Platforms & Tools

D2L (DEN) : Syllabus ✓
Lecture Notes ✓
Lecture Videos ✓
HW Assignments ✓
HW Submissions ✓
Any other reference material ✓
~~Exams~~

Piazza : Discussion Board

Roles & Responsibilities

- Instructor lectures & Discussions
- TAs HWs & Exams
- Graders grading of HW assignments
- Course Producers *
- CS Dept. Advisors any Reg. issues
- DEN Support tech support issues

Textbooks

- Algorithms Design by Jon Kleinberg & Eva Tardos

- Supplemental Textbook: Introduction to Algorithms, 3rd edition, by Cormen et al.

Your Responsibilities

- Attending lectures and discussions
- Completing reading assignments
- Doing HW problems
- Doing as many other practice problems as possible

Your Grade

Exam 1	30%	Sep. 30
Exam 2	30%	Nov 4
Exam 3	25%	Dec 2
HW	10%	Weekly
Fin. Project	5%	
	100%	

Grading Scale

90 - 100	A	60 - 64.99	C^+
85 - 89.99	A^-	55 - 59.99	C
75 - 84.99	B^+	50 - 54.99	C^-
70 - 74.99	B	45 - 49.99	D
65 - 69.99	B^-	Below 45	F

- Grade boundaries will be reduced by the difference between class average grade and 75 if class average falls below 75
- At least 20% of the class will receive an A grade.
- At least 10% of the class will receive an A^- grade

Prerequisites

- Discrete Math – Mathematical Induction
- Sorting methods
- Basic data structures: Arrays, stacks, queues, linked lists
- Basics of graphs: Trees, cycles, DAG, adjacency list/matrix, etc.
- Graph search algorithms: BFS, DFS

High level Syllabus

Today!

- Introduction
- Review of some prereqs + asymptotic notation
- Major algorithmic techniques
  - Greedy
  - Divide & Conquer
  - Dynamic Programming

Exam 1

- Network Flow
- Computational Complexity Theory
- Approximation Algorithms
- Linear Programming

Exam 2

Exam 3

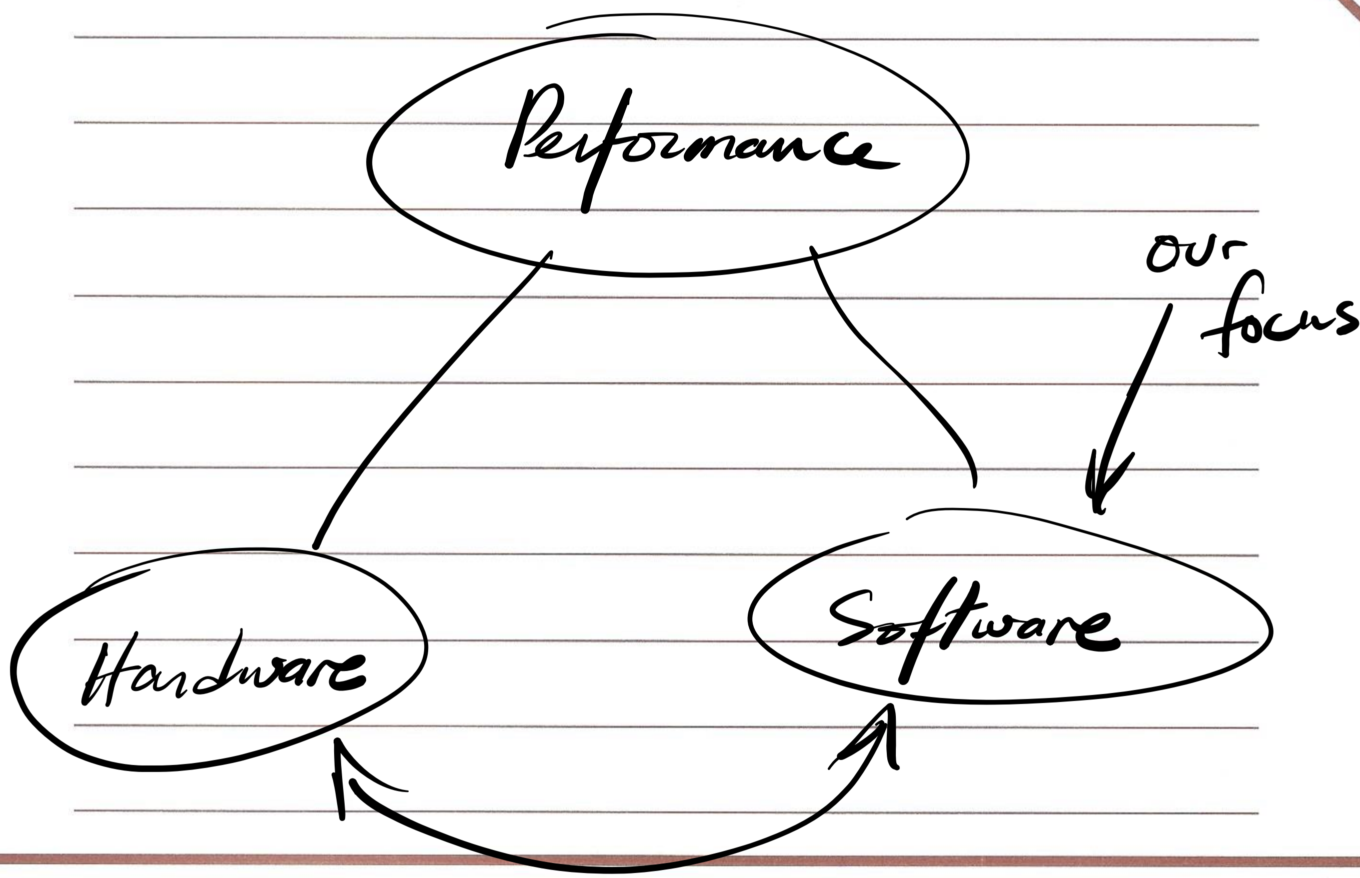
Corrections

1- ~~An algorithm is a set of instructions in machine language.~~

Kharazmi ~ 780-850

2- ... ~~Algorithmic science advanced on Wall Street ...~~

3- ... ~~Invite 6 million algorithms for a listen ...~~



- Parallel processing
  - SMP
  - DMP
- Memory hierarchy

When studying a problem, we go through the following steps:

1- Come up with a concise problem statement

2- Present a solution

3- Prove correctness

4- Perform complexity analysis

Stable Matching

Stable Matching Example

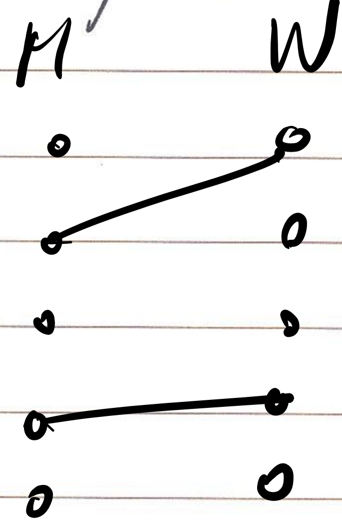
Problem: We are interested in matching $\underline{n}$ men with $\underline{n}$ women so that they could stay happily married ever after.

Step 1: Come up with a concise problem statement.

We have a set of $\underline{n}$ men, $M = \{m_1, \dots m_n\}$

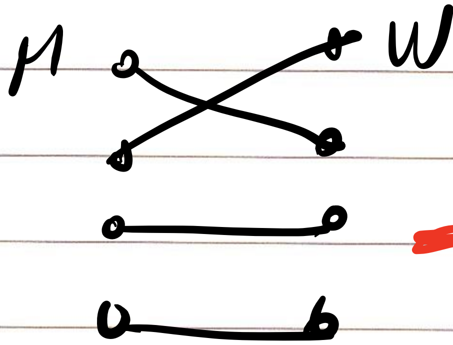
We have a set of $\underline{n}$ women, $W = \{w_1, \dots w_n\}$

Def. A Matching S is a set of ordered pairs.



A matching

Def. A perfect matching S' is a matching with the property that each member of M and each member of W appear in exactly one pair in S'.



A Perfect Matching

Add notion of preferences

Each man $m \in M$ ranks all women

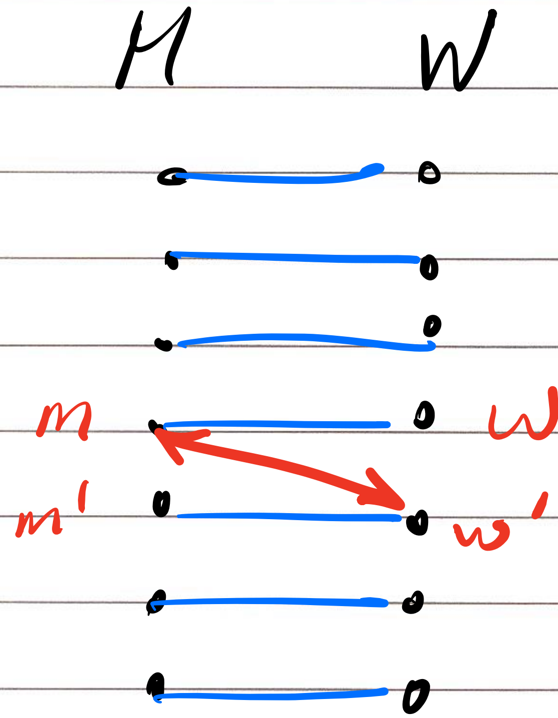
- $\underline{m}$ prefers $\underline{w}$ to $\underline{w'}$ if $\underline{m}$ ranks $\underline{w}$ higher than w'.

- Ordered ranking of $\underline{m}$ is his preference list

$$P_{m_i} = \{\underbrace{w_{i_1}}_{\text{highest ranking}}, w_{i_2}, \ldots \underbrace{w_{i_n}}_{\text{lowest ranking}}\}$$

Same for women, i.e. each woman $w \in W$ ranks all men ...

$\underline{S}$



Such a pair (m, w') is called an instability WRT $\underline{S}$.

Def. Matching S is stable if

1. It is perfect

2. There are no instabilities WRT S

Step 1: Input: Preference lists for a set of $\underline{n}$ men & $\underline{n}$ women.

Output: Set of $\underline{n}$ marriages w/ no instabilities

Step 2: Gale-Shapley Alg.

Step 3 Proof of Correctness

1. From the woman's perspective, she starts single, and once she gets engaged and she can only get into better engagements

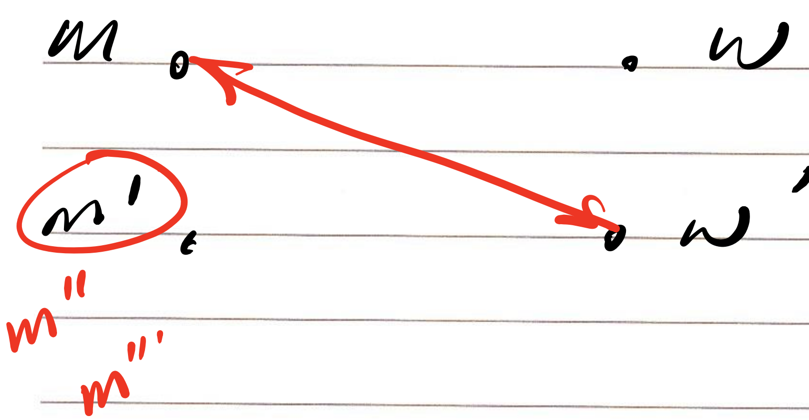
2. From the man's perspective, he starts single, gets engaged, and may get rejected repeatedly only to settle for a lower ranking woman

3. Solution will terminate in at most n^2 iterations to find a perfect matching

4. Need to show that our matching has no instabilities

Proof by Contradiction

Assume an instability exists in our solution involving two pairs (m,w), (m',w')

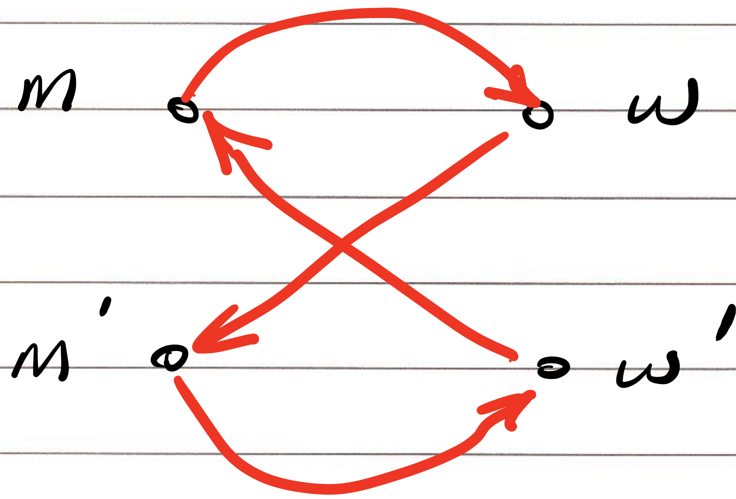


Q: Did $\underline{m}$ propose to w' at some point in the execution?

If no, then $\underline{w}$ must be higher than $\underline{w'}$ on his list $\Rightarrow$ contradiction!

If yes, he must have been rejected in favor of m'' and due to (1) either $m'' = m'$ or m' is better than m''

$\Rightarrow$ contradiction!



→ women proposing: (m, w'), (m', w)

→ men ″ : (m, w), (m', w')

Step 4 Complexity Analysis

1- Identify a free man $O(1)$

2- For a man $\underline{m}$, identify the highest ranked woman to whom he has not yet proposed. $O(1)$

3- For a woman $\underline{w}$, decide if $\underline{w}$ is engaged, and if so to whom $O(1)$

4- For a woman $\underline{w}$ and two men $\underline{m}$ & $\underline{m}'$, decide which man is preferred by $\underline{w}$ $O(1)$

5- Place a man back in the list of free men. $O(1)$

step 4 step 5

1. Identify a free man

	Get	Put
Linked list	$O(1)$	$O(1)$
queue	$O(1)$	$O(1)$
stack	$O(1)$	$O(1)$
array	$O(1)$	$O(1)$

2. Identify the highest ranked woman to whom <u>m</u> has not yet proposed.

Keep an array <u>Next[1..n]</u> where Next[m] points to the position of the next woman that <u>m</u> will be proposing to on his pref. list.

Men's preference list: ManPref[1..n, 1..n], where

ManPref[m, i] denotes the i^{th} woman on man m's preference list.

To find next woman w to whom m will be proposing to:

$$w = ManPref[m, Next[m]]$$

Takes O(1)

3_ Determine woman w's status

Keep an array called Current[1..n]
where Current[w] is Null
if w is single & set to m
if w is engaged to m.

takes O(1)

4_ Determine which man is preferred by w.

WomanPref:

3	8	4	32	1	-	-	-	...
1	2	3	4	5	6	-	-	-

WomanRanking

5		1	3					
1	2	3	4	5	6	-	-	-

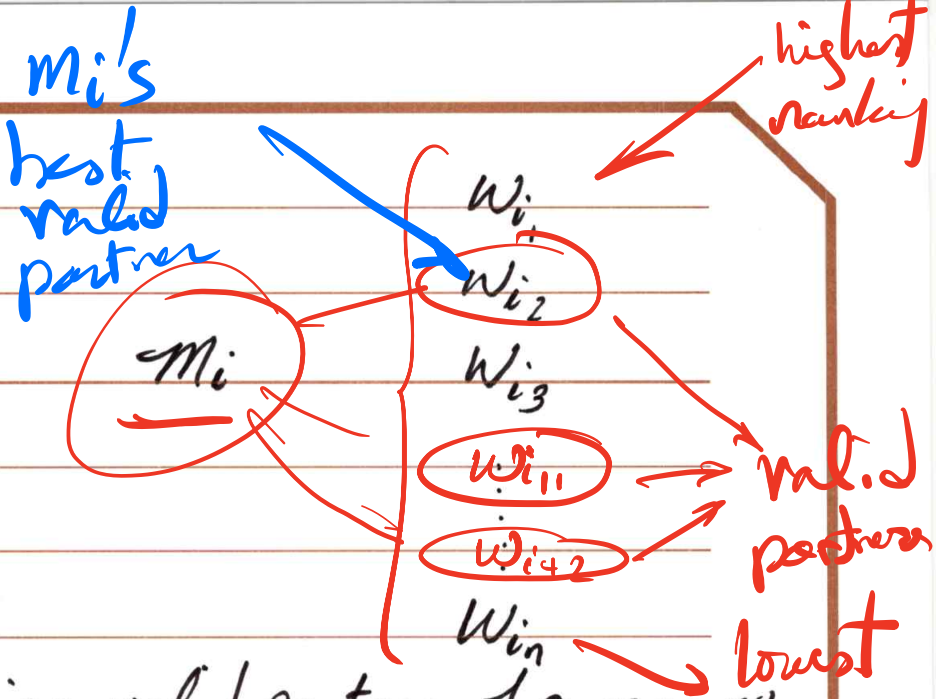
Preparation before entering GS iterations

Create a Ranking array where Ranking [w,m] contains the rank of man m based on w's preference

Preparation + GS iteration

$O(n^2)$ $O(n^2)$

overall complexity = $O(n^2)$



Def. Woman $\underline{w}$ is a valid partner of a man $\underline{m}$ if there is a stable matching that contains the pair (m, w)

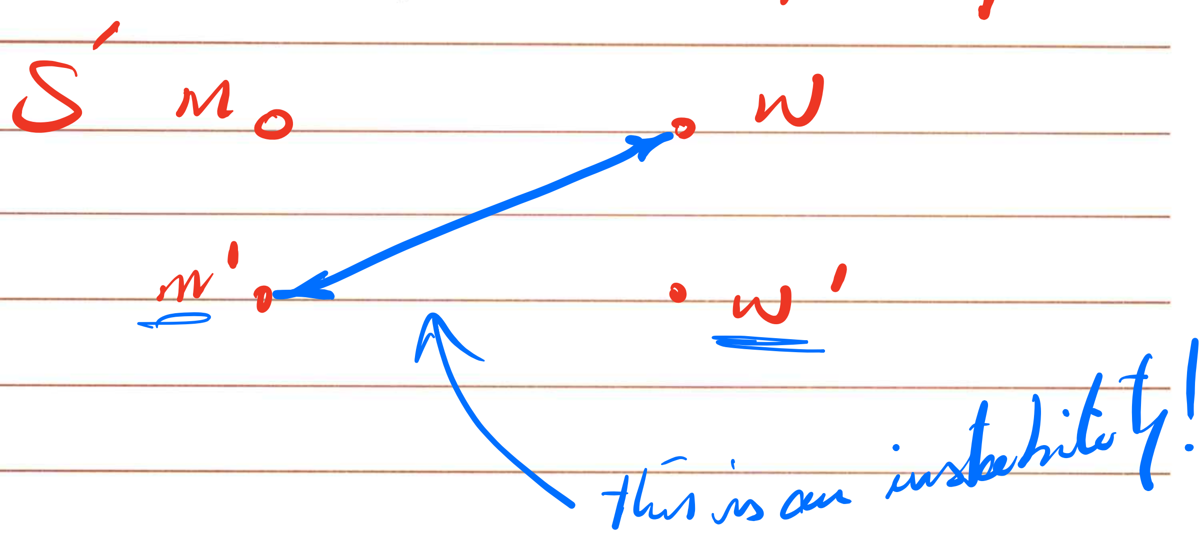
Def. m's best valid partner is

Claim: Every execution of the G-S algorithm (When men propose) results in the same stable matching regardless of the order in which men propose.

Plan: to prove this, we will show that when men propose, they always end up with their best valid partner.

Proof by contradiction:

Say $\underline{m}$ is the first man rejected by a valid partner w in favor of m'.

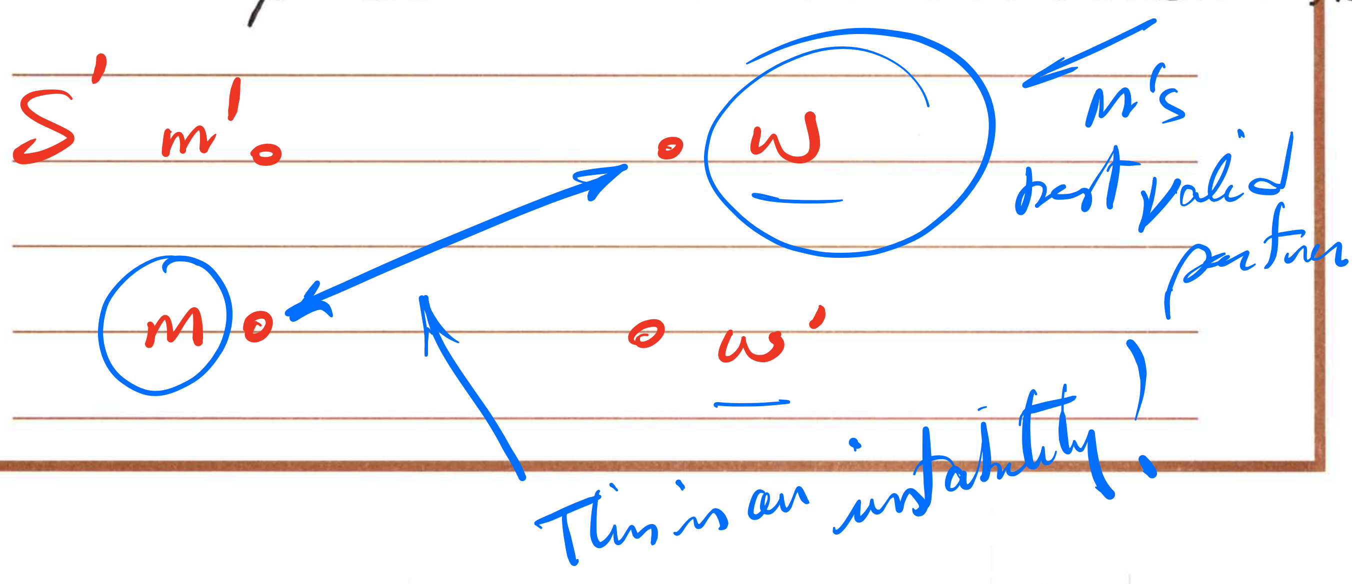


Claim: When men propose, women end up with their worst valid partners

Proof: By contradiction . . .

Suppose we end up with a matching S where for a pair (m, w) in S, m is not w's worst valid partner.

So there must be another matching S' where w is paired with a man m' whom she likes less.



Discussion 1

1. Prove that every execution of the G-S algorithm (when men are proposing) results in the same stable matching regardless of the order in which men propose.

2. Prove that when we run the G-S algorithm with men proposing, women end up with their worst valid partners.

3. True or False:
   In every stable matching that Gale–Shapley algorithm may end up with when men propose, there is a man who is matched to his highest-ranked woman.

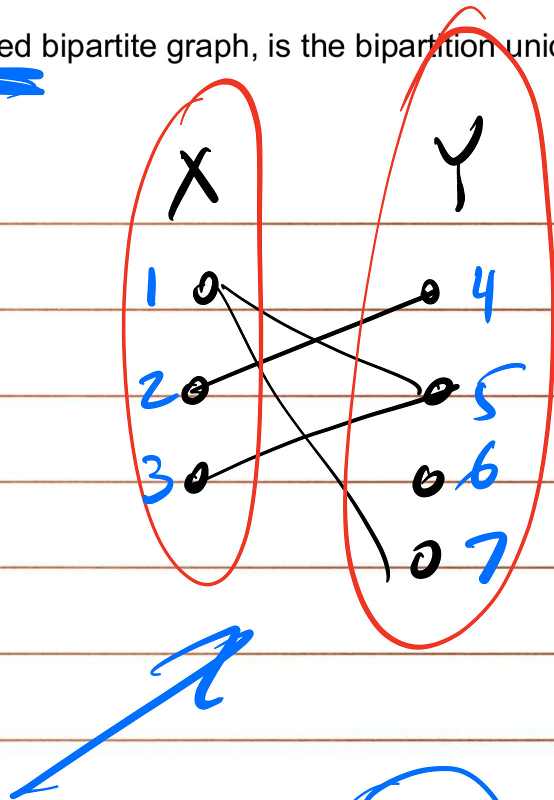
4. In a connected bipartite graph, is the bipartition unique? Justify your answer.

3. True or False:
In every stable matching that Gale–Shapley algorithm may end up with when men propose, there is a man who is matched to his highest-ranked woman.

m_1	m_2	m_3	w_1	w_2	w_3
w_1	w_1	w_2	m_3	m_2	m_1
w_2	w_2	w_1	m_2	m_1	m_2
w_3	w_3	w_3	m_1	m_3	m_3

w_1	w_2	w_3
m_3 $\cancel{m_2}$ $\cancel{m_1}$	m_2 $\cancel{m_1}$	m_1

4. In a connected bipartite graph, is the bipartition unique? Justify your answer.



$X = \{1, 2, 3\}$

$Y = \{4, 5, 6, 7\}$

$X' = \{ - \; - \; . \}$

$Y' = \{ \quad \}$