DOKUMENTACIJA

Programski prevodioci - predmetni zadatak

Osnovni podaci

Broj indeksa	Ime i prezime	Tema
SW28/2019	Katarina Komad	Skener i parser za HTML upotrebom ANTLR alata

Korišćeni alati

Naziv	Verzija
ANTLR - jar	4.7

Opis ANTLR alata

ANTLR (Another tool for language recognition) je alat koji izgeneriše parser i lekser za odabrani jezik, u ovom slučaju HTML, na osnovu date gramatike. Takođe, automatski gradi i stablo parsiranja za koje postoji lepa podrška za GUI prikaz.

Obezbeđena je i dobra podrška za obradu grešaka, gde se precizno ispiše koje pravilo nije zadovoljeno i na kojoj liniji.
Automatski generiše i tzv. *tree walkers* i *listener*-e koji se mogu koristiti za obilazak čvorova ako je potrebno izvršavati specifičan kod za određene vrste čvorova.

ANTLR je fleksibilan alat koji ima prostora za custom implementacije svakog od koraka analize odabranog jezika jer se izgenerišu vrlo jasne i ljudski čitljive Java klase (moguće podesiti i drugi odredišni jezik promenom parametra npr –Dlanguage=JavaScript).

ANTLR vs BISON

Najznačajnija razlika između Bison-a koji je korišćen na vežbama i ANTLR alata koji je obrađen u ovom projektu je vrste gramatike koje mogu da obrađuju.

Bison podrazumevano generiše LALR(Look Ahead Right Left to Right)parser,
dok ANTLR radi sa LL gramarikama
Dakle, ANTLR-u je potrebna gramatika koja nije levo rekurzivna, dok Bison nema problem sa njom, ali zato je mana što može doći do shift-reduce i reduce-reduce problema.

Evidencija implementiranog dela

Kako se tema direktno odnosi na skener i parser za HTML upotrebom ANTLR alata, oba ova dela su odrađena u potpunosti.

Implementirana je gramatika za markup jezik HTML. Za samu leksičku i sintaksnu analizu korišćena je defaultna implementcaija od ANTLR alata.

Obezbeđeni su test primeri za validne i nevalidne HTML fajlove u folderu tests.

Takođe, kreirana je i .exe shell skirpta koja ima mogućnost izbora željenog test fajla sa diska, koja zatim pokreće sintaksnu i leksičku analizu, i na kraju otvara lep GUI prikaz AST stabla parsiranja. Sve što je potrebno uraditi je pokrenuti startAnalisys.exe i iskočiće prozor za izbor test fajla koji mora imati ekstenziju html.

Dodatno je započeta je semantička analiza, međutim zbog neočekivanih nepoklapanja verzija Jave, semantička analiza je ostavljena i opisana kao ideja za nastavak i dalji razvoj

Detalji implementacije

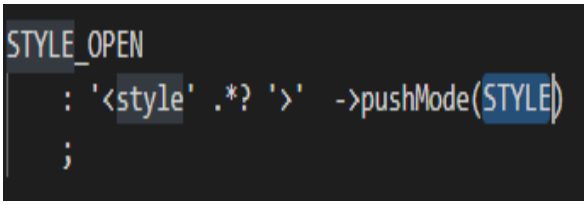
Neki od najbitijih elemenata HTML jezika koju su omogućeni i opisani u gramatici:

- Tag
- Atributi unutar tagova
- Skripta
- Stil
- Komentar

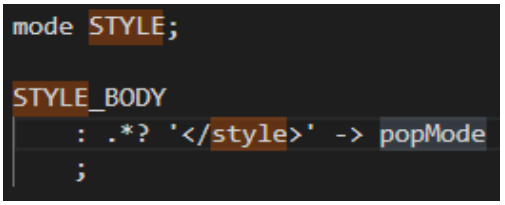
Opis dodatnih mogućnosti u okviru GRAMATIKE

1. PuhsMode, PopMode i mode

Gramatika leksera je sastavljena od pravila koja se mogu dalje razlagati na više *modova.*
Za ulaz u željeni mode koristi se metoda → pushMode(željeni_mode) sa slika1
A za izlaz iz trenutnog i povratak u prethodni mode → popMode sa slika2
Za deklaraciju novog moda koristi se ključna reč mode



1. Slika



2. Slika

2. Channel(HIDDEN)

Token koji se prebaci u ovaj kanal se praktično samo preskače, ali za razliku od →skip moguće mu je pristupiti kasnije po potrebi. Primer upotrebe imamo na slika3

3. Fragment

Fragmenti su ponovno upotrebljivi delovi pravila leksera, koji ne mogu biti samostalni tokeni

```
TAG_WHITESPACE
    : [ \t\r\n] -> channel(HIDDEN)
    ;
```

3. Slika

```
fragment HEXCHARS
    : '#' [0-9a-fA-F]+
    ;
```

4. Slika

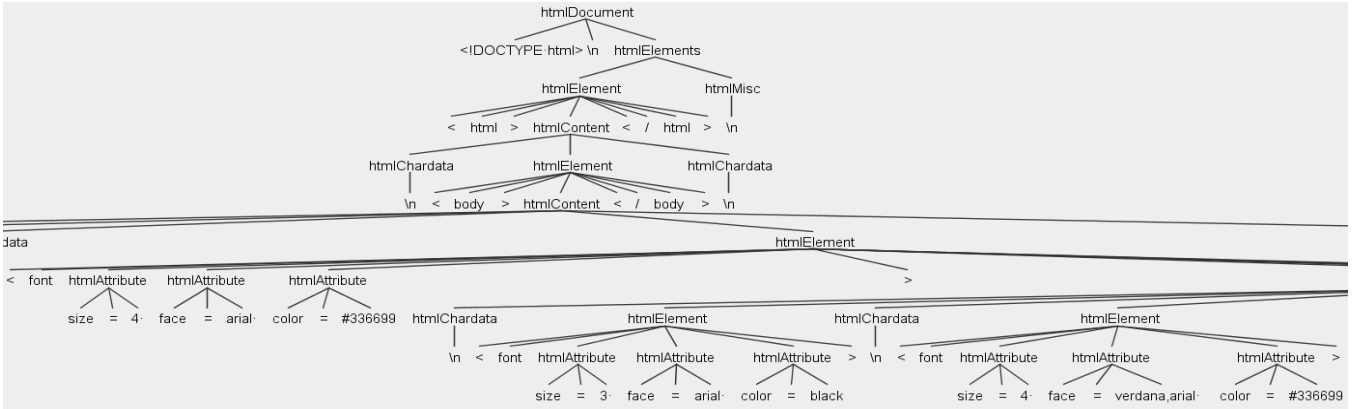
Parser → primer jednog pravila:

```
htmlElement
    : TAG_OPEN TAG_NAME htmlAttribute*
    | (TAG_CLOSE (htmlContent TAG_OPEN TAG_SLASH TAG_NAME TAG_CLOSE)? | TAG_SLASH_CLOSE)
    | script
    | style
    ;
```

5. Slika

Stablo parsiranja

Primer izgenerisanog stabla parsiranja za test primer *attvalues.html*



6. Slika

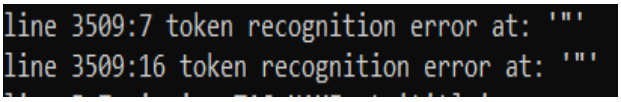
TEST FAJLOVI:

Test fajlovi su organizovani u 2 foldera: valid i invalid

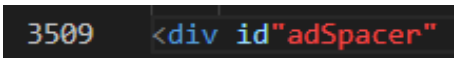
U valid folderu se nalazi 17 pravilno napisanih html fajlova različitih dužina i kompleksnosti. Za pojedinačno pokretanje svakog od tih fajlova, analiza se uspešno završi bez prijavljenih grešaka, s toga se može smatrati da je gramatika pravilno konstruisana.

Invalid folder sadrži 5 gramatički nepravilno napisanih fajlova.
Primeri prijave grešaka:

1. Primer nepravilnog atributa unutar taga (slika7 i slika8) u *invalid/aljazeera_com.html*.



7. Slika



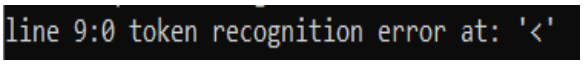
8. Slika

2. Primer nepravilnog imena tokena (slika9 i slika10) u fajlu *invalid/style1.html*

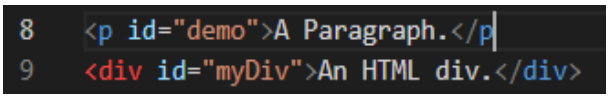
9. Slika

10. Slika

3. Primer nepravilnog zatvaranja taga (slika11 i slika12) u fajlu *invalid/script1.html*



11. Slika



12. Slika

Ideje za nastavak

1. Semantička analiza

Glavni prostor za unapređenje ovog projekta bi bilo uvođenje semantičke analize.

Prilikom generisanja koda za lekser i parser u Javi (u našem slučaju), moguće je navesti opciju i za generisanje *Visitor-a* i *Listener-a.* Oni su izuzetno korisni jer omogućavaju da se eksplicitno razdvoji gramatika od koda vezanog za aplikaciju i time olakšavajući čitljivost i smanjuje međuzavisnost.

Svaki Listener sadrzi metodu koja se poziva pri ulazu i pri izlazu iz pravila. Potrebno bi bilo u okviru Listener-a definisati proveru semantičke ispravnosti strukture fajla (npr da li se vrednost otvarajućeg taga poklapa sa vredošću zatvarajućeg taga) i takođe rekaciju na greške.

Takođe, moguće je izvršavanje i poziv listener-a tokom parsiranja, umesto da se čeka “šetnja” kroz stablo parsiranja (*tree walker walks*), međutim

Druga ideja za implementaciju semantičke analize je upotreba paketa *org.antlr.symtab*.
To je paket koji sadrži generičku tabela simbola, međutim potrebno detaljno osmisliti sturkturu takve tabele kako bi se na što jednostavniji način proverila ispravnost semantike jednog html fajla.

2. Vizualizacija - antlrworks/custom

Iako u okviru samog ANTLR-a postoji podrška za GUI prikaz stabla parsiranja,
za ranije verzije je postojao i poseban IDE *antlrworks* koji kombinuje editor za gramatiku sa intepreterom i vizualizacijom za bržu i lakšu izradu gramatika.

Ideja je proširiti prikaz izgenerisanog stabla da bude prijamčiviji ljudskom oku i da bude omogućena direktna manipulacija same gramatike.

3. Uvođenje sinatksne i leksičke analize javascript-a i css-a unutar tela html taga

Poslednja ideja nije toliko specifična za izabrani domen, ali da bi se ovaj analizator html fajlova smatrao kompletnim potrebno je omogućiti i analizu *javascript* i *css* delova koda unutar html tagova, koji će se neminovno pojavljivati u datim fajlovima.

Literatura

Tutorijal: https://tomassetti.me/antlr-mega-tutorial/
Baza za gramatiku: https://github.com/antlr/grammars-v4/tree/master/html
Test fajlovi: https://github.com/antlr/grammars-v4/tree/master/html/examples
Html tagovi: https://www.w3schools.com/TAgs/ref_attributes.asp
Symtab: https://github.com/antlr/symtab
Antlrworks: https://www.antlr3.org/works/