

Όνομα : Μιχάλης Μήτσιος

A.M.: 1115201500096

3η Εργασία Compilers 2020

Η 3η εργασία καλεί την 2η όπως είναι με κάποιες βελτιώσεις τις οποίες αναφέρω παρακάτω:

Το EvalVisitor.java έχει ένα break στην σειρά 228.

Στο typechecking έχει επιτραπεί η περίπτωση όπου

```
class A{}  
Class B extends A{}
```

```
A foo(){  
    return B;  
}
```

Δηλαδή να γίνει return type το παιδί στην θέση τύπου parent.

Μια ακόμα διαφορά υπάρχει στην σειρά 517 όπου περνάω στο expression list τον πατέρα της συνάρτησης εάν αυτός είναι ο μοναδικός που την έχει.

Τέλος η μεγαλύτερη αλλαγή που έγινε είναι η χρήση του ArrayList<Integer> all_expression_counters αντί του var_counter. Το var_counter ήταν ένας counter για να γνωρίζω σε ποια από τις μεταβλητές βρίσκομαι στην κλήση της συνάρτησης. Εάν όμως υπήρχε κλήση expression list μέσα σε κάποιο άλλο expression list τότε αυτό το counter επειδή είναι global θα χανόταν. Τώρα με το all_expression_counters η κάθε expression list έχει και τον δικό της counter ανεξάρτητο από τις υπόλοιπες. Το total_expressions είναι το πλήθος των παραπάνω counters.

Το νέο κομμάτι της 3ης εργασία αποτελεί το `llvm_compiler.java`

Έγινε αρκετή χρήση των παραδειγμάτων που δόθηκαν από το φροντιστήριο.

Αρχικά έχει γίνει ορισμός μιας συνάρτησης `emit` για να γράφω στο αρχείο.

Έπειτα επειδή η γλώσσα είναι `single assignment` έχουν οριστεί `global` μεταβλητές με συνοδεία των αντίστοιχων συναρτήσεων για την παραγωγή των `registers` (που η ονομασία τους αλλάζει ανάλογα την χρήση τους).

Αρχικά καλείται ο `Goal visitor`

Σε αυτόν δημιουργούνται όλα τα `v_tables` για όλες τις κλάσεις με την βοήθεια του `symboltable` που έχουμε από πριν. Όλες οι μέθοδοι της κλάσης καταλήγουν σε έναν βοηθητικό `LinkedHashMap` τον `v_table_helper` και τοποθετούνται στην θέση του `offsets/8` (για να γίνει `actual position 0,8,16=>0,1,2`) τους. Πρώτα γίνεται σκαν των `parents` και μπαίνουν όλες οι συναρτήσεις και μετά της ίδιας της κλάσης που είμαστε μέσα η οποία κάνει και `override` την προηγούμενη αποθήκευση σε περίπτωση που μια συνάρτηση έχει ξανά δηλωθεί.

Τέλος στο `readme` θα ήθελα να πω για το `Var_Type` που είναι ένα βοηθητικό `LinkedHashMap<String,String>` το οποίο αποθηκεύει όλους τους `registers` με το αντίστοιχο `type` τους ώστε όταν θέλω να καλέσω κάποιος από κάποιον `visitor` να φαίνεται σε όλους. Το παραπάνω γίνεται `clear` στο τέλος των μεθόδων (μαζί και της `main`), όπου και οι ίδιοι οι `registers` εκεί μηδενίζονται. Αυτό γίνεται σε αυτό το σημείο καθώς όταν φύγουμε από την συνάρτηση που έχει οριστεί σταματάει το πεδίο επιρροής ενός `register`. Με λίγα λόγια δεν μας ενδιαφέρουν πια με το που βγούμε εκτός της συνάρτησης που χρησιμοποιήθηκαν.

Σας ευχαριστώ για τον χρόνο σας.