CS 860: Formal Languages and Number Theory
Course Project
Winter Term 2020

January 21, 2020

Handed out Thursday, January 23 2020. Choose project by Thursday, February 6 2020. Present project in class in the last three weeks; hand in written portion by last day of class.

CS 860 students must complete a term project that involves either (i) reading research papers in the areas of the course and presenting the results in class or (ii) doing independent research in the area and presenting results in class or (iii) writing or re-writing Wikipedia pages on various topics.

This handout addresses some possibilities for (i) and (iii). For (ii), try one of the problems marked as “open” in the course textbook, or on the open problems page of the course website.

You are welcome to choose a problem not on this list, but please check with me first.

Part of the goal of this project is for you to learn something about how to do bibliographic search in the area.

How much information is enough? If your problem is well-studied and you only bother to look at one source (one paper or book) then you haven’t done enough. (Even if the paper claims to have the ultimate solution, you should explore that for yourself.) On the other hand, if you collect a list of more than 10 papers, then you should narrow the field somehow, either by restricting the scope of the problem or the type of solution, or by focusing on the most recent or the most relevant work.

The second step of your exploration is then to make sense of the available results, judging which are most useful. Then you will write a brief report (5 – 15 pages) and present your results in a talk of 15-30 minutes in class.

For both the oral presentation and written project, you will be marked on clarity, presentation, depth, and understanding displayed.

You will be marked on depth (20%), understanding (30%), clarity (30%), and the presentation itself (20%). You might want to read Ian Parberry, “How to present a paper in Theoretical Computer Science”, available at https://www.csee.umbc.edu/csee/research/cra/etw98/speaker.pdf.
On February 6 2020, you should hand in a 1-page sheet in class with your name, the title of your project, a brief description of what you plan to do, and at least 3 relevant references to the literature you plan to read.

1 Writing hints for the written version of the project

Here are the most common errors students make when writing a mathematics paper for the first time:

– never start clauses or sentences with notation.
– punctuate equations as if they were ordinary English sentences.
– Both “i.e.” and “e.g.” need a comma after them, always.
– All mathematics in latex must be in math mode. This includes mathematics in the titles of papers in the bibliography. If you don’t observe this basic rule, then variables will appear in the wrong font.
– Use the built-in latex commands for things like max, min, etc.
– Multi-letter mathematics functions (in analogy with sin, cos, etc.) should appear in the Roman font. Use the latex \texttt{DeclareMathOperator} command if the particular function you need has not yet been defined.
– Avoid splitting equations over lines.
– Read about “run-on sentences” in Wikipedia to learn how to recognize and avoid this common stylistic error.
– Use the correct left quotes in latex.
– Give complete bibliographic references including journal name, volume, year, and page numbers, etc.
– Run your paper through a spell-checker before turning it in.
– If your English is not very strong, ask a friend to proofread your report before you turn it in.

When simultaneously citing multiple references, use syntax similar to \cite{ref1,ref2,ref3} to combine all references in a single pair of brackets; do not write \cite{ref1}, \cite{ref2}, \cite{ref3}.

When citing a theorem or page number in another work, say \cite[p. 123]{ref1} or something similar. Note in particular the backslash and space after the dot. This is needed because LaTeX assumes that a dot following a lowercase letter indicates the end of a sentence, and hence inserts extra space.

2 Some suggestions for doing a reading project

1. Many people have proposed the use of continued fractions for computer arithmetic. See, for example, J. E. Vuillemin, “Exact real computer arithmetic with continued fractions”, IEEE Trans. Computers 39 (1990), 1087–1105; P. Kornerup and D. W. Matula, “LCF: A

2. Read more about words without repetitions. The following book


has an extensive bibliography in section E 21. Also see the course textbook.


7. Deciding periodicity and ultimate periodicity. Here you are somehow given a description of an infinite word and you must decide if that infinite word is periodic or ultimately periodic. See, for example,


6. J. Honkala, A decision method for the recognizability of sets defined by number systems, *RAIRO Info. Théorique* **20** (1986), 395–403. For $k$-automatic sequences this can be done with *Walnut*, so the interesting case is non-$k$-automatic.

8. Exotic numeration systems. There have been many unusual number systems proposed. For example, see V. S. Dimitrov, G. A. Jullien, and W. C. Miller, Theory and applications for a double-base number system, *Proc. 13th IEEE Symp. Computer Arithmetic*, 1997, 44–51, for a number system based on $2^i3^j$.


10. Study the relationship between Bresenham’s algorithm from computer graphics for drawing a digitized straight line, and the theory of Sturmian words. See, for example,


12. Normal numbers. A real number is normal in base-$k$ if every digit (or block of digits) occurs equally often in its base-$k$ representation. Study the kinds of transformations that preserve normality. See, for example, F. Blanchard, J. M. Dumont, and A. Thomas, Generic sequences, transducers and multiplication of normal numbers, *Israel J. Math.* **80** (1992), 257–287.


14. Explore the connection between finite automata and unsolvable tilings of the plane. See, for example, J.-P. Allouche and O. Salon, Finite automata, quasicrystals, and Robinson tilings, in I. Hargittai, ed., *Quasicrystals, Networks, and Molecules of Fivefold Symmetry*,


21. Look into the properties of numbers expanded in non-integer bases (called β-expansions). See the references in the course textbook.

22. Read about the hierarchy induced by finite-state transducers and the Thue-Morse sequence. See Endrullis, Hendriks, Klop, Degrees of streams, *INTEGERS* (2011), and other papers by these authors.

23. Read about automatic sequences and music. Both the Danish composer Per Nørgård and the American composer Tom Johnson have composed music using these sequences.


31. Jason Bell has done some work connecting Amitsur-Levitzki’s theorem on matrices (see the Wikipedia page) and the \(d\)-power property on matrices to prove results on \(k\)-regular and \(k\)-automatic sequences. For example, see the talks in the video archive https://www.fields.utoronto.ca/video-archive/event/1041.

### 3 Some suggestions for Wikipedia articles

You can develop your own list of articles, but here are some suggestions.

*Baum-Sweet sequence*

*Rudin-Shapiro sequence* (currently doesn’t refer to the original papers of Rudin, Shapiro, or Golay; doesn’t give the original motivation for the problem (see paper of Rudin), etc.).

*Automatic sequence:* add info about the logical approach.

*\(k\)-regular sequence:* expand what is currently there.

*Systems of numeration:* a survey, extracting a few items from Chapter 3 of the course textbook.
Unavoidable pattern.

Square-free word.

Subword complexity. (You will have to create it.)

Sturmian word.