



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 8 Δηλωτική σημασιολογία

Προθεσμία παράδοσης: 3/2/2013

Υλοποιήστε δυο διερμηνείς για τη γραμμική γραφή M, μεταφράζοντας πιστά σε κώδικα τους κανόνες δύο παραλλαγών της δηλωτικής σημασιολογίας της: μίας για αποτίμηση call-by-value και μίας για αποτίμηση call-by-name. Υποβάλετε τις λύσεις σας στο σύστημα αυτόματης υποβολής και ελέγχου προγραμμάτων grader.softlab.ntua.gr, σε όποια γλώσσα προγραμματισμού θέλετε (από αυτές που υποστηρίζει ο grader — αν θέλετε άλλη, μιλήστε με τους διδάσκοντες). Για δική σας διευκόλυνση, σας προτείνουμε να επιλέξετε μια συναρτησιακή γλώσσα: Haskell, Standard ML (MLton), OCaml ή F#.

Περιγραφή του προβλήματος. Πώς προέκυψε η φήμη ότι κατά το χειμερινό ηλιοστάσιο της 21/12/2012 ο κόσμος θα καταστρεφόταν; Γιατί η περίφημη συντέλεια του κόσμου δε συνέβη τελικά; Η νέα ταινία της γνωστής σειράς με τίτλο “Tomb Raider IV: Mayas Reinterpreted” απαντά σε όλα αυτά τα καντά ερωτήματα που μας ταλανίζουν. (Και ελπίζουμε ότι το επόμενο sequel θα απαντήσει και σε άλλα καντά ερωτήματα, όπως π.χ. ποιος πείραξε τη λίστα Lagarde και γιατί, ποιος πείραξε τη λίστα του Schindler και αν αληθεύει ότι αυτές οι δύο λίστες, πριν πειραχθούν, ήταν άπειρες και χρειάζονται Haskell για να τις επεξεργαστείς και ότι τελικά αυτός ήταν ο μόνος λόγος που τις πειράξαμε — όλοι είμαστε αθώοι!)

Η ταινία ξεκινά με ένα flashback. Ο πατέρας της γνωστής και προικισμένης (ε-χεμ) αρχαιολόγου Lara Croft είχε επισκεφθεί το θαμμένο ναό των Maya για πρώτη φορά πριν από πολλές δεκαετίες και, μαζί με τη γραμμική γραφή Λ που είδαμε περίσυ (αν δεν την έχετε δει κοιτάζτε την, πιθανώς θα βρείτε τη σημασιολογία της χρήσιμη), βρήκε άλλο ένα είδος γραφής στο ναό: τη γραμμική γραφή M! Συγκεκριμένα, η παρακάτω επιγραφή ήταν αυτή που δημιούργησε το φαινόμενο του 2012:

@/x@x/o/i+i+#0@x/x/x+o+x+o+i+i+oo/o-/i/x@@xoi
can you count to 4?

Φυσικά, το ερώτημα πώς προέκυψαν τα άπταιστα αγγλικά στην αρχαία επιγραφή των Maya διόλου δεν απασχόλησε το επιτελείο του Hollywood (και καλά θα κάνει να μη μας απασχολήσει κι εμάς). Η αρχαία περγαμινή που βρήκε ο πατέρας της Lara αποκαλύπτει σχετικά λίγες πληροφορίες για τη γραφή, μετά την αποκρυπτογράφησης του από τη γνωστή ομάδα επιστημόνων στο Princeton:

$e ::= \# c \mid + e e \mid - e \mid x \mid / x e \mid @ e e$	Dear Lara, where did you find this stuff again? It's a simple functional language with I/O: # is for constants, where $c \in X$ is a character; + and - are for output and input, respectively, see continuation-passing I/O [Andrew Gordon's thesis , §7.6]; $x \in Var$ is a letter standing for a variable; / and @ are abstraction and application in the λ -calculus. In my semantics, a state is a pair of the input remaining to be read and the output so far. Be careful: I'm assuming the language is eager! Glad I helped... — D.
$V = X + (V \rightarrow D)$	values
$D = S \rightarrow V \times S$	denotations
$S = X^* \times X^*$	states
$Env = Var \rightarrow V$	environments
$\llbracket e \rrbracket : Env \rightarrow D$	meanings

Η Lara δεν είναι καλή στις γλώσσες προγραμματισμού (όπως έχουμε πει, όμως, έχει άλλα χαρίσματα, γκουχ γκουχ). Κάποιος καλός της φίλος (εσείς!) τη βοηθάει να ορίσει μία δηλωτική σημασιολογία σε αποτίμηση call-by-value, ακολουθώντας τις συμβουλές του David και, βάσει αυτής, να κατασκευάσει μία υλοποίηση για τη γραμμική γραφή M. Έτσι προκύπτει η ημερομηνία 21/12/2012 (δοκιμάστε το!).

Όμως, όταν αυτή η ημερομηνία περνάει, η Lara παίρνει ένα νέο e-mail από τον David:

Dear Lara, the language wasn't eager, after all! Fix the semantics! Kisses... — D.

Βοηθήστε ξανά τη Lara να ορίσει και να υλοποιήσει μία δηλωτική σημασιολογία για την ίδια γλώσσα, σε αποτίμηση call-by-name. Πότε τελικά θα έρθει η συντέλεια του κόσμου; Προλαβαίνω;

Είσοδος και έξοδος. Κάθε ένα από τα δύο προγράμματά σας (ένα για τη σημασιολογία call-by-value και ένα για τη σημασιολογία call-by-name) θα διαβάζει τα δεδομένα από την τυπική είσοδο (stdin) και θα τυπώνει τα αποτελέσματα στην τυπική έξοδο (stdout).

Η πρώτη γραμμή της εισόδου θα περιέχει μία πρόταση στη γραμμική γραφή M. Υποθέστε ότι η πρόταση θα είναι έγκυρη. Η δεύτερη γραμμή της εισόδου θα περιέχει τους χαρακτήρες που θα δοθούν ως είσοδος για την αποτίμηση της πρότασης.

Το πρόγραμμά σας πρέπει να τυπώνει ακριβώς μία γραμμή που θα περιέχει τους χαρακτήρες που θα προκύψουν ως έξοδος κατά την αποτίμηση της πρότασης.

Παράδειγμα 1.

Είσοδος	Έξοδος (call-by-value)	Έξοδος (call-by-name)
-/x+x+#o+xx	1o1	1o1
1		

Εξήγηση: Κατά την αποτίμηση, διαβάζεται ένας χαρακτήρας από την είσοδο. Στη συνέχεια, ο χαρακτήρας αυτός τυπώνεται δύο φορές, μεταξύ των οποίων παρεμβάλλεται η εκτύπωση του χαρακτήρα “o”. Το αποτέλεσμα της αποτίμησης (το οποίο δε χρησιμοποιείται ούτε εμφανίζεται) είναι ο χαρακτήρας που διαβάστηκε. Το παράδειγμα αυτό έχει την ίδια συμπεριφορά και στις δύο σημασιολογίες (call-by-value και call-by-name).

Παράδειγμα 2.

Είσοδος	Έξοδος (call-by-value)	Έξοδος (call-by-name)
@/x+x+#o+x#i-/yy	1o1	1om
1m		

Σημείωση: Κατά την αποτίμηση call-by-value, ο δεύτερος χαρακτήρας της εισόδου “m” δε διαβάζεται.