Υπολογισμός στο Λογικό Προγραμματισμό

Πώς υπολογίζεται η έξοδος ενός Λογικού Προγράμματος;

Herbrand Universe H_L

- Είναι τα <u>δεδομένα</u> που μεταχειρίζεται ένα Λογικό Πρόγραμμα, προκειμένου να απαντήσει μια ερώτηση. Όλες οι <u>σταθερές</u> που διαθέτει ο μεταφραστής προκειμένου να κατασκευάσει αποδείξεις.

- Το σύνολο των <u>σταθερών όρων</u> που κατασκευάζονται χρησιμοποιώντας τα σύμβολα σταθερών και συναρτήσεων του αλφαβήτου της L.

ΛΠ για ακέραιο

```
int(0).
int(s(X)) :- int(X).
int(p(X)):- int(X).
```

- L_A={0} και s(X)=X+1, p(X)=X-1
- H_L={0, s(0), p(0), s(s(0)), s(s(s(0))), p(p(0)), p(p(p(0))), s(p(0)), p(s(0)), ….}

Herbrand Base of P: B(P)

- Η B(P) ενός προγράμματος P είναι το σύνολο όλων των σταθερών ατομικών προτάσεων που τα κατηγορήματά τους ορίζονται στο P, και τα ορίσματά τους ανήκουν στο H_L του P.

- Η B(P) αντιπροσωπεύει το σύνολο όλων των ατομικών κλειστών ερωτήσεων που μπορούν να απαντηθούν από το πρόγραμμα P.

- Κλειστή είναι η ερώτηση που απαντάται με ναι/όχι. Οι κλειστές ερωτήσεις είναι σταθερές. Σταθερή είναι η ερώτηση που δεν περιέχει μεταβλητές.

Παράδειγμα

```
likes(chris, Anyone):-
  buys(Anyone, this_book).
buys(Anyone, this_book):-
  sensible(Anyone).
sensible(you).
```

H_L={chris, this_book, you}

Κατηγορήματα του P={likes/2, buys/2, sensible/1}

B(P)

{ likes(chris, chris), likes(chris, you), likes(chris, this_book), likes(you, chris), likes(you, you), likes(you, this_book), likes(this_book, chris), likes(this_book, you), likes(this_book, this_book), buys(chris, chris), buys(chris, you), buys(chris, this_book), buys(you, chris), buys(you, you), buys(you, this_book), buys(this_book, chris), buys(this_book, you), buys(this_book, this_book),

sensible(chris), sensible(you), sensible(this_book)}

- Κάποιες από τις σταθερές ατομικές προτάσεις της B(P) θα απαντηθούν θετικά, και κάποιες αρνητικά, εφόσον τεθούν ως ερωτήσεις στο P.

- (παράδειγμα με βάση το B(P)).

- Πώς υπολογίζονται απαντήσεις σε ερωτήσεις;

Στιγμιότυπο Προγράμματος G(P)

- Η CNF μπορεί να προσεγγισθεί από την προτασιακή λογική, δηλαδή, κάθε πρόταση της μορφής:

$$(\forall X_1)\ldots(\forall X_n)\ W(X_1\ldots X_n)$$

μπορεί να αντικατασταθεί από το σύνολο έμμεσα συζευγμένων σταθερών προτάσεων της μορφής:

$$W(t) \text{ όπου } t \in H_L^{\,n}$$

- Οπότε κάθε πρόγραμμα P έχει σταθερό <u>στιγμιότυπο</u> G(P), που αποτελείται από όλα τα σταθερά στιγμιότυπα των προτάσεων του P.

Παράδειγμα G(P)

```
likes(chris,X):- likes(X,logic).
likes(bob, logic).

H={chris, bob, logic}.
```

Το στιγμιότυπο αυτού του προγράμματος είναι G(P):

likes(chris,chris):-likes(chris,logic).
likes(chris,bob):-likes(bob,logic).
likes(chris,logic):-likes(logic,logic).
likes(bob,logic).

Τρόποι υπολογισμού

- Επανειλημμένες εφαρμογές modus ponens. (forward reasoning)
- Modus tollens + απαγωγή σε άτοπο (backward reasoning, proof by contradiction).

Modus ponens

- Απαλοιφή συνεπαγωγής: {p→q, p} / q
- Εφαρμόζοντας στο G(P)

```
likes(chris,chris):-likes(chris,logic).
likes(chris,bob):-likes(bob,logic).
likes(chris,logic):-likes(logic,logic).
likes(bob,logic).
```

Ποιες προτάσεις προκύπτουν στο <u>σύνολο απαντήσεων</u>; (Αντιστοιχούν σε κλειστές ερωτήσεις που απαντώνται καταφατικά. Οι υπόλοιπες κλειστές ερωτήσεις απαντώνται αρνητικά).

Modus tollens + proof by contradiction

- $\{p \rightarrow q, \neg q\} / \neg p$.
- Απόδειξη μέσω του κανόνα μετακίνησης λεκτικών:

$$p \rightarrow q \equiv \neg q \rightarrow \neg p$$

Εφαρμογή στο G(P):

likes(chris,chris):-likes(chris,logic).
likes(chris,bob):-likes(bob,logic).
likes(chris,logic):-likes(logic,logic).
likes(bob,logic).

Έστω ότι θέλουμε να επιβεβαιώσουμε το αποτέλεσμα **likes(chris,bob)**

{likes(chris,bob)←likes(bob,logic), ¬**likes(chris,bob)**} / ¬likes(bob, logic)
{likes(bob, logic) ← True, ¬likes(bob, logic)} / **False**

Άρα η υπόθεσή μας ότι ¬**likes(chris,bob)** δεν ισχύει, επομένως ισχύει **likes(chris,bob)**

Modus Tollens + proof by contradiction

- Για πρόγραμμα P με πιθανή απάντηση A:

$P \cup \{\neg A\} \models$ False iff $P \models$ (False $\leftarrow \neg \mathbf{A}$)

Δηλαδή

$P \cup \{\neg A\} \models$ False iff $P \models A$

Δηλαδή

$P \cup \{\neg A\} \models$ False iff $P \vdash A$

Θεώρημα Herbrand

Μπορεί να αποδειχθεί μια αντίφαση (False) από ένα σύνολο προτάσεων $P \cup \{\neg A\}$

αν και μόνο αν

μπορεί να αποδειχθεί μια αντίφαση από κάποιο πεπερασμένο υποσύνολο του σταθερού στιγμιοτύπου $G(P \cup \{\neg A\})$

- Πρακτική Χρησιμότητα: Αν το $G(P \cup \{\neg A\})$ περιέχει προτάσεις g1, g2, ….gk τότε υπάρχει κάποιο m<k τέτοιο ώστε το {g1,…gm} περιέχει αντίφαση. Δηλαδή, στη χειρότερη περίπτωση, εξετάζοντας το $G(P \cup \{\neg A\})$ 'σειριακά', θα εντοπίσουμε την αντίφαση μετά από πεπερασμένο αριθμό βημάτων (τερματισμός, γραμμική πολυπλοκότητα). Σε αυτό το πνεύμα αναπτύχθηκαν οι <u>διαδικασίες Herbrand (διαδικασίες κορεσμού),</u> για τον υπολογισμό των απαντήσεων ενός λογικού προγράμματος.

Ανάλυση

- Πιο αποδοτικός τρόπος υπολογισμού.
- Ο modus ponens μπορεί να παραγάγει πολλές απαντήσεις που δεν μας ενδιαφέρουν (ανάλωση μνήμης).
- Οι διαδικασίες Herbrand τερματίζουν αλλά μπορεί να είναι χρονοβόρες, καθώς εφαρμόζονται πάνω στο G(P).
- Μπορούν να απαντηθούν και **ανοιχτές** ερωτήσεις (χωρίς προηγουμένως να έχουμε σκεφτεί όλες τις πιθανές απαντήσεις τους!).
- Ανοιχτές είναι οι ερωτήσεις που περιέχουν μεταβλητές και δεν απαντώνται μόνο ναι/όχι. Απαντώνται επιστρέφοντας τις τιμές εκείνες που, αν αντικατασταθούν στις μεταβλητές, καθιστούν την ερώτηση κατάφαση.
- Γενική Μορφή: **$P \cup \neg\{Q\}$ |- False αν και μόνο αν P|-Q**

Μορφές Ανοιχτής Ερώτησης

```
likes(chris,X) :- likes(X, logic).
likes(bob, logic).
```

- ?-likes(chris,X) Ποιος/τι αρέσει στον chris?
  - αναμενόμενη απάντηση {X=bob}
- ?-likes(Z, logic) Σε ποιον αρέσει η λογική?
  - αναμενόμενη απάντηση {Z=bob}
- ?-likes(X, Y)   Ποιος/τι αρέσει σε ποιον?
  - αναμενόμενη απάντηση {(X=chris and Y=bob), (X=bob and Y=logic)}

Γράφημα ανάλυσης

```
likes(chris,X) :- likes(X, logic).
likes(bob, logic).
```

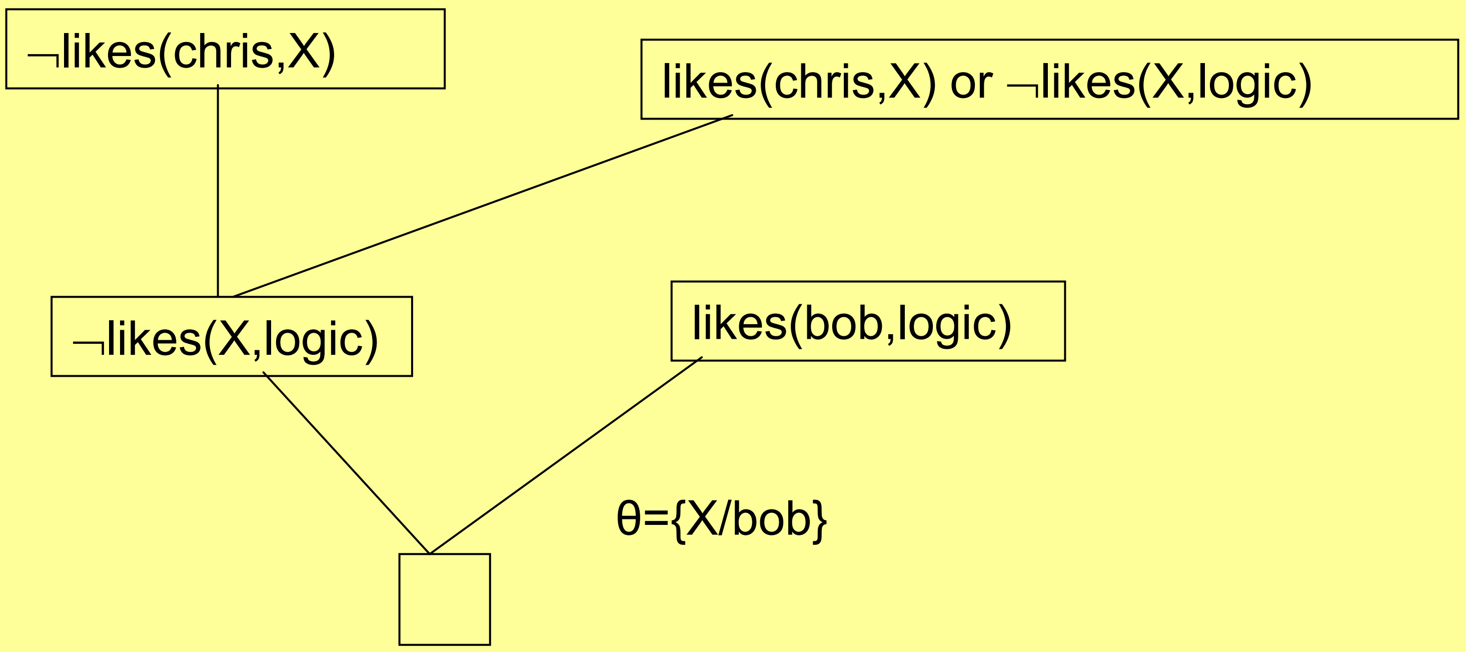
Σε CNF:

(C1) likes(chris,X)or$\neg$ likes(X, logic).

(C2) likes(bob, logic).

Έστω η ερώτηση ?- likes(chris, X)

(Q) $\neg$likes(chris,X)


¬likes(chris,X)
likes(chris,X) or ¬likes(X,logic)
¬likes(X,logic)
likes(bob,logic)
θ={X/bob}

Δέντρο αναζήτησης ανάλυσης

- Η ερώτηση τίθεται στη ρίζα
- Ο διάδοχος ενός κόμβου είναι εκείνος που περιέχει τις συνθήκες της πρότασης του προγράμματος, της οποίας το συμπέρασμα ταιριάζει (ενοποιείται) με τον κόμβο. Η ενοποίηση εφαρμόζεται στις συνθήκες της πρότασης πριν αυτές τοποθετηθούν στο διάδοχο κόμβο.

- παράδειγμα:

```
likes(chris,X) :- likes(X, logic).
likes(bob, logic).
```

- ερώτηση `?-likes(chris, X)`

?-likes(chris,X)

ενοποίηση με C1

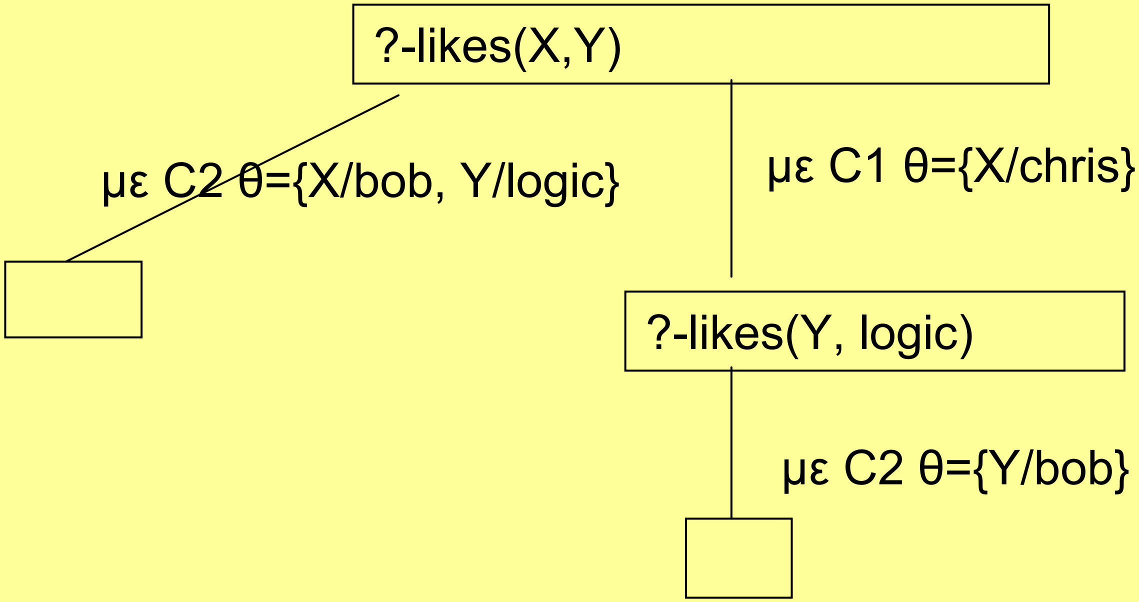
?-likes(X,logic)

ενοποίηση με C2 για θ={X/bob}

Δέντρο ανάλυσης με πολλαπλά κλαδιά

- Εμφανίζονται διακλαδώσεις σε έναν κόμβο όταν υπάρχουν περισσότερες από μια ενοποιήσεις του με το συμπέρασμα κάποιας πρότασης του προγράμματος.

- Οι απαντήσεις που επιστρέφει το πρόγραμμα συντίθενται από τις ενοποιήσεις που εμφανίζονται κατά μήκος κάθε μονοπατιού από την κενή πρόταση προς τη ρίζα.


?-likes(X,Y)
με C2 θ={X/bob, Y/logic}
με C1 θ={X/chris}
?-likes(Y, logic)
με C2 θ={Y/bob}