Programski prevodioci: Vežbe 1

Sadržaj

1. Uvod

Na vežbama iz predmeta Programski prevodioci će se razvijati kompajler. Vežbe će se sastojati iz dva dela. U prvom delu vežbi asistenti pokazuju kako se razvija neki deo kompajlera, a u drugom delu studenti treba da to implementiraju.

Na ovim vežbama biće prikazani regularni izrazi, kao i njihova upotreba u implementaciji skenera.

1.1. Alati potrebni za vežbe

Svi alati koji će biti korišćeni na ovom predmetu su slobodni softver.

Slobodni softver podrazumeva da korisnik ima 4 osnovne slobode:

1. Slobodu da program koristi u bilo koje svrhe.
2. Slobodu da izučava kako program radi i modifikuje ga, što podrazmeva dostupnost izvornog koda programa.
3. Slobodu da distribuira originalne verzije programa.
4. Slobodu da distribuira svoje modifikovane verzije programa.

Spisak korišćenih alata:

- Flex
- Bison
- GCC
- GNU make

- Tekst editor po izboru

Svi gore navedeni alati su dostupni u repozitorijumima popularnih GNU/Linux distribucija.

Alternativno, moguće je instalirati Ubuntu distribuciju sa sajta katedre gde su svi potrebni alati već instalirani.

2. Skener

Flex je program koji generiše leksički analizator, odnosno skener. Flex čita ulazna pravila i kao izlaz vraća kod koji implementira lekser u programskom jeziku C. Nastao je 1987. godine kao slobodna verzija programa Lex. Dostupan je pod modifikovanom BSD licencom, a njegov originalni autor je Vern Paxson.

Zadatak leksičkog analizatora je skeniranje sekvence tokena u ulaznom tekstu i pronalaženje određenih šablona. Kada se šablon prepozna, izvršava se zadata akcija. Šabloni se zadaju u vidu regularnih izraza.

Primer jednostavnog Flex programa:

```
 /* Deo za definicije */
%{ ①
    #include <stdio.h>
    int brojac;
%}
%% ②
 /* Deo za pravila */
[0-9]+  { brojac++; } ③
%% ②
 /* Deo za proizvoljan C kod */
int main() { ④
    yylex();
    printf("%d", brojac);
    return 0;
}
```

① U okviru `%{` i `%}` delimitiranog bloka može se navesti proizvoljan C kod koji će se nepromenjen ubaciti na vrh izgenerisane datoteke. Koristi se za dodavanje zaglavlja i definisanje globalnih promenljivih.

② Flex program se sastoji iz 3 sekcije, delimitirane karakterima `%%`. Prva sekcija predstavlja deo za definicije, druga sekcija predstavlja deo za pravila, dok treća omogućava navođenje proizvoljnog C koda koji će biti dodat na kraj izlazne datoteke.

③ Regularni izraz `[0-9]+` prepoznaje sve celobrojne dekadne vrednosti. Kada se ovo pravilo prepozna, izvršava se akcija navedena između vitičastih zagrada. Akcija može biti proizvoljan kod u programskom jeziku C.

④ Funkcija `main()` startuje leksičku analizu pozivom funkcije `yylex()`, a potom ispisuje koliko puta se na ulazu pojavio nerazlomljeni dekadni broj.

3. Regularni izrazi

Regularni izrazi omogućavaju pretraživanje šablona u ulaznom tekstu.

Flex koristi sebi specifičnu sintaksu za zadavanje regularnih izraza, ali je ona u velikoj meri slična uobičajenoj ERE sintaksi koja je definisana u IEEE POSIX standardu.

Regularni izrazi imaju široku primenu i ne koriste se samo za implementaciju kompajlera. Na primer, tekst editor koji koristite verovatno poseduje mogućnost pretrage teksta pomoću regularnih izraza. Takođe, ako pravite bilo koji program gde korisnik ima mogućnost unosa podataka u formu, validaciju unosa je pogodno implementirati pomoću regularnih izraza.

3.1. Sintaksa regularnih izraza

Specijalni karakter	Značenje
`.`	Jedan, bilo koji karakter *osim* karaktera za novi red (`\n`).
`[abc]`	Jedan, bilo koji karakter iz skupa karaktera navedenih u uglastim zagradama. Karakter `-` ima specijalno značenje u uglastim zagradama — predstavlja raspon karaktera. Na primer, `[a-f]` ima isto značenje kao i `[abcdef]`.
`[^abc]`	Jedan, bilo koji karakter *osim* karaktera navedenih u uglastim zagradama.
`?`	Kvantifikator `0..1`.
`*`	Kvantifikator `0..n`.
`+`	Kvantifikator `1..n`.
`{x,y}`	Kvantifikator—minimalno `x` a maksimalno `y` ponavljanja, gde su `x` i `y` pozitivne celobrojne vrednosti.
`()`	Grupisanje.
`|`	Alternativa.
`""`	Svi karakteri unutar dvostrukih navodnika gube svoje specijalno značenje.
`\`	Karakter neposredno iza `\` gubi svoje specijalno značenje.

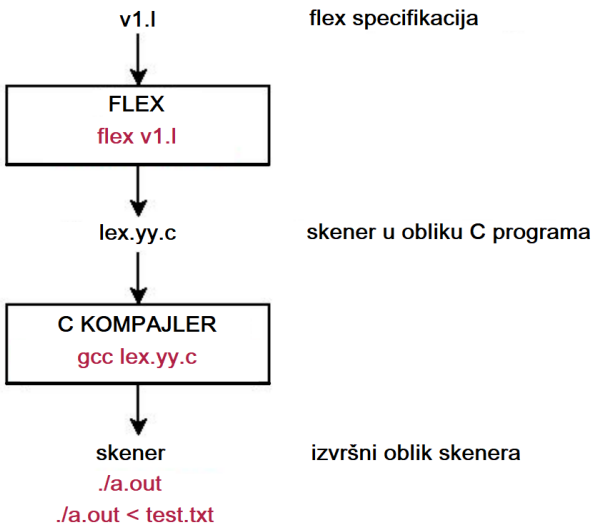
Prioriteti:

1. Kvantifikatori se odnose samo na 1 krakater koji se nalazi neposredno ispred samog kvantifikatora.

2. Alternativa `|` se odnosi na sve što se nalazi levo i desno od `|`, a ne samo na 1 karakter neposredno levo i neposredno desno.
3. Moguće je koristiti grupisanje `()` kako bi se gore navedeni prioriteti promenili.

Gore navedena tabela nije kompletna, već sadrži samo najčešće korišćene izraze. Za pregled kompletne liste izraza ukucati `info flex` u terminalu i otvoriti sekciju "*Patterns*".

4. Kompajliranje, pokretanje i testiranje



Kako bismo izbegli da prilikom testiranja svaki put ručno kucamo test primer, moguće je test primer iskucati u nekoj datoteci, a zatim upotrebom redirekcije proslediti tu datoteku na standardni ulaz programa:

```
./a.out < test.txt
```

5. Zadaci

5.1. Zadatak 1

Napraviti skener koji prepoznaje:

1. Cele brojeve. Na primer: `+45645`, `-4356`, `642642`, `+1`.
2. Heksadecimalne brojeve—započinju cifrom `0`, iza koje sledi malo ili veliko slovo `x`, a zatim najmanje jedna, a najviše 4 heksadecimalne cifre. Na primer: `0x7A3F`, `0X1234`, `0xf`, `0x123`, `0xffce`.
3. Realne brojeve (tačka i bar jedna cifra pre tačke su obavezni). Na primer: `2.345`, `0.0`, `5.`, `123.456`, `+123.456`, `-123.456`.

   Dodatno: realni brojevi su i brojevi oblika `12.345e+123`, `5.e-4`, `-1.23E+04` gde eksponencijalni deo

može imati 1-3 cifre.

4. Ključnu reč `break`, ali case-insensitive. Na primer: `break`, `BrEaK`, `BREAK`.

5.2. Zadatak 2

Napraviti skener koji prepoznaje single-line komentar (`//`) i izbacuje ga iz koda. Za testiranje se može koristiti `test2.c`. Izlaz treba da bude početni kod, sa izostavljenim linijama koje počinju stringom `//`.

5.3. Zadatak 3

Napraviti skener koji u ulaznom tekstu temperaturu izraženu u Farenhajtima prepravlja u temperaturu izraženu u Celzijusima. Ostatak teksta treba da ostane isti.

Formula za konverziju temperature: `(°F - 32) x 5/9 = °C`.

Na primer, tekst:

```
Normalna temperatura ljudskog tela je 98F.
Temperatura od 50F je daleko ispod proseka za mesec maj.
Voda kljuca na 212F.
```

treba da se prevede u:

```
Normalna temperatura ljudskog tela je 36C.
Temperatura od 10C je daleko ispod proseka za mesec maj.
Voda kljuca na 100C.
```

5.4. Zadatak za domaći

Rešiti prepoznavanje C blok komentara. Na primer: `/* komentar */`.

Testirati rešenje i sa sledećim primerima:

- `/* komentar * komentar */ nije komentar */`
- `/* komentar **/ nije komentar */`

5.5. Dodatni zadaci za vežbanje

Dodatne zadatke za vežbanje možete pronaći na linku: https://regexcrossword.com/