Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Γλωσσες Προγραμματισμου 2
Ασκηση 4 Συστηματα τυπων

Γεώργιος Καράκος

Αριθμός Μητρώου: 03113204

Μέρος 1ο - Αυτοεφαρμογή

Έχουμε σύστημα τύπων με απλές συναρτήσεις όπως περιγράφεται στη σελίδα 14 των διαφανειών. Προσπαθούμε να αποδείξουμε αν μπορεί να υπάρχει κάποιο περιβάλλον Γ και κάποιος τύπος τ ώστε να ισχύει:

$$\Gamma| - x\ x : \tau$$

Ακολουθούμε τη μέθοδο της εις άτοπο απαγωγής.

Αν υποθέσουμε ότι υπάρχει περιβάλλον Γ για το οποίο να ισχύει: $\Gamma| - x\ x : \tau$ τότε πρέπει να ισχύουν τα ακόλουθα: $\Gamma| - x : \tau' \rightarrow \tau$ και $\Gamma| - x : \tau'$,όπου τ′ ένας τύπος. Σύμφωνα με τα παραπάνω η έκφραση x έχει τύπους $\tau \rightarrow \tau'$ και τ′. Για να ικανοποιούνται και οι δύο σχέσεις, οι τύποι αυτοί πρέπει να είναι δομικά ίδιοι. Παρατηρούμε όμως ότι αυτό δεν είναι δυνατόν, καθώς ο τύπος $\tau' \rightarrow \tau$ υποδηλώνει συνάρτηση που παίρνει είσοδο ένα δεδομένο τύπου τ′ και επιστρέφει τιμή τύπου τ, οπότε δε μπορεί να είναι ίδιος με τον τύπο τ′ στο σύστημα τύπων που έχουμε διαθέσιμο.

Μέρος 2ο - Αναφορές και αναδρομή

1.

Πρέπει να κατασκευάσουμε ένα πρόγραμμα το οποίο να μην τερματίζει. Γράφτηκε χρησιμοποιώντας το syntactic sugar let , καθώς γνωρίζουμε ότι είναι ισοδύναμο με λάμδα έκφραση.

```
let r = ref (λn: Int.n)
   in let f = Int → Int.(λn. !r n)
         in (r:=f; f 1)
```

Το παραπάνω πρόγραμμα δε θα τερματίσει ποτέ καθώς η κλήση της συνάρτησης f με παράμετρο 1, θα καλέσει την συνάρτηση την οποία κάνει reference η r με παράμετρο πάλι 1. Εφόσον η συνάρτηση που κάνει reference η r είναι η f, αυτό σημαίνει πως θα καλείται συνεχώς η f με το ίδιο όρισμα, οπότε το πρόγραμμα δεν τερματίζει.

2.

Στην ίδια γλώσσα γράφουμε τη συνάρτηση που να υπολογίζει το παραγοντικό ενός φυσικού αριθμού.

```
let r = ref (λn: Int.n)
   in let f = Int → Int.(λn. if n<=1 then 1 else n*(!r (n-1)))
         in (r:=f; f)
```

Η r τώρα μπορεί να καλεστεί με όρισμα έναν φυσικό αριθμό και επιστρέφει το παραγοντικό αυτού του αριθμού.