

Skripta za usmeno ispitivanje 3. i 4. laboratorijske vježbe

Što je semantička analiza i navedi primjer?

- Semantička analiza povezuje faze analize izvornog i ciljnog programa. Ona provjeri npr. zbrajaju li se varijable istih tipova, je li predan dobar broj parametara funkciji itd.

Gdje se u 4. vježbi spremaju parametri funkcija i povratne adrese?

- Parametri na stog, povratna adresa u registar R6.

Koja je razlika između deklaracije i definicije?

- Deklaracija funkcije je u biti zaglavlje funkcije. Definicija funkcije je kompletna funkcija (zaglavlje + tijelo). Deklaracija daje samo informacije potrebne u pozivu funkcije: ime funkcije, tip vraćene vrijednosti i tipove parametara. Definicija uključuje sve informacije o funkciji, uključujući i način rada (naredbe koje je čine). Definicija je implicitno i deklaracija.

Što sve semantički analizator gleda kod funkcije, npr. $f(1,2)$?

- Odgovaraju li deklarirani tipovi parametara funkcije tipovima parametara koji su predani funkciji i je li točan broj parametara.

Navedite primjer kad semantički analizator izbaci grešku.

- Na primjer kad se pokuša obaviti operacija $char = char + int$ (upute za lab – 48.str)

Koji je ulaz u sintakсни analizator, a koji u semantički i koja je njihova razlika?

- Ulaz u sintakсни analizator je niz uniformnih znakova sa standardnog ulaza, a u semantički generativno stablo.
- Sintakсни analizator konstruira generativno stablo
- Na izlazu semantičkog analizatora ne smije biti ništa ako je program ispravan, a ako nije mora ispisati produkciju u kojoj je otkrivena prva semantička pogreška.

Što su nasljedna i izvedena svojstva i gdje ih koristimo?

- U atributnoj prijevodnoj gramatici završnim i nezavršnim znakovima dodjeljuju se svojstva i pravila računanja tih svojstava. Pravilima računanja svojstava određen je i smjer njihovog prijenosa po sintaksnom stablu. Ako se njihove vrijednosti prenose od dna prema vrhu onda se svojstva nazivaju **izvedena**, u suprotnom se nazivaju **nasljedna svojstva**.
- Koristimo ih za opis semantičkih pravila

Što se dogodi ako se zbrajaju float i int?

- Int se pretvori u float i zbroje se kao floatovi

Koja je razlika kada u FRISCUu želimo spremi polje intova i polje charova?

- Spremimo prvi element na neku adresu, a ostale elemente indeksiranjem s odmakom
- Budući da je char 8 bita, a int 32 bita onda koristimo indeksiranje s odmakom +1 za char odnosno +4 za int.

Što se sve provjerava kod poziva funkcije?

- Provjerava se: je li definirana, odgovara li ono što return vraća povratnom tip funkcije, odgovara li broj argumenata onom u deklaraciji i odgovaraju li su parametri koji su u pozivu funkcije po tipu argumentima.

Kako smo odredili koja je važeća vrijednost varijable? Pojasniti djelokruge i koliko ih ima?

- Postoje 3 pravila djelokruga: statički djelokrug bez ugniježđenih procedura, statički djelokrug ugniježđenih procedura i dinamički djelokrug.
- U vježbi se koristi statički djelokrug bez ugniježđenih procedura, ali podržava ugniježđene blokove
- Pročitati na 42. i 43. stranici uputa za labose (Upute za izradu laboratorijskih vježbi iz predmeta Prevodenje programskih jezika) objašnjenje kako odrediti važeću vrijednost.

Opisati naredbe CALL, RET i što je PC.

- PC je programsko brojilo koje ima spremljenu memorijsku adresu naredbe koja se **sljedeća** treba izvesti.
- Prilikom poziva potprograma, tj. izvođenja naredbe CALL, povratna adresa iz PC-a se sprema na stog kako bi nakon izvođenja potprograma RET mogao dohvatiti sa stoga povratnu adresu i skočiti na nju.

Kako se prenose parametri u potprogram u FRISCUu?

- Tri najčešća načina prijenosa su preko registara, preko stoga i preko fiksnih memorijskih lokacija.
- U vježbi se zbog jednostavnosti koristi stog i registar R6 za povratnu vrijednost.

Navesti primjere leksičkog, sintaksnog i semantičkog pravila.

- Leksičko: ako, onda i inače su ključne riječi. Identifikatori moraju počinjati sa slovom...
- Sintakšno: da u naredbi zbrajanja mora biti + između dva identifikatora ili konstante
- Semantičko: nije moguće zbrajati varijable tipa int i char

Koja je razlika između JUMP i CALL?

- JUMP skače na labelu ili zadanu adresu. CALL mora spremiti prvo povratnu adresu na stog pa tek onda skočiti na labelu potprograma.

Koliko razina međukoda imamo?

- Općenito, postoje 3 vrste međukoda: viši, srednji i niži
- Na laboratorijskoj vježbi nemamo niti jednu razinu međukoda

Opisati djelokrug varijabli na primjeru.

- ```
int main(void) {
 return f(); // greška
}
int f(void) {
 return 0;
}
```
- Funkcija f nije deklarirana prije korištenja.

### Koja je razlika između generativnog i sintaksnog stabla i koje je praktičnije koristiti u semantičkoj analizi (i zašto)?

- Koristi se generativno stablo, sintakсни analizator obično izravno generira sintakšno stablo, ali se ono može dobiti i dodatnim obilaskom generativnog stabla.

### Što su ulaz i izlaz za generator koda (4. labos)?

- Ulaz je isti kao i u semantički analizator, tj. generativno stablo ppjC programa.
- Izlaz je generirani mnemonički program za procesor FRISC u datoteku a.frisc u istom direktoriju u kojem se nalazi sam generator koda.

### Kako zapisati strukture?

- Strukture ne možemo spremati u registre, nego na neku memorijsku lokaciju. To onda moramo zapisati u tablici

### Po čemu su povezani semantički analizator i generator koda?

- Imaju isti ulaz, a semantički analizator dodatno provjerava kod prije generiranja ciljnog programa.

### Kako se pronalaze globalne varijable u FRISCU?

- Preko labela

### **Objasniti kako radi semantički analizator?**

- Semantička analiza povezuje faze analize izvornog i sinteze ciljnog programa. Semantički analizator određuje značenje sintaksnim cjelinama, provjerava semantička pravila pridružena sintaksnim cjelinama i priprema podatke generatoru međukoda ili ciljnog programa. Ovisno o izboru strukture međukoda, semantički analizator priprema sve podatke potrebne za generiranje atributnog sintaksnog stabla, niza naredbi apstraktnog stogovnog stroja ili niza slijednih naredbi.

### **Što je lijena evaluacija?**

- Dijelovi programa se prevode tek kada su potrebni, ne prije.

### **Kako se provjerava izlaz generatora?**

- Pokretanjem programa

### **Kako spremi 15 podataka u 7 registara?**

- Korištenjem stoga. Kada koji treba dohvatiti se u registar naredbom POP.

### **Kako implementirati množenje u FRISCu?**

- MNOZI ADD R0, R2, R2  
SUB R1, 1, R1  
JR\_NZ MNOZI
- Množenje uzastopnim pribrajanjem: R1 je brojač, umnošku R2 pribraja se R0 u kojem je fakt(n-1) (\_NZ not zero)

### **Koja je razlika u pozivanjima potprograma u assembleru i višim programskim jezicima?**

- U višim programskim jezicima najčešće ne moramo voditi računa o kako se potprogram poziva i kako se vraćamo iz njega, gdje se sprema povratna adresa, kako se prenose parametri i vraćaju rezultati iz potprograma, gdje se čuvaju lokalne adrese itd.
- U assembleru se koriste naredbe CALL i RET za poziv i povratak iz potprograma

### **S koje tri naredbe uvijek počinje program u FRISCu?**

- MOVE 40000, R7  
CALL F\_MAIN  
HALT

## Kako realizirati while i for petlju u FRISCu?

|                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>while ( i &lt; j ) {<br/>    i++ ;<br/>}</pre>                                                                                                                                                                                                                                               | <pre>for (brojac = 1; brojac &lt;= 5; brojac++)<br/>    naredbe;</pre>                                                                                                                                                                                                                                                                                                                                      |
| <pre>LABELA CMP R0, R1<br/>        JP_UGE DALJE<br/>        ADD R0,1,R0<br/>        JP LABELA<br/>DALJE ....</pre> <p>Usporedi registre u kojima su i i j. Ako je i veće ili jednako j ide dalje, ako ne izvest će se ADD i skočiti opet natrag na LABELA. (_UGE – Unsigned Greater or Equal)</p> | <pre>        MOVE 1, R0<br/>        MOVE 5, R1<br/>LAB     CMP R0, R1<br/>        JP_UGT DALJE<br/>        //naredbe koje su u tijelu//<br/>        ADD R0,1,R0<br/>        JP LAB<br/>DALJE  ...</pre> <p>U registre R0 i R1 spremi 1 i 5 i njih uspoređuje. Ako je brojač veći od 5 skače na DALJE, a ako nije izvršava naredbe i povećava brojač, pa se vraća na LAB. (_UGT – unsigned greater than)</p> |

## Koje pravilo djelokruga smo koristili?

- Statičko bez ugniježđenih procedura.

## Što znači naredba ORG u FRISCu?

- To je pseudonaredba koja označava početak punjenja strojnog koda u memoriji. Ako se ispusti pretpostavlja se da kod kreće od nule. Adresa mora biti djeljiva sa 4 (jer svaka naredba nosi 4 memorijske lokacije).
- Npr. `ORG 100  
 ADD ...  
 STORE ...  
 U memoriji će biti na adresi 100 upisano ADD, a na adresi 104 STORE.

## Navesti elemente strukture generatora ciljnog programa.

- Generator ciljnog programa definira se ulazom i izlazom, a čine ga
  - postupci izrade adrese,
  - izbora naredbe,
  - izbora redoslijeda izvođenja naredbi
  - i dodjele registara postupcima.

**Ako imamo samo deklariranu funkciju koja nije definirana, je li semantičkom analizatoru to dovoljno da ispita je li poziv funkcije ispravan?**

- Da, jer je dovoljno da analizator zna kakav tip argumenata i broj treba biti, te povratni tip.

**Kako bi u FRISC-u izveli čitanje podatka na koji pokazuje pointer?**

- Učitamo vrijednost pointera s neke adrese, i onda napravimo još jedan load koji koristi tu učitano vrijednost kao adresu za čitanje.

**Što su opisnici procedura?**

- Svaka aktivirana procedura sprema podatke u svoj opisnik. Naredbe potpore izvođenju ciljnog programa povezuju aktiviranu proceduru i njen opisnik te osiguraju memoriju za opisnik, zapišu podatke u opisnik prije izvođenja procedure, zapišu podatke u opisnik nakon što završi izvođenje procedure i oslobode memoriju. Opisnik ima 7 polja, među kojima aktualne vrijednosti ulaznih i izlaznih parametara procedure, kazaljka na opisnik pozivajuće procedure, kazaljka nelokalnih imena, stanje racunala i sadržaj registara, lokalni i privremeni podaci.

**Dodatna neodgovorena/neprovjerena pitanja:**

Kod na ploči koji je bio ovako napisan:

```
int main(){
 return 0;
}
```

i pitao je jel tu sve okej, (nije, fali void u zagradi ili parametri), množenje i dijeljenje u friscu kako bi se ostvarilo, što predstavlja `_l` izraz u trećem labosu, što je short circuit evaluacija (podvrsta lijene evaluacije rekao je Pavlič), kako smo prenosili parametre u četvrtom labosu (pomoću stoga, pa je potpitanje bilo kako se još može - registrom ili samo upišemo rezultate u neku memorijsku lokaciju)

Kako povezujemo labele varijabli s imenom identifikatora u izvornom programu?

Kad se u deklaraciji/definiciji zapisuju parametri u tablicu znakova?

Kako ostvariti rekurziju u friscu?

Što znači ključna riječ void u ppjc-u?

Kako ostvariti rekurziju u friscu?

Koliko bytova smo koristili za char, 1 ili 4 tako nešto... i nešto sa float i int zbrajanjima, i implicitnim pretvorbama nešto sitno mi se čini.

***Napomena: preporučljivo je ipak nakon učenja ovih pitanja pročitati uputu za labos. Puno je jasnije o čemu se radi. Nadopunjavajte i popravljajte skriptu za buduće generacije.***

**Sile**