

Άσκηση Prolog

Δίνονται δύο, αρχικά κενά, δοχεία που μπορούμε να τα γεμίσουμε με νερό. Το ένα χωράει 3 λίτρα και το άλλο χωράει 2 λίτρα. Αυτό αποτυπώνεται ως εξής:

```
cup(a,3,0).  
cup(b,2,0).
```

Αν γεμίσουμε ένα δοχείο βάζοντάς το κάτω από βρύση, μπορούμε να ξέρουμε πόσο νερό θα περιέχει μόνο αν γεμίσει πλήρως (δηλαδή δε μπορούμε να σταματήσουμε το γέμισμα στα 2μισι λίτρα). Αυτό, επίσης, είναι ανεξάρτητο του πόσο νερό είχε πριν. Αυτό αποτυπώνεται ως εξής:

```
fill(X):-  
    cup(X,MaxVal,CurVal),  
    assert(cup(X,MaxVal,MaxVal)),  
    retract(cup(X,MaxVal,CurVal)).
```

Μας ζητάνε να γράψουμε ένα κατηγορημα που να μεταφέρει νερό από το ένα δοχείο στο άλλο. Η πρώτη μας προσπάθεια είναι η ακόλουθη:

```
move(From,To):-  
    cup(From,MaxFrom,CurFrom),  
    cup(To,MaxTo,CurTo),  
    EmptyTo is MaxTo-CurTo,  
    CurFrom >= EmptyTo,  
    fill(To),  
    NewFrom is CurFrom-EmptyTo,  
    assert(cup(From,MaxFrom,NewFrom)),  
    retract(cup(From,MaxFrom,CurFrom)).
```

(α) (3/10) Για να δοκιμάσετε το παραπάνω πρόγραμμα (συνενώνοντας τα παραπάνω τρία τμήματα) το φορτώνετε στον Prolog interpreter του ΗΥ σας και δίνετε τις ακόλουθες εντολές, τη μία μετά την άλλη:

```
fill(a).  
move(a,b).
```

Αποτυπώστε τα αποτελέσματα των εντολών `assert` και `retract` που παίρνουν μέρος στην εκτέλεση με τις αναγκαίες εξηγήσεις (δηλ. χωρίς να εξηγήσετε τις αποδόσεις τιμών στις μεταβλητές).

Από την εκτέλεση του `fill(a)`, λόγω του ορισμού `fill(X)`, η `assert` έχει ως αποτέλεσμα την εισαγωγή του γεγονότος `cup(a, 3, 3)` και η `retract` την διαγραφή του γεγονότος `cup(a, 3, 0)`.

Από την εκτέλεση του `move(a, b)`, λόγω του ορισμού του `move(From, To)`, καλείται η `fill(b)`, η `assert` και `retract` της οποίας έχουν σαν αποτέλεσμα την εισαγωγή του γεγονότος `cup(b, 2, 2)` και τη διαγραφή του `cup(b, 2, 0)` αντίστοιχα. Στη συνέχεια η `assert` και `retract` της `move(a, b)` έχουν σαν αποτέλεσμα την εισαγωγή του γεγονότος `cup(a, 3, 1)` και τη διαγραφή του `cup(a, 3, 3)`.

Στο τέλος, δηλαδή, το A θα περιέχει 1 λίτρο και το B θα περιέχει 2 λίτρα.

(β) (2/10) Υποθέστε πως αντί των παραπάνω εντολών είχατε δοκιμάσει τις ακόλουθες εντολές, τη μία μετά την άλλη:

```
fill(b).  
move(b,a).
```

Ποιό θα ήταν το αποτέλεσμα; Εξηγήστε.

A	B	Γ
Το Α θα περιέχει 0 λίτρα Το Β θα περιέχει 2 λίτρα	Το Α θα περιέχει 2 λίτρα Το Β θα περιέχει 0 λίτρα	Το Α θα περιέχει 3 λίτρο Το Β θα περιέχει 0 λίτρα

Η απάντηση είναι η Α. Λόγω του `CurFrom >= EmptyTo (2 >= 3)`, δε μπορεί να ολοκληρωθεί η εκτέλεση του `move`.

(γ) (2/10) Σε αντιστοιχία με το κατηγορήμα `fill`, γράψτε πως θα πρέπει να υλοποιηθεί το κατηγορήμα `empty`. Περιγράψτε την υλοποίησή σας πολύ συνοπτικά.

```
empty(X) :-  
    cup(X, MaxVal, CurVal),  
    assert(cup(X, MaxVal, 0)),  
    retract(cup(X, MaxVal, CurVal)).
```

Κατ' αναλογία με το γέμισμα, και το άδειασμα εννοείται ως πλήρες. Άρα προσθέτουμε το `fact` που μας λέει πως το δοχείο είναι πλέον άδειο και αφαιρούμε το `fact` που είχε την προηγούμενη τιμή για περιεχόμενο.

(δ) (3/10) Ολοκληρώστε την υλοποίηση του κατηγορήματος `move` προσθέτοντας ότι άλλο θεωρείτε απαραίτητο. Περιγράψτε την υλοποίησή σας πολύ συνοπτικά.

```
move(From, To) :-  
    cup(From, MaxFrom, CurFrom),  
    cup(To, MaxTo, CurTo),  
    EmptyTo is MaxTo - CurTo,  
    CurFrom < EmptyTo,  
    empty(From),  
    NewTo is CurTo + CurFrom,  
    assert(cup(To, MaxTo, NewTo)),  
    retract(cup(To, MaxTo, CurTo)).
```

Η υλοποίηση του `move` που δίνεται στην εκφώνηση καλύπτει μόνο τις περιπτώσεις που το δοχείο-προορισμός θα γεμίσει πλήρως, ενδεχομένως αφήνοντας κάποιο υπόλοιπο στο δοχείο-αφετηρία.

Με τον κώδικα που μόλις προσθέσαμε, καλύπτουμε την περίπτωση που το δοχείο-αφετηρία θα αδειάσει πλήρως, ενδεχομένως αφήνοντας κάποιο κενό στο δοχείο-προορισμό.

Κατ' αναλογία με τα κατηγορήματα `fill` και `empty`, δε νοείται περίπτωση να υπάρχει άλλο ενδεχόμενο (δηλαδή μετά από μετακίνηση να έχουμε δύο μισογεμάτα δοχεία).