Lecture 10 Date 1 Feb 2024

Stable Matching

Two set S and C. $|C| = |S|$.

students (S), colleges (C)

$S = \{s_1, \dots, s_n\}$ $C = \{c_1, \dots, c_n\}$

Each student $s \in S$ has a preference list.

$c_{i_1} > c_{i_2} > \dots > c_{i_n}$. $>_s$ $c_{21} >_s c_5$

Each college $c \in S$ has a preference list.

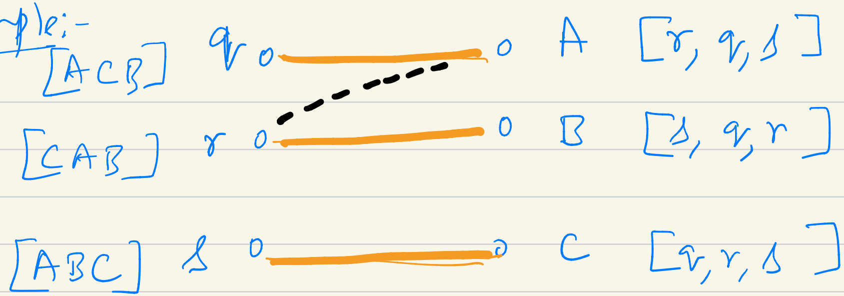
$s_{i_1} > s_{i_2} > \dots > s_{i_n}$

Defn:- A matching M is a pairing of S & C.

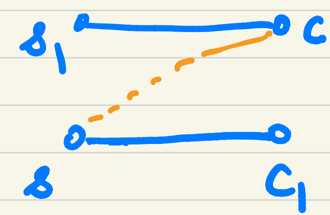
[Note:- In GT the word matching is used differently]

JE

Example:-

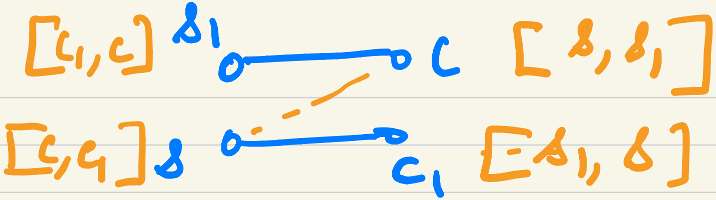


Def^n:- Suppose in a matching M s is paired with c_1 and c is paired with s_1. We say that (s, c) is an unstable pair if

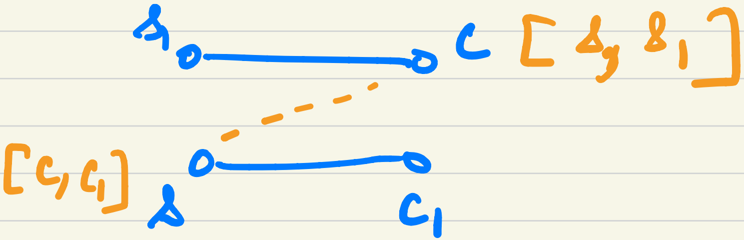


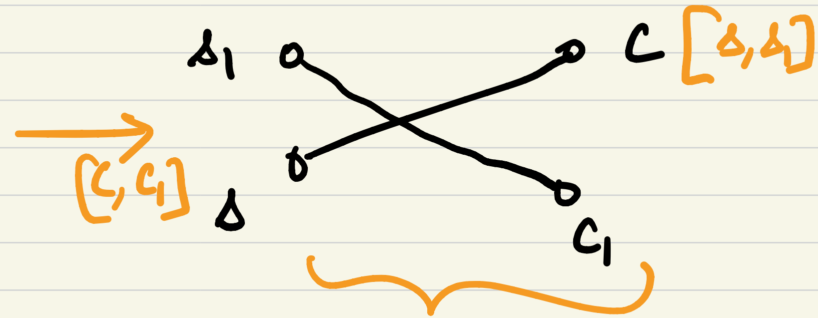
$$c >_s c_1 \quad \& \quad s >_c s_1$$

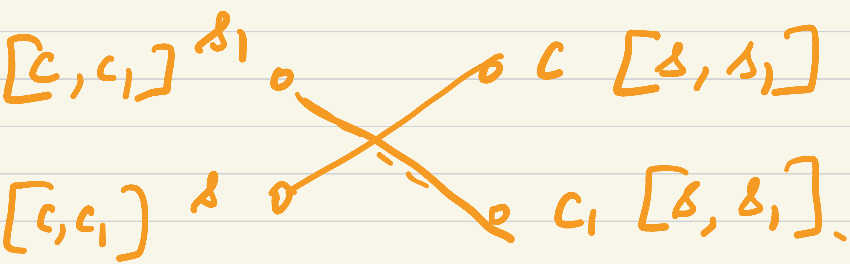
Def^n: A matching is called stable if it has no unstable pair.

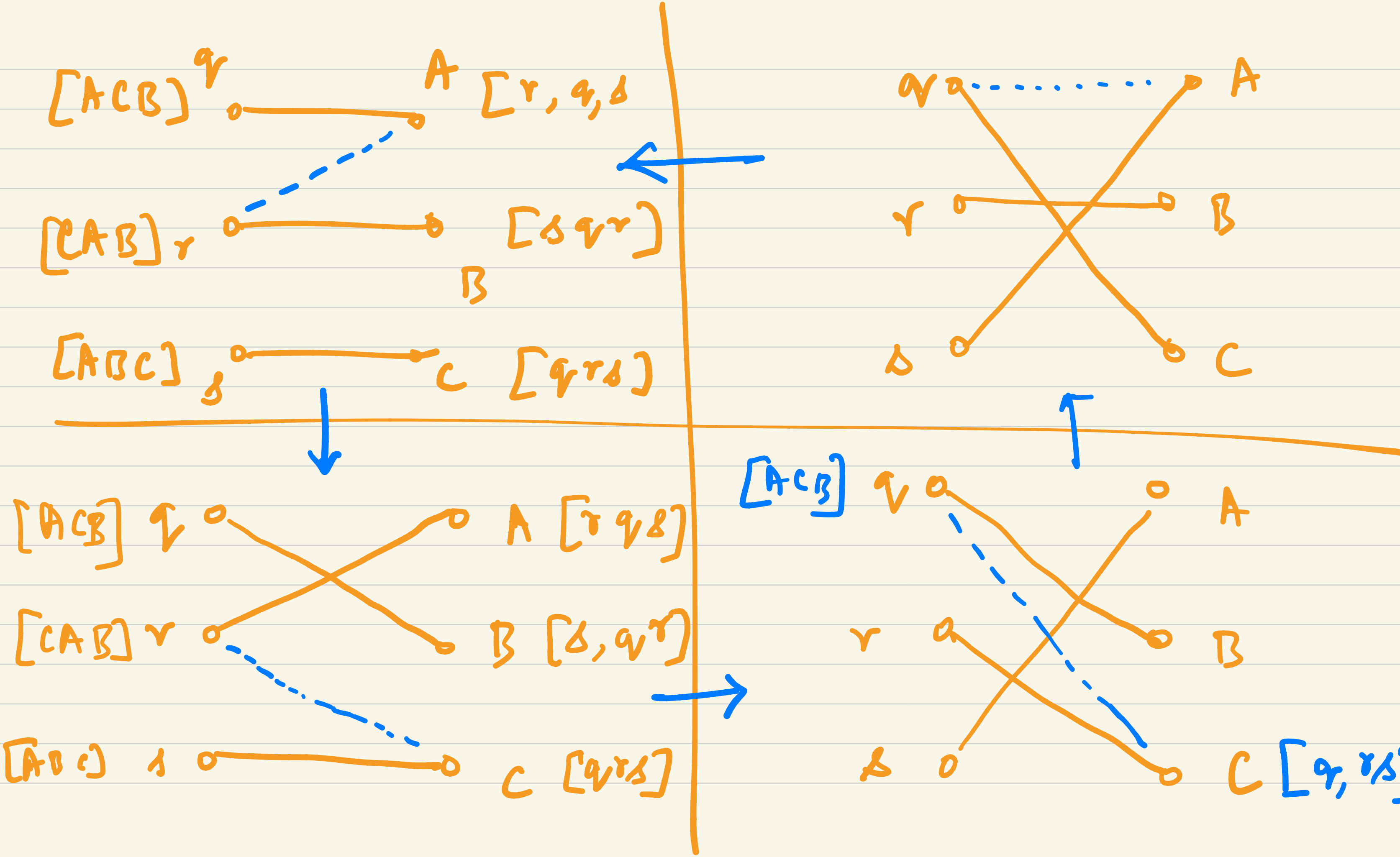


can (s1, c1) be unstable.








[ACB] q
A [r, q, s]
[CAB] r
B [s q r]
[ABC] s
C [q r s]
q
A
r
B
s
C
[ACB] q
A [r q s]
[CAB] r
B [s, q r]
[ABC] s
C [q r s]
[ACB] q
A
r
B
s
C [q, r s]

$M \leftarrow \emptyset$

while (some college c is unmatched and $\boxed{\text{has not offered to every student}}$)

*

$s \leftarrow$ 1st student in c's list who was not offered by c yet.

<u>if</u> (s is unmatched)

add (s, c) to M.

<u>else if</u> s is matched to c_1 and $c >_s c_1$

add (s, c) to M.

tag c_1 unmatched.

<u>else:</u> s rejects c's offer.

<u>Obs 1:</u> Colleges offers in decreasing order of preference.

<u>Obs 2:</u> Student once matched never becomes unmatched.

<u>Claim:</u>- The algorithm terminates in $O(n^2)$ iteration of the while loop.

<u>Proof:</u>- For a fixed college (*) can be true at most n times.

$\Rightarrow$ The condition inside the while loop can be true at most n^2 times. $\square$

Algo terminates.

Claim:- The algo returns a matching.

Proof: Suppose not.

Then some college c remains unmatched.

$\Rightarrow$ Some student s remains "

s was never matched [by obs 2]

the list of c is exhausted.

c had offered to s.

This cannot happen.

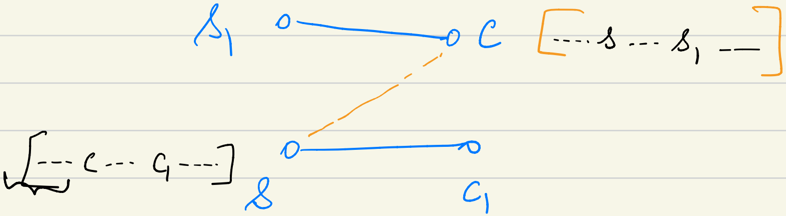
The algo returns a matching.

Thm:- The matching returned by the algo is stable.

Pf:- Suppose the matching M is not stable.

Let (s, c) be an up.

Suppose, $(s_1, c), (s, c_1) \in M$



c must have made an offer to s

+ c must have made this offer to s before c_1 made an offer to s.

$\Rightarrow$ not possible.