

Simple Merge Return Pass

Projekat u okviru kursa Konstrukcija kompilatora

Konstantin Klima 476/2018

Matematički fakultet, Univerzitet u Beogradu

31.10.2025.

Uvod

- *mergereturn* - standardni LLVM optimizacijski pass
- Spaja više return ili unreachable naredbi u jednu
- Cilj je poboljšanje čitljivosti generisanog IR-a kao i daljih analiza i optimizacija
- Pripoznajemo tri slučaja koja treba obraditi:
 - Više unreachable instrukcija - nastaju kao posledica exceptiona-a, instrukcija koje prekidaju tok izvršavanja ili prethodnih optimizacija
 - Više return instrukcija bez povratne vrednosti (void)
 - Više return instrukcija sa povratnom vrednošću različitom od void

Algoritam

Algorithm Simple Merge Return Pass

$unrBlocks \leftarrow$ niz blokova terminiranih unreachable instrukcijom
 $retBlocks \leftarrow$ niz blokova terminiranih return instrukcijom

za **svaki** *basicBlock* u funkciji *F* :

ako je terminirajuća instrukcija za *basicBlock* unreachable **onda**
dodaj *basicBlock* u *unrBlocks*

ako je terminirajuća instrukcija za *basicBlock* ret **onda**
dodaj *basicBlock* u *retBlocks*

ako je $|unrBlocks| > 1$ **onda**

kreiraj novi blok *unrBlock* sa unreachable instrukcijom

za svaki *basicBlock* u *unrBlocks* :

zameni terminirajuću instrukciju u *basicBlock* sa grananjem na *unrBlock*

ako je $|retBlocks| > 1$ **onda**

kreiraj novi blok *retBlock* sa return instrukcijom

ako je tip povratne vrednosti *F* različit od void **onda**

kreiraj novu phi instrukciju

izmeni return instrukciju u *retBlock* da učitava vrednost iz phi

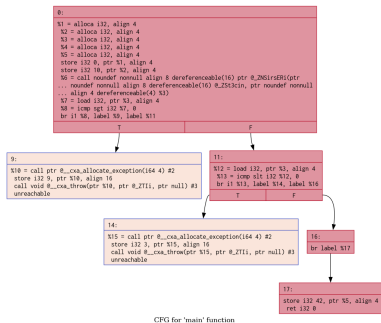
za svaki *basicBlock* u *retBlocks* :

zameni return instrukciju u *basicBlock* sa grananjem na *retBlock*

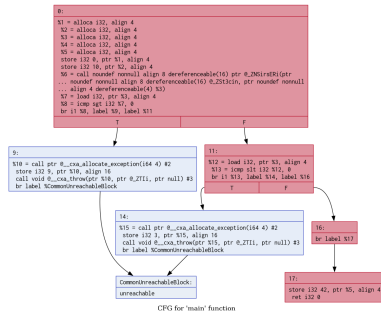
Napomene

- Implementacija je testirana korišćenjem LLVM New Pass Managera - dodata je i legacy verzija koja zbog ograničenja lokalnog okruženja nije mogla da se testira.
- IR za primer sa višestrukim return instrukcijama je ručno izmenjen jer je frontend emitovao već optimizovan IR sa spojenim return instrukcijama (ovo nije bio slučaj kod unreachable primera)

Primer 1: više Unreachable Instrukcija

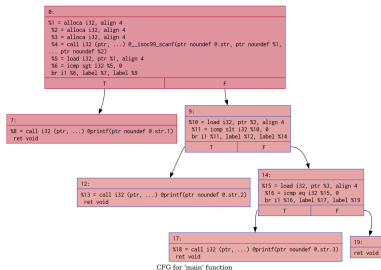


(a) Pre optimizacije

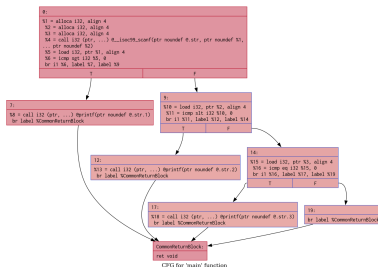


(b) Posle optimizacije

Primer 2: više Return Instrukcija

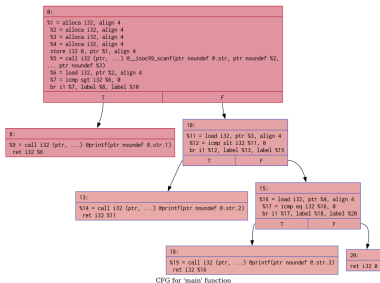


(c) Pre optimizacije

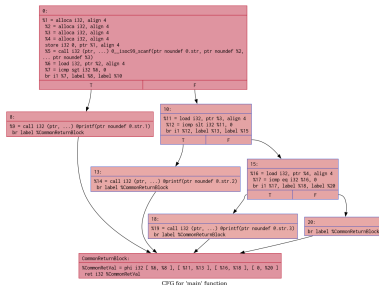


(d) Posle optimizacije

Primer 3: više Return Instrukcija sa povratnom vrednošću



(e) Pre optimizacije



(f) Posle optimizacije

Zaključak

- Pri implementaciji neophodno je obratiti pažnju na različite tipove povratnih vrednosti i obezbediti da se propisno propagiraju pri zameni return instrukcija
- Optimizacija treba da obradi i unreachable instrukcije
- Kako ova optimizacija menja CFG, neophodno je ponoviti prethodno obavljene analize koje se oslanjaju na CFG