CPSC 320 Notes, The Stable Matching Problem, Part II

Here you'll analyze the Gale-Shapley (G-S) algorithm for the Stable Matching Problem, and learn principles for constructing proofs more generally. You've already read the analysis provided in the text. It's best not to look it up now, but see if you can reconstruct the analysis yourself, come up with alternative ways to reason about the algorithm, and uncover new properties of the algorithm.

Algorithm G-S(n,P_E, P_A) // return a stable matching M for the input instance, where:
// $n \geq 1$ is the number of employers and also the number of applicants
// P_E is the collection of complete preference lists of the employers
// P_A is the collection of complete preference lists of the applicants

```
M ← ∅ // matching M is initially empty
While some employer is unmatched and has not considered every applicant
    Choose any such employer e
    Let a be the highest-ranked applicant in e's preference list that e has not yet considered
    // e now considers a (i.e., "makes an offer" to a) as follows:
    If a is unmatched
        Add match e:a to M                    // a accepts e's offer
    Else // a is matched
        Let a be currently matched to e'
        If a prefers e to e'
            Remove match a:e' from M          // a rejects e''s offer
            Add match a:e to M                // a accepts e's offer
    // Else a prefers e' to e, in which case M does not change
Return M
```

1 Correctness Conditions

Write down what it means for an algorithm for the Stable Matching Problem to be correct on a valid iput.

2 The G-S Algorithm Always Terminates

Prove that algorithm G-S terminates on every valid input.

3 True or False

Choose whether you think each of the following statements about Algorithm G-S is true or false for all valid inputs. Do your best to also write down explanations of your answers. (Get in the practice of explaining your answers to true/false or multiple-choice questions such as this, and always include short explanations in assignments and exams unless explicitly asked not to.) If you don't have convincing justification, just write down your best intuition - some of these may be hard to justify.

1. Every applicant has been considered at least once when the algorithm terminates.

2. Some employer gets its highest-ranked applicant.

3. When the algorithm terminates, each employer is matched with the last applicant it considered.

4 The G-S Algorithm Always Outputs a Valid Solution

Prove that the G-S algorithm always outputs a valid solution, i.e., a perfect matching.

5 The G-S Algorithm Always Outputs a Good Solution

Prove that the G-S algorithm always outputs a a perfect matching M such that there are no instabilities with respect to M.

1. Recall and write down the definition of instability with respect to M.

2. Using your definition of instability, show that there are no instabilities with respect an output M of the G-S algorithm.

   One natural approach is to show that the (partial) matching constructed after each iteration of the While loop avoids instabilities. Let M_k be the matching at the end of iteration $k \geq 1$ of the While loop, and let E_k and A_k be the set of employers and the set of applicants, respectively, that are matched in M_k. Show by induction that there is no instability in $E_k \times A_k$ with respect to M_k.

   Claim: There is no instability in $E_k \times A_k$ with respect to M_k.

6 True or False Again

You've now proved that the G-A algorithm is correct. But there are other interesting properties of the algorithm, and stable matchings more generally, that are valuable to understand. For example, choose whether you think the following statements are true or false.

1. Any valid instance of the Stable Matching Problem has exactly one good solution.

2. The output of the G-A algorithm is independent of the order in which employers are chosen in the While loop.

7 Employer-Optimal Matchings

1. Write down more precisely what it means for an algorithm to be "employer optimal". Specifically, for a given employer e, define what is e's "best stable applicant". We'll denote this applicant by best(e).

2. It turns out that no two employers have the same best stable applicant. Prove this.

   Claim: Each employer e has a distinct best(e).

8 The G-S Algorithm Is Optimal for Employers

We can use induction also to show that the G-S algorithm outputs the (unique) employer-optimal stable matching in which every employer e is matched with best(e).

1. Prove the following claim.

   Claim: At the end of every iteration k of the While loop, every employer e has only considered applicants that it ranks at least as high as best(e). Moreover, if e has considered best(e) on or before iteration k, then e is matched with best(e) at the end of iteration k.

2. Using the claim, show that the output of the G-S algorithm must be employer-optimal.

9 Useful Proof Strategies

Stepping away from Stable Matching now, discuss and summarize useful things that you've learned more generally about writing proofs, while doing this worksheet.