LSCI 109/209: Formal Languages

Connor Mayer

Fall 2022

E-mail: `cjmayer@uci.edu`
Student hours: W 10am–11am
Student hours location: SSPB 2211

Class Hours: MWF 9am–9:50am
Clasroom: SSL 115

Prerequisites: LSCI 3
Some basic programming knowledge
(e.g., ICS 31/ICS 32A/EECS 12 or similar)

Website: **https://canvas.eee.uci.edu/courses/49169**

Course Description

Computational linguistics is a large, multifaceted and rapidly expanding field. There are many different kinds of work that might be classified as "computational linguistics", differing in

- goals (e.g., build a useful tool, test a linguistic theory), and
- empirical domains (e.g., sounds, words, sentences),

but there are certain core analytical concepts, tools, and techniques that frequently appear throughout. In this course, we will consider these foundational concepts through the lens of Formal Language Theory (FLT), which is concerned with the mathematical properties and computational complexity of formal languages and the formal grammars that describe them. Broadly speaking, the relevant foundational concepts concern the computational nature and properties of the systems of rules and restrictions that govern the patterns we see in natural language. This course aims to highlight these common ideas, starting with their simplest instantiations and gradually building up towards the more complex cases. The simple cases will probably closely parallel certain things you may have learned about in mathematics or computer science courses, and the more complex cases will hopefully look similar to things you have learned about in linguistics courses. An important goal is to highlight the connections between these areas.

As a preview, some of the biggest ideas that will come up repeatedly are:

- recursive generation of infinitely many expressions by a finite machine,
- interchangeability/intersubstitutability of subexpressions within larger expressions, and
- the relationship between discrete structures and probabilistic models.

Course schedule

This schedule may be adjusted based on the progress of class discussions and student interests.

Week	*Date*	*Topic*	*Comment*	*Weekly readings*
0	9/23	*Review of Python basics*		
1	9/26	*Recursion and induction*	Quizlet 1 due	HMU Ch. 1
	9/28	*Formal language theory*		PMW Ch. 16
	9/30	*Finite state automata*	Homework 1 due	
2	10/3	*Recursive generation using FSAs*	Quizlet 2 due	HMU Ch. 2&3
	10/5	*Relating regular expressions and ϵ-FSAs*		PMW Ch. 17
	10/7	*Intersecting FSAs*		
3	10/10	*Forward and backward sets*	Quizlet 3 due	HMU Ch. 4
	10/12	*Are natural languages regular?*		
	10/14	*Probabilistic FSAs*	Homework 2 due	
4	10/17	*Semirings*	Quizlet 4 due	PMW Ch. 9&10
	10/19	*More semirings*		
	10/21	*Context-free grammars*		
5	10/24	*More CFGs*	Quizlet 5 due	HMU Ch. 5
	10/26	*Inside and backward values*		PMW Ch. 18
	10/28	*Probabilistic CFGs*	Homework 3 due	
6	10/31	*More PCFGs; relating FSAs and CFGs*	Quizlet 6 due	HMU Ch. 6&7
	11/2	*Pushdown automata*		
	11/4	*Bottom-up parsing with CFGs*		
7	11/7	*Top-down and left-corner parsing*	Quizlet 7 due	Resnik (1992)
	11/9	*Comparing parsing algorithms*		
	11/11	**No class: Veteran's Day**	Homework 4 due	
8	11/14	*Tree languages*	Quizlet 8 due	Engelfriet (1975)
	11/16	*Tree grammars*		
	11/18	*Applications of tree grammars*		
9	11/21	*Subregular languages*	Quizlet 9 due	Heinz (2018)
	11/23	*Tier-based strictly local languages*		
	11/25	**No class: Thanksgiving**	Homework 5 due	
10	11/28	*Subregular relations*	Quizlet 10 due	PMW Ch. 20&21
	11/30	*Context-sensitive grammars*		Clark (2015)
	12/2	*Multiple context-free grammars*		
11	12/7	**Final exam (8:00am – 10:00am)**		
	12/9	**Final paper due**		

Learning outcomes

- Knowledge outcomes:
  - Students should know the various formal grammars/languages in the Chomsky hierarchy (finite-state, context-free, context-sensitive, sub-regular, etc.), and know the limitations of each.
  - Students should understand the relationship between grammatical structure, recursion and dynamic programming.
- Skills outcomes:
  - Students should be able to read descriptions of grammatical systems in traditional mathematical notation and write corresponding programs using recursion and/or dynamic programming.
- Attitudes and values outcomes:
  - Students should come to appreciate the kind of understanding of the human mind that can come from trying to express its workings in a completely formalized system.
- Behavioral outcomes:
  - By combining the skills and knowledge outcomes above, students should be able to construct programmed implementations of simple grammatical models.

Course Materials

Readings

There is no required textbook for this course. All materials (assignments, notes, readings) will be distributed through the course website on Canvas. Course readings are optional (all the material you need will be contained in the handouts), but you may find them useful.

For background on Python, I recommend the first five (or so) chapters of *Automate the Boring Stuff with Python* by Al Sweigart (`https://automatetheboringstuff.com/`).

Hopcroft, Motwani & Ullman's *Automata Theory: Languages and Computation* is an excellent introduction to many of the formalisms we will explore in this course, and Partee, Meuleun & Wall's *Mathematical Methods in Linguistics* shows how these can be applied to linguistic problems. There are PDFs of these textbooks on Canvas, and the schedule indicates corresponding readings.

There are two well-known textbooks that are often used for introductory computational linguistics courses: Jurafsky & Martin's *Speech and Language Processing* (Prentice Hall) and Manning & Schütze's *Foundations of Statistical Natural Language Processing* (MIT Press). Both take an approach that is more oriented towards specific applications than we will be taking in this course, but you may find them interesting and useful.

Software

Most of the homework exercises will involve writing or modifying small programs. The programming language we will use is Python 3. You will need access to a computer with a text editor and Python 3 installed (**https://www.python.org/downloads/**).

Requirements and grading

There are two grading options available for students. Graduate students are required to take the project-and-exams option, while undergraduates can choose either one.

Exam-only

Component	**Proportion of grade**
Weekly quizlets	10%
Five homework exercises	70%
Final exam	20%

Project-and-exams

Component	**Proportion of grade**
Weekly quizlets	10%
Five homework exercises	70 %
Final exam	10%
Final mini-project	10%

Quizlets

There will be ten short take-home quizzes (quizlets) which will be posted on Canvas on Friday and will be due on Cavnas at 11:59 pm on the following Monday. The aim of these is to keep you thinking about the course material, check your understanding of something from the previous week, and sometimes to get the ball rolling for the new day's content. **No late quizlets will be accepted.**

Homework assignments

Assignments will be posted to Canvas at least a week prior to their due dates. **Homework can be turned up to 7 days late**. 10% of your score will be deducted for each 24 hours of lateness (rounded up). For example, if an assignment is worth 100 points, you turn it in two days late, and earn an 80 before lateness is taken into account, your score will be $(1 - 0.2) * 80 = 64$.
Students are permitted (encouraged, even!) to collaborate on homework assignments. However:

- Each person must hand in their own assignment that is reflective of their own understanding (no direct copies or jointly authored assignments are allowed).

- You must list at the top of your assignment all of the people you've collaborated with.

Your discussions should abide by (both the letter and spirit of) the "whiteboard policy": you may work together on a whiteboard and discuss things for as long as you wish and in as much detail as you wish, but then you must erase the whiteboard and not take any written notes away from this discussion. The idea is that being able to write up your solution individually establishes that you understand what you submit.

Final exam

There will be a cumulative final exam. The exam is scheduled to be held from 8:00am – 10:00am on Wednesday December 7th.

Final mini-project

Graduate students are required to do a final mini-project. Undergraduate students have the option to do so. It will be due at the end of exam week.

The requirements for the project are fairly open-ended. Some possibilities include:

- An analysis of an interesting linguistic phenomenon using formal language theory.
- A software implementation of an algorithm related to topics we discuss in class (this must go beyond what's covered in the homeworks and include a brief write-up).
- A pilot experiment that frames its hypothesis in terms of formal complexity.

If you are interested in doing a final project, **please email me by the end of Week 7 with a brief description of your proposed project.**

Attendance of lectures

You will not be graded on lecture attendance, but it is *crucial* to your mastery of the course material and your success in the class. This is a difficult course, and if you do not attend lectures and stay on top of the material, **you will fall behind**. That being said, I will record lectures using Zoom and post them to Canvas. This is not intended to replace in-person attendance, but will give you greater flexibility if external factors prevent you from attending class. With the exception of the final exam, all quizlets and homeworks will be distributed and submitted on Canvas.

Numeric and letter grades

Letter grades are calculated from numeric grades as follows:

Numeric grade	Letter grade
$\geq 90\%$	A
$\geq 80\%$	B
$\geq 70\%$	C
$\geq 60\%$	D
$< 60\%$	F

I reserve the right to scale final grades if I think it is necessary. I will only scale grades up: that is, your final grade can only *improve* as the result of scaling.

Getting help

- The first place you should seek help is using the discussion board on Canvas. If you have a question, it's likely that someone else has the same question. Posting on the discussion board allows everyone to see the answer. I also strongly encourage you to try to answer your peers' questions on the discussion board. This gives you valuable practice engaging with the course material, utilizing online resources, and synthesizing information, all of which will serve you well down the road.

- The second place you should come for help is my student hours. Please feel free to drop by as frequently as you like, even if you don't have any specific questions and you just want to work on an exercise or chat.

- If neither the discussion board or student hours are viable, you can email me with questions or concerns. I will reply to you within 24 hours.

- In certain circumstances I may be willing to arrange a meeting with you outside of normal class times and office hours. For the sake of my schedule (and yours!), please consider this a last resort, and do your best to seek help using the resources in the previous three points.

Academic integrity

All students are expected to adhere to the UCI Academic Dishonesty Policies (for more information, please visit https://aisc.uci.edu/students/academic-integrity/index.php).

Disability

Any student requesting academic accommodations based on a disability is required to apply with Disability Service Center at UCI. For more information, please visit http://disability.uci.edu/.

Acknowledgements

Much of the material used in the lectures and assignments in this course comes from previous iterations designed by Tim Hunter and Ethan Poole at UCLA. Charles Torres contributed to lecture handouts and content.