21 Extensions

21.1 Why principal operation counting is not enough

- Looking only at principal operation counts ignores important hardware differences, in

  available instruction sets, number and speed of arithmetic units, registers, size and speed of caches and memory, etc.

- This abstractness can lead to suboptimal choices of algorithms for a particular task.

21.2 Extending principal operation counting

- How we might extend principal operation counting to take better account of hardware differences:
  - Express algorithms with additional parameters that **capture key hardware characteristics as a concept**, e.g., a cache concept.
  - Then study the performance of different algorithms or algorithm variants as assumptions about these parameters are varied—i.e., **are refined into different subconcepts**.

21.3 Organizing details and summarizing statistics

- Main drawback to introducing and varying hardware parameters: the **amount of detail** that must be reported to give a fully accurate picture of an algorithm's performance.
- But organization of information using concept lattices could help in
  - suppression of details at one level while revealing them fully at deeper levels;

  - summarization, aggregation of statistics.
  - providing a database of detailed performance statistics, corresponding to different hardware platforms, that can be accessed at both compile time and run time:
    * to help make the most appropriate choice of algorithms from a library depending on the specific context in which an computation is to be performed, and thus
    * to assist overall in optimizing applications for a particular platform.

22 Conclusion

22.1 Recap

- Concept lattices are a means to classify, present, and use knowledge of abstractions based on incrementally defined sets of requirements.
- Several applications of concept lattices we have found useful:
  - generic component library design,
  - high-level compiler optimizations,
  - library transformations,
  - algorithm performance specification, especially for generic algorithms

22.2 Next steps

- Refinement of algorithm concepts based on a variety of hardware parameters.
- Conceptual specification of distributed and parallel applications.

- Representation of concept-based information in a form that best supports the program analysis that the optimizing concept-based compiler performs.