# Lecture 0 - Course Policy and Introduction

**Vaishnavi Sundararajan**

COL703 - Logic for Computer Science

1. Logistics

2. Actual logic!

# Teaching staff

| | |
|---|---|
| Instructor | Vaishnavi Sundararajan |
| TAs | Karan Agrawal |
| | Shivam Jain |
| | Kushagra Karar |
| | Gokul Sankar |

**Instructor Office Hours:** 1400–1500 Mondays and Thursdays

# Pages to check regularly

Course webpage: For lecture notes and other material:

https://vaishs.github.io/courses/col703-jul24/

Moodle: All course-related announcements will happen on Moodle. You should already be enrolled there.

Piazza: All discussion and Q&A will be done on Piazza. You should be enrolled to the page for COL703 on Piazza.

Send me an email ASAP if you have not been enrolled to Moodle or Piazza.

# Q&A and contacting us

- Do not contact me via means other than email.
- Email is only for queries of an individual nature – anything technical or logistical must be posted on Piazza.
- It is especially not okay to use email to ask about logistics issues that might affect the entire class. Stick to Piazza for those concerns.
- Before posting on Piazza, check if the same question has been asked by someone else before. Do not repeat posts.
- We are but human, and sometimes it might take us up to two days to respond to a query. Be patient, and account for such delays.

# Prerequisites

- Ability to write and understand rigorous formal proofs (especially those involving induction)
- Some minimal programming background; but no prior knowledge of any specific language is required
- Should have taken **at least one** of the following courses (or equivalent)
    - COL106 (Data Structures & Algorithms)
    - COL202 (Discrete Mathematical Structures)
- It is helpful to have done some course where you have had to write rigorous proofs and/or analyze abstract formal objects

# References

- Will mostly follow slides/notes presented in class
- Will upload notes and supplementary material to the course webpage
- External references:
  - Introduction to Formal Logic, by Peter Smith: https://www.logicmatters.net/ifl/
  - Formal Logic: Its Scope and Limits, by Richard C. Jeffrey: https://archive.org/details/formallogicitssc0000jeff_d0z0/ (also available in the IIT library)
  - Hyper-notes for Logic in Computer Science by Prof. S. Arun-Kumar: https://www.cse.iitd.ac.in/~sak/courses/ilcs/2020-21/ilcs.pdf

# Evaluation policy

- Assignments: 20%
  - **Graceful degradation policy**
  - For every day that you use beyond the indicated deadline, you lose 20%.
  - Anything submitted later than 5 days past the deadline will not be evaluated.
- Minor: 20%
  - No re-minor will be conducted.
  - Produce appropriate documentation within one week of the minor if you miss it for a medical reason.
  - In the above case, marks for the major will be scaled up accordingly.
- Major: 40%. Will include everything covered in the course.

# Evaluation policy (Contd.)

- Quizzes: 15–20%.
  - Surprise quizzes conducted in class (How many? Surprise!)
  - Absence in a quiz directly marked 0.
  - No make-up quizzes will be conducted.
- In-class participation, attendance etc: 0–5%.
- Pedagogical experiment: 0–5% **Bonus**
- "A few of my favourite things": 0–5% **Bonus**
- **No phone use in class**: One percent docked per offence, five offences directly gets you down a grade.

# Pedagogical experiment

- Each student gets three (virtual) tokens at the start of the semester
- 0–5% bonus based on fraction of tokens spent **before the midterm**
- Ways to spend a token? Next slide!

# Pedagogical experiment: Ways to spend a token

1. Ask a question (in class, in office hours, or on Piazza)
**which sparks a technical discussion**

Positive and negative examples:

- "I think <concept> should be <this way> instead of <that way>. The reason I think this is..." ✓

# Pedagogical experiment: Ways to spend a token

1. Ask a question (in class, in office hours, or on Piazza) **which sparks a technical discussion**

Positive and negative examples:

- “I think <concept> should be <this way> instead of <that way>. The reason I think this is…” ✓
- “I tried to prove <statement> and got stuck <here>, but I think this technique should work because...” ✓

# Pedagogical experiment: Ways to spend a token

1. Ask a question (in class, in office hours, or on Piazza) **which sparks a technical discussion**

Positive and negative examples:

- "I think <concept> should be <this way> instead of <that way>. The reason I think this is..." ✔
- "I tried to prove <statement> and got stuck <here>, but I think this technique should work because..." ✔
- "How do I solve <question>?" without even starting an attempt: ✗

# Pedagogical experiment: Ways to spend a token

1. Ask a question (in class, in office hours, or on Piazza) **which sparks a technical discussion**

Positive and negative examples:

- "I think <concept> should be <this way> instead of <that way>. The reason I think this is..." ✓
- "I tried to prove <statement> and got stuck <here>, but I think this technique should work because..." ✓
- "How do I solve <question>?" without even starting an attempt: ✗
- "Will the minor cover <concept>?": ✗

# Pedagogical experiment: Ways to spend a token

## 2. Model a real-world example in logic

- Present in class/in office hours

# Pedagogical experiment: Ways to spend a token

## 2. Model a real-world example in logic

- Present in class/in office hours
- Choose a real-world situation (an advertisement, an article, a movie plot, a WhatsApp forward etc.)

# Pedagogical experiment: Ways to spend a token

## 2. Model a real-world example in logic

- Present in class/in office hours
- Choose a real-world situation (an advertisement, an article, a movie plot, a WhatsApp forward etc.)
- Formally model it in some logic

# Pedagogical experiment: Ways to spend a token

## 2. Model a real-world example in logic

- Present in class/in office hours
- Choose a real-world situation (an advertisement, an article, a movie plot, a WhatsApp forward etc.)
- Formally model it in some logic
- Draw at least one interesting conclusion as formal inference

# Pedagogical experiment: Ways to spend a token

## 2. Model a real-world example in logic

- Present in class/in office hours
- Choose a real-world situation (an advertisement, an article, a movie plot, a WhatsApp forward etc.)
- Formally model it in some logic
- Draw at least one interesting conclusion as formal inference
- Could even be (and often is) that some reasonable statement is inconsistent with initial assumptions! Prove this formally.

# Pedagogical experiment: Ways to spend half a token

- Answer a difficult question in class
- Point out some error or ambiguity in the notes
  - Do this even if you are slightly unsure about it being an error!
  - **Worst** case it is an error and you spend half a token
  - **Best** case it is not an error, and you spend **ONE** token if you can sensibly discuss why you thought it was one.
  - Do not misuse this option. If you cannot walk me through why you think it is an error, I will **increment** your remaining token count.
- There will be a handful of surveys over the course of the semester; please fill those out carefully!

# “A few of my favourite things”

- Bonus points for some “extra” items I really liked over the semester.
- Each student can get up to 5% for these.
- Include, but not limited to
  - Best answer(s) to a starred exercise from the notes
  - Best answer(s) to bonus questions in assignments
  - Best question(s) asked that sparked a discussion
  - Best clarification(s) about ambiguity/errors in the notes
  - Best example(s) of real-world situations modelled in logic
- Multiple bests possible under each head

AND NOW FOR SOMETHING
COMPLETELY DIFFERENT.

1 Logistics

2 Actual logic!

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world
- “What constitutes a valid proof?”

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world
- “What constitutes a valid proof?”
- “Is my set of statements internally consistent?”

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world
- “What constitutes a valid proof?”
- “Is my set of statements internally consistent?”
- Valid inference and internal consistency becomes paramount when we **model complex systems**

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world
- “What constitutes a valid proof?”
- “Is my set of statements internally consistent?”
- Valid inference and internal consistency becomes paramount when we **model complex systems**
- Logic allows us to **verify** that systems work correctly...

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world
- “What constitutes a valid proof?”
- “Is my set of statements internally consistent?”
- Valid inference and internal consistency becomes paramount when we **model complex systems**
- Logic allows us to **verify** that systems work correctly...
- ...without testing each possible execution!

# Why study logic as a computer scientist?

- Logic allows us to make sense of our world
- “What constitutes a valid proof?”
- “Is my set of statements internally consistent?”
- Valid inference and internal consistency becomes paramount when we **model complex systems**
- Logic allows us to **verify** that systems work correctly...
- ...without testing each possible execution!
- Important to know when inference is sound!

# The knights and the knaves

- Vera City is populated by knights, who always tell the truth, and knaves who always lie. Alice and Bob are two inhabitants.

# The knights and the knaves

- Vera City is populated by knights, who always tell the truth, and knaves who always lie. Alice and Bob are two inhabitants.
- Alice says, “Bob is a knave”

# The knights and the knaves

- Vera City is populated by knights, who always tell the truth, and knaves who always lie. Alice and Bob are two inhabitants.
- Alice says, “Bob is a knave”
- Bob says, “Neither Alice nor I are knaves”

# The knights and the knaves

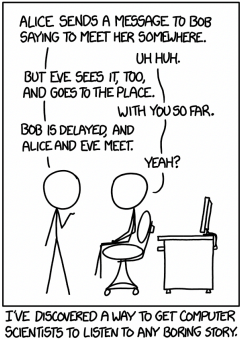
- Vera City is populated by knights, who always tell the truth, and knaves who always lie. Alice and Bob are two inhabitants.
- Alice says, “Bob is a knave”
- Bob says, “Neither Alice nor I are knaves”
- What are Alice and Bob?

# The knights and the knaves

- Vera City is populated by knights, who always tell the truth, and knaves who always lie. Alice and Bob are two inhabitants.
- Alice says, “Bob is a knave”
- Bob says, “Neither Alice nor I are knaves”
- What are Alice and Bob?

# The knights and the knaves

- Vera City is populated by knights, who always tell the truth, and knaves who always lie. Alice and Bob are two inhabitants.
- Alice says, “Bob is a knave”
- Bob says, “Neither Alice nor I are knaves”
- What are Alice and Bob?


ALICE SENDS A MESSAGE TO BOB SAYING TO MEET HER SOMEWHERE.
UH HUH.
BUT EVE SEES IT, TOO, AND GOES TO THE PLACE.
WITH YOU SO FAR.
BOB IS DELAYED, AND ALICE AND EVE MEET.
YEAH?
I'VE DISCOVERED A WAY TO GET COMPUTER SCIENTISTS TO LISTEN TO ANY BORING STORY.

# 2 knights 2 knaves

- Alice says, “Bob and I are both knights or both knaves”

# 2 knights 2 knaves

- Alice says, “Bob and I are both knights or both knaves”
- Bob says, “Alice and I belong to the same category”

# 2 knights 2 knaves

- Alice says, “Bob and I are both knights or both knaves”
- Bob says, “Alice and I belong to the same category”
- What are Alice and Bob?

# How do we solve these?

- Can do case analysis with a truth table
- Only Alice is a knight, only Bob is a knight, both are knights, both are knaves
- Allows us to draw contradictions/impossibilities (using some **valid system of inference**)
- Can rule out all but one (the correct!) case
- In some cases: can do smarter analysis than looking through the entire truth table.

# ~~Jedi~~ Knights and knaves

- Alice says, “Bob is telling the truth.”

# ~~Jedi~~ Knights and knaves

- Alice says, “Bob is telling the truth.”
- Bob says, “Han shot first.”

# ~~Jedi~~ Knights and knaves

- Alice says, “Bob is telling the truth.”
- Bob says, “Han shot first.”
- Carol says, “Greebo shot first.”

# ~~Jedi~~ Knights and knaves

- Alice says, “Bob is telling the truth.”
- Bob says, “Han shot first.”
- Carol says, “Greebo shot first.”
- If only one of Alice, Bob, and Carol is a knight, is it possible to conclusively determine who shot first?

# ~~Jedi~~ Knights and knaves

- Alice says, “Bob is telling the truth.”
- Bob says, “Han shot first.”
- Carol says, “Greebo shot first.”
- If only one of Alice, Bob, and Carol is a knight, is it possible to conclusively determine who shot first?
- Probably not a great idea to do exhaustive case analysis here!

# River crossings

- A farmer with a wolf, a goat, and a cabbage must cross a river by boat.

# River crossings

- A farmer with a wolf, a goat, and a cabbage must cross a river by boat.
- The boat can carry only the farmer and one other entity.

# River crossings

- A farmer with a wolf, a goat, and a cabbage must cross a river by boat.
- The boat can carry only the farmer and one other entity.
- If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.

# River crossings

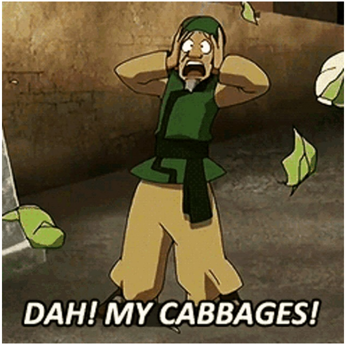
- A farmer with a wolf, a goat, and a cabbage must cross a river by boat.
- The boat can carry only the farmer and one other entity.
- If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.
- How can they cross the river without anyone suffering any damage?

# River crossings

- A farmer with a wolf, a goat, and a cabbage must cross a river by boat.
- The boat can carry only the farmer and one other entity.
- If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.
- How can they cross the river without anyone suffering any damage?

# River crossings

- A farmer with a wolf, a goat, and a cabbage must cross a river by boat.
- The boat can carry only the farmer and one other entity.
- If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.
- How can they cross the river without anyone suffering any damage?



# ⋆ The knights and the knaves: Triplet drift

- Three identical sisters: Alice, Bella, and Carol

# ⋆ The knights and the knaves: Triplet drift

- Three identical sisters: Alice, Bella, and Carol
- Alice and Bella always lie, but Carol always tells the truth

# ⋆ The knights and the knaves: Triplet drift

- Three identical sisters: Alice, Bella, and Carol
- Alice and Bella always lie, but Carol always tells the truth
- You run into one of them unexpectedly, and want to know if it is Bella

# ⋆ The knights and the knaves: Triplet drift

- Three identical sisters: Alice, Bella, and Carol
- Alice and Bella always lie, but Carol always tells the truth
- You run into one of them unexpectedly, and want to know if it is Bella
- You can ask a question of up to three words to find out who it is

# ⋆ The knights and the knaves: Triplet drift

- Three identical sisters: Alice, Bella, and Carol
- Alice and Bella always lie, but Carol always tells the truth
- You run into one of them unexpectedly, and want to know if it is Bella
- You can ask a question of up to three words to find out who it is
- What question would you ask?