Optimization Techniques

Advanced Compiler Techniques

2004

Erik Stenman

EPFL

Optimization Techniques Summary

- The most important aspect of an optimization is that it is correct.
- The subject is confusing:
  - The notion of optimality.
  - Huge number of possible optimization.
  - Many intricate and NP-complete problems.
- In this course we have tried to give an overview of some common optimization techniques.

Optimization Techniques Summary

- Suggested method for compiler optimization:
  1. Look at the generated code – try to find sources of inefficient code. (Better yet profile.)
  2. Look in the literature for solutions to these inefficiencies. (Most likely someone has already solved the problem.)
  3. Implement the solution.
  4. Repeat from 1.

Optimization Techniques Summary

- Some techniques are useful for many different problems.
  - Dataflow analysis.
  - Dominators.
  - Liveness.
  - SSA form.
  - Reverse post order traversal.
  - Graph coloring.

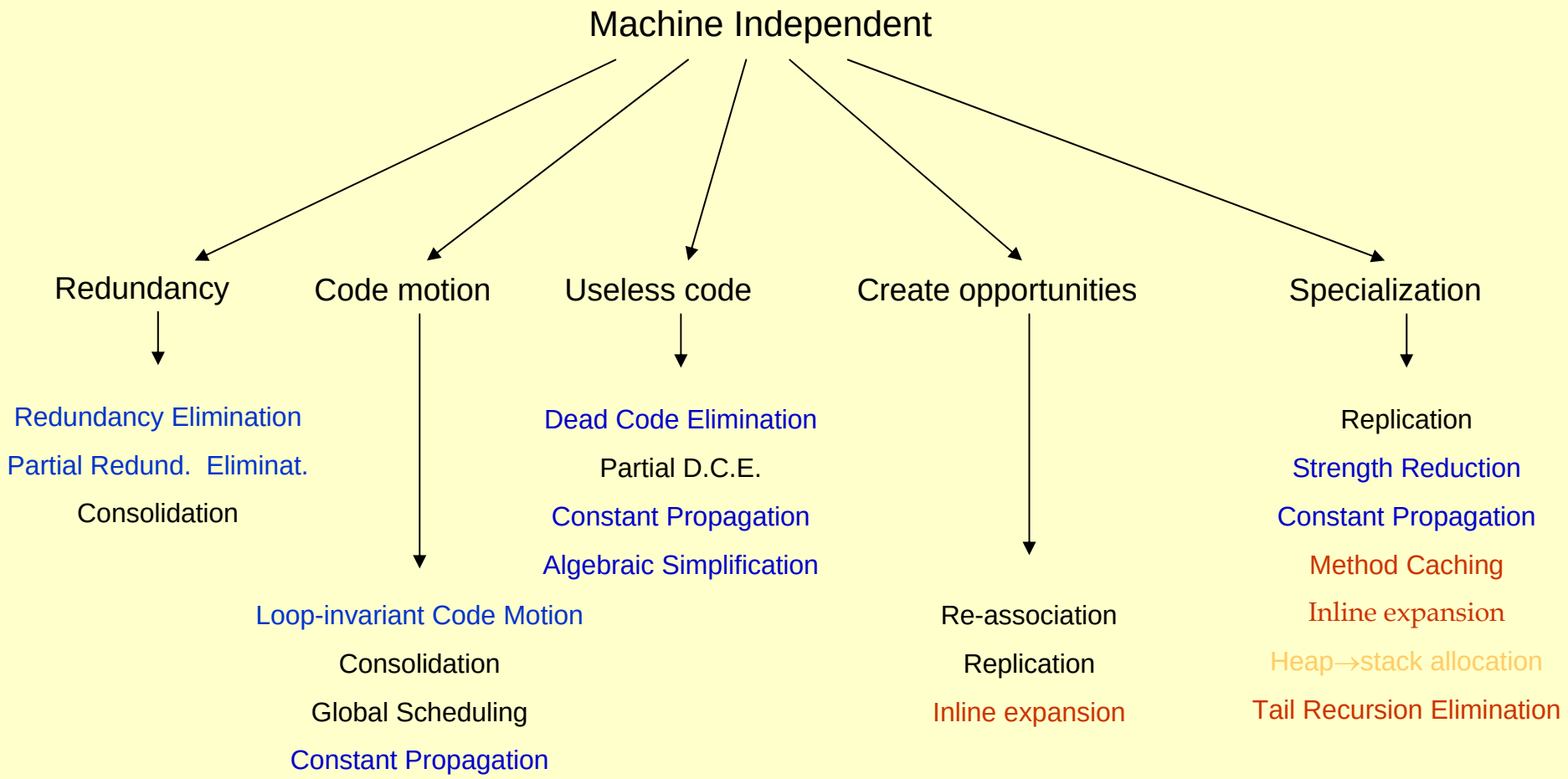
Optimization Techniques Taxonomy

- We can divide optimizations into:
  - Machine independent optimizations.
    - Decrease ratio of overhead to real work.
    - Example: dead code elimination.
  - Machine dependent optimizations.
    - Take advantage of specific machine properties.
    - Work around limitations of a specific machine.
    - Example: instruction scheduling.

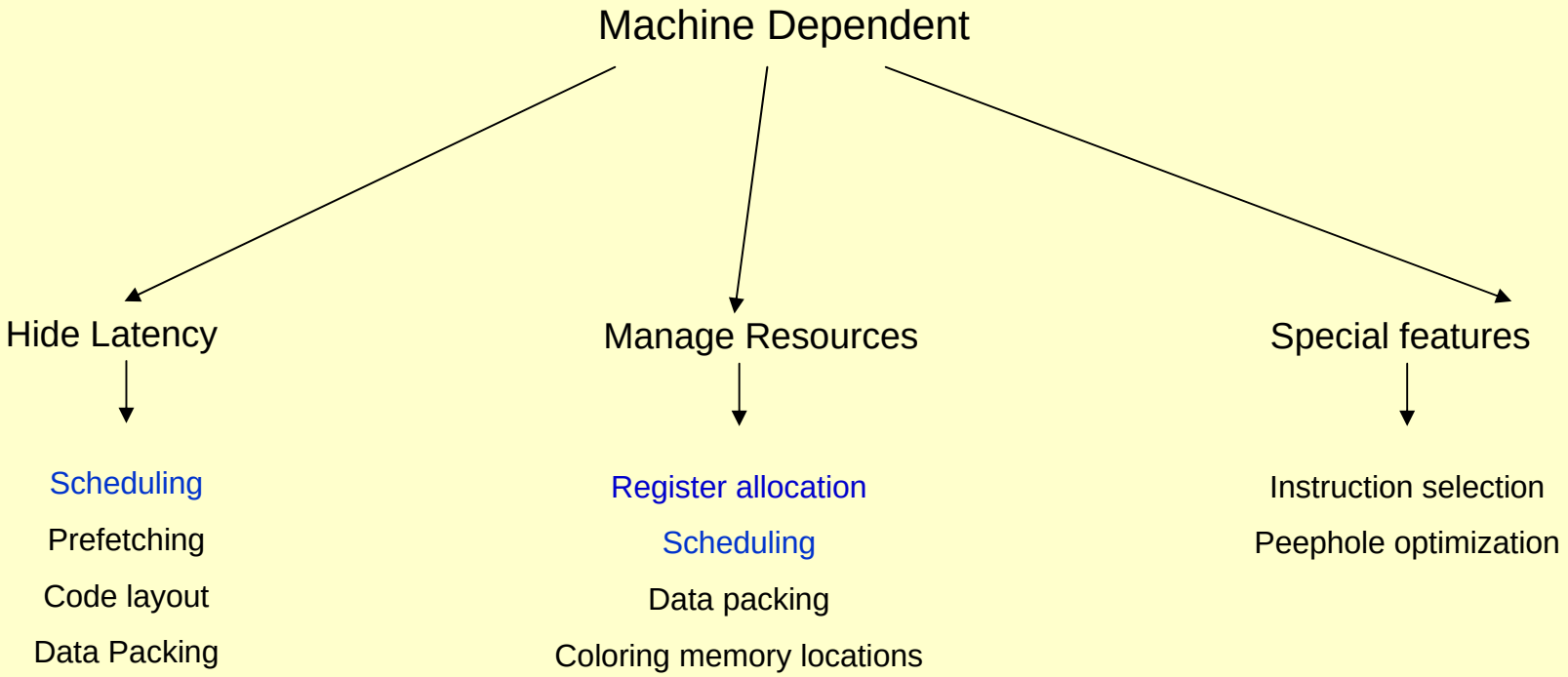
Optimization Techniques Taxonomy

- We can further divide the optimizations on their intended effect.
  - Machine independent optimizations.
    1. Eliminating redundant computations.
    2. Move code to execute it less.
    3. Eliminate dead code.
    4. Specialize on context.
    5. Enable other optimizations.
  - Machine dependent optimizations.
    1. Manage or hide latency.
    2. Take advantage of special hardware features.
    3. Manage finite resources.

Taxonomy of Global Compiler Optimizations



Taxonomy of Global Compiler Optimizations



Optimization Techniques Summary

- The aim of the lectures have been to give you an insight into and overview of some of the most important concepts in optimizing compilers.
- You might also have discovered that the topic is complex and often difficult.
- The project will probably really show you how difficult it is.
- Hopefully the project will also show you how fun it can be.