# Framing The Register Allocation Problem

- Problem
  - Register access is significantly faster than memory access.
  - Use GPRs to reduce the number of memory loads and stores.
  - Assign a large number of program variables and temporary values to a small number of GPRs.
- Key idea
  - Generate “abstract” assembly code assuming an unbounded number of available virtual registers.
  - Perform register allocation i.e., map the virtual registers to the fixed finite number of GPRs, satisfying semantic correctness, architectural idiosyncrasies, and linkage protocol constraints.
- Key questions
  - Scope: Local, global, or interprocedural?
  - When: Static or dynamic?
- Key metrics
  - Size of generated code, execution speed of generated code, speed of allocation.

# Common Issues

- Machine idiosyncrasies
  - [Aliasing] Assigning a value to one register can affect the value of another, e.g., `%rax`/`%eax`/`%ax`/`%al` on x86.
  - [Register configurations] Register pairs, e.g., full-width multiplication.
  - [Miscellaneous] Destructive operations, condition flags.
- Pre-coloring
  - Forcing some variables to be assigned to particular registers, e.g., procedure linkage on x86.
- Problem complexity
  - Global register allocation is NP-complete, by a reduction from the standard NP-complete problem [GT4] of *graph k-colorability* ("Given a graph $G = (V, E)$ and a natural number $k$ such that $2 < k \leq |V|$, determine whether or not there is an $k$-coloring of $G$").
  - Not a significant problem in practice, but worst-case scenarios can be constructed.

# Design Space Dimensions

- [Ref: “Register Allocation Deconstructed”, D. R. Koes and S. C. Goldstein, Proc. 12th International Workshop on Software & Compilers for Embedded Systems (SCOPES), pp. 21–30. 2009.]

- Assignment
  - The action of assigning a register to a variable.
  - E.g., integrated optimal, graph heuristic, linear scan heuristic.
- Spilling
  - The action of storing a variable into memory instead of registers.
  - E.g., integrated optimal, separate optimal, separate heuristic.
- Move Insertion
  - The action of inserting register-register moves, i.e., making a variable live in different registers during its lifetime.
  - E.g., Integrated optimal, integrated optimal ignoring uncoalescable, separate optimal, separate aggressive, none.
- Coalescing
  - The action of limiting the number of moves between registers, thus limiting the total number of instructions.
  - E.g., Full, limited, none.