

# CS70 Course Notes

Anmol Parande

Fall 2019 - Professors Alistair Sinclair and Yun Song

**Disclaimer:** These notes reflect CS60 when I took the course (Fall 2019). They may not accurately reflect current course content, so use at your own risk. If you find any typos, errors, etc, please report them on the GitHub repository.

## Contents

<b>1</b>	<b>Logic and Proofs</b>	<b>2</b>
1.1	Propositional Logic . . . . .	2
1.2	Proofs . . . . .	3
1.2.1	Direct Proof . . . . .	3
1.2.2	Proof by Contraposition . . . . .	3
1.2.3	Proof by Contradiction . . . . .	3
1.2.4	Proof by Cases . . . . .	4
1.2.5	Proof by Induction . . . . .	4
1.2.6	Strong Induction . . . . .	4
1.3	Case Study: Stable Marriage Algorithm . . . . .	4
1.3.1	The algorithm . . . . .	4
1.3.2	Proving the Algorithm . . . . .	5

# 1 Logic and Proofs

## 1.1 Propositional Logic

**Definition 1** A proposition  $P$  is a statement that is either true or false

Propositions can depend on one or more variables. This is denoted by  $P(x, y, \dots)$

**Definition 2** A connective is an operator which joins two or more propositions together in some way

Connectives are fully defined by a **Truth Table** which enumerates the values of the connective given all possible combination of inputs.

**Definition 3** The base connectives are those which can be combined to create any other connective.

- $\wedge$ :  $P$  AND  $Q$
- $\vee$ :  $P$  OR  $Q$
- $\neg$ : NOT  $P$

One important connective is the **implies** connective.

P	Q	$P \implies Q$
T	T	T
T	F	F
F	T	T
F	F	T

Notice that  $P \implies Q$  has the same truth table as  $\neg P \vee Q$ . This means they are equivalent.

**Definition 4** The contrapositive of  $P \implies Q$  is  $\neg Q \implies \neg P$

**Notice:** The contrapositive is logically equivalent to the original statement

**Definition 5** The converse of  $P \implies Q$  is  $Q \implies P$

**Notice:** This is not always equivalent to the original statement The **if and only if** connective  $P \iff Q$  is equivalent to  $(P \implies Q) \wedge (Q \implies P)$

P	Q	$P \iff Q$
T	T	T
T	F	F
F	T	F
F	F	T

Quantifiers help introduce variables into our propositions.

- $\forall$ : For all

- $\exists$ : There exists

**Important:** The order of quantifiers matters.

- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
- $\neg(\forall x P(x)) \equiv \exists x(\neg P(x))$
- $\neg(\exists x P(x)) \equiv \forall x(\neg P(x))$

## 1.2 Proofs

**Definition 6** A proof is a sequence of statements, each of which follows from the preceding ones by a valid law of reasoning.

**Definition 7** An Axiom is a basic fact which can be assumed without proof

### 1.2.1 Direct Proof

**Goal:** Prove  $P \implies Q$

**Approach:**

1. Assume  $P$
2. Deduce  $Q$  through logical steps

### 1.2.2 Proof by Contraposition

Since the contrapositive of a statement is logically equivalent to the original statement, we can prove the original statement by proving the contrapositive.

**Goal:** To prove  $P \implies Q$

**Approach:**

1. Assume  $\neg Q$
2. Deduce  $\neg P$  through logical steps

**Theorem 1** The pigeonhole principle says that if  $n$  objects are placed in  $k$  boxes and  $n > k$ , then some box contains  $\geq 1$  object.

### 1.2.3 Proof by Contradiction

**Goal:** Prove  $P \implies Q$  **Approach:**

1. Assume  $\neg P$
2. Deduce  $\neg P \implies (R \wedge \neg R)$

$R$  and  $\neg R$  are two facts which can be deduced from assumption  $\neg P$

### 1.2.4 Proof by Cases

**Goal:** Prove  $P \implies Q$

**Approach:**

1. Break  $P$  into cases  $1 \dots n$
2. Prove  $P$  holds in all cases

### 1.2.5 Proof by Induction

**Goal:** Prove  $\forall n, P(n)$  **Approach:**

1. Prove a based case
2. Assume  $P(k)$  (Induction hypothesis)
3. Prove  $P(k) \implies P(k+1)$  (Inductive Step)

### 1.2.6 Strong Induction

Because induction requires that  $P(k)$  be true to prove  $P(k+1)$ , it might be easier to just assume  $P(0) \wedge P(1) \wedge \dots \wedge P(k)$  when proving  $P(k+1)$ .

## 1.3 Case Study: Stable Marriage Algorithm

### 1.3.1 The algorithm

The stable marriage algorithm uses a list of preferences to make pairings from two disjoint groups of people.

Input :

- $n$  men,  $n$  women
- For each man, a ranked list of the women
- For each woman, a ranked list of the men

Goal: Pair men and women up in a way such that two people from different pairings would be happier together if they switched partners

Algorithm :

Each Day:

1. Each man proposes to the first woman on his list who has not yet rejected him
2. Each woman says "maybe" to the man she likes best among her proposals and rejects the others
3. Each man crosses off his list the woman who just rejected him

Repeat until there are no rejections

### 1.3.2 Proving the Algorithm

There are four things we must prove in order to make sure that the stable marriage algorithm works.

1. SMA terminates
2. SMA must output a pairing
3. The pairing outputted by SMA is stable

Beginning with proving the SMA terminates,

**Proof 1 (SMA Terminates)** *On each iteration, when SMA doesn't halt, some man got rejected. If a man gets rejected, he crossed a woman off his list. If every man gets rejected by every woman on his list, then there is a maximum of  $n^2$  rejections. This will take a maximum of  $n^2$  days, therefore the algorithm terminates after  $n^2$  days.*

Before proving that the SMA must output a pairing, it is first useful to prove something called the improvement lemma

**Lemma 1 (Improvement lemma)** *Suppose  $M$  proposes to  $W$  on iteration  $K$ . On every day  $J \geq K$ ,  $W$  has said "maybe" to a man she likes at least as much as  $M$*

**Proof 2 (Improvement Lemma)** *Proceed with induction on the number of days  $k$*

**Base Case:**  $J = K$

$M$  proposes to  $W$ , so the statement holds because even if  $M$  is her worst proposal, she will only say "maybe" to a better man.

**Induction Hypothesis:** *Assume the lemma is true for day  $J$*

**Inductive Step:**

*On day  $J$ ,  $W$  has said "maybe" to  $M'$  she likes at least as much as  $M$ .  $M'$  will propose again on day  $J + 1$ , so no matter who else proposes to  $W$ , she will at least have  $M'$  who is better than  $M$*

**Proof 3 (SMA outputs a pairing)** *Suppose the algorithm produces no pairing. Then there exists  $M$  who is rejected by all  $n$  women when the algorithm terminates. By the improvement lemma, all women at the end of the algorithm have a man whom they've said maybe to. So  $n$  women can only have rejected  $n - 1$  men which is a contradiction, therefore SMA must output a pairing.*

Before proving that the output of SMA is stable, we should define what stability means.

**Definition 8** *A pairing outputted by SMA is stable if there is not a man and a woman who would rather be with each other than their current partners (a rogue couple).*

**Proof 4 (SMA outputs a stable pairing)** Suppose the pairing outputted by SMA is  $\{...(M, W)...(M', W')\}$  and  $M$  likes  $W'$  more than  $W$ . Since  $M$  likes  $W'$  more than  $W$ ,  $M$  must have proposed to  $W'$ . By the improvement lemma,  $M'$  must be better than  $M$ , so  $(M, W')$  is not a rogue couple, so there are no rogue couples and the pairing is stable.

**Definition 9** For a given  $M$  the optimal  $W$  is the highest woman on his list that he is paired with in any stable pairing

**Definition 10** The male optimal pairing is the one in which every man is paired with his optimal woman.

**Proof 5 (SMA is male-optimal)** Proceed by induction on  $k$  (the number of days).  $\forall k \geq 0$ , on day  $k$ , no man gets rejected by his optimal woman.

**Base Case:**  $k = 0$

Nothing to prove, no rejections have happened

**Inductive Step:**

Assume for all  $0 \leq k \leq j$ , no man is rejected by his optimal woman. Suppose for contradiction that on day  $j + 1$   $M$  gets rejected by his optimal woman  $W^*$  meaning some other man was accepted  $M^*$ . Since  $W^*$  is  $M$ 's optimal woman, there exists a stable pairing  $\{...(M, W^*)...(M^*, W')\}$  since  $W^*$  prefers  $M^*$  to  $M$  because she rejected  $M$ . By the inductive hypothesis,  $M^*$  has not yet been rejected by his optimal woman, so he likes  $W^*$  at least as much as  $W'$  because there is a stable pairing with  $W'$ . Hence  $(M^*, W^*)$  is rogue because they like each other better than their partners.

**Theorem 2** The male-optimal pairing is the female-pessimal pairing.