$\mathcal{G}$. A *circle* is a cycle that visits no node twice except for the starting node (which is the end node at the same time). For instance, in Figure 2 the sequence of vertices T, GT, A, TG, T is a circle while the sequence T, GT, A, GT, A, TG, T is a cycle that is not a circle.

Lemma 2 (Fimmel, Michel, Strüngmann, 2015). *Let $X \subseteq B^n$ be a circular code. Then its representing graph is a simple oriented graph without circles of length* 2.

Let us remark that the above Lemma 2 simply says that for circular codes the representing graph has an *underlying* simple unoriented graph.

Example 2. *In Figure 4 we show two examples of trinucleotide codes and their representing graphs. The code $\{ATG, CAC, CAT, GTG\}$ (left) is circular and has a simple graph, while the code $\{ATG, ATT, TGA, TGT\}$ (right) is non-circular and its graph is not simple*

Bad.

Good

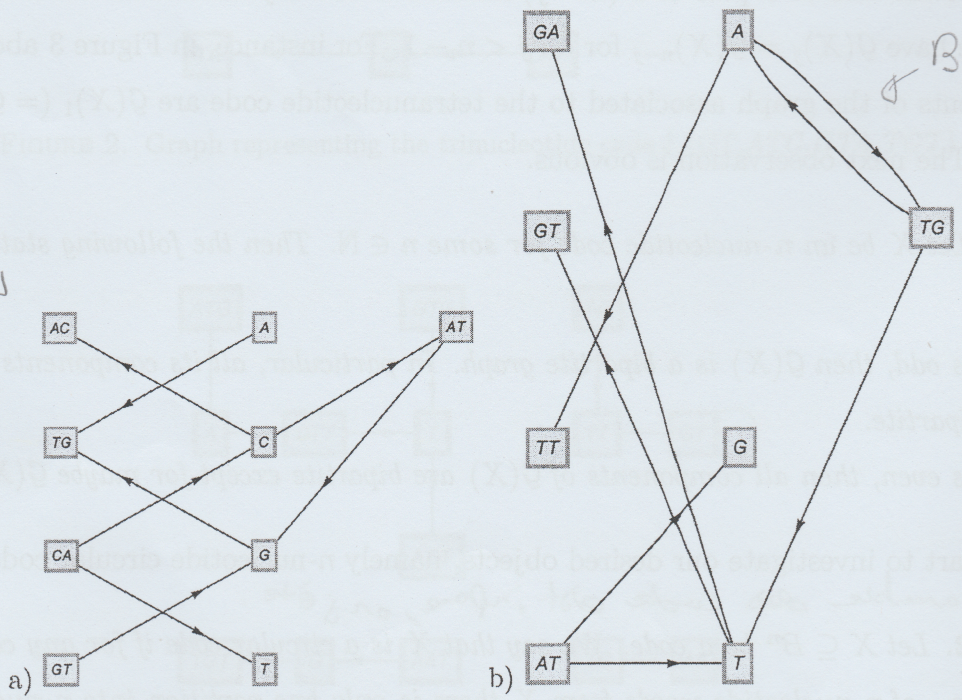

FIGURE 4. a) The trinucleotide code $\{ATG, CAC, CAT, GTG\}$ is circular and has a simple graph, b) The trinucleotide code $\{ATG, ATT, TGA, TGT\}$ is non-circular and its representing graph is not simple.

We now state our first main theorem which proves the connection between circularity of codes and acyclicity of graphs. Recall from graph theory (Clark and Holton, 1991) that a graph is called *acyclic* if it does not contain cycles, i.e. oriented closed paths.

Theorem 1 (Fimmel, Michel, Strüngmann, 2015). *Given a code $X \subseteq B^n$ the following statements are equivalent:*