



Άσκηση 3

Μεταγλωττιστές

Εαρινό Εξάμηνο 2019-2020

Παντελιάδη Ευφροσύνη

- Αριθμός Μητρώου: 1115201400301
- Email: sdi1400301@di.uoa.gr

Εισαγωγή

Στα πλαίσια του τρίτου και τελικού project έχουν δημιουργηθεί τα παρακάτω αρχεία:

- VTables.java
- ClassTables.java
- Translator.java
- Makefile

Για να εκτελεστεί η άσκηση πρέπει πρώτα να μεταγλωτιστεί. Για τον σκοπό αυτό έχει δημιουργηθεί ένα *Makefile*, το οποίο το τρέχουμε γράφοντας `make` στη γραμμή εντολών. Στη συνέχεια, μπορούμε να εκτελέσουμε την άσκηση πληκτρολογώντας `java Main <input_file>`, όπου *<input_file>* το αρχείο που επιθυμούμε να “μεταφράσουμε” σε LLVM κώδικα. Το αποτέλεσμα θα είναι να δημιουργηθεί το αρχείο *<input_file>.ll*. Για να εκτελέσουμε το νέο αρχείο γράφουμε `clang -o <result_file> <input_file>.ll` ακολουθούμενο από την εντολή `./<result_file>`. Μόλις ολοκληρωθεί η εκτέλεση μπορούμε να διαγράψουμε όλα τα *.class* αρχεία που δημιουργήθηκαν τρέχοντας την εντολή `make clean`.

Περιγραφή Αρχείων

VTables.java:

Εδώ, βρίσκονται οι υλοποιήσεις ορισμένων δομών από τις οποίες η σημαντικότερη είναι η **VTables**. Η συγκεκριμένη κλάση αναφέρεται στο σύνολο των πινάκων (πεδίων και v-tables) κάθε κλάσης.

Η εκτύπωση των v-tables στο *.ll* αρχείο που θα δημιουργηθεί γίνεται από τη μέθοδο **writeVTables**. Τα v-tables πρέπει να έχουν συγκεκριμένη σειρά στο αρχείο. Συγκεκριμένα, πρώτα θα πρέπει να εμφανίζονται οι θυγατρικές κλάσεις μιας κλάσης, στη συνέχεια η γονική και τελευταία η κλάση που περιέχει τη *main* μέθοδο. Για τον σκοπό αυτό έχει δημιουργηθεί μια στοίβα στην κορυφή της οποίας βρίσκονται οι κλάσεις των οποίων τα v-tables θα πρέπει να εκτυπωθούν πρώτα. Κατασκευάζουμε βήμα-βήμα το κατάλληλο String, το μετατρέπουμε σε έναν πίνακα από bytes και το εκτυπώνουμε στο αρχείο.

Μέσα στο συγκεκριμένο αρχείο βρίσκονται και οι κλάσεις **Functions** και **VariableType**. Η πρώτη περιέχει τη static μέθοδο **declareFunctions**, η οποία χρησιμοποιείται σε συνδυασμό με

την εντολή **declare** για να δηλώσουμε όλες τις εξωτερικές μεθόδους που θα χρησιμοποιηθούν. Η δεύτερη κλάση λειτουργεί ως βοηθητική και χρησιμοποιείται στη συνέχεια στον visitor όποτε θέλουμε να εντοπίσουμε αν μια μεταβλητή είναι τοπική μεταβλητή μιας μεθόδου ή πεδίο κάποιας κλάσης.

ClassTables.java:

Στο παραπάνω αρχείο οργανώνονται οι πίνακες πεδίων και μεθόδων (v-tables) μιας κλάσης.

Translator.java:

Πρόκειται για έναν επιπλέον visitor που δημιουργήθηκε για την υλοποίηση του συγκεκριμένου project. Στα πεδία του εμπεριέχονται διάφορα references τα οποία αναφέρονται στην κλάση με την οποία δουλεύουμε (**currentClass**), τη μέθοδο με την οποία δουλεύουμε (**currentMethod**) και τη μεταβλητή/πεδίο με την οποία δουλεύουμε (**currentVariable**), αντίστοιχα.

Εδώ, κάθε visit μέθοδος δέχεται ως δεύτερο όρισμα ένα αντικείμενο της κλάσης **Info** και επιπλέον επιστρέφει ένα αντικείμενο της συγκεκριμένης κλάσης.

Στις visit μεθόδους χρησιμοποιείται πολύ συχνά η ήδη υπάρχουσα κλάση **FieldInfo** για να περιγράψουμε το αντικείμενο το οποίο επιστρέφεται.

Ο κώδικας που θα εκτυπωθεί μέσα στο `./` αρχείο δημιουργείται σιγά-σιγά ως String, το οποίο μετατρέπεται στη συνέχεια σε έναν πίνακα από bytes. Τέλος, γράφεται στο αρχείο μέσω της μεθόδου **writeOutput**.

Σημειώσεις

Έχουν δοκιμαστεί όλα τα αρχεία που είχαν δοθεί στο φροντιστήριο (classes.java, arrays.java, and.java, if.java, simple.java), το παράδειγμα με τους boolean πίνακες που αναφέρθηκε στο Piazza (<https://piazza.com/class/k6qh6av05cj4kw?cid=136>), καθώς και ορισμένα αρχεία που δημιούργησα.

