Let *F* and *G* be two rooted trees with a left-to-right order among siblings and where each vertex is assigned a label from an alphabet∑. The *edit distance* between *F* and *G* is the minimum cost of transforming *F* into *G* by a sequence of elementary operations consisting of deleting and relabeling existing nodes, as well as inserting new nodes (allowing at most one operation to be performed on each node).

the alphabet in my code is terminals + nonterminals

Formally, given a node *v* in *F* with parent *v*’

*relabel* changes the label of *v*

*delete* removes a nonroot node *v* and sets the children of *v* as the children of *v*’ (the children are inserted in the place of *v* as a subsequence inthe left-to-right order of the children of *v*’)

*insert* (the complement of delete) connects a newnode *v* as a child of some *v*’ in *F* making *v* the parent of a consecutive subsequence of the children of *v*’.

The cost of the elementary operations is given by two functions, *c*del and *c*match , where *c*del (*τ* ) is the cost of deleting or inserting a vertex with label *τ* , and *c*match (*τ*1*, τ*2) is the cost of changing the label of a vertex from *τ*1 to *τ*2. Since a deletion in *F* is equivalent to an insertion in *G* and vice versa, we can focus on finding the minimum cost of a sequence of just deletions and relabelings in both trees that transform *F* and *G* into isomorphic trees.