

불가능의 의미

장봉준

어떤 문제에 대해서 유한시간 안에 모든 가능한 입력에 올바른 답을 내는 알고리즘은 존재하지 않는다. 그러나 프로그램 분석가는 가능한 모든 프로그램에서 잘못된 점을 찾아내거나 정말로 프로그램이 원하는 일을 하고 있는지 알아내는 방법을 찾는다. 모순이다. 프로그램 분석가는 결코 해낼 수 없는 일에 도전하는 사람이다. 프로그램 분석가들이 하는 일은 어떤 의미가 있는 것일까?

1931년 괴델(Gödel)은 불완전성 정리를 발표하여 수학기계를 뒤흔들었다. 괴델은 불완전성 정리를 통해 자기모순이 없는 공리계에서 참이지만 증명할 수 없는 명제가 존재한다는 것을 밝혔고, 이를 통해 힐베르트의 바램이었던 완전한 수학 체계는 불가능하다는 것이 알려졌다. 또한, 앨런 튜링(Alan Turing)은 1936년 정지 문제가 결정 불가능함을 밝혔으며 이후 1951년 라이스(Henry Gordon Rice)는 이를 일반화하여 프로그램의 중요한(non-trivial) 성질을 증명하는 것은 결정 불가능하다는 것을 세상에 알렸다. 이와 같은 20세기의 업적들은 프로그램 분석의 한계를 명확히 드러낸다. 자동 프로그램 분석을 통해 우리가 원하는 궁극의 답을 찾는 것은 애초에 가능하지 않다. 어디를 가나 자동화된 시스템을 볼 수 있는 현대 사회를 생각해 보면 이러한 결론은 가히 충격적이다. 우리가 매일 사용하는 소프트웨어를 온전히 신뢰할 방법이 없다는 뜻이기 때문이다. 누군가 내 스마트폰에서 개인정보를 탈취할 수 없음을 보장할 수 없고, 비행기 컨트롤러가 제대로 작동하는지 완벽하게 알 수 없다는 말이다.

하지만, 이 모든 상황을 눈뜨고 지켜볼 수는 없다. 범죄를 예방하고, 소중한 인명을 지키기 위해서는 개인정보 탈취를 막고 비행기가 똑바로 날 수 있음을 증명해야 한다. 하지만 불가능하다고 판명된 일을 어찌 해낸다는 말인가? 흥미롭게도 조금만 관점을 바꾸어 생각해 보면 이런 한계가 수학적으로 잘 정의되어 있는 것은 매우 유용한 사실로 다가온다. 세상 어떤 일이든지 해낼 수 있는 것과 해낼 수 없는 것이 있기 마련이지만, 그 경계가 명확히 드러나 있는 일은 드물다. 무엇이 가능한 일인지 모른다면 우리는 영원히 우리가 관찰하고 있는 것이 코끼리인지 아닌지 알 수 없을 것이다. 결정불가능성처럼 수학적으로 잘 정의된 한계점이 있기에 우리는 주어진 자원을 현명하게 사용할 수 있다. 이 명확한 한계 속에서 프로그램 분석가들은 프로그램을 넉넉히 근사(over-approximate)하는 요약해석(Abstract Interpretation)이나, 검증 대상의 크기나 검증에 걸리는 시간을 제한하는 방법 등을 제시하여 왔다. 오탐이나 미탐의 문제가 있을 수 있지만, 소프트웨어 신뢰도를 높일 수 있는 실현 가능하면서도 창의적인 방법들이다.

결정불가능성조차도 막지 못한 프로그램 분석가들의 연구를 보다 보면 이들의 창의성에 감탄하게 된다. 그들은 결정불가능성이라는 결론에 도달하게 된 이유를 깊이 이해하고 질문하였다. 모든 프로그램을 완벽하게 검사할 수 없지만, 정말로 모든 프로그램을 검사해야 하는가? 프로그램을 기계적으로 완벽하게 검사할 수는 없지만, 프로그램을 꼭 기계적으로 검사해야 하는가? 프로그램을 항상 올바르게 검사할 수는 없지만, 사실 어느정도 수준만 보장해도 되는 것 아닌가? 이런 질문들을 통해 프로그램 분석가들은 인류가 작성하는 프로그램의 특징, 인간이 만들어내는 오류, 현실적으로 허용할 수 있는 오류 등 소프트웨어에 대해 더 깊이 이해할 수 있게 되었으며 지금도 새로운 아이디어와 방법들을 창조해내고 있다.

나는 지금 프로그램 분석가들이 마주한 도전들에서 소프트웨어와 계산에 대한 본질적인 질문들을 발견할 수 있다고 믿는다. 18세기 증기기관이 발전하고 나서야 19세기에 열역학 법칙들이 발견되었던 것처럼 지금 인류는 마치 증기기관을 만들어내던 것과 같이 프로그램 분석의 걸음마를 떼고 있는 것일 수도 있다. 여기서 에너지 보존법칙과 같은 결정불가능성을 먼저 이해했다는 사실은 사실 어마어마한 축복일 수도 있는 것이다. 무한동력기관이 없다고 좌절하는 사람이 없듯이, 결정불가능하다고 프로그램 분석가의 일이 끝나는 것은 아니다. 열기관의 한계에 도전해 수많은 기술이 발전한 것처럼 꾸준히 결정불가능성의 한계에 도전한다면 컴퓨터 과학은 눈부시게 발전할 수 있다고 본다. 이를 위해 프로그램 분석가로서, 또한 컴퓨터 과학자로서 우리의 약점을 극복하기 위한 본질적인 원인을 찾기 위해서 예리한 질문을 끊임없이 던져야 할 것이다.