

3η Σειρά Ασκήσεων

Οι παρακάτω ασκήσεις είναι για δική σας εξάσκηση και όχι για παράδοση. Οι λύσεις τους θα δοθούν αργότερα κατά τη διάρκεια του εξαμήνου.

Ασκηση 1.

Περιγράψτε τα βήματα του αλγόριθμου ταυτοποίησης για καθένα από τα παρακάτω ζεύγη όρων:

- (α) $f(X, g(a), Y)$ και $f(c, Y, g(X))$
- (β) $a(X, f(X), f(Y))$ και $a(g(b), Y, f(f(X)))$
- (γ) $p(s(2), X, Y * X)$ και $p(s(Y), 4, X + X)$
- (δ) $f(A, B/A)$ και $f(0, 1/A)$
- (ε) $[X, x, x | T]$ και $[T, Y, Z, 0, 0]$
- (στ) $[X, 2 | T]$ και $[T, X, 2]$

Ασκηση 2.

Έστω το παρακάτω πρόγραμμα Prolog:

```
p(X,Y) :- q(Z,X), q(Z,Y), q(Y,W), q(W,X). %1
q(d,f). %2
q(d,a). %3
q(f,b). %4
q(a,b). %5
q(a,c). %6
q(c,f). %7
```

Περιγράψτε τα βήματα που θα ακολουθήσει η Prolog για να απαντήσει στην ερώτηση:

?- p(A,B).

Ασκηση 3.

Έστω το παρακάτω πρόγραμμα Prolog:

```
p(A) :- b(A,B), r(B,C), b(C,D), r(D,A). % 1
b(2,1). % 2
b(3,4). % 3
b(5,2). % 4
b(1,6). % 5
r(4,5). % 6
r(4,1). % 7
r(6,3). % 8
r(6,6). % 9
```

Περιγράψτε τα βήματα που θα ακολουθήσει η Prolog για να απαντήσει στην ερώτηση:

?- p(Y).

Άσκηση 4.

Περιγράψτε τα βήματα που θα ακολουθήσει η Prolog για να απαντήσει στις παρακάτω ερωτήσεις (όπου τα `sum`, `gcdEuc`, `gcdFast`, `conc` και `delete` ορίζονται όπως στις σημειώσεις):

(α) `?- sum(X,s(s(0)),s(s(s(s(s(0))))))`.

(β) `?- gcdEuc(72,120,X)`.

(γ) `?- gcdFast(455,196,X)`.

(δ) `?- conc([1,a,[3,4],[]], [], L)`.

(ε) `?- delete(0,L,[1])`.

Στο (ε) υποθέστε ότι ο χρήστης δίνει ; μετά από κάθε απάντηση, ώστε να υπολογιστούν όλες οι τιμές της L.

Άσκηση 5.

Ορίστε κατηγορήματα σε Prolog για των υπολογισμό των συναρτήσεων `s` και `h` οι οποίες υλοποιούνται σε Haskell με τον παρακάτω τρόπο:

```
s :: Int->Int->Int
s m n
  | m <= 0 || n <= 0 = 0
  | m == 1 || n == 1 = n*m
  | otherwise = m * n * s (m 'div' 2) (n 'div' 2)
```

```
h :: [Int]->[Int]->Bool->[Int]
h [] s True = []
h (a:[]) s True = a:s
h (a:b:t) s True
  | a>b = h (a:t) (b:s) False
  | otherwise = h (b:t) (a:s) True
h q (x:y:w) False
  | x<y = h (y:q) (x:w) False
  | otherwise = h q (x:y:w) True
h q s False = h q s True
```

Άσκηση 6.

Το κατηγορήμα `p` στο παρακάτω πρόγραμμα Prolog υπολογίζει μία συνάρτηση με είσοδο το πρώτο όρισμα και έξοδο το δεύτερο όρισμα. Γράψτε ένα πρόγραμμα σε Haskell που να υπολογίζει την ίδια συνάρτηση.

```
p(N,[]) :- N =< 1.
p(N,[H|T]) :- N>1, d(N,2,H), e(N,H,M), p(M,T).
```

```
d(N,I,I) :- N mod I == 0.
d(N,I,H) :- N mod I > 0, J is I+1, d(N,J,H).
```

```
e(N,R,N) :- N mod R > 0.
e(N,R,M) :- N mod R == 0, K is N // R, e(K,R,M).
```