



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 Δηλωτική σημασιολογία

Προθεσμία παράδοσης: 13/7/2014

Να ορίσετε μία δηλωτική σημασιολογία για τη γλώσσα της μηχανής στοίβας που περιγράφεται στην άσκηση 5. Στη συνέχεια, να συμπληρώσετε το πρόγραμμα Haskell που δίνεται από τους διδάσκοντες (στη σελίδα του μαθήματος) για να υλοποιήσετε ένα διερμηνέα για αυτή τη γλώσσα, μεταφράζοντας πιστά σε κώδικα τους κανόνες της δηλωτικής σημασιολογίας.

Η σύνταξη της γλώσσας υλοποιείται από τον τύπο δεδομένων `P`:

```
data P = Pint Integer | Ptrue | Pfalse | Padd | Pneg | Pmul | Pdiv | Plt | Peq | Pand
       | Pnot | Pnop | Pdup | Ppop | Pswap | Pswap2 | Pseq P P | Pcond P P | Ploop P
```

για τον οποίο έχουν ήδη υλοποιηθεί ένας parser και ένας pretty printer (δηλαδή, instances των κλάσεων `Read` και `Show`).

Για τη δηλωτική σημασιολογία απαιτούνται δύο σημασιολογικά πεδία: οι τιμές που αποθηκεύονται στη στοίβα (ακέραιες ή λογικές) και η ίδια η στοίβα. Υλοποιούνται αντίστοιχα από τους τύπους δεδομένων `V` και `S`.

```
data V = VI Integer | VB Bool
type S = [V]
```

Η σημασιολογική συνάρτηση υλοποιείται από τη συνάρτηση `sem` την οποία καλείστε να συμπληρώσετε. Ο τύπος της και η εσφαλμένη υλοποίηση των διδασκόντων δίνεται παρακάτω:

```
sem :: P -> S -> S
sem p s = [VI 42]
```

Ο τελεστής σταθερού σημείου `fix` (που θα σας χρειαστεί για να ορίσετε τη σημασιολογία του `Ploop`) μπορεί να υλοποιηθεί εύκολα σε Haskell ως εξής:

```
fix :: (a -> a) -> a
fix f = f (fix f)
```

Όταν συμπληρώσετε τη συνάρτηση `sem`, ο διερμηνέας σας θα διαβάζει το πρόγραμμα από την τυπική είσοδο (`stdin`) και θα εκτυπώνει (`stdout`) τα περιεχόμενα της στοίβας μετά την εκτέλεση. Για παράδειγμα:

```
$ cat ex.in
1 3 true loop [dup 2 < cond [false | dup 1 - + swap2 * swap true]] pop dup 1 + *
$ ./densem < ex.in
42
```

(Μήπως τελικά η συνάρτηση `sem` των διδασκόντων λειτουργεί σωστά;)