Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών
https://courses.softlab.ntua.gr/pl2/

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 7 Δηλωτική σημασιολογία

Προθεσμία παράδοσης: 17/3/2019

Έστω η προστακτική γλώσσα προγραμματισμού με την εντολή `while`, που ορίστηκε στην παρουσίαση της 5/12/2018 (ας την ονομάσουμε WHILE). Η δηλωτική σημασιολογία της WHILE ορίζεται στις διαφάνειες, αρχικά χρησιμοποιώντας μερικές συναρτήσεις μεταξύ συνόλων και στη συνέχεια βασισμένη στη θεωρία πεδίων (διαφάνειες 29–31). Ένας διερμηνέας για τη WHILE γραμμένος σε Haskell και βασισμένος σε μία άμεση υλοποίηση της δηλωτικής σημασιολογίας είναι διαθέσιμος από τη σελίδα του μαθήματος. Για λόγους απλότητας, ο διερμηνέας αυτός υποθέτει ότι υπάρχουν μόνο ακέραιες μεταβλητές.

Στην άσκηση αυτή, ορίζουμε τη γλώσσα WHILE^{cons} που είναι η επέκταση της WHILE προσθέτοντας ζεύγη (cons cells), σαν αυτά της άσκησης 5. Επομένως, οι τιμές των εκφράσεων στη WHILE^{cons} είναι τριών ειδών:

- Ακέραιοι αριθμοί.
- Λογικές τιμές (true και false).
- Ζεύγη (cons cells) αποτελούμενα από δύο τιμές.

Επίσης, οι εκφράσεις παύουν να διαχωρίζονται βάσει του τύπου τους, δηλαδή η γλώσσα γίνεται επισήμως dynamically typed. Η αφηρημένη σύνταξη της WHILE^{cons} είναι η εξής:

$$
\begin{aligned}
C &::= \texttt{skip} \mid x \texttt{ := } E \mid C_0 \texttt{ ; } C_1 \mid \texttt{if } E \texttt{ then } C_0 \texttt{ else } C_1 \mid \texttt{for } E \texttt{ do } C \mid \texttt{while } E \texttt{ do } C \\
E &::= \texttt{0} \mid \texttt{succ } E \mid \texttt{pred } E \mid \texttt{true} \mid \texttt{false} \mid E_0 \texttt{ < } E_1 \mid E_0 \texttt{ = } E_1 \mid \texttt{not } E \mid \texttt{if } E_0 \texttt{ then } E_1 \texttt{ else } E_2 \\
&\quad \mid E_0 \texttt{ : } E_1 \mid \texttt{hd } E \mid \texttt{tl } E
\end{aligned}
$$

Οι τελεστές `:` (cons), `hd` και `tl` στη δεύτερη γραμμή της σύνταξης των εκφράσεων κατασκευάζουν και αποδομούν ζεύγη, όπως ακριβώς στην άσκηση 5.

Ορίστε τη δηλωτική σημασιολογία της WHILE^{cons}. Αναφέρετε τυχόν παραδοχές ή σχεδιαστικές επιλογές που κάνατε. Δώστε ιδιαίτερη προσοχή στην κατασκευή του πεδίου D που θα περιέχει τις σημασιολογικές τιμές των εκφράσεων! Ο ορισμός του είναι προφανώς αναδρομικός (οι τιμές μπορεί να είναι ζεύγη αποτελούμενα από τιμές).

Στη συνέχεια, κατασκευάστε σε Haskell έναν διερμηνέα για τη WHILE^{cons} βασισμένο και πάλι σε μία άμεση υλοποίηση της δηλωτικής σημασιολογίας. Ο διερμηνέας σας θα πρέπει να διαβάζει από την τυπική είσοδο (standard input) ένα πρόγραμμα WHILE^{cons}, να το εκτελεί και να εκτυπώνει την τελική τιμή της μεταβλητής `result`, όπως φαίνεται στα παρακάτω παραδείγματα. Για διευκόλυνσή σας, μπορείτε να χρησιμοποιήσετε το δοθέν αρχείο που υλοποιεί τη σύνταξη της WHILE^{cons}, έναν parser και έναν pretty-printer για αυτήν.

Παραδείγματα χρήσης του διερμηνέα.

`$ cat ex1`
`result := 0;`
`for succ succ succ succ succ succ 0 do`
`  for succ succ succ succ succ succ succ 0 do`
`  result := succ result`
	`$ runhaskell densem.hs < ex1`
	`42`

`$ cat ex2`
`x := succ succ succ succ succ succ succ 0;`
`result := false;`
`while 0 < x do (`
`  result := x : result;`
`  x := pred x`
`)`
	`$ runhaskell densem.hs < ex2`
	`1 : 2 : 3 : 4 : 5 : 6 : 7 : false`

Τι να παραδώσετε.

- Ένα αρχείο PDF (έστω και χειρόγραφο σκαναρισμένο) που να περιέχει τον ορισμό της σημασιολογίας της WHILE^{cons}— μην ξεχάσετε να συμπεριλάβετε έναν αυστηρό μαθηματικό ορισμό του πεδίου D που χρησιμοποιήσατε.
- Ένα αρχείο πηγαίου κώδικα Haskell που να υλοποιεί τον ζητούμενο διερμηνέα.