Compilers

Topic: Global Register Allocation

Monsoon 2011, IIIT-H, Suresh Purini

Scope of Register Allocation

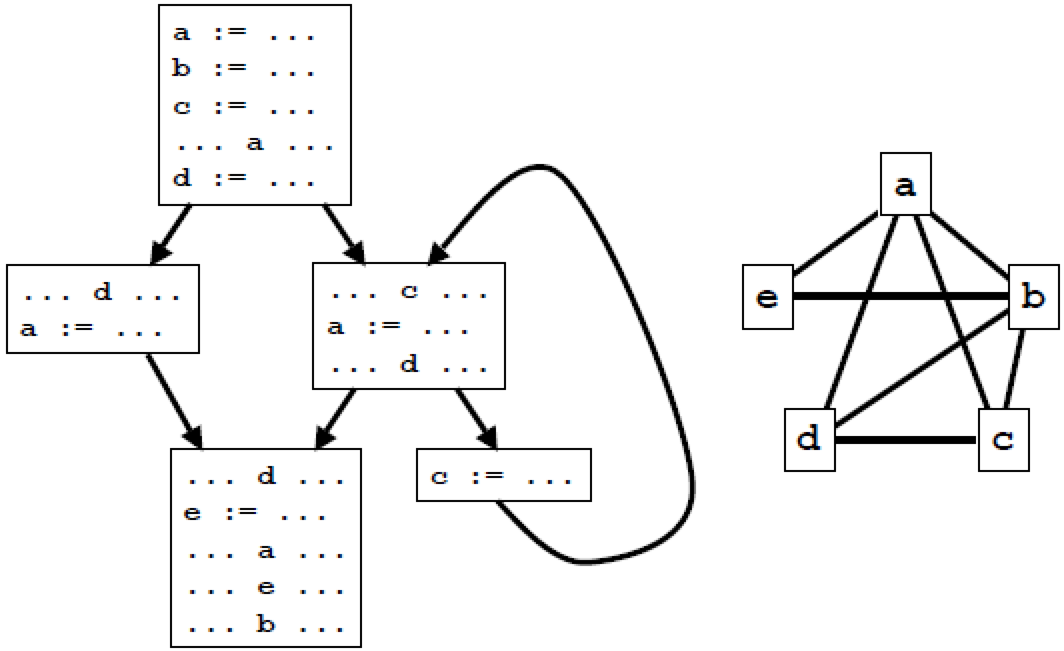
Scopes at which register allocation can be done

- Expression level
- Basic block level
- Loop Level
- Function level (Global Allocators)
- Inter-procedural level

Granularity of Allocation

- Variables
- Webs (or Live ranges)
- Values (corresponds to definitions in SSA form)

Interference Graph(Variables)



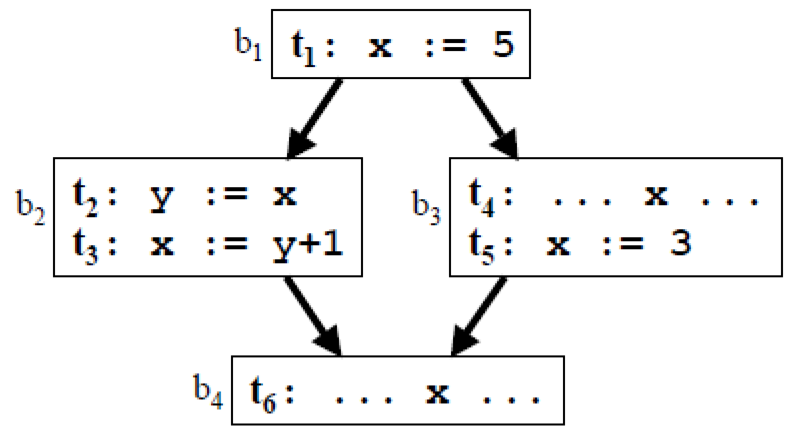
Source: CS553 from Colorado State University

Web

- Def-Use Chain: A def-use chain connects the definition of a virtual register to all of its uses.
- Def: When several definitions reach a single use, we say they are connected by that use.
- All the chains originating from the same definition are considered connected
- Web: The union of all connected def-use chains
- Basic Idea: Allocating registers at the web level prevents are register naming conflicts

Webs

Exercise: Rewrite the code using webs as the new name space.

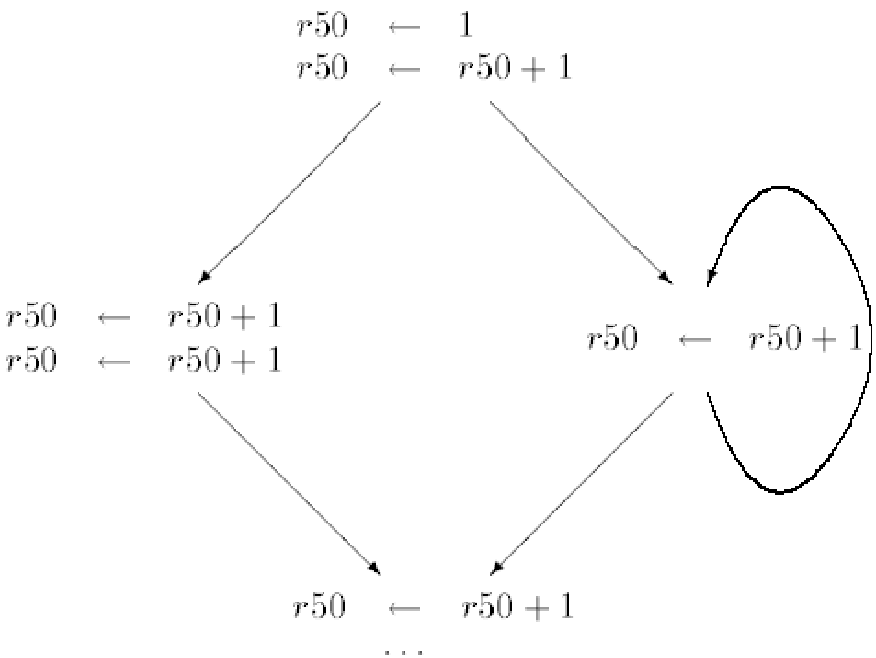


- w_1: (t_1 -> t_2, t_4)
- w_2 : (t_2 -> t_3)
- w_3 : (t_3, t_5 -> t_6)

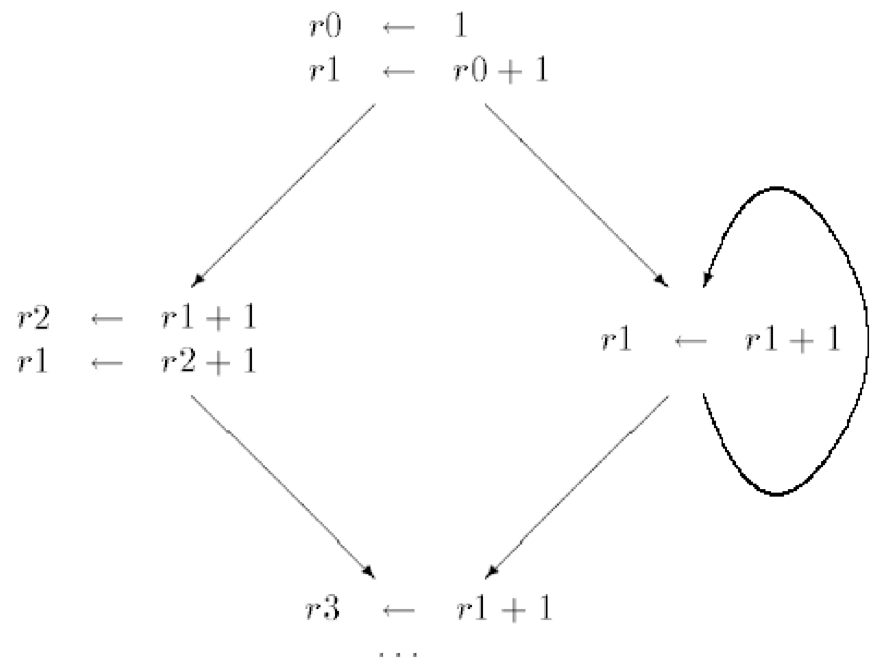
Key Ideas

1. In the webs name-space register allocation can be done without register naming conflicts.
2. Granularity of allocation is smaller than that of variables resulting in possible better register utilization

Webs



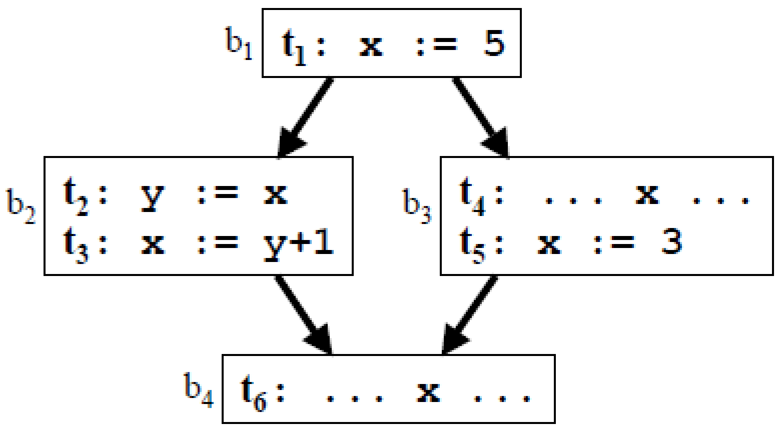
Code in Virtual Register Name Space



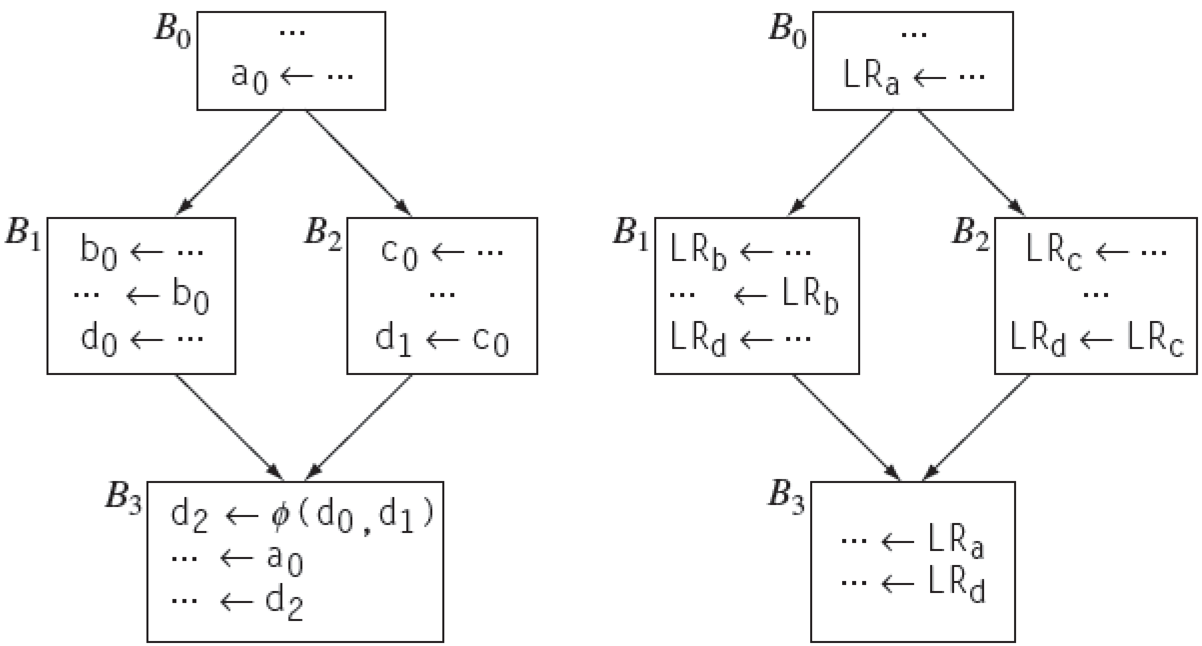
Code in Virtual Web Name Space

SSA Name Space

- In SSA Name Space, def-use chains have simple structure
  - A definition can have multiple uses
  - But only one definition can reach a use
- A web is simply equivalent to an SSA name.



SSA Name Space



(a) Code Fragment in Pruned SSA Form

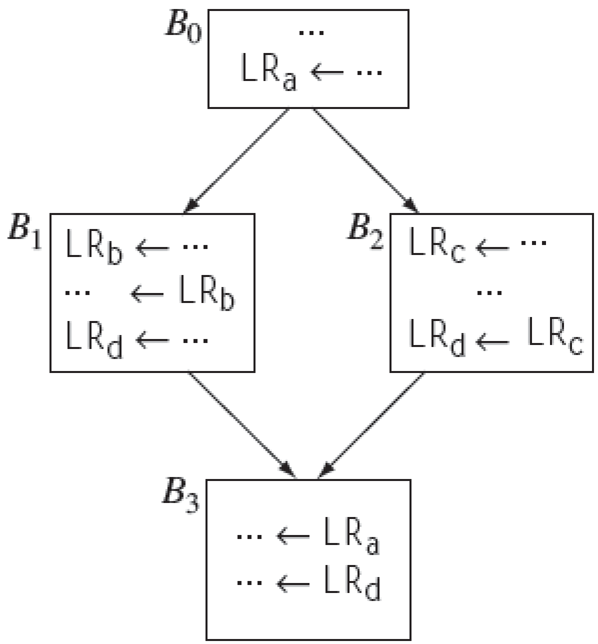
(b) Rewritten in Terms of Live Ranges

Construction of Interference Graph

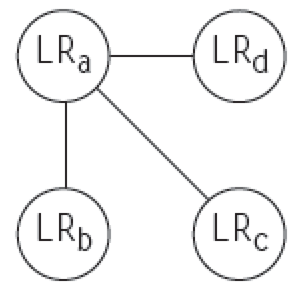
- We say that two live ranges interfere with each other if there exists some point in the function and a possible execution of the function such that:

1. both live ranges have been defined,
2. both live ranges will be used, and
3. the live ranges have different values

Interference Graph



(a) Code Fragment with Live-Range Names



(b) Corresponding Interference Graph

Coalescing

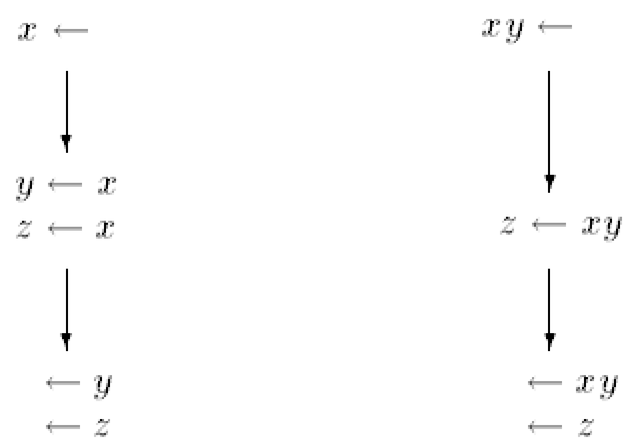
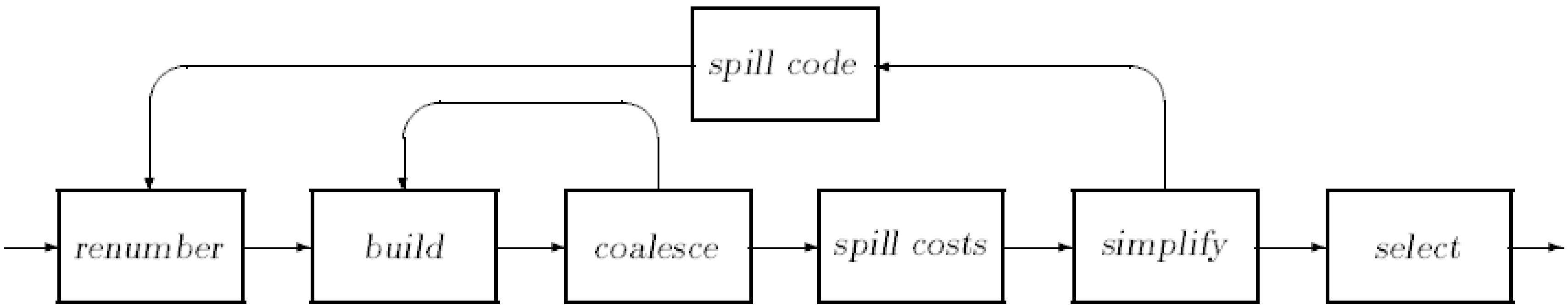
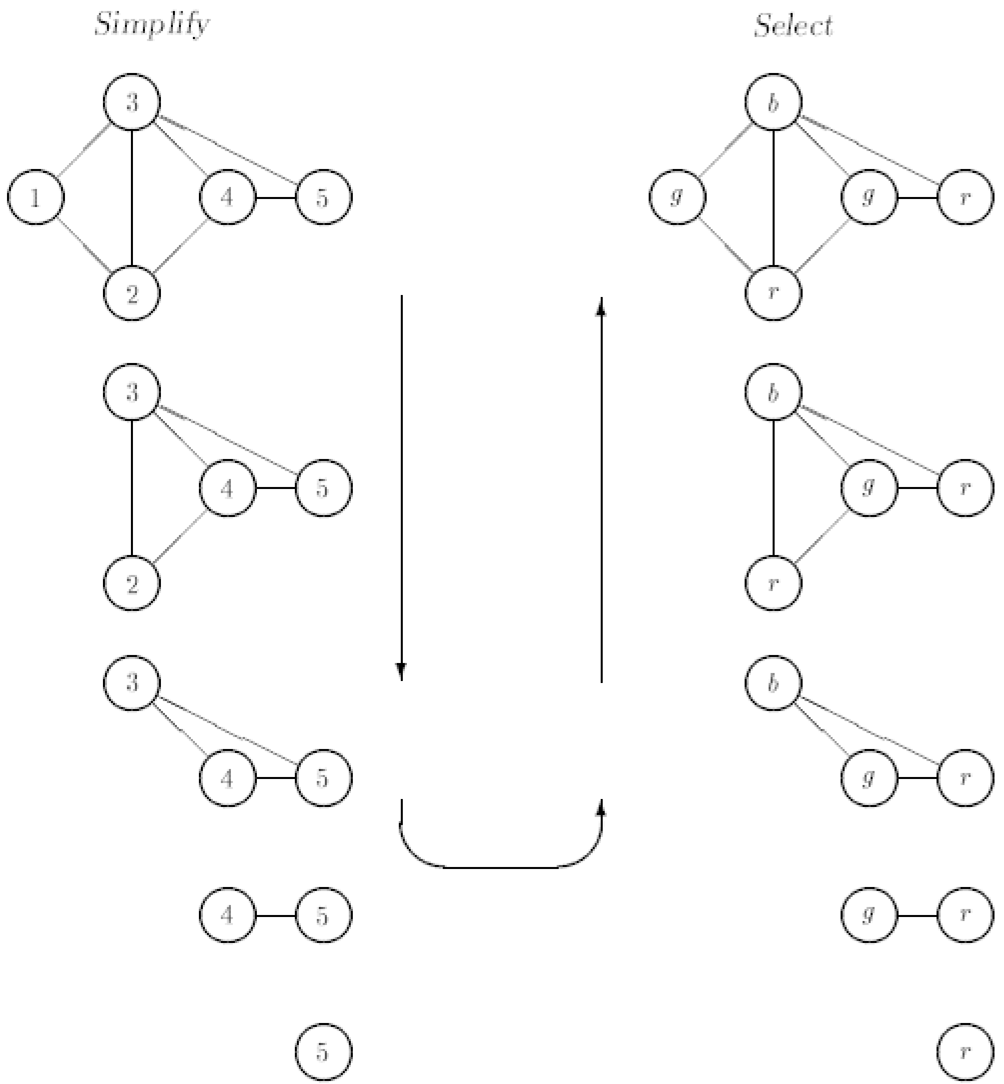


Figure 2.3 Effects of Coalescing

Yorktown Allocator



Simplification



Choosing Spill Nodes

$$m_n = \frac{cost_n}{degree_n}$$

$$\begin{aligned} m_n &= \frac{cost_n}{degree_n^2} \\ m_n &= \frac{cost_n}{degree_n \mathrm{area}_n} \\ m_n &= \frac{cost_n}{degree_n^2 \mathrm{area}_n} \end{aligned}$$

$$area_n = \sum_{\substack{i \in instructions \\ n \text{ is alive at } i}} 5^{depth_i} width_i$$