

교육 및 연구계획서

전민석

April 8, 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.

학생들에게도 특화 프로그래밍 언어 디자인 및 프로그램 합성 알고리즘 개발을 통한 문제 접근을 전달하고자 한다. 이를 위해 기반이 되는 과목인 계산이론, 프로그래밍 언어, 프로그램 분석, 컴파일러, 소프트웨어 공학, 알고리즘을 강의하고자 한다. 이후 특화 프로그래밍 언어 디자인 및 프로그램 합성 과목을 개발하여 학생들에게 전달할 것이다.

1 강의 가능 과목

본 연구자의 배경은 프로그래밍 언어/소프트웨어 공학 분야이기에 아래와 같은 과목들을 강의하고자 한다.

계산이론. 이 과목의 목표는 언어의 문법을 정의하는 법 및 튜링머신과 계산 가능성에 대한 이해를 제공하는 것이다. 과목의 첫번째 목표는 학생들은 오토마타, 정규표현식, 문맥 자유 문법을 전달하고, 이들의 표현력과 한계에 대해 이해하게 하는 것이다. 이 과정에서 (특화) 프로그래밍 언어에서의 문법을 정의하는 방법에 대한 기초적인 이해를 제공한다. 이후 컴퓨터과학의 기초가 되는 튜링머신을 배우게 되고, 이를 통해 계산 가능성에 대해 이해하게 된다. 마지막으로 계산 불가능성에 대해 배우게 되고, 이를 다양한 분야의 문제들(특히 정적 프로그램 분석)과 연관지어 설명할 것이다.

프로그래밍 언어. 이 과목의 목표는 프로그래밍 언어의 주요 개념들을 이해하고, 이를 통해 프로그래밍 언어를 디자인하는 방법을 배우는 것이다. 학생들은 우선 기본적인 프로그램 언어의 컨셉을 배우게 되고, 실제로 프로그래밍 언어를 구현해보는 과정을 통해 프로그래밍 언어의 문법과 의미를 정의하는 방법을 익히게 된다. 이후 타입 시스템과같이 진보된 프로그래밍 언어 개념들을 배우고 실제로 구현해봄으로써 프로그래밍 언어에 대한 이해를 높인다. 마지막으로, 학생들은 자신들만의 문제를 해결하기 위한 특화 프로그래밍 언어를 디자인해보는 프로젝트를 수행하게 하여 특화 언어 디자인을 통한 문제 접근을 경험해볼 수 있도록 할 것이다.

프로그램 분석. 이 과목의 목표는 프로그램 분석의 기본적인 개념들을 이해하고, 이를 통해 프로그램 분석 도구를 디자인하는 방법을 배우는 것이다. 학생들은 다양한 프로그램 분석 기술들을 배우게 되고, 이를 실제로 구현해봄을 통해 프로그램 분석 기술의 동작 방식을 이해하고 개선하는 방향을 고민해보게 된다. 더 나아가 프로그램 분석 기술이 다양한 분야에 어떻게 적용되는지 배우게 될 것이다. 특히 프로그램 자동 합성에 프로그램 분석 기술이 어떻게 활용되는지에 대해 이해하게 될 것이다.

컴파일러. 컴파일러 과목의 목표는 컴파일러의 중요성 및 주요 개념들을 이해하고, 컴파일러를 설계하는 방법을 배우는 것이다. 학생들은 어떻게 하나의 프로그래밍 언어가 다른 프로그래밍 언어로 의미 변화 없이 안전하게 번역되는지에 대해 배우게 된다. 학생들은 컴파일러의 주요 구성 요소인 렉서, 파서, 번역기, 최적화기를 구현해보는 프로젝트를 수행하게 되어 컴파일러의 동작 방식을 이해하게 된다. 더 나아가 정적 분석 기술이 컴파일러 최적화에 어떻게 활용되는지에 대해 배우게 될 것이다.

소프트웨어 공학. 이 과목의 목표는 소프트웨어 개발의 주요 단계들을 이해하고, 소프트웨어 개발 방법론을 배우는 것이다. 학생들은 소프트웨어 요구사항, 소프트웨어 디자인, 소프트웨어 개발, 소프트웨어 분석, 소프트웨어 유지보수의 주요 개념들을 배우게 된다. 이 과정에서 최근의 소프트웨어 공학 연구들을 소개하고, 학생들은 최신 소프트웨어 공학 연구 위에서 이를 개선해보는 프로젝트를 수행하게 하여 최신 소프트웨어 공학 기술 연구에 도전해볼 수 있는 기회를 제공할 것이다.

2 개발하고자 하는 과목

위 과목들을 기반으로 본 연구자의 연구 방향인 특화 프로그래밍 언어 디자인 및 프로그램 합성 과목을 개발하고자 한다.

특화 프로그래밍 언어 디자인 및 프로그램 합성.

이 연구 방향의 중단기적 목표는 지금까지 개발해온 특화 언어 및 합성 알고리즘을 개선하거나 적용 범위를 다양한 분야로 확장시키는 것이다. 현재까지 개발한 특화 언어들은 초기 단계의 프로토타입 언어들이며,

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.[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.