MJCompiler

Programski prevodioci 1

Stefan Todorović

426/2017

# 1.Postavka projekta

Cilj projektnog zadatka je realizacija kompajlera za programski jezik Mikrojava. Programski prevodilac za Mikrojavu ima četiri funkcionalnosti: leksičku analizu, sintaksnu analizu, semantičku analizu i generisanje koda.

Leksički analizator treba da prepoznaje jezičke lekseme i vrati skup tokena izdvojenih iz izvornog koda, koji se dalje razmatraju u okviru sintaksne analize.

Sintaksni analizator ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu formiraju gramatički ispravne sentence. Rezultat sintaksne analize je apstraktno sintaksno stablo.

Semantička analiza se sprovodi implementacijom metoda za posećivanje  
čvorova apstraktnog sintaksnog stabla. Stablo je formirano na osnovu gramatike implementirane u prethodnoj fazi.

Generator koda prevodi sintaksno i semantički ispravne programe u izvršni oblik za odabrano izvršno okruženje Mikrojava VM. Generisanje koda se implementira na sličan način kao i semantička analiza, implementacijom metoda koje posećuju čvorove.

# 2.Opis komandi

* build.xml, target = lexerGen – generisanje leksičkog analizatora Yylex.java
* build.xml, target = compile – generisanje sintaksnog analizatora MJParser.java i prevođenje svih napisanih java fajlova
* build.xml, target = runObj – pokretanje disasemblera, ispis sadržaja objektnog fajla i pokretanje tog koda
* MJParserTest.java – pokretanje testa koji prevodi Mikrojava program u objektni fajl (argumenti: test/program.mj test/program.obj)
* MJTest.java – testiranje leksičkog analizatora
* rs.etf.pp1.mj.runtime.Run – izvršavanje objektnog fajla bez dibager opcije (argumenti: test/program.obj)
* rs.etf.pp1.mj.runtime.Run – izvršavanje objektnog fajla sa dibager opcijom (argumenti: test/program.obj -debug)

# 3.Test primeri

* test1.mj – prikaz funkcionalnosti mnozenja skalara s vektorom, vektora s vektorom i do...while petlje
* test2.mj – prikaz funkcionalnosti poziva procedure, if naredbe i break naredbe
* test3.mj – prikaz optimizacije u slucaju racunanja uslova za if naredbu i do…while petlju
* test302jul.mj – test za B nivo iz julskog roka (obrisana switch naredba)

# 4.Novouvedene klase

* Table.java – klasa koja proširuje klasu Tab.java. Omogućava dodavanje bool tipa u tabelu simbola. Pored toga proverava da li je definisan simbol zadatog imena u najužem opsegu
* TSDumpSymbolTableVisitor.java – klasa koja proširuje klasu DumpSymbolTableVisitor.java i redefiniše njenu metodu visitStructNode. Na taj način omogućava da u ispisu bude ispisan strukturni čvor bool tipa
* OperatorType.java – enumerator koji sadrži dve vrednosti koje odgovaraju logičkim operatorima AND i OR
* LogicalOperator.java – klasa koja definiše jedan logički operator. Definiše tip operatora i fleg koji nam govori da li smo ga već posetili u obilasku.
* LogicOperatorsCounter.java – klasa koja proširuje klasu VisitorAdaptor.java. Vodi evidenciju o tome da li smo posetili AND ili OR logički operator i to beleži u listi svih logičkih operatora u izrazu. Čuva informaciju o tome koliko smo
* ServiceObject.java – klasa koja vodi evidenciju o adresama koje bi trebalo da se preprave kako bi se skokovi izvršavali korektno. Vodi evidenciju i o trenutnom bloku naredbi koji se izvršava u okviru if naredbe ili do...while petlje
* CompilerImpl.java – implementira intefejs Compiler i metodu compile
* MethArgs.java – klasa koja vodi evidenciju o definisanim funkcijama i tipovima argumenata te funkcije
* AllMethArgs.java – lista svih MethArgs objekata koji su definisani