Τμήμα: Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Μάθημα: *Λογικός Προγραμματισμός*
Διδάσκοντες: Μανόλης Μαρακάκης, Χαρίδημος Κονδυλάκης
Ημερομηνία παράδοσης 16 Ιανουαρίου 2022
Χειμερινό εξάμηνο 2021-22

Εργασία 5

Άσκηση 1

Να υλοποιήσετε την *ταξινόμηση με συγχώνευση (merge sort)* με ακολουθίες. Να χρησιμοποιήσετε τις πράξεις των ακολουθιών που ορίστηκαν και υλοποιήθηκαν στην Ενότητα 8.5.1 του βιβλίου σας και αυτές που ορίστηκαν στην άσκηση 8.12 του βιβλίου σας. Όποια πράξη χρησιμοποιήσετε από την άσκηση 8.12 να την υλοποιήσετε. Το κατηγόρημα «merge_sort(Q:seq(T), S:seq(T))» είναι αληθές εάν η ακολουθία S έχει τα στοιχεία της ακολουθίας Q ταξινομημένα κατ' αύξουσα σειρά. Εάν τα στοιχεία της ακολουθίας Q είναι άτομα να χρησιμοποιήσετε τη *ταξινόμηση όρων της Prolog*, T = atom, «selectionSort(Q:seq(atom), S:seq(atom))». Εάν τα στοιχεία της ακολουθίας Q είναι ακέραιοι να χρησιμοποιήσετε τη *ταξινόμηση ακεραίων*, δηλαδή T = integer, «merge_sort(Q:seq(integer), S:seq(integer))». Η στρατηγική «*διαίρει και βασίλευε*» ακολουθείται για την επίλυση *της ταξινόμησης με συγχώνευση* ως εξής. Έχουμε μια ακολουθία Q την οποία θέλουμε να την ταξινομήσουμε. Μπορούμε να χωρίσουμε την ακολουθία Q σε δύο υπο-ακολουθίες Q1 και Q2 περίπου στο ίδιο μέγεθος, στη συνέχεια, θα προσπαθήσουμε να ταξινομήσουμε την κάθε υπο-ακολουθία Q1 και Q2. Αυτή η διαδικασία μπορεί να επαναληφθεί μέχρι να επιτευχθεί η βασική περίπτωση που μπορεί να είναι μία ακολουθία με ένα στοιχείο που δεν μπορεί να διαιρεθεί περαιτέρω. Η ιδέα είναι ότι μια ακολουθία μικρότερου μεγέθους ταξινομείται πολύ πιο εύκολα. Στη συνέχεια, οι ταξινομημένες υπο-ακολουθίες μπορούν να συγχωνεύονται ανά δύο δημιουργώντας μια ταξινομημένη ακολουθία μέχρι την τελική λύση. Εάν οι ταξινομημένες υπο-ακολουθίες των Q1 και Q2 είναι οι S1 και S2, τότε η συγχώνευση τους θα δώσει την ταξινομημένη ακολουθία S.

4.0 μονάδες.

Άσκηση 2

Να γράψετε κατηγόρημα parenthesis_LR_same(S) το οποίο είναι αληθές εάν η συμβολοσειρά των παρενθέσεων τις οποίες δέχεται μέσω της παραμέτρου S περιέχει ίσο πλήθος αριστερών και δεξιών παρενθέσεων. Να μη χρησιμοποιήσετε μετρητές ώστε να μετράτε τις παρενθέσεις αλλά να χρησιμοποιήσετε την δομή δεδομένων στοίβα. Επιπλέον τις παρενθέσεις να τις βάλετε σε μια λίστα την οποία θα χειριστείτε ως ουρά. Στην υλοποίηση σας να χρησιμοποιήσετε τις πράξεις για τις στοίβες και τις ουρές όπως υλοποιήθηκαν στις ενότητες 8.5.7 και 8.5.8 αντίστοιχα του βιβλίου σας. Παραδείγματα στόχων: 1) «?-parenthesis_LR_same(‘)))(()((‘).» yes. 2) «?-parenthesis_LR_same(‘((())‘).» no**.**

3.0 μονάδες.

Άσκηση 3

Να γράψετε κατηγόρημα queue_element_cardinality(Q1, Q2) το οποίο είναι αληθές εάν Q1 είναι μια ουρά με στοιχεία ζεύγη μορφής (Στοιχείο, Πλήθος_επαναλλήψεων) όπου το «Στοιχείο» είναι ένας ακέραιος αριθμός και το «πλήθος_επαναλλήψεων» είναι ένας θετικός ακέραιος αριθμός. Τέλος, Q2 είναι μια ουρά που περιέχει κάθε «Στοιχείο» της Q1 με πληθικότητα «πλήθος_επαναλλήψεων» σε διαδοχικές θέσεις. Στην υλοποίηση σας να χρησιμοποιήσετε τις πράξεις για τις ουρές όπως υλοποιήθηκαν στην ενότητα 8.5.8 του βιβλίου σας. Παράδειγμα στόχου, «?- queue_element_cardinality([(5,4), (-8,3), (0,2)], Q2).» Q2 = [5,5,5,5,-8,-8,-8,0,0].

3.0 μονάδες.