Let \((T; \{V_t \mid t \in T\})\) be the given tree decomposition rooted at \(r\). There are \(k\) \((s_i; t_i)\) terminal pairs. We focus on the tree-width 2 case. For convenience, we assume that there are no two pieces \(V_{t_1}\) and \(V_{t_2}\) where \((t_1, t_2)\) is an edge and \(V_{t_1} \subset V_{t_2}\). Consider the subgraph \(G_t\). Note that there can be at most one \(i\) for which \(P_i\) both enters and leaves \(G_t\), since any such path uses up at least 2 vertices of \(V_t\). Note also that there can be at most 3 \(s_i; t_i\) terminal pairs that have one end in \(G_t\) and the other one outside (as the paths connecting such pairs must go through \(V_t\).

- If there are 3 such pairs, that each node \(v \in V_t\) must be connected via disjoint paths to one of them, and terminal pairs inside \(G_t\) must connect via paths inside \(G_t\). There are \(O(1)\) cases here to consider depending on which of the nodes in \(V_t\) is used to connect which of the 3 separated terminal pairs.

- If there are 2 such terminal pairs, than of the at most 3 nodes in \(V_t\) 2 must be connected via disjoint paths to one of them, the third is either not used in any of the paths or is used by path connecting two terminals \(s_i\) and \(t_i\) inside, or two terminals outside \(G_t\). Now there are \(O(k)\) cases to consider, depending on which terminal pair \(i\) is using the extra node.

- If there is only one such pair, than we can have one pair \(i\) with both terminals inside or outside of \(G_t\), that uses 2 nodes in \(V_t\), or the one path leaving \(G_t\), can leave, come back and leave again, or one or two paths can use just one node in \(V_t\) while having both terminals inside or both outside of \(G_t\). There are \(O(k^3)\) cases to consider here.

- If there are no such pairs, than one path can use 2 or 3 nodes in \(V_t\), or multiple paths can use one node each. Now there are \(O(k^3)\) cases to consider.

We define multiple subproblems for each \(t\) according to the possibilities discussed above. For at most 3 nodes in \(V_t\) there are at most \(O(k^3)\) possible cases. This defines \(O(k^3n)\) subproblems. The value of a subproblem is simply 0 or 1 (or true or false) depending whether or not there are disjoint paths in \(G_t\) that satisfy the state of the nodes in \(V_{\bar{t}}\) corresponding to the subproblem, that is, connect each \(v \in V_t\) to the terminal in question inside \(G_t\) (and possibly connect the two nodes in \(V_t\) to each other, if needed), via disjoint paths inside \(G_t\). The desired disjoint paths exists if and only if the value of one of the subproblems that connects all terminal pairs within the subgraph \(G_t\) (which is the whole graph).

Given values for all the subproblems associated with the children \(t_1, \ldots, t_d\) of a node \(t\), we want to get the value of the given subproblem efficiently. To do this consider a node \(v \in V_t\), the subproblem under question wants a particular paths \(P_i\) to go through this vertex in \(O(d)\) time.

\footnote{ex209.650.476}