- for a set of taxa M, we use T^*_M to denote an optimum phylogeny on M

We say that an edge e *mutates* character i if $i \in \mu(e)$. We will use the following well known definition and lemma on phylogenies.

Definition 7: Given matrix I, the *set of gametes* $G_{i,j}$ for characters i, j is defined as: $G_{i,j} = \{(r[i], r[j])|r \in R(I)\}$. Two characters i, j *share* t gametes in I i.f.f. $|G_{i,j}| = t$.
In other words, the set of gametes $G_{i,j}$ is a projection on the i, j dimensions.

Lemma 2.1: [14] An optimum phylogeny for input I is not perfect i.f.f. there exists two characters i, j that share (all) four gametes in I.

Definition 8: **(Conflict Graph [17])**: A *conflict graph* G for matrix I with character set C is defined as follows. Every vertex v of G corresponds to unique character $c(v) \in C$. An edge (u, v) is added to G i.f.f. $c(u), c(v)$ share all four gametes in I. Such a pair of characters are defined to be in *conflict*.

Note that if the conflict graph G contains no edges, then a perfect phylogeny can be constructed for I. Gusfield [14] provided an efficient algorithm to reconstruct a perfect phylogeny in such cases.

Simplifications: We assume that the all zeros taxon is present in the input. If not, using our freedom of labeling, we convert the data into an equivalent input containing the all zeros taxon (see section 2.2 of Eskin et al [9] for details). We also remove any character that contains only one state. Such characters do not mutate in the whole phylogeny and are therefore useless in any phylogeny reconstruction. The BNPP problem asks for the reconstruction of an unrooted tree. For the sake of analysis, we will however assume that all the phylogenies are rooted at the all zeros taxon.

III. SIMPLE ALGORITHM

This section describes a simple algorithm for the reconstruction of a binary near-perfect phylogenetic tree. Throughout this section, we will use the first definition of a phylogeny (Definition 1).

We begin by performing the following pre-processing step. For every pair of characters c', c'' if $|G_{c',c''}| = 2$, we (arbitrarily) remove character c''. After repeatedly performing the above step, we have the following lemma:

Lemma 3.1: For every pair of characters c', c'', $|G_{c',c''}| \geq 3$.