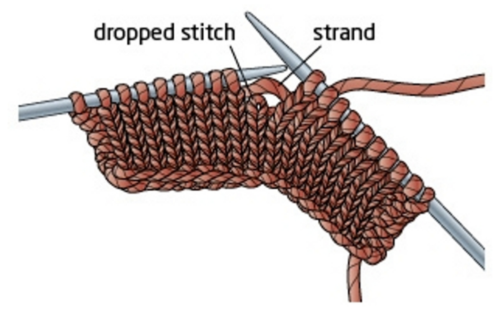
Retroactive Data Structures

Michael T. Goodrich, Joseph A. Simons
Department of Computer Science, University of California, Irvine

Motivation

We may need to alter the historical sequence of transactions.

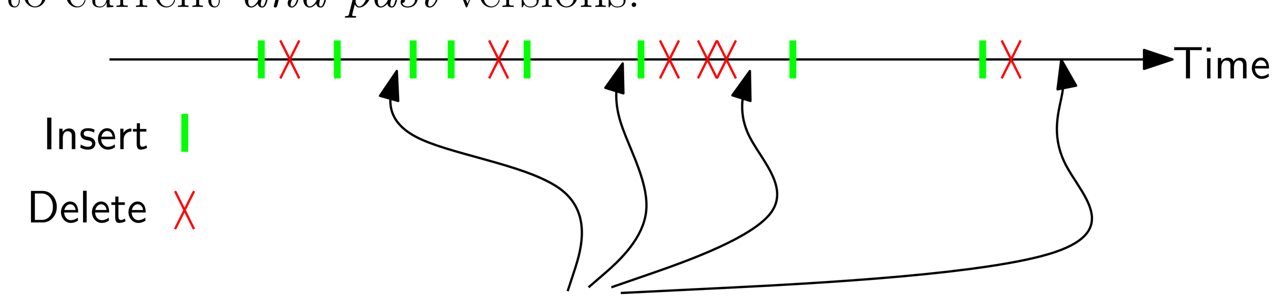
- **Dynamization.** Some static algorithms require a dynamic data structure to process the input. These algorithms can be made dynamic by using a retroactive data structure instead.
- **Bad Data.** Data is either missing, was entered incorrectly, or comes from a tainted source (e.g. a sensor has malfunctioned).
- **Efficient Adaptation.** We may want to undo an error made early on without redoing all the work since then.



Goal: Maintain a large amount of data which changes over time such that we can efficiently update and query the data. Support updates and queries on current and past versions of the data structure.

Definition

A data structure is **fully retroactive** if it supports queries and updates to current *and past* versions.



Query and/or **Update** at any point in time.

Persistent data structures also maintain past versions, but only retroactive data structures propigate updates forward in the timeline.

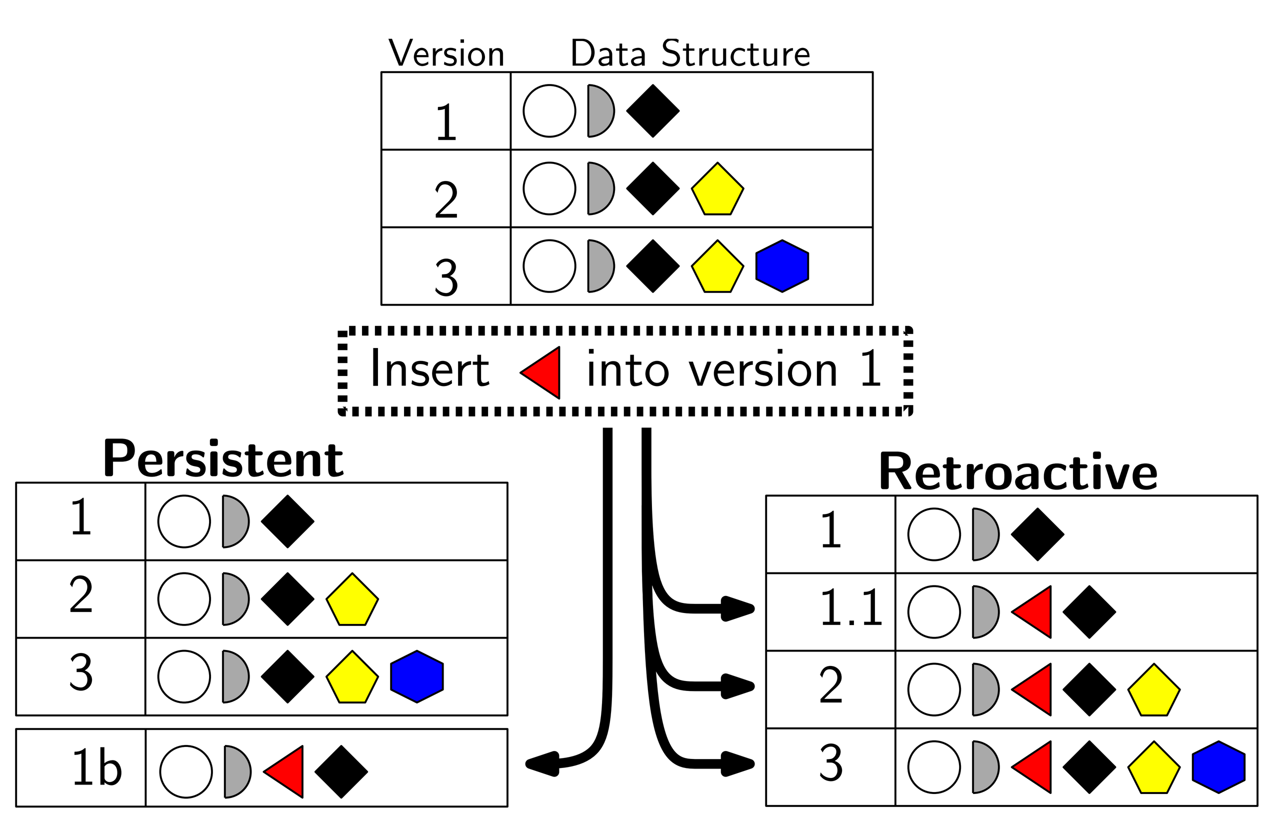
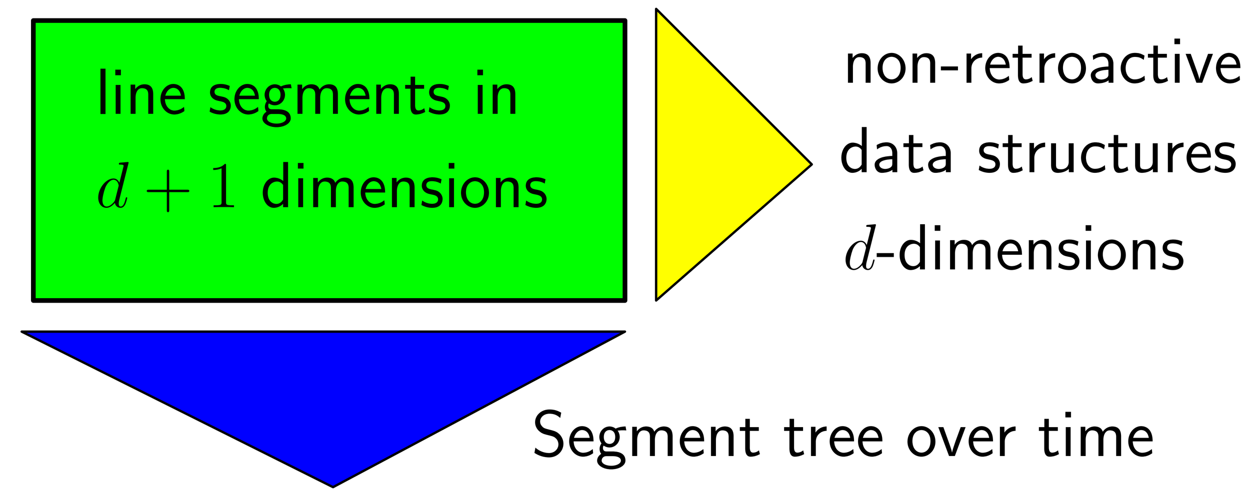


Figure: All later versions reflect the update in a fully retroactive data structure.

General Approach



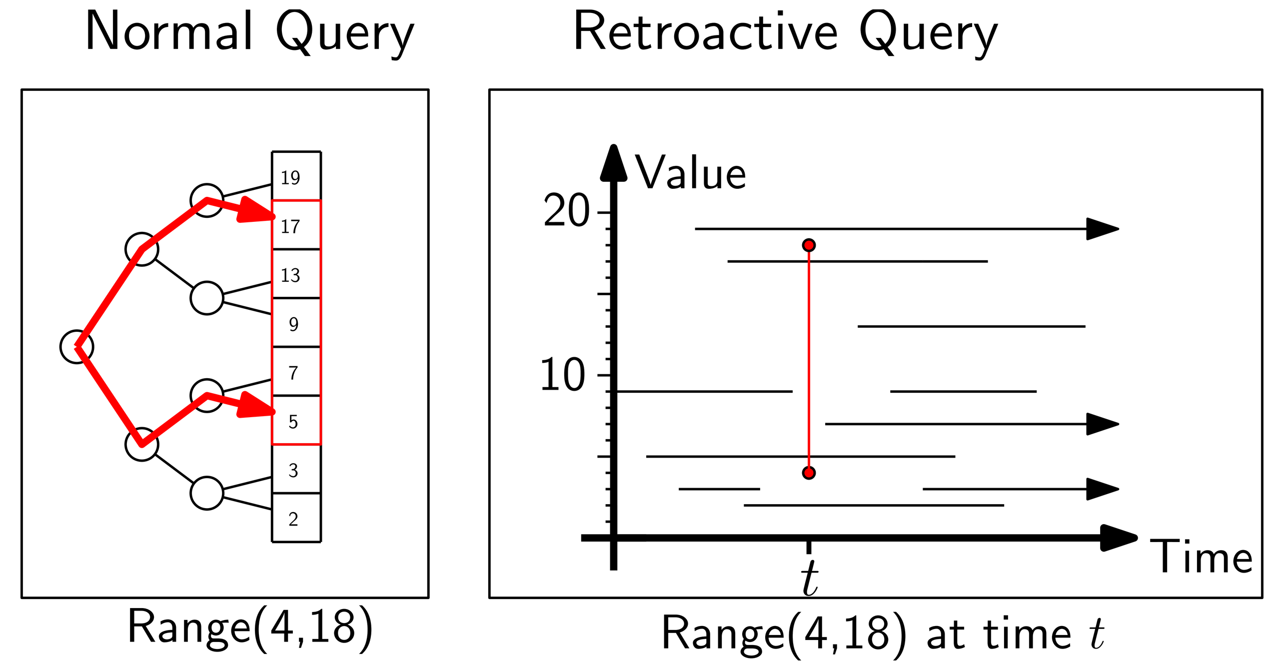
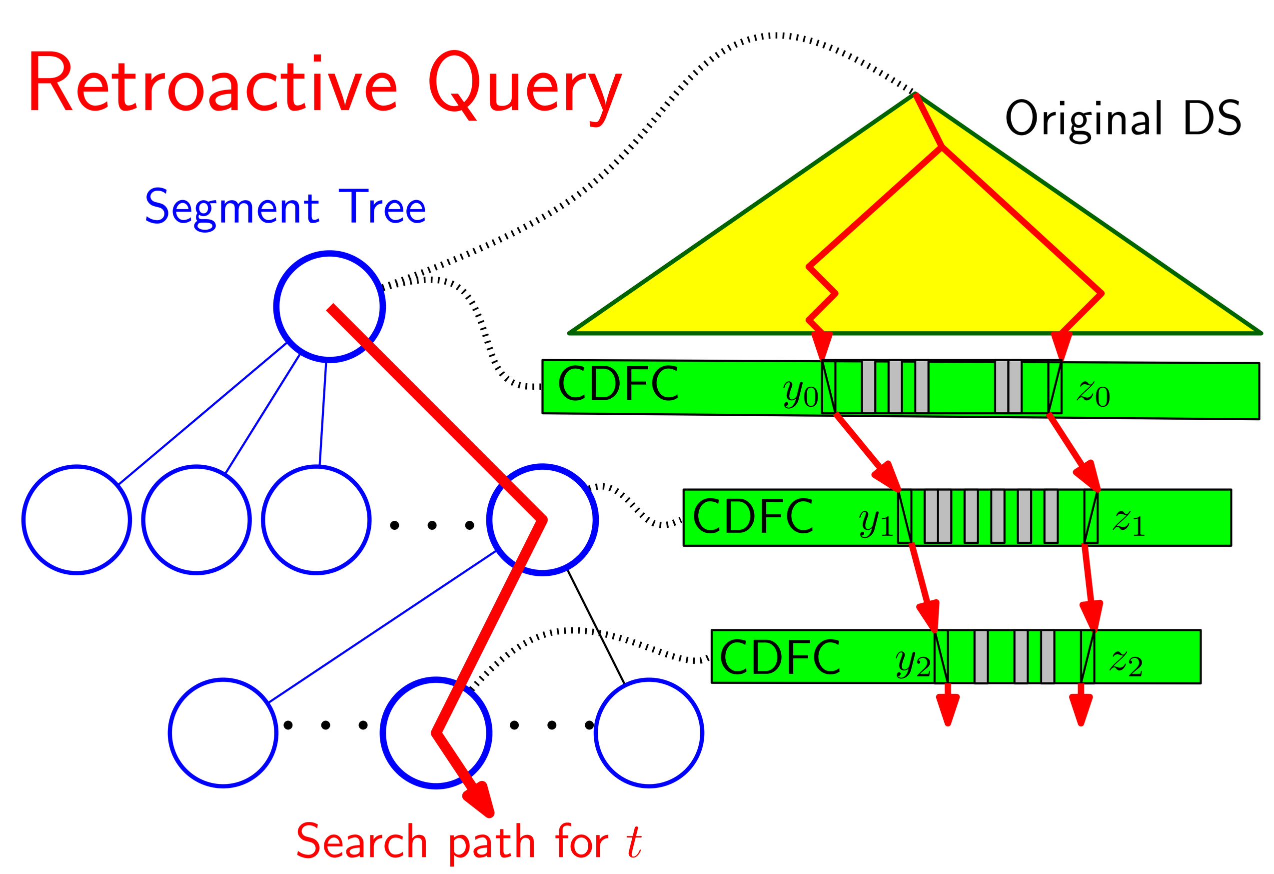


Figure: Illustrating the conversion to a fully retroactive query.

Avoiding extra $O(\log n)$ overhead:

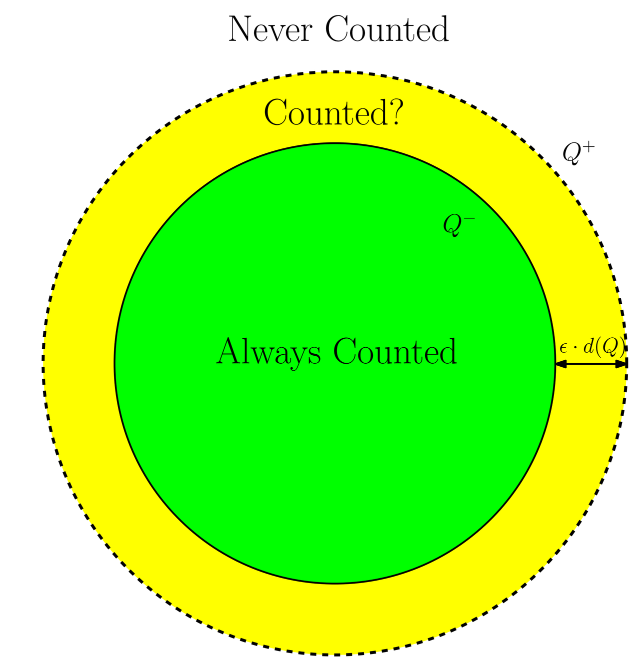
- Build high-degree segment tree over segments.
- Augment root with traditional data structure.
- Augment nodes with CDFC structures.



Query at time t:

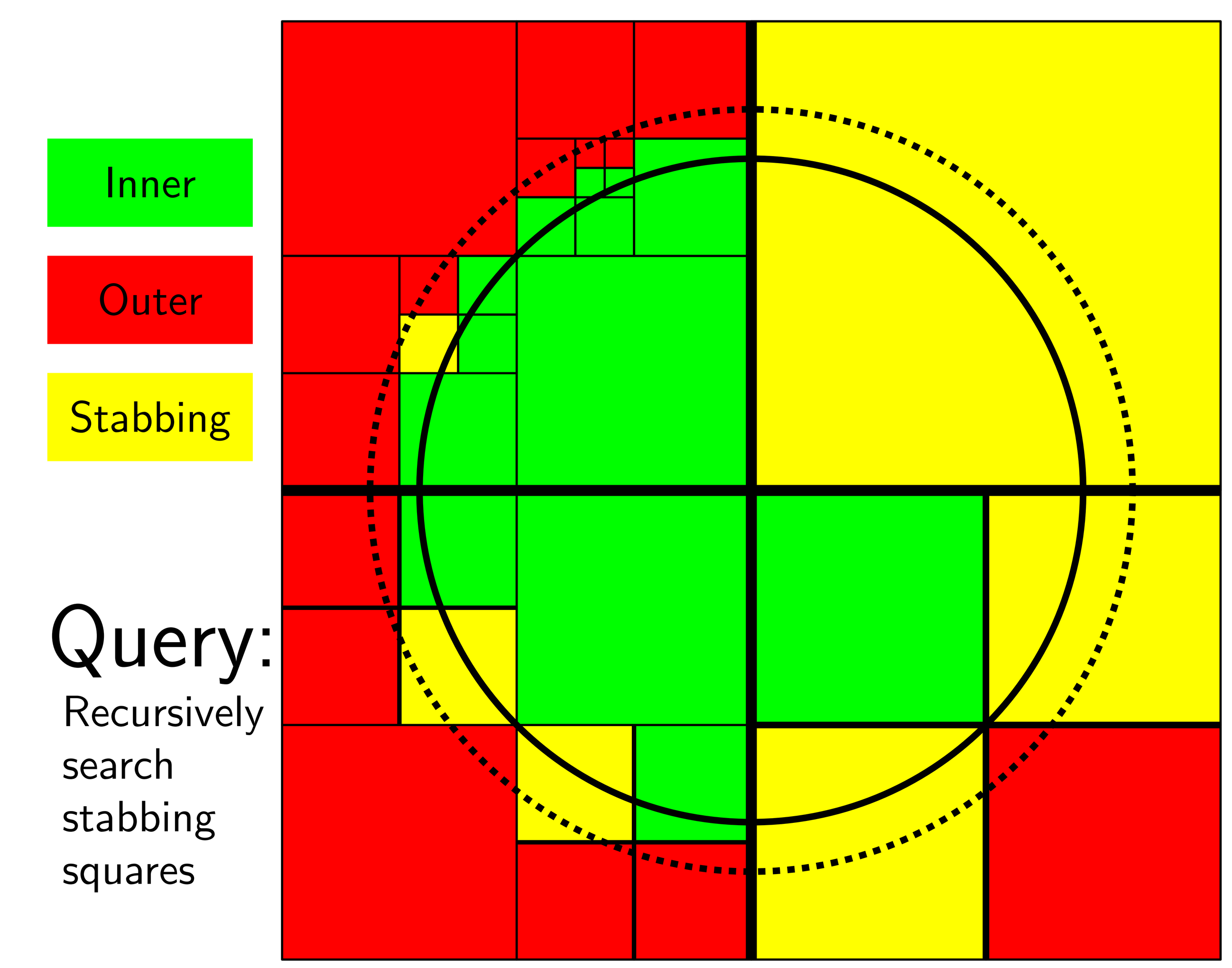
- Search for t in segment tree.
- non-retroactive query at root.
- fractional cascading in each CDFC.

Approximate Range Searching



- Input: set of points in $\mathbb{R}^d$.
- Updates can insert or delete points at any time.
- Query: return points in approximate range at time t.

Figure: Points within Q^- must be returned. Points outside Q^+ are never returned. Points between Q^- and Q^+ may or may not be returned.



For each Inner Square:

- Reduce to 1-d query using Z-order
- Perform 1-d query in augmented segment tree

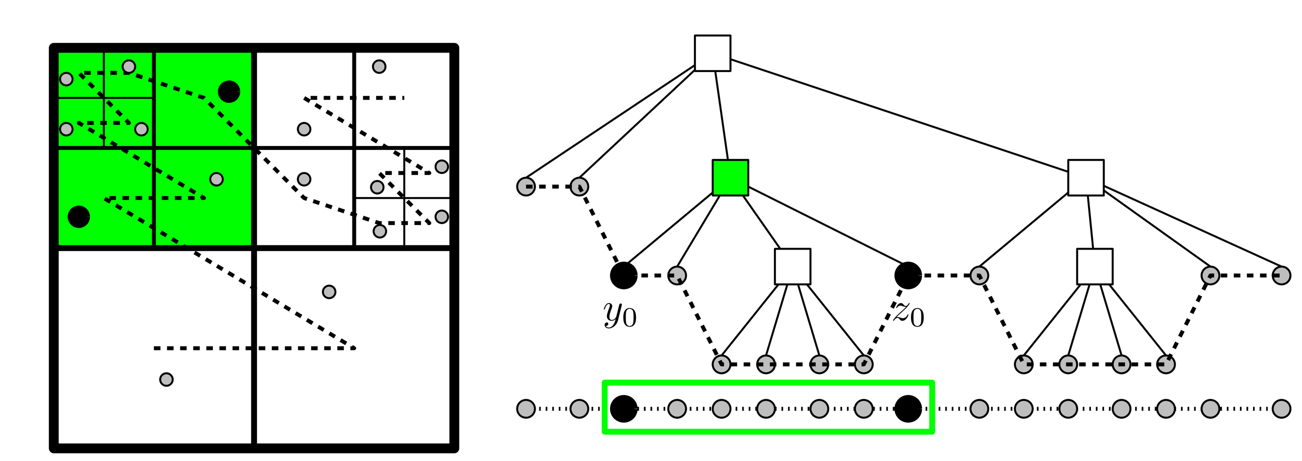


Figure: Z-order curve projects quadtree down to one dimension.

[1] M. T. Goodrich and J. A. Simons. Fully retroactive approximate range and nearest neighbor searching. *Algorithms and Computation*, pp. 292–301. Springer, Lecture Notes in Computer Science 7074, 2011.

University of California · Irvine
Center for Algorithms and Theory of Computation
Supported by: Office of Naval Research award N00014-08-1-1015.
Scalable Methods for the Analysis of Network-Based Data