



Γλώσσες Προγραμματισμού II

Αν δεν αναφέρεται διαφορετικά, οι ασκήσεις πρέπει να παραδίδονται στους διδάσκοντες σε ηλεκτρονική μορφή μέσω του συνεργατικού συστήματος ηλεκτρονικής μάθησης `moodle.softlab.ntua.gr`. Η προθεσμία παράδοσης θα τηρείται αυστηρά. Έχετε δικαίωμα να καθυστερήσετε το πολύ μία άσκηση.

Άσκηση 6 Συστήματα τύπων — `letrec`

Προθεσμία παράδοσης: 10/1/2021

Στις περισσότερες γλώσσες προγραμματισμού επιτρέπεται ο ορισμός *αμοιβαία αναδρομικών* συναρτήσεων, όπως για παράδειγμα οι παρακάτω (σε OCaml):

```
let rec even n = if n == 0 then true  else odd  (n-1)
and   odd  n = if n == 0 then false else even (n-1)
```

Προσθέστε τη δυνατότητα ορισμού αμοιβαία αναδρομικών συναρτήσεων στη δομή `letrec` που περιγράφεται στη διαφάνεια 31 της παράδοσης της 25/11/2020. Συγκεκριμένα:

1. Προσθέστε στη σύνταξη της γλώσσας τον παρακάτω κανόνα (όπου $n \geq 1$):

$$e ::= \dots \mid \mathbf{letrec} \ x_1 = e_1 \ \dots \ x_n = e_n \ \mathbf{in} \ e$$

2. Ορίστε κατάλληλο κανόνα τύπων, γενικεύοντας τον κανόνα της διαφάνειας 31.
3. Ορίστε κατάλληλους κανόνες λειτουργικής σημασιολογίας, call-by-value και call-by-name. Βασιστείτε στους αντίστοιχους κανόνες για το **fix** (διαφάνειες 26 και 27).
4. Περιγράψτε πώς η γενικευμένη δομή **letrec** μπορεί να υλοποιηθεί ως *παραγόμενη μορφή* (syntactic sugar) βάσει του **fix**. Μπορείτε να υποθέσετε ότι η γλώσσα σας διαθέτει tuples, records, ή οποιονδήποτε άλλο τύπο δεδομένων χρειάζεστε από αυτούς των διαφανειών 1–31.
5. Εξηγήστε γιατί η υλοποίηση ως παραγόμενη μορφή που προτείνετε στο ερώτημα 3 έχει την ίδια σημασιολογία με τις απαντήσεις σας στα ερωτήματα 2 (κανόνας τύπων) και 3 (λειτουργική σημασιολογία).

Δώστε ιδιαίτερη προσοχή στην ακρίβεια των τυπικών ορισμών σας. Αν χρησιμοποιήσετε σχετικές εργασίες τρίτων που βρήκατε π.χ. στο διαδίκτυο, μην ξεχάσετε να αναφέρετε λεπτομερώς τις πηγές σας.