# homework_9_Baoshu Feng

March 11, 2022

## 1 Q1

1. Let $M$ and $M'$ be perfect matchings in a tree. Form the symmetric difference of the edge sets, $M \Delta M'$.
2. The perfect matchings cause each vertex has degree 0 or 2 in the symmetric difference. Therefore, every component is an isolated vertex or a cycle.
3. We had tree that has no cycle. Every vertex must have degree 0 in the symmetric difference, which means that the two matchings are the same.

## 2 Q2

1. When $M$ is a maximal matching, then the vertices saturated by $M$ form a vertex cover. In the vertex cover, the edge that had no vertex in set could be added to $M$ .
2. Since every vertex cover has size at least $\alpha'(G)$, we obtain $2|M| \geq \beta(G) \geq \alpha'(G)$.

## 3 Project Proposal

In this project, we cover register allocation and assignment, which are among the most important compiler optimizations for almost all computer architectures. The problem addressed is how to minimize traffic between the CPU registers, which are usually few and fast to access, and whatever lies beyond them in the memory hierarchy, including one or more levels of cache and main memory, all of which are slower to access and larger, generally increasing in size and decreasing in speed the further we move away from the registers.

Register allocation is best carried out on low-level intermediate code or assembly language. It is essential that all loads from and stores to memory, including their address computations, be represented explicitly. The project's central focus, global register allocation by graph coloring, usually results in very effective allocations without a major cost in compilation speed. It views the fact that two quantities must be in registers simultaneously as excluding them from being in the same register. It represents the quantities by nodes in a graph and the exclusions (called interferences) by arcs between the corresponding nodes; the nodes may represent real registers also, and the arcs may represent exclusions such as the base address in a memory access may not be the register r0. Given the graph corresponding to an entire procedure, this method then attempts to color the nodes, with the number of colors equal to the number of available real registers. Every node is assigned a color that is distinct from those of all the nodes adjacent to it. If this cannot be achieved, additional code is introduced to storequantities to memory and to reload them as needed, and the process is repeated until a satisfactory coloring is achieved. As we will see, even very simple formulations of graph-coloring problems are NP-complete, so one of the most important facets of making global register allocation as effective as possible is using highly effective heuristics.

This project will review the register allocation strategy of the state-of-the-art modern compiler, LLVM. We will implement a graph coloring algorithm and benchmark its efficiency.