# Algorithms and Data Structures (WS15/16)

Example Solutions for Unit 28

## Problem 1

### Part i)

Let us consider the symmetric difference of $M_1$ and $M_2$, that is, the set $G' = (M_1 \cup M_2) \setminus (M_1 \cap M_2)$. This set contains all the edges that are either in $M_1$ or $M_2$, but not in both. The degree of each vertex in $G'$ is at most 2, since the degree in each of $M_1$ and $M_2$ is at most 1. Therefore, $G'$ will be a collection of paths and cycles. Furthermore, we know that the cycles must have even length, since the graph is bipartite, and that all paths and cycles are alternating, that is, each pair of consecutive edges belongs to different matchings.

We now show how to select a subset of $G'$, which we denote by $M'$, such that $M' \cup (M_1 \cap M_2)$ satisfies the requirements in the statement. There are three possibilities for each connected component of $G'$:

- **Cycle**: Since each vertex has degree 2, then all vertices in the cycle is matched by $M_1$ and $M_2$. Therefore, by adding to $M'$ all the edges in the cycle that belong to $M_1$, we match the same vertices that $M_1$ and $M_2$.
- **Path of odd length**: W. l. o. g. we assume that the path starts (and ends) with an edge in $M_1$. Then, this path will have one more edge in $M_1$ than in $M_2$, and furthermore, all vertices that are matched by an edge in $M_2$ will also be matched by $M_1$, since they are not the first or last vertices. Therefore, by picking all the edges in $M_1$, and adding them to $M'$, we match all vertices in $M_1$ and $M_2$.
- **Path of even length**: We first assume that the path starts and ends in $A$. The first and last edges are in different matchings. We say that the last edge is the one in $M_2$. Now, if we add all the edges in $M_1$ to $M'$, the last vertex will not be matched, since it is only matched by an edge in $M_2$. However, this vertex is not in $M_1 \cap A$ (as otherwise it would be matched in $M_1$ and the path would not end). Therefore, adding all the edges in $M_1$ satisfies the requirements, since it matches every vertex except the last. Now, for the case in which the path starts and ends in $B$, we pick all the edges in $M_2$ and the same reasoning applies (since the first vertex is not in $M_2 \cap B$).

We conclude that the matching $M'$ matches all of the required vertices in $(M_1 \cup M_2) \setminus (M_1 \cap M_2)$, and consequently, $M' \cup (M_1 \cap M_2)$ matches all the required vertices in $(M_1 \cup M_2)$.

### Part ii)

Let $M'$ be any maximum matching for $G$. We use part i) to obtain a matching $M''$ that matches all of the vertices in $M \cup A$ and in $M' \cup B$. Since $M''$ matches all the vertices in $M' \cup B$, it must be at least as large as $M'$, so it must be a maximum matching. We repeat this, but now we obtain a matching $M^*$ that matches all the vertices in $M'' \cup A \supseteq M \cup A$ and in $M \cup B$. By the same reasoning, $M^*$ is a maximum matching and matches all the vertices in $M$.

Alternatively, we can prove this by considering augmenting paths in $M$. If $M$ has no augmenting path, then it is already a maximum matching, so we are done. Otherwise, any augmenting path will be an odd length path between $A$ and $B$, starting and ending with an edge not in $M$. By replacing all the edges in the path that are in $M$ by all not in $M$, we match all the vertices that were previously matched, and additionally match two more vertices that were not matched in $M$. Therefore, the maximum matching obtained by augmenting matching $M$ still matches all of the vertices in $M$, since every step of the process matches additional vertices without unmatching any vertex.

## Problem 2

- $\Rightarrow$ If Player 1 has a winning strategy, then $G$ does not have a perfect matching.

  We prove the contraposition, if $G$ has a perfect matching, then Player 2 has a winning strategy. Let $M$ denote any perfect matching in $G$. In every turn Player 2 will simply pick the vertex that is matched to Player 1 last picked vertex in $M$. As $M$ is a perfect matching there will always be such a choice.

- $\Leftarrow$ If $G$ has no perfect matching, then Player 1 has a winning Strategy. Let $M'$ denote a maximum matching in $G$. Player one will start the game by picking a vertex that is not matched in $M'$ (such a vertex must exist, as $M'$ can not be perfect). In his following turns Player 1 will always pick the vertex that is matched to Player 2's last pick in $M'$. Clearly Player 2 can only win, if he at some point picks a vertex that is not matched in $M'$. The existence of such a vertex would contradict the maximality of $M'$ as there would then exist an augmenting path from Player 1's first picked vertex to Player 2's last picked vertex. Therefore Player 2 can not win the game.

# Exercises for Unit 28

(3.) M... maximum matching, size m

C... minimum vertex cover, size c

**$c \geq m$** a vertex cover cannot cover more than one edge from a matching with one vertex $\Rightarrow c \geq m$

**$m \geq c$** we know how to get a maximum matching in a bipartite graph by turning the graph into a flow network $\longrightarrow$ max-flow in this network is m and it consists of m vertex-disjoint paths (the paths are, of course, also edge-disjoint ...)

$\Rightarrow$ by removing one vertex from each of the m vertex-disjoint paths, we can disconnect s and t which in particular means that these m vertices are touching all the edges of our bipartite graph $\Rightarrow m \geq c$

$$\Rightarrow m = c$$

remark: if we want to "explicitly" find a vertex cover from the max. matching, we go through the same process as in the proof of Hall's thm. presented in class

$\hookrightarrow$ start with an $(s,t)$-cut that is minimal - we can transform it into a min-cut whose cut edges either leave s or enter t

- now, it is easy to see that $(A \cap T') \cup (B \cap S')$ is a vertex cover (with cardinality m)