CS Theory: Proof Review

Sophia Kolak

Roadmap

1. Mathematical Notation
2. Propositional Logic
3. Proof Techniques
4. Examples

For more proof writing info:
http://www.math.columbia.edu/~ums/introproofs2020.html

Mathematical Notation

Theorem (Thm): A substantial mathematical statement which has been proven true.

Lemma: A smaller statement that needs to be proved as an intermediate step to proving a theorem.

Corollary (Cor): A consequence of a theorem which follows either immediately from it or from the theorem combined with other established facts.

RHS: right hand side of an equation

LHS: left hand side of an equation

WLOG: without loss of generality, an assumption that does not limit the scope of your proof to specific cases

Mathematical Notation

- $\forall$ ↠ for all
- $\exists$ ↠ there exists
- $\nexists$ ↠ there does not exist
- $\Leftrightarrow$ ↠ if and only if (iff)
- $\Rightarrow$ ↠ LHS implies RHS
- $\Leftarrow$ ↠ RHS implies LHS
- $:$, $|$ ↠ such that

- $\{\,\}$ ↠ set (languages are sets of strings)
- $\subset$ ↠ LHS is proper subset of RHS
- $\not\subset$ ↠ LHS is not proper subset RHS
- $\subseteq$ ↠ LHS is a subset of RHS (not proper, meaning LHS might also equal RHS, similar to $\leq$ sign

Common CST Notation

Σ, alphabet	A finite set of symbols (all strings of length 1 in the language)
String	Any finite sequence of 0 or more symbols
Σ*	Set of possible strings that can be made with the alphabet, ex {a,b}*
Language L over **Σ***	Any subset of strings $L \subseteq \Sigma^*$
L*	Set of strings that can be obtained by concatenating 0 or more strings from L. Ex: L={a,bbb}, L* = {ε, a, bbb, aa, abbb, bbba, bbbbbb, aaa, aabbba, ...}

Common CST Notation

String $w = w_1 w_2 \ldots w_n$	$w_1 w_2 \ldots w_n$ are the symbols that compose w
$w \in L$	The string w is in the language L
$\lvert w \rvert$	The length of the string w (Ex: $\lvert \varepsilon \rvert = 0$, $\lvert aaab \rvert = 4$)
$P \cup R$	The union of P and R
PR, $P.R$, $P \circ R$	The concatenation of P and R
$\{a,b\}^*$	All strings that can be made with a's and b's

Mathematical Notation - Example

How would you read the following language definition (in English words)?

$$ins(L) = \{w \mid w = xat, \text{ where } x \in \Sigma, a \in \Sigma, t \in \Sigma^*, \text{ and } xt \in L\}$$

Mathematical Notation - Example

How would you read the following language definition (in English words)?

$$ins(L) = \{w \mid w = xat, \text{ where } x \in \Sigma, a \in \Sigma, t \in \Sigma^*, \text{ and } xt \in L\}$$

The language called "ins(L)" is composed of strings w such that w=xat, where x is some symbol in the alphabet, a is some symbol in the alphabet, t is a string in the alphabet, and xt is in the language L

Roadmap

1. Mathematical Notation
2. Propositional Logic
3. Proof Techniques
4. Examples

Propositional Logic

Proofs build on logical statements that are either true or false.
Let P and Q be two statements

NOT, ¬ : NOT P

- True when P is false
- False when P is true

AND, ∧ : (P and Q)

- True only when P and Q are both True

OR, ∨: (P or Q)

- False only when both P and Q are False

Propositional Logic

If/then, $\Rightarrow$: (P implies Q)

- True when Q is true or P is false
- False only when P is true and Q is false

If/only If (IFF), $\Leftrightarrow$: (P iff Q)

- True when P and Q are both True or both False
- False when P and Q have different truth values

Converse: The converse of $P \Rightarrow Q$ is $Q \Rightarrow P$

- True when Q and P are both True or Q is False
- False when Q is True and P is False

Contrapositive: The contrapositive of $P \Rightarrow Q$ is $(-Q \Rightarrow -P)$

- Same truth value as $P \Rightarrow Q$

Why?

Propositional Logic

P = John ate healthy = True

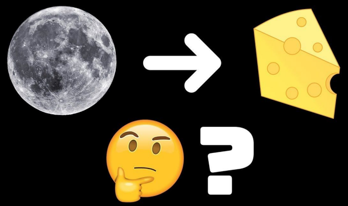
Q = John is in good health = True

If John ate healthy then John is in good health (T)
If John ate healthy then John is not in good health (F)
If John did not eat healthy then John is in good health (T)
If John did not eat healthy then John is in not in good health (T)

P	Q	P ⇒ Q
T	T	T
T	F	F
F	T	T
F	F	T

Cannot be shown False unless the Premise is True. If the premise never occurred the conclusion is true by default

This is called a **vacuous truth**



Ex: If John did not eat healthy, then the moon is cheese

Propositional Logic

P = John ate healthy = True

Q = John is in good health = True

John ate healthy ⇔ John is in good health
If John is in good health, then John ate healthy *(both sides imply each other)*
If John ate healthy, then John is in good health

P	Q	P⇒Q	Q⇒P	(P⇒Q) ∧ (Q⇒P)	P⇔Q
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

Quick Note on Sets:

Given a set S and a propositional statement P
we can define the new set R

$$R := \{s \in S \mid P(s)\}$$

What is R?

The set of all elements in S for which the statement P is True

S = {1,2,3,4,5,6,7,8,9,10}   P = the number is even

Then R = {2,4,6,8,10}

Roadmap

1. Mathematical Notation
2. Propositional Logic
3. Proof Techniques
4. Examples

Proof Techniques – Quantifiers

Existential: "There Exists", $\exists$, you want to show that there is at least one element within a given set satisfies a certain statement

$$\exists x \in S : P(x) \Leftrightarrow P(x) \text{ is true for at least one } x \in S$$

Universal: "For all", $\forall$, you want to show that every element in the set satisfies a certain statement

$$\forall \mathrm{x} \in \mathrm{S} : \mathrm{P(x)} \Leftrightarrow P(x) \textit{ is true for every } x \in S$$

Proof By Construction

- Used when a theorem States that a particular type of object exists
- Can be proved by a demonstration of how t construct the object
- For any Y there is an X
- Should argue that the construction is actually an X
- Should show that it works for all possible Y
- Last two steps are not always necessary in this class

Proof By Construction- Example

Prove that regular languages are closed under reversal

Why can we use proof by construction here?

What do we need to show? Where to start?

RL/NFA/DFA construction proofs often involve manipulating the symbolic definition of a DFA or a NFA

*Start by symbolically defining an NFA $M = (Q, \Sigma, \delta, q_o, \{q_f\})$ that accepts the regular language L

Proof By Construction- Solution

Prove that regular languages are closed under reversal

Let $M = (Q, \Sigma, \delta, q_o, \{ q_f \})$ be an NFA accepts the regular language L

We can construct another NFA M\` such that

M\`   M

Q\` = Q

Σ\` = Σ

δ\` = δ (the same transitions with arcs reversed)

q_o\` = $q_f$ (the start state of M\` is the final state of M)

q_f\` = $q_o$ (the final state of M\` is the start state of M)

There is a path from q to q in M if and only if there is a path from q to q in M\`. Thus, L(M\`) = L^R

Proof By Contradiction

- Suppose you wanted to prove that $P \Rightarrow Q$
- Assume the opposite of what you want to prove
- Show that this assumption leads to a larger contradiction.

- Negating Quantified Statements:
  - $(-\exists)x, P(x)$ = P is not true for any x
  - $(-\forall)x, P(x)$ = P is true for some x

- Baby example:
  - Suppose $2+2 \neq 4$
  - Subtract 2 from both sides
  - Then we have derived $2 \neq 2$, which is a contradiction
  - Therefore, it must be the case that $2+2 = 4$

Proof By Contradiction - Example

Prove that the following language is not regular:
L = The language of binary strings with the same number of 1's and 0's

We use the pumping lemma!

Logic of the pumping lemma: (if L is regular, then L satisfies the pumping lemma)

Contrapositive (always has same truth value!)
If L does not satisfy the pumping lemma, then L is not regular.

We suppose L is regular, then if L does not satisfy the pumping lemma, we will have reached a contradiction, which means L must not be regular.

Proof By Contradiction - Solution

Prove that the following language is not regular:
L = The language of binary strings with the same number of 1's and 0's

Suppose, for the sake of contradiction, that L is regular. Then it must satisfy the pumping lemma.

For any positive integer n, we consider the string $w = 0^n 1^n$. Clearly $w \in L$. If we partition w in any way such that $|xy| \leqslant n$ and y is non-empty, then y must contain only 0's. Thus xy^2z will contain more zeros than ones and will not be in L.

Why was this a valid proof?

Induction Proof

- State what you're trying to prove (this is your inductive hypothesis)
- Establish a base case for a simple example (often in this class, strings of length one or zero)
- Assume the inductive hypothesis is true, and use it to prove the inductive step (show that P(n) implies P(n+1))

Induction Proof - Example

If L is regular, then L^{k} is regular

Induction Proof - Example

If L is regular, then L^{k} is regular

Base Case: let k =1 Then L^{1}=L, hence regular.

Inductive Step: Assume L^{k} is regular.
L^{k+1} = L^{k}.L
Since the regular languages are closed under concatenation,
L^{k+1} is regular.

HW1 - Problem #3

Let $L_1, L_2 \subseteq \Sigma^*$ be arbitrary languages, prove that $L_2 \subseteq L_1 \circ L_2$ if and only if $\varepsilon \in L_1$ or $L_2 = \varnothing$

Remember, this is an IFF (if and only if) so we have to prove both sides

HW1 - Problem #3 - First Side

Let $L_1, L_2 \subseteq \Sigma^*$ be arbitrary languages, prove that $L_2 \subseteq L_1 \circ L_2$ if and only if $\varepsilon \in L_1$ or $L_2 = \varnothing$

1. *If $\varepsilon \in L_1$ or $L_2 = \varnothing$, then $L_2 \subseteq L_1 \circ L_2$*

(start with the easier side first!)
If $L_2 = \varnothing$, then $L_2 \subseteq L_1 \circ L_2$ (any string in empty set is also in any other set, including $L_1 \circ L_2$)

If $\varepsilon \in L_1$, then we can write any element $x \in L_2$ as εx, which is $\in L_1 \circ L_2$

HW1 - Problem #3 -Second Side

Let $L_1, L_2 \subseteq \Sigma^*$ be arbitrary languages, prove that $L_2 \subseteq L_1 \circ L_2$ if and only if $\varepsilon \in L_1$ or $L_2 = \varnothing$

2. If $L_2 \subseteq L_1 \circ L_2$, then $\varepsilon \in L_1$ or $L_2 = \varnothing$

If $L_2 = \varnothing$, then this is clearly true because the RHS is true.
If $L_2 \neq \varnothing$, then L_2 contains at least one string.
Choose $x \in L_2$ with the smallest length (if multiple, choose one arbitrarily)
Now, since $L_2 \subseteq L_1 \circ L_2$, it must be that $x \in L_1$ concat L_2
So we can write $x = yz$ where $y \in L_1$, $z \in L_2$
If y not ε, then $|y| \geqslant 1$, and so z is a string in L_2 with a length smaller than x.
However, we chose x to be a string in L_2 with the smallest length so this is impossible!
This means the only possibility is that $y = \varepsilon$ and $z = x$ so $\varepsilon \in L_1$.

HW1 - Problem #3 - Discussion

Let $L_1, L_2 \subseteq \Sigma^*$ be arbitrary languages, prove that $L_2 \subseteq L_1 \circ L_2$ if and only if $\varepsilon \in L_1$ or $L_2 = \varnothing$

How did we prove each side
(first side, direct proof)

Second side (direct proof)

When proving and IFF and only if, you have to prove both that P↣Q and that Q↣P,
One proof for the price of two….

but you can also use different techniques for each proof if necessary.

Final Questions?

For more proof writing info:
http://www.math.columbia.edu/~ums/introproofs2020.html