README

Project 3 Compilers

Παναγιώτης Ευαγγελίου

1115201500039

#Γενικά Σχόλια

- Το πρόγραμμα τρέχει σωστά για όλες τις περιπτώσεις , δηλαδή δοκιμάστηκαν τόσο τα παραδείγματα για τα οποία είχαν δοθεί και τα αντίστοιχα ll αρχεία καθώς επίσης και τα παραδείγματα minijava-extra καθώς και κάποια δικά μου επιπλέον.
- Προκειμένου να γίνει η μεταγλώττιση θα πρέπει **στον φάκελο Project3_sdi1500039/** να **μπουν τα αρχεία javacc5.jar και jtb123di.jar** .
- Εφόσον μπουν, **μπαίνουμε στον φάκελο Project3_sdi1500039/miniJava/** και η εντολή για μεταγλώττιση είναι **make all**.
- Η εντολή για εκτέλεση είναι java Main file1 [file2] [file3] ... [fileN] , δηλαδή θα πρέπει να δοθεί τουλάχιστον 1 αρχείο προς μεταγλώττιση και όσο πλήθος αρχείων θέλει ο χρήστης προκειμένου να παραχθεί για ένα-ένα το αντίστοιχο αρχείο ll.
- Παρόλο που τα αρχεία που θα δοθούν δεν θα έχουν κάποιο error από θέμα type check , έχω αφήσει τους 2 πρώτους visitor από το Project2 μια και μας στείλανε τους βαθμούς και δεν είχα κάποιο λάθος. Ο πρώτος visitor , ο TypeGathererVisitor, είναι απαραίτητος μια και είναι αυτός που θα μαζέψει τις απαραίτητες πληροφορίες για τις κλάσεις κλπ οι οποίες θα χρειαστούν οπωσδήποτε και στο κομμάτι της μετατροπής σε llvm bitcode. Ο δεύτερος visitor , ο TypeCheckerVisitor , κάνει το type checking και μπορεί να παραλειφθεί αν θέλουμε.
- Ο visitor που αφορά την μετατροπή σε llvm bitcode είναι ο **LlvmVisitor**.
- Για τα αρχεία που δόθηκαν ως είσοδο, δημιουργείται για το καθένα, αρχείο της μορφής **specialClassName.ll** όπου εκεί είναι ο αντίστοιχος κώδικας σε llvm bitcode.Τα αρχεία αυτά βρίσκονται στον φάκελο Project3_sdi1500039/miniJava/Output/ .

#Παρατηρήσεις υλοποιήσης

- Οι δομές που χρησιμοποιήθηκαν έμειναν όπως είναι από το project2 απλώς προστέθηκαν κάποιες επιπλέον μέθοδοι όπως στο αρχείο Classinfo.java οι putMyMethodsInVtable, fieldClassOwner, methodUpperClassOwner, getTotalFieldBytes, getTotalMethodNumber.
- Αρχικά , στον constructor του LlvmVisitor , αξιοποιώντας τις πληροφορίες που έχουμε ήδη από τους visitors του Project2, κάνουμε emit τους v_tables για κάθε class καθώς και τις έτοιμες helper methods.
- Στα Statement() περνάμε σαν όρισμα (String[] argu) το όνομα της class (argu[0]) και της method (argu[1]) που βρίσκεται το statement ώστε να περαστούν με την σειρά τους στα expressions/primaryExpressions ώστε όταν χρειαστεί να βρούμε τον τύπο κάποιας μεταβλητής ή πληροφορίες για αυτήν όπως offset κλπ.
- Εφόσον οι registers και τα labels είναι “τοπικά” και ανεξάρτητα σε κάθε συνάρτηση του llvm bitcode, όπως αναφέρθηκε και στο piazza , στο τέλος κάθε MethodDeclaration τα μηδενίζουμε.
- Όταν μας επιστρέφεται ένας identifier, αν είναι τοπική μεταβλητή τότε δεν έχουμε κάποιο πρόβλημα και τον χρησιμοποιούμε από το όνομα του register , που είναι το ίδιο με το όνομα του identifier. Αν όμως δεν είναι τοπική μεταβλητή πρέπει να το φορτώσουμε από τον χώρο που δεσμεύσαμε για το κάθε αντικείμενο άρα χρειαζόμαστε το offset της μεταβλητής. Για το λόγο αυτό:
  - Αρχικά βρίσκουμε την class που έκανε τελευταία φορά hide το field , διότι μια κλάση μπορεί να μην έχει σαν “δικό της καθεαυτό” ένα πεδίο και να το έχει κληρονομήσει από

κάποια superClass της, και μετά παίρνουμε το offset, διότι για τα κληρονομημένα πεδία το offset βρίσκεται μόνο στην “αρχική” class που έχει αυτούσιο το πεδίο.

- Τα expressions/primeExpressions επιστρέφουν String της μορφής “type register” ή “type literalValue” , όπου μερικές φορές όμως δεν χρειαζόμαστε το type σε μερικές εντολές llvm bitcode οπότε και το κάνουμε split και χρησιμοποιούμε μόνο το register/literalValue.
- Για το messageSend:
  - Στο PrimaryExpresion() περνάμε String[] argu μεγέθους 3 και όχι μεγέθους 2 όπως τις άλλες φορές διότι χρειαζόμαστε να ξέρουμε όχι μόνο τον register και τον τύπου του register αλλά και τον τύπο – όνομα κλάσης – του αντικειμένου που καλούμε την method του. Η πληροφορία αυτή θα μπεί στην 3η θέση argu[2] του ορίσματος. Και την πληροφορία αυτή την βάζουμε, ελέγχοντας στα αντίστοιχα PrimaryExpressions τα οποία μπορούν να βρίσκονται αριστερά από την “.” του messageSend (όπως το messageSend, this, γενικά PrimaryExpression όταν είναι identifier, AllocationExpression) ελέγχοντας αν το argu έχει length 3, πράγμα που σημαίνει ότι ήρθαμε από κάποιο messageSend άρα πρέπει να αποθηκεύσουμε την συγκεκριμένη πληροφορία.
  - Για να βρούμε το offset της method που πρέπει να καλέσουμε θα το βρούμε από την πρώτη ιεραρχικά class που βρίσκεται η method, γιατί μόνο εκεί αποθηκεύουμε το offset των μεθόδων που γίνονται override.