Columbia Computer Science

Combination Track Proposal
“Knowledge” & The Computer

Name: Arman Jindal Advisor: Professor. Luis Gravano Last Update: 09/17/22

Introduction

Guiding Question - How do we transmit and represent “knowledge” to a Computer?

By asking this question as a computer scientist, philosopher and historiographer, I aim to understand the foundations of modern computers and the full extent of their benefits and limitations.

One way to understand the development of the computer is by a progressively abstract way of transmitting and representing knowledge to a computer. When computers entered universities in the late 1960s, few had the skills to communicate with computers. Fast forward to today, everyone is talking to computers at some level of granularity.

By understanding each level of abstraction and the “language” we use to transmit and represent “knowledge” to a computer, I hope to better understand the kinds of “knowledge” a computer can have, if any, and the implications of these innovations.

Computer Science Electives

List at least three 4000-level or above computer science courses that you propose to take as part of your program. These must add up to at least 9 points. At least two of these courses should come from the list below. Furthermore, a third CS elective must have substantial technical content.

1. ***Operating Systems I COMS 4118***
2. ***Natural Language Processing COMS 4705***
3. ***Types Languages and Compilers COMS 6998***

Rational:

Viewed as an extension of the CS Core, these CS electives expose me to all the levels of abstraction through which we communicate knowledge with computers and how we came to communicate with them in this way.

Beginning with bits, logic gates, and assembly (*Fundamentals of Computer Systems),* then to higher level mathematical representations (*Computer Science Theory/Introduction to Computational Complexity)*. *Operating Systems I* are about the interface between software and hardware. It shows how high-level languages of different programming paradigms were developed and compiled into assembly language. *Natural Language Processing* represents the most human-like way to interact with computers and is also a way computers can interact with us. Finally, *Types Languages and Compilers* looks at paradigms of languages and the addition of typing to make them safer.

External Electives

List at least three upper level courses outside of computer science that you propose to take as part of your program. These are typically 4000 level courses and must be at least at the 3000 level. The point total for these courses must add up to at least 9 points. For each course provide a brief course description including its prerequisites.

1. Symbolic Logic PHIL3411 - 4 Credits

Corequisites: PHILV3413 Required Discussion Section (0 points). Advanced introduction to classical sentential and predicate logic. No previous acquaintance with logic is required; nonetheless a willingness to master technicalities and to work at a certain level of abstraction is desirable.

2. Epistemology PHIL UN 3960 - 4 Credits

Corequisites: PHIL W3963 Required Discussion Section (0 points). What can we know? What is knowledge? What are the different kinds of knowledge? We will read classic and contemporary texts for insight into these questions.

3. Information-Computing-Infrastructure (HISTGR8479) - 4 Credits

The course introduces the major works in the history of computing and information technologies, with particular attention to transformative methodologically important texts. Students will be likewise introduced to major current works in the history of technology and media studies. The course along the way provides an outline of the development of computing from the late nineteenth century.

Prerequisites

None. The course assumes only an interest in the history of computing or technology and a willingness to go beyond text book history. Additional readings will be available for students with the substantial technical expertise of substantial expertise in science and technology studies. Historians and more humanities-trained students should expect some challenges from the technical material and computer scientists from the historical methodological considerations.

Rational:

Each course exposes me to a critical aspect of my guiding question. *Symbolic Logic* is the foundational language of computation. *Epistemology* studies “knowledge” and questions what can be known and by whom. Knowledge is an essential term in this course of study and needs a considered definition. Finally, the historical perspective from *Information-Computing-Infrastructure* ties the whole track together. The computer is a relatively recent invention. By studying its history by reading the primary texts of the most influential figures in Computer Science, I hope to contextualize all the technical information from my study as a computer scientist. By better understanding the past — specifically how the computer went from highly specialized and technical individuals to the world — I hope to better understand and actively shape its future.