CAS CS 131
FALL 2019
Combinatoric Structures

Charalampos (Babis) Tsourakakis
CS 131

September 3rd, 2019


WELCOME TO BU!
WELCOME TO CS 131!

Instructor – Professor

Babis Tsourakakis

tsourakakis.com

Instructors – Teaching Fellows

Tolik Zinovyev
tolik@bu.edu

Arsenii Mustafin
aam@bu.edu

Hassan Saadi
hsaadi13@bu.edu

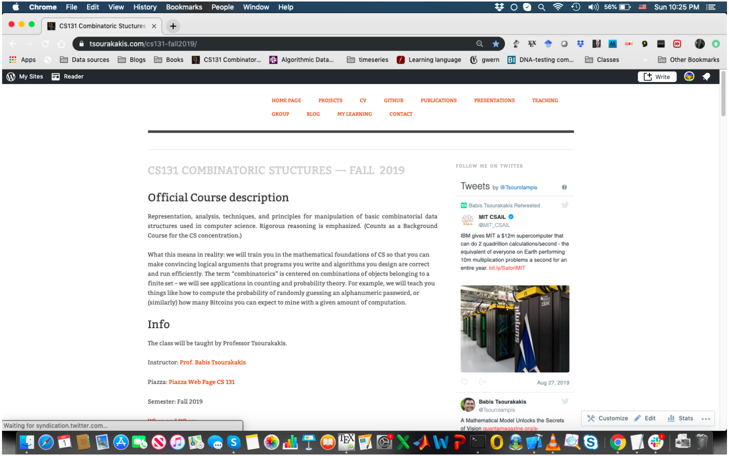
Great team of Teaching Fellows!

Discussion Labs

- On each Wednesday you will participate in a discussion lab.
- The TFs will hand out a set of problems related to the lectures, and you will solve it.
- Make sure you participate, ask your questions, share your ideas.
  - Participation is required
- Please attend the lab you have been assigned to.

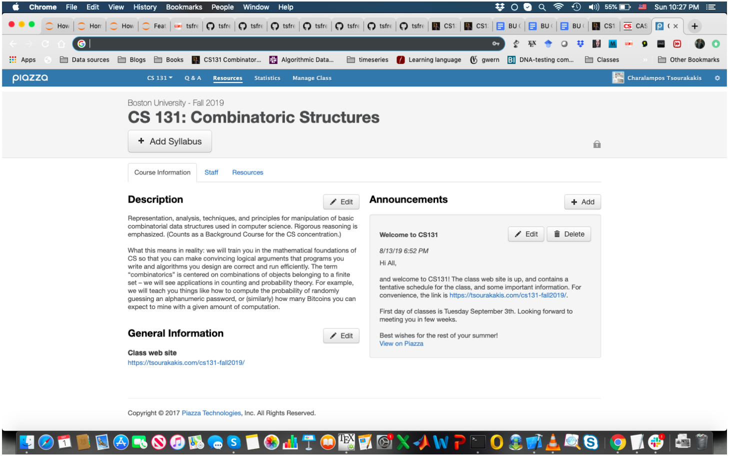
Class Web page

https://tsourakakis.com/cs131-fall2019/



Piazza

https://piazza.com/class/fall2019/cs131



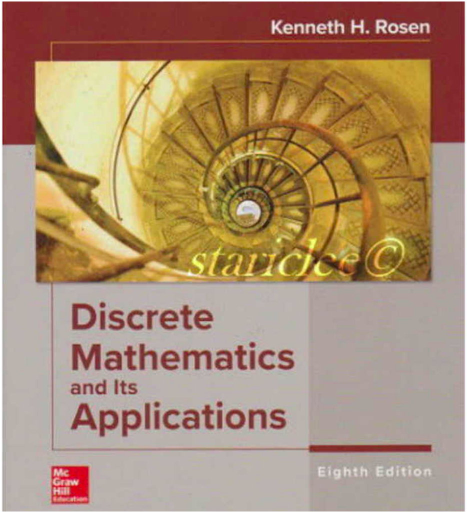
Grading and Attendance

The course grade will break down as follows:

- Problem sets: 25%
- Two midterms: 40% (20+20%)
- Final exam: 30%
- Lab attendance and participation in lab, lecture, Piazza: 5%

Fun class but also hard work!

Textbook


Kenneth H. Rosen
Discrete Mathematics and Its Applications
Eighth Edition
Mc Graw Hill Education

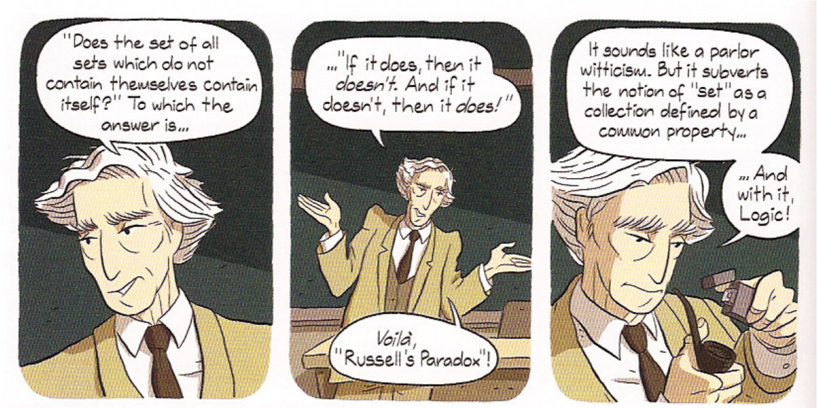
Topics

CS 131 covers **fundamental** topics in Computer Science. The main goal is to introduce you to important proof techniques.

1. Logic and proof
2. Induction
3. Recursive algorithms, and recurrences
4. Number theory
5. Counting
6. Probability
7. Graph theory

You are building foundations in this class!

These are Fun Topics ...

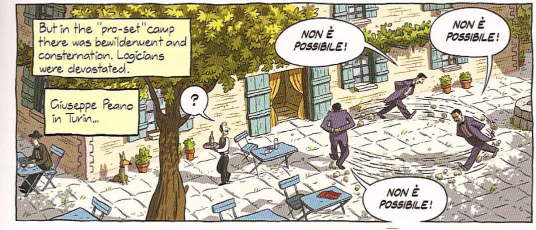


Who is this guy?

Bertrand Russell

The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts.

that have lead to a lot of frustration too!



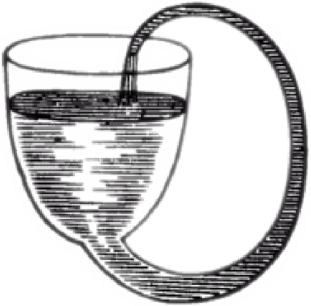
Who are these guys?

Giuseppe Peano and **David Hilbert**

- Hilbert's problems are twenty-three problems in mathematics published by German mathematician David Hilbert in 1900. The problems were all unsolved at the time, and several of them were very influential for 20th-century mathematics.

Paradox using inductive logic



Sorites paradox

- A person with 0 hairs is bald.
- For any number n, if a person with n hairs is bald, then a person with $n + 1$ hairs is also bald.

Therefore, we are all bald!

Academic Conduct

- Academic standards and the code of academic conduct are taken very seriously by our university
- Bottom line:
  - Healthy collaboration for doing homeworks, is fine, and actually encouraged. You can learn from each other.
  - No collaboration during exams.
  - Do not cheat! Besides being a dishonest act, it may also have severe consequences on your academic trajectory.

Office hours

- Prof. Tsourakakis: TR 8.30-10.00

- Arsenii: M 9:00-10.30, F 15.30:17.30
- Hassan: T 17:15 18:15, R 17:15 19:15
- Tolik: W 17:45 18:45, F 13:30 15:30

This week: Space 135 (lounge) at MCS.

Lecture 1

Propositional logic

We start our study of mathematical reasoning with deductive reasoning. Let's see few examples:

1. It will either rain or snow tomorrow. Its too warm for snow. Therefore, it will rain.

2. Either the butler is guilty or the maid is guilty. Either the maid is guilty or the cook is guilty. Therefore, either the butler is guilty or the cook is guilty.

Logical form – Premises and conclusion

1. It will either rain or snow tomorrow. Its too warm for snow. Therefore, it will rain.
2. Either the butler is guilty or the maid is guilty. Either the maid is guilty or the cook is guilty. Therefore, either the butler is guilty or the cook is guilty.

Valid argument: The premises cannot all be true without the conclusion being true as well.

Argument 1 is valid.

Argument 2 is invalid (i.e., were the maid guilty, then both premises are true, but the conclusion is false).

Propositional logic

By replacing statements by letters, we can study the logical structure of the arguments. These are called *propositions*.

Logical form of valid arguments 1.

Premises: P or Q. Not Q

Conclusion: Therefore P.

Logical form of valid invalid argument 2.

Premises: P or Q. Q or S

Conclusion: Therefore P or S.

Propositional logic

- A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

  Which ones are propositions?

  - Boston is the capital of MA (yes)
  - $1 + 1 = 2$ (yes)
  - $x + 1 = 2$ (no)
  - Read this carefully (no)
- Propositions are represented by propositional variables (or sentential variables), e.g., $P =$ it will rain.
- The value of a proposition is either true (T) or false (F)
- Compound propositions are composed by propositions using logical operators (e.g., P **and** Q)

Propositional logic

Symbol	Meaning
$\vee$	or
$\wedge$	and
$\neg$	not

- $P \vee Q$ stands for P **or** Q
- $P \wedge Q$ stands for P **and** Q
- $\neg P$ stands for **not** P

Important remark: Logical **and, or, not** do not correspond to all uses of the words *and, or, not* in English.

- E.g., Tolik and Arsenii are friends (Here the word *and* does not connect two propositions as the logical **and**)
- Or in English can be used both as disjunctive or exclusive. In logic **or** is disjunctive, and we also have **xor** for exclusive or.

Translating English sentences

Let's translate some English sentences into logic.

Either Tolik went to the coffee shop, or Hassan ate noodles.

1. We introduce the necessary propositional variables
   - $P =$Tolik went to the coffee shop
   - $Q =$ Hassan ate noodles.
2. Now we express our compound statement using logical **or**
   - Our compound statement is $P \vee Q$

Translating English sentences

Either Bill is at work and Jane isnt, or Jane is at work and Bill isnt.

1. We introduce the necessary propositional variables
   - $B =$Bill is at work
   - $J =$Jane is at work

2. Now we express our compound statement step by step
   - *Either Bill is at work and Jane isnt* translates to $B \wedge \neg J$
   - *Either Bill is not at work and Jane is* translates to $\neg B \wedge J$
   - The whole proposition is therefore

$$(B \wedge \neg J) \vee (\neg B \wedge J).$$

Precedence of logical operators

Consider the proposition $\neg p \wedge q$. How should we interpret it?

- As $\neg(p \wedge q)$...
- or $(\neg p) \wedge q$?

Precedence of operators is as follows.

1. $\neg$
2. $\wedge$
3. $\vee$

Therefore, the correct way to interpret it as $(\neg p) \wedge q$.

Remark: For now, we will be using parentheses, but you should get familiar with the precedence of these three operators (more to follow).

Translating Logic into English sentences

1. $((\neg S) \wedge L) \vee S$ where S stands "John is smart", and L "John is lazy".

   Either John is smart, or John lazy and not smart

2. $(\neg(S \wedge L) \vee S$

   Either John is smart, or he is not both smart and lazy.

Truth tables

- We wish to evaluate the true or falsity of a compound proposition.
- The first way we learn about doing this is through truth tables.

TABLE 1 The Truth Table for the Negation of a Proposition.

p	$\neg p$
T	F
F	T

Truth tables

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Truth tables

Practice: Make a truth table for the formula $\neg(P \wedge Q) \vee \neg R$.

P	Q	R	$P \wedge Q$	$\neg(P \wedge Q)$	$\neg R$	$\neg(P \wedge Q) \vee \neg R$
F	F	F	F	T	T	T
F	F	T	F	T	F	T
F	T	F	F	T	T	T
F	T	T	F	T	F	T
T	F	F	F	T	T	T
T	F	T	F	T	F	T
T	T	F	T	F	T	T
T	T	T	T	F	F	F

Exclusive OR ($\oplus$)

Definition: The exclusive or of p and q, denoted by $p \oplus q$ is the proposition that is true when exactly one of p and q is true and is false otherwise.

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

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

Exclusive OR ($\oplus$)

Example: I will use all my savings to travel to Europe or to buy an electric car.

- $P =$ I will use all my savings to travel to Europe
- $Q =$ I will use all my savings to buy an electric car.

Our proposition *I will use all my savings to travel to Europe or to buy an electric car* can be expressed as $P \oplus Q$

Conditional statement (aka implication)

Definition: Let p and q be propositions. The conditional statement $p \to q$ is the proposition

if p, then q.

p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).

TABLE 5 The Truth Table for the Conditional Statement $p \to q$.

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

Conditional statement (aka implication)

There are many ways to express a conditional statement in English.

"if p, then q"
"if p, q"
"p is sufficient for q"
"q if p"
"q when p"
"a necessary condition for p is q"
"q unless $\neg p$"

"p implies q"
"p only if q"
"a sufficient condition for q is p"
"q whenever p"
"q is necessary for p"
"q follows from p"
"q provided that p"

Conditional statement (aka implication)

Example: Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job." Express the statement $p \rightarrow q$ as a statement in English.

Conditional statement (aka implication)

Example: Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job." Express the statement $p \to q$ as a statement in English.

- If Maria learns discrete mathematics, then she will find a good job.
- Maria will find a good job when she learns discrete mathematics.
- Maria will find a good job unless she does not learn discrete mathematics.

Converse, contrapositive, and inverse

Consider the implication $p \to q$.

1. Converse of $p \to q$: $q \to p$
2. Contrapositive of $p \to q$: $\neg q \to \neg p$
3. Inverse of $p \to q$: $\neg p \to \neg q$

$p \rightarrow q$ and $\neg q \rightarrow \neg p$

- Implication $p \rightarrow q$: If John cashed the check I wrote then my bank account is overdrawn

- Contrapositive $\neg q \rightarrow \neg p$: If my bank account isnt overdrawn then John hasnt cashed the check I wrote

Contrapositive law

$$P \rightarrow Q \text{ is equivalent to } \neg Q \rightarrow \neg P.$$

Exercise: Truth tables on blackboard!

Converse, contrapositive, and inverse

Proposition: The home team wins whenever it is raining. Express the following in English.

- Converse $q \to p$: ??
- Inverse $\neg p \to \neg q$: ??
- Contrapositive $\neg q \to \neg p$: ??

Converse, contrapositive, and inverse

Proposition: The home team wins whenever it is raining. Express the following in English.

- Converse $q \rightarrow p$: If the home team wins, then it is raining.
- Inverse $\neg p \rightarrow \neg q$: If it is not raining, then the home team does not win.
- Contrapositive $\neg q \rightarrow \neg p$: If the home team does not win, then it is not raining.

Biconditional statement (bi-implications)

Example: Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition

p if and only if q

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

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

Example: $\underbrace{\text{You can take the flight}}_{P}\ \underbrace{\text{if and only if}}_{\leftrightarrow}\ \underbrace{\text{you buy a ticket}}_{Q}.$

Precedence of logical operators

Precedence of operators is as follows.

1. $\neg$
2. $\wedge$
3. $\vee$
4. $\rightarrow$
5. $\leftrightarrow$

Example: $\neg p \vee q \rightarrow r$ should be interpreted as $((\neg p) \vee q) \rightarrow r$.

Bits (binary digits)

- Computers represent information using bits. A bit is a symbol with two possible values, namely, 0 (zero) and 1 (one).
- Our logical operations can be expressed using 0/1s in addition to T/F.

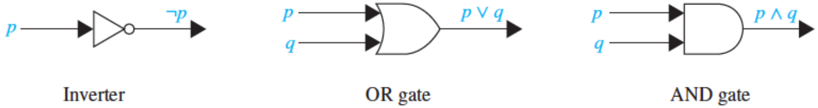
Truth Value	*Bit*
T	1
F	0

TABLE 9 **Table for the Bit Operators** ***OR*****,** ***AND*****, and** ***XOR*****.**

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Logic circuits – Gates

- Single output logic circuits receive binary input signals and output 0/1.
- Their basic components are the following three gates:



Logic circuits – Example

- Using the basic gates we can produce combinatorial circuits for compound propositions.
- For now, parantheses will help you, but you should start getting familiar with the precedence of logical operators.