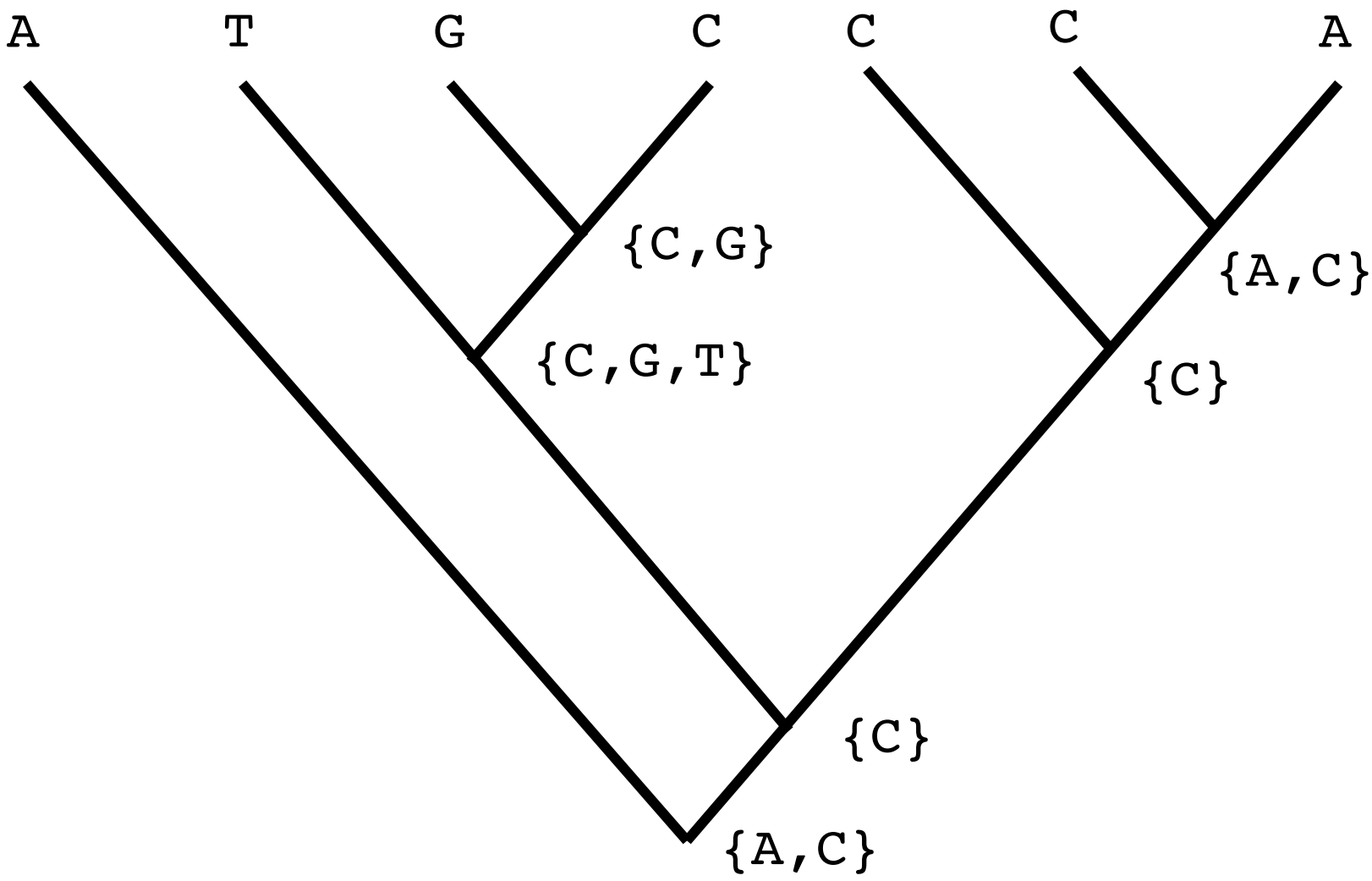
2. Phylogenetic networks

Trees and traversals

Fitch downpass (postorder)



Tree traversal

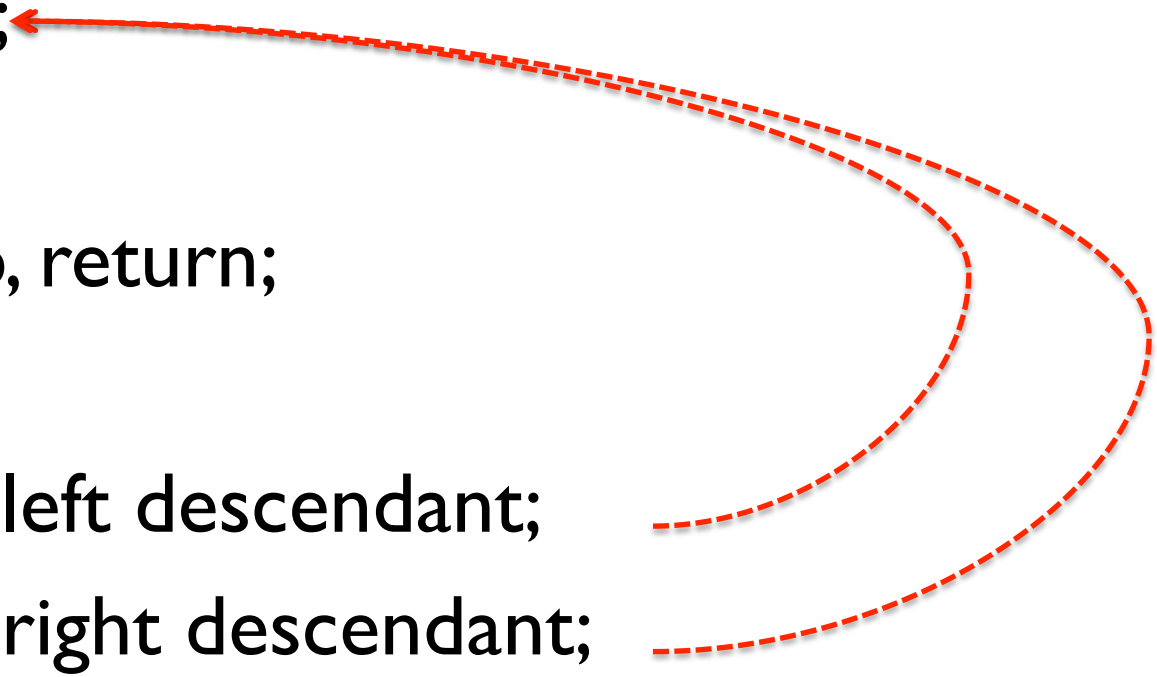
- An **recursive** algorithm that visits every tip and every node in a tree

- **Downpass (postorder) traversal**
  - Proceeds from the tips towards the root

- **Uppass (preorder) traversal**
  - Proceeds from the root to all of the tips.

Tree traversal

```
1 Begin at a node;
2 If :
3       node is a tip, return;
4 Else:
5       proceed up left descendant;
6       proceed up right descendant;
7 Return;
```



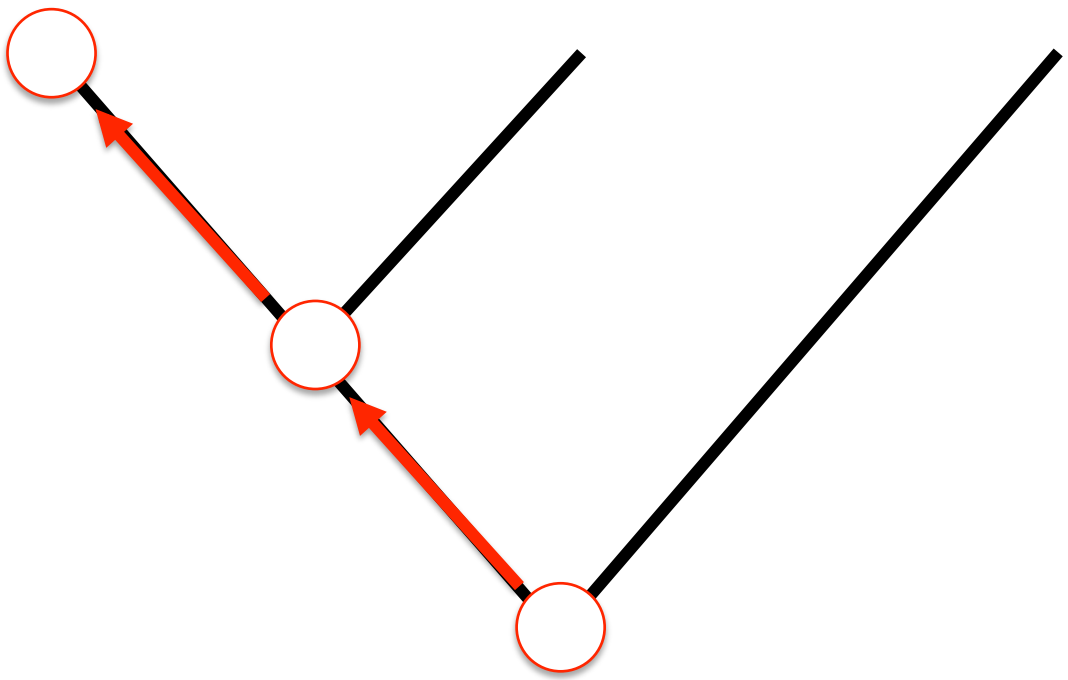
Uppass (preorder)

```
1 Begin at a node;
2 Some F(x)
3 If :
4       node is a tip, return;
5 Else:
6       proceed up left descendant, go to 1;
7       proceed up right descendant, go to 1;
8 Return;
```

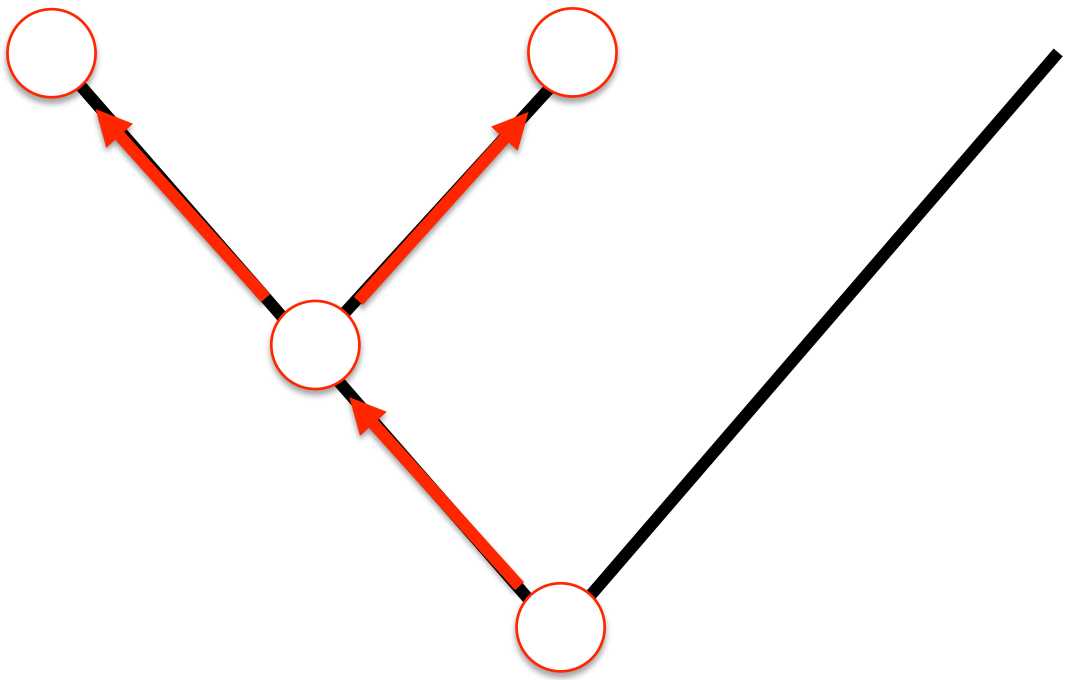
Downpass (postorder)

```
1 Begin at a node;
2 If :
3       node is a tip, return;
4 Else:
5       proceed up left descendant, go to 1;
6       proceed up right descendant, go to 1;
7  Some F(x)
8 Return;
```

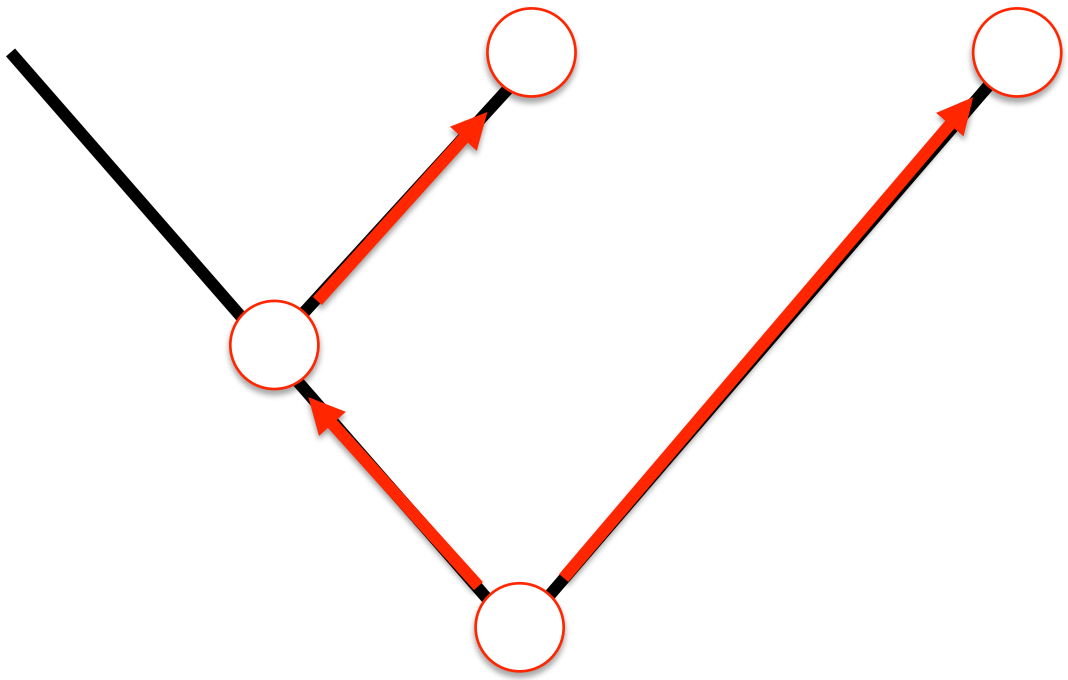
Tree traversal animation



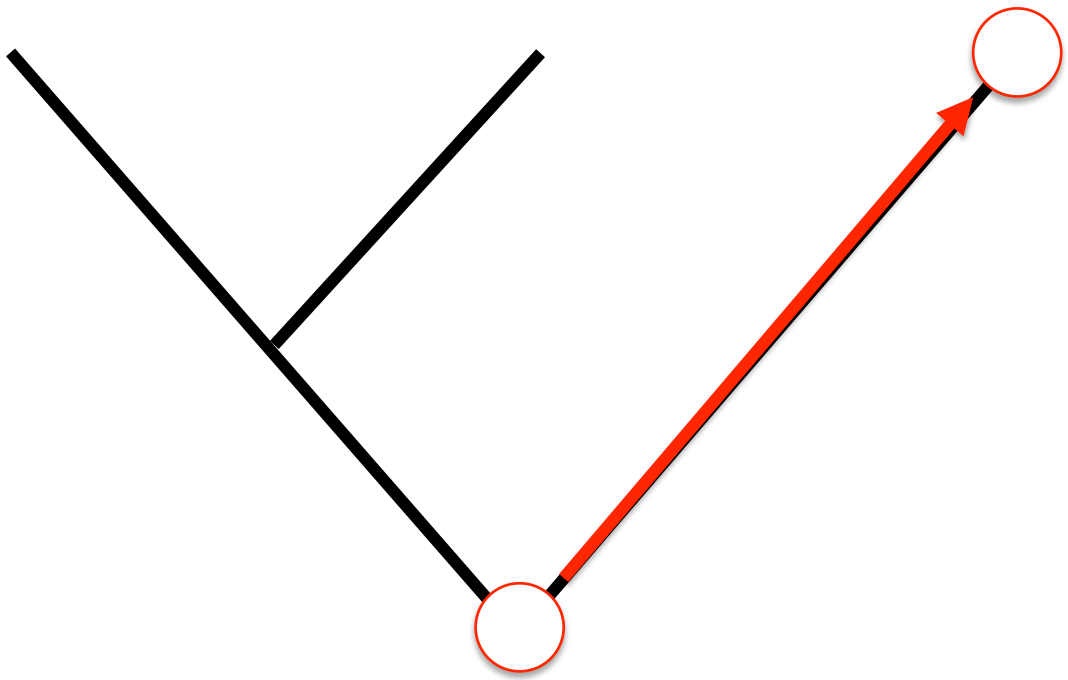
Tree traversal animation



Tree traversal animation



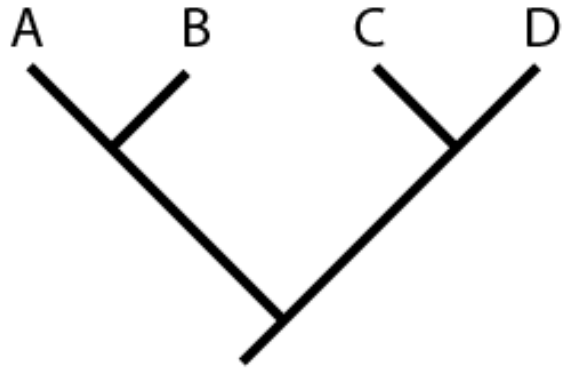
Tree traversal animation



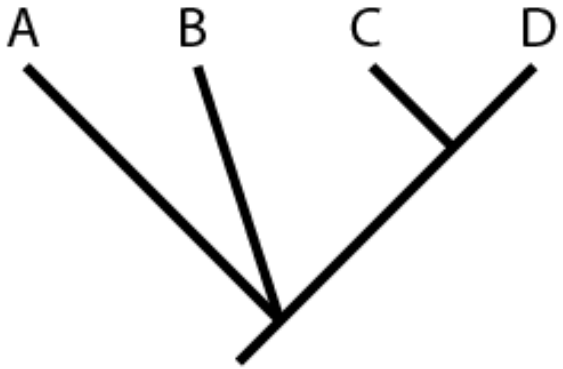
Tree traversals are used to…

- Optimise data on a tree
- Generate all tree topologies
- Perform all branch-breaking and reinsertion operations
- Anything else where you need to visit all nodes in a tree

Tree encoding

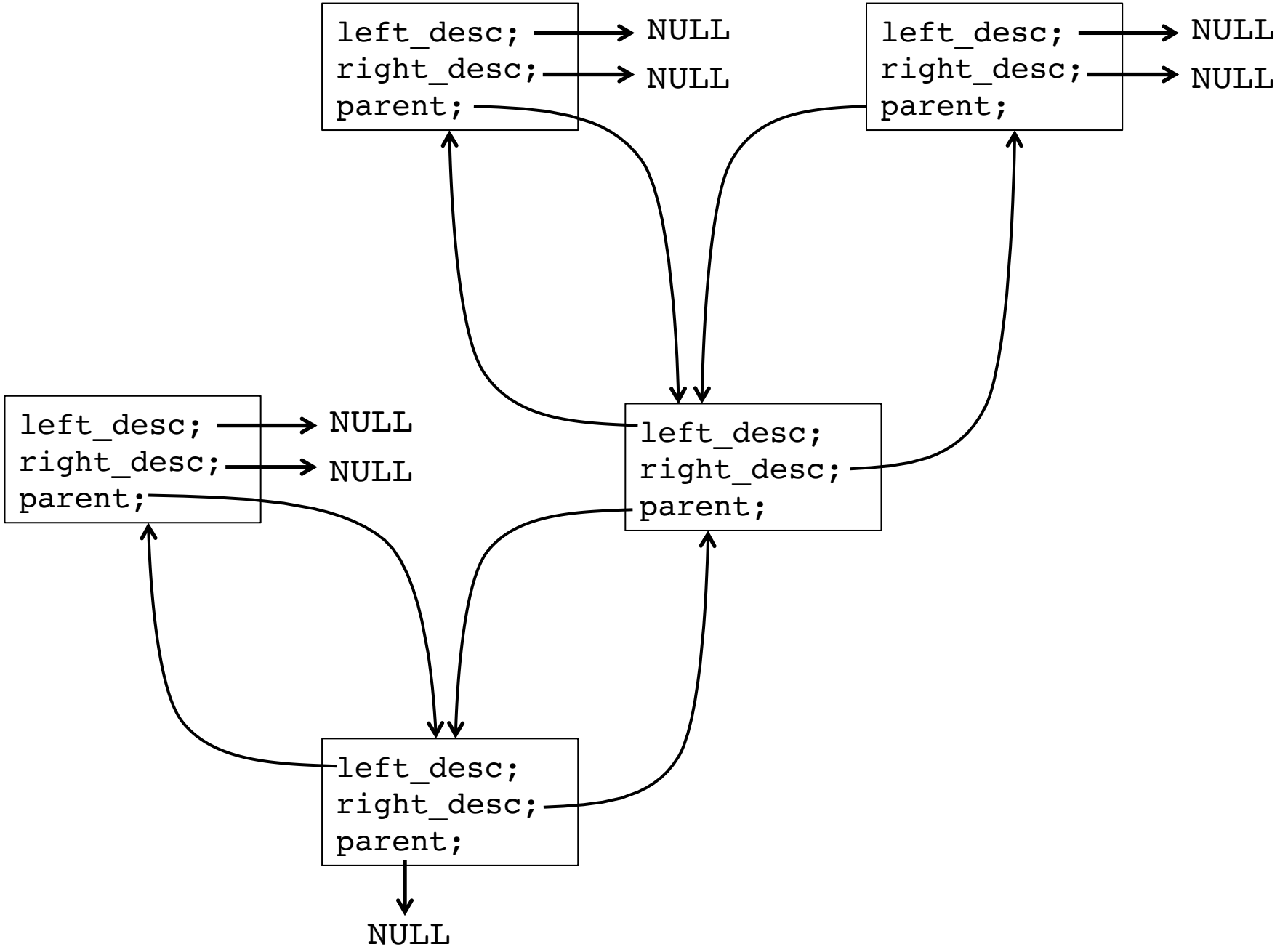



```
((A,B),(C,D));
```




```
(A,B,(C,D));
```

- Newick format
- Text based
- Hierarchical parentheses


left_desc;
right_desc;
parent;
NULL
NULL
left_desc;
right_desc;
parent;
NULL
NULL
left_desc;
right_desc;
parent;
NULL
NULL
left_desc;
right_desc;
parent;
left_desc;
right_desc;
parent;
NULL