

컴퓨터공학의 철학 버클리 학생 모임

백 종민 제롬, 트란 알렉산드라, 달라르 니하르
지도 교수진: 데네로 존 & 알신 세스
pocab.org

서기 2017년 3월

1 왜 또 새로운 학생 모임인가?

컴퓨터공학에서 컴퓨터가 차지하는 역할은 천문학에서 망원경이 차지하는 역할 이상이 되지 못한다. (다익스트라 에드스거)

“컴퓨터공학”이라는 말을 듣고 철학이나 시를 떠올리는 사람은 별로 없을 것입니다. 비슷한 레로 “컴퓨터공학의 아버지”인 튜링 앨런이 철학자였다는 사실이나, “컴퓨터공학의 어머니”인 러브레이스 아다가 시인이었다는 사실을 아는 사람은 별로 없을 것입니다. 실리콘 벨리의 화려한 빌딩과 묵직한 돈뭉치 뒤로 그렇게, 컴퓨터공학의 역사와 창시자들의 질문은 서서히 잊혀가고 있습니다. 이 건 참 안된 일인데, 왜냐하면 컴퓨터공학의 창시자들이 컴퓨터공학을 만든 이유는 돈이나 명예가 아니라 어쩌면 시적이고, 기본적이고, 아이같은, 그러니까 다음과 같은 질문들이었기 때문입니다 –

다음 사항을 묻고자 한다. 기계는 생각을 할 수 있는가? (튜링 앨런)

나는 무엇보다 가장 **화음적으로** 훈련된 군사를 갖게 될 것이다; – 그러니까, 거대한 **숫자**로 구성된, 그리고 **음악**에 맞추어 막을 수 없는 힘으로 전진하는. 이것은 정말 알 수 없는 것이 아닌가? 하지만 그러고 보면, 도대체 **무엇**인가, 이 **숫자**들은? 여기에 수수께끼가 존재한다 – (러브레이스 아다의 유언 중)

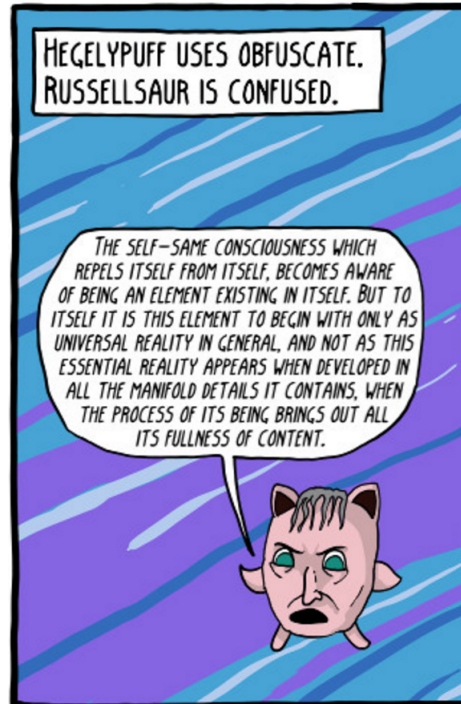
이 철학적인 질문들이 무시되는 현실은 두번 안된 일인데, 왜냐하면 창시자들이 죽은 뒤 수십 년의 시간이 흘렀고, 그동안 컴퓨터공학의 빛은 계속해서 전진했고, 그래서 지금 우리는 그들 창시자들보다 훨씬 멀리 내다 볼수 있는 상황에서 그들의 질문들을 더 잘 이해하고 더 잘 대답할 수 있기 때문입니다! 컴퓨터공학에서 파생된 여러가지 아이디어들, 예를 들어 계산 불가능성, P vs. NP, 그리고 양자 컴퓨팅은 아동발달 심리학, 언어, 진화론, 문화, 인식론, 형이상학, 도덕론 등등의 다채로운 학문들 속으로 이미 스며들고 있습니다. 하지만 컴퓨터공학이 철학적인 뿌리를 가지고 있고 철학적인 질문에 대한 해답을 제시한다는 개념은 우리의 시대정신 속으로 좀처럼 들어오지 못하고 있습니다.

컴퓨터공학의 철학에 대한 이야기가 부족하다는 사실보다도 컴퓨터공학의 철학이 존재한다는 것 자체를 모르는 사람이 대부분이라는 사실을 우리는 바꾸고자 합니다. 컴퓨터공학의 철학이 존재한다는 것 자체를 모르는 사람이 대부분이기 때문에 컴퓨터공학은 물질주의적이고, 엘리트 의식이 가득한, 그리고 목적에 달하기 위한 수단으로 보통 생각들을 합니다. 우리는 많은 수의 학생들이 컴퓨터공학의 철학에 대해 이야기를 하고 생각을 하는 학술적인 문화를 발전시키고 싶습니다. 그러면 컴퓨터공학이 목적을 위한 수단이 아닌, 그 자체의 목적이 되겠지요. 이 변화된 학술적인 문화를 통해 보다 많은 학생들에게 컴퓨터공학을 공부할 동기를 부여할 뿐더러, 더 다채로운 배경을 가진, 다채로운 시각을 가진, 그리고 다채로운 집중 분야와 역량을 가진 사람들을 우리 컴퓨터공학부가 환영할 수 있도록 바꿀 것이라 믿습니다.

2 왜 하필이면 컴퓨터공학의 철학인가?

“컴퓨터공학”은 사실 좀 잘못 붙여진 이름이에요. 좀 더 나은 이름을 제안해보자면, “정략적인 인식론.”
(아아론슨 스콧, 버클리대 박사 2004)

근데 왜 하필이면 컴퓨터공학의 철학을 공부하자는 것이고, 누가 여기에 관심이 있을 이유가 있냐고요? 따지고 보면 철학은 이상한 말만 주절주절 늘어놓고, 엘리트 의식에 가득 차 있고, 우리의 보편적인 삶과 아무런 관련이 없잖아요.



계산이라는 관점을 통해서 묻는 철학적인 질문은 다음과 같은 이유로 조금 특별하다고 믿습니다. 이 관점은 다른 어느 관점보다 정량적이고, 정규적이고, 정확하기 때문입니다. 그래서 컴퓨터공학을 공부한 사람, 컴퓨터적인 생각을 할 줄 아는 사람이라면 철학적인 아이디어를 쉽게 이해할 수 있습니다. 어떻게 보든 컴퓨터공학의 이론에서 나온 여러가지 중요한 아이디어들, 예를 들어 P vs. NP, 는 정말이지 직관적이라서 컴퓨터공학을 한 번도 공부해보지 못한 사람도 금세 무슨 말을 하는 것인지 알아차릴 수 있습니다. 게다가 이 관점은 수학적이기 때문에 우주적인 보편성을 지니고 있어, 다채로운 문화적인 사고방식이 다 같이 존중되고 이해될 수 있습니다 - 현재 철학 학풍은 문화적인 다양성이 거의 없다는 치명적인 약점을 가지고 있는데, 컴퓨터공학의 철학은 그래서 이 가려운 점을 긁어줄 수 있죠. 또 하나 중요한 이유는, 꽤 이른 시간 내에 **아무도** 컴퓨터공학이 던지는 철학적인 질문을 피해갈 수 없을 것이기 때문입니다: 앨런이 **기계는 생각을 할 수 있는가?**하고 질문을 던진 후 70년이라는 시간이 흘렀고, 이제 그 질문에 대한 해답이 어찌면 엄청난 속도로 우리 모두를 향해 다가오고 있습니다. 그리고 물론 컴퓨터공학의 철학은 정말 재미있다는 이유가 있지요. 우리가 통상 던지는 철학적인 질문을 계산의 관점으로 보면 전혀 예상치 못한 해답이, 우리가 세상을 보는 시선을 극단적으로 뒤집어엎을 수도 있는 해답이, 그런 해답이 저기 멀리서 우리를 가만히 기다리고만 있었을 수도 있다는 것을 알게 될 수도 있습니다.

3 학생모임의 구조

간략하게 말하자면, 우리 모임은 P vs. NP, 괴델의 불완전성 증명, 혹은 인류를 위협하지 않는 인공지능을 만드는 방법 같은 이야기를 담은 에세이나 책을 몇 챕터 읽고 일주일에 한번씩 만나 두시간 동안 이야기를 나눌 것입니다. 교수님이나 연구원님의 초청강연도 할 수 있는대로 할 것입니다. 아이디어가 있는 모임 멤버는 자신의 아이디어를 담은 에세이나 책을 쓰는데 도움을 받을 수 있고, 그것을 세상에 내놓을 공간이 주어질 것입니다.

4 몇 가지 주제 및 책

4.1 배경지식

정보의 역사, 정보의 이론, 정보의 홍수, 글레익 제임스
로지코믹스: 진리를 찾아서, 독시아디스 & 파파디미트리우
The Information: a History, a Theory, a Flood, James Gleick
Logicomix: An Epic Search for Truth, Doxiadis & Papadimitriou

4.2 계산 복잡성 이론

철학자들이 계산 복잡성 이론에 관심을 가져야 하는 이유, 아아론슨 스콧
Why Philosophers Should Care About Computational Complexity, Scott Aaronson

4.3 양자 컴퓨팅

데모크르토스부터 알아보는 양자 컴퓨팅, 아아론슨 스콧
양자 튜링 기계 안에 있는 귀신, 아아론슨 스콧
Quantum Computing Since Democritus, Scott Aaronson
The Ghost in the Quantum Turing Machine, Scott Aaronson

4.4 튜링, 괴델, 자기언급과 계산불능성

괴델, 에셔, 바흐, 호프스대터 더글라스
Gödel, Escher, Bach, Douglas Hofstadter

4.5 강인공지능, 약인공지능, 인공지능 도덕, 계산적 도덕

마음, 뇌, 그리고 프로그램, 시얼 존
초지능, 보스트롬 닉
지식을 찾는 보편적인 행위자, 오르수 로렌트
Minds, Brains, and Programs, John Searle
Superintelligence, Nick Bostrom
Universal Knowledge-Seeking Agents, Laurent Orseau

4.6 진화론, 밈, 밈 이론

박테리아부터 바흐까지, 그리고 다시, 데넷 데니엘
From Bacteria to Bach and Back, Daniel Dennett

4.7 계산적 문화, 은유, 아동 발달

생각의 발달, 피아제 장

삶으로서의 은유, 레이코프 조지

논어, 공자

The Development of Thought, Jean Piaget

Metaphors We Live By, George Lakoff

The Analects, Confucius