

ZADAĆA 5, ZADATAK 3

Belmin Selimović 18393

Ahmed Pašić 18527

Imamo dvije klase: Main i Instruction. U klasi Instruction se nalaze atributi za svaki dio instrukcije, atribut koji govori kojeg je tipa instrukcija te string koji sadrži cijelu instrukciju(njega smo koristili radi lakšeg zapisivanja u odredišnu datoteku).

Pretpostavili smo da je format instrukcija takav da se iza eventualne labele nalazi jedan razmak, kao i iza naziva instrukcije, a da su parametri odvojeni sa jednim zarezom i jednim razmakom. Dvotačka kod labele je obavezna inače program neće ispravno raditi.

Primjer formata:

- ADD R1, R2, R3
- label: SUB R3, R4, R5
- BEQ R1, R2, label1

Metoda „dajSekvencuIzDatoteke“ čita sadržaj datoteke te kreira listu čiji su elementi instrukcije(tipa Instruction).

Metoda „dajInstrukcijeZadrške“ pronalazi sve instrukcije grananja i za svaku određuje njenu instrukciju zadrške, ukoliko je to moguće.

Metoda „zapisInstrukcijeZadrške“ zapisuje u datoteku instrukcijeZadrške.txt instrukcije grananja zajedno sa svojim instrukcijama zadrške. Za one instrukcije grananja za koje nije pronađena instrukcija zadrške, umjesto instrukcije zadrške piše „nema instrukciju zadrške“.

Metoda „dajInstrNakonInstrGrananjaZadrška“ provjerava da li postoji instrukcija zadrške tako što gleda slučaj „C“ iz materijala vježbe. Odnosno provjerava da li sljedeća instrukcija može biti instrukcija zadrške.

Metoda „dajInstrIznadInstrGrananjaZadrška“ provjerava slučaj A i ona se prva izvršava jer je slučaj A idealan. Prvo se provjerava da li je tekuća instrukcija grananja ujedno i prva instrukcija u datoteci. Ukoliko jest, ova metoda nam ne daje instrukciju zadržke jer ne postoji nijedna instrukcija iznad te se prelazi na preostala dva slučaja. Npr. za sekvencu:

```
add r4, r5, r6
beq r2, r3, LABELA
LABELA sub r3, r3, r3
```

Instrukcija zadržke za drugu instrukciju po redu će biti prva instrukcija. Za sekvencu:

```
add r2, r5, r6
beq r2, r3, LABELA
LABELA sub r3, r3, r3
```

instrukcija zadržke neće biti prva instrukcija zbog njenog odredišnog registra kojeg koristi druga instrukcija, odnosno instrukcija grananja. Ovdje će se zbog slučaja B za instrukciju zadržke uzeti treća instrukcija.

Metoda „dajInstrSaOdredistaGrananja“ pokriva slučaj B. Ona će dati instrukciju zadržke samo onda kada to ne uradi prethodna metoda, kao što je prikazano na prethodnom primjeru.

Metoda „dajInstrukcijuIzStringa“ za odgovarajući red pročitani iz datoteke razdvaja parametre na način da uzima ime instrukcije, registre, ukoliko postoje labela te konstante. Ova metoda vraća instrukciju tipa Instruction.

Metoda „dajRegistarLwSw“ vraća bazni registar ukoliko se radi o metodi LW ili SW

Metoda „provjeriPotrebuInstrZadrške“ prolazi kroz instrukcije i ukoliko nema niti jedne instrukcije grananja govori nam da nije potrebna instrukcija zadržke.

Projekat smo radili zajedno preko githuba. Dogovarali smo se međusobno oko implementacije i raspoređivali poslove. Commite i odrađeni posao možete vidjeti

na githubu ukoliko Vam je tako lakše pregledati, a link od githuba je(program je također poslan i na Zamger) : https://github.com/ahmedpasic/RAZadaca5_zad3

Testiranje načina „C“ traženja instrukcije zadržke, instrukcija nakon instrukcije grananja :

```
ADD R3, R2, R1
SUB R1, r2, r4
beq R1, R2, label4
SUB R5, R5, R3
LW R5, 5(R6)
label4: ADDI R3, R4, 1
ADDI R1, R6, 5
```

U ovom slučaju imamo instrukciju zadržke
BEQ R1, R2, LABEL4----->SUB R5, R5, R3

```
ADD R3, R2, R1
SUB R1, r2, r4
beq R1, R2, label4
SUB R1, R5, R3
LW R5, 5(R6)
label4: ADDI R3, R4, 1
ADDI R1, R6, 5
```

Nema instrukcije zadržke za slučaj C

```
LW R1, 5(R3)
label4 : ADD R2, R4, R5
beq R6, R3, label4
```

Nema instrukcije zadržke za slučaj C