Register Allocation

Anurag Kashyap, Mia de Leon, Austin Kao

April 20, 2021

Description

In writing a compiler of the Tiger programming language, we previously generated an interference graph, which tells which temporaries interfere with each other. Using this, we are able to perform register allocation, where we assign temporaries to available registers.

Color

Our color module is meant to "color" each of the temporaries in the interference graph so that each temporary will have a "color" that represents the register the temporary will be in at run time. Temporaries that are live at the same time cannot have the same "color" and be stored in the register. We keep track of a list of the 18 available registers MIPS has, and if a register is available, we will assign a temporary to that register. Caller saved registers have priority over callee saved registers. We did not handle spilling, nor did we attempt to coalesce move instructions.

RegAlloc

The register allocation uses the color module as a subroutine to figure out which temporaries are assigned to which registers. Because we did not implement coalescing, our allocation does not remove any instructions in the list of instructions. It only modifies the instructions to specify the MIPS registers that will be used at run time.