

## Bugs or features that are not operational

არ მაქვს იმპლემენტირებული global briggs allocation. -g ფლეგის დროს ვაკეთებ იგივეს რასაც -b გააკეთებდა.

## Mips Functions calling implementation

სტანდარტული მიდგომისგან განსხვავებით, \$s რეგისტრებს ინახავს caller და არა callee, რადგან ასეთი შეთანხმებით, ჩემს შემთხვევაში, უფრო მარტივი იყო intra block allocation-ს იმპლემენტაცია. დანარჩენი ისევეა როგორც სტანდარტულში.

**Callee** გამოყოფს სტეკში ადგილს ცვლადებისთვის და \$ra რეგისტრისთვის (ინახავს \$ra რეგისტრს). შემდეგ return-მდე ამოაქვს \$ra რეგისტრში შენახული მნიშვნელობა, თუ საჭიროა \$v0 რეგისტრში წერს დასაბრუნებელ მნიშვნელობას, მნიშვნელობას \$sp-ს ზრდის \$sp-ს ზრდის და ასრულებს jr \$ra ინსტრუქციას.

**Caller** ინახავს \$s რეგისტრებში ამოტანილ ცვლადებს ისევ მეხსიერებაში, გამოყოფს ადგილს პარამეტრებისთვის (\$a რეგისტრებს არ ვიყენებ) **Callee**-ს და იძახებს მას. კონტროლის დაბრუნების შემდეგ კი \$sp-ს ზრდის.

## Class design and general code structure

**\*კოდში ჩახედვას არ გირჩევთ\***

**Codegen** აგვარებს კოდის გენერაციას, არ არის დამოკიდებული რეგისტრის ალოკაციის ალგორითმზე. გადაცემა **AllocatorFactory** ტიპის ობიექტი. და სხვადასხვა **Allocator**-ის გადაცემისას აგენერირებს სხვადასხვა კოდს. ანუ იდეაში ისე უნდა ყოფილიყო, რომ Allocator ინტერფეისის გაუკუთებული სხვადასხვა იმპლემენტაციებს, codegen უკვე მზად მექნებოდა და შეხება აღარ მომიწევდა, მაგრამ აღმოჩნდა რომ არც ისე კარგად მქონდა ინტერფეისი მოფიქრებული და შეცვლა მომიწია intra block ალოკაციის დამატებისას. Global briggs ალოკაცია რომ დამეწერა მაგისტრის კიდე შეცვლა მიწევდა. სხვა მგონი საინტერესო არაფერი არ არის

## Design of data structures used to implement each register allocation scheme: naïve, block, Briggs.

**Naïve: Allocator.allocate()** აბრუნებს MemoryTable კლასის ობიექტს, სადაც წინასწარ განსაზღვრულია რომელი ცვლადი რა ოფსეტზე წერია, რეგისტრებში არცერთი ცვლადი არ გვაქვს.

**Intra block: Allocator.allocate()** აბრუნებს MemoryTable კლასის ობიექტს, სადაც წინასწარ განსაზღვრულია რომელი ცვლადი რა ოფსეტზე წერია, თუმცა თითოეული basic block-ის შემდეგ ვიძახებთ **Allocator**. () მეთოდს, რომელსაც რაღაც ცვლადებს რეგისტრებში აიტანს, და რაღაცებს მეხსიერებაში დააბრუნებს (შესაბამის mips ინსტრუქციებს CodeGenerator აგენერირებს ოღონდ, ეს უბრალოდ ეტყვის CodeGenerator-ს რაები ამოიტანოს წინასწარ ამ basicblock-ისთვის და რაები დააბრუნოს).