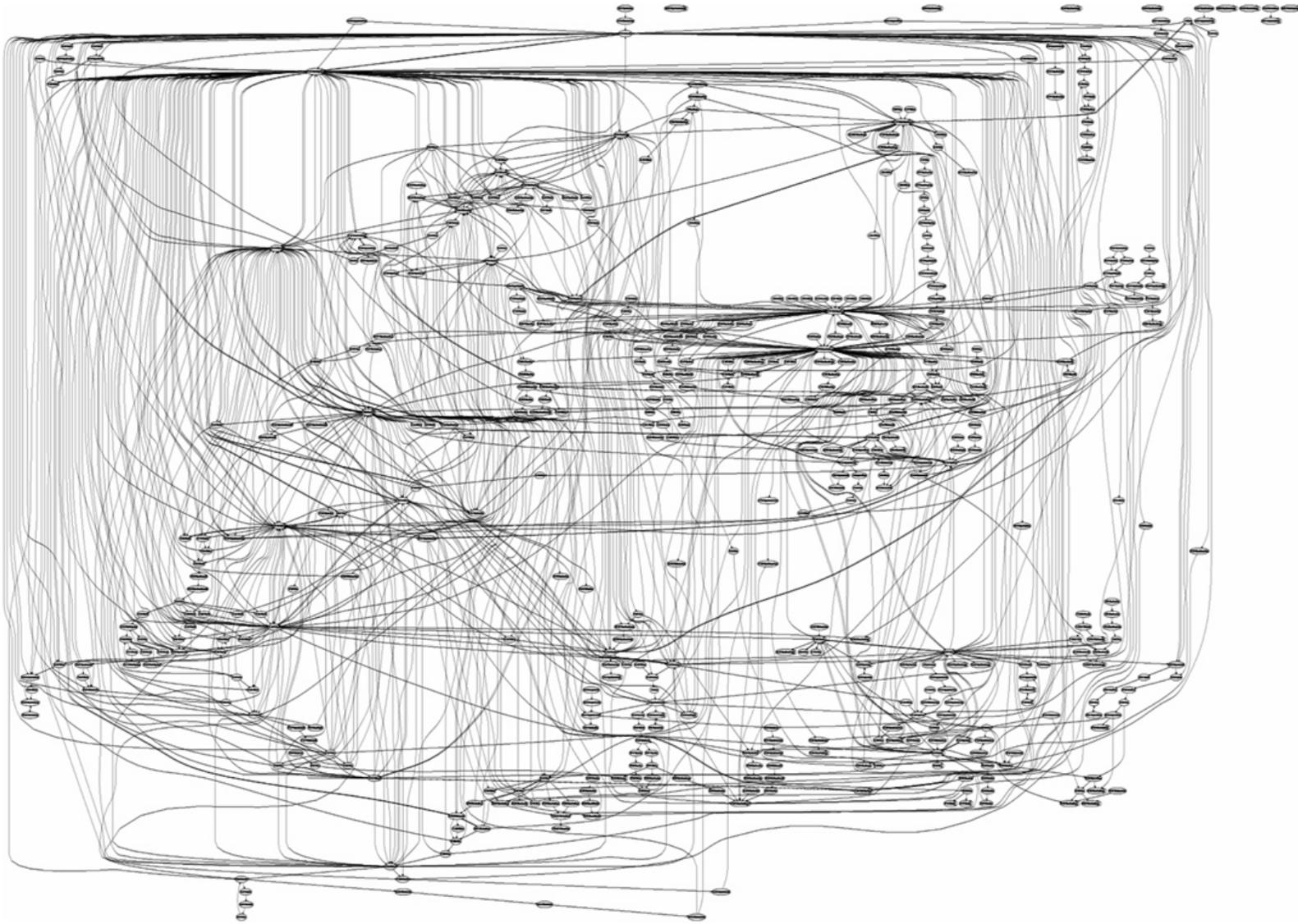
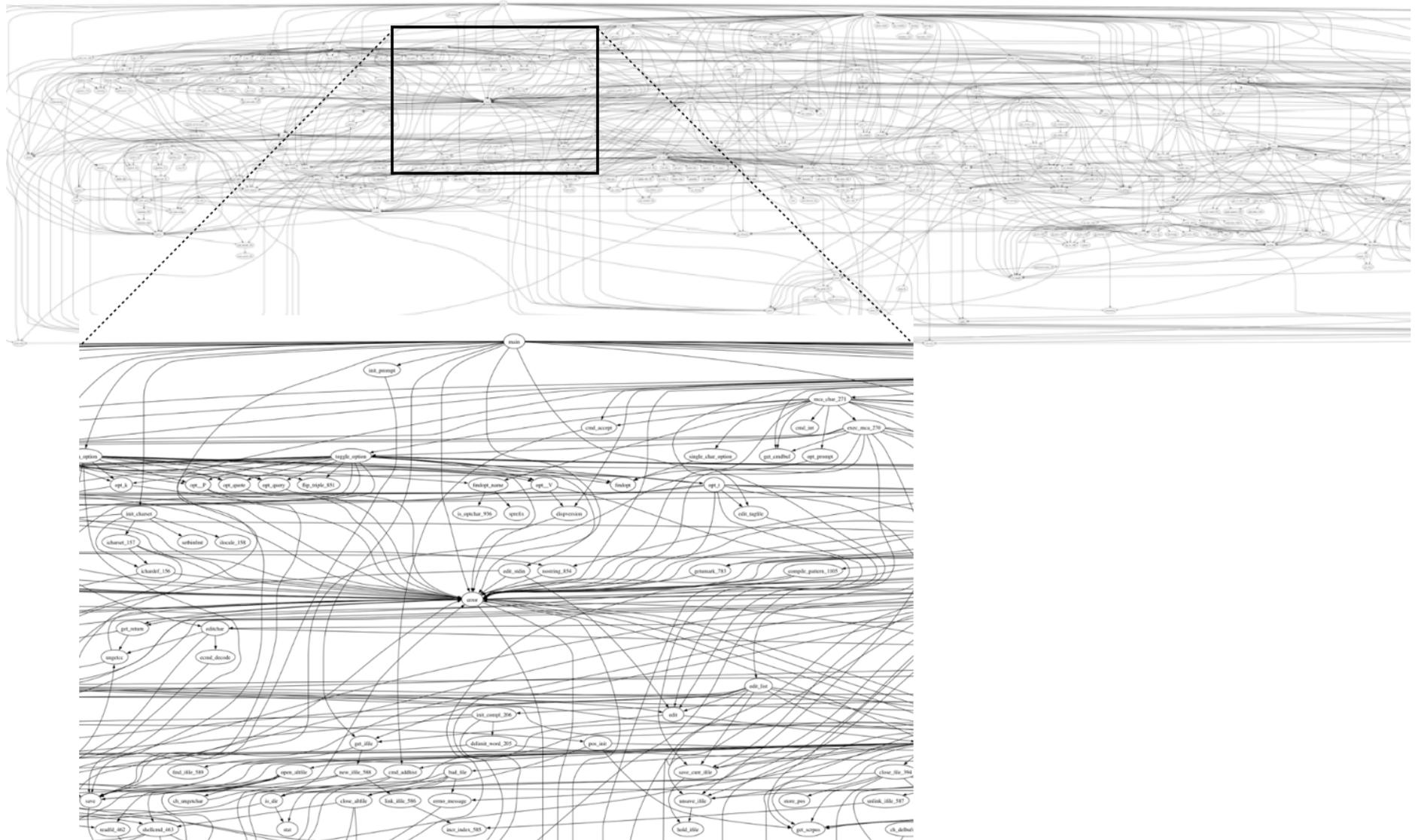


SW는 복잡하다



SW는 복잡하다

less-382(23,822 LoC)



자연물 만큼 복잡

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



SW 제작의 어려움

- ▶ 프로그램의 규모와 복잡도가 점점커짐
 - ▶ 프로그램 복잡성의 증가속도 >> hw 성능의 성장속도
 - ▶ “sw는 가스다.” “Software is gas.”
- ▶ 프로그램은 기계가 자동으로 실행함
 - ▶ 기계는 우리가 바라는 바를 실행하지 않음
 - ▶ 기계는 sw에 적합한 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 확인해가야: 다양한 도구로 비용절감
 - ▶ 현재의 기술수준: 걸음마 “3발짝”, 초보수준

고백: 컴퓨터분야는 아직 미숙합니다

분야의 역사 ~ 겨우 5-60년

그래서

- ▶ 미숙하기 때문에 \implies 기회가 많다.
- ▶ 미숙하기 때문에 \implies 지금 “Newton”, “Galileo”, “Curie”, 와 같이 살고있다.

컴퓨터분야 발전 속도가 빠르다? (2/3)

다른 분야와 얼추 비슷: $10^7 \sim 10^9$ 배 발전/100년

- ▶ 에너지분야: 10^7 배/100년 발전

10^{-1} hp/hr (하인) $\rightarrow 1.35 \times 10^6$ hp/hr (원전 1GW/hr)



- ▶ 교통분야: 10^9 배/100년 발전

10^2 j (가마) $\rightarrow 10^{11}$ j (Delta II)



컴퓨터분야 발전 속도가 빠르다? (3/3)

- ▶ 하드웨어: 다른 분야와 비슷한 발전
- ▶ 소프트웨어: 많이 미개함
 - ▶ 크기만 증가:
 - 휴대폰 100만줄
 - 아래아한글 200만줄
 - 스마트폰 1000만줄
 - 윈도우 3000만줄
- ▶ 소프트웨어: 다른 분야가 이루어 놓은 것을 아직 이루지 못했슴

무엇일까?

다른 분야의 현재 수준

- ▶ 제대로 작동할 지를 미리 검증할 수 없는 기계 설계는 없다.
- ▶ 제대로 작동할 지를 미리 검증할 수 없는 회로 설계는 없다.
- ▶ 제대로 작동할 지를 미리 검증할 수 없는 공장 설계는 없다.
- ▶ 제대로 서있을 지를 미리 검증할 수 없는 건축 설계는 없다.

뉴턴방정식, 미적분 방정식, 통계역학, 무슨무슨 방정식...

모든 기술의 질문

우리가 만든 것이
우리가 의도한대로 움직인다는 것을
어떻게 미리
확인할 수 있는가?

(사실, 일상에서도 잦은 질문과 답: 입학시험, 입사시험, 궁합, 클럽관리)

소프트웨어 분야에서의 이 질문

작성한 SW의 오류를
자동으로 미리 모두 찾아주거나,
없으면 없다고 확인해주는
기술들은 있는가?

그래서, SW의 오류때문에 발생하는
개인/기업/국가/사회적 비용을
절감시켜주는 기술들은 있는가?

SW 오류의 예

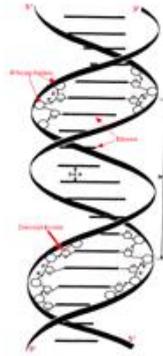
어머니가 전해준 슈퍼마켓 심부름 목록(프로그램)

1. 우유 1리터 2병
2. 우유가 없으면 오렌지쥬스 1리터 3병
3. 우유가 있으면 식빵 500그램 1봉
4. 쌀과 자두

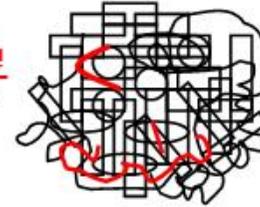
만약에: 우유 1리터 1병만 있으면? 쌀과 자두? 쌀과자두?

SW 오류의 문제

DNA 오류



SW 오류



살다가 병이든다
사회적 비용
해결책 = 경제적기회

SW로 동작하는 제품이 고장난다
사회적 비용
해결책 = 경제적기회



SW 오류: 사람이 손으로 손쉽게 잡을 수 없다

- ▶ 엄청나게 커지고 복잡해 진 소프트웨어
- ▶ 새로운 컴퓨팅 환경: 지구 = 컴퓨터
 - ▶ 전지구적 네트워크를 타고
 - ▶ 불특정 다수의 코드가 내게로 온다(스마트폰 게임/apps)

자동 기술이 필요

SW 오류 검증 기술: 1세대

문법 검증기: 70년대에 완성된 기술. lexical analyzer & parser

오류 = 소프트웨어의 생김새가 틀린 것

1. 우유 1리터 2병
2. 우유가 없으면 오렌지주스 1리터 3병
3. 있으면 빵식 우유가 500그램 1봉
4. 쌀과 자두

SW 오류 검증 기술: 2세대

타입 검증(type checking): 90년대에 완성.
(30년간 프로그래밍언어 분야의 대표 성과)

오류 = 생긴것은 멀쩡한데, 잘못된 값이 계산에 흘러드는
경우

1. 우유를 담은 얼음을 가스 불위에 올려놓고 데운다
2. 식빵을 유리접시에 담고 방아로 빵는다
3. 2의 접시에 1의 우유를 튀긴다
4. 남자와 소나무를 결혼시킨다

SW 오류 검증 기술: 3세대

프로그램분석검증: 아직 미완성

(static analysis, verification, model checking, ...)

오류 = 생긴것도 멀쩡하고 타입도 맞지만, 바라는 바가 아닌 것

오류 = 필요한 조건을 만족시킬 수 없는 경우

1. 우유를 담은 냄비를 가스 불위에 데운다
2. 식빵을 스텐접시에 담고 방아로 빵는다
3. 남자1명과 여자1명을 결혼 시킨다
 - ▶ 가스불이 포항제철 용광로가 된다면?
 - ▶ 방아가 현대중공업의 프레스가 된다면?
 - ▶ 남녀 나이의 차이가 100살이 된다면?