Универзитет у Београду

Електротехнички факултет



**Програмски преводиоци 1**

Лука Шотра 2019/0204

**Опис поставке пројектног задатка**

Циљ овој пројекта је реализација компајлера за микројаву. Компајлер треба да преводи улазне микројава фајлове (.mj) у извршне (.obj) фајлове.

Пројекат се састоји из 4 фазе:

**1) Лексичка анализа** – очитава све улазне симболе из улазног фајла и групише их у лексичке токене. Уколо се на улазу појави симбол који није дефинсан у нашој граматици, лексичка анализа ће пријавити грешку.

**2) Синтаксна анализа** – проверава синтаксну исправност улазног микројава фајла уз помоћ LALR(1) парсерa. Токени који се налазе у излазном фајлу који је генерисао лексер се користе како би оформиле исправне граматичке смене дефинисане поставком пројекта. Уколико се од лексичких токена не може формирати исрравна граматичка смена парсер ће на том месту да пријави грешку. Излаз синтаксне анализе је апстрактно синтаксно стабло (AST).

**3) Семантичка анализа** – проверава семантичке услове задате поставком пројекта и тестира исправност синтаксно исправног написанког улазног кода. Семантичка анализа се реализује рефесинисањем методе visit() за сваки чвор који нам је потребан током обиласка кроз апстракстно синтаксно стабло. Уколико се нађе на неку семантичку грешку, наша семантичка анализа ће пријавити грешку.

**4) Генерисање кода** – генерише код који може да се изврши на микројава VM за све синтаксно и сематички исправне улазне микројава програме. Излаз генератора кода је .obj фајл који је извршив на микројава виртуелној машини.

**Опис команди за покретање и превођење**

Прва и друга фаза се преводи покретањем команде lexerGen и команде compile унутар build.xml фајла, који покреће извршавање свих неопходних програма (parserGen, repackage, delete) као што су лексер и парсер који генеришу своје излазне фајлове. Трећа фаза се извршава покретањем Compiler класе која се налази унутар test фолдера која нам исписује табелу симбола на излаз и пријављује све сематичке грешке уколико постоје. Четврта фаза се преводи покретањем runObj команде унутар build.xml фајла који у позадини покрене све неопходне ствари везане за генерисање кода који изврши превођење нашег улазног микројава кода у .obj фајл унутар истог директоријума где се налазио и наш тест пример.

**Опис новоуведених класа**

**1) mjlexer.flex** – упарује симболе са улазног фајла са предефинисаним лексичким симболима и пријављује грешку уколико се у улазном фајлу појави симбол који није у складу са граматиком задатом у спецификацији пројекта.

**2) mjparser.cup** – упарује лексичке токене са граматичким сменама и пријављује грешку уколико од лексичких токена са излаза лексера не може да се формира исправна граматичка смена.

**3) SemanticAnalyzer.java** – пролази кроз апстркатно синтаксно стабло у post-order поретку и проверава да ли је код поред тога што је синтаксно исправан такође и семантички исправан и пријављује грешку уколико наиђе на неку семантичку грешку.

**4) CodeGenerator.java** – извршава генерисање кода за све синтаксно и семантичко исправне улазне фајлове обиласком апстракног синтаксног стабла у post-order поретку и креира инструкције које се касније извршавају на микројава виртуелној машини.

**5) Compiler.java** – класа извршава лексичку, синтаксну i семантичку анализу улазног програма чије је име прослеђено кроз аргумент командне линије и уколико је улазни програм исправно написан класа креира .obj фајл који може да се изврши на микројава виртуалној машини.

**Опис тестова**

**1) test\_semantic\_errors.mj** – тестира већину семантичких грешака које могу да се јаве унутар улазног микројава фајла.

**2) test\_semantic\_errors.mj** – тестира синтаксне грешке које могу да се јаве унутар улазних микројава фајлова и приказује успешно извршен опоравак од тих грешака.

**3) program.mj** – представља јавни тест са сајта предмета (test301.mj), који тестира већину граматичких смени које наш парсер подржава и приказује успешно формирање извшној .obj фајла.