Randomized Algorithms. Exercises for 20.10

October 19, 2009

A *(network) topology* is a rooted directed acyclic graph in which exactly one vertex has indegree 0 and outdegree 2 (the root) and all other vertices have either indegree 1 and outdegree 2 (*split vertices*), indegree 2 and outdegree 1 (*reticulation vertices*) or indegree 1 and outdegree 0 (*leaves*). A *(phylogenetic) network* $\mathcal{N} = (N, \gamma)$ on X is an ordered pair consisting of a topology N on X and a *labelling* γ of N, that is, a a bijective map from the leaf set L^N of N to X. Clearly, $|X| = |L^N|$. For ease of terminology, we will sometimes call X the *leaf-label set* of $\mathcal{N}$. Unless stated otherwise, throughout the paper the range of a labelling of a topology is X. In case there is no confusion, we will refer to a vertex or edge of N as a vertex or edge of $\mathcal{N}$, respectively.

Calling two distinct phylogenetic trees (T_1, γ_1) and (T_2, γ_2) *isomorphic* if there exists a bijection $\phi : V(T_1) \to V(T_2)$ which induces a bijection between $E(T_1)$ and $E(T_2)$ such that $\gamma_2 = \phi \circ \gamma_1$, it can be easily checked that there are precisely 3 non-isomorphic (rooted) triplets on $Y := \{x, y, z\}$. Following common practice, we will denote the unique triplet on Y in which the parent vertex of x and y is a proper descendant of the parent vertex of x and z by $xy|z$ or, equivalently, by $yx|z$. Further, for any set T of triplets, we denote by $X(T)$ the union of the leaf-label sets over all triplets in T. For ease of terminology, we say for a triplet t whose leaf-label set is contained in X that t *is on* X.

Obviously by deleting edges and suppressing resulting degree 2 vertices, any level-k network on X gives rise to a collection of triplets. However the converse need not hold in general. We therefore define a a triplet $xy|z$ on X to be *consistent* with a network $\mathcal{N}$ on X (interchangeably: $\mathcal{N}$ is consistent with $xy|z$) if $\mathcal{N}$ contains a subdivision of $xy|z$. In other words, if $\mathcal{N}$ contains vertices $u \neq v$ and pairwise internally vertex-disjoint paths $u \to x$, $u \to y$, $v \to u$ and $v \to z$[1]. More generally, we define a set T of triplets to be *consistent* with a network $\mathcal{N}$ (interchangeably: $\mathcal{N}$ is consistent with T) if every triplet $t \in T$ is consistent with $\mathcal{N}$.

ex.1 For a given triplet set give an algorithm that returns a tree consistent with 1/3 of the triplets.

ex.2 Given a triplet set and a tree-topology give an algorithm to produce a labeling consistent with a big fraction of triplets.

[1]Where it is clear from the context, as in this case, we may refer to a leaf of a phylogenetic network by the element in X that it is labelled by.

ex.3 Show that there exist sparse triplet sets (with $|T| = O(|X|^2)$) such that there is no tree consistent with $1/3 - \epsilon$ fraction of the triplets.

ex.4 Give a randomized LP-rounding algorithm for the (general) facility location problem.