# Computational Linguistics 2

Spring 2024

| Course | Info |
|---|---|
| Course# | lin637 |
| Time | T/R 10:00-11:20am |
| Location | CompLab (SBS N-250) |
| Website | https://stonybrook-lin539.github.io/main/ |
| | |
| Instructor | Thomas Graf |
| Email | [coursenumber]@thomasgraf.net |
| Office hours | T 12:00–1:00 |
| | W 12:00–1:00 |
| | T 12:00–1:00 |
| Office | SBS N-249 |

## 1 Course outline

### 1.1 Bulletin description

An introduction to the theoretical foundation of computational linguistics. The course emphasizes the importance of algorithms, algebra, logic, and formal language theory in the development of new tools and software applications. Empirical phenomena in phonology and syntax are sampled from a variety of languages to motivate and illustrate the use of concepts such as strictly local string languages, tree transducers, and semirings. Students will develop familiarity with the literature and tools of the field.

### 1.2 Full description

This course serves a specific purpose in our program: it acts as the bridge from introductory courses in linguistics (Syntax 1, Phonology 1, Phonetics) and computational methods (Statistics, Mathematical Methods in Linguistics, Computational Linguistics 1) to advanced courses and seminars in computational/mathematical linguistics. In contrast to the NLP courses offered by the department of computer science, this course focuses on studying the properties of natural language from a computationally informed perspective. The question is not how computers can solve language-related tasks, but how language can be conceptualized as a computational problem. This emphasis is also reflected in the selection of topics for this course.

- **What this course is not about**
  - Computer-assisted research methods in linguistics
  - Software development for natural language tasks
- **What is not covered but benefits from what is covered**
  - Speech recognition
  - OCR
  - Text generation
  - Parsing
  - Semantic analysis
  - Machine translation
  - Neural networks
- **List of topics**
  - *Phonology and morphology*
    - ⋆ The role of formalization
    - ⋆ String languages
    - ⋆ Subregular hierarchy
    - ⋆ Regular languages
    - ⋆ Generative capacity of phonology
    - ⋆ String transductions
    - ⋆ 2-level morphology
    - ⋆ Equivalence of SPE and OT
    - ⋆ Weighted automata and transducers
  - *Syntax*
    - ⋆ Tree languages
    - ⋆ Syntax is more complex than phonology
    - ⋆ Mildly context-sensitive formalisms (TAG, MGs)
    - ⋆ Tree transductions
    - ⋆ Regular representations of MCS formalisms
    - ⋆ Reinterpreting the T-model

A rough outline of the course progression is given in the table below. Many of the topics we cover draw from very specialized areas of formal language theory that even most mathematicians and computer scientists do not know about, e.g. the correspondence between finite-state machines and monadic second-order logic, or the logical characterization of tree transductions. So this course might be of value to you even if you do not particularly care about natural language. Make no mistake, though, we'll talk a lot about language and linguistics — this is not a math class.

| *Wk* | *Formal* | *Linguistics* |
|---|---|---|
| 1 | What is computation? | Marr's Three Levels |
| 2 | Formalizing phonology | Why formalize? |

| *Wk* | *Formal* | *Linguistics* |
|---|---|---|
| 3 | Strictly local languages | Local dependencies |
| 4 | Subregular hierarchy | How powerful is phonology? |
| 5 | Regular languages | Abstractness |
| 6 | String transductions | SPE-OT equivalence |
| 7 | Two-level morphology | Null morphemes |
| 8 | (Spring Break) | |
| 9 | Weak Generative Capacity | Phonology $<$ Syntax |
| 10 | Tree languages | Headedness, feature percolation |
| 11 | Local tree languages | GPSG |
| 12 | Recognizable tree languages | GB |
| 13 | TAG and MGs | Minimalist syntax |
| 14 | Tree transductions | Reinterpreting the T-model |
| 15 | Unification via first-order logic | Strictly Derivational Minimalism |

### 1.3 Prerequisites

The only official prerequisite is Computational Linguistics 1 (Lin 537) or comparable programming skills in Python. Python will be used to illustrate formal concepts, and some of the homework assignments will require you to implement an algorithm or procedure in Python. Prior experience with git and markdown is useful for the homework assignments but not required.

It is also helpful to have some basic familiarity with linguistics (phonemes, phrase structure rules, syntactic trees) and mathematics (sets, functions, relations, and first-order logic as covered in Semantics 1, for instance).

## 2 Teaching goals

- **Practical skills**
  - conceptualize a problem in mathematical terms
  - optimize your programs through the use of adequate algorithms and data structures (dynamic programming techniques, hash tables, etc.)
  - a more abstract and theoretically informed perspective on current tools and techniques in NLP
  - an understanding for how linguistic insights can be invoked to simplify NLP tasks
- **Research skills**
  - assess linguistic phenomena from a computational perspective
  - evaluate linguists' claims about computational efficiency
  - basic overview of current research in theoretical computational linguistics
  - use computational concepts to identify new empirical generalizations

  - bring linguistic data to bear on computational claims
  - mathematically informed understanding of linguistic theories

## 3 Grading

This course cannot be taken for variable credit. Students who want to take the course for less than 3 credits should contact me as soon as possible.

Student grades are determined by the following components:

1. **Attendance & class participation**
   - Everybody is expected to attend every class. Attendance here means both physical and mental presence.
   - You are expected to be actively engaged, ask questions, and participate in discussions.
   - Simply parking your body in the CompLab at the scheduled time does not constitute attendance.
2. **Class summary**
   - At the beginning of each lecture, a randomly chosen student will be asked to give a 5 minute summary of the last lecture.
   - The summary may require you to present key ideas at the board.
3. **Assignments**
   - approximately one every other week (whenever we hit a natural capstone)
   - exercises, programming assignments, or critical evaluations of assigned readings
   - Homework submission and grading is done in person, or via GitHub for programming assignments.
   - No late hand-ins!
   - Solutions will be discussed in class.
   - Collaboration on homework problems is encouraged as long as you write up the solutions by yourself, using your own words, examples, notation, and code.
   - Under no circumstances should you copy code from the internet. That's a severe case of academic dishonesty.
4. **Readings**
   - we may have occasional readings, but this is not the focus of the course
   - Reading comprehension may be tested as part of the homework assignments.
5. **Lecture Note Feedback**
   - I plan to publish the lecture notes as an open-access textbook with *Language Science Press*.
   - Every week, you should file issues on GitHub for the relevant units, where you spot typos, suggest exercises, pictures, examples, etc.
   - This is a collaborative enterprise: comment on other student's suggestions if you (dis)approve, expand their ideas, and so on.

## 4 Online component

This class uses some online tools to facilitate homework submission and dynamic lecture evaluation.

### 4.1 Homework submission

- *How it works:* Each homework assignments is in its own private GitHub repository. You can fork this repo and upload your own code. In order to submit a homework you upload your solution to your fork and issue a pull request. After the due date, I'll upload my solution to the repository.
- *Why we do it:* This setup mimics the modern workflow in collaborative development projects. Git is one of the best-known version control systems, and GitHub is the biggest online service for hosting git repositories. Familiarity with version control systems is an essential job requirement for computational linguists.

  Git skills are also very helpful for academic work. See this discussion on Stackoverflow for some ideas how git can be used in conjunction with Latex: http://stackoverflow.com/questions/6188780/git-latex-workflow
- *What you'll need:* A GitHub account (the free tier is enough) and a way of uploading your code to a GitHub repository. Linux users can install git via the command line. Mac users can use `homebrew` or a GUI app. Windows uses can install `posh-git` to use git in the Powershell, or install a GUI app. Frankly, though, I would discourage you from using a GUI for git, they tend to make things harder to understand than the command line.

### 4.2 Lecture Notes

- *How it works:* The lecture notes are made available online Monday and Wednesday before 3pm. You can look at them on your laptop/tablet or make a hardcopy before class. I will not bring handouts to class.
- *Why we do it:* Mostly because I just don't like paper. Also keep in mind that you can fork the lecture notes repository and include your own notes directly in the Latex source files. Then you can compile a version of the lecture notes with your own notes already included.

## 5 Miscellaneous

### 5.1 Homework feedback and discussion

- We will discuss solutions in class. You will be expected to be ready to present and justify your solution.

### 5.2 Class announcements

- Class announcements will be distributed by email.

## 6 Policies

### 6.1 Contacting me

- Emails should be sent to [coursenumber]@thomasgraf.net. Disregarding this policy means late replies and is a sure-fire way to get on my bad side.
- Reply time < 24h in simple cases, possibly more if meddling with bureaucracy is involved.
- If you want to come to my office hours and anticipate a longer meeting, please email me so that we can set apart enough time and avoid collisions with other students.

### 6.2 Student Accessibility Support Center Statement

If you have a physical, psychological, medical, or learning disability that may impact your course work, please contact the Student Accessibility Support Center, Stony Brook Union Suite 107, (631) 632-6748, or at sasc@stonybrook.edu. They will determine with you what accommodations are necessary and appropriate. All information and documentation is confidential.

Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and the Student Accessibility Support Center. For procedures and information go to the following website: https://ehs.stonybrook.edu//programs/fire-safety/emergency-evacuation/evacuation-guide-disabilities and search Fire Safety and Evacuation and Disabilities.

### 6.3 Academic Integrity Statement

Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty is required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty please refer to the academic judiciary website at http://www.stonybrook.edu/commcms/academic_integrity/index.html

### 6.4 Critical Incident Management

Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Student Conduct and Community Standards any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Until/unless the latest COVID guidance is explicitly amended by SBU, during Fall 2021 “disruptive behavior” will include refusal to wear a mask during classes.

For the latest COVID guidance, please refer to: https://www.stonybrook.edu/commcms/strongertogether/latest.php