CS 580
The Specification of Software Systems

Modern software systems exhibit high levels of complexity, are distributed, and support concurrent execution of multiple communicating and coordinating processes. The challenges facing programmers today are further exacerbated by our limited ability to reason about concurrent computations. Yet, concurrent algorithms are central to the development of software executing on modern multiprocessors or across computer networks. This course reviews several important classes of concurrent algorithms and presents a formal method for specifying, reasoning about, verifying, and deriving concurrent algorithms. The selected algorithms are judged to have made significant contributions to our understanding of concurrency. Rigorous treatment of the design and programming process is emphasized. Students entering this course must be familiar with predicate calculus and sequential algorithms. Upon completion of this course students will be able to reason completely formally about small concurrent programs and to apply systematically and correctly formal skills to large software systems.

Credit: 3 units
Optional Textbook: *Parallel Program Design—A Foundation* by K. M. Chandy and J. Misra

January 13, 2021

Syllabus

A. INTRODUCTION

1. Overview and motivation (Ch. 1)

B. UNITY MODEL

2. A programming notation (Ch. 2 & 6)
3. A programming logic (Ch. 3)
4. Sample proofs (Ch. 3)
5. More on programming logic (Ch. 3)

C. ARCHITECTURES AND MAPPINGS

6. Program schemas (Ch. 4)
7. Shortest path case study (Ch. 5)
8. Program structuring and composition (Ch. 7)

D. ALGORITHMS

9. Termination detection (Ch.9)
10. Global snapshots (Ch. 10)
11. Mutual exclusion (Ch. 15)
12. Dining philosophers (Ch. 12)
    Drinking philosophers (Ch. 13)
    Garbage collection (Ch. 16)

E. SYSTEM SPECIFICATIONS

13. Case study
14. Case study

F. OTHER MODELS

15. Mobile UNITY