Description and Proof of Correctness of `treedrawing`

poypoyan

4 Dec 2021

Abstract

This note contains an overview of my Github repo `treedrawing` [1], which is a naive algorithm that neatly draws rooted trees. Then a proof of correctness of the algorithm is presented.

1 Definitions

We start by introducing some standard and non-standard terminologies:

Definition 1.1. *A **rooted tree** is a triple $T = (V, E, r)$ where (V, E) is a tree (a graph without cycles), and $r \in V$ is called the **root**.*

Remark. We adopt terminologies from computer science: we call V the set of **nodes** (instead of vertices), and E the set of **branches** (instead of edges).

Definition 1.2. *For nodes $x, y \in V$ connected by a branch, x is the **parent** of y and y is a **child** of x if x has closer path to the root than y.*

Definition 1.3. *A **leaf** is a node that does not have a child.*

Definition 1.4. *A non-root node is **minor** iff it is the only child of its parent and it has exactly one child node. A node is **major** iff it is the root or it is not minor.*

Definition 1.5. *The **sub-rooted tree** by a major node $x \in V$ is a rooted tree $T' = (V', E', x)$ consisting of x itself (as root) and all of its the 'descendant' nodes.*

Remark. (1) The sub-rooted tree by the root node is the whole rooted tree. (2) The sub-rooted tree by the leaf node is the trivial branch-less tree with itself as the only node.

Definition 1.6. *The **weight** of a major node is the number of major nodes of the sub-rooted tree by that node. The weight of a minor node is 0.*

Remark. The weight of every leaf node is 1.

We show the converse of the last sentence in Definition 1.6: all nodes of weight 0 are minor. This is equivalent to the statement that all major nodes have weights of at least 1. This is indeed the case because the major node itself is contained in the sub-rooted tree by itself. In conclusion, a node is minor iff its weight is 0.

To aid in understanding the concepts, an example rooted tree is provided in Figure 1.

2 Algorithm Description

The algorithm consists of three main functions that are executed in order: (1) `analyzeNode`, which records relevant information for each node recursively, (2) `setInitCoord`, which sets initial coordinates to each node recursively, and (3) `fixCoord`, which fixes coordinates of nodes so that there are no nodes with same location.

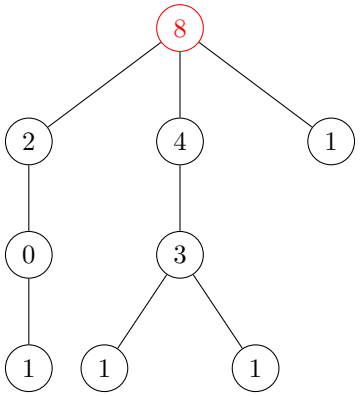

Figure 1: Example rooted tree with red-colored node as the root. The number in node indicate its weight.

2.1 Function `analyzeNode`

The main input of the algorithm is any data with a structure of rooted tree. Note that nodes in the input data must be *labelled*. What the `analyzeNode` function mainly does is to perform tree traversal on the data and record the following information:

- `connecNodes` is an array of arrays of natural numbers. An index `idx` in `connecNodes[idx]` corresponds to a node (with `idx = 0` being the root). Now `connecNodes[idx]` itself is the array of indices that correspond to the *children* of the `idx` node. If the `idx` node is a leaf, then `connecNodes[idx]` is the empty array.
- `labelDict` is a function that aligns node *labels* (strings) to indices for `connecNodes`, `weightNodes`, and `majorNodes`.
- `weightNodes` is an array of natural numbers. Like in `connecNodes`, an index `idx` in `weightNodes[idx]` corresponds to a node (with `idx = 0` being the root). Now `weightNodes[idx]` itself is the weight of the `idx` node.
- `majorNodes` is an array of objects. Like in `connecNodes`, an index `idx` in `majorNodes[idx]` corresponds to a node (with `idx = 0` being the root). Now for major nodes, `majorNodes[idx]` itself is the array of major nodes along the (unique) path from the root to the `idx` node (`idx` itself excluded). Since the root has no parent, `majorNodes[0]` is always the empty array. For minor nodes, `majorNodes[idx]` is the *null* object.

2.2 Function `setInitCoord`

[To be continued.]

2.3 Function `fixCoord`

[To be continued.]

3 Proof of Correctness

[To be continued.]

References

[1] poypoyan. treedrawing: naive rooted tree drawing algorithm. `https://github.com/poypoyan/treedrawing`.