Algorithmic Game Theory
COMP6207

Lecture 10: Stable Matching

Baharak Rastegari
b.rastegari@soton.ac.uk
Electronics and Computer Science
University of Southampton

Learning Outcomes

- By the end of this session, you should be able to
  - ***Describe*** the stable matching problem and its objective.
  - ***Identify*** blocking pairs
  - ***Compute*** a stable matching by Gale-Shapley algorithm

College Admission

- How to assign students to universities/colleagues:
  - Every student wants to go to the university s/he likes the best
  - Universities want students they think are the best
  - Every program in every university has a limited number of seats
  - There are many ways of allocating students to different programs in various universities!
  - Which one makes more sense?

Too complicated?
Lets assume that each university has only **1** seat!

College admission => Dance Gala

- Let us assume that each university has only **1** seat!!!
- Now lets replace ``students’’ with ``leaders’’ and ``universities’’ with ``followers’’, or the other way around (doesn't matter)
- For now, assume that leaders can only lead and followers can only follow

Setting

- **Participants**
  - A set of leaders $\boldsymbol{L = \{1, \dots, n\}}$
  - A set of followers $\boldsymbol{F = \{1, \dots, n\}}$
- **Preferences**
  - Each leader has strict preferences over all followers
  - Each follower has strict preferences over all leaders

All preferences together: preference profile

- **Objective**
  - To find a **one-to-one** ***stable*** matching
    - (one-to-one) Matching: each leader is paired with at most one follower and vice versa
    - Stable: no pair $\boldsymbol{(l, f)}$ wants to deviate

Wizarding Schools' Spring Ball

Harry

Cho

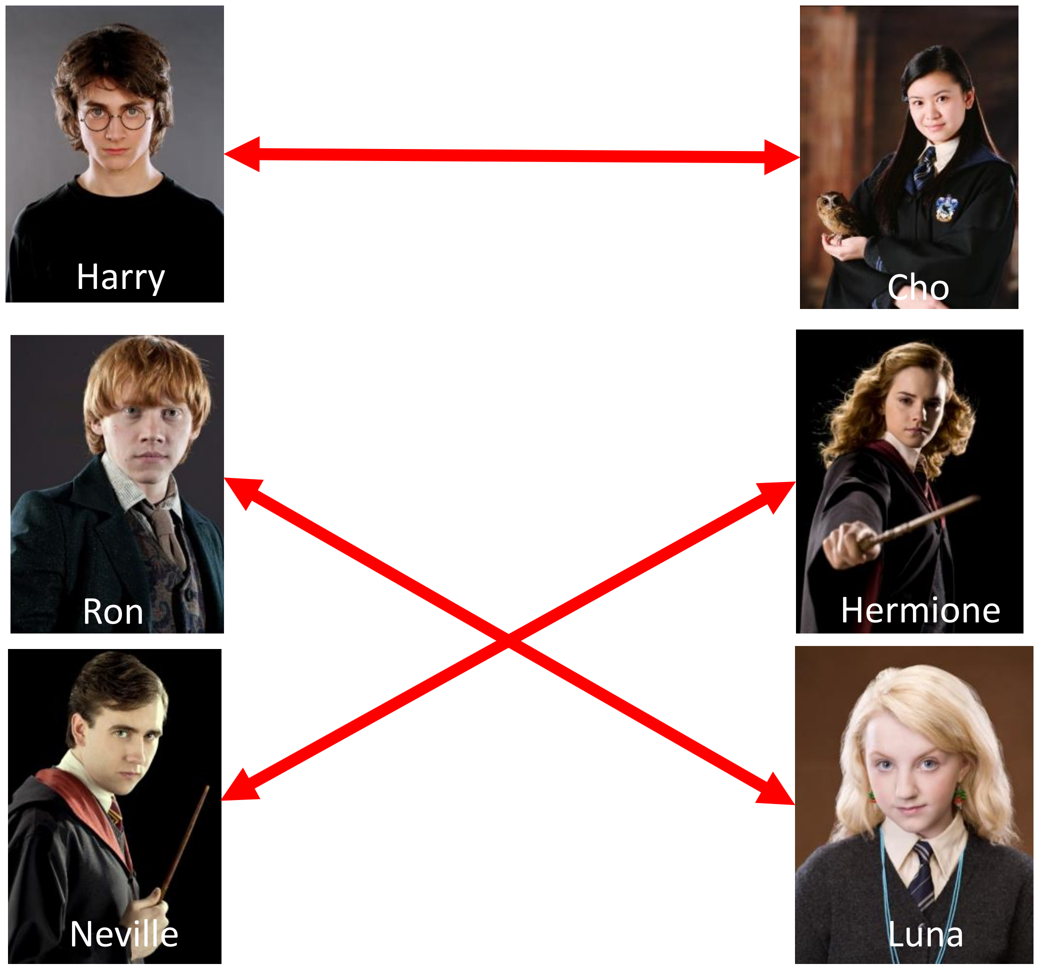
Ron

Hermione

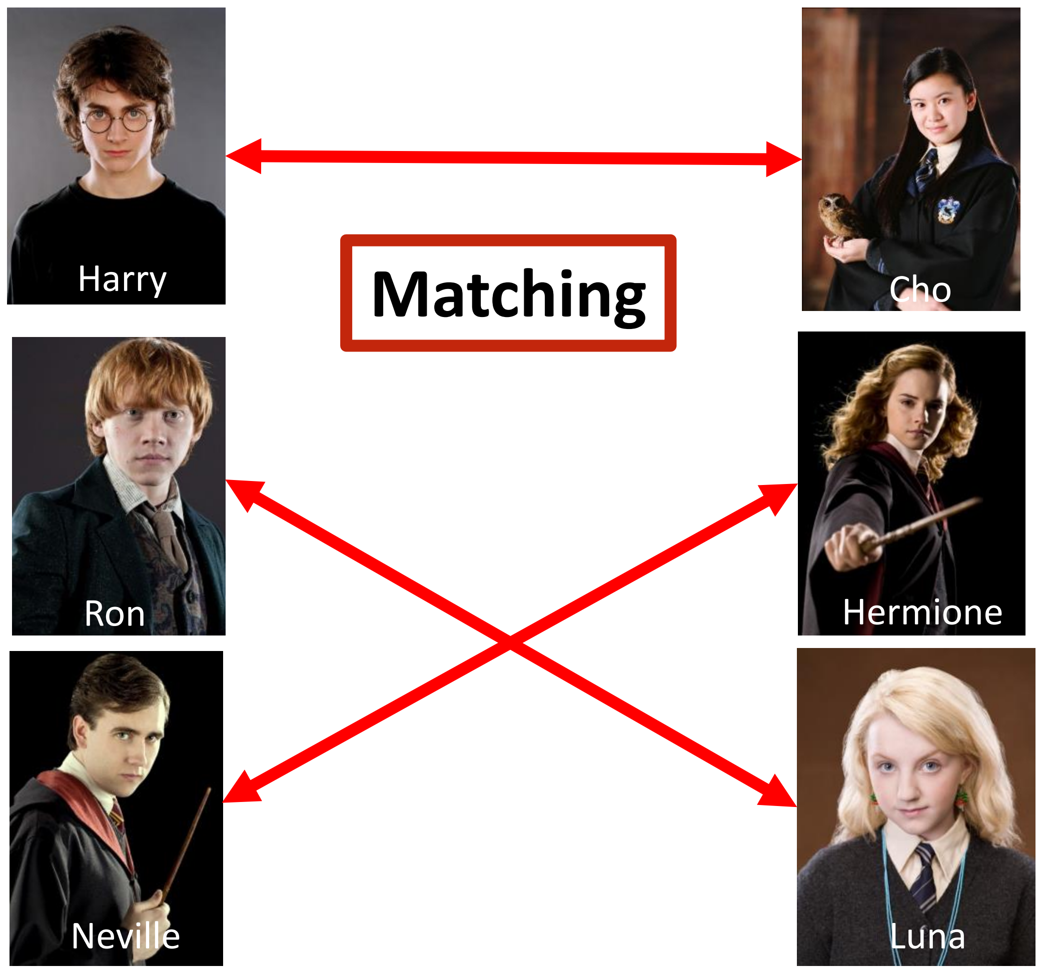
Neville

Luna

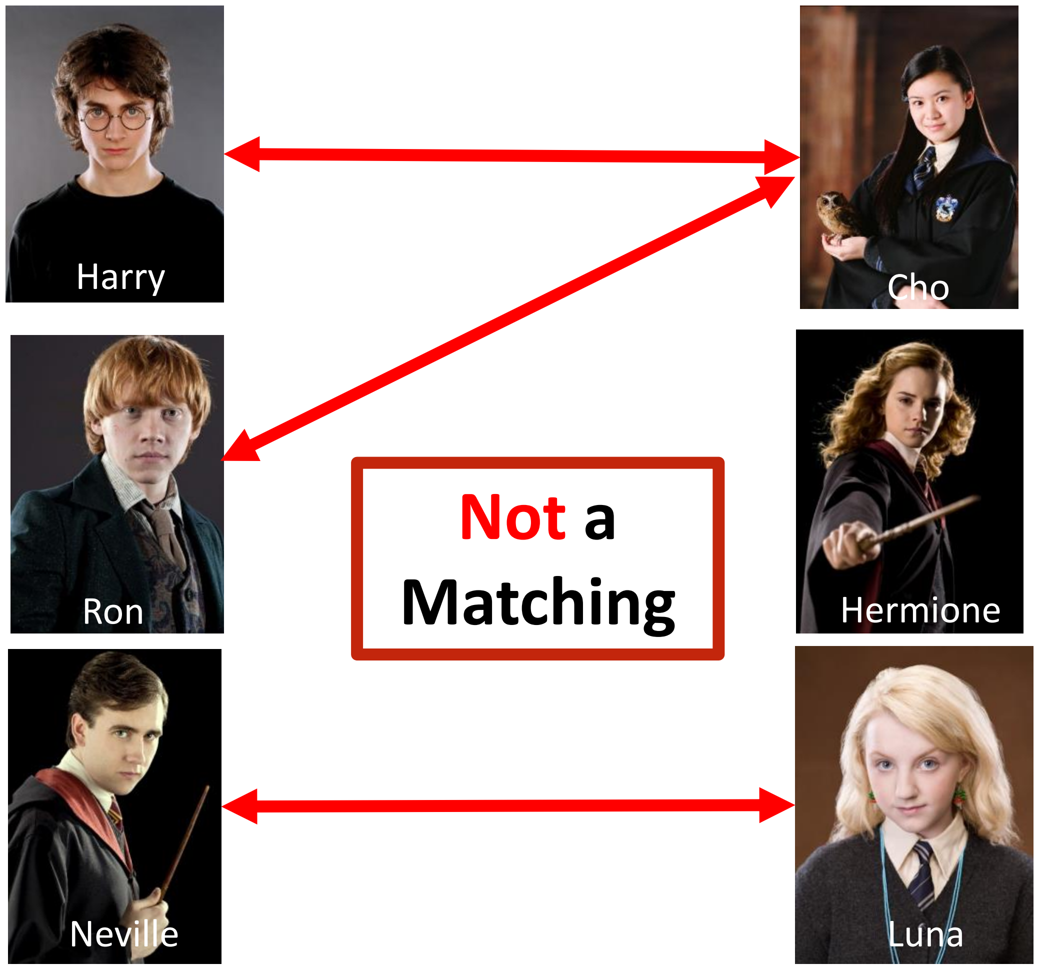
Wizarding Schools' Spring Ball



Wizarding Schools’ Spring Ball



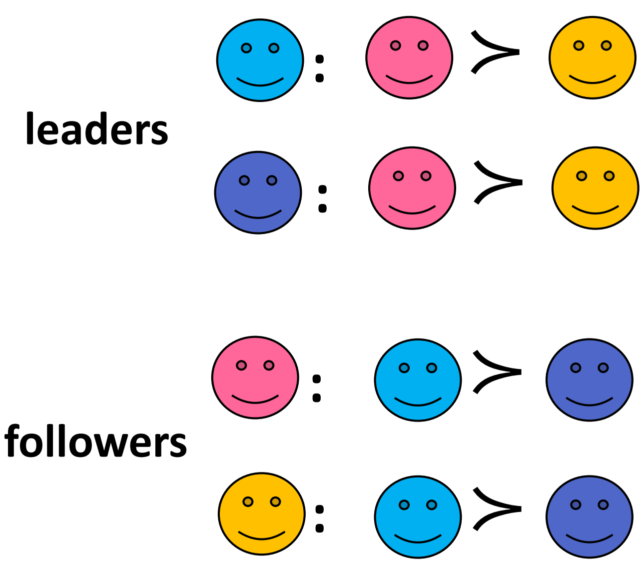
Wizarding Schools’ Spring Ball

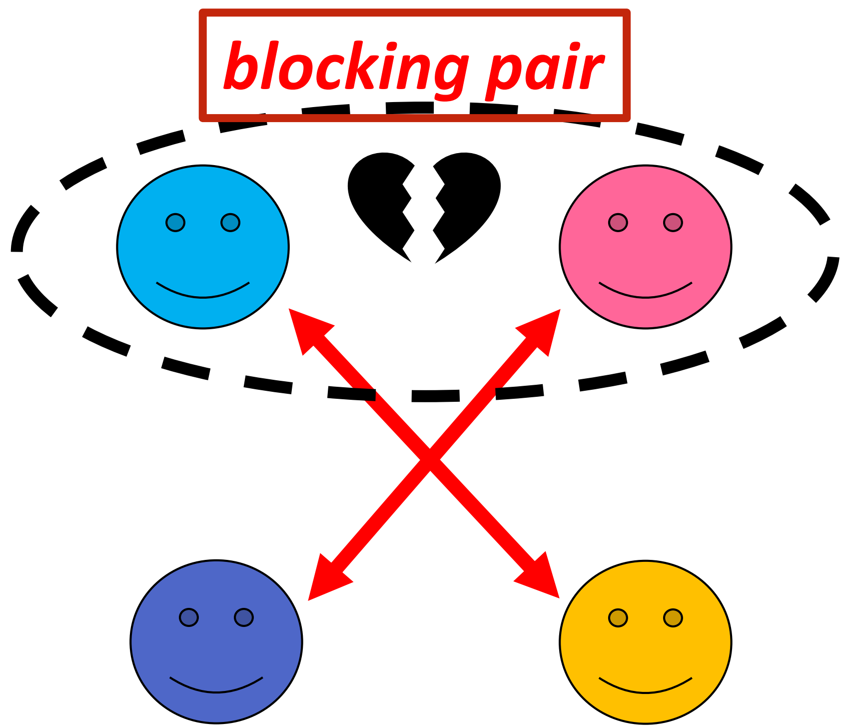


Not a Matching

Stable Matching

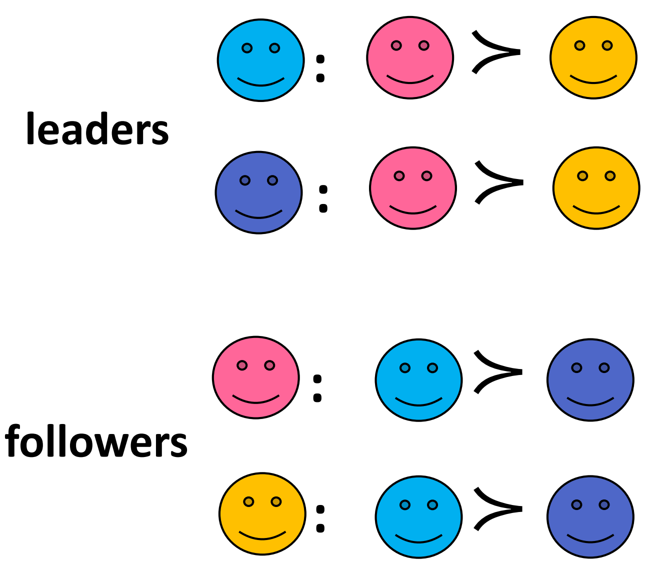
- A matching is ***stable*** if
  - There is no leader-follower pair, each of whom would prefer to match with each other rather than their assigned partner.
- Such a pair is called a ***blocking pair***

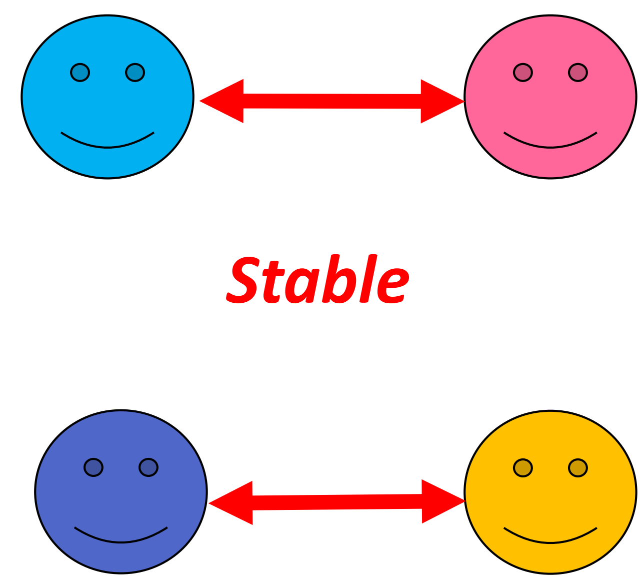




Stable Matching

- A matching is ***stable*** if
  - There is no leader-follower pair, each of whom would prefer to match with each other rather than their assigned partner.
  - Such a pair is called a ***blocking pair***





Stable Matching Problem (SM)

- Does a stable matching always exist?

- Can we find a stable matching efficiently, if it exists?

These two questions answered in 1962

Theorem (Gale & Shapley, 1962)

A stable matching always exists, and can be found in polynomial time.

Lloyd Shapley

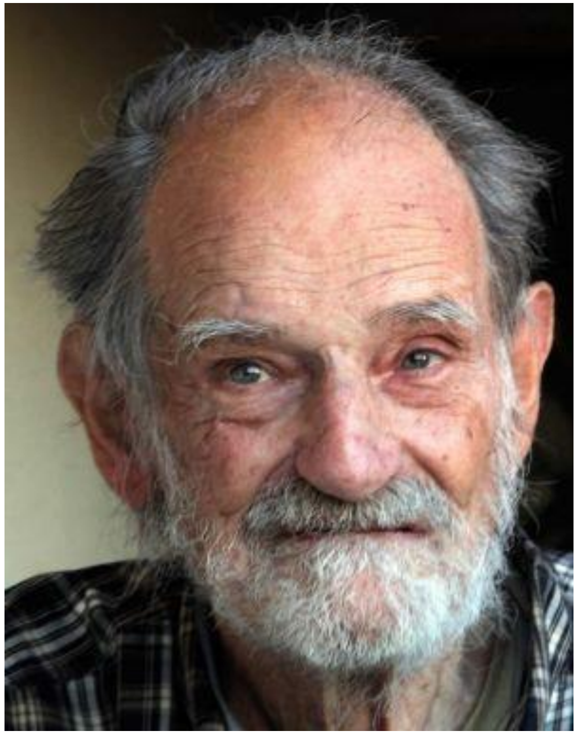
David Gale

COLLEGE ADMISSIONS AND THE STABILITY OF MARRIAGE

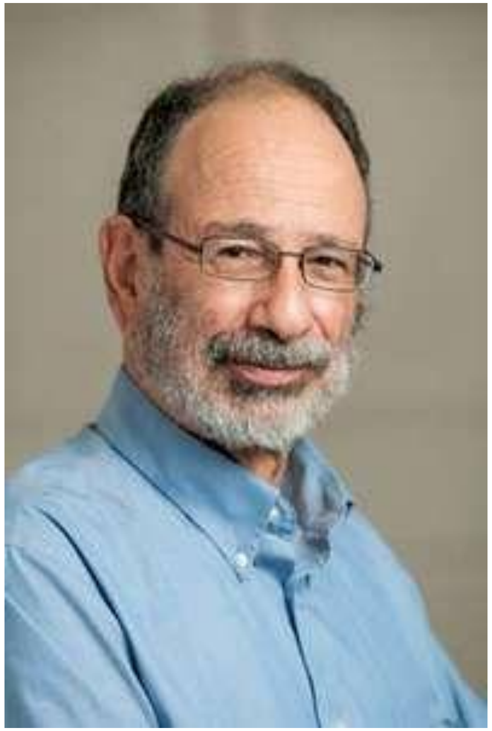
D. GALE* AND L. S. SHAPLEY, Brown University and the RAND Corporation

1. Introduction. The problem with which we shall be concerned relates to the following typical situation: A college is considering a set of n applicants of

2012 Nobel Prize Economic Sciences



Loyd S. Shapley

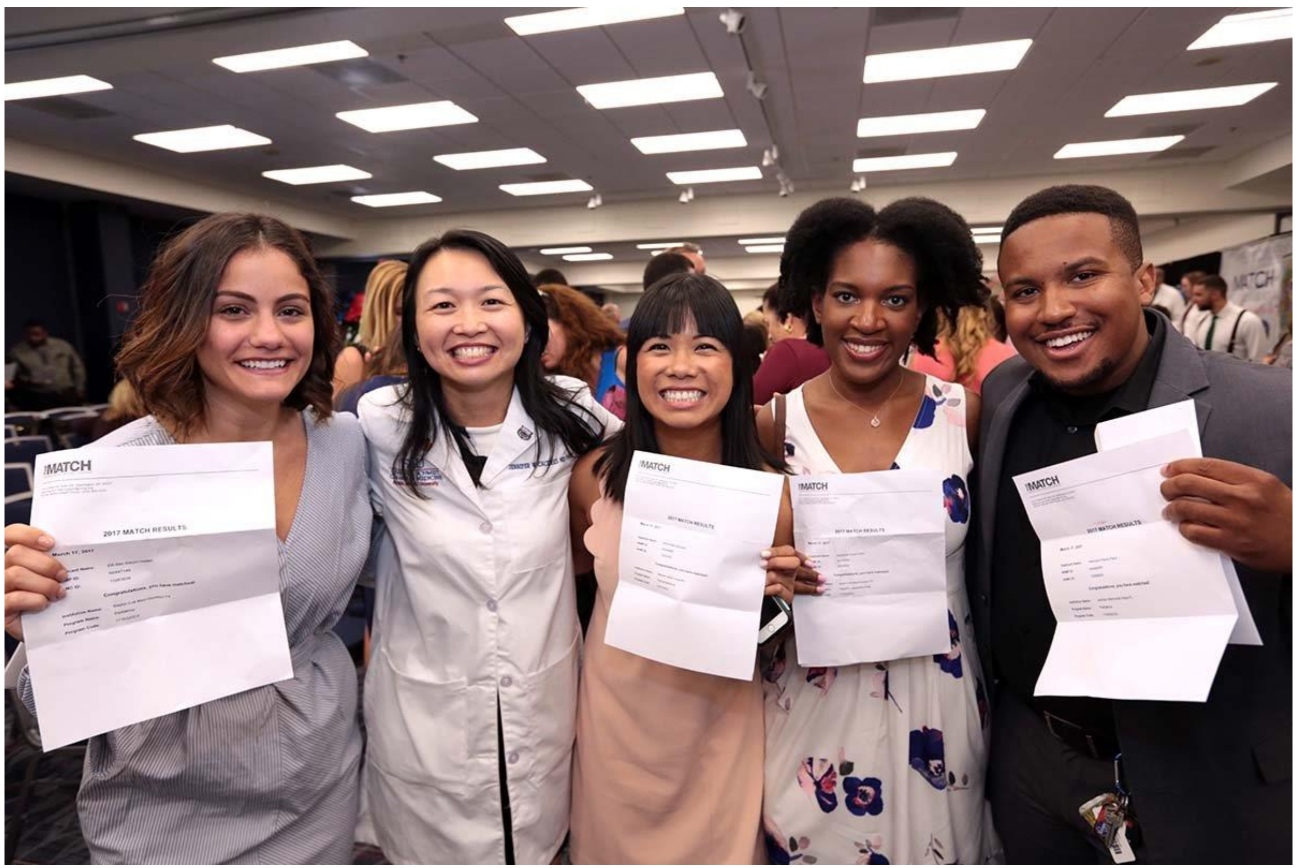


Alvin Roth

``for the theory of stable allocations and the practice of market design"

Applications

- Student-college admission
- School choice
- Hospitals/Residents problem
- …



Match Day 2017. *Credit: Charles E. Schmidt College of Medicine, FAU.*
For more photos of this important day of medical students' life click [here](#).

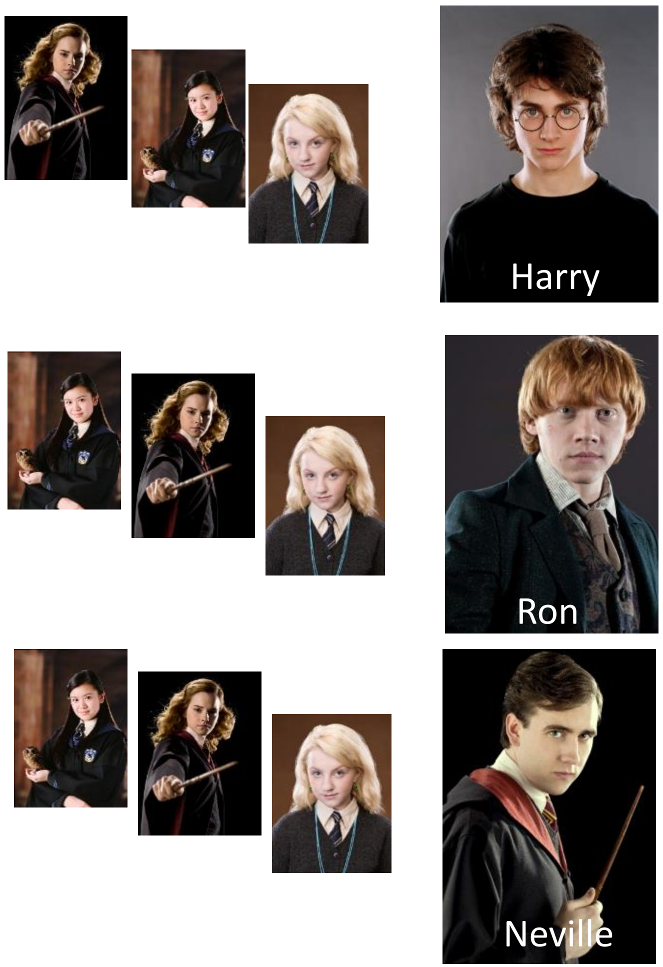
Gale-Shapley algorithm

Deferred-acceptance-leader-oriented (leaders, followers, preferences)

```
Assign all leaders and followers to be free; //initial state
While (some leader l is free and hasn’t proposed to every follower)
     f = first follower on l's list to whom l hasn't yet proposed;
     // next: l proposes to f
     If (f is free)
          assign l and f to be engaged; //tentatively matched
     else if (f prefers l to her fiancé l') { //f is engaged
          set l and f to be engaged;
          set l' to be free;
     else  f rejects l; //and l remains free
output the n engaged pairs, who form a stable matching;
```

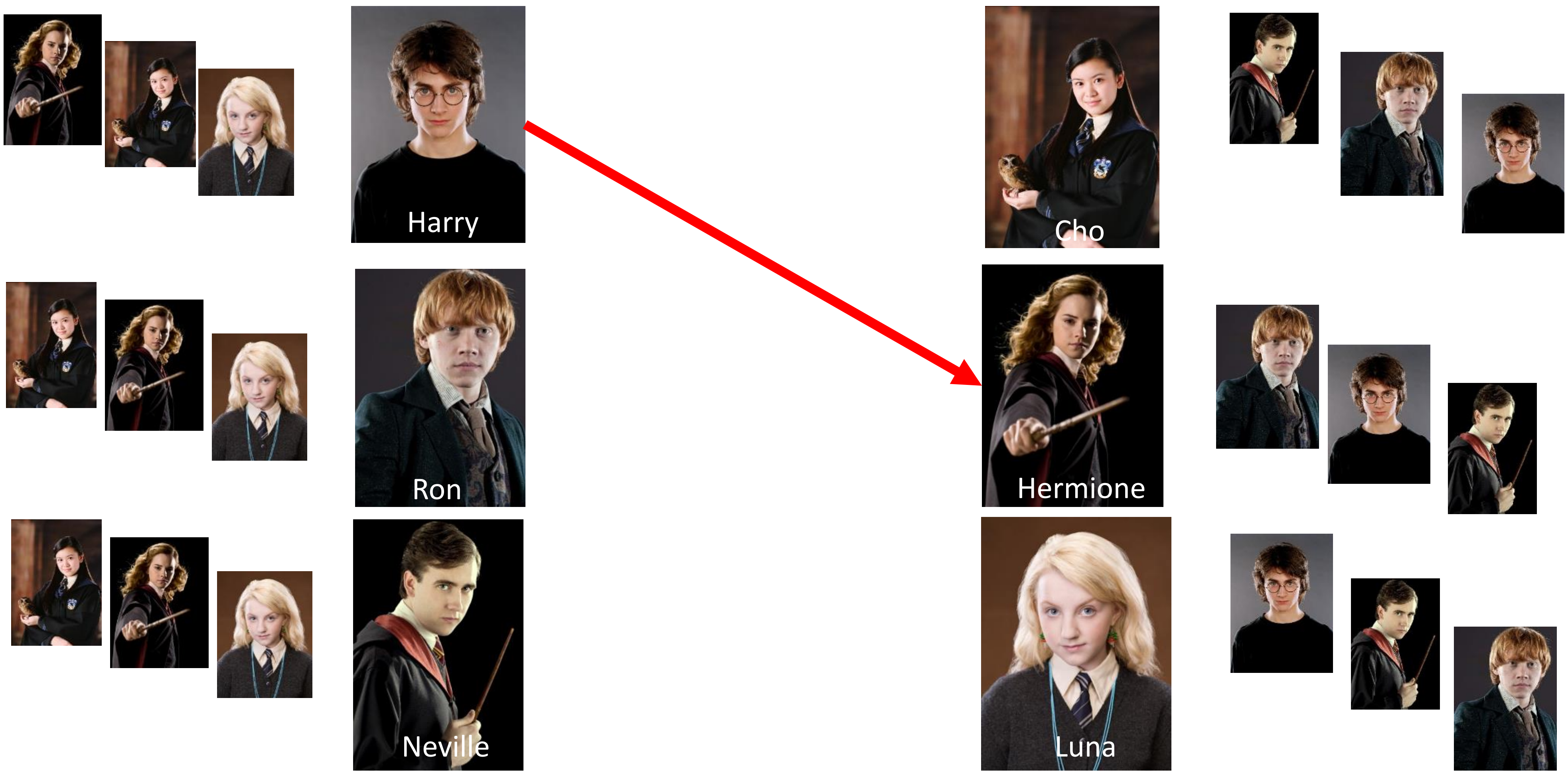
Example

Leaders

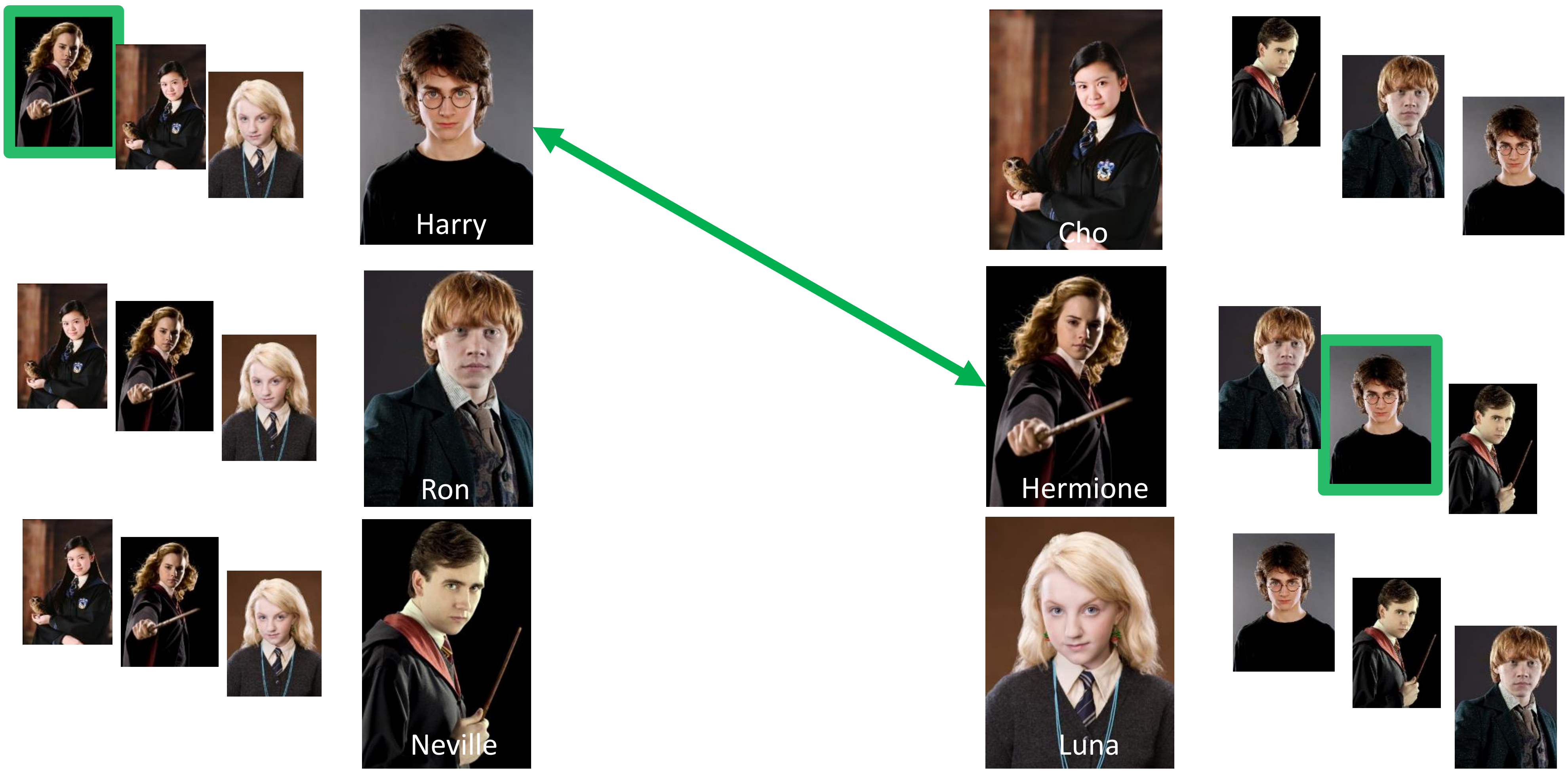


Followers

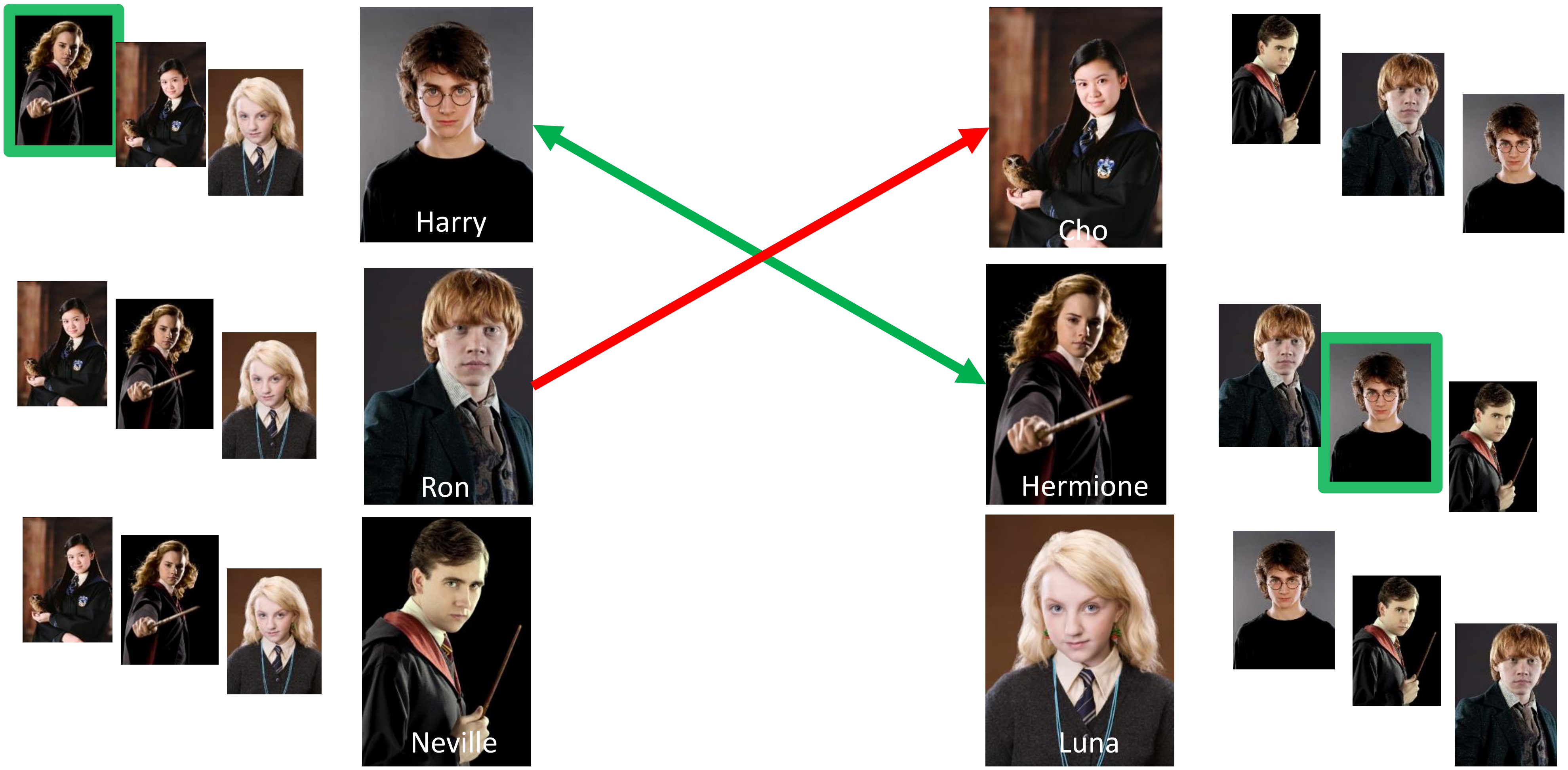
Example



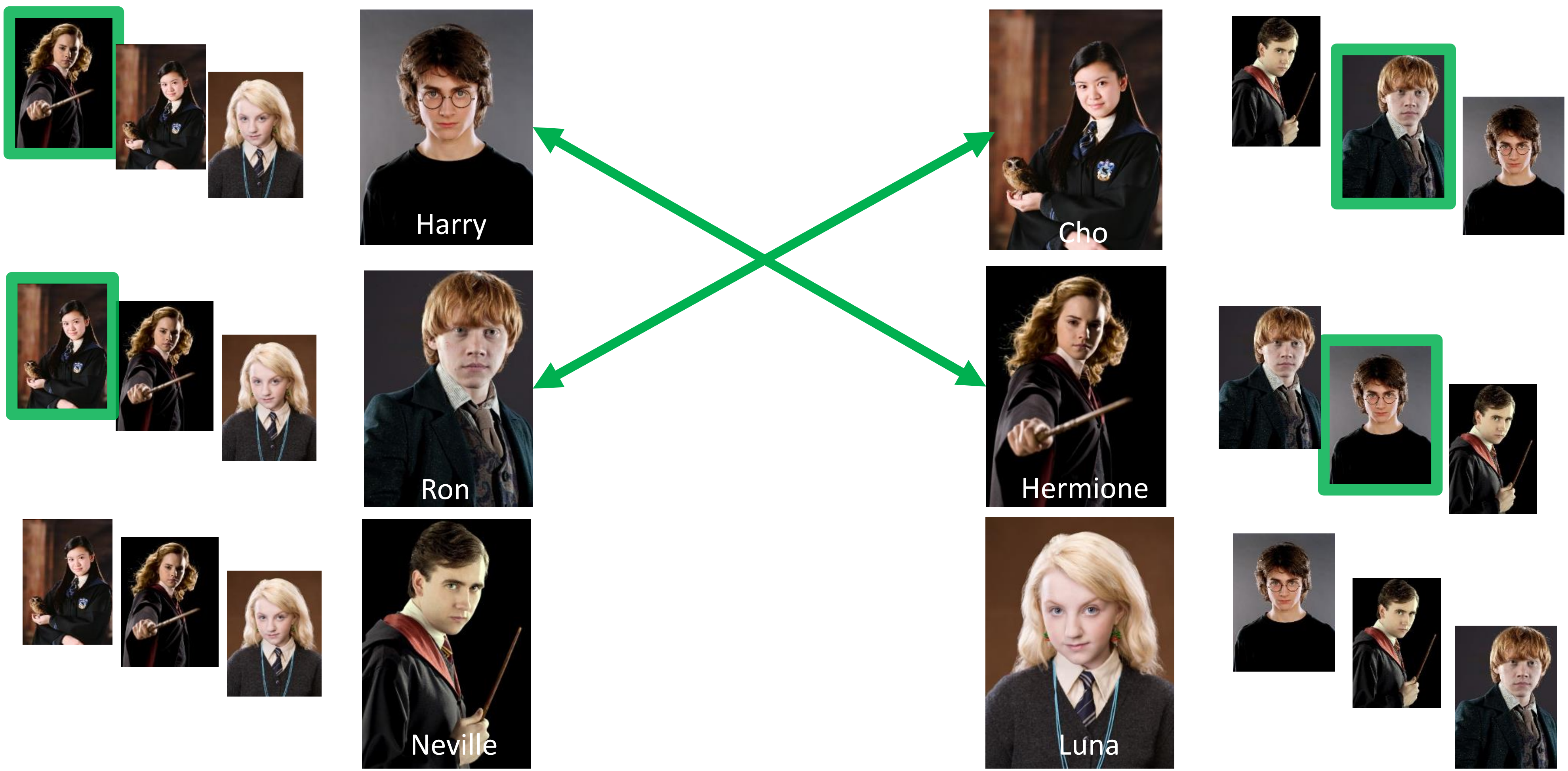
Example



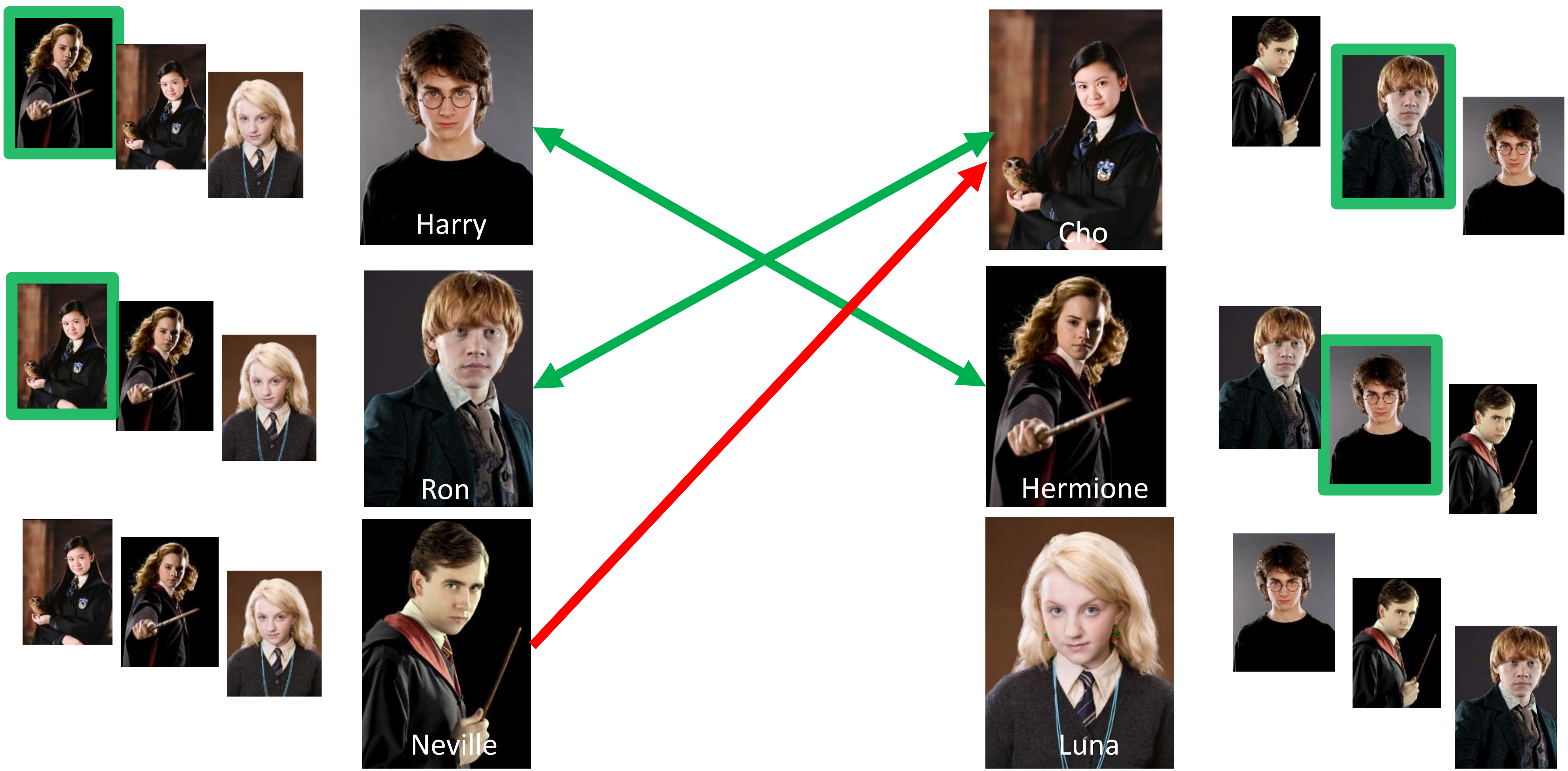
Example



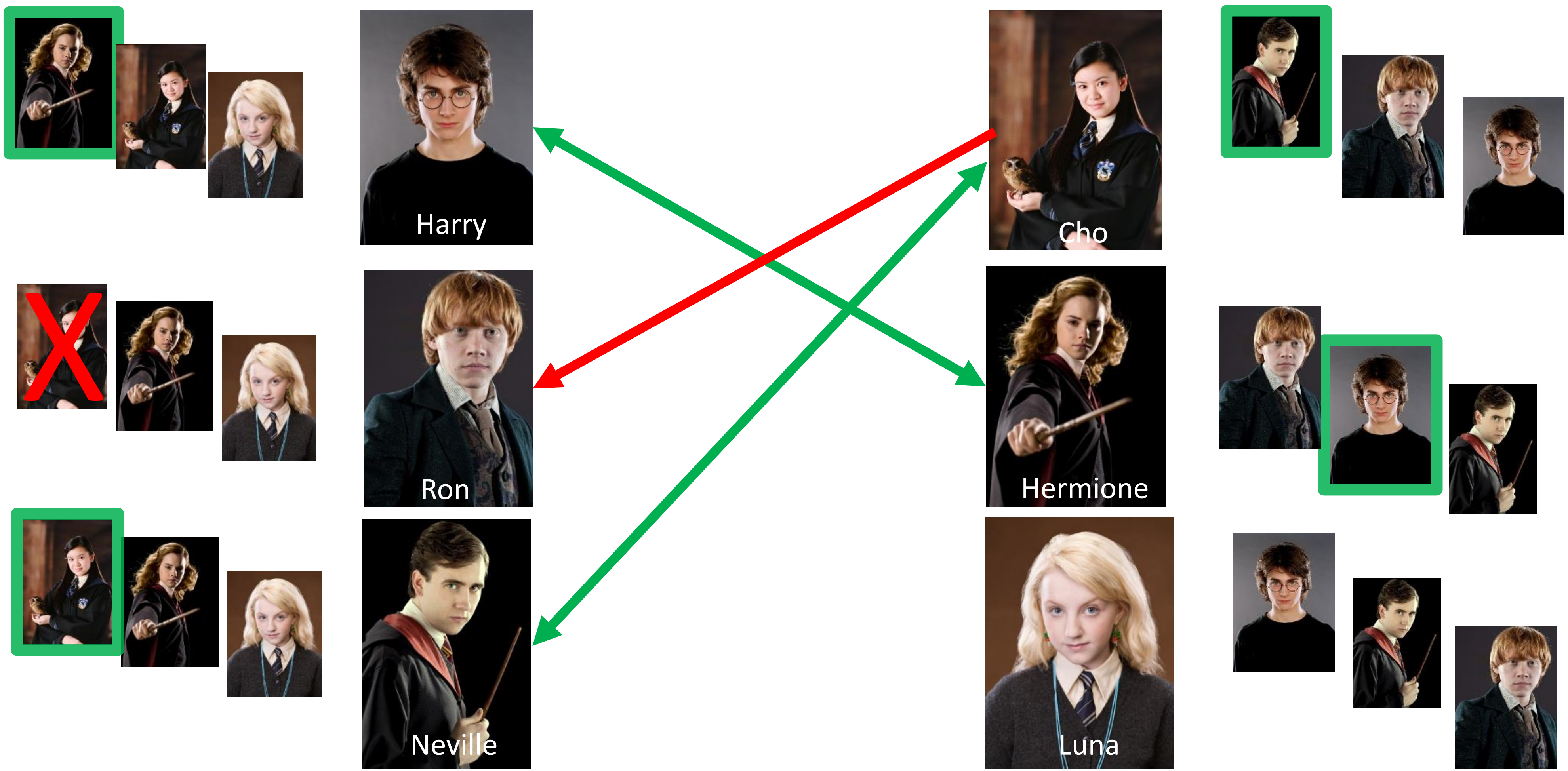
Example



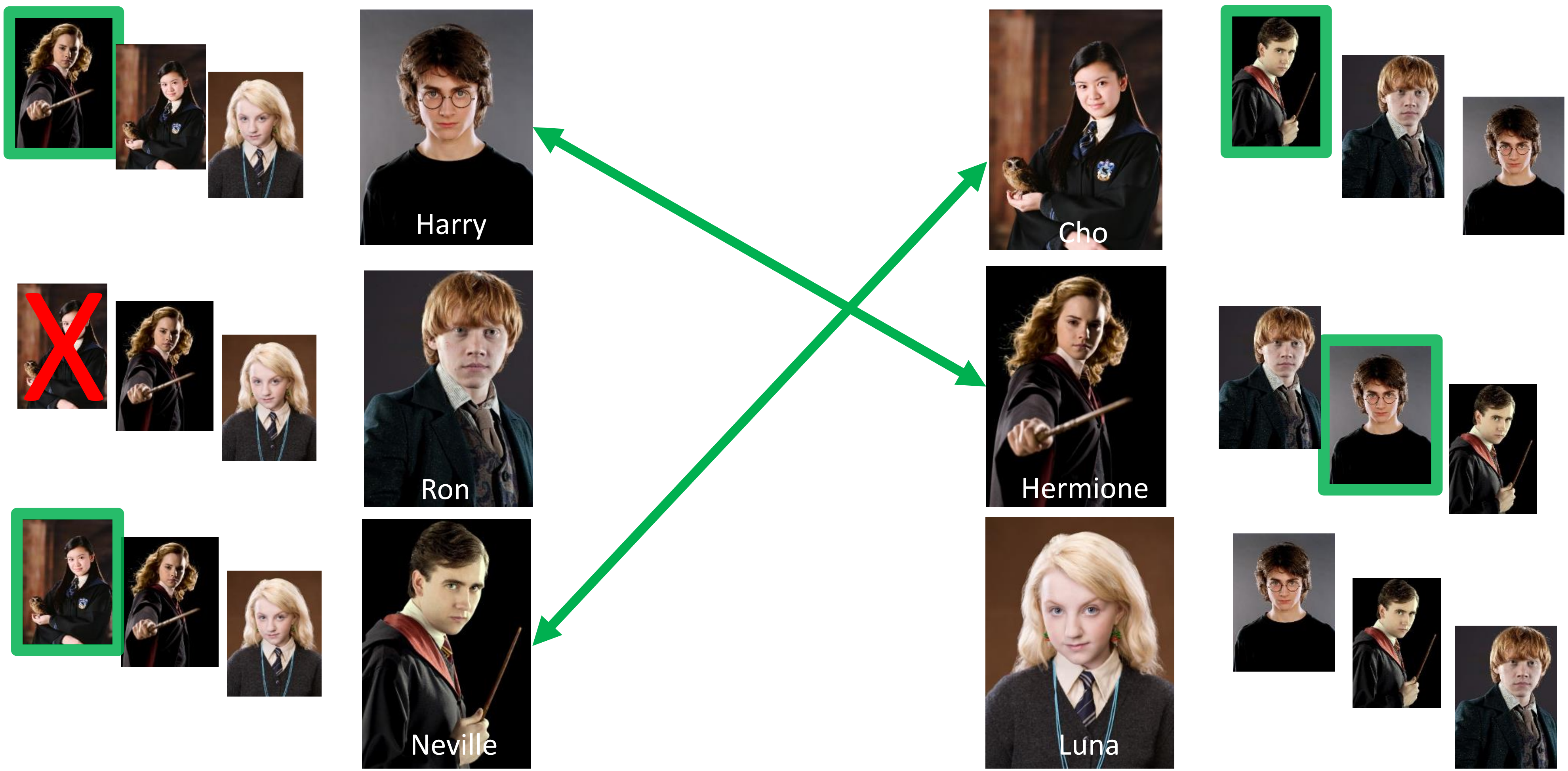
Example



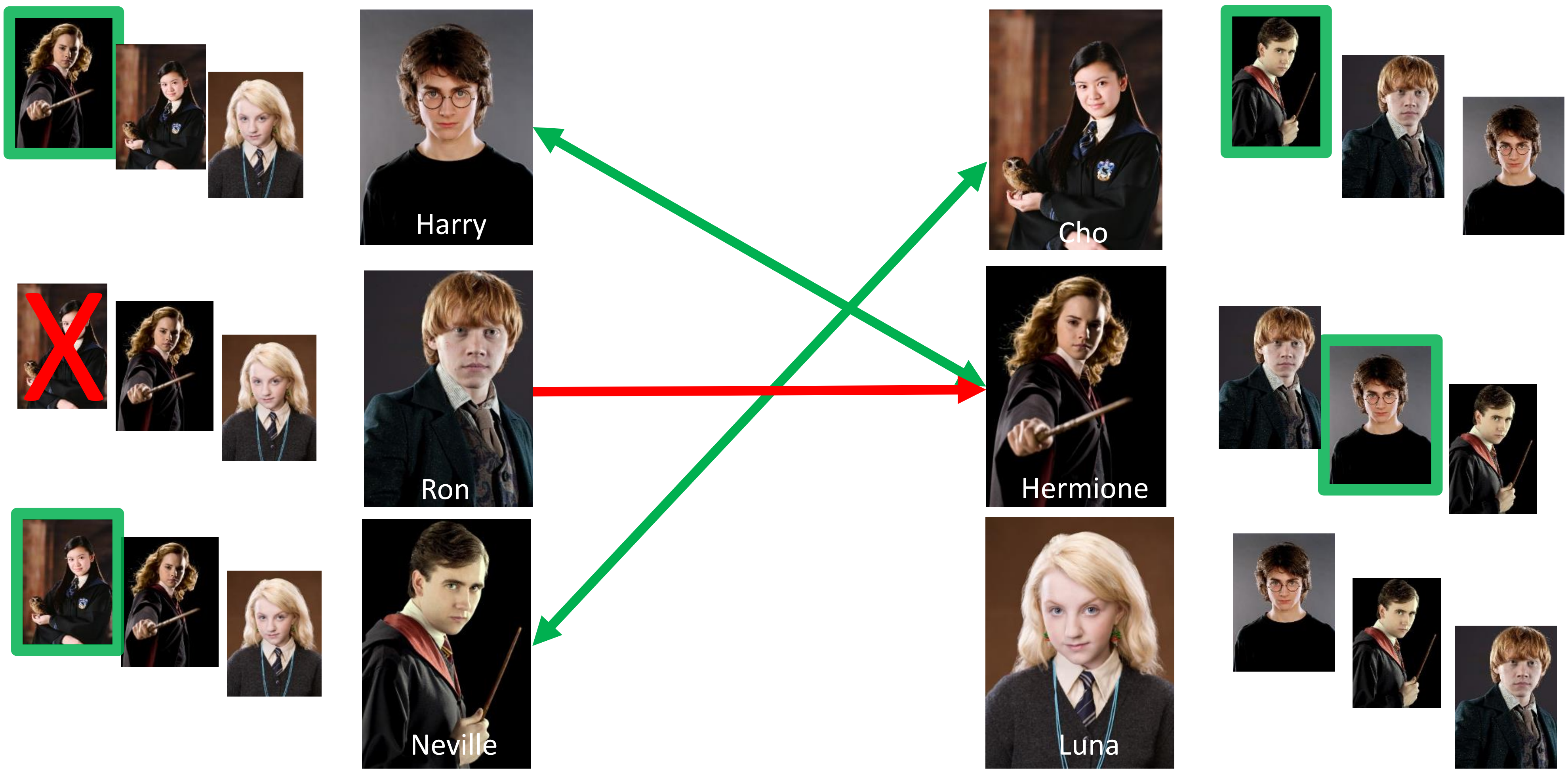
Example



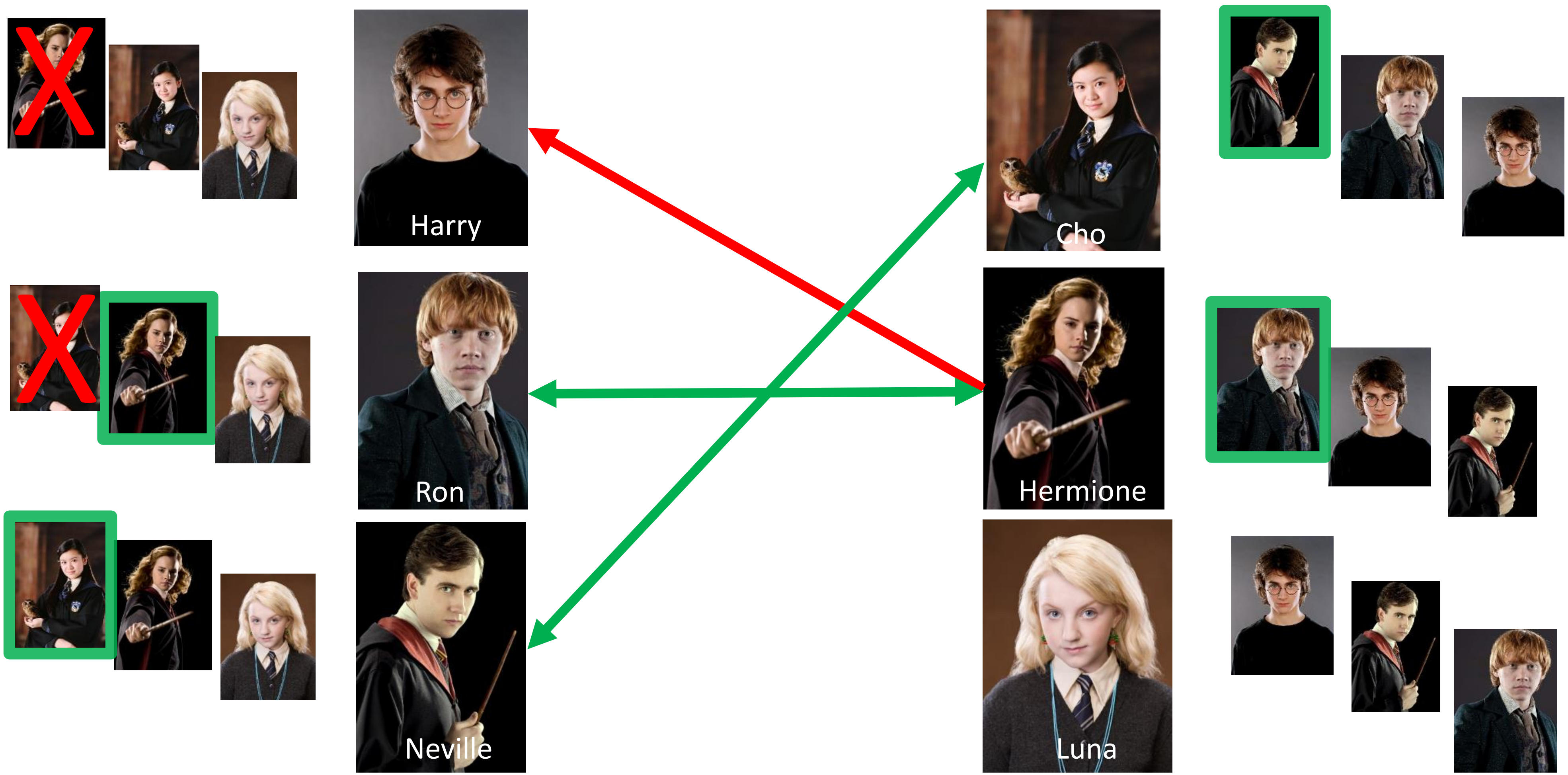
Example



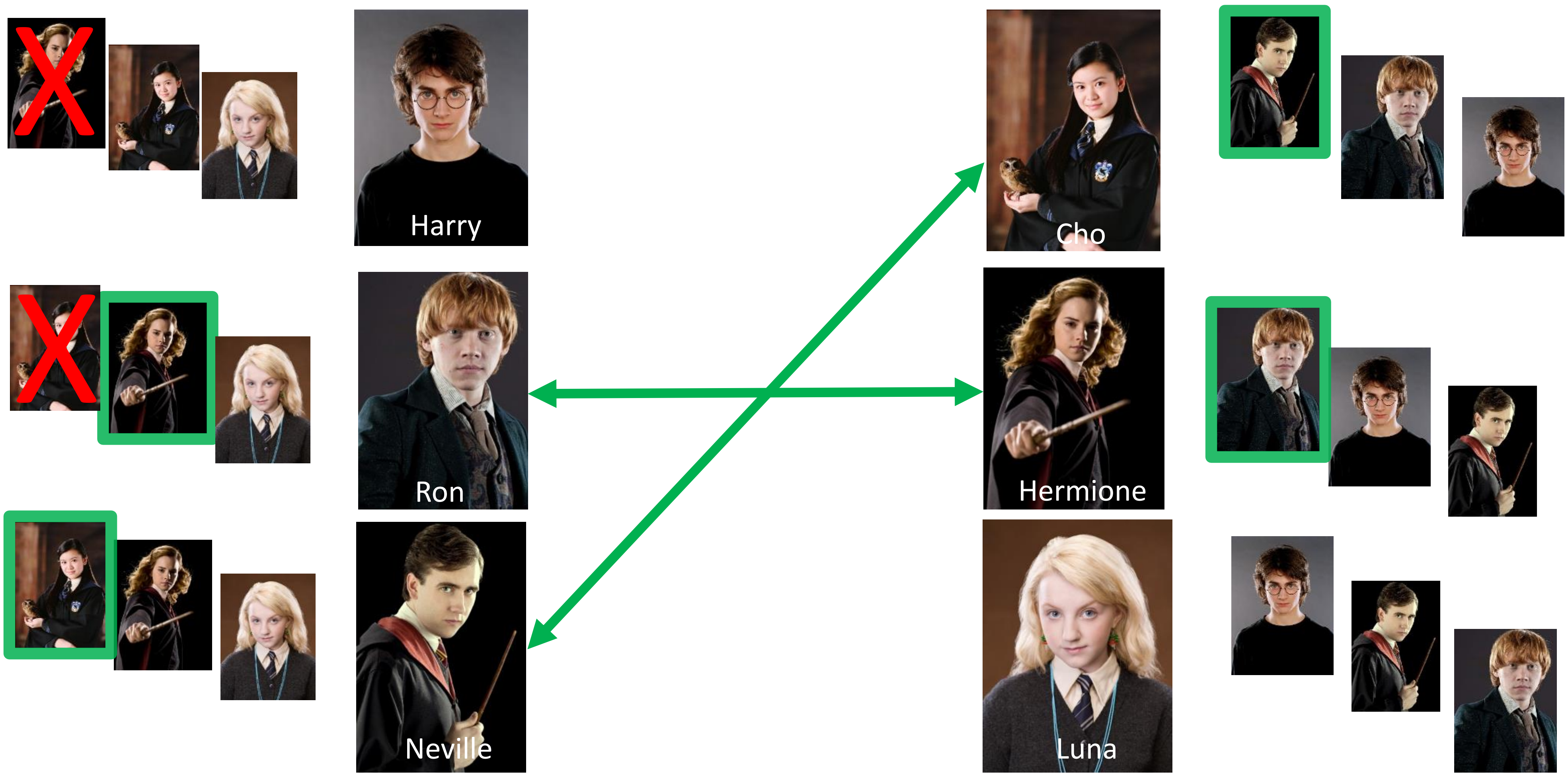
Example



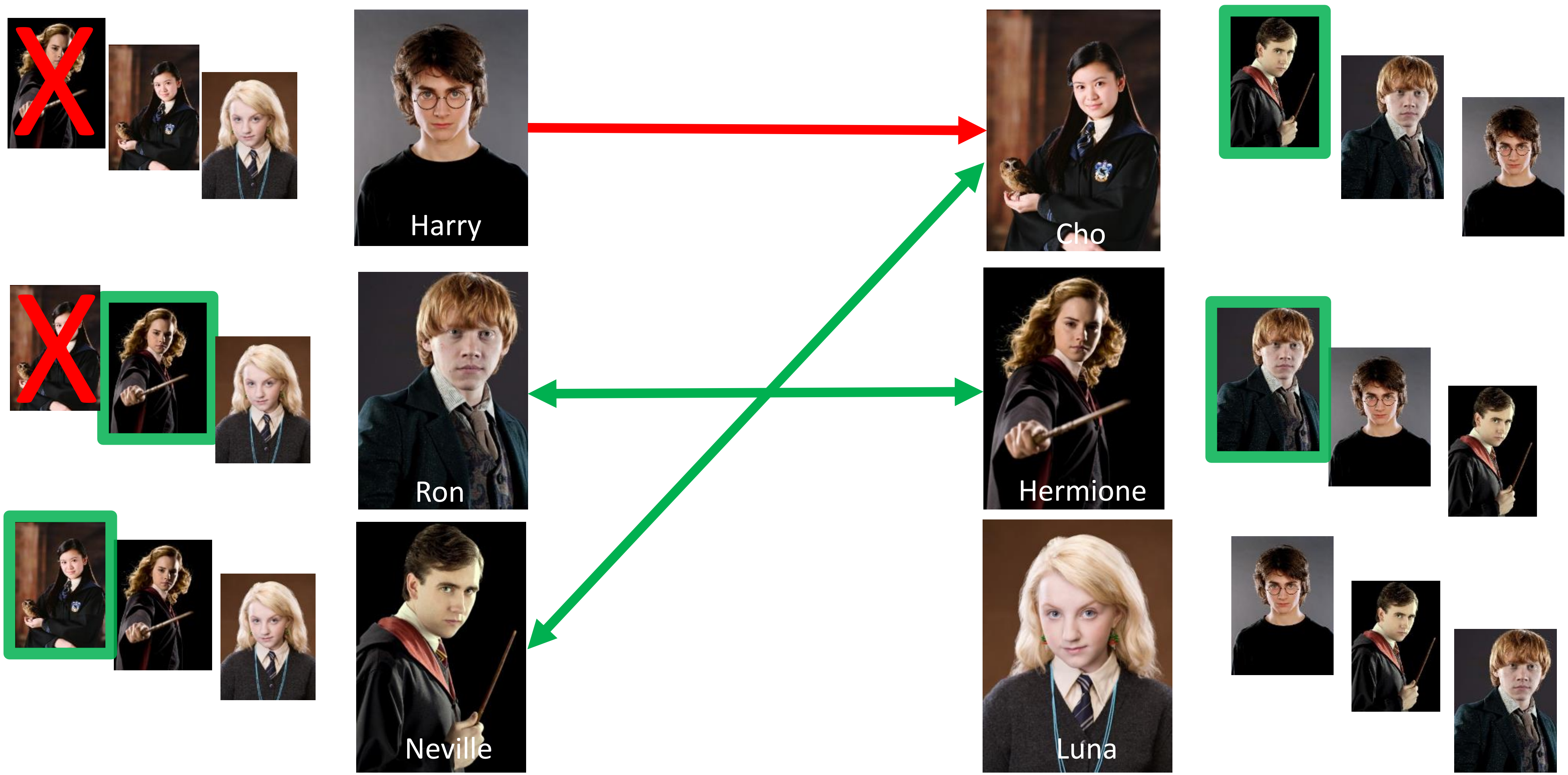
Example



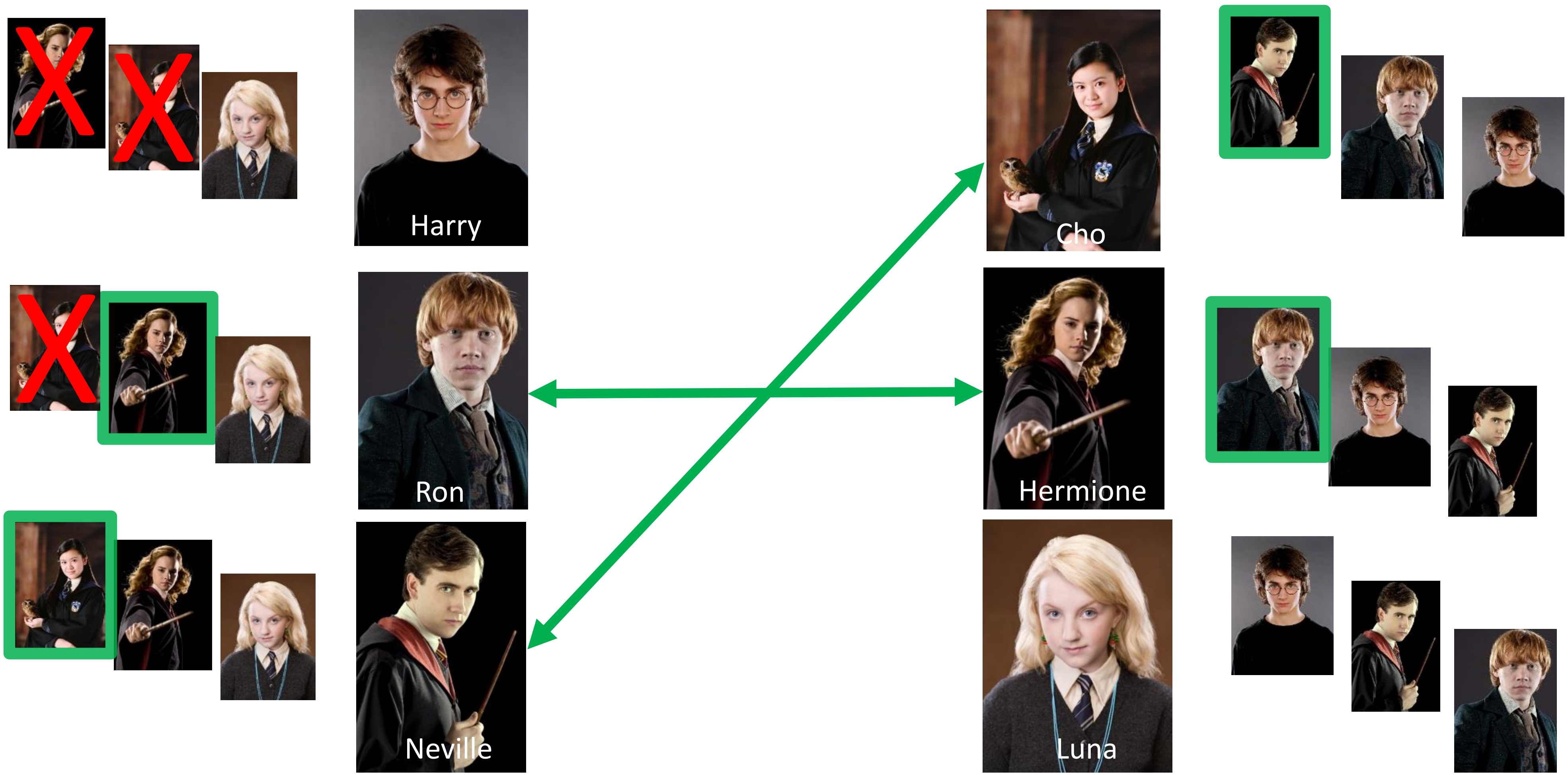
Example



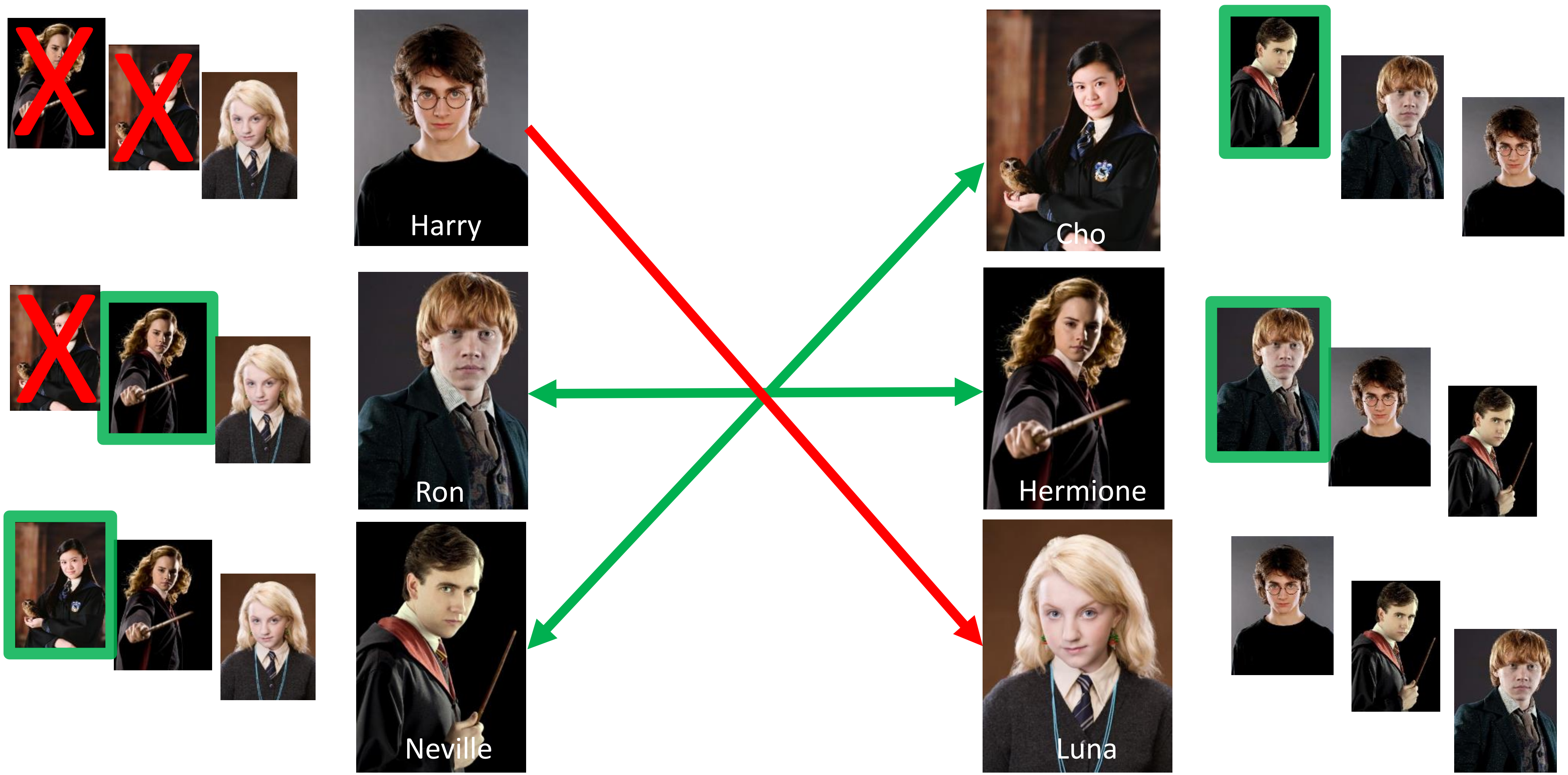
Example



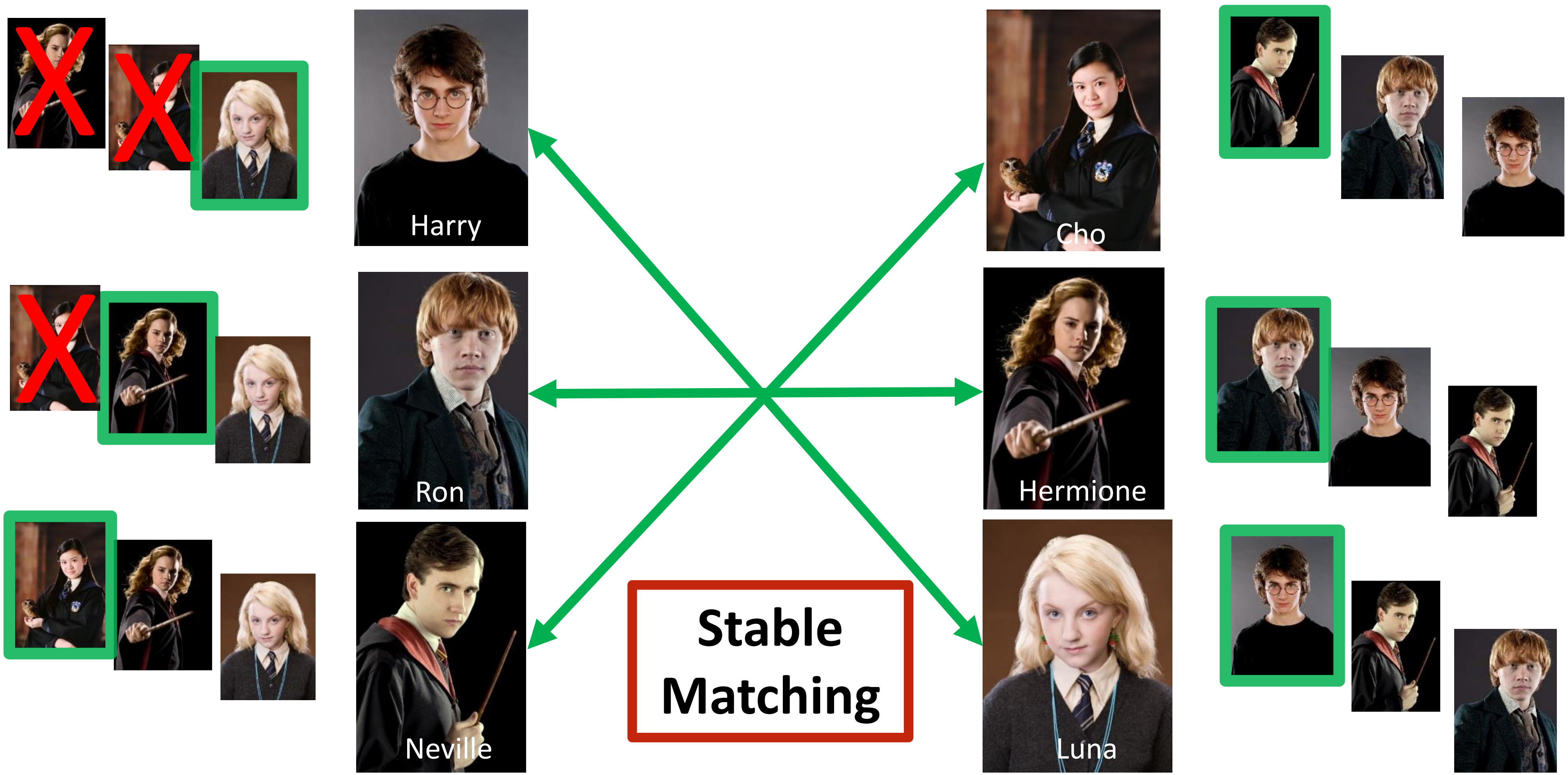
Example



Example



Example



Quiz

Go to vevox.com
Meeting ID: 110-844-851

- Which matching does GS return when leaders propose?

$f_1 > f_2 > f_3$	l_1	f_1	$l_1 > l_2 > l_3$
$f_1 > f_2 > f_3$	l_2	f_2	$l_1 > l_3 > l_2$
$f_1 > f_3 > f_2$	l_3	f_3	$l_1 > l_2 > l_3$

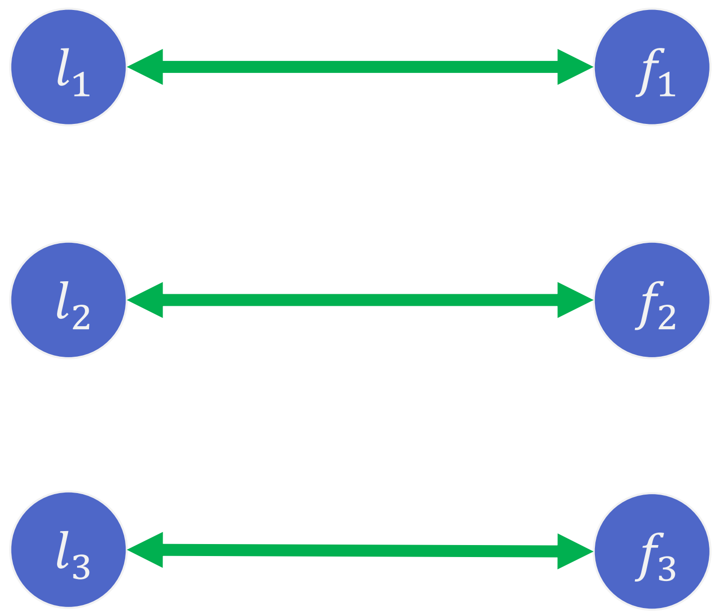
Answer

- Which matching does GS return when leaders propose?

$f_1 > f_2 > f_3$

$f_1 > f_2 > f_3$

$f_1 > f_3 > f_2$



$l_1 > l_2 > l_3$

$l_1 > l_3 > l_2$

$l_1 > l_2 > l_3$

Quiz

- What if followers propose?

$f_1 > f_2 > f_3$ l_1 f_1 $l_1 > l_2 > l_3$

$f_1 > f_2 > f_3$ l_2 f_2 $l_1 > l_3 > l_2$

$f_1 > f_3 > f_2$ l_3 f_3 $l_1 > l_2 > l_3$

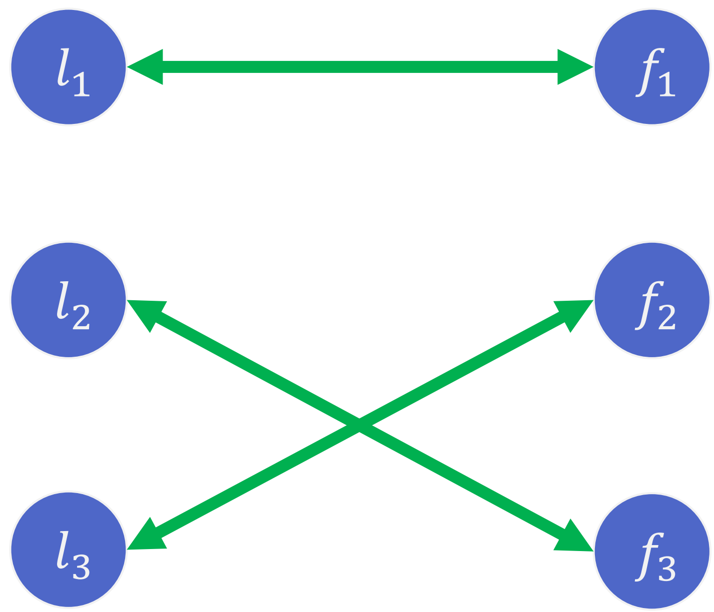
Answer

- What if followers propose?

$f_1 > f_2 > f_3$

$f_1 > f_2 > f_3$

$f_1 > f_3 > f_2$



$l_1 > l_2 > l_3$

$l_1 > l_3 > l_2$

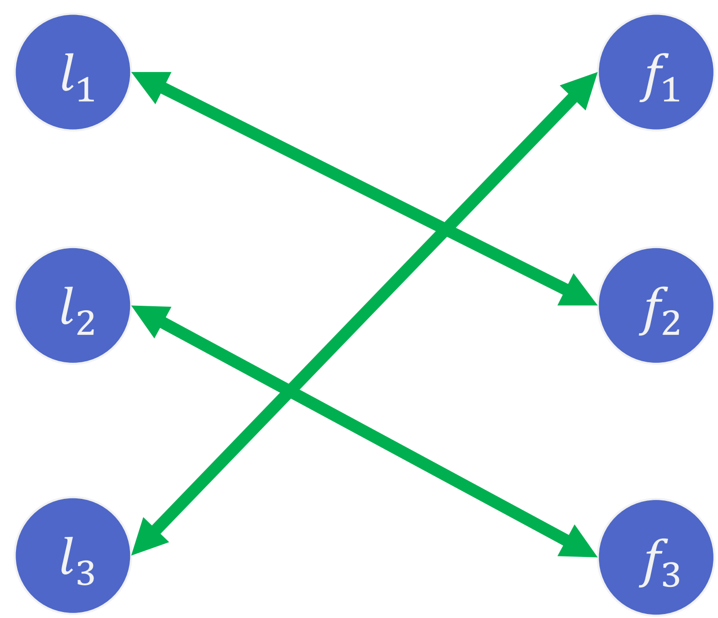
$l_1 > l_2 > l_3$

How many blocking pairs?

$f_1 > f_2 > f_3$ $\quad$ $l_1 > l_2 > l_3$

$f_1 > f_2 > f_3$ $\quad$ $l_1 > l_3 > l_2$

$f_1 > f_3 > f_2$ $\quad$ $l_1 > l_2 > l_3$



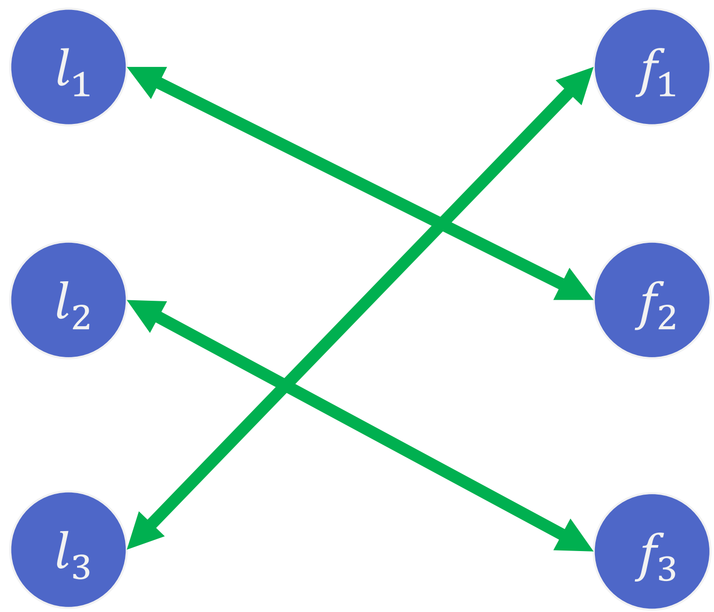
Answer

- 2 blocking pairs: (l_1, f_1) and (l_2, f_1)

$f_1 > f_2 > f_3$

$f_1 > f_2 > f_3$

$f_1 > f_3 > f_2$



$l_1 > l_2 > l_3$

$l_1 > l_3 > l_2$

$l_1 > l_2 > l_3$

Gale-Shapley algorithm

Deferred-acceptance-leader-oriented (leaders, followers, preferences)

```
Assign all leaders and followers to be free; //initial state
While (some leader l is free and hasn’t proposed to every follower)
    f = first follower on l's list to whom l hasn't yet proposed;
    // next: l proposes to f
    If (f is free)
        assign l and f to be engaged; //tentatively matched
    else if (f prefers l to her fiancé l') { //f is engaged
        set l and f to be engaged;
        set l' to be free;
    else  f rejects l; //and l remains free
output the n engaged pairs, who form a stable matching;
```

Gale-Shapley returns a stable matching

Claim. Gale-Shapley algorithm always terminates and returns a stable matching.

In order to prove this we need to show that:

1. The algorithm always terminates.
2. The algorithm returns a matching (every leader is matched with at most one follower and vice versa).
3. The matching it returns is stable (no blocking pair).

Proof of correctness: 1. Termination

Observation 1. Leaders propose to followers in decreasing order of preference

Observation 2. Once a follower is matched up, s/he never becomes unmatched; only ``trades up''.

Claim. Algorithm terminates after at most n^2 iterations of While loop.

Proof. Each time through the while loop, a leader proposes to a new follower. Thus there are at most n^2 possible proposals.

Proof of correctness: 2. Matching

Claim. Gale-Shapley outputs a matching.

Proof.

- Leader proposes only if unmatched $\Rightarrow$ matched to ≤ 1 follower.
- Follower keeps only best leader $\Rightarrow$ matched to ≤ 1 leader.

The matching is perfect (i.e. everyone is matched)

Claim. In Gale-Shapley matching, all leaders get matched.

Proof. [by contradiction]

- Suppose, for a contradiction, that some leader $\boldsymbol{l}$ is unmatched when Gale-Shapley terminates.
- Then some follower, say $\boldsymbol{f}$, is unmatched upon termination.
- By Observation 2, $\boldsymbol{f}$ was never proposed to.
- But, $\boldsymbol{l}$ proposes to every follower, since $\boldsymbol{l}$ ends up unmatched
- A contradiction!

Claim. In Gale-Shapley matching, all followers get matched.

Proof. [by counting] By previous claim, all **n** leaders get matched. Thus all **n** followers get matched.

Proof of correctness: 3. Stability

Claim. in Gale-Shapley matching μ, there are no blocking pairs.

Proof. Consider any pair (l, f) that is not in μ.

- Case 1: l never proposed to f

  $\Rightarrow$ l prefers $\mu(l)$ to f. (since leaders propose in decreasing order of preferences}

  $\Rightarrow$ (l, f) is **not blocking**

  Partner of l in μ

- Case 2: l proposed to f

  $\Rightarrow$ f rejected l (either right away or later)

  $\Rightarrow$ f prefers $\mu(f)$ to l (as followers only trade up)

  $\Rightarrow$ (l, f) is **not blocking**

  Partner of f in μ

- In either case, the pair (l, f) is **not a blocking pair**

Books

- Algorithmics of Matching under Preferences by David F. Manlove.

- Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis by Alvin E. Roth, Marilda A. Oliviera Sotomayor.

- Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations (**MAS**) by Yoav Shoham and Kevin Leyton-Brown

- Algorithmic Game Theory (**AGT**), edited by Noam Nisan, Tim Roughgarden, Eva Tardos,Vijay V. Vazirani

Acknowledgeleaderst

Some of the slides in this lecture were based on the slides by **Jie Zhang** and **Kevin Wayne**.