

Αρχές Γλωσσών Προγραμματισμού

3η Εργασία

Αστέριος Γιαννούδης Α.Μ.: 1115201200025

Νικόλαος Θεοδώρου Α.Μ.: 1115201000030

Η άσκηση έχει υλοποιηθεί πλήρως με βάση τα ζητούμενα της εκφώνησης, στο αρχείο `skeleton.hs` το οποίο βρίσκετε στον φάκελο με τα παραδοτέα αρχεία.

Έχουμε υλοποιήσει το β και η reduction με τις συναρτήσεις `beta` και `eta` αντίστοιχα.

Η `reduce` υπολογίζει την κανονική μορφή ενός λ-όρου εφαρμόζοντας πρώτα τα απαιτούμενα β reductions και στη συνέχεια αν γίνεται εφαρμόζει η reduction.

Δεν έχουμε φτιάξει σε μία συνάρτηση το α reduction, αλλά παρ' όλα αυτά σε περίπτωση που χρειαστεί μπορούμε να αλλάξουμε χειροκίνητα εμείς τις μεταβλητές και ο υπολογισμός της κανονικής μορφής να συνεχιστεί κανονικά.

Αυτό γίνεται εμφανώς άμα θέλετε να τρέξετε το παράδειγμα `not true = false`, όπου κάποια στιγμή θα χρειαστεί να γίνει ένα α reduction χειροκίνητα και να δοθεί σαν νέο όρισμα στην συνάρτηση `reduce`.

Για παράδειγμα εάν σαν είσοδο στην `reduce` δοθεί το εξής :

```
reduce (myparse "(\\z.z(\\x.\\y.y)(\\x.\\y.x))(\\x.\\y.x)")
```

θα βγει λάθος κανονική μορφή. Παρατηρώντας όμως την λίστα με τους λ-όρους μετά από κάθε reduction βλέπουμε πως οι δύο πρώτες είναι σωστές. Έτσι λοιπόν, αν αλλάξουμε μεταβλητή μόνοι μας στο αποτέλεσμα αυτό :

```
reduce (myparse "(\\a.\\x.\\y.y)(\\x.\\y.x)")
```

θα έχουμε τη σωστή κανονική μορφή.

Σε άλλες περιπτώσεις γίνεται σωστά όπως σε αυτή:

```
reduce (myparse "(\\x.\\y.zxy)(wy)")
```