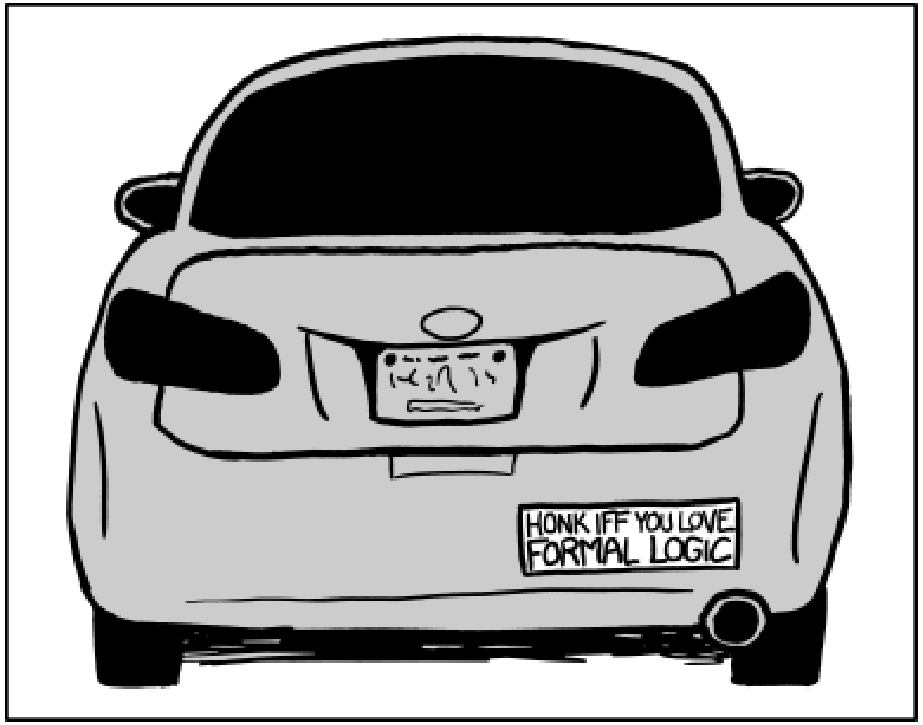
Here Early?

Here for CSE 311?

Welcome! You’re early!

Want a copy of these slides to take notes?

- You can download them from the calendar webpage cs.uw.edu/311



Intro and Propositional Logic

CSE 311: Foundations of Computing I
Lecture 1

Outline

Course Logistics

Start of Propositional Logic

Staff

Instructor: Parker Gustafson (he/him)

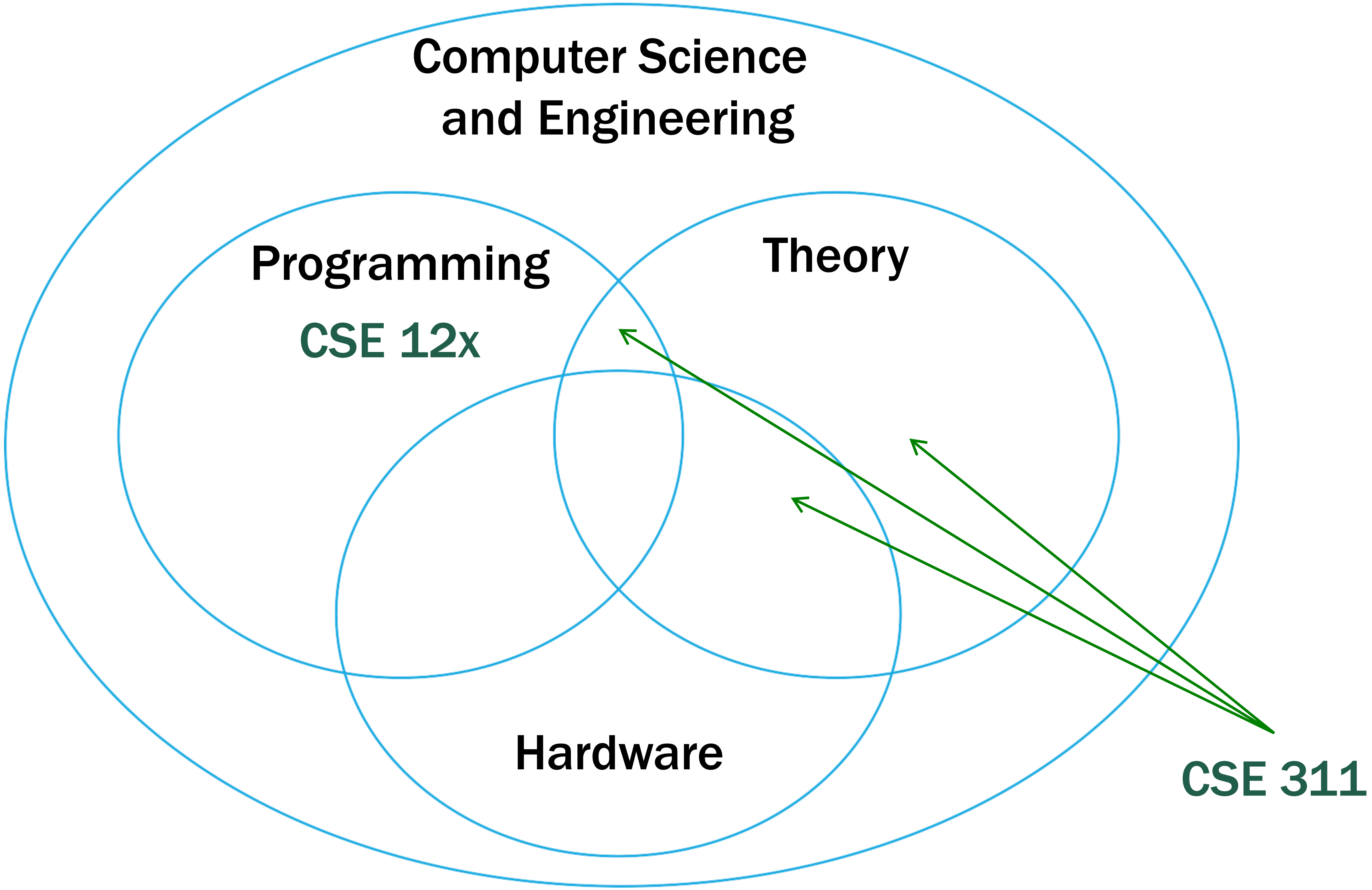
MS student at the Allen School
Undergrad in CS & Math
13-time CSE TA, 2-time 311 TA,
1st time teaching 311 as an instructor

Office: CSE 204
Email: pgusto34@uw.edu

TAs

Alice Zhu
Emma Huang
Evan Wu
Lisa Elkin
Rushil Arun
Zareef Amyeen

Perspective



Course Goals

1. Learn to make & clearly communicate rigorous formal arguments
   - Mathematical Proofs
2. Understand mathematical objects that are widely used in CS
   - Number Theory, Set Theory, Recursively-Defined Functions
3. Explore and analyze models of computation
   - Regular Expressions, Context Free Grammars, Finite-State Machines
4. Develop a toolkit for approaching computational problems
   - Programmer → Computer Scientist

Lectures

- Monday, Wednesday, Friday from 12:00 – 1:00 pm in DEM 102
- Recorded!
  - Lecture recordings can be found on Canvas via the Panopto tab
- Attendance is highly recommended but not required
- Concept Check after every lecture

Concept Checks

Each Lecture will have a “concept check” associated with it.

Goal: Make sure you’ve understood today’s topic before we build on it in the next lecture.

Available on Gradescope at 1:00 pm (after lecture). Due the day of the next lecture at 12:00 pm (start of the lecture).

- You can submit as many times as you want.
- When you get an answer correct the explanation will appear.

Worth a small amount of your grade, and an 80% average for the quarter is all you need for full credit.

- More details in the syllabus.

Quiz Sections

- TA-led sections meet on Thursdays
- Opportunity to practice material and ask questions
- Materials are posted, but sections aren’t recorded
- (sync or async) Participation is required and part of the course grade
- You only need to participate in 7/8 sections for full credit

Exams

- In-class midterm on August 1st
  - There will be an in-person midterm retake opportunity during week 8 (8/11 – 8/15). More details will be announced later.

- In-class final on August 21st and 22nd
  - The final will be taken over two days: part 1 will be taken in your quiz section and part 2 will be taken in lecture.

Homework

- 7 written assignments (no programming)
- Posted Wednesday evenings, due the follow Wednesday at 11:59 pm
- **Typesetting is highly recommended**
  - There are many options: LaTeX (via Overleaf), Parchmynt, Notion
  - Start early and try to finish at least one day before the due date to typeset
- You will be able to revise and resubmit two questions per homework for an improved grade (more details to come)

Late Policy

- You have 6 late days for the quarter
- You can use a late day to turn in an assignment up to 24 hours late with no penalty
- You can use at most 3 late days per assignment
- In the case of extenuating circumstances, please reach out

Collaboration Policy

- Collaboration with others is encouraged
  - Pro Tip: form study groups for this course ASAP
  - Do help other students learn
  - Do not help other students *avoid* learning
- Policy:
  - List all names of those you worked with
  - Don't take away pictures or notes from discussions
  - Write up the final solutions on your own

ChatGPT Policy

- **You are prohibited from using ChatGPT or other LLMs for your assignments.**
- Why?
  - The best way to learn discrete math is by *personally* working with concepts by solving (and often getting stuck on) related problems.
  - ChatGPT is very good at solving these types of problems, so its use would rob you of tons of learning if we permitted its use.
  - Setting yourself up for academic success down the road.
- Permitted uses of ChatGPT for this course
  - As a study tool to learn or review course concepts
  - Help with typesetting
  - Other?

Getting Help

Office Hours

- Both in-person and on Zoom
- Schedule posted on course website
- Begin this Wednesday

Ed Discussion

- Post any content, homework, or logistical questions

Study Groups

- 311 (discrete math) is *different* than programming

Course Grades

Mastery-Based Grading System

- Two components: ESN grading and resubmissions
- Instead of a points-based grading, we will be grading on an ESN scale, where each letter reflects a level of mastery of a course topic or skill.
  - **E** – Excellent (Meeting or exceeding all learning objectives)
  - **S** – Satisfactory (Close to mastery, but there may be some minor mistakes or misunderstandings)
  - **N** – Not Yet (Demonstrated significant knowledge gaps or made major mistakes in work)

Course Grades

- After receiving feedback, we want to give you an opportunity to learn from it and demonstrate improved mastery of the material through resubmissions
  - Everything in this course will have some form of resubmission or reattempt besides the final and section participation
- Your accumulated **E**’s and **S**’s will determine your course grade
  - For example, earning 32 E’s and 10 additional S+ grades will guarantee you a grade of a 2.5
- More grading details are on the course syllabus

Textbook

- There is no official textbook for this course
- Optional: *How to Prove It* by Daniel Velleman, 3rd edition
  - Good option if you're looking for a source of additional practice problems or a second set of explanations for the same material

Course Tools

Course Website

(assignments, calendar, resources)

Gradescope

(submissions, feedback)

Canvas

(lecture recordings)

Ed Discussion

(discussion board)

If something unusual happens

In a 50-person class, a few of you will have something significant happen during the quarter.

- Illness or family emergency or something else.
- We can give some kinds of accommodations (e.g., extra late days) in some cases.
- We can only help if you tell us something is going on
- Either email Parker or create an Ed board post to let us know.

Now is a great time to:

- Ask DRS for accommodations if you think you might need formal ones.
- Start forming study groups!

Symbolic Logic

What is logic and why do we need it?

Symbolic Logic (synonymous with propositional logic) is a language, like English or Java, with its own

words and rules for combining words into sentences (syntax)

ways to assign meaning to words and sentences (semantics)

Symbolic Logic will let us **mechanically** simplify expressions and make *precise*, *concise*, and *unambiguous* arguments.

The new language will let us focus on the (sometimes familiar, sometimes unfamiliar) rules of logic.

Once we have those rules down, we'll be able to apply them "intuitively" in English and won't need the symbolic representation as often

but we'll still go back to it when things get complicated.

Why not use English?

English can be ambiguous or imprecise.

- Turn right here.

  Does “right” mean the direction, or “right now”?

- We saw her duck.

  Does “duck” mean the animal, or the action of ducking down?

- I saw the man with the long telescope.

  Does the man have the long telescope, or did I use the long telescope to see the man?

Propositions: building blocks of logic

Proposition

A statement that has a truth value (i.e. is true or false) and is “well-formed”

Propositions are the basic building blocks in symbolic logic.

Here are two propositions.

All cats are mammals

True, (and a proposition)

All mammals are cats

False, but is well-formed and has a truth value, so still a proposition.

Analogy

In Intro Programming you talked about a variable type that could be either true or false.

You called it a "`Boolean`"

Boolean variables are a useful analogy for propositions.

They aren't identical, but they're very similar.

Are These Propositions?

2 + 2 = 5 This is a proposition. It's okay for propositions to be false.

x + 2 = 5 Not a proposition. Doesn't have a fixed truth value

Akjsdf! Not a proposition because it's gibberish.

Who are you?

This is a question which means it doesn't have a truth value.

There is life on Mars.

This is a proposition. We don't know if it's true or false, but we know it's one of them!

Proposition Notation

We need a way of talking about *arbitrary* propositions…

To make statements easier to read we'll use propositional variables to represent propositions like $p, q, r, s, \ldots$

Lower-case letters are standard.

Usually start with p (for proposition), and avoid t, f, because…

Truth Values:

T for true (note capitalization)

F for false

Analogy

We said propositions were a lot like Booleans…

How did you connect Booleans in code?

`&&`

`||`

`!`

Logical Connectives

And (&&) works exactly like it did in code.

- But with a different symbol ∧

Or (||) works exactly like it did in code.

But with a different symbol ∨

Not (!)works exactly like it did in code.

- But with a different symbol ¬

Some Truth Tables

p	$\neg p$

p	q	$p \land q$

p	q	$p \lor q$

Truth tables are the simplest way to describe how logical connectives operate.

Some Truth Tables

p	**¬*p***
T	F
F	T

p	***q***	***p* ∧ *q***
T	T	T
T	F	F
F	T	F
F	F	F

p	***q***	***p* ∨ *q***
T	T	T
T	F	T
F	T	T
F	F	F

Truth tables are the simplest way to describe how logical connectives operate.

Atomic and Compound Propositions

Definitions:

An atomic proposition is a proposition that can’t be broken down any further.

A compound proposition is a proposition that can be divided into simpler propositions.

logical connective combines propositions into compound propositions.

Compound Proposition Example

It is raining in Seattle and it's June.

Atomic Propositions:

____

____

Logical Connectives:

____

Translation into Logic:

____

Analogy
Boolean variables p, q
Java and connective &&

All Logical Connectives

Name	Logical Symbol	Java Symbol
Not	$\neg p$	!p
And	$p \land q$	p && q
Or	$p \lor q$	p || q
XOR	$p \oplus q$	p ^ q
Implication	$p \rightarrow q$	
Biconditional	$p \leftrightarrow q$	$p == q$

Implication

Another way to connect propositions

If p then q.

"If it is raining, then I have my umbrella."

$p \to q$

Think of an implication as a promise.

p = antecedent, hypothesis

q = consequent, conclusion

Implication

The first two lines should match your intuition.

The last two lines are called "vacuous truth." For now, they're the definition. We'll explain why in a few lectures.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

This is the definition of implication. When you write "if…then…" in a piece of mathematical English, this is how you will be interpreted.

Implication ($p \to q$)

"If it's raining, then I have my umbrella"

It's useful to think of implications as promises. An implication is false exactly when you can ***demonstrate*** *I'm lying.*

	It's raining	It's not raining
I have my umbrella		
I do not have my umbrella		

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Implication ($p \to q$)

"If it's raining, then I have my umbrella"

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

*It's useful to think of implications as promises. An implication is false exactly when you can **demonstrate** I'm lying.*

	It's raining	It's not raining
I have my umbrella	No lie. True	No lie. True
I do not have my umbrella	**LIE! False**	No lie. True

$p \rightarrow q$

$p \rightarrow q$ and $q \rightarrow p$ are different implications!

"If the sun is out, then we have class outside."

"If we have class outside, then the sun is out."

Only the first is useful to you when you see the sun come out.

Only the second is useful if you forgot your umbrella.

$p \rightarrow q$

Implication:

p implies *q*

whenever *p* is true *q* must be true

if *p* then *q*

q if *p*

p is sufficient for *q*

p only if *q*

q is necessary for p

p	q	$p \rightarrow q$
T	**T**	**T**
T	**F**	**F**
F	**T**	**T**
F	**F**	**T**

Implications are super useful, so there are LOTS of translations.
You'll learn these in detail in section.

Logical Connectives

Negation (not)	$\neg p$
Conjunction (and)	$p \wedge q$
Disjunction (or)	$p \vee q$
Exclusive Or (xor)	$p \oplus q$
Implication (if-then)	$p \longrightarrow q$
Biconditional	$p \longleftrightarrow q$

These ideas have been around for so long most have at least two names.

Two more connectives to discuss!

Biconditional: $p \leftrightarrow q$

Think: $(p \rightarrow q) \land (q \rightarrow p)$

p if and only if *q*

p iff q

p is equivalent to *q*

p implies *q* and *q* implies *p*

p is necessary and sufficient for *q*

p	q	$p \leftrightarrow q$

Biconditional: $p \leftrightarrow q$

Think: $(p \rightarrow q) \wedge (q \rightarrow p)$

p if and only if *q*

p iff q

p is equivalent to *q*

p implies *q* and *q* implies *p*

p is necessary and sufficient for *q*

p	q	$p \leftrightarrow q$
T	**T**	**T**
T	**F**	**F**
F	**T**	**F**
F	**F**	**T**

$p \leftrightarrow q$ is the proposition: "p" and "q" have the same truth value.

Exclusive Or

Exactly one of the two is true.

$p \oplus q$

p	q	$p \oplus q$

In English "either p or q" is the most common, but be careful.

Often translated "p or q" where you're just supposed to understand that exclusive or is meant (instead of the normal inclusive or).

Try to say "either…or…" in your own writing.

Exclusive Or

Exactly one of the two is true.

$p \oplus q$

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

In English "either p or q" is the most common, but be careful.

Often translated "p or q" where you're just supposed to understand that exclusive or is meant (instead of the normal inclusive or).

Try to say "either…or…" in your own writing.

Order of Operations

Just like you were taught PEMDAS

e.g. $3 + 2 \cdot 4 = 11$ not 24.

Logic also has order of operations.

Parentheses

Negation

And

Or, exclusive or

Implication

Biconditional

For this class: each line is its own level!
e.g. "and"s have precedence over "or"s

Within a level, apply from left to right.

Some textbooks place And, Or at the same level – it's good practice to use parentheses even if not required.

A More Complicated Statement

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

Is this a proposition?

We'd like to *understand* what this proposition means.

In particular, is it true?

A Compound Proposition

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

We'd like to *understand* what this proposition means.

First find the simplest (**atomic) propositions** and substitute them into the English statement:

p	"Robbie knows the Pythagorean Theorem"
q	"Robbie is a mathematician"
r	"Robbie took geometry"

(p if (q and r)) and (q or (not r))

$$(p \text{ if } (q \wedge r)) \wedge (q \vee (\neg r))$$

A Compound Proposition

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

$$(p \text{ if } (q \wedge r)) \wedge (q \vee (\neg r))$$

p	**"Robbie knows the Pythagorean Theorem"**
q	**"Robbie is a mathematician"**
r	**"Robbie took geometry"**

How did we know where to put the parentheses?

- Subtle English grammar choices (top-level parentheses are independent clauses).
- Context/which parsing will make more sense.
- Conventions

A reading on this is on the webpage!

Back to the Compound Proposition...

"Robbie knows the Pythagorean Theorem if he is a mathematician and took geometry, and he is a mathematician or did not take geometry."

$(p \text{ if } (q \land r)) \land (q \lor (\neg r))$

p	**"Robbie knows the Pythagorean Theorem"**
q	**"Robbie is a mathematician"**
r	**"Robbie took geometry"**

What promise am I making?

$((q \land r) \rightarrow p) \land (q \lor (\neg r))$ $(p \rightarrow (q \land r)) \land (q \lor (\neg r))$

The first one! Being a mathematician and taking geometry goes with the "if." Knowing the Pythagorean Theorem is the consequence.

Analyzing the Sentence with a Truth Table

p	q	r	$\neg r$	$q \vee \neg r$	$q \wedge r$	$(q \wedge r) \rightarrow p$	$((q \wedge r) \rightarrow p) \wedge (q \vee \neg r)$
F	F	F	T	T	F	T	T
F	F	T	F	F	F	T	F
F	T	F	T	T	F	T	T
F	T	T	F	T	T	F	F
T	F	F	T	T	F	T	T
T	F	T	F	F	F	T	F
T	T	F	T	T	F	T	T
T	T	T	F	T	T	T	T

Todo

Tonight:

Make sure you can access the Ed discussion board.

If you can’t, send an email to Parker.

Make sure you can access Gradescope, and do the concept check (due date is Friday, but we encourage you to check by Wednesday).

If you can’t, make a private post on Ed.

Thursday:

Go to section (in-person)

Soon:

Form a study group! Threads to organize on the Ed board.