CSE 311 LaTeX Template

I collaborated with Jacob Berg and Alysa Meng

- This is hopefully a (slightly) comprehensive guide to typing up assignments for 311 in LaTeX!
- If you have any questions/things to add, feel free to post on Ed, or email me at jacob33@uw.edu
- To use this, Copy the template and then delete all of the items inside the document section.
- **You will not understand most of the example problems right away.** That's totally fine! I made sure to include problems up to week 10 so that you can see example formatting for each of them.
- Whenever you see the tbs command in the code, for example, \LaTeX, I am using the tbs command to render the backslash in the PDF. You should instead write LaTeXdirectly to use the command.

1 Basic Formatting

In LaTeX, the \section command is used to create a new numbered section. You can put a * after the section command, i.e., \section*, to create non-numbered sections.
To create a new line, you can use the (\\) command at the end of the line
You can also create a line break with one blank line (both shown above). Multiple blank lines are the same as 1 blank line indicating a new paragraph to LaTeX. To create multiple blank lines, you can use the ˜ character to chain them together, i.e \\˜\\.

In LaTeX, single line breaks without a blank line inbetween and tabs/indentation do not matter and are ignored. Use the % symbol to comment in your document. You can also use the \vspace{xmm} command to insert white space directly.

To center items, put them inside the \center environment.

To create a new page, use the \newpage command.

You can also use the solution environment I created above to put your solutions in a green box, but that is optional.

Note: the tbs command here is used to escape the backslash. Read the comment on line 6 to see the definition. Commands can be more complicated and take arguments. They are not necessary to understand, but documentation is available here.

1.1 Text Formatting

You can also use the \subsection command to make a subsection in your document, although you probaly won't need this.
Bold text: **bold**, italicized text: *italic*, and underlined text: <u>underline</u>, sans-serif font sans-serif font.
You can also define commands using the \define\func-name{func content}. An example is above. Hello World!!

2 Math Formatting

There are a few options to format mathematical expressions. To format math inline, you can surround your math with singular dollar signs, i.e $x+2 \equiv -1^3$. To create centered major equations, you can either surround your equation with double dollar signs, or \[expression here \]. Examples below:

$$\cos(\theta) = \pi$$

or

$$\frac{\pi}{2} = 90^{\circ}$$

Another important environment is the align* environment. Note that the * removes the numbering, same as it does to the \section command. You are automatically in math mode inside this environment and can create multiple centered lines. Here are a few examples:
Multiple lines of equations:

$$\begin{aligned} x &= y \\ y &= z \\ z &= a \end{aligned}$$

Proof with justification:

$$\begin{aligned} x &= y && \text{thing 1} \\ y &= z && \text{thing 2} \\ z &= a && \text{thing 3} \end{aligned}$$

Other alignment:

$$\begin{aligned} &z \\ &\quad = x && \text{thing 1} \\ &\qquad = \pi && \text{thing 2} \\ &\qquad\quad = z && \text{thing 3 with symbol } \sum_i = 0^1 \end{aligned}$$

Inside the align environment, the & symbol indicates a point of alignment or column separation. The double and indicates seperate important alignment point. You can use \quad to add indentation, which will be useful in things like formal proofs. Be sure to not include blank lines in your align environment as this will cause errors.

$$\text{Here is an example} \tag{1}$$

$$\text{align without the * symbol.} \tag{2}$$

3 Symbols

Symbols in latex usually need to be in a math environment. In order to both super script, and subscript, you can use the underscore and carrot symbol. 2_1^{x+1}. The curly braces indicate the entire part in curly braces is the exponent or subscript.

Here are the symbols (I can think of) that you will need to use for this class. I am also using an enumerate environment which you can use to list things like parts of a question:

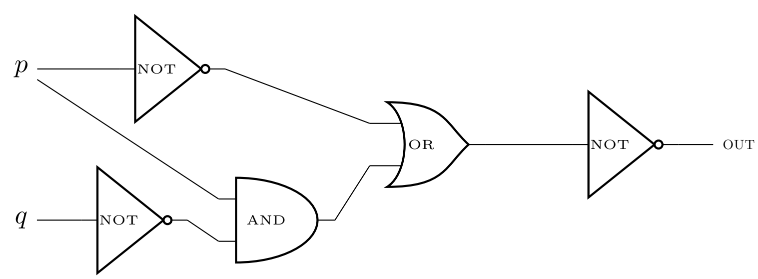
1. And, Or, Not, Xor, Implication, Biconditional: $\wedge, \vee, \neg, \oplus, \rightarrow, \leftrightarrow$
2. Equivalence, equivalence with subscripts, divides, sum: $\equiv, \equiv_4, |, \sum_i^j$
3. Less than, less than equal to, greater than, greater than equal to, times: $<, \leq, >, \geq, \cdot, \bullet$
4. Most special characters like pi can be created with their name: $\pi, \sigma, \Sigma, \cos, \max$
5. For all and exists: $\forall x, \exists y$
6. Cross product, union, intersection, set difference, empty set, curly braces, complement, in, not in: $\times, \cup, \cap, \setminus, \emptyset, \{\}, \overline{S}, \in, \notin$
7. Example Regex: $(1 \cup 0)^* \cup \varepsilon$
8. Natural, Integers, and Reals: $\mathbb{N}, \mathbb{Z}, \mathbb{R}$

You can also use the \mathcal or \mathbb to create certain symbols i.e $\mathbf{O}, \mathcal{O}$

To create a table, you can use the tabular environment where the c — c c part represents the columns and where the lines are. \hline is a horizontal line:

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

You can also use LaTeXfor circuits, but I personally draw and take a screenshot and then include them in my document with the \includegraphics[width=.x\linewidth]path/to/image. Inside the [] of the include graphic command, you can specify the width and the height with a number and unit, i.e., 12px, or 0.2\linewidth. There are also ways to include multiple subfigures, which is explained well here Here is an example circuit:



4 Example Translation

Translate the English sentences below into symbolic logic.

(a) If I am lifting weights this afternoon, then I do a warm-up exercise.

Since we're in "if...then..." form, the sentence is an implication.
p: I am lifting weights
q: I do a warm-up exercise

$$p \to q$$

(b) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.

Let the domain be all applications. Let `Firewall`(x) be "x is the firewall", and let `ProxyServer`(x) be "x is the proxy server." Let `Diagnostic`(x) be "x is in a diagnostic state".

$$\forall x \; \forall y \; ((\texttt{Firewall}(x) \land \texttt{Diagnostic}(x)) \to (\texttt{ProxyServer}(y) \to \texttt{Diagnostic}(y))$$

5 Example Equivalence Proof

1. $\neg p \to (q \to r) \equiv q \to (p \lor r)$

$$
\begin{aligned}
\neg p \to (q \to r) &\equiv \neg\neg p \lor (q \to r) && \text{[Law of Implication]} \\
&\equiv p \lor (q \to r) && \text{[Double Negation]} \\
&\equiv p \lor (\neg q \lor r) && \text{[Law of Implication]} \\
&\equiv (p \lor \neg q) \lor r && \text{[Associativity]} \\
&\equiv (\neg q \lor p) \lor r && \text{[Commutativity]} \\
&\equiv \neg q \lor (p \lor r) && \text{[Associativity]} \\
&\equiv q \to (p \lor r) && \text{[Law of Implication]}
\end{aligned}
$$

6 Formal Set Proof / English Set Proof

1. Prove that $(A \cap B) \times C \subseteq A \times (C \cup D)$ for any sets A, B, C, D.

1.	Let x be arbitrary	
2.1	$x \in (A \cap B) \times C$	Assumption
2.2	$(y, z) \in (A \cap B) \times C$	Def of $\times$
2.3	$y \in (A \cap B) \wedge z \in C$	Def of $\times$
2.4	$y \in (A \cap B)$	Elim $\wedge$
2.5	$y \in A \wedge y \in B$	Def of $\cap$
2.6	$y \in A$	Elim $\wedge$
2.7	$z \in C$	Elim $\wedge$
2.8	$z \in C \vee z \in D$	Intro $\vee$
2.9	$z \in (C \cup D)$	Def of $\cup$
2.10	$y \in A \wedge z \in (C \cup D)$	Intro $\wedge$
2.11	$(y, z) \in A \times (C \cup D)$	Def of $\times$
2.12	$x \in A \times (C \cup D)$	Def of $\times$
2.	$x \in (A \cap B) \times C \rightarrow x \in A \times (C \cup D)$	Direct Proof
3.	$\forall x (x \in (A \cap B) \times C \rightarrow x \in A \times (C \cup D))$	Intro $\forall$
4.	$(A \cap B) \times C \subseteq A \times (C \cup D)$	Def of $\subseteq$

Let x be an arbitrary element of $(A \cap B) \times C$. Then, by definition of the Cartesian product, x must be of the form (y, z) where $y \in A \cap B$ and $z \in C$. Since $y \in A \cap B$, $y \in A$ and $y \in B$ by definition of $\cap$; in particular, all we care about is that $y \in A$. Since $z \in C$, by definition of $\cup$, we also have $z \in C \cup D$. Therefore since $y \in A$ and $z \in C \cup D$, by definition of Cartesian product we have $x = (y, z) \in A \times (C \cup D)$.

Since x was an arbitrary element of $(A \cap B) \times C$ we have proved that $(A \cap B) \times C \subseteq A \times (C \cup D)$ as required.

7 Example GCD problem

1. Find the multiplicative inverse y of 7 mod 33. That is, find y such that $7y \equiv 1 \pmod{33}$. You should use the extended Euclidean Algorithm. Your answer should be in the range $0 \leq y < 33$.

First, we find the gcd:

$$\begin{aligned}
\gcd(33,7) &= \gcd(7,5) & 33 &= \boxed{7} \bullet 4 + 5 \quad (3)\\
&= \gcd(5,2) & 7 &= \boxed{5} \bullet 1 + 2 \quad (4)\\
&= \gcd(2,1) & 5 &= \boxed{2} \bullet 2 + 1 \quad (5)\\
&= \gcd(1,0) & 2 &= 1 \bullet 2 + 0 \quad (6)\\
&= 1 & & \quad (7)
\end{aligned}$$

Next, we re-arrange equations (1) - (3) by solving for the remainder:

$$1 = 5 - \boxed{2} \bullet 2 \quad (8)$$
$$2 = 7 - \boxed{5} \bullet 1 \quad (9)$$
$$5 = 33 - \boxed{7} \bullet 4 \quad (10)$$
$$(11)$$

Now, we backward substitute into the boxed numbers using the equations:

$$\begin{aligned}
1 &= 5 - \boxed{2} \bullet 2\\
&= 5 - (7 - \boxed{5} \bullet 1) \bullet 2\\
&= 3 \bullet \boxed{5} - 7 \bullet 2\\
&= 3 \bullet (33 - \boxed{7} \bullet 4) - 7 \bullet 2\\
&= 33 \bullet 3 + 7 \bullet -14
\end{aligned}$$

So, $1 = 33 \bullet 3 + \boxed{7} \bullet -14$. Thus, $33 - 14 = 19$ is the multiplicative inverse of 7 mod 33.

2. Now, solve $7z \equiv 2 \pmod{33}$ for all of its integer solutions z.

We already computed that 19 is the multiplicative inverse of 7 mod 33. That is, $19 \cdot 7 \equiv 1 \pmod{33}$.

If z is a solution to $7z \equiv 2 \pmod{33}$, then multiplying by 19 on both sides, we have $19 \cdot 7 \cdot z \equiv 19 \cdot 2 (\text{mod } 33)$.

Substituting $19 \cdot 7 \equiv 1 \pmod{33}$ into this on the left gives $1 \cdot z \equiv z \equiv 19 \cdot 2 \equiv 38 \equiv 5 \pmod{33}$.

This shows that every solution z is congruent to 5. In other words, the set of solutions is $\{5 + 33k \mid k \in \mathbb{Z}\}$.

8 Example Induction Problems

1. **Normal Induction** Show using induction that $0 + 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$ for all $n \in \mathbb{N}$.

For $n \in \mathbb{N}$ let $P(n)$ be "$0 + 1 + \cdots + n = \frac{n(n+1)}{2}$". We show $P(n)$ for all $n \in \mathbb{N}$ by induction on n.

Base Case: We have $0 = 0 = \frac{0(0+1)}{2}$ which is $P(0)$ so the base case holds.

Inductive Hypothesis: Suppose $P(k)$ holds for some arbitrary integer $k \geq 0$.

Inductive Step: $\boxed{\text{Goal: Show } 0 + 1 + \cdots + (k+1) = \dfrac{(k+1)(k+2)}{2}}$.

We have

$$\begin{aligned}
0 + 1 + \cdots + k + (k+1) &= (0 + 1 + \cdots + k) + (k+1) \\
&= \frac{k(k+1)}{2} + (k+1) && \text{[Inductive Hypothesis]} \\
&= \frac{k(k+1)}{2} + \frac{2(k+1)}{2} \\
&= \frac{k(k+1) + 2(k+1)}{2} \\
&= \frac{(k+1)(k+2)}{2} && \text{[Factor out } (k+1)\text{]}
\end{aligned}$$

This proves $P(k+1)$.

Conclusion: $P(n)$ holds for all $n \in \mathbb{N}$ by the principle of induction.

2. **Strong Induction**
Xavier Cantelli owns some rabbits. The number of rabbits he has in any given year is described by the function f:

$$\begin{aligned}
&f(0) = 0 \\
&f(1) = 1 \\
&f(n) = 2f(n-1) - f(n-2) \text{ for } n \geq 2
\end{aligned}$$

Determine, with proof, the number, $f(n)$, of rabbits that Cantelli owns in year n. That is, construct a formula for $f(n)$ and prove its correctness.

Let $P(n)$ be "$f(n) = n$". We prove that $P(n)$ is true for all $n \in \mathbb{N}$ by strong induction on n.

Base Cases ($n = 0, n = 1$)**:** $f(0) = 0$ and $f(1) = 1$ by definition.

Inductive Hypothesis: Assume that $P(0) \wedge P(1) \wedge \ldots P(k)$ hold for some arbitrary $k \geq 1$.

Inductive Step: We show $P(k+1)$:

$$\begin{aligned}
f(k+1) &= 2f(k) - f(k-1) && \text{[Definition of } f\text{]} \\
&= 2(k) - (k-1) && \text{[Induction Hypothesis]} \\
&= k + 1 && \text{[Algebra]}
\end{aligned}$$

Conclusion: $P(n)$ is true for all $n \in \mathbb{N}$ by principle of strong induction.

3. **Structural Induction**:
Consider the following definition of a (binary) **Tree**:

Basis Step: $\bullet$ is a **Tree**.

Recursive Step: If L is a **Tree** and R is a **Tree** then $\texttt{Tree}(\bullet, L, R)$ is a **Tree**.

The function leaves returns the number of leaves of a **Tree**. It is defined as follows:

$$\begin{aligned} \mathsf{leaves}(\bullet) &= 1 \\ \mathsf{leaves}(\texttt{Tree}(\bullet, L, R)) &= \mathsf{leaves}(L) + \mathsf{leaves}(R) \end{aligned}$$

Also, recall the definition of size on trees:

$$\begin{aligned} \mathsf{size}(\bullet) &= 1 \\ \mathsf{size}(\texttt{Tree}(\bullet, L, R)) &= 1 + \mathsf{size}(L) + \mathsf{size}(R) \end{aligned}$$

Prove that $\mathsf{leaves}(T) \geq \mathsf{size}(T)/2 + 1/2$ for all Trees T.

For a tree T, let $\texttt{P}(T)$ be $\mathsf{leaves}(T) \geq \mathsf{size}(T)/2 + 1/2$. We prove $\texttt{P}(T)$ for all trees T by structural induction on T.

Base Case (T = $\bullet$): By definition of $\mathsf{leaves}(\bullet)$, $\mathsf{leaves}(\bullet) = 1$ and $\mathsf{size}(\bullet) = 1$. So, $\mathsf{leaves}(\bullet) = 1 \geq 1/2 + 1/2 = \mathsf{size}(\bullet)/2 + 1/2$, so $\texttt{P}(\bullet)$ holds.

Inductive Hypothesis: Suppose $\texttt{P}(L)$ and $\texttt{P}(R)$ hold for some arbitrary trees L, R.

Inductive Step: $\boxed{\text{Goal: Show that } \texttt{P}(\texttt{Tree}(\bullet, L, R)) \text{ holds.}}$

$$\begin{aligned} \mathsf{leaves}(\texttt{Tree}(\bullet, L, R)) &= \mathsf{leaves}(L) + \mathsf{leaves}(R) && \text{[By Definition of } \mathsf{leaves}] \\ &\geq (\mathsf{size}(L)/2 + 1/2) + (\mathsf{size}(R)/2 + 1/2) && \text{[By IH]} \\ &= (1/2 + \mathsf{size}(L)/2 + \mathsf{size}(R)/2) + 1/2 && \text{[By Algebra]} \\ &= \frac{1 + \mathsf{size}(L) + \mathsf{size}(R)}{2} + 1/2 && \text{[By Algebra]} \\ &= \mathsf{size}(T)/2 + 1/2 && \text{[By Definition of } \mathsf{size}] \end{aligned}$$

This proves $\texttt{P}(\texttt{Tree}(\bullet, L, R))$.

Conclusion: Thus, $\texttt{P}(T)$ holds for all trees T by structural induction.

9 Regex and CFG, and DFA

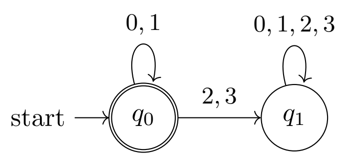
1. Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

$$((01)^*(0 \cup \varepsilon)) \cup ((10)^*(1 \cup \varepsilon))$$

2. All binary strings that contain at least three 1's.

$$\mathbf{S} \to \mathbf{TTT}$$
$$\mathbf{T} \to 0\mathbf{T} \mid \mathbf{T}0 \mid 1\mathbf{T} \mid 1$$

3. All binary strings.



q_0**:** binary strings

q_1**:** strings that contain a character which is not 0 or 1.

Note: For DFA/NFA, you can draw them and then take a picture and use the \includegraphic command to include the picture.