

Inapplicable algorithm description

1 First downpass

1. Enter on any cherry (i.e. pair of tips) on the tree and move to its most recent common ancestor; **then**, *go* to 2.
2. **If** both node's descendants share at least one token in common, *go* to 3; **else**, *go* to 4.
3. **If** the state in common is only the inapplicable token and that any of the descendants has an applicable token, *set* the node's state to be the union between both descendants; **else**, *set* the node's state to the state in common between both descendants. **Then** *go* to 5.
4. **If** any of the descendants have an applicable token, *set* the node's state to be the union of its descendants without any inapplicable token; **else**, *set* the node state to be the inapplicable token only. **Then** *go* to 5.
5. **If** possible, move to the node's ancestor and *go* to 2; **else** move to the next unvisited cherry's ancestor and *go* to 2. Once all nodes have been visited, end the first downpass.

2 First uppass

1. Enter the tree on its root. **If** the root has any applicable token, *set* it's state to be the applicable token(s) only. Move to one of the root's descendants and *go* to 2.
2. **If** the node has an inapplicable token, *go* to 3; **else**, leave the node's state unchanged and *go* to 7.
3. **If** the node also has an applicable token, *go* to 4; **else**, *go* to 5.
4. **If** the node's ancestor has an inapplicable token, *set* the node's state to be the inapplicable token only; **else** *set* the node's state to be the applicable token(s) only. **Then** *go* to 7.
5. **If** the node's ancestor has an inapplicable token, *set* the node's state to be the inapplicable token only and *go* to 7; **else** *go* to 6.

6. **If** any of the descendants have an applicable token, the the node's state to be its descendants union without an inapplicable token; **else** *set* the node's state to be the inapplicable token only. **Then** *go* to 7.
7. **If** any of the node's descendant is not a tip, move to the next node and *go* to 2. **If** both descendants are tips, move to the closest non-visited node and *go* to 2. Once all nodes have been visited, end the first uppass.

3 Second downpass

1. Enter on any cherry on the tree and move to its most recent common ancestor; **then**, *go* to 2.
2. **If** there is any token in common between both descendants, *go* to 3; **else**, *go* to 4.
3. **If** any the token(s) in common are applicable, *set* the node's state to be the common applicable token(s) only; **else** *set* the node's state to be the inapplicable token only. **Then** *go* to 6.
4. *Set* the node's state to be the union of both descendants tokens without the inapplicable token and *go* to 5.
5. **If** both descendants have an applicable token, ***increment*** the tree length; **else if** both the descendants have are an active region, ***increment*** the tree length. **Then** *go* to 6.
6. **If** possible, move to the node's ancestor and *go* to 2; **else** move to the next unvisited cherry's ancestor and *go* to 2. Once all nodes have been visited, end the second downpass.

4 Second uppass

1. Enter the tree on its root and move to one of the root's descendants. **Then** *go* to 2.
2. **If** the node has any applicable token(s), *go* to 3; **else**, *go* to 11.
3. **If** the node's ancestor has any applicable token(s), *go* to 4; **else**, *go* to 10.

4. **If** the node's token(s) is the same as its ancestor, *go* to 12; **else**, *go* to 4.
5. **If** there is any token(s) in common between the node's descendants, *go* to 6; **else** *go* to 7.
6. **If** there is any token(s) in common between the node, its ancestor and both its descendants, *set* it (them) to be the node's token(s); **else** *set* the node to be the token(s) in common between its descendants. *The go* to 12.
7. **If** the union between the node's descendants have an inapplicable token, *go* to 8; **else** *go* to 9.
8. **If** there is any token(s) in common between the node's descendants and ancestor, *set* the node's token(s) to be its ancestor's; **else** *set* the node to be the union of the tokens from its descendants and ancestor without any inapplicable token. **Then** *go* to 12.
9. *Set* the node to be the union its tokens and its ancestor ones. **Then** *go* to 12.
10. **If** there is any tokens in common between both descendants, *set* the node's state to these tokens. *Go* to 12.
11. **If** both the descendants have are an active region, ***increment*** the tree length. *Go* to 12.
12. **If** any of the node's descendant is not a tip, move to the next node and *go* to 2. **If** both descendants are tips, move to the closest non-visited node and *go* to 2. Once all nodes have been visited, end the second uppass.

The tree length is then equal to the number of states changes and the number of additional applicable regions.