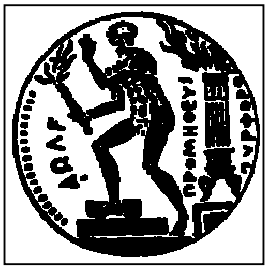

Εθνικό Μετσόβιο Πολυτεχνείο
Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Ηλεκτρονικών Υπολογιστών

Γλώσσες Προγραμματισμού 2
Άσκηση 3

Μάρτιος 2004

Τζάννες Αλέξανδρος

3. απλή μηχανική μετάφραση με ενοποιητική γραμματική

α. Χρησιμοποιώντας τον φορμαλισμό της ενοποιητικής γραμματικής **PATR II** σχεδιάστε μια γραμματική που να δέχεται ως είσοδο την *αριθμητική γραφή* (δεκαδικά ακέραια ψηφία) και να την μεταφράζει στην αντίστοιχη *ολογραφική* μορφή. Π.χ.:

είσοδος :	**15**	μετάφραση :	**δέκα πέντε**
είσοδος :	**342**	μετάφραση :	**τριακόσια σαράντα δύο**

β. Το αντίστροφο πώς θα μπορούσε να υλοποιηθεί. Περιγράψτε με δυο λόγια τις αλλαγές.

Για το πρώτο ερώτημα σχεδιάσαμε και υλοποιήσαμε σε φορμαλισμό ενοποιητικής γραμματικής PATR II μια γραμματική για μετατροπή αριθμών από την αριθμητική γραφή στην ολογραφική μορφή στα ελληνικά. Οι αριθμοί μπορούν να έχουν το πολύ 6 ψηφία, δηλαδή περιλαμβάνονται οι περιπτώσεις των αριθμών από το 1 ως το 999999. Οι κυριότερες δυσκολίες που συναντήσαμε και ξεπεράσαμε ήταν οι εξής:

1. Περιπτώσεις του 11, 12 που αποτελούν εξαίρεση σε σχέση με τους άλλους κανόνες μιας και αποτελούνται από 2 αριθμητικά ψηφία.
2. Η περίπτωση του 100 [εκατό και όχι εκατόν]
3. Η περίπτωση του χίλια

Αρχικά, να σημειώσουμε ότι τα ψηφία των αριθμών στην είσοδο πρέπει να χωρίζονται μεταξύ τους με κενά. Έχοντας παρατηρήσει ότι επιτρέπονται τα κενά μέσα στις λέξεις και για να αντιμετωπίσουμε το πρόβλημα των 11 και 12 προσπαθήσαμε να ξεγελάσουμε το σύστημα εισάγοντας στο λεξικό τις λέξεις “1 1” και “1 2”. Όμως αποδείχθηκε ότι αυτό όχι μόνο δεν έλυνε το πρόβλημα, αλλά ότι έκανε τη λύση αδύνατη. Παρότι για αριθμούς με το πολύ δύο ψηφία το τέχνασμα αυτό δούλεψε άψογα, για αριθμούς με 3 η περισσότερα ψηφία είχαμε το εξής πρόβλημα. Επειδή ο λεκτικός αναλυτής αναγνωρίζει τη μεγαλύτερη σε μήκος δυνατή λέξη κάθε φορά, και δεν έχει τη δυνατότητα να δώσει διαφορετικές λεκτικές αναλύσεις σε μια πρόταση, αριθμοί όπως ο 115 τεχνολογούνταν ως “ένδεκα πέντε” ενώ όταν μπήκαν οι αναγκαίοι περιορισμοί για το ποια θέση μπορεί να καταλαμβάνει το “ένδεκα” δεν ήταν δυνατό να γίνει η τεχνολόγηση. Αυτό συνέβη όπως είπαμε εξ’ αιτίας του λεκτικού αναλυτή του PCPATR. Έπρεπε επομένως να εκφραστούν τα 11 και 12 σαν εξαίρεση του γενικού κανόνα.

Ρίχνοντας μια ματιά στο λεξικό παρατηρούμε ότι όλα τα τερματικά σύμβολα ανήκουν στην κατηγορία (\c) DIGIT. Παρατηρούμε επίσης ότι για κάθε αριθμητικό ψηφίο έχουμε περισσότερα τερματικά. Για το ψηφίο “1” για παράδειγμα έχουμε το “ένα”, “μία”, “δέκα” και “εκατόν”. Παρατηρούμε ότι τα “ένδεκα”, “δώδεκα” και “εκατό” δεν βρίσκονται στο λεξικό. Αυτό συμβαίνει γιατί, όπως είπαμε, πρόκειται για εξαιρέσεις και βρίσκονται μέσα στη γραμματική, όπως θα δούμε αργότερα. Παρατηρούμε ότι το γένος περιέχεται ως ιδιότητα μόνο όταν υπάρχουν διαφορετικές επιλογές για ένα ψηφίο (π.χ. ένα - μία). Το πιο σημαντικό feature των ψηφίων είναι το count. Αυτό υποδεικνύει σε ποια θέση μπορεί να εμφανιστεί το εν λόγω ψηφίο. Αυτό προϋποθέτει ότι χωρίζουμε νοητά τον αριθμό σε τριάδες ψηφίων, όπως κάνουμε όταν γράφουμε τον αριθμό με

τελείες στις χιλιάδες (π.χ. 54.325). Επειδή δεν έχουμε τη δυνατότητα να μετράμε με άμεσο τρόπο με το PCPATR χρησιμοποιούμε τη στοίβα. Έτσι <count> = 1 σημαίνει ότι το ψηφίο αυτό (με την αντίστοιχη ερμηνεία) μπορούμε να το συναντήσουμε στην πρώτη θέση, ενώ <count count> = 1 στη δεύτερη.

ΠΑΡΑΤΗΡΗΣΗ : Η τιμή **1** της ιδιότητας count δεν παίξει κανέναν ρόλο.

Τέλος παρατηρούμε τις ιδιότητες "iszero", "isone" και "istwo" που είναι true για τα αντίστοιχα ψηφία και δεν υπάρχουν για τα υπόλοιπα. Μια πιο ευνόητη υλοποίηση θα είχε και τις τιμές false των ιδιοτήτων στα υπόλοιπα ψηφία (δηλαδή κάθε τερματικό θα είχε τις 3 αυτές ιδιότητες με true ή false). Αυτό όμως θα μεγάλωνε σημαντικά το λεξικό, και αφού η υλοποίηση χωρίς αυτά έγινε εύκολα, προτιμήθηκε αυτή η οδός.

Η γραμματική στηρίζεται αρχικά στην ακόλουθη δομή :
NUMBER → TRIPLET TRIPLET
NUMBER → TUPLE TRIPLET
NUMBER → DIGIT TRIPLET
NUMBER → TRIPLET
NUMBER → TUPLE
NUMBER → DIGIT

Και εμπλουτίζεται με αρκετούς κανόνες οι οποίοι πηγάζουν από τις εξαιρέσεις που προαναφέραμε. Η γραμματική έχει συνολικά 29 κανόνες που καλύπτουν πλήρως όλες τις περιπτώσεις για το πολύ 6 ψηφία.

Ακολουθούν κάποια επιλεγμένα παραδείγματα εκτέλεσης:

```
0

1:
NUMBER
   |
 DIGIT
   0

NUMBER:
[ cat:   NUMBER
  label: Miden ]

1 parse found
```

```
1

1:
NUMBER
   |
 DIGIT
   1

NUMBER:
[ cat:   NUMBER
  label: Ena ]

1 parse found
0 0 (επιτρέπουμε και leading zeros)
```

```
1:
   NUMBER
      |
    TUPLE
      |
   TUPLES
   ___|____
 DIGIT  DIGIT
   0      0

NUMBER:
[ cat:   NUMBER
  label: Miden ]

1 parse found
```

```
0 1

1:
   NUMBER
      |
    TUPLE
   ___|____
 DIGIT  DIGIT
   0      1

NUMBER:
[ cat:   NUMBER
  label: Ena ]

1 parse found
```

```
1 1

1:
   NUMBER
      |
    TUPLE
      |
   TUPLES
   ___|____
 DIGIT  DIGIT
   1      1

NUMBER:
[ cat:   NUMBER
  label: Enteka ]

1 parse found
```

```
1 2

1:
   NUMBER
      |
```

```
    TUPLE
      |
   TUPLES
   ___|____
 DIGIT  DIGIT
   1      2

NUMBER:
[ cat:   NUMBER
  label: Dwdeka ]

1 parse found
```

```
1 3

1:
   NUMBER
      |
    TUPLE
   ___|____
 DIGIT  DIGIT
   1      3

NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Deka
             2:     Tria ] ]

1 parse found  (Οι αριθμοί από το 13 ως το 20, ενώ κανονικά γράφονται σε μία λέξη, τις γράφουμε σε δύο, όπως το 15 στην εκφώνηση της άσκησης).
```

```
6 1

1:
   NUMBER
      |
    TUPLE
   ___|____
 DIGIT  DIGIT
   6      1

NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Eksinta
             2:     Ena ] ]

1 parse found
```

```
1 0 0

1:
     NUMBER
        |
     TRIPLET
   _____|_____
 DIGIT     TUPLE
   1         |
```

```
          TUPLES
          ___|____
        DIGIT  DIGIT
          0      0

NUMBER:
[ cat:   NUMBER
  label: Ekato ]

1 parse found
```

```
1 0 1

1:
     NUMBER
        |
     TRIPLET
   _____|_____
 DIGIT     TUPLE
   1       ___|____
         DIGIT  DIGIT
           0      1

NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Ekaton
             2:     Ena ] ]

1 parse found
```

```
5 1 2

1:
     NUMBER
        |
     TRIPLET
   _____|_____
 DIGIT     TUPLE
   5         |
           TUPLES
           ___|____
         DIGIT  DIGIT
           1      2

NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Pentakosia
             2:     Dwdeka ] ]

1 parse found
```

```
1 0 0 0

1:
      NUMBER
   ______|______
 DIGIT       TRIPLET
   1       _____|_____
```

```
        DIGIT     TUPLE
          0         |
                  TUPLES
                  ___|____
                DIGIT  DIGIT
                  0      0
```

```
NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Xilia ] ]
```

1 parse found

---

1 0 0 1

1:

```
      NUMBER
   ______|______
 DIGIT       TRIPLET
   1       _____|_____
         DIGIT     TUPLE
           0       ___|____
                 DIGIT  DIGIT
                   0      1
```

```
NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Xilia
             2:     Ena ] ]
```

1 parse found

---

4 6 7 6

1:

```
      NUMBER
   ______|______
 DIGIT       TRIPLET
   4       _____|_____
         DIGIT     TUPLE
           6       ___|____
                 DIGIT  DIGIT
                   7      6
```

```
NUMBER:
[ cat:   NUMBER
  label:   [ 1:     Tessereis
             2:     Xiliades
             3:       [ 1:     Eksakosia
                        2:       [ 1:     Ebdominta
                                   2:     Eksi ] ] ] ]
```

1 parse found

---

7 6 9 1 2

1:

```
           NUMBER
```

```
        _________|________
     TUPLE              TRIPLET
   ___|____          _____|_____
 DIGIT  DIGIT  DIGIT       TUPLE
   7      6      9            |
                            TUPLES
                             ___|____
                          DIGIT  DIGIT
                            1      2

NUMBER:
[ cat:   NUMBER
  label:   [ 1:        [ 1:     Ebdominta
                          2:     Eksi ]
             2:     Xiliades
             3:        [ 1:     Enniakosia
                          2:     Dwdeka ] ] ]

1 parse found
```

```
9 9 9 9 9 9

1:
                 NUMBER
          ____________|____________
      TRIPLET                  TRIPLET
   _____|_____              _____|_____
 DIGIT      TUPLE        DIGIT       TUPLE
   9          ___|____     9          ___|____
           DIGIT  DIGIT             DIGIT  DIGIT
             9      9                 9      9

NUMBER:
[ cat:   NUMBER
  label:   [ 1:        [ 1:     Enniakosies
                          2:        [ 1:     Eneninta
                                      2:     Ennia ] ]
             2:     Xiliades
             3:        [ 1:     Enniakosia
                          2:        [ 1:     Eneninta
                                      2:     Ennia ] ] ] ]

1 parse found
```

Ακολουθεί ο κώδικα από τα αρχεία .lex και .grm

Ex3.lex

```
\w 0
\c DIGIT
\f
   <count> = 1
   <iszero> = true

\w 0
\c DIGIT
\f
   <count count> = 1
   <iszero> = true

\w 0
\c DIGIT
\f
   <count count count> = 1
   <iszero> = true

\w 1
\c DIGIT
\f
   <count> = 1
   <label> = Ena
   <genos> = neut
   <isone> = true

\w 1
\c DIGIT
\f
   <count> = 1
   <label> = Mia
   <genos> = fem
   <isone> = true

\w 1
\c DIGIT
\f
   <count count> = 1
   <label> = Deka
   <isone> = true

\w 1
\c DIGIT
\f
   <count count count> = 1
   <label> = Ekaton
   <isone> = true

\w 2
\c DIGIT
\f
   <count> = 1
   <label> = Duo
```

<istwo> = true

\w 2
\c DIGIT
\f
<count count> = 1
<label> = Eikosi
<istwo> = true

\w 2
\c DIGIT
\f
<count count count> = 1
<label> = Diakosia
<istwo> = true
<genos> = neut

\w 2
\c DIGIT
\f
<count count count> = 1
<label> = Diakosies
<istwo> = true
<genos> = fem

\w 3
\c DIGIT
\f
<count> = 1
<label> = Tria
<genos> = neut

\w 3
\c DIGIT
\f
<count> = 1
<label> = Treis
<genos> = fem

\w 3
\c DIGIT
\f
<count count> = 1
<label> = Trianta

\w 3
\c DIGIT
\f
<count count count> = 1
<label> = Triakosia
<genos> = neut

\w 3
\c DIGIT
\f
<count count count> = 1

<label> = Triakosies
<genos> = fem

\w 4
\c DIGIT
\f
<count> = 1
<label> = Tessera
<genos> = neut

\w 4
\c DIGIT
\f
<count> = 1
<label> = Tessereis
<genos> = fem

\w 4
\c DIGIT
\f
<count count> = 1
<label> = Saranta

\w 4
\c DIGIT
\f
<count count count> = 1
<label> = Tetrakosia
<genos> = neut

\w 4
\c DIGIT
\f
<count count count> = 1
<label> = Tetrakosies
<genos> = fem

\w 5
\c DIGIT
\f
<count> = 1
<label> = Pente

\w 5
\c DIGIT
\f
<count count> = 1
<label> = Peninta

\w 5
\c DIGIT
\f
<count count count> = 1
<label> = Pentakosia
<genos> = neut

\w 5
\c DIGIT
\f
  <count count count> = 1
  <label> = Pentakosies
  <genos> = fem

\w 6
\c DIGIT
\f
  <count> = 1
  <label> = Eksi

\w 6
\c DIGIT
\f
  <count count> = 1
  <label> = Eksinta

\w 6
\c DIGIT
\f
  <count count count> = 1
  <label> = Eksakosia
  <genos> = neut

\w 6
\c DIGIT
\f
  <count count count> = 1
  <label> = Eksakosies
  <genos> = fem

\w 7
\c DIGIT
\f
  <count> = 1
  <label> = Epta

\w 7
\c DIGIT
\f
  <count count> = 1
  <label> = Ebdominta

\w 7
\c DIGIT
\f
  <count count count> = 1
  <label> = Eptakosia
  <genos> = neut

\w 7
\c DIGIT
\f
  <count count count> = 1

<label> = Eptakosies
<genos> = fem

\w 8
\c DIGIT
\f
<count> = 1
<label> = Oxtw

\w 8
\c DIGIT
\f
<count count> = 1
<label> = Ogdonta

\w 8
\c DIGIT
\f
<count count count> = 1
<label> = Oktakosia
<genos> = neut

\w 8
\c DIGIT
\f
<count count count> = 1
<label> = Oktakosies
<genos> = fem

\w 9
\c DIGIT
\f
<count> = 1
<label> = Ennia

\w 9
\c DIGIT
\f
<count count> = 1
<label> = Eneninta

\w 9
\c DIGIT
\f
<count count count> = 1
<label> = Enniakosia
<genos> = neut

\w 9
\c DIGIT
\f
<count count count> = 1
<label> = Enniakosies
<genos> = fem

Ex3.grm

```
;-------------------------------------------------------------------------------------------
RULE ; {(tr1, tr2), tr1>1 }
NUMBER -> TRIPLET_1 TRIPLET_2
<TRIPLET_1 genos> = fem
<TRIPLET_1 iszero> = false
<TRIPLET_1 isone>  = false
<TRIPLET_2 genos> = neut
<NUMBER label 1> = <TRIPLET_1 label>
<NUMBER label 2> = Xiliades
<NUMBER label 3> = <TRIPLET_2 label>

RULE ; {(tr1, tr2), tr1=1 }
NUMBER -> TRIPLET_1 TRIPLET_2
<TRIPLET_1 genos> = fem
<TRIPLET_1 label> = Mia
<TRIPLET_1 iszero> = false
<TRIPLET_2 genos> = neut
<NUMBER label 1> = Xilia
<NUMBER label 2> = <TRIPLET_2 label>

RULE ; {(tr1, tr2), tr1=0, tr2<>0 }
NUMBER -> TRIPLET_1 TRIPLET_2
<TRIPLET_1 genos> = fem
<TRIPLET_1 label> = Miden
<TRIPLET_2 genos> = neut
<TRIPLET_2 iszero> = false
<NUMBER label> = <TRIPLET_2 label>

RULE ; {(tr1, tr2), tr1=0, tr2=0 }
NUMBER -> TRIPLET_1 TRIPLET_2
<TRIPLET_1 genos> = fem
<TRIPLET_1 label> = Miden
<TRIPLET_2 genos> = neut
<TRIPLET_2 label> = Miden
<NUMBER label> = Miden

;-------------------------------------------------------------

RULE ; {(t, tr), t>1}
NUMBER -> TUPLE TRIPLET
<TUPLE genos> = fem
<TUPLE iszero> = false
<TUPLE isone> = false
<TRIPLET genos> = neut
<NUMBER label 1> = <TUPLE label>
<NUMBER label 2> = Xiliades
<NUMBER label 3> = <TRIPLET label>

RULE ; {(t, tr), t=1}
NUMBER -> TUPLE TRIPLET
<TUPLE genos> = fem
<TUPLE label> = Mia
<TRIPLET genos> = neut
```

<NUMBER label 1> = Xilia
<NUMBER label 2> = <TRIPLET label>

RULE ; {(t, tr), t=0, tr<>0 }
NUMBER -> TUPLE TRIPLET
<TUPLE genos> = fem
<TUPLE label> = Miden
<TRIPLET genos> = neut
<TRIPLET iszero> = false
<NUMBER label> = <TRIPLET label>

RULE ; {(t, tr), t=0, tr=0 }
NUMBER -> TUPLE TRIPLET
<TUPLE genos> = fem
<TUPLE label> = Miden
<TRIPLET genos> = neut
<TRIPLET label> = Miden
<NUMBER label> = Miden

;---

RULE ; {(d1,tr), d1>1}
NUMBER -> DIGIT TRIPLET
<DIGIT genos> = fem
<DIGIT count> = 1
<DIGIT isone> = false
<DIGIT iszero> = false
<TRIPLET genos> = neut
<NUMBER label 1> = <DIGIT label>
<NUMBER label 2> = Xiliades
<NUMBER label 3> = <TRIPLET label>

RULE ; {(d1,tr), d1=1}
NUMBER -> DIGIT TRIPLET
<DIGIT genos> = fem
<DIGIT count> = 1
<DIGIT lex> = 1
<TRIPLET genos> = neut
<NUMBER label 1> = Xilia
<NUMBER label 2> = <TRIPLET label>

RULE ; {(d1,tr), d1=0, tr<>0 }
NUMBER -> DIGIT TRIPLET
<DIGIT genos> = fem
<DIGIT lex> = 0
<DIGIT count> = 1
<TRIPLET iszero> = false
<TRIPLET genos> = neut
<NUMBER label> = <TRIPLET label>

RULE ; {(d1,tr), d1=0, tr=0 }
NUMBER -> DIGIT TRIPLET
<DIGIT genos> = fem
<DIGIT lex> = 0
<DIGIT count> = 1
<TRIPLET label> = Miden

<TRIPLET genos> = neut
<NUMBER label> = Miden

;---

RULE ; { tr | tr<>0 }
NUMBER -> TRIPLET
<TRIPLET genos> = neut
<TRIPLET iszero> = false
<NUMBER label> = <TRIPLET label>

RULE ; { tr | tr = 0 }
NUMBER -> TRIPLET
<TRIPLET genos> = neut
<TRIPLET label> = Miden
<NUMBER label> = Miden

RULE ;
NUMBER -> TUPLE
<TUPLE genos> = neut
<NUMBER label> = <TUPLE label>

RULE ; { (d), d<>0}
NUMBER -> DIGIT
<DIGIT iszero> = false
<DIGIT count> = 1
<DIGIT genos> = neut
<NUMBER label> = <DIGIT label>

RULE ; { (d), d=0}
NUMBER -> DIGIT
<DIGIT lex> = 0
<DIGIT count> = 1
<DIGIT genos> = neut
<NUMBER label> = Miden

;---
; TRIPLETS
;---

RULE ; {(d1, t), d1=0, t <> 0}
TRIPLET -> DIGIT TUPLE
<DIGIT genos> = <TUPLE genos>
<DIGIT count count count> = 1
<DIGIT lex> = 0
<TUPLE iszero> = false
<TRIPLET label> = <TUPLE label>
<TRIPLET genos> = <TUPLE genos>
;<TRIPLET iszero> = false
<TRIPLET isone> = <TUPLE isone>

RULE ; {(d1, t), d1<>0, t <> 0}
TRIPLET -> DIGIT TUPLE
<DIGIT genos> = <TUPLE genos>
<DIGIT count count count> = 1
<DIGIT iszero> = false

<TUPLE iszero> = false
<TRIPLET label 1> = <DIGIT label>
<TRIPLET label 2> = <TUPLE label>
<TRIPLET genos> = <TUPLE genos>
;<TRIPLET iszero> = false
;<TRIPLET isone> = false

RULE ; {(d1,t), d1 > 1, t = 0}
TRIPLET -> DIGIT TUPLE
<DIGIT genos> = <TUPLE genos>
<DIGIT count count count> = 1
<DIGIT isone> = false
<DIGIT iszero> = false
<TUPLE label> = Miden
<TRIPLET label> = <DIGIT label>
<TRIPLET genos> = <TUPLE genos>
;<TRIPLET iszero> = false
;<TRIPLET isone> = false

RULE ; {(d1,t), d1 = 0, t = 0}
TRIPLET -> DIGIT TUPLE
<DIGIT genos> = <TUPLE genos>
<DIGIT count count count> = 1
<DIGIT lex> = 0
<TUPLE label> = Miden
;<TRIPLET label> = Miden
<TRIPLET genos> = <TUPLE genos>
<TRIPLET iszero> = true
;<TRIPLET isone> = false

RULE ; rule for 100
TRIPLET -> DIGIT TUPLE
<DIGIT genos> = <TUPLE genos>
<DIGIT count count count> = 1
<DIGIT lex> = 1
<TUPLE label> = Miden
<TRIPLET label> = Ekato

;---

RULE ; {(d1, d2), d1>1}
TUPLE -> DIGIT_1 DIGIT_2
<TUPLE count count> = 1
<DIGIT_1 count count> = 1
<DIGIT_2 count> = 1
<DIGIT_1 iszero> = false
<DIGIT_1 isone> = false
<DIGIT_1 genos> = <DIGIT_2 genos>
<TUPLE genos> = <DIGIT_2 genos>
<TUPLE label 1> = <DIGIT_1 label>
<TUPLE label 2> = <DIGIT_2 label>
;<TUPLE iszero> = false
;<TUPLE isone> = false

RULE ; {(0, d2), d2<>0}
TUPLE -> DIGIT_1 DIGIT_2

<TUPLE count count> = 1
<DIGIT_1 count count> = 1
<DIGIT_2 count> =1
<DIGIT_1 lex> = 0
<DIGIT_2 iszero> = false
<TUPLE genos> = <DIGIT_2 genos>
<TUPLE label> = <DIGIT_2 label>
;<TUPLE iszero> = false
<TUPLE isone> = <DIGIT_2 isone>;????????????????????????????

RULE ; {(1, d2), d2 > 2 | d2 = 0}
TUPLE -> DIGIT_1 DIGIT_2
<TUPLE count count> = 1
<DIGIT_1 count count> = 1
<DIGIT_2 count> = 1
<DIGIT_1 lex> = 1
<DIGIT_2 isone> = false
<DIGIT_2 istwo> = false
<TUPLE genos> = <DIGIT_2 genos>
<TUPLE label 1> = <DIGIT_1 label>
<TUPLE label 2> = <DIGIT_2 label>
;<TUPLE iszero> = false
;<TUPLE isone> = false

RULE
TUPLE -> TUPLES
<TUPLE label> = <TUPLES label>
<TUPLE iszero> = <TUPLES iszero>
<TUPLE isone> = <TUPLES isone>

RULE ; rule for 00
TUPLES -> DIGIT_1 DIGIT_2
<TUPLES label> = Miden
<TUPLES count count> = 1
<DIGIT_1 count count> = 1
<DIGIT_2 count> = 1
<DIGIT_1 lex> = 0
<DIGIT_2 lex> = 0
<TUPLES iszero> = true
;<TUPLES isone> = false

RULE ; rule for 11
TUPLES -> DIGIT_1 DIGIT_2
<TUPLE count count> = 1
<DIGIT_1 count count> = 1
<DIGIT_2 count>=1
<DIGIT_2 genos> = neut
<DIGIT_1 lex> = 1
<DIGIT_2 lex> = 1
<TUPLES label> = Enteka
;<TUPLES iszero> = false
;<TUPLES isone> = false

RULE ; rule for 12
TUPLES -> DIGIT_1 DIGIT_2
<TUPLE count count> = 1

<DIGIT_1 count count> = 1
<DIGIT_2 count>=1
<DIGIT_1 lex> = 1
<DIGIT_2 lex> = 2
<TUPLES label> = Dwdeka
;<TUPLES iszero> = false
;<TUPLES isone> = false

β. Για το αντίστροφο έχουμε υλοποιήσει μια γραμματική σε PCPATR που να δέχεται αριθμούς το πολύ τριψήφιους ολογράφως και τους μετατρέπει στη αριθμητική μορφή. Επειδή όμως, όπως ξέρουμε, το PCPATR δεν μπορεί να κάνει προσθέσεις ή άλλες αριθμητικές πράξεις, πρέπει να κάνουμε κάποια σύμβαση. Η σύμβαση που κάνουμε είναι ότι το αποτέλεσμα προκύπτει ως το άθροισμα από τις τιμές της ιδιότητας label της ρίζας. Αυτό γίνεται κατανοητό με το ακόλουθο παράδειγμα :

```
Eksakosia Ebdominta Ena

1:
             S
          ___|_____
  NUMBER           S
 Eksakosia      ___|____
             NUMBER     S
            Ebdominta   |
                      NUMBER
                        Ena

S:
[ cat:   S
  count:   [ count:   [ count: 1 ] ]
  label:   [ 1:     600
             2:       [ 1:     70
                        2:     1 ] ] ]

1 parse found
```

Η τιμή είναι 600+70+1 = 671.

Βλέποντας τα αρχεία του λεξικού και της γραμματικής που ακολουθούν διαπιστώνουμε ότι το προηγούμενο ερώτημα ήταν σημαντικά πιο δύσκολο ενώ το παρόν ήταν απρόσμενα εύκολο, αν και εκ πρώτης όψης ίσως να έμοιαζε πιο δύσκολο.

Ακολουθούν τα αρχεία του λεξικού (.lex) και της γραμματικής (.grm) :

Ex3b.grm

```
RULE
S -> NUMBER
<S count> = <NUMBER count>
<S label> = <NUMBER label>


RULE
S_1 -> NUMBER S_2
<S_1 count count> = <S_2 count>
<NUMBER count count> = <S_2 count>
<S_1 label 1> = <NUMBER label>
<S_1 label 2> = <S_2 label>
```

RULE
S_1 -> NUMBER S_2
<S_1 count count count> = <S_2 count>
<NUMBER count count count> = <S_2 count>
<S_1 label 1> = <NUMBER label>
<S_1 label 2> = <S_2 label>

Ex3b.lex

```
\w 0
\c NUMBER
\f
   <count> = 1
   <label> = -

\w 0
\c NUMBER
\f
   <count count> = 1
   <label> = -

\w 0
\c NUMBER
\f
   <count count count> = 1
   <label> = -

\w Ena
\c NUMBER
\f
   <count> = 1
   <label> = 1

\w Deka
\c NUMBER
\f
   <count count> = 1
   <label> = 10

\w Enteka
\c NUMBER
\f
   <count count> = 1
   <label> = 11

\w Dwdeka
\c NUMBER
\f
   <count count> = 1
   <label> = 12

\w Dekatria
\c NUMBER
\f
```

<count count> = 1
<label> = 13

\w Dekatessera
\c NUMBER
\f
<count count> = 1
<label> = 14

\w Dekapente
\c NUMBER
\f
<count count> = 1
<label> = 15

\w Dekaeksi
\c NUMBER
\f
<count count> = 1
<label> = 16

\w Dekaepta
\c NUMBER
\f
<count count> = 1
<label> = 17

\w Dekaoktw
\c NUMBER
\f
<count count> = 1
<label> = 18

\w Dekaennia
\c NUMBER
\f
<count count> = 1
<label> = 19

\w Ekaton
\c NUMBER
\f
<count count count> = 1
<label> = 100

\w Ekato
\c NUMBER
\f
<count count count> = 1
<label> = 100

\w Duo
\c NUMBER
\f
<count> = 1
<label> = 2

\w Eikosi
\c NUMBER
\f
  <count count> = 1
  <label> = 20

\w Diakosia
\c NUMBER
\f
  <count count count> = 1
  <label> = 200

\w Tria
\c NUMBER
\f
  <count> = 1
  <label> = 3

\w Trianta
\c NUMBER
\f
  <count count> = 1
  <label> = 30

\w Triakosia
\c NUMBER
\f
  <count count count> = 1
  <label> = 300

\w Tessera
\c NUMBER
\f
  <count> = 1
  <label> = 4

\w Saranta
\c NUMBER
\f
  <count count> = 1
  <label> = 40

\w Tetrakosia
\c NUMBER
\f
  <count count count> = 1
  <label> = 400

\w Pente
\c NUMBER
\f
  <count> = 1
  <label> = 5

\w Peninta

\c NUMBER
\f
  <count count> = 1
  <label> = 50

\w Pentakosia
\c NUMBER
\f
  <count count count> = 1
  <label> = 500

\w Eksi
\c NUMBER
\f
  <count> = 1
  <label> = 6

\w Eksinta
\c NUMBER
\f
  <count count> = 1
  <label> = 60

\w Eksakosia
\c NUMBER
\f
  <count count count> = 1
  <label> = 600

\w Epta
\c NUMBER
\f
  <count> = 1
  <label> = 7

\w Ebdominta
\c NUMBER
\f
  <count count> = 1
  <label> = 70

\w Eptakosia
\c NUMBER
\f
  <count count count> = 1
  <label> = 700

\w Oxtw
\c NUMBER
\f
  <count> = 1
  <label> = 8

\w Ogdonta
\c NUMBER
\f

```
   <count count> = 1
   <label> = 80

\w Oktakosia
\c NUMBER
\f
   <count count count> = 1
   <label> = 800

\w Ennia
\c NUMBER
\f
   <count> = 1
   <label> = 9

\w Eneninta
\c NUMBER
\f
   <count count> = 1
   <label> = 90

\w Enniakosia
\c NUMBER
\f
   <count count count> = 1
   <label> = 900
```

Παρατηρούμε ότι ενώ η γραμματική είναι στοιχειώδης, το λεξικό είναι αρκετά πιο μεγάλο.