**Projekat**

Kompajler za Mikrojavu

Programski prevodioci 1

Elektrotehnički fakultet u Beogradu, Avgust 2023.

Nemanja Krivokapić 2019/0588

*Opis zadatka*

Programski prevodilac za Mikrojavu ima četiri osnovne 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. Ukoliko se tokom leksičke analize detektuje leksička greška, potrebno je ispisati odgovarajuću poruku na izlaz.

**Sintaksni analizator** ima zadatak da utvrdi da li izdvojeni tokeni iz izvornog koda programa mogu formiraju gramatički ispravne sentence. Tokom parsiranja Mikrojava programa potrebno je na odgovarajući način omogućiti i praćenje samog procesa parsiranja na način koji će biti u nastavku dokumenta detaljno opisan. Nakon parsiranja sintaksno ispravnih Mikrojava programa potrebno je obavestiti korisnika o uspešnosti parsiranja. Ukoliko izvorni kod ima sintaksne greške, potrebno je izdati adekvatno objašnjenje o detektovanoj sintaksnoj grešci, izvršiti oporavak i nastaviti parsiranje. Semantički analizator se formira na osnovu apstraktnog sintaksnog stabla koje je nastalo kao rezultat sintaksne analize.

**Semantička analiza** se sprovodi implementacijom metoda za posećivanje čvorova apstraktnog sintaksnog stabla. Stablo je formirano na osnovu gramatike implementirane u prethodnoj fazi. Ukoliko izvorni kod ima semantičke greške, potrebno je prikazati adekvatnu poruku o detektovanoj semantičkoj grešci.

**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.

*Pokretanje*

<?xml version="1.0" encoding="UTF-8"?>

<project name="MJCompiler" default="compile" basedir=".">

<target name="delete">

<delete>

<fileset dir="src/rs/ac/bg/etf/pp1">

<exclude name="sym\_old.java"/>

<exclude name="util/Log4JUtils.java"/>

<exclude name="Yylex.java"/>

<exclude name="RuleVisitor.java"/>

<exclude name="SemanticPass.java"/>

<exclude name="CodeGenerator.java"/>

<exclude name="CounterVisitor.java"/>

</fileset>

</delete>

<delete>

<fileset dir="spec">

<exclude name="mjlexer.flex"/>

<exclude name="mjparser.cup"/>

</fileset>

</delete>

</target>

<target name="lexerGen" depends="delete">

<java jar="lib/JFlex.jar" fork="true">

<arg value="-d"/>

<arg value="./src/rs/ac/bg/etf/pp1"/>

<arg value="spec/mjlexer.flex"/>

</java>

</target>

<target name="parserGen" depends="delete">

<java jar="lib/cup\_v10k.jar" fork="true">

<arg value="-destdir"/>

<arg value="src/rs/ac/bg/etf/pp1"/>

<arg value="-ast"/>

<arg value="src.rs.ac.bg.etf.pp1.ast"/>

<arg value="-parser"/>

<arg value="MJParser"/>

<arg value="-dump\_states"/>

<arg value="-buildtree"/>

<arg value="spec/mjparser.cup"/>

</java>

</target>

<target name="repackage" depends="parserGen">

<!-- Replaces all of the references to the old package name in files in the "src" directory -->

<replace dir="src" value="rs.ac.bg.etf.pp1.ast" token="src.rs.ac.bg.etf.pp1.ast" summary="true"/>

</target>

<target name="compile" depends="repackage">

<javac srcdir="src/rs/ac/bg/etf/pp1" includeantruntime="false">

<classpath>

<pathelement path="lib/JFlex.jar"/>

<pathelement path="lib/cup\_v10k.jar"/>

<pathelement path="lib/log4j-1.2.17.jar"/>

<pathelement path="lib/symboltable-1-1.jar"/>

<pathelement path="lib/mj-runtime-1.1.jar"/>

</classpath>

</javac>

</target>

<target name="disasm">

<java classname="rs.etf.pp1.mj.runtime.disasm">

<arg value="test/program.obj"/>

<classpath>

<pathelement location="lib/mj-runtime-1.1.jar"/>

</classpath>

</java>

</target>

<target name="runObj" depends="disasm" >

<java classname="rs.etf.pp1.mj.runtime.Run">

<arg value="test/program.obj"/>

<redirector input="ulaz.txt" output="izlaz.out">

</redirector>

<classpath>

<pathelement location="lib/mj-runtime-1.1.jar"/>

</classpath>

</java>

</target>

</project>

Build.xml fajl se pokreće kako bi se sve stvari izvršile.

Ispis rezultata se radi na izlaz.out.

*Testovi*

**Program.mj** javni test primer projekta.

**Test1.mj** osnovni test koji radi sa osnovnim operacija, printovanjem i citanjem.

**Test2.mj** test koji radi nad nizovima.