CS180, Winter 2018
Homework 2
Problems 1.5,1.8,2.4,2.5

Derek Xu

January 25, 2018

1 Problem 1.5

(a) Given:
Strong instability in a perfect matching S consists of a m and w s.t. each m and w prefers the other to their current partner in S.
We can modify the Gale-Shapely Algorithm a little to get an algorithm to solve this problem:

Let m ϵ M and w ϵ W be unmatched sets of size n
While there exists mϵM s.t. m has no pair:
    Let m' be a mϵM who has proposed the least number of times
    Let m' propose to the highest w'ϵW on his list.
    If w' is free:
        Let (m',w') be a pair in set, S
    If w' is not free:
        If w' strictly prefers m' to her current partner:
            Free the current pair with w' in S
            Let (m', w') be a pair in set, S
Return S

Proof of correctness (by contradiction):
Let there be a strong instability.
This means that there exists a pair (m, w) and (m', w') where m prefers w' and vice versa, or m' prefers w and vice versa. Because these two situations are symmetric, if one of them reaches a contradiction, the other does too for the same reason. Examine the former situation. Two cases arise.
Case 1:
w or m is unpaired. This is false because the while loop will only exit once all $m\epsilon M$ are paired. If all $m\epsilon M$ are paired and each m can only be paired with one w (as whenever a new pair is formed $w\epsilon W$ either leaves her current pair or never had a pair) and there are n $m\epsilon M$ and n $w\epsilon W$, all $w\epsilon W$ must also be paired. Thus this case cannot be true
Case 2:
$w'\epsilon W$ prefers $m\epsilon M$ over her current partner and $m\epsilon M$ prefers $w'\epsilon W$ over his current partner. This is false, because had m prefer w' to his current partner, he would have proposed to her before being matched with his current partner. This is because any $m''\epsilon M$ proposes to $w''\epsilon W$

from his first choice until his last choice. If w' had been proposed to by m, she would have accepted him, because w' prefers m to her current partner, satisfying both if statements. Thus this case cannot be true.
$\therefore$ The algorithm is correct.
$\therefore$ There always exist a perfect matching.

(b) Given:
Weak instability in a perfect matching S consists of a m and w s.t. either m or w prefers the other to their current partner in S.
There does not always exist a perfect matching. Consider the following counter example:
Let the following elements constitute a set M and W: m_1, m_2, w_1, w_2
Let both m_1 and m_2 prefer w_1 to w_2
Let w_1 prefer m_1 and m_2 equally
Observe all possible pairings that exist:
$(m_1, w_1), (m_2, w_2)$
This has weak instability as m_2 prefers w_1 to w_2 and w_1 holds no preference between m_1 and m_2
$(m_1, w_2), (m_2, w_1)$
This has weak instability as m_1 prefers w_1 to w_2 and w_1 holds no preference between m_1 and m_2
In this case, no possible pairing can be made such that no weak instability exists $\therefore$ there does not always exist a perfect matching.

2 Problem 1.8

Switching would improve the partner of wϵW. Consider the following example:
Let the following be a set of M and W: $m_1, m_2, m_3, m_4, w_1, w_2, w_3, w_4$
Let the preference list of m_1 be:
$w_1 > w_2 > w_3$
Let the preference list of m_2 be:
$w_1 > w_2 > w_3$
Let the preference list of m_3 be:
$w_2 > w_1 > w_3$
Let the preference list of w_1 be:
$m_3 > m_1 > m_2$
Let the preference list of w_2 be:
$m_1 > m_3 > m_2$
Let the preference list of w_3 be:
$m_3 > m_1 > m_2$
..where $a > b$ represents that a is preferred over b
..where the order that the men propose for the first round is: m_3, m_2, m_1
Running through the Gale-Shapely Algorithm we obtain the following result:
$(m_1, w_1), (m_2, w_3), (m_3, w_2)$
Now, we change the preference list of w_1 to the following list (assume that the previous order is w_1's actual ranking):
$m_3 > m_2 > m_1$
Running through the Gale-Shapely Algorithm again we obtain the following answer:
$(m_1, w_2), (m_2, w_3), (m_3, w_1)$
Note, the second iteration of the algorithm give a better pairing for w_1.
$\therefore$ lying about her preference list can improve the pairing of the woman.

3 Problem 2.4

The following list of functions are arranged in ascending growth rate:
Analysis of g_1 vs g_3:
$g_1(n) = 2^{\sqrt{logn}}$
$log(g_1(n)) = \sqrt{logn} \cdot log(2)$
$g_3(n) = n(logn)^3$
$log(g_3(n)) = log(n(logn)^3)$
$= log(n) + log(log(n)^3)$
$\therefore g_1(n) = O(g_3(n))$

Analysis of g_4 vs g_5 vs g_2:
By L'Hoptial's Rule:
$\lim_{n\to\infty} \frac{n^{log(n)}}{2^n} = 0$
$\therefore g_5(n) = O(g_2(n))$
$\lim_{n\to\infty} \frac{n^{4/3}}{n^{log(n)}} = 0$
$\therefore g_4(n) = O(g_5(n))$

The order is thus: $g_1(n), g_3(n), g_4(n), g_5(n), g_2(n), g_7(n), g_6(n)$

4 Problem 2.5

(a) False, Counter-example:
Let $f(n) = 99$
and $g(n) = 1$
Note: $f(n) = O(g(n))$
Now we substitute values:
$log_2(f(n)) = log_2(99) > 0$ for n$\epsilon\mathbb{R}$
$log_2(g(n)) = 0$ for n$\epsilon\mathbb{R}$
$\lim_{n\to\infty} \frac{0}{log_2(99)} = 0$
In this case:
$log_2(f(n)) \neq O(log_2(g(n)))$

(b) False, Counter-example:
Let $f(n) = 99n$
and $g(n) = n$
Note: $f(n) = O(g(n))$
Now we substitute values:
$2^{f(n)} = (2^{99})^n$ for n$\epsilon\mathbb{R}$
$2^{g(n)} = 2^n$ for n$\epsilon\mathbb{R}$
$\lim_{n\to\infty} \frac{2^n}{(2^{99})^n} = 0$
In this case:
$2^{f(n)} \neq O(2^{g(n)})$

(c) True, Proof:
$f(n) = O(g(n))$
$f(n) \leq c \cdot g(n)$
$f(n)^2 \leq (c \cdot g(n))^2$
$f(n)^2 \leq c' \cdot g(n)^2$
...where $c' = c^2$
$\therefore f(n)^2 = O(g(n)^2)$