**PROGRAMSKI PREVODIOCI 1**

**Kompajler za programski jezik MikroJava**

**Elektrotehnički Fakultet Univerziteta u Beogradu,**

**Katedra za Računarsku tehniku i informatiku**

Profesor: dr Dragan Bojić

Asistenti: Kristijan Žiža, Maja Vukasović

Student: Dušan Damljanović 526/2014

**Zadatak**

Projektni zadatak podrazumeva kreiranje kompajlera za programski jezik mikroJava.

Projekat ima četiri funkcionalnosti:

* leksički analizator
* sintaksni analizator
* semantički analizator
* generator koda

*Leksička analiza* podrazumeva prepoznavanje i izdvajanje tokena iz ulaznog fajla u .mj formatu uz proveru leksičkih grešaka. Za ove potrebe koristimo alat JFlex. Nakon toga, prepoznate tokene ispisujemo na standardni izlaz.

*Sintaksna analiza* sastoji se od uparivanja delova .mj fajla sa odgovarajućim smenama. Sintaksne greške ispisujemo na standardni izlaz, a za ovu fazu koristimo java\_cup alat. Rezultat ove faze je apstraktno sintaksno stablo.

*Semantička analiza* vrši se obilaskom apstraktnog sintaksnog stabla, i vrši proveru ispravnosti semantike programa.

Četvrta faza, *generisanje koda*, podrazumeva generisanje bajtkoda za mikroJavu korišćenjem mj-runtime alata.

**Komande za pokretanje**

**Lexer Generator**

Main Class: JFlex.Main

Program arguments: -d src\rs\ac\bg\etf\pp1 spec\mjlexer.flex

**Parser Generator**

Main Class: java\_cup.Main

Arguments: -destdir rs\ac\bg\etf\pp1 -dump\_states -parser MJParser -ast rs.ac.bg.etf.pp1.ast -buildtree ..\spec\mjparser.cup

Working directory: ${workspace\_loc:MJCompiler\src}

**Compile MJ Source:**   
Main Class: rs.ac.bg.etf.pp1.Compiler  
Arguments: test\program.mj test\program.obj >izlaz.out 2>izlaz.err

**Disasm**

Main Class: rs.etf.pp1.mj.runtime.disasm

Arguments: test\program.obj

**Run**

Main Class: rs.etf.pp1.mj.runtime.Run

Arguments: -debug test\program.obj

**MJTest**

Main Class: rs.ac.bg.etf.pp1.MJTest

**Testovi**

**Oporavak od grešaka**

Prvi test ilustruje oporavak od grešaka koji je realizovan u toku semantičke analize. Testira se oporavak od grešaka u svim slučajevima kada je to zahtevamo za nivo B i ispisuje informaciju o oporavku, o broju linije gde se greška desila i o karakteru do kog ignorišemo ulaz da bismo izvršili oporavak.

**Nivo A**

Ovaj test nalazi se u fajlovima test1.mj i test1.out i ispituje funkcionalnosti zahtevane za nivo A. U pitanju je modifikovana verzija priloženog fajla test301.mj. Testiraju se deklaracije promenljivih sva tri osnovna tipa, nizova i nabrajanja i konstantne vrednosti. Osim deklaracija, za navedene tipove proveravamo i dodelu vrednosti, aritmetičke operacije, komplikovanije izraze i različite sintaksne oblike i smene. Pozivaju se i funkcije čitanja i pisanja. Ono što ne potpada pod zahtevani nivo a nalazi se u tekstu je if naredba I ispitivanje uslova, koja je dodata u ovaj test da bismo proverili ispravnost bool tipa.

**Test2**

Ovaj test nalazi se u fajlovima test2.mj I test2.out. Ispituje funkcionalnosti zahtevane za nivo B. U pitanju je modifikovana verzija priloženog fajla test302.mj. I u ovom testu imamo razne elemente koje smo testirali prethodnim fajlom uz dodatak, sada imamo funkcije, inicijalizatorske liste i petlje. Cilj ovog testa je da, kao što je navedeno u zadatku, budemo sigurno da argumenti i povratne vrednosti funkcija u pozivu odgovaraju formalnim parametrima i tipu pri njihovoj deklaraciji. Testiramo relativne operacije u uslovima i if i else blokove u odnosu na date uslove, korektan broj ponavljanja bloka kod for petlji, korektan format inicijalizatorskih listi.

**Test3**

Ovaj test predstavlja nadogradnju prethodnog. On ilustruje neke od netrivijalnih grešaka koje prepoznaje, kao što su pogrešni povratni tipovi, različit broj elemenata inicijalizatorske liste i njene veličine kao i različiti tipovi elemenata inicijalizatorske liste, različit broj i tip formalnih I stvarnih argumenata funkcije, višestruka definicija člana nabrajanja.