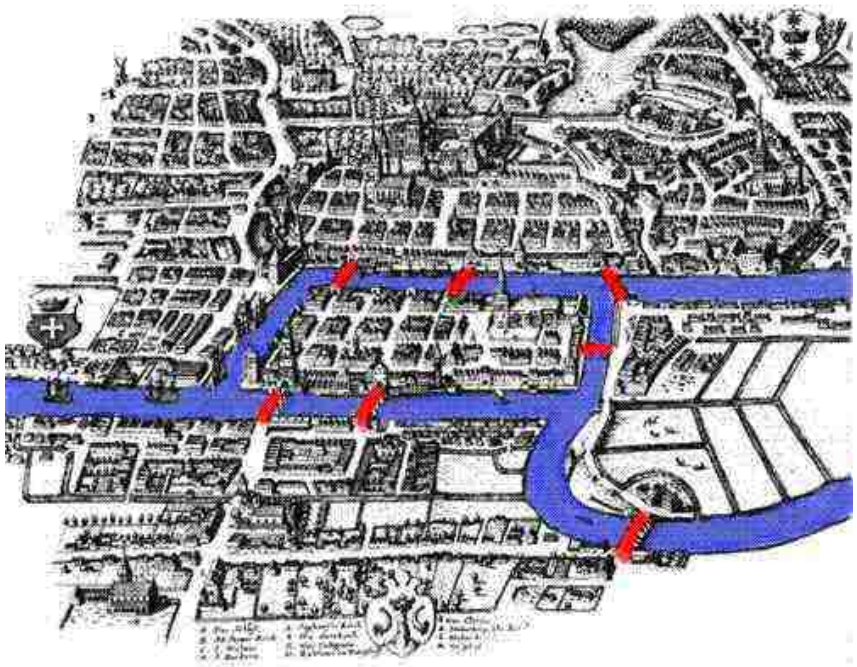
The Hunt for Right Abstractions

Keshav Pingali
Department of Computer Science
University of Texas, Austin

Overview

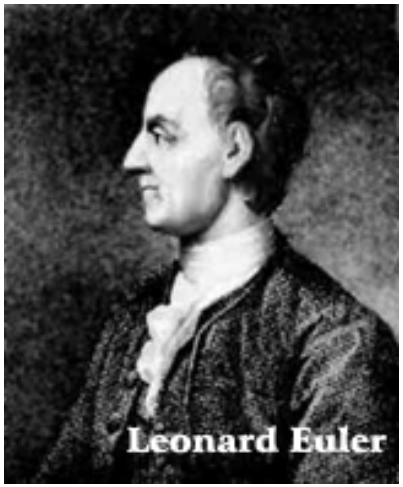
- Abstraction means different things in different areas
  - mathematics/sciences: ignoring some properties of an object so as to focus on the important ones
  - art: representation of object that may be as interesting as the object itself
- Abstractions in Computer Science
  - have something in common with both of these kinds of abstractions
  - abstractions for parallelism in irregular programs
- Good and bad abstractions
  - abstractions can be very powerful and beautiful
  - but they can be misleading if they are the wrong abstractions

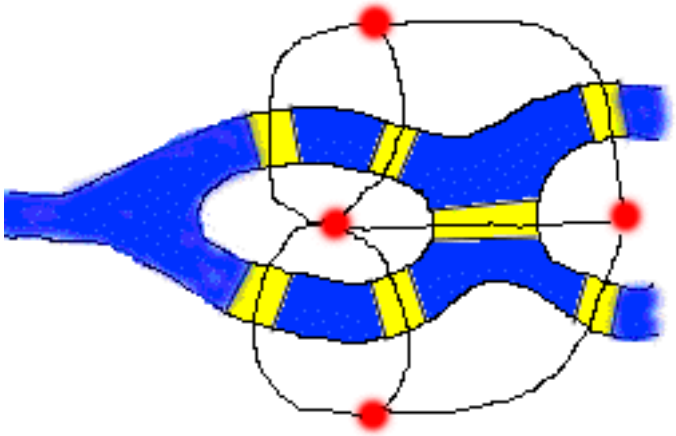
Abstraction in Mathematics



- Bridges of Konigsberg
  - town in Prussia
  - now named Kaliningrad in Russia
- Problem:
  - Is there a walk that crosses each bridge exactly once?
- Citizens of Konigsberg in the 17th and 18th centuries spent lots of time trying to solve this problem

Solution by Euler





- Key insight: connectivity between land masses is what is important, not the actual distances or the orientations of the bridges
- Create an abstraction: graph
  - One node for each land mass
  - Edge between two nodes if there is a bridge connecting the two land masses
- Graph has nodes of odd degree, so there is no walk with desired property
- Led to field we now call topology

Abstraction in Mathematics/science

Bridges of Madison County (Iowa)

- Problem-solving technique
  - the act or process of leaving out of consideration one or more properties of a complex object so as to focus on others
    - (e.g.) Euler left out distances and orientations
  - a general concept formed by extracting common features from specific examples
    - (e.g.) Topology is an abstraction of geometry

Abstraction in art

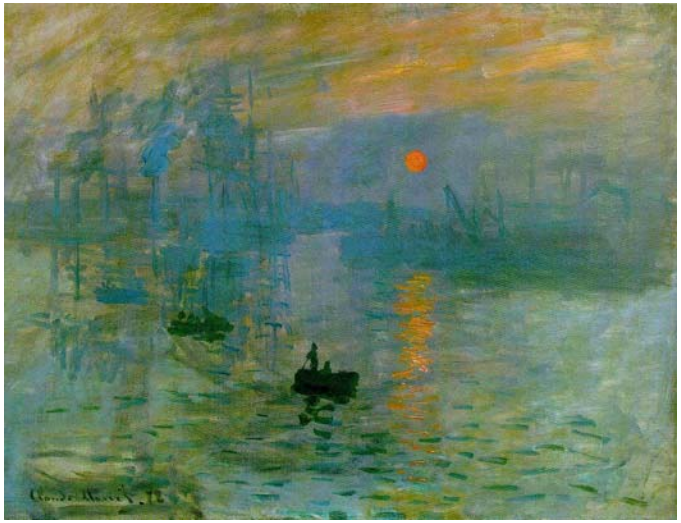
Raphael : Madonna and Child

Claude Monet: Haystacks

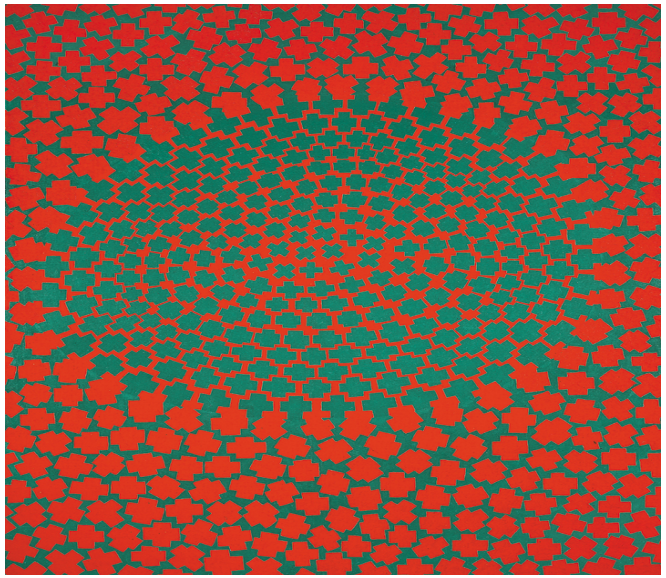
- "That it was a haystack the catalogue informed me. I could not recognize it. This non-recognition was painful to me. I considered that the painter had no right to paint indistinctly. I dully felt that the object of the painting was missing. And I noticed with surprise and confusion that the picture not only gripped me, but impressed itself ineradicably on my memory. Painting took on a fairy-tale power and splendour."

  – Wassily Kandinsky

Abstraction in art: luminance



Claude Monet: Impression Sunrise



Richard Anuszkiewicz: Plus Reversed

- How we see objects
  - what: contrast
  - where: luminance
- Impressionists abstracted away objects and exploited how light is perceived by the eye and brain
  - eye has difficulty finding edges of objects if object and background have the same luminance
- Human-centered abstraction: how the human eye/brain sees the representation of an object may be as interesting as the object itself

Abstraction in literature

Shorter Moby Dick (Ben Hoyle, Times April 14, 2007)

Moby-Dick Ishmael: Whaling's cool.

Queequeg: Tattoos are cool.

Starbuck: Coffee's cool.

Ahab: Fools! Stop yer philosophizin' and help me fight this fish.

Moby-Dick (rising from waves): Screw you, Pegleg!

All: At last! Some action!

Moby-Dick: [Crash! Chomp! Blow!]

All: Aaargh!

Ishmael (later, alone, clinging to wreckage): Whaling's cool.

Bad abstractions

"Naked Blue IV" Henri Matisse (1952)

- Abstraction is bad if it has thrown away some essential feature of the problem
  - topologist is someone who does not know the difference between a doughnut and coffee-cup
- What is essential depends on the use you intend to make of the abstraction

Abstractions in PL

- My opinion: most important advances in PL have introduced new abstractions
- Examples:
  - Procedures (1950?)
    - abstraction: parameterized code module (λ–abstraction)
    - abstracted away: implementation code
  - Instruction-set architecture (IBM 360)
    - abstraction: machine language
    - abstracted away: micro-architecture
  - FORTRAN I (1957)
    - abstraction: high-level programming language
    - abstracted away: machine language

Abstractions in PL (contd.)

- Examples (contd.):
  - Structured programming (1967)
    - abstraction: structured control-flow constructs like if-then-else, while-loops, for-loops etc.
    - abstracted away: conditional jumps (machine language relic)
  - Object-oriented programming (1970-)
    - abstraction: abstract data type
    - abstracted away: representation of data type
  - Automatic storage management (1960-)
    - abstraction: objects
    - abstracted away: machine addresses (pointers)

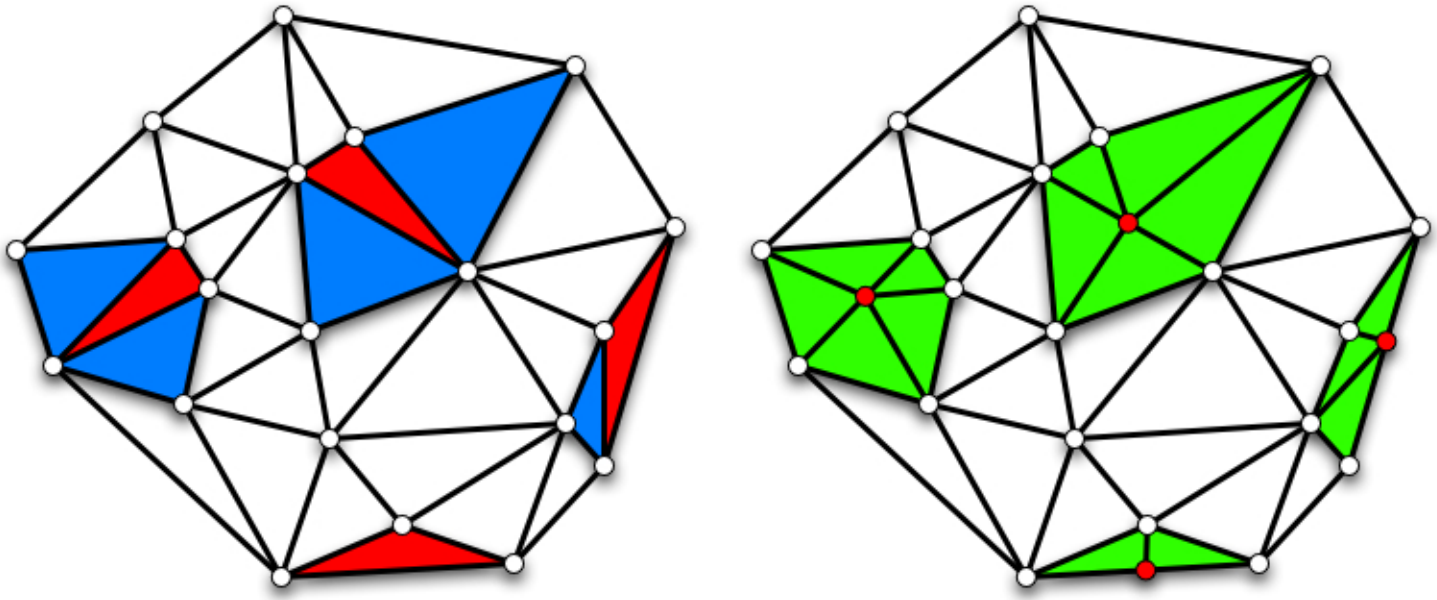
Abstractions for parallelism

- Irregular programs:
  - pointer-based data structures
  - parallelism is organized around worklists
  - kind of data-parallelism but more complex than array-based data parallelism
  - parallelism may be very data-dependent
    - whether or not two worklist elements can be processed in parallel may depend on input data

    ➔purely compile-time parallelization cannot work

    ➔runtime dependence checks are needed

Delaunay Mesh Refinement

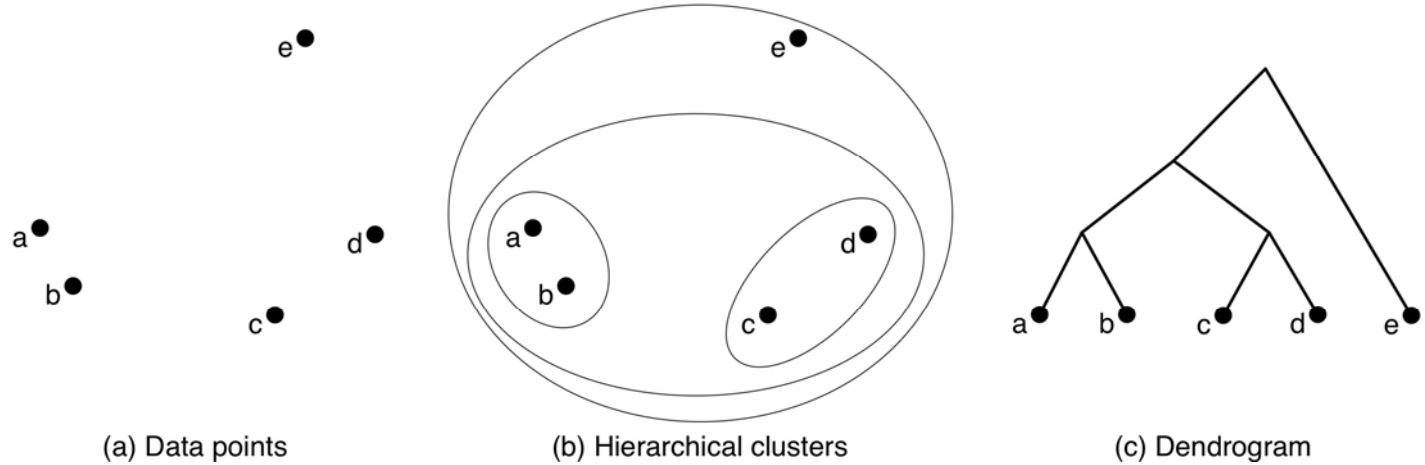


- Bad triangles with non-overlapping cavities can be processed in parallel
- Whether or not two cavities overlap depends on the mesh: need speculation
- However, thread-level speculation (TLS) however has high abort ratio

Sequential Algorithm

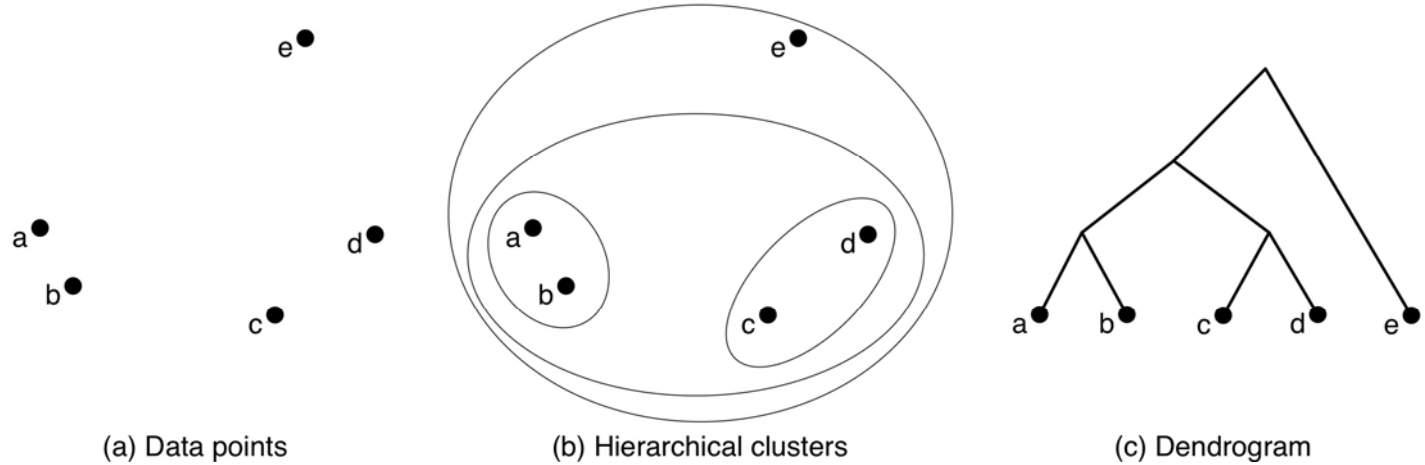
```
Mesh m = /* read in mesh */
WorkList wl;
wl.append(mesh.badTriangles());

while (true) {
        if ( wl.empty() ) break; //done

        Element e = wl.get-first();
        if (e no longer in mesh) continue;

        Cavity c = new Cavity(e);//determine new cavity
        c.expand();                 //determine affected triangles
        c.retriangulate();          //re-triangulate region
        m.update(c);                //update mesh
        wl.append(c.badTriangles()); //add new bad triangles
}
```

Agglomerative Clustering



- Input:
  - Set of data points
  - Measure of “distance” (similarity) between them
- Output: dendrogram
  - Tree that exposes similarity hierarchy
- Applications:
  - Data mining
  - Graphics: lightcuts for rendering with large numbers of light sources

Clustering algorithm



- Sequential algorithm: iterative
  - Find two closest points in data set
  - Cluster them in dendrogram
  - Replace pair in data set with a “supernode” that represents pair
    - Placement of supernode: use heuristics like center of mass
  - Repeat until there is only one point left
- Key data structure: priority queue

Solution: set iterators

- *for each e in Set S do B(e)*
  - evaluate block B(e) for each element in set S
  - sequential implementation
    - set elements are unordered, so no a priori order on iterations
    - there may be dependences between iterations
  - set S may get new elements during execution
- *for each e in PoSet S do B(e)*
  - evaluate block B(e) for each element in set S
  - sequential implementation
    - perform iterations in order specified by poSet
    - there may be dependences between iterations
  - set S may get new elements during execution
- PLDI 2007 paper: “Optimistic Parallelism Requires Abstractions”

Bad abstractions in PL?

- Abstractions that are
  - difficult to reason about
  - hard to implement efficiently
- Example: functional languages for parallel programming
  - Abstract away the notion of storage: only values and functions on values
  - Elegant parallel execution models: reduction, dataflow
  - Big problems:
    - data structure manipulation can be very inefficient if you view data structures as values
    - hard to get a handle on locality
  - Unfortunately, parallelism in algorithms is mostly data parallelism ☹
  - Notion of storage might be an essential feature of program execution that should not be abstracted away by the programming language

Summary

- Abstraction means different things in different areas
  - mathematics/sciences: ignoring some properties of an object so as to focus on the important ones
  - art: representation of object that may be as interesting as the object itself
- Abstractions in Computer Science
  - have something in common with both of these kinds of abstractions
- How I learnt to value abstraction
  - most of the world is hostile to abstractions
  - working with Arvind, Jack and the dataflow group taught me the power and the perils of abstraction

The Hunt For Right Abstractions