Project Specification

Theory of Computation (CSCI 2210)

due: 2023-12-14

Introduction

In lieu of a final exam, students will complete a term project in which they research a model of computation of their choice that we did not study in our course. The main aspects of this project are as follows:

- Students may work alone or in groups of up to three.
- Students may choose their own research topic, subject to instructor approval.
- Students will prepare both a classroom presentation and a written report of their findings.

Topics

Students should choose a model of computation that interests them. A list of some possible topics appears at the end of this document, but is not intended to be comprehensive or normative. Students are encouraged to pick a topic related to their own academic or personal interests.

Groups

Students may, but need not, work in small groups. Students interested in researching the same topic should consider either working together in a group, or else coordinating their separate presentations to avoid duplication. The Instructor will inform student whose topics are closely related of any potential overlap. Students working together in a group are jointly responsible for the outcome of their project.

Presentations

In-class presentations will commence on December 4. As that date approaches, the instructor will schedule individual presentations for specific dates. It is anticipated that presentations will fill the last week of the semester as well as the meeting time of our scheduled final exam. Students are expected to attend all of the presentations, not just their own.

Presentations should last approximately 12–15 minutes per person. Teams with more members are therefore afforded more time to make their presentations. Each team member should participate meaningfully in the presentation. At the end of each presentation there will be time for questions from the audience.

Written Reports

In addition to the classroom presentation, students will submit a written report. The report should be submitted as an electronic document in pdf format no later than the day of our scheduled final exam, December 14.

There is no length requirement, but the written report should clearly describe the topic researched by the student(s). This includes an introduction containing context and motivation, a clear presentation of the *model* *of computation* in question, as well as some selected aspects of the *theory* of this model of computation. You should include proper citations of the sources consulted as well as a bibliography.

Evaluation

Research projects will be evaluated for both ambition and execution. The classroom presentation and written components will be given equal weight. All team members working together on a project will receive the same grade.

Schedule

- Please inform your instructor of your proposed topic and collaborators no later than November 8.
- We will agree a schedule for presentations together in class on November 27. Willingness to present toward the beginning of the allotted days will be looked on favorably.
- Presentations will commence at the class meeting of December 4, and continue on December 6 and December 14.
- Written reports are due no later than midnight on December 14.

Some Possible Topics

logic programming A family of declarative models of computation where computation involves exploring a search space for an expression satisfying a set of constraints. This is the model of computation for programming languages such as PROLOG.

combinator systems A family of models of computation without variable binding. The S–K system of combinators is universal in that it is interdefinable with the λ-calculus.

reversible computation A family of model of computation where all operations are reversible. Surprisingly, this turns out to be universal, and has implications for the minimum thermodynamic cost that an algorithm can have.

quantum computation A model of computation that harnesses the inherent parallelism of reality, at the cost of producing only probabilistic results.

first-class continuations A *continuation* is an object representing “the rest of the program”. In conventional programming languages continuations are implicit and represented by the state of the call stack. However in some models of computation continuations are data that can be manipulated by programs directly.

tree automata The finite automata that we have studied in this course act on *strings*; that is, their input consists of candidates for string languages. One possible generalization is to describe finite automata for *trees*, such machines take as inputs candidates for expression languages.

finite state transducers Models of computation for these resemble those for finite state automata, except that instead of returning a bit (accept/reject) a machine transforms a string over its input alphabet into a string from its output alphabet.

dependent type theory A typed λ-calculus expressive enough to encode higher-order logic. It has been proposed as a practical foundation for constructive mathematics, and most modern proof assistants, including Coq, Agda, and Lean are based on a version of this.

SECD machines An influential family of stack-based models of computation that serve as virtual machine targets of compilation for programming languages.

Petri nets A model of computation for discrete-event dynamical systems. They are used to model distributed systems in computing, population ecology, and epidemiology.

cellular automata A family of models of computation involving a regular array of cells each of which can be in a given state, and where the state of each cell influences the states of its neighbors. This is the basis of Conway's famous *game of life*.

rewrite systems A family of models of computation involving repeatedly replacing some aspect of the state of a system by another that has the same boundary. The λ-calculus together with the operation of β-reduction is an example of such a system. Rewrite systems are the abstract setting for the study of properties such as confluence and termination.

process calculi These are models of computation useful for describing the behavior of interacting systems engaged in a *protocol*. A famous example is the π-calculus.

game semantics A model of computation presented in the form of an adversarial "game" between two parties, where an effective algorithm is represented by a winning strategy.