

---

## ΑΣΚΗΣΗ ΠΡΑΞΗΣ 5: ΕΦΑΡΜΟΓΗ: ΟΡΘΟΓΡΑΦΟΣ

---

Μια απλή αλλά ιδιαίτερα χρήσιμη εφαρμογή είναι η διόρθωση ορθογραφικών λαθών μέσα σε ένα κείμενο. Στην Prolog η εφαρμογή αυτή υλοποιείται πολύ εύκολα με τη χρήση του κατηγορήματος `append`, και του γεγονότος ότι ο **μηχανισμός ενοποίησης (unification)** δίνει τη δυνατότητα να καθορίζονται τα ορίσματα των κατηγορημάτων άλλοτε ως εισόδου και άλλοτε ως εξόδου. Έτσι στη συγκεκριμένη εφαρμογή το `append` χρησιμοποιείται είτε για να συνενώσει δύο λίστες σε μία τρίτη είτε να χωρίσει μία λίστα σε δύο επι μέρους.

Για απλούστευση, θα θεωρήσουμε ότι οι λέξεις βρίσκονται σε μορφή λίστας από γράμματα. Βασικό συστατικό ενός ορθογράφου είναι η ύπαρξη κάποιου λεξικού, δηλαδή κάποιας βάσης δεδομένων που περιέχει τις αποδεκτές λέξεις. Για παράδειγμα:

```
word([h,e,l,l,o]).  
word([m,a,n]).  
word([w,o,m,a,n]).  
word([b,o,o,k]).  
word([m,o,t,h,e,r]).  
word([c,o,m,p,u,t,e,r]).  
word([c,a,r]).  
word([p,r,o,l,o,g]).  
word([p,e,n,c,i,l]).  
word([t,a,b,l,e]).
```

Τα παραπάνω γεγονότα δηλώνουν τη λίστα των αποδεκτών λέξεων. Φυσικά σε ένα πραγματικό ορθογράφο η παραπάνω λίστα είναι πολύ μεγαλύτερη και πληρέστερη, καλύπτοντας – αν αυτό είναι δυνατό – όλες τις λέξεις της γλώσσας που μας ενδιαφέρει. (Η σειρά με την οποία δηλώνονται οι λέξεις δεν έχει φυσικά στην Prolog καμία σημασία).

Μια απλή ορθογραφική διόρθωση είναι η ανίχνευση μιας λέξης που γράφτηκε με ένα γράμμα λάθος. Για παράδειγμα θα μπορούσαμε να ορίσουμε το κατηγορήμα

```
changeletter(Word, CorrectWord)
```

το οποίο δέχεται σαν πρώτο όρισμα μια λέξη γραμμένη με ένα γράμμα λάθος και επιστρέφει στο δεύτερο όρισμα τη λέξη (ή τις λέξεις) του λεξικού που ταιριάζουν μ' αυτήν αν αλλάξουμε ένα μόνο γράμμα. Για παράδειγμα:

```
?- changeletter([m,o,n],X).  
X = [m,a,n];  
false
```

Οι λέξεις "mon" και "man" διαφέρουν σε ένα μόνο γράμμα, και η λέξη "man" είναι μια λέξη που βρίσκεται στο λεξικό. Το κατηγορήμα `changeletter` υλοποιείται ως εξής:

```
changeletter(Word, CorrectWord) :-  
  append(Begin, [Letter|End], Word),  
  word(CorrectWord),  
  append(Begin, [Letter1|End], CorrectWord),  
  Letter1 \= Letter.
```

Η κλήση του κατηγορήματος `word(CorrectWord)` απαιτεί η λίστα `CorrectWord` να ανήκει στη λίστα των αποδεκτών λέξεων του λεξικού.

Οι δύο κλήσεις του κατηγορήματος `append` πριν και μετά την κλήση της `word(CorrectWord)` τεμαχίζουν τις λίστες `Word` και `CorrectWord` σε δύο τμήματα αντίστοιχα.

Πράγματι η `append(Begin, [Letter|End], Word)` τεμαχίζει την λίστα `Word` σε δύο τμήματα, τις λίστες `Begin` και `[Letter|End]`. Η δεύτερη λίστα αποτελείται από την κεφαλή της που είναι

το γράμμα `Letter` και την ουρά που είναι η λίστα `End`. Άρα συνολικά το `Word` έχει τεμαχιστεί σε τρία κομμάτια δηλαδή τα `Begin` (λίστα), `Letter` (ένα στοιχείο/γράμμα) και `End` (λίστα).

Αντίστοιχα η `append(Begin, [Letter1|End], CorrectWord)` τεμαχίζει τη λίστα `CorrectWord` σε δύο τμήματα, τις λίστες: `Begin` και `[Letter1|End]` και ουσιαστικά στα τρία κομμάτια `Begin`, `Letter1` και `End`.

Αυτό σημαίνει ότι οι δύο λίστες `Word` και `CorrectWord` έχουν την ίδια αρχή και το ίδιο τέλος (`Begin` και `End`) αλλά διαφέρουν σε κάποιο γράμμα στη μέση και συγκεκριμένα το γράμμα `Letter` το οποίο διαφέρει από το γράμμα `Letter1` (Το `Letter` δεν ενοποιείται με το `Letter1`).

Παρατηρήστε ότι με βάση τη λογική του `append` τόσο η λίστα `Begin` όσο και η λίστα `End` μπορεί να είναι κενές. Στην πρώτη περίπτωση το λάθος εντοπίζεται στο πρώτο γράμμα, ενώ στη δεύτερη λάθος είναι το τελευταίο γράμμα.

## ΤΙ ΠΡΕΠΕΙ ΝΑ ΚΑΝΕΤΕ

**1.** Χρησιμοποιήστε το κατηγορήμα `append` για να υλοποιήσετε τα παρακάτω κατηγορήματα τεμαχίζοντας τις λέξεις σε κατάλληλο αριθμό κομματιών. Τα κατηγορήματα αυτά μπορούν να θεωρηθούν ότι ορίζουν κανόνες εμπειρίας για τη διόρθωση λαθών.

(α) **swapletters (Word, CorrectWord)**. Το κατηγορήμα αυτό δέχεται ως είσοδο τη λίστα `Word` και επιστρέφει στο δεύτερο όρισμα μια λέξη του λεξικού που είναι ίδια με την `Word` με εξαίρεση το γεγονός ότι έχει γίνει ανταλλαγή δύο γειτονικών γραμμάτων. Παράδειγμα:

```
?- swapletters([w,m,o,a,n],X).
X = [w,o,m,a,n];           %ανταλλαγή των γραμμάτων o, m
false
```

(β) **exchangeletters (Word, CorrectWord)**. Το κατηγορήμα αυτό δέχεται ως είσοδο τη λίστα `Word` και επιστρέφει στο δεύτερο όρισμα μια λέξη του λεξικού που είναι ίδια με την `Word` με εξαίρεση το γεγονός ότι έχει γίνει ανταλλαγή δύο γραμμάτων που δεν είναι κατ' ανάγκη γειτονικά. Παράδειγμα:

```
?- exchangeletters([w,a,m,o,n],X).
X = [w,o,m,a,n];           %ανταλλαγή των γραμμάτων o, a
false
```

(γ) **delete2 (W, CW)**. Το κατηγορήμα αυτό θα δέχεται ως είσοδο τη λίστα `W` που αντιστοιχεί σε μία πιθανά ανορθόγραφη λέξη και επιστρέφει στη λίστα `CW` μια σωστά ορθογραφημένη λέξη η οποία προήλθε μετά την αφαίρεση δύο γραμμάτων από τη λέξη `W` (τα γράμματα μπορούν να αφαιρεθούν από οποιαδήποτε σημεία της λέξης που δεν είναι αναγκαστικά γειτονικά). Παράδειγμα:

```
?- delete2([m,o,o,t,h,e,r,w],L).
L = [m, o, t, h, e, r]
?- delete2([w,i,o,m,e,a,n],L).
L = [w, o, m, a, n]
```

(δ) **movefront (W, CW)**. Το κατηγορήμα αυτό θα δέχεται ως είσοδο τη λίστα `W` που αντιστοιχεί σε μία πιθανά ανορθόγραφη λέξη και επιστρέφει στη λίστα `CW` μια σωστά ορθογραφημένη λέξη η οποία προήλθε από τη μετακίνηση προς τα εμπρός ενός γράμματος της λέξης `W`. Παράδειγμα:

```
?- movefront([m,t,h,e,o,,r],L).
L = [m, o, t, h, e, r]
?- movefront([o,m,w,a,n],L).
L = [w, o, m, a, n]
```

**2.** Υλοποιήστε ένα κατηγορήμα το οποίο θα δέχεται μία πρόταση με τη μορφή λίστας από λέξεις και θα τυπώνει στην οθόνη τα λάθη (αν υπάρχουν) σε κάποιες από αυτές. Για παράδειγμα:

```
?- spell_check([[t,h,e], [m,o,n], [a,n,d], [e,h,t], [w,m,o,a,n]]).
```

```
Spelling Error (Mistyping): [m,o,n] Suggest [m,a,n]
```

```
Unknown: [a,n,d]
```

```
Spelling Error (Exchange): [e,h,t] Suggest [t,h,e]
```

```
Spelling Error (Swap): [w,m,o,a,n] Suggest [w,o,m,a,n]
```

```
true
```