Intra Block Allocation

კარგად მუშაობს თუ ბეიზიქ ბლოქში თითოეულ ცვლადს ბევრჯერ ვიყენებთ.

თუ მაგალითად ბლოკში ცვლადს ერთხელ ვიყენებთ, მისი წინასწარ რეგისტრში ამოტანა არაფერს გვიცვლის (არ აუმჯობესებს naive-ს). თუმცა ამას თუ ფუნქციის გამოძახება დაემატა ბლოკში, არათუ აუმჯობესებს, არამედ აუარესებს naive-ს, ვინაიდან call-ის დროს ამ ცვლადების ისევ მეხსირებაში დაბრუნება და მერე ისევ რეგისრებში ამოტანა გვიწევს.

ჩემი იმპლემენტაციის შემთხვევაში read write ზოგ პროგრამაში მცირდება, თუმცა move ინსტრუქციების რაოდენობა ყოველთვის მეზრდება (ამაზე უფრო დეტალურად ქვემოთ მიწერია).

Benchmark 1

naive: #reads: 158 #writes 113 #branches 61 #other 158

Program ib: #reads: 158 #writes 112 #branches 61 #other 325

პირველ ბენჩმარკში მარტო write გაუმჯობესდა 1-ით. ეს იმიტომ ხდება, რომ ib allocation ეფექტურია მაშინ, როცა ბეიზიქ ბლოქში თითოეულ ცვლადს ბევრჯერ ვიყენებთ, სხვა შემთხვევაში არაეფექტურა. თუ ისეა, რომ თითო ცვლადს თითოჯერ ვიყენებთ გამოდის, რომ იგივე შედეგს მოგვცემს რასაც ნეივი. **#other ინსტრუქციების რაოდენობა გაიზარდა, რადგან $s რეგისტრებში ვინახავ ცვლადებს, და მათზე მოქმედებებს ვატარებ $t რეგისტრებში. ამიტომ გვემატება move ინსტრუქციები. თუმცა მიუხედავად ამისა თუ read და write ტიპის ინსტრუქციების შემცირებას შევძლებთ move ინსტრუქციების გაზრდას არაუშავს, რადგან ბევრად უფრო სწრაფია. ასევე არც ისე რთულია move-ების მოშორება და პირდაპირ $s რეგისტრებზე მოქმედებები, მაგრამ თავიდან ასეთი იმპლემენტაცია გავაკეთე CodeGenerator-ის და აღარ შემიცვლია, რადგან უფრო მნიშვნელოვანი read write ტიპის ინსტრუქციების შემცირებაა.**

Benchmark 2

naive: #reads: 344 #writes 237 #branches 61 #other 220

Program ib: #reads: 220 #writes 203 #branches 61 #other 727

აქ read-იც და write-იც გაუმჯობესდა. რადგან აქ გვაქვს ბეიზიქ ბლოქები, რომლებშიც გვაქვს რამდენიმე ცვლადის ხშირი გამოყენება.

მაგალითად:

add, _2_a, _2_b,

_t5 assign, _2_a,_t5,

add, _2_a, _2_b, _t6

assign, _2_a, _t6,

add, _2_a, _2_b, _t7

assign, _2_a, _t7,

assign, _t8, 1,

brlt, _2_i, 5, _L5

მაგალითად _2_a უკვე რეგისტრში გვექნება ამ ბეიზიქ ბლოქში შემოსვლის დროს, და აღარ მოგვიწევს ყოველ ჯერზე მემორიდან წაკითხვა და ჩაწერა. **#other ინსტრუქციებზე აქაც იგივე. Move ინსტრუქციების დამატება იწვევს ზრდას.**