Algorithm for the d8 method

1. Check that there are no conflicts in the fixage, minage or maxage nodes of the input tree. If so, terminate the program and display an error message.

2. Remove redundant constraints. This means that if for a minnode m_1 there exists a younger minnode m_2 whose minage exceeds the minage for m_1, then m_1 is not regarded as a minnode anymore. Also, if for a maxnode m_1 there exists a younger maxnode m_2 whose maxage exceeds the maxage for m_1, then m_2 is not regarded as a maxnode anymore.

3. If the root is not a fixage node then the root age a_r is estimated by a weighted average of fixage node ages multiplied by relative MPL's of the root and the fixage node, where the weights are proportional to the number of leaves of the fixage nodes. More specifically, suppose there are k adjacent fixage nodes of the root, where we by adjacent mean that there are no fixage nodes in between, and note that by assumption $k > 0$. Let p_r denote the MPL of the root and $p_1, \ldots, p_k$ denotes the MPL of the k fixage nodes having $n_1, \ldots, n_k$ leaves and fixages $a_1, \ldots a_k$. The estimated root age is then defined by

$$a_r = (n_1 \cdot a_1(p_r/p_1) + \ldots + n_k \cdot a_k(p_r/p_k))/(n_1 + \ldots + n_k).$$

   Check that the calculated root age a_r is not in conflict with any of the fixages $a_1, \ldots, a_k$. If it is, set a_r equal to the oldest of $a_1, \ldots, a_k$ and set all undated nodes in between these nodes equal to a_r. Treat the root as a fixage node with age a_r from now on.

4. Go through the tree, starting next to the root and moving towards the leaves, and calculate the age estimate of a given (non-fixage) node x as follows. Let 0 be the fixage node closer to the root of x and a_0 its age, and let $a_1, \ldots, a_k$ be the ages of fixage nodes between x and the leaves (k can be 0 or positive). Let n_x be the number of paths from x to the leaves without passing through any fixage node, and let n_i denote the number of leaves of fixage node $i, i = 1, \ldots, k$. The age a_x is estimated by a weighted average of the relative age of x and the root, and of the age of each fixage node i plus the relative additional age of x from the fixage node up to 0, where the weights are proportional to the number of paths going through the fixage nodes. More precisely we have

$$a_x = (n_x \cdot a_0(p_x/p_0) + s_1 + \ldots + s_k)/(n_x + n_1 + \ldots + n_k),$$

   where

$$s_i = n_i \cdot (a_i + (a_0 - a_i) \cdot (p_x - p_i))/(p_0 - p_i), 1 \leq i \leq k.$$

   Check that a_x is older than $a_1, \ldots, a_k$. If not set a_x equal to the oldest of them and set all nodes in between equal to the same age. Check also that all estimated ages between x and its leaves are younger than a_x. If not, change the older age estimates to equal a_x.

5. For all minage and maxage nodes, check that the calculated age estimates are not in conflict with them. If there are conflicts, set all violated node ages equal to the constrained minage or maxage and treat these modified node ages as fixage nodes and redo steps 3 and 4 treating the original fixage nodes and the new fixage nodes as original fixage nodes and all other node ages as unknown.