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

Γλώσσες Προγραμματισμού II

Οι ασκήσεις πρέπει να παραδοθούν 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 Γλώσσες συναρτησιακού προγραμματισμού

Προθεσμία παράδοσης: 26/4/2010

Το ταίριασμα προτύπων (pattern matching) είναι ένα από τα βασικότερα συστατικά των συναρτησιακών γλωσσών προγραμματισμού και ένα σημείο που οι υλοποιήσεις τους πρέπει να επιδείξουν προσοχή. Κάποιοι αλγόριθμοι για ικανοποιητική μεταγλώττιση του ταιριάσματος προτύπων περιγράφονται στα παρακάτω papers (σε χρονολογική σειρά):

1. Lennart Augustsson. Compiling pattern matching. In Functional Programming Languages and Computer Architecture, LNCS, vol. 201, pages 368–381, September 1985. Springer.

2. Marianne Baudinet and David MacQueen. Tree pattern matching for ML. December 1985. Available at http://www.smlnj.org/compiler-notes/85-note-baudinet.ps

3. Le Fessant and Luc Maranget. Optimizing pattern matching. In Proceedings of the Sixth ACM SIGPLAN International Conference on Functional Programming, pages 26–37, September 2001. ACM Press.

4. Luc Maranget. Compiling pattern matching to good decision trees. In Proceedings of the Workshop on ML, pages 35–46, September 2008. ACM Press.

Για τη διευκόλυνσή σας links για τα παραπάνω papers βρίσκονται και στην ιστοσελίδα του μαθήματος. Αυτό που η άσκηση σας ζητάει να κάνετε είναι να τα διαβάσετε και να γράψετε μια (συγ)κριτική περιγραφή (π.χ. 3–4 σελίδες) των κυριότερων αλγορίθμων που έχουν προταθεί για την επίλυση του προβλήματος της αποδοτικής μεταγλώττισης του ταιριάσματος προτύπων. Ένα από τα σημεία στα οποία η άσκηση σας ζητάει να επικεντρώσετε την προσοχή σας είναι να δείξετε/κατασκευάσετε περιπτώσεις συνόλων προτάσεων (case clauses) για τα οποία οι διαφορετικοί αλγόριθμοι δείχνουν τα συγκριτικά πλεονεκτήματά/μειονεκτήματά τους σε χώρο και σε χρόνο.

Άσκηση 9 Γλώσσες ταυτοχρονισμού

Προθεσμία παράδοσης: 26/4/2010

Επιλέξτε *μία* από τις παρακάτω δύο ασκήσεις:

1. Ίσως κάποιοι από σας να αναπολούν τις ασκήσεις των Γλωσσών Προγραμματισμού I. Τέρμα η νοσταλγία! Στο φετεινό μάθημα, η πρώτη σειρά ασκήσεων περιλαμβάνει μια άσκηση που ζητάει να γραφεί σε ML ένα πρόγραμμα που να βρίσκει τον ελάχιστο αριθμό κινήσεων για την επίλυση του παιχνιδιού "Rush Hour" (βλέπε την εκφώνηση της άσκησης στη σελίδα

`http://courses.softlab.ntua.gr/pl1/Exercises/exerc10-1.pdf`). Αυτό που σας ζητάει η συγκεκριμένη άσκηση είναι να γράψετε ένα παράλληλο πρόγραμμα στη γλώσσα της επιλογής σας που να επιλύει το συγκεκριμένο πρόβλημα. Για να βαθμολογηθεί το παράλληλο πρόγραμμά σας με άριστα σε αυτή την άσκηση θα πρέπει:

(α) είτε να καταφέρνει κάποιο ικανοποιητικό speedup όταν ο αριθμός των cores/threads στα οποία εκτελείται αυξάνεται (στο εργαστήριο λογισμικού μπορούμε να τρέξουμε το πρόγραμμά σας σε μηχανήματα από 1–16 cores)

(β) ή, αν δεν έχει κάποιο εντυπωσιακό speedup, τουλάχιστον να καταφέρνει να επιλύσει το πρόβλημα σε χρόνο μικρότερο από το μέσο όρο των 5 ταχύτερων σειριακών προγραμμάτων σε ML που οι σπουδαστές των Γλωσσών Προγραμματισμού Ι θα υποβάλλουν.

Με άλλα λόγια το παράλληλο πρόγραμμά σας στη γλώσσα της επιλογής σας δε μπορεί να είναι εντελώς "για τα μπάζα".

Εκτός από το πρόγραμμά σας και οδηγίες για το πως μπορεί κάποιος να το τρέξει, ως μέρος της υποβολής σας πρέπει να περιλαμβάνεται και ένα πρόγραμμα/script το οποίο να μπορεί να μετατρέψει την είσοδο από το format που η άσκηση των Γλωσσών Ι ορίζει για την ML στο format που το πρόγραμμά σας τη θέλει. (Για το script αυτό επιτρέπεται συνενόηση και κοινή δουλειά με άλλους συμφοιτητές σας.)

2. Στην ιστοσελίδα `http://shootout.alioth.debian.org/` υπάρχει το λεγόμενο "Computer Language Benchmarks Game" (πιο παλιά "Computer Language Shootout") όπου διάφορες υλοποιήσεις γλωσσών προγραμματισμού "διαγωνίζονται" μεταξύ τους σε ταχύτητα, απαιτήσεις μνήμης και, τελευταία, σε ταχύτητα όταν το πρόγραμμα εκτελείται σε περισσότερους από έναν πυρήνες. Τουλάχιστον για την άσκηση αυτή εμάς μας ενδιαφέρει περισσότερο η τελευταία αυτή παράμετρος των benchmarks.

   Επικεντρώστε την προσοχή σας στα benchmarks που εκτελούνται σε Quad-core μηχανήματα (`http://shootout.alioth.debian.org/u64q/`). Σχετικά σύντομα θα παρατηρήσετε ότι μόνο κάποια από τα benchmarks (`fannkuch`, `k-nucleotide`, `mandelbrot`, `spectral-norm`, `chameneos-redux`, `redex-dna`, `binary-trees`, `reverse-complement`) φαίνονται να μπορούν να παραλληλοποιηθούν.

   Η άσκηση είναι η εξής: επιλέξτε κάποιο από τα παραπάνω προβλήματα και όποια γλώσσα θέλετε (από αυτές που αναφέρονται στην ιστοσελίδα) και γράψτε ένα πρόγραμμα που να "βελτιώνει" κάπως την κατάσταση όπως αυτή αναφέρεται στην παραπάνω ιστοσελίδα. Βελτίωση της κατάστασης θεωρείται κάτι από τα παρακάτω (η λίστα δεν είναι εξαντλητική):

   - Να γράψετε ένα πρόγραμμα για το πρόβλημα Π στη γλώσσα Γ τέτοιο ώστε να τρέχει πιο γρήγορα από το καλύτερο υπάρχον πρόγραμμα για το πρόβλημα Π στη γλώσσα Γ.
   - Να τροποποιήσετε ένα ήδη υπάρχον Π+Γ πρόγραμμα ώστε να τρέχει πιο γρήγορα.
   - Να γράψετε ένα πρόγραμμα για το πρόβλημα Π στη γλώσσα Γ για κάποιο συνδυασμό Π και Γ για το οποία δεν υπάρχει κάποια λύση.
   - Να παραλληλοποιήσετε ένα ήδη υπάρχον πρόγραμμα για το πρόβλημα Π στη γλώσσα Γ.
   - ... *Χρησιμοποιήστε τη φαντασία σας* ...

Αν χρειάζεστε κάποιο πολυπυρηνικό μηχάνημα για την ανάπτυξη του προγράμματός σας, μπορούμε να σας φτιάξουμε κάποιο προσωρινό account στο Εργαστήριο Τεχνολογίας Λογισμικού (SoftLab).