

SNU 4190.210 프로그래밍 원리(Principles of Programming) Part 0

Prof. Kwangkeun Yi

School of Computer Science & Engineering

목표

- ▶ 익히기:
 - ▶ 프로그램 작성의 기초와 핵심
 - ▶ 프로그램 실행의 개념
- ▶ 균형잡기:
 - ▶ 프로그램이 컴퓨터를 사용하는 도구
 - ▶ 컴퓨터가 프로그램을 실행하는 도구
- ▶ 눈높이 맞추기:
 - ▶ 프로그래밍 도구 기술의 현재와 미래

목표: 이 과목을 수강후 “아하”

- ▶ 프로그램은 이렇게 짜는 것이구나
 - ▶ 이렇게 생각하면서 프로그램을 구성하는 것이구나
- ▶ 프로그래밍 기술의 현재수준이 이렇구나
- ▶ 프로그래밍 기술의 미래가 이렇게 되겠구나
- ▶ 어떤 언어로 짤다고 해도 자신이 생긴다

내용

▶ 프로그램 작성의 원리

간단한 부품의 반복구성	elements & compound
재귀와 반복	recursion & iteration
속내용 감추기	procedural & data abstraction
계층구조로 속내용 감추기	modularity & hierarchy
맞는 프로그램인지 확인하기	program proof
타입으로 정리+이해하기	types & typeful pgm'ng
물건중심의 프로그래밍	objects & imperative pgm'ng
값중심의 프로그래밍	value & applicative pgm'ng
예외상황 관리	exceptions & advanced control

▶ 프로그램 실행의 개념

- ▶ 이름(name, variable), 환경(environment), 메모리(memory, state), 실행(evaluation, interpretation, semantics)

Not

- ▶ 특정언어를 가르치는 과목이 아님
- ▶ 집중하자: 원리(기초와 핵심)
 - ▶ 실습언어는 그것을 전달하는 도구일 뿐
- ▶ 이 강의에서 익힌 내용은 장차 어떤 언어를 사용하더라도 도움+적용된다

실습언어의 선택: Scheme + ML

- ▶ 작고 간단
 - ▶ 집중할 수 있다: 프로그래밍 원리 배우기에
 - ▶ 필요없다: 복잡한 문법과 혼동스런 잔재주
- ▶ 강력하고 실제적
 - ▶ 상위레벨 프로그래밍
 - ▶ 상용 소프트웨어 시스템에 사용됨
- ▶ 복음: 생소한 실습언어
 - ▶ C/Java 경험이 많다고 유리하지 않음
 - ▶ C/Java 경험이 없다고 불리하지 않음
 - ▶ 모두 같은 선에서 출발

프로그래밍

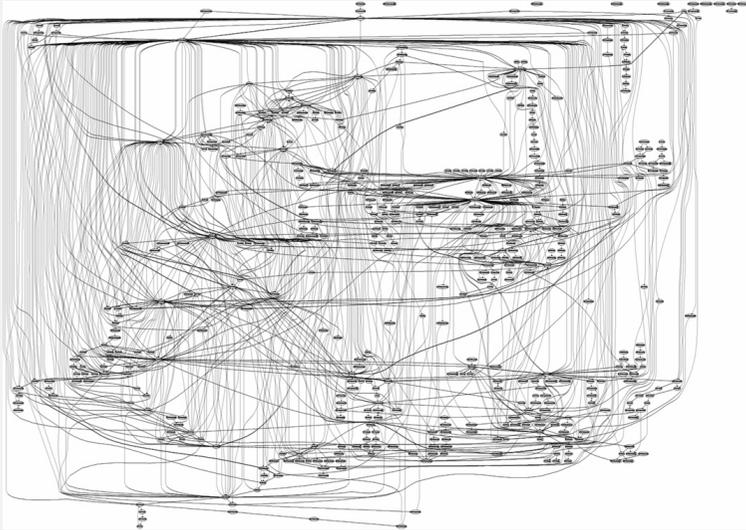
- ▶ 뭔가를 해내는("계산하는") 방법을 서술하기
 - ▶ $\sqrt{2}$ 를 계산하는 방법
 - ▶ 자장면을 만드는 방법
- ▶ 뭔가를 정의하기,와는 다르다
 - ▶ $\sqrt{2}$ 는 제곱하면 2가 되는 수
 - ▶ 자장면은 가장 싸고 맛있는 중국음식

프로그래밍의 어려움

- ▶ 프로그램의 규모와 복잡도가 점점 커짐
 - ▶ sw 복잡성의 증가속도 >> hw 성능의 성장속도
 - ▶ "sw는 가스다." "Software is gas."
- ▶ 프로그램은 기계가 자동으로 실행함
 - ▶ 기계는 우리가 바라는 바를 실행하지 않음
 - ▶ 기계는 프로그램에 적힌 바를 실행할 뿐

"장보기 = 우유 1리터, 신라면 4봉지, 그리고 쌀과자두 사오기."
- ▶ 모든 상황을 고려해야 × 사소한 실수가 없어야
- ▶ 프로그램의 실행을 "미리 완벽히 알기"가 어려움
 - ▶ 자동으로는 불가능: 증명됨(1936년, Alan Turing)
 - ▶ 사람이 확인해야: 다양한 도구로 비용절감
 - ▶ 현재의 기술수준: 걸음마 "3발짝", 초보수준

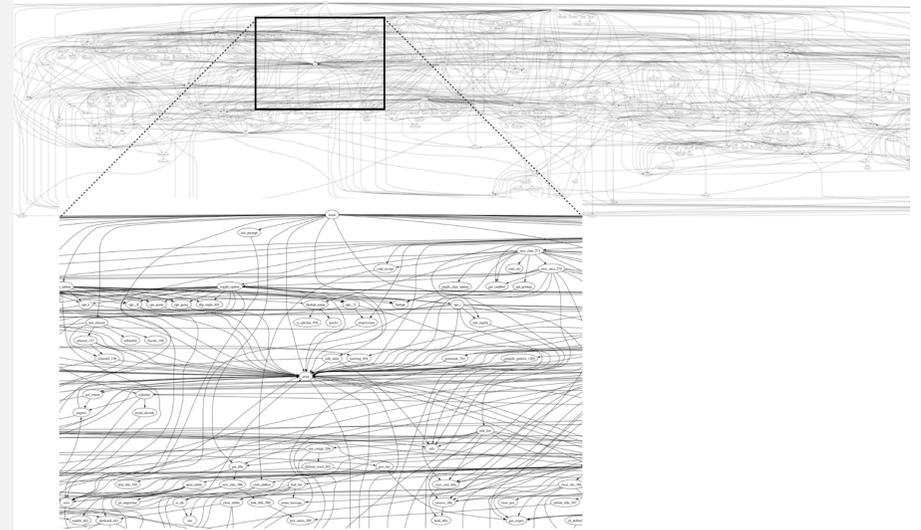
SW는 복잡하다



SNU 4190.210 ©Kwangkeun Yi

SW는 복잡하다

less-382(23,822 LoC)



SNU 4190.210 ©Kwangkeun Yi

자연물 만큼 복잡

포유류 뇌의 1만개 뉴런의 네트워크, Ecole Polytechnique Fédérale de Lausanne, *Blue Brain Project*, 2008



SNU 4190