Analyzing the effect of the topology on succinct tree encodings

José Fuentes , Alexander Irribarra, Diego Seco

September 2020

Many succinct tree encodings exist, but it is known that Fully Functional [5] is, in general, the best solution in practice at the moment [1]. Similar to other solutions, it works by encoding a tree as a balanced parentheses sequence [3], and to navigate the tree is equivalent to navigate the parentheses sequence. One of the key operations for this is to find the matching parenthesis of a given one. Fully Functional representation supports this query in constant time, but in practice it is implemented in a time that is logarithmic to the distance between the queried parenthesis and its matching one [4], which has been proven to be fast enough in practice. It can be seen that the distance between parentheses of a node is usually higher for nodes closer to the tree's root, as they tend to have more descendants. In other words, every node increments its ancestors' parentheses distance, thus affecting their query time detrimentally. This leads to shallower trees having lower average query time, since nodes will have fewer ancestors.

We exploit this property in Ferres *et al* [2]. encoding for planar embeddings. This encoding works by storing a spanning tree T of the graph G, a spanning tree T' of the dual of G where the edges in T' are dual to the edges in $G - T$, and a bitvector representing the interleaving between the two trees in a counter-clockwise DFS traversal of G. This encoding allows navigation queries on the primal and dual of G. Originally, T is obtained with a DFS traversal of G, but by instead obtaining T with a BFS traversal of G or its dual we can decrease the height of both trees, consequently decreasing time of navigation operations on the embedding from about 20% up to 50%. We also obtain further speedups on queries on G by performing the BFS on G, while further speedups on queries on its dual can be obtained by performing the BFS on the dual.

As future work, it would be interesting to analyze how other succinct tree encodings, such as LOUDS, are affected by the tree topology. On the other hand, we could also study the effect of using flatter trees when possible in other compact data structures using succinct trees.

References

[1] Diego Arroyuelo, Rodrigo Cánovas, Gonzalo Navarro, and Kunihiko Sadakane. Succinct trees in practice. In *2010 Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 84–97. SIAM, 2010.

[2] Leo Ferres, José Fuentes-Sepúlveda, Travis Gagie, Meng He, and Gonzalo Navarro. Fast and compact planar embeddings. *Computational Geometry*, page 101630, 2020.

[3] G. Navarro and K. Sadakane. *Compressed Tree Representations*, pages 397–401. Springer, 2nd edition, 2016.

[4] Gonzalo Navarro. *Compact data structures: A practical approach*. Cambridge University Press, 2016.

[5] Gonzalo Navarro and Kunihiko Sadakane. Fully functional static and dynamic succinct trees. *ACM Transactions on Algorithms (TALG)*, 10(3):1–39, 2014.