



Chapter 5 - Summary

Alexandre David
1.2.05





Optimization?

- Rule #1: write correct code.
 - Efficient code means clear for the compiler and **other people**.
- Rule #2: keep it simple.
 - Pick the right algorithm with the right complexity.
 - Only what is needed.



The job of the compiler

PP5.1
PP5.4

- *Safe* optimizations.
 - Know what it means.
 - Keep the semantics – see the twiddle examples.
- Good to know:
 - Code motion.
 - Reduce memory references – fig 5.10.



Wrap-up on CPU

- Know what
 - Superscalar means,
 - Out-of-order execution means.



Profiling

- Know what it is.
- Know how that works.
 - Input, options, output.
- 90/10 rule:
 - It is common to have 10% of the code taking 90% of the time.
 - Profiling will show you that.

- Understand what it means (and then you can find it again).
 - If we speed-up part of the system, the results depends on
 - the relative size of that part,
 - the speed-up obtained on that part.
 - If you can only improve α of your system by a factor k the speed-up is
$$S = 1 / ((1 - \alpha) + \alpha / k)$$
 - Conclusion: you need to improve a lot of the original system to see any effect.