**Introduction to Automatic Software Verification**

이번 강의는 software verification에 대해 대략적인 소개를 하는 강의로, software verification의 두 가지 핵심적인 기법인 intermediate assertion과 abstraction에 대한 설명을 들을 수 있었다. Software verification이란 프로그램 실행 도중 오류로 인해 예기치 못한 상황이 일어나지 않도록 소프트웨어를 미리 분석하고 검증하는 작업에 대해서 연구하는 분야를 말한다. 이 강의를 듣기 전까지는 software verification에 대해 들어본 적이 없었고, 규모가 크거나 중요한 프로그램을 다뤄본 적이 없어 소프트웨어 검증에 대한 필요성을 크게 느끼지 못하였다. 그러나 이 분야가 앨런 튜링의 연구를 시초로 하여 오래 전부터 형성되었다는 사실을 알게 되자 비로소 소프트웨어의 올바른 실행을 위한 장치가 중요한 이유를 간단하게나마 스스로 생각해 볼 수 있게 되었다. 지금 이 시대에는 소프트웨어의 역할이 다양한 분야로 퍼져나가 소프트웨어 없이는 삶이 불편해지거나 경우에 따라 정상적인 생활이 불가능하게 될 정도에 이르렀다. 산업계에서 처리되는 상당수의 업무가 컴퓨터 및 여러 응용 프로그램을 통해 처리되고, 심지어 현대인의 일상 생활과 아주 밀접한 관계에 있는 스마트폰 뿐 아니라 가전 제품, 자동차 등에도 수많은 소프트웨어가 이용되고 있다. 그러므로 소프트웨어의 코드가 올바르게 작성되었는 지를 점검하는 방법은 철저히 연구되어야 한다는 점을 깨달았다. 또한, 강의 시간에 소개된 기법 중 abstraction와 관련된 내용은 조금 이해가 어려웠던 점이 아쉬웠지만, intermediate assertion을 이용해 코드의 중간 중간에 software verification을 위한 문장을 끼워넣는 방식은 앞으로 프로그램을 설계하는 과정에서 꼭 숙지를 하고 습관처럼 만들어야겠다는 생각이 들었다.