Graph traversals as universal constructions

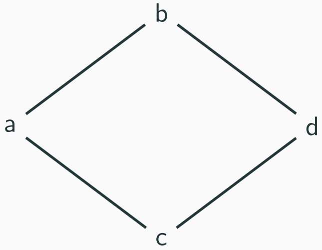
Siddharth Bhaskar with Robin Kaarsgaard

Københavns Universitet and the University of Edinburgh

Searching through graphs

- A graph search is an algorithm for systematically visiting each vertex in a graph.
- Searches are *local* (each vertex is in the neighborhood of previously visited ones).
- The resulting vertex orders are called *traversals*.

Example



Traversals

a b c d
a c b d
a b d c
a c d b

Non-traversals

a d b c
a d c b

Motivating question

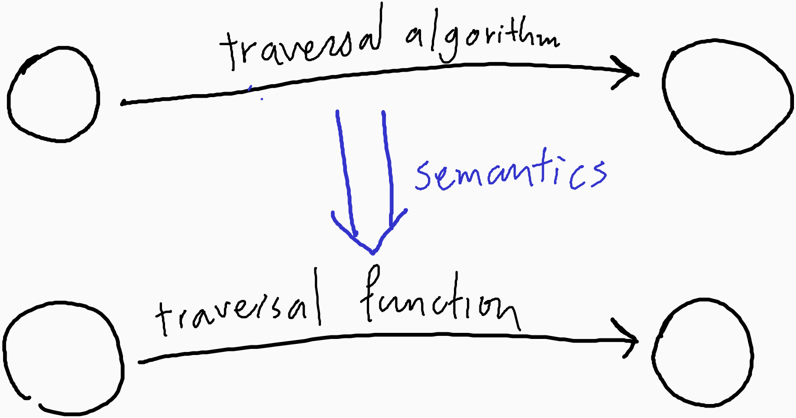
- In what sense is a traversal (or BFT/DFT) canonical in its original graph?
- Observe that there is no canonical way to choose between BFTs or DFTs in the previous graph.
- Solution: linearly order each neighborhood.
- Our contribution: the resulting *lexicographic* BFTs and DFTs are canonical (functorial) in the edge-ordered graph

Structure vs. Power

- **Power.** Traversals are *expressive*.
  - Graph searches are used as "subroutines" in all sorts of algorithms (connectivity, planarity, . . .).
  - Connection to complexity classes.
  - Various interesting *algorithms* (sequential, parallel, . . .)
- **Structure.** By endowing traversals with the right categorical structure, we might hope to
  - "read off" algorithms computing them
  - clearly visualize obstructions to efficient computation.

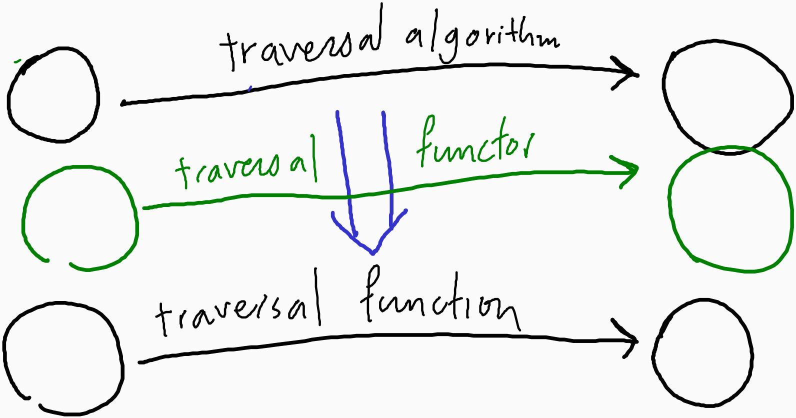
Structure vs. Power

I see our work as *interpolating* between the problem statement and algorithms solving it.



Structure vs. Power

I see our work as *interpolating* between the problem statement and algorithms solving it.



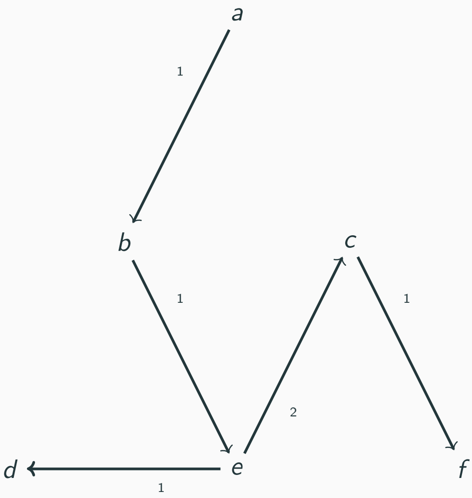
Structure vs. Power

- *Inferring algorithms from problems* is a very tall order.
- Moderate version: infer **stacks** from DFTs and **queues** from BFTs.
- I believe our work is a first step in this direction.

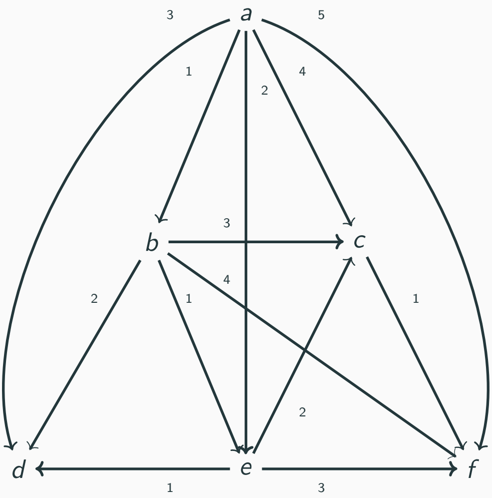
Our construction—DFT

$$\begin{tikzcd}
 & & a \arrow[dl, "1"'] \arrow[dr, "2"] & & \\
 & b \arrow[dl, "2"'] \arrow[dr, "1"] & & c \arrow[dr, "1"] & \\
d & & e \arrow[ll, "1"] \arrow[ur, "2"'] & & f \arrow[ll, "1"]
\end{tikzcd}$$

Our construction—DFT



Our construction—DFT



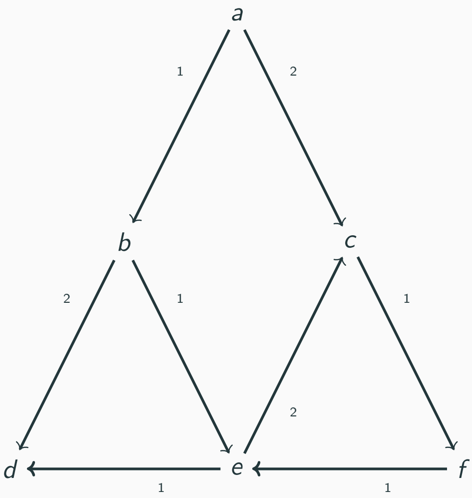
Our construction—DFT

- It turns out that both transformations can be formulated as universal constructions in the right category.

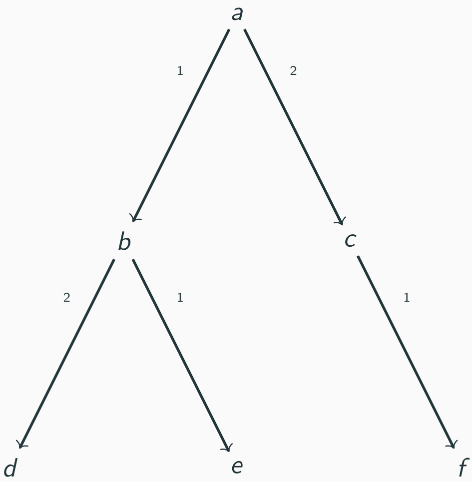
$$\begin{tikzcd}
\mathbf{FinGraph}^{lex}_{\star} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \mathbf{FinArb}^{<}_{\star} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathbf{TLexGraph} \arrow[l, bend left]
\end{tikzcd}$$

- The output DFT as a linear order can be extracted by a forgetful functor out of **TLexGraph**.

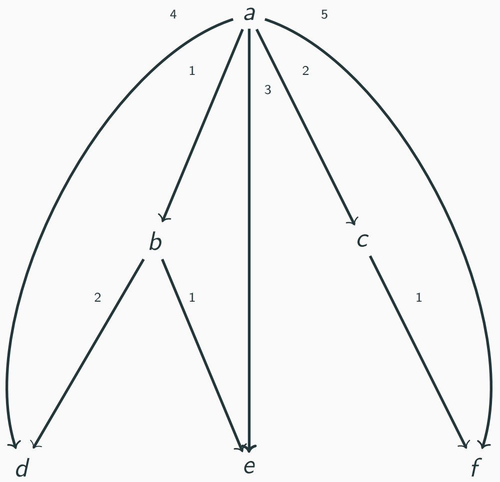
Our construction—BFT



Our construction—BFT



Our construction—BFT



Our construction—BFT

- It turns out that both transformations can be formulated as universal constructions in the right category.

$$\begin{tikzcd} \mathbf{FinGraph}^{slex}_{\star} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \mathbf{FinArb}^{<}_{\star} \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathbf{TArb} \arrow[l, bend left] \end{tikzcd}$$

- The output BFT as a linear order can be extracted by a forgetful functor out of **TArb**.

Remarks

- The universal constructions in the BFT and DFT setting are analogous, but imperfectly so.
- Both lexicographic BFTs and DFTs are known to have a general setting as *path algebra problems* relative to an arbitrary semiring.
- The natural next step is to see whether we can express a general path algebra problem via universal constructions.

Remarks

- What does the categorical decomposition suggest about (sequential or parallel) algorithms that solve these problems?
- A single universal construction suggests a greedy strategy. What about two? (One free one cofree)
- Can we already extract some upper bounds from our decomposition?
- If we could show that we could *not* express a lex-BFT/DFT as a *single* universal construction, could we extract lower bounds from that?

Remarks

- Lex-BFTs naturally extend from finite edge-ordered graphs to infinite edge-well-ordered graphs.
- Resulting BFTs always well-ordered.
- However, not all infinite graphs have well-ordered DFTs. (Problem: well-orderings are not closed under taking lists and ordering them lexicographically.)
- Can we use our machinery to find the “correct” notion of a lex-DFT for infinite edge-ordered graphs, or show that none exist?

Remarks

Thank you for your attention!