρώνει οποιεσδήποτε αντικαταστάσεις προέκυψαν από τη διαδικασία αυτής της ενοποίησης. Στη συνέχεια ξαναθεωρεί τον υποστόχο A_i και προσπαθεί να βρεί εναλλακτικό τρόπο εκτέλεσης του. (Προσπαθεί δηλαδή να βρεί την επόμενη φράση του προγράμματος της οποίας η κεφαλή μπορεί επίσης να ενοποιηθεί με τον A_i).

Παραδείγματα εκτέλεσης δίνονται στη συνέχεια:

Παράδειγμα 1:

Εάν θεωρήσουμε το παρακάτω πρόγραμμα :

(P1) **big(horse).**
(P2) **big(elephant).**
(P3) **big(bear).**
(P4) **small(cat).**
(P5) **brown(bear).**
(P6) **brown(horse).**
(P7) **black(cat).**
(P8) **gray(elephant).**
(P9) **dark(X) :- blacK(X).**
(P10) **dark(X) :- brawn(X).**

τα διαδοχικά βήματα που ακολουθεί η PROLOG για την εκτέλεση της ερώτησης:

(G) **?- dark(Z), big(Z).**

είναι τα εξής:

B1] Η PROLOG αναζητεί μέσα στο πρόγραμμα την πρώτη φράση της οποίας η κεφαλή μπορεί να ενοποιηθεί με τον πρώτο υποστόχο **dark(Z)** της **G**. Η φράση (P9) πληρεί αυτές τις προϋποθέσεις. Το σώμα της φράσης αυτής αντικαθιστά τον πρώτο υποστόχο και η αρχική ερώτηση μετασχηματίζεται στην:

(G⁽¹⁾) **?- black(Z), big(Z).**

{ Εδώ πρέπει να παρατηρήσουμε ότι η φράση (P10) αποτελεί εναλλακτική λύση. Για το λόγο αυτό η PROLOG σημειώνει τη φράση αυτή σαν σημείο οπισθοδρόμησης (backtracking point). }

B2] Αναζητείται φράση που να μπορεί να ενοποιηθεί με τον υποστόχο **black(Z)** της **G⁽¹⁾**. Η φράση (P7) πληρεί τις προϋποθέσεις με την

αντικατάσταση $\{Z=cat\}$ και επειδή η φράση αυτή είναι γεγονός η $G^{(1)}$ μετασχηματίζεται στην:

$(G^{(2)}) \quad ?- \text{big}(cat).$

- B3]** Η εκτέλεση του υποστόχου $G^{(2)}$ αποτυχαίνει, διότι δεν υπάρχει φράση με την οποία να μπορεί να ενοποιηθεί. Η διαδικασία εκτέλεσης οπισθοδρομεί στο βήμα **B1** και η αντικατάσταση $\{Z=cat\}$ ακυρώνεται. Τώρα η ερώτηση είναι ξανά η:

$?- \text{black}(Z), \text{big}(Z).$

Αναζητείται φράση μετά την **(P7)** που να μπορεί να ενοποιηθεί με τον υποστόχο $\text{black}(Z)$. Τέτοια φράση δεν υπάρχει, οπότε η διαδικασία εκτέλεσης οπισθοδρομεί και αναζητεί εναλλακτική φράση μετά την **(P9)** με την οποία μπορεί να ενοποιηθεί ο υποστόχος $\text{dark}(Z)$ της αρχικής ερώτησης G . Η φράση **(P10)** επιλέγεται και η G μετασχηματίζεται τώρα στην:

$(G^{(3)}) \quad ?- \text{brawn}(Z), \text{big}(Z).$

{ Στην πράξη μπορούμε να θεωρήσουμε ότι η διαδικασία εκτέλεσης της PROLOG οπισθοδρομεί κατ' ευθείαν στον υποστόχο $\text{dark}(Z)$ της G και επιλέγει τη φράση **(P10)** γιατί εκεί βρίσκεται το πιο πρόσφατο σημείο οπισθοδρόμησης. }

- B4]** Αναζητείται φράση που να μπορεί να ενοποιηθεί με τον υποστόχο $\text{brawn}(Z)$ της $G^{(3)}$. Η φράση **(P5)** πληρεί τις προϋποθέσεις με την αντικατάσταση $\{Z=bear\}$ και η $G^{(3)}$ μετασχηματίζεται στην:

$(G^{(4)}) \quad ?- \text{big}(bear).$

{ Η φράση **(P6)** αποτελεί σημείο οπισθοδρόμησης. }

B5] Ο υποστόχος **big(bear)** μπορεί να ενοποιηθεί με την φράση (**P3**) και επειδή η φράση αυτή είναι γεγονός η $G^{(4)}$ μετασχηματίζεται στην κενή ερώτηση. Αυτό σημαίνει ότι η εκτέλεση τερματίζεται με επιτυχία και η αντικατάσταση $\{Z=bear\}$ αποτελεί μία λύση.

Στο σημείο αυτό εάν ο χρήστης ζητήσει από το σύστημα της PROLOG εναλλακτική λύση, ο μηχανισμός εκτέλεσης θεωρεί ότι η τελική ερώτηση $G^{(4)}$ έχει αποτύχει και επιστρέφει στο πιό πρόσφατο σημείο οπισθοδρόμησης για να συνεχίσει από εκεί την εκτέλεση:

B6] Επιλέγεται η φράση (**P6**) που μπορεί να ενοποιηθεί με τον υποστόχο **brown(Z)** της $G^{(3)}$ με την αντικατάσταση $\{Z=horse\}$ και η $G^{(3)}$ μετασχηματίζεται στην:

$G^{(5)}$?- **big(horse).**

B7] Ο υποστόχος **big(horse)** μπορεί να ενοποιηθεί με την φράση (**P1**) και επειδή η φράση αυτή είναι γεγονός η $G^{(5)}$ μετασχηματίζεται στην κενή ερώτηση. Αυτό σημαίνει ότι η εκτέλεση τερματίζεται με επιτυχία και η αντικατάσταση $\{Z=horse\}$ αποτελεί επίσης λύση.

Παράδειγμα 2:

Ας θεωρήσουμε το πρόγραμμα 1.2 που δόθηκε σε προηγούμενη παράγραφο και την ερώτηση:

(G) ?- **located_in(sindos,northGreece).**

Τα βήματα της εκτέλεσης της ερώτησης είναι τα εξής:

Η αρχική ερώτηση **G** μετά από την ενοποίηση της με τη φράση (**L8**) και με αντικατάσταση $\{X=sindos\}$ μετασχηματίζεται στην:

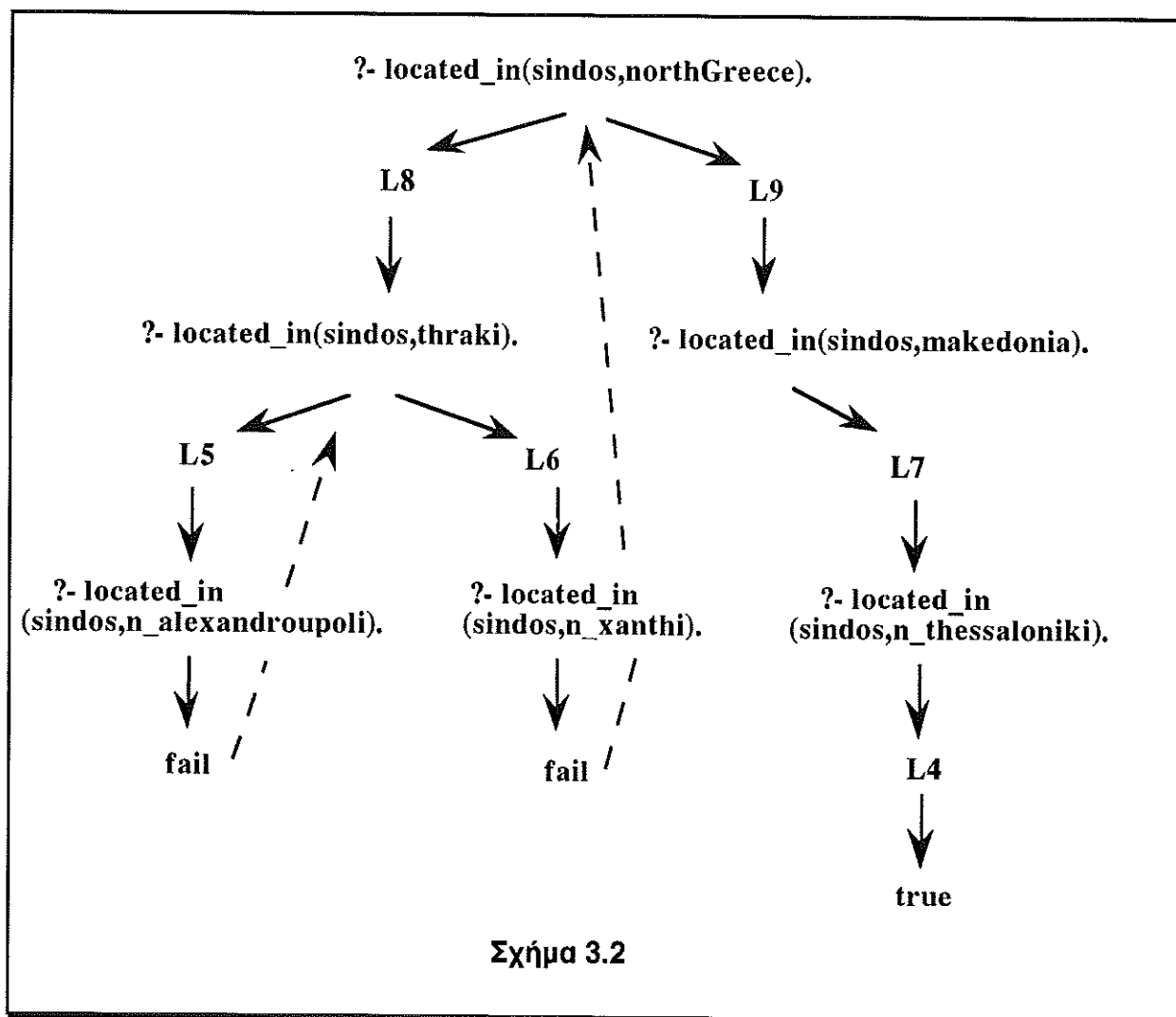
($G^{(1)}$) ?- **located_in(sindos,thraki).**

Με τη βοήθεια της φράσης (L5) η $G^{(1)}$ μετασχηματίζεται στην:

($G^{(2)}$) $?- \text{located_in}(\text{sendos}, \text{n_alexandroupoli}).$

Η $G^{(2)}$ αποτυχαίνει και η διαδικασία της εκτέλεσης οπισθοδρομεί στην ερώτηση $G^{(1)}$, οποία με τη βοήθεια της (L6) μετασχηματίζεται στην:

($G^{(3)}$) $?- \text{located_in}(\text{sendos}, \text{n_xanthi}).$



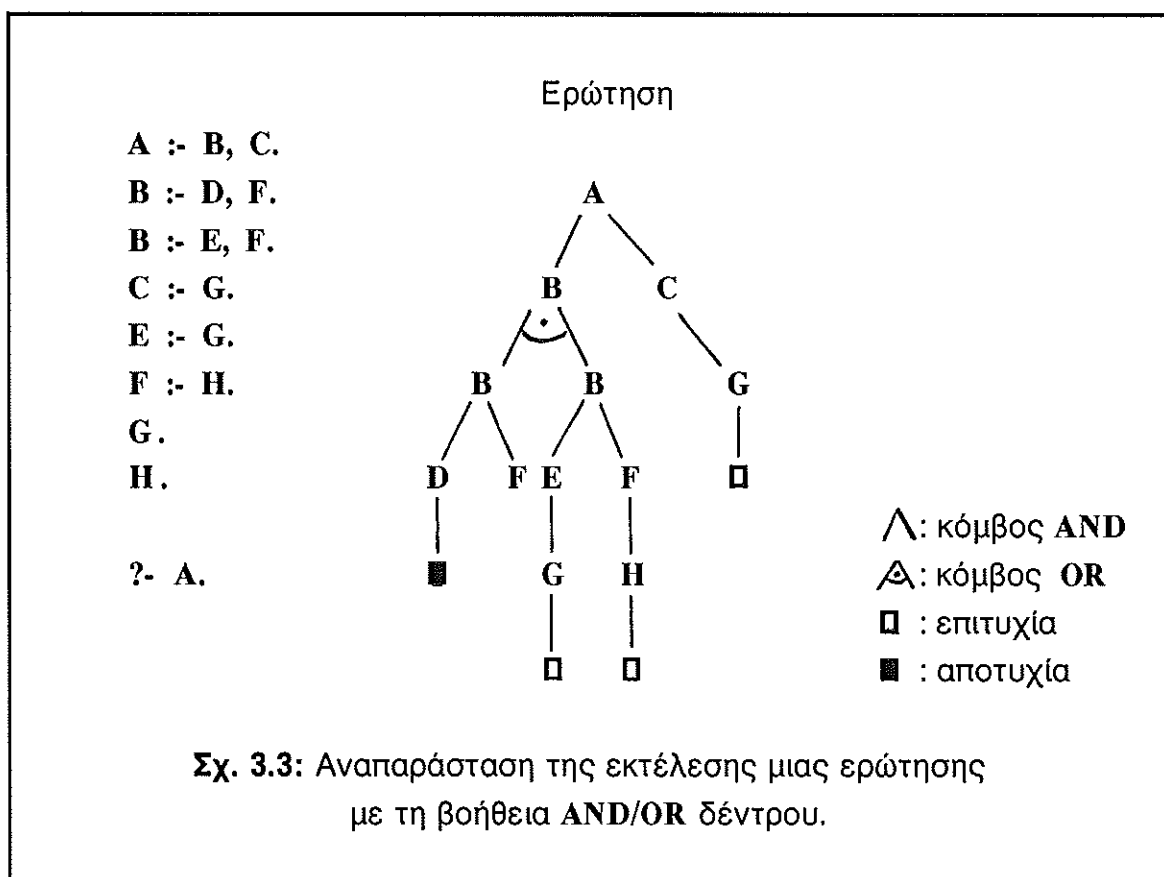
Η $G^{(3)}$, επίσης αποτυχαίνει και η διαδικασία εκτέλεσης οπισθοδρομεί στην αρχική ερώτηση G . Αυτή με τη βοήθεια της φράσης (L9) μετασχηματίζεται στην:

($G^{(4)}$) ?- located_in(sindos,makedonia).

Η $G^{(4)}$ με τη βοήθεια της φράσης (L7) μετασχηματίζεται στην:

($G^{(5)}$) ?- located_in(sindos,n_thessaloniki).

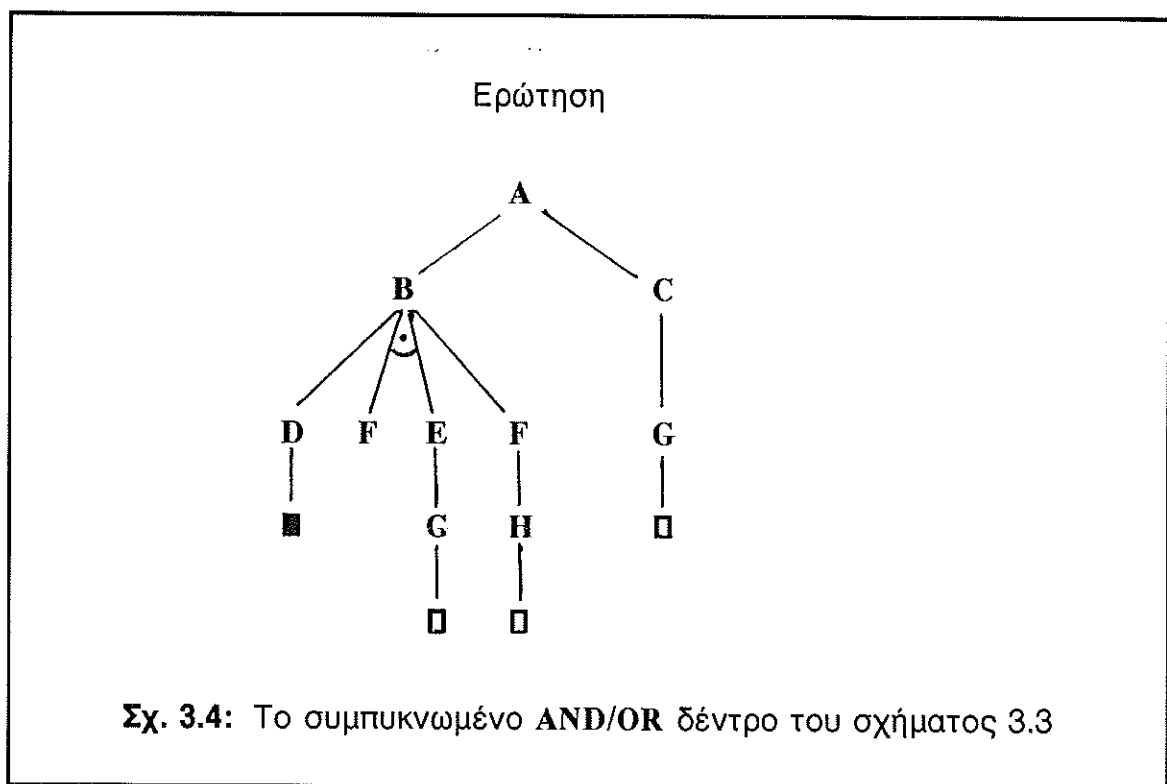
Η $G^{(5)}$ μπορεί να ενοποιηθεί με το γεγονός (L4) και μετασχηματίζεται στην κενή φράση, που σημαίνει ότι η εκτέλεση της αρχικής ερώτησης τερματίζεται με επιτυχία.



Η διαδικασία εκτέλεσης της ερώτησης φαίνεται σχηματικά στο σχήμα 3.2 (Η διαδικασία της οπισθοδρόμησης σημειώνεται στο σχήμα με τη βοήθεια διακεκομμένης γραμμής.)

Γενικά η όλη διαδικασία της εκτέλεσης στην PROLOG μπορεί να παρασταθεί με την βοήθεια ενός **AND/OR** δέντρου. Σε κάθε κόμβο **AND** αντιστοιχεί κάποιος στόχος **G**. Στα κλαδιά του κόμβου **AND** αντιστοιχούν οι υποστόχοι που η σύζευξή τους μπορεί να αντικαταστήσει το στόχο **G**. Σε κάθε στόχο **G** μπορεί επίσης να αντιστοιχεί ένας **OR** κόμβος, τα κλαδιά του οποίου παριστάνουν τις κεφαλές των φράσεων που μπορούν να ενοποιηθούν εναλλακτικά με το στόχο. Παράδειγμα χρήσης **AND/OR** δέντρου δίνεται στο σχήμα 3.3.

Σε πολλές περιπτώσεις οι κεφαλές των φράσεων που μπορούν να ενοποιηθούν εναλλακτικά με κάποιο στόχο δεν αναγράφονται στο **AND/OR** δέντρο, με αποτέλεσμα, το δέντρο αυτό να έχει μία πιά συμπυκνωμένη μορφή (σχήμα 3.4).



3.3 Ασκήσεις

1] Θεωρήστε το παρακάτω πρόγραμμα:

```
a(1,1).    a(2,1).  
a(3,2).    a(4,4).  
b(1,2).    b(1,3).  
b(2,3).    b(3,2).  
b(4,4).
```

Ποιές είναι όλες οι απαντήσεις, που θα δώσει η PROLOG στις παρακάτω ερωτήσεις:

- a) ?- a(X,Y), b(X,Y).
- b) ?- a(X,Y), b(X,Y), a(Y,Y).

2] Εστω το παρακάτω πρόγραμμα:

```
boss(dick,harry).  
boss(tom,dick).  
boss(ann,mary).  
boss(marry,harry).  
boss(harry,nick).
```

Περιγράψτε αναλυτικά τα βήματα που θα κάνει η PROLOG για να απαντήσει την παρακάτω ερώτηση (Θεωρήστε ότι ζητούνται όλες οι λύσεις.):

?- boss(X,Y),boss(Y,Z).

3] Εστω το σύνολο των φράσεων:

```
human(turing).  
human(socrates).  
greek(socrates).  
fallible(X) :- human(X).
```

Τι παράγει η Αρχή της Ανάλυσης αν εφαρμοστεί στην παρακάτω ερώτηση:

?- fallible(X), greek(X).

(Δώστε όλα τα βήματα μέχρι να απαντηθεί η ερώτηση)