ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ (Ακαδ. Έτος 2014-15)

3η Σειρά Ασκήσεων - Λύση 4ης άσκησης

(α)

```
[0] sum(X,s(s(0)),s(s(s(s(s(0))))))
    {X = X}
    1 --> 4
    sum(X1,s(Y1),s(Z1)) :- sum(X1,Y1,Z1).
    {X1 = X, Y1 = s(0), Z1 = s(s(s(s(0))))}

[1] sum(X,s(0),s(s(s(s(0)))))
    {X = X}
    1 --> 4
    sum(X2,s(Y2),s(Z2)) :- sum(X2,Y2,Z2).
    {X2 = X, Y2 = 0, Z2 = s(s(s(0)))}

[2] sum(X,0,s(s(s(0))))
    {X = X}
    1 --> 3
    sum(X3,0,X3) :- nat(X3).
    {X = s(s(s(0))), X3 = s(s(s(0)))}

[4] nat(s(s(s(0))))
    {X = s(s(s(0)))}
    1 --> 2
    nat(s(X4)) :- nat(X4).
    {X4 = s(s(0))}

[4] nat(s(s(0)))
    {X = s(s(s(0)))}
    1 --> 2
    nat(s(X5)) :- nat(X5).
    {X5 = s(0)}

[5] nat(s(0))
    {X = s(s(s(0)))}
    1 --> 2
    nat(s(X6)) :- nat(X6).
    {X6 = 0}

[6] nat(0)
    {X = s(s(s(0)))}
    1 --> 1
    nat(0).
    {}
```

```
[7] #
    {X = s(s(s(0)))}
------------------------------------------------------------
X = s(s(s(0)))
```

(β)

```
 [0] gcdEuc(72,120,X)
     {X = X}
     1 --> 2
     gcdEuc(M1,N1,D1) :- M1 < N1, K1 is N1-M1, gcdEuc(M1,K1,D1).
     {M1 = 72, N1 = 120, D1 = X}

 [1] 72 < 120, K1 is 120-72, gcdEuc(72,K1,X)
     {X = X}
     < : build-in
     {}

 [2] K1 is 120-72, gcdEuc(72,K1,X)
     {X = X}
     is : build-in
     {K1 = 48}

 [3] gcdEuc(72,48,X)
     {X = X}
     1 --> 2
     gcdEuc(M2,N2,D2) :- M2 < N2, K2 is N2-M2, gcdEuc(M2,K2,D2).
     {M2 = 72, N2 = 48, D2 = X}

 [4] 72 < 48, K2 is 48-72, gcdEuc(72,K2,D2)
     {X = X}
     < : build-in
     failure - backtracking

 [3] gcdEuc(72,48,X)
     {X = X}
     3 --> 3
     gcdEuc(M3,N3,D3) :- M3 > N3, K3 is M3-N3, gcdEuc(N3,K3,D3).
     {M3 = 72, N3 = 48, D3 = X}

 [4] 72 > 48, K3 is 72-48, gcdEuc(48,K3,X)
     {X = X}
     < : build-in
     {}
```

```
 [5] K3 is 72-48, gcdEuc(48,K3,X)
     {X = X}
     is : build-in
     {K3 = 24}


 [6] gcdEuc(48,24,X)
     {X = X}
     1 --> 2
     gcdEuc(M4,N4,D4) :- M4 < N4, K4 is N4-M4, gcdEuc(M4,K4,D4).
     {M4 = 48, N4 = 24, D4 = X}

 [7] 48 < 24, K4 is 24-48, gcdEuc(48,K4,X).
     {X = X}
     < : build-in
     failure - backtracking

 [6] gcdEuc(48,24,X)
     {X = X}
     3 --> 3
     gcdEuc(M5,N5,D5) :- M5 > N5, K5 is M5-N5, gcdEuc(N5,K5,D5).
     {M5 = 48, N5 = 24, D5 = X}

 [7] 48 > 24, K5 is 48-24, gcdEuc(24,K5,X)
     {X = X}
     < : build-in
     {}

 [8] K5 is 48-24, gcdEuc(24,K5,X).
     {X = X}
     is : build-in
     {K5 = 24}

 [9] gcdEuc(24,24,X)
     {X = X}
     3 --> 3
     gcdEuc(N6,N6,N6).
     {X = 24, N6 = 24}

[10] #
     {X = 24}
------------------------------------------------------------
X = 24
```

(γ)

```
[0] gcdFast(455,196,X)
    {X = X}
    1 --> 2
    gcdFast(M1,N1,D1) :- M1 < N1, K1 is N1 mod M1, gcdFast(K1,M1,D1).
    {M1 = 455, N1 = 196, D1 = X}

[1] 455 < 196, K1 is 196 mod 455, gcdFast(K1,455,X)
    {X = X}
    < : build-in
    failure - backtracking

[0] gcdFast(455,196,X)
    {X = X}
    3 --> 3
    gcdFast(M2,N2,D2) :- M2 >= N2, K2 is M2 mod N2, gcdFast(K2,N2,D2).
    {M2 = 455, N2 = 196, D2 = X}

[1] 455 >= 196, K2 is 455 mod 196, gcdFast(K2,196,X)
    {X = X}
    >= : build-in
    {}

[2] K2 is 455 mod 196, gcdFast(K2,196,X)
    {X = X}
    is : build-in
    {K2 = 63}

[3] gcdFast(63,196,X)
    {X = X}
    1 --> 2
    gcdFast(M3,N3,D3) :- M3 < N3, K3 is N3 mod M3, gcdFast(K3,M3,D3).
    {M3 = 63, N3 = 196, D3 = X}

[4] 63 < 196, K3 is 196 mod 63, gcdFast(K3,63,X)
    {X = X}
    < : build-in
    {}

[5] K3 is 196 mod 63, gcdFast(K3,63,X)
    {X = X}
    is : build-in
    {K3 = 7}
```

```
 [6] gcdFast(7,63,X)
     {X = X}
     1 --> 2
     gcdFast(M4,N4,D4) :- M4 < N4, K4 is N4 mod M4, gcdFast(K4,M4,D4).
     {M4 = 7, N4 = 63, D4 = X}

 [7] 7 < 63, K4 is 63 mod 7, gcdFast(K4,7,X)
     {X = X}
     < : build-in
     {}

 [8] K4 is 63 mod 7, gcdFast(K4,7,X)
     {X = X}
     is : build-in
     {K4 = 0}

 [9] gcdFast(0,7,X)
     {X = X}
     1 --> 1
     gcdFast(0,M5,M5).
     {X = 7, M5 = 7}

[10] #
     {X = 7}

------------------------------------------------------------
X = 7
```

(δ)

```
[0] conc([1,a,[3,4],[]], [], L)
    {L = L}
    1 --> 2
    conc([H1|T1],L1,[H1|Q1]) :- conc(T1,L1,Q1).
    {H1 = 1, T1 = [a,[3,4],[]], L1 = [], L = [1|Q1]}

[1] conc([a,[3,4],[]], [], Q1)
    {L = [1|Q1]}
    1 --> 2
    conc([H2|T2],L2,[H2|Q2]) :- conc(T2,L2,Q2).
    {H2 = a, T2 = [[3,4],[]], L2 = [], Q1 = [a|Q2]}

[2] conc([[3,4],[]], [], Q2)
    {L = [1,a|Q2]}
    1 --> 2
    conc([H3|T3],L3,[H3|Q3]) :- conc(T3,L3,Q3).
    {H3 = [3,4], T3 = [[]], L3 = [], Q2 = [[3,4]|Q3]}

[3] conc([[]], [], Q3)
    {L = [1,a,[3,4]|Q3]}
    1 --> 2
    conc([H4|T4],L4,[H4|Q4]) :- conc(T4,L4,Q4).
    {H4 = [], T4 = [], L4 = [], Q3 = [[]|Q4]}

[4] conc([],[],Q4)
    {L = [1,a,[3,4],[]|Q4]}
    1 --> 1
    conc([],L5,L5).
    {L5 = [], Q4 = []}

[5] #
    {L = [1,a,[3,4],[]]}
------------------------------------------------------------
L = [1,a,[3,4],[]]
```

(ε)

```
 [0] delete(0,L,[1])
     {L = L}
     1 --> 1
     delete(X1,[X1|T1],T1).
     {L = [0,1], X1 = 0, T1 = [1]}

 [1] #
     {L = [0,1]}
------------------------------------------------------------
L = [0,1] ;
------------------------------------------------------------
 [0] delete(0,L,[1])
     {L = L}
     2 --> 2
     delete(X2,[H2|T2],[H2|S2]) :- delete(X2,T2,S2).
     {L = [1|T2], X2 = 0, H2 = 1, S2 = []}

 [1] delete(0,T2,[])
     {L = [1|T2]}
     1 --> 1
     delete(X3,[X3|T3],T3).
     {T2 = [0], X3 = 0, T3 = []}

 [2] #
     {L = [1,0]}
------------------------------------------------------------
L = [1,0] ;
------------------------------------------------------------
 [1] delete(0,T2,[])
     {L = [1|T2]}
     2 --> EOP
     failure - backtraching

 [0] delete(0,L,[1])
     {L = L}
     EOP
     failure
------------------------------------------------------------
no
```