

교육 및 연구계획서

전민석

April 5, 2024

교육과 연구는 거대한 순환관계를 형성한다. 교육은 연구의 결과물들을 학생들에게 전달하고, 학생들은 이를 소화하여 자신의 문제를 해결하는 능력을 키우고 새로운 문제에 도전하며 이 과정에서 연구를 통해 새로운 지식을 창출한다. 새로이 창출된 지식은 다시 교육에 반영되어 새로운 학생들에게 전달된다. 나 또한 새롭고 가치 있는 순환을 일으키는 것을 목표로 하고 있다.

새롭고 가치있는 연구와 교육의 순환을 만들어내기 위해선 문제의 본질을 꿰뚫어볼 수 있는 능력을 길러야 한다. 연구는 문제를 풀어내는 과정이지만, 문제의 본질을 이해하는 과정이기도 하다. 주어진 문제를 제대로 해결하기 위해서는 현재 문제가 발생한 본질적인 원인이 무엇인지, 왜 이것이 본질적인 원인인지 이해하며, 이를 근본적으로 해결하기 위한 방법이 무엇인지를 고민해야 한다. 사람마다 문제의 본질에 대한 고민과 이해가 제각기 다르기 때문에, 제각기 다른 해결책이 제시되고, 이를 통해 기존의 지식이 발전되기도 하고 뒤집어지기도 하며 교육과 연구의 건강한 순환을 일으킨다. 반면, 문제의 본질을 스스로 고민하지 않은 채 다른이들이 내놓은 연구들을 따라 문제에 접근하는 것은 제대로된 해결책을 내놓을 수 없을 뿐더러 교육과 연구의 건강한 순환을 만들어낼 수 없다.

문제의 본질을 꿰뚫어보기 위한 나의 방법은 문제에 적합한 특화 프로그래밍 언어 디자인 (domain-specific programming language design) 및 특화 언어 프로그램 합성 알고리즘 개발이다. 문제에 대한 본질적인 해결책을 찾기 위해서는 본질적인 해결책을 표현할 수 있는 언어가 필수적이다. 특화 프로그래밍 언어 디자인이란 문제의 정답을 하나의 프로그램으로써 표현할 수 있는 언어를 디자인하는 것이고, 특화 언어 프로그램 합성 알고리즘 개발이란 특화 프로그래밍 언어로 표현된 공간 안에서 문제의 해답이 되는 정답 프로그램을 자동으로 찾아내는 알고리즘을 개발하는 것이다. 문제의 정답을 표현할 수 있는 언어를 디자인하는 과정속에서 문제의 본질에 대한 고민을 하게 되고, 정답 프로그램을 합성하는 알고리즘을 개발하는 과정에서 문제의 성질을 이해하게 된다. 위 철학을 기반으로 지금까지 프로그램 분석, 설명가능한 기계학습, 프로그래밍 교육 등 다양한 컴퓨터학 분야의 문제에 대한 해결책들을 제시해왔다.

이 연구 방향의 중단기적 목표는 지금까지 개발해온 특화 언어 및 합성 알고리즘을 개선하거나 적용 범위를 다양한 분야로 확장시키는 것이다.

궁극적으로는 주어진 문제에 대해서 자동으로 특화 프로그래밍 언어 및 특화 언어 프로그램 합성 알고리즘을 생성하는 시스템을 개발하고자 한다. 현재까지 개발한 특화 언어 및 합성 알고리즘들은 모두 내가 직접 수동으로 문제를 분석해 디자인한 것들이고, 이 과정은 시간과 노력이 매우 많이 소모되는 작업이었다. 자동으로 언어 디자인 및 알고리즘 개발 작업을 수행하는 시스템을 개발하여 위 수고를 덜어주고, 특화언어 기반 문제 접근법을 하나의 대중적인 방법론으로 만들어내는 것이 중장기적 목표이다.

학생들에게도 특화 프로그래밍 언어 디자인 및 프로그램 합성 알고리즘 개발을 통한 문제 접근을 방법을 가르치고자 한다.

교육의 본질은 학생들이 스스로 문제를 해결할 수 있는 힘을 길러주는 것이고, 연구의 교육은 학생뿐만 아니라 교수, 조교, 학계에도 큰 영향을 미치는 가장 중요한 학술적 활동입니다. 수업에서 학생들은 기존 지식에 직면하며, 그들의 이해는 시험이나 과제로 평가됩니다. 그러나 저는 교육의 목표가 기존 지식을 전달하는 것뿐만 아니라 학생들이 습득한 지식을 활용하거나 새로운 지식을 창출하여 자신의 문제를 해결할 수 있도록 하는 것이라고 믿습니다. 잘 설계된 수업 자료가 이 목표를 달성하는 데 중추적인 역할을 한다는 것을 경험적으로 알고 있습니다 [1].

References

- [1] Sehun Jeong, Minseok Jeon, Sungdeok Cha, and Hakjoo Oh. Data-driven context-sensitivity for points-to analysis. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA), 2017.