

컴퓨터 과학이 여는 세계

기말 팀 프로젝트

: *컴퓨터 과학이 우리에게 남긴 것들*

2021. 09. ~ 2021. 12.

사회교육과 2021-12552 강승윤

기계공학부 2021-17317 이진무

자유전공학부 2014-16200 이전형

목차

- 01 가르침처럼 가혹한 가르쳐짐은 없다 (이건무)
- 02 시나리오 기획 (강승윤)
- 03 시 사냥 (강승윤)
- 04 완벽한 하인 (이전형)
- 05 영화 속 컴퓨터 과학이 여는 세계 (이전형)
- 06 암호기술로 책을 마치며 (강승윤, 이건무, 이전형)

가르침처럼 가혹한 가르쳐짐은 없다

이건무

건무의 친구 동훈이는 고등학교 때 건무와 함께 컴퓨터 프로그래밍 동아리를 했었다. 동훈이는 프로그래밍에 능숙하게 적응했고, 코딩으로 문제를 해결하는 것을 좋아했다. 그러나 항상 동훈이는 마음 한 켠에 이런 의문을 가지고 있었다. ‘이런 컴퓨터 언어는, 어떤 원리로 작동하는 거지? 단순히 글씨를 쓰는 것만으로 어떻게 컴퓨터가 인식하는 걸까? 나아가, 컴퓨터 자체는 어떤 원리로 만들어지고 작동되는 걸까?’ 그런 의문을 간직한 채, 어느덧 시간이 흘러 둘 다 대학생이 되었고, 마침 동훈이는 건무가 ‘컴퓨터 과학이 여는 세계’라는, 컴퓨터의 기원부터 그 활용까지 다루는 강의를 듣고 있는 것을 우연히 알게 됐다. 지금이 오랫동안 간직했던 의문을 해결할 수 있는 기회라고 생각했던 동훈이는, 그렇게 건무에게 질문을 했다. ‘컴퓨터라는 건, 어떻게 작동하는 거야?’

<프롤로그>

동훈 : (건무에게 전화거는 중) 야, 지금 좀 잠깐 들어와봐

건무 : 응? 왜 갑자기? 뭐하려고?

동훈 : 뭐 좀 물어보게. 수업시간에 이해 안되는게 있었던 말야

건무 : 일단 알겠음 지금 들어갈게.

(비대면으로 줌에서 만난 건무와 동훈)

건무 : 그래서 물어볼게 뭔데?

동훈 : 아니, 너 그 우리 고등학교 때 같이 컴퓨터 동아리 했던 거 기억나지?

건무 : 응 당연히 기억나지?

동훈 : 그때 내가 맨날 투덜투덜거리지 않았었나?

건무 : (웃으면서) 맞아. 너 맨날 불만이 엄청 많았었지. 뭐라 그랬더라? 이게 왜 이렇게 작동하냐 뭐 그런 문제 아니었어?

동훈 : 맞아. 정확히 말하면 이제 우리가 그때 C++을 배웠었잖아?

건무 : 그치.

동훈 : 나는 C++을 그때 처음해봤는데, 뭐 물론 다른 컴퓨터 언어들도 경험이 없었고. 그래서 뭔가 컴퓨터가 어떻게 작동하는지, 언어들을 프로그래밍하는 건 어떤 원리인지에 대해서 알아볼거라고 기대하고 들어갔었는데, 막상 그런 건 없었지.

건무 : 아아 맞아 기억난다. 우리가 C++을 처음 배웠을 때 'printf'부터 시작했는데, 너가 처음에 바로 애들한테 왜 하필 'printf'는 이름이 'printf'냐고 물어봤었지?

동훈 : 맞아. 나는 그때 그런 것들에 대한 원리가 너무 궁금했어. 왜 하필 'printf'는 이름이 'printf'인지, 내가 'printf'라고 쓴 걸 이 컴퓨터가 어떻게 받아들여서 결과를 출력하는건지, 이런 것들이 너무 궁금했었지. 근데 애들이 다 정확히 모르더라고? 프로그래밍할 줄만 알지. 그래서 그때는 무작정 따라하기는 했었는데, 뭐 printf 나 scanf 같은거 있잖아. 그랬는데 되게 회의감이 많이 들었었지... 나는 이게 대학에 오면 좀 달라질 줄 알았다? 그리고 내가 또 전기과잖아. 프로그래밍을 1학년 때부터 들어야하는데, 오늘

python을 처음 배웠거든? 근데 또 그때랑 똑같이 하더라. 그래서 너한테 꾸념을 늘어놓고 있는 중 인거고.

건무 : (웃으면서) 아, 그럼 너 아직도 그것 때문에 답답했던거야?

동훈 : 당연하지.... 진짜 너무 답답함...

건무 : 이야 넌 운도 좋다. 내가 마침 학교에서 듣는 수업이 있는데!

동훈 : 응? 뭔데뭔데?

건무 : ‘컴퓨터과학이 여는 세계’라고, 딱 너가 궁금해 할 만한 것들을 알려주는 느낌인데? 컴퓨터의 기본 원리부터 시작해서 요즘에 어떻게 활용되고 있는지까지, 그러니까 컴퓨터에 대한 전반적인 걸 모두 알려주는 강의야.

동훈 : 와 부럽네... 아니 그럼 내 고민도 해결이 가능하고?

건무 : 응 그럴거 같은데? 한번 내가 배운 내용 토대로 이야기해줘봐?

동훈 : 와 그러면 나는 진짜 땡큐지. 미리 고맙다 나중에 밥한번 살게~~~~

건무 : 그러면 비싼거 먹어야지~~~~

<컴퓨터의 작동원리에 대해 - 튜링머신의 기계적 계산을 중심으로>

건무 : 일단 너가 궁금해하는 컴퓨터 언어들의 원리를 알아보기 전에, ‘컴퓨터의 작동 원리’를 알아보는게 맞겠지?

동훈 : 그렇지? 아무래도 컴퓨터 언어들은 결국 소프트웨어를 만드는 방법인거고, 컴퓨터의 작동 원리를 통해서 소프트웨어를 어떻게

실행시키는지만 알아도 내 의문은 풀리지 않을까?

건무 : 그러면 결국은 컴퓨터가 어떻게 만들어지게 되었는지, 그 역사를 따라 가 봐야겠네! 음, 사실 컴퓨터라는 건 두 방향으로 나누어져서 개발이 되었다가, 나중에 합쳐지게 되는? 그런 느낌이야. 한쪽에서는 컴퓨터의 개념, 그러니까 ‘만능 기계’의 원리 / 디자인에 대해 생각했고, 다른 한쪽에서는 이런 원리를 어떻게 실제로 구현 가능할지에 대해 생각했어. 이론과 실험의 구분이라고 생각할 수 있지. 우리는 우선 이론에 대해 이야기해보고, 그러한 이론이 어떻게 직접 실험적으로, 현실에 나타나게 되었는지 생각해보자.

동훈 : 그래, 좋아. 어차피 시간은 많으니까!

건무 : 그래그래. 그럼 이론에 대해 이야기해보려면, 이 사람이 빠질 수가 없어. 너 혹시 앨런 튜링에 대해서 들어본적이 있지?

동훈 : 당연하지, 그 사람을 왜 몰라! ‘이미테이션 게임’으로 많은 사람들이 알 것 같은데? 그 ‘애플’의 한 입 베어먹은 사과 로고도 그 사람에서 유래된 걸로 알고 있고. 요즘 핫한 인공지능 분야에서도 많은 업적을 세웠잖아! 튜링 테스트라던지 말이야.

건무 : 오, 잘 알고 있네? 그렇다면 ‘튜링 머신’이라는 것에 대해서도 들어본적이 있지?

동훈 : 음, 그건 못들어 본 것 같은데? ‘튜링이 개발한 기계’ 뭐 이런건가?

건무 : 아주 틀린 말은 아니지. 이 ‘튜링 머신’이라는 게, 컴퓨터라는 만능 도구의 탄생을 이끌어낸, 최초의 모터브라고 생각하면 돼.

동훈 : 오오? 튜링이 컴퓨터도 발명했던 말이야? 진짜 못하는게 없는 사람이었구나.

건무 : 정확히 말하면 튜링이 ‘발명’을 한 건 아니지만, 세상에 없

던, 중요한 개념을 제시했다고 봐야지. 그리고 웃기게도, 이 튜링 머신이란 건 ‘만능 도구를 발명하겠다!’라는 마인드에서 만든게 아니었어.

동훈 : 정말? 그럼 어찌다가 나오게 된건데?

건무 : 튜링은 케임브릿지 수학과에 다녔었어. 당연하게도 수학에 관심이 엄청 많았겠지? 그 당시에 수학기에는 증명되지 않은 난제들이 많았어. ‘페르마의 마지막 정리’와 같은 문제들 말이야. 그 중에 하나가 바로 ‘수리 명제 자동 판결 문제’였는데, 모든 명제들의 참/거짓을 기계적으로 판명할 수 있는지, 나아가 모든 참인 명제들을 밑바닥에서부터 기계적인 방법만을 이용하여 찾을 수 있는지에 관한 문제였어. 그리고 이 문제는, 너도 한번쯤 들어봤을 ‘괴델의 불완전성 정리’에 의해 증명이 되어버려. 참인 모든 명제를 만들 수 없다는 거였지.

동훈 : 근데 이미 증명한 명제랑 튜링하고는 무슨 관련이 있어?

건무 : 튜링이 이 명제를 증명하는 수업, 그러니까 괴델의 불완전성 정리를 설명하는 수업을 들으면서, 생각을 했어. ‘이 방식과는 다른, 아예 다른 방식으로 수리 명제 자동 판결 문제를 증명할 수 있을 것 같은데!’라고 말이야. 이 증명 과정에서 튜링이 생각한게, 바로 튜링 머신이야.

동훈 : 아, 이제야 너가 왜 갑자기 수학 이야기를 하는지 좀 이해가 되네. 근데 수학 증명에 기계가 쓰인다고? 그것도 좀 특이하네.

건무 : 그치? 이 기계는 너가 생각하는 것처럼 진짜 철 같은 금속으로 만들어진 그런 기계가 아니고, 그냥 가상의 개념이라고 생각하면 돼. 만능 도구 같은, 컴퓨터 같은거 말이야! 튜링은 이걸로 일종의 사고 실험을 진행한 거지.

동훈 : (웃으며) 그러네 그 기계가 아니었구나. 그럼 정확히 튜링 머신은 어떤걸 해주는 기계인거야?

건무 : 이제 그거에 대해 이야기해줄게. 뜬금없을 수도 있는데, 너는 ‘기계적’인게 뭐라고 생각해?

동훈 : 좀 뜬금없긴 하네. 한번도 진지하게 생각해본 적이 없는데? 음... 아마도 뭔가 생각없이, 두뇌의 일부분을 사용하지 않고 몸만으로 할 수 있는? 뭐 이런 느낌 아닐까?

건무 : 그렇지. 누구한테 물어봐도 그런 식으로 답변을 할거야 아마도. 그런데, 튜링은 이걸 좀 다르게 생각했어. ‘기계적’이라는 걸, ‘튜링 머신’으로 정의를 해버린거야. 그러니까, ‘튜링머신으로 실행할 수 있다’ = ‘기계적이다’라고 정의해버린거지. 그리고 이 정의를 이용해서 ‘기계적으로 참인 모든 명제들을 만들 수 없다’를 증명한 거고.

동훈 : 와... ‘기계적’이라는 걸 그렇게 정의해버릴 줄은 상상도 못했대 야. 어떻게 저런 생각을 한거지? 역시 비범한 사람들은 뭔가 대단하긴 하네! 그래서 이 수리 명제 자동 판결 문제 증명의 일부분에 튜링 머신의 개념이 도입된 거구나.

건무 : 맞아, 정확해. 우리는 튜링이 어떻게 수리 명제 자동 판결 문제를 증명했는지까지 알아야할 필요는 없으니까, 컴퓨터의 원천 설계도인 튜링 머신에 대해서 더 파고들어보자.

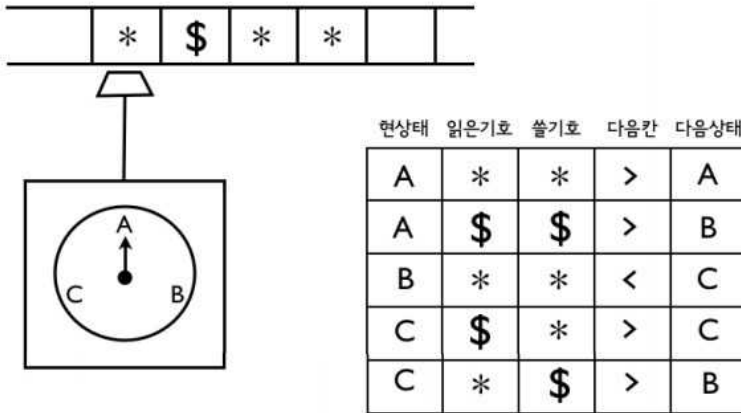
동훈 : 그래, 나도 그게 더 좋을 것 같아.

건무 : 사실 튜링 머신이라는 거는, 진~~~~짜 단순해. 별거 없어. 부품 몇 개로 계속해서 작업을 실행하는 거야. 이걸 기계적이라고 정의하는 느낌인거고. 부품들의 정확한 구성을 말해보면, 기본적으로 ‘십볼’이 있어. 약간 컴퓨터 언어에서의 변수 느낌이지. 이러한

심볼들을 적을 수 있는 테이프와, 테이프에 기록된 심볼들을 읽고 쓸 수 있는 장치, 마지막으로 테이프에 심볼을 어떻게 쓸지 정하는 규칙표가 구성의 끝이야.

동훈 : 전혀 안 와닿는군.

건무 : (웃으며) 너무 추상적이었나? 기다려봐 내가 화면공유로 보여줄게!



(출처 : ‘컴퓨터 과학이 여는 세계’)

건무 : 요거 보이지?

동훈 : 응 잘 보이네. 이거 사진 한 장만으로도 다 알겠는데? 음 그냥 규칙표대로 따라가는 거구만! 지금 상태는 A인데 *을 읽었으니까 *을 다시 쓸거고, 다음칸으로 저 화살표가 넘어가면서 상태는 A로 유지되겠네!

건무 : 역시 바로 이해하는구만? 정확해! 그럼 계속해서 규칙표대로 반복하면 어떻게 될까?

동훈 : 음, *써지고 두 번째 칸(상태는 A), \$써지고 세 번째 칸(상태는 B), *써지고 두 번째 칸(상태는 C), *써지고 세 번째 칸(상태는 C), \$써지고 네 번째 칸(상태는 B) ... 여기까지 했을 때 **\$*이고, 앞으로는 \$써지고 세 번째 칸 갔을 때랑 똑같은 규칙이 반복 되겠네. 그러니까 최종적으로는 \$가 한 칸 더 옮겨진, ***\$가 될 것 같은데?

건무 : 오오 맞아 정확해. 끝까지 다 해보지 않고도 규칙을 이용해서 파악했구나!

동훈 : 근데 약간 걸리는게, 이렇게 계속되면 마지막에는 네 번째 칸에 \$이 써진 다음에 아무 심볼도 없는 다섯 번째 칸으로 갈텐데, 그 뒤로는 규칙이 어떻게 적용되는거야?

건무 : 만약 더 이상 적용되는 규칙이 없다면, 그 기계는 멈추게 돼. 사실 이렇게 튜링기계가 멈추게 되는 성질도 튜링의 수리 명제 자동 판결 문제 증명에 쓰여.

동훈 : 그렇구나. 아니 이거 직접 내가 체험해보니까 계속해서 규칙표 읽어들이고 써놓고 하는게 진짜 컴퓨터 언어같네? 벌써 컴퓨터스러운걸? 그래서 이 기계를 컴퓨터의 원천 설계라고 하는 거구나. 컴퓨터가 작동하는 방식을 정의해버린 느낌이네.

건무 : 그러니까. 나도 들으면서 엄청 놀랐었어. 왜 진짜 이런걸 생각해보지 못했을까 했을 정도로.

동훈 : 좋아, 그럼 이제 튜링머신이 대충 어떤 건지는 감을 잡은 것 같아. 근데 이걸로 할 수 있는게 그렇게 많은 거야? 난 사실 아직 이런 예시만으로는 잘 모르겠어.

건무 : 음, 내가 강의를 들었을 땐 말야, 우리가 생각하는 ‘기계적’이라고 할 수 있는 것들은 전부 다 이걸로 할 수 있다고 교수님이

말씀해주셨긴 했어. 물론 엄밀한 증명?은 없다고 봐야할 것 같아.

동훈 : 음... 난 별로 마음에 들지는 않네...

건무 : 그럼 다른 예시를 한 번 들어볼까? 이런건 어때. 우리가 고딩 때 같이 했던 C++에서 가장 간단하고 기본적인 함수가 뭐라고 생각해?

동훈 : 아무래도 printf하고 scanf지?

건무 : 그치. 그러면 이런걸 생각해볼까? a를 받아들이면 "Hello World"를 출력하는거야.

동훈 : 아, 그러니까 지금 컴퓨터 언어하고 일대일대응을 한번 시켜보겠냐?

건무 : 그렇지. 아무리 너라도 컴퓨터 언어로 만드는 소프트웨어들은 기계적이라고 생각하지 않겠어?

동훈 : 음... 어느 정도는 납득할 것 같기는 하다. 컴퓨터가 곧 기계고, 소프트웨어는 컴퓨터가 할 수 있는 것들이니까.

건무 : 아, 참. 진짜 좋은 사이트가 있거든? 내가 튜링 머신의 규칙표랑 테이프 심볼 이런걸 다 직접 짜서 돌려볼 수 있는 사이트야. 일단 링크를 보내줄게.

<http://morphett.info/turing.html>

우리가 튜링머신을 짰 다음에, 이걸로 확인해보면 좋을 것 같아.

동훈 : 진짜 좋은 생각인데? 저 사이트 앞으로 애용해야겠다.

건무 : 그래. 그러면 a를 받아들이면 "Hello World"를 출력시키는 걸 한번 만들어보자.

동훈 : 근데 그럼 처음의 테이프는 우리가 아무렇게나 설정해도 되는건가?

건무 : 응, 아무런 상관없어.

동훈 : 아니 그럼 너무 쉬운거 아니야? 그냥 심볼에 a가 있으면 그 다음부터 상태가 한칸씩 변하면서 Hello World를 한 글자씩 출력하면 될 것 같은데.

건무 : 맞아 정확해. 엄청 쉽지? 그냥 상태만 몇 개를 설정하고, 테이프는 썩 다 빈칸이었다가 a가 입력되면 그 다음부터 빈칸을 한 글자씩 바꾸는 거지. 튜링머신을 설정할 때는, 1)초기 테이프, 2) 규칙표, 3)초기 상태, 이렇게 3가지만 설정해주면 돼. 너가 말한 것들로, 튜링머신을 설정해봐 그럼. 규칙표는 ‘현재 상태 / 현재 심볼 / 다음 심볼 / 헤더 방향(l : left, r : right, * : don't move) / 다음 상태’ 순으로 하면 좋을 것 같아. 이러면 바로 위의 사이트에 접속해서 제대로 했나 확인해볼 수 있거든.

동훈 : 그럼 일단 초기 테이프는 ‘_(빈칸) _ a _ _ ...’ 이렇게 설정을 하고, 규칙표를 설정해야겠네. 조금 귀찮긴하지만 상태 11개를 이용해서 표현해볼게. 그럼 이렇게 하면 될 거 같은데? 현 상태를 1이라고 한다면,

1 _ _ r 1
1 a _ r 2
2 _ h r 3
3 _ e r 4
4 _ l r 5
5 _ l r 6
6 _ o r 7
7 _ _ r 8
8 _ w r 9

9 _ o r 10

10 _ r r 11

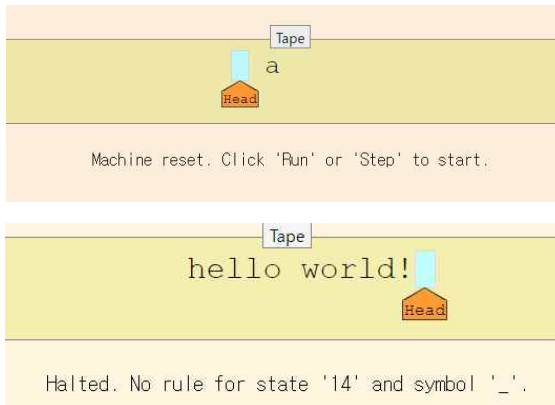
11 _ l r 12

12 _ d r 13

13 _ ! r 14

이렇게 하면 빈칸은 무시하다가 a를 받아들일 때부터 'hello world'를 출력시킬 수 있을 것 같네.

건무 : 오 정확히 잘했는걸? 실제로 이렇게 만들어 놓으면 저 위에 사이트에서 바로 확인해볼 수 있어! 이런 식으로 말이야.



동훈 : 뭐야 김빠진다. 너무 쉽잖아.

건무 : 이것뿐만 아니라 다양한 것들을 할 수 있다고 교수님께서 말씀해주셨어. 이번에는 진짜 scanf를 대응시키는 느낌인데, 어떤 심볼을 읽어들이 수 있는 '마커'를 만드는 거야. 그리고 우리가 컴퓨터로 입력하는건, 바로 튜링기계 위의 테이프가 되는거지. 우리가 테이프에 입력하는 걸 이 머신이 읽고, 뭔가를 할 수 있게 만드는

거야.

동훈 : 그렇게 진짜 가능할까? 말로만 들어서는 잘 모르겠는데.

건무 : 당연히 가능하지! 일단 이렇게 생각하는 거야. 테이프를 두 칸씩 다 묶고, 앞 칸에는 우리가 쓰고자하는 심볼, 뒷 칸에는 마커 역할을 하는 심볼, 뒤 예를 들어 #을 넣는 거야. 만약 'hello world'를 읽고 싶으면 'h#e_l_l_o__w_o_r_l_d_'가 초기 테이프 상태가 되는 거지. 그 뒤에는 이 마커가 하나의 심볼로 움직이면서 'hello world'를 읽어들이 수 있을 거야!

동훈 : 근데 그건 일종의 라이브러리 느낌 아니야? 그냥 마커를 '읽어들이는 것'이라고 막 정의해도 되는거야?

건무 : 강의에서는 교수님이 마커를 정의하시고, 바로 규칙표에 마커 칸을 새로 만드는 느낌으로 하셨는데.... 흠.... 니 말도 맞네. 마커가 과연 오리지널 규칙표로 나타낼 수 있는지에 대해선 생각해보는 것 없는 것 같아. 그럼 우리가 같이 라이브러리 자체를 만들어보는 것도 괜찮을 것 같은데? 이 마커에 관한 걸 라이브러리로 만드는 거야. 마커가 읽는 행위를, 가장 low level에서의 규칙표로는 어떻게 나타낼지를 생각해보는거지. 이정도면 만족하지?

동훈 : 좋아. 한번 해보자.

건무 : 음, 그러면 일단은 마커 라이브러리를 이용한 규칙을 한번 생각해보고, 이런 라이브러리를 어떻게 생각해낼지 고민해보는게 좋을 것 같다. 이런 예시를 들어볼게. 초기 테이프가 10101_____일 때, 빈칸에 앞의 패턴을 그대로 복사하는 튜링머신을 생각해보자. 오리지널 튜링머신은 헤더만 계속 움직이지만, 여기서는 마커까지 움직이게 해버리는 거야. 규칙표의 순서도, '현재 상태 / 현재 심볼 / 다음 심볼 / 헤더 방향(l : left, r : right, * : don't move) / 다

음 상태'에서, 뒤에 '현재 마커가 읽는 심볼 / 마커 이동'을 추가하자. 현재 상태를 a라고 하면, 헤더는 일단 빈칸까지 움직여야하니까 이런 규칙이 필요해.

a 1 1 r a 상관없음 *

a 0 0 r a 상관없음 *

이 두 규칙을 시행하고 나면, 일단 마커는 빈칸 앞에 가있을 거야. 첫 번째 빈칸은 복사 끝을 알리기 위해 다른 심볼(\$)을 하나 넣어 줘야해.

a _ \$ r a 1 상관없음 *

그 후에는 다음 단계가 필요하지. 복사가 가능한 상태를 b라고 하자. 그러면 헤더가 빈칸을 인식할 때 b가 되어야하니까

a 1 _ _ * b 상관없음 *

가 필요해. 그 다음은 이제 마커로 읽고, 헤더로 쓰는거야. 1을 읽으면 1을 쓰고 헤더와 마커 모두 한칸 옮기고, 0을 읽으면 0을 쓰고 마찬가지로. 그리고, \$를 마커가 읽으면 끝나도록!

b _ 1 r b 1 r

b _ 0 r b 0 r

이런 식으로 규칙 몇 개만으로도 마커가 있으면 저런 복사를 바로 표현할 수 있어.

동훈 : 맞네. 그러면 이런 마커 라이브러리를 low level로 표현해보는 건 내가 해볼게. 테이프의 모양부터 벌써 달라지겠네? 1#0_1_0_1_____로 말이야. 이 테이프를 그대로 복사해볼게 한번. 마커로 숫자를 '읽는다'라는 것부터 생각을 해봐야할 것 같은데? 마커를 숫자로 읽는다는 건, 결국 #이라는 심볼을 헤더가 읽으면 그 왼쪽으로 헤더가 이동하고 다시 심볼을 읽어서 특정한 심볼

을 쓸 수 있는 상태로 바뀌어야한다는 말과 똑같다고 생각해. 그렇게 바뀐 상태는, 쪽 다른 심볼들을 무시하다가, 헤더가 원래 있던 자리에 가서 읽은 숫자를 복사하는 거지.

건무 : 뭐, 그건 그렇다고 쳐. 그러면 처음에 헤더가 두 칸씩 띄워져서 가는건 어떻게 해결할거야?

동훈 : 그건 이렇게 해결할 수 있을 것 같은데? 빈칸이 아닌 심볼들 있잖아. 0이나 1같은거. 헤더의 원래 상태가 a였다하면, 0이나 1을 읽으면 상태를 b로 만드는 거야. 이 상태는 빈칸을 건너뛴 수 있는 상태인거지. 그리고 빈칸을 b 상태로 읽으면, 다시 a상태로 돌아가서 다른 심볼들을 지나갈 수 있도록 하는거야. 이렇게하면 헤더가 쪽~이동하고, b 상태에서 숫자가 아닌 빈칸을 인식할 때 복사 상태로 돌입하도록 하면되는 거지.

건무 : 계속 듣고 있으니깐 우리가 직접 라이브러리를 만드려면 약간 '상태'에 대한 정의가 되게 중요할 것 같아. 복사할 수 있는 상태라던지, 이동하는 상태라던지 이런 것 말야. 단순히 a, b, c, ...로 의미없이 하는게 아니라. 저기 너가 세운 a, b상태는 이동과 관련되고, c 상태는 복사와 관련된 상태가 되는거지 그럼. 그리고 참고로 상태, 심볼 같은건 단순 a, b 같은 문자뿐만 아니라 글자 + 숫자 조합같은 것도 충분히 가능해. 'move1' 같은거 말이야.

동훈 : 옹 그렇구나. 그렇게 의미를 부여하고 일반화하면 라이브러리를 바로 만들 수 있을 것 같은데? 일단 라이브러리 일반화는 생각 안하고, 저 복사를 해보는 코드를 짜볼게. 초기 테이프 상태는 아까 말했듯이 1#0_1_0_1_____이고, 초기 상태는 move1이야.

```
move1 1 1 r move2
```

```
move1 0 0 r move2
```

; 각 첫 번째 심볼 쓰여있는 칸에서 움직이기

```
move2 # # r move1
```

```
move2 _ _ r move1
```

; 각 두 번째 빈칸 or 마커 들어있는 칸에서 움직이기. 여기까지
거치면 헤더가 복사를 하게될 빈칸까지 움직임

```
move1 _ @ l copy1
```

```
copy1 _ _ l copy1
```

```
copy1 1 1 l copy1
```

```
copy1 0 0 l copy1
```

```
copy1 # # l copy2
```

; 헤더가 다시 돌아올 자리를 지정하기 위해 새로운 심볼(@)을
도입하고, copy1 상태에 돌입. 그리고 이 copy1 상태에서는 #을
찾을때까지 쪽~오면 됨. 이렇게 #을 찾은 헤더는 copy2 상태로
들어가서, 옆 칸에 있는 숫자를 인식하도록 하면 됨.

```
copy2 1 1 r 1
```

```
copy2 0 0 r 0
```

; 상태 n --> n을 복사하도록 한다!

```
1 # _ r 1-1
```

```
1-1 1 1 r 1-1
```

1-1 0 0 r 1-1

1-1 _ # r 1

; 이제 상태 1에서 #을 두 칸 옮기도록!

0 # _ r 0-1

0-1 1 1 r 0-1

0-1 0 0 r 0-1

0-1 _ # r 0

; 마찬가지로 상태 0에서 #을 한 칸 옆으로 옮긴다.

1 1 1 r 1

1 0 0 r 1

1 _ _ r 1

1 @ 1 r move2

0 1 1 r 0

0 0 0 r 0

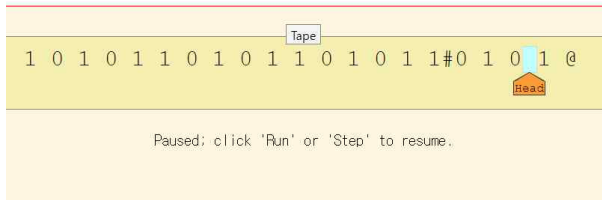
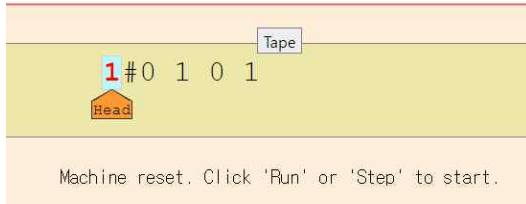
0 _ _ r 0

0 @ 0 r move2

; 마지막으로 상태 n에서 @를 찾을때까지 계속 움직이고, @를 n으로 바꾸면 하나를 복사한 것이 된다! 그 다음은 다음 칸에서 복사를 반복해야하므로 마치 하나의 숫자를 지난 것처럼 move2 상태로 보낸다.

동훈 : 이런 식으로 코드를 짜봤는데, 어때? ‘;’ 뒤에는 주석을 달아서 각각의 상태가 뭘 의미하는지를 써봤어.

건무 : 오 너무 좋은데? 그럼 프로그램에 한 번 돌려보자.



건무 : 이거... 멈추지를 않는데? 어찌냐 요거?

동훈 : 그러네...? 복사는 확실히 잘되는데, 언제 끝나는지에 대한 설정은 생각해보니까 안한 것 같다. 흠 그럼 어떻게 해야할까..

건무 : 그러면 이렇게 해야지 뭐. 시작과 끝이 정해져있지 않다는 게 지금의 문제니까, 맨 처음의 1 0 1 0 1 바로 다음 칸에 ! 같은 다른 심볼을 하나 박고, 마커가 그걸 읽으면 끝내는 거야. 그럼 딱 한번밖에 복사를 못하는 거지. 그럼 코드를 좀 변경해야할 것 같은데..?

```

move1 _ @ l copy1 --> move1 _ ! r move4
      move4 _ _ r move3
      move3 _ @ l copy1

```

; 처음에 갈 때 ! 하나 찍고 @를 설정하도록.

```
move3 1 1 r move4
move3 0 0 r move4
move4 # # r move3
move4 _ _ r move3
```

; move3과 move4가 자유롭게 움직일 수 있도록.

```
copy1 !!! copy1
1 !!! r 1
0 !!! r 0
1-1 !!! r 1-1
```

; 새로 추가된 심볼 !에 대해서도 헤더가 자유롭게 움직일 수 있도록.

```
1 @ 1 r move2 -> 1 @ 1 r move4
0 @ 0 r move2 -> 0 @ 0 r move4
```

; 그리고 이제 마지막으로 복사 끝난 후 다시 move 상태로 올 때를 move4로 바꿔주기만 하면 된다.

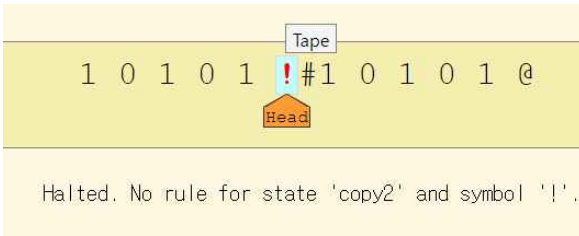
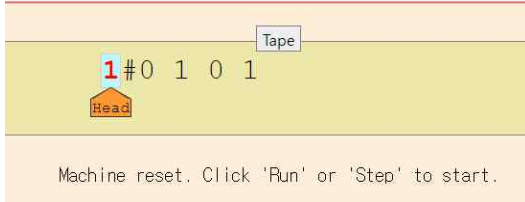
그냥 move1, move2를 일회용(한번만 쪽 갈 때)으로 쓰고 move3, move4라는 새로운 상태를 추가하는 거지. 역할은 move1, move2랑 똑같은데 !를 지나갈 수 있는 상태말이야.

동훈 : 그렇게되면 그냥 move1, move2를 !에서 자유롭게 움직일 수 있도록 하면 되는거 아냐?

건무 : 아니지. move1이랑 move2는 이제 빈칸에 닿으면 !가 되니

까 더 이상은 쓸 수 없어.

동훈 : 아 그러네. 그러면 이렇게 프로그램을 돌려볼까? 이번엔 내가 돌려볼게.



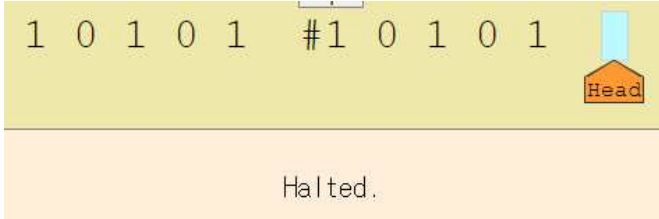
동훈 : 오? 이제 진짜 딱 한번만 복사가 되네? 그리고 마커가 !옆에 있으니까 이러면 마커가 #을 읽는 순간 끝나는 것도 맞고.

건무 : 그치. 프로그램 잘 짰거같은데?

동훈 : 한가지 걸리는게, 저 우리가 새로 도입한 !하고 @ 심볼도 없애는게 좋을 것 같아. 코드 몇 개 만 더 추가하면 될 것 같은데? finish라는 새로운 상태를 도입해서 말이야.

```
copy2 ! _ r finish
finish 1 1 r finish
finish # # r finish
finish _ _ r finish
finish 0 0 r finish
```

동훈 : 이러면 진짜 진짜 완성이다. 한번 더 돌려볼게!



건무 : 오ㅋㅋㅋㅋㅋ 확실히 우리가 의도한 대로 되었네! 이제 우리는 복사에 대해서 완벽한 하나의 프로그램을 짤거야. 두 번, 세 번 복사하고 싶다? 그럼 이 복사를 반복하면 돼 이제!

동훈 : 확실히 진짜 프로그래밍 하는 기분이긴 하다. 근데 너두 코드가 긴 것 같은 느낌은 지울 수가 없네...

건무 : 맞아. 너무 너무 길어. 만약 c++로 했다면 scanf 다섯 번하고 print 다섯 번만 있으면 바로 했을 텐데 말야. 그래서 이 튜링머신이라는게 굉장히 low level이라는 거야. 이렇게 모여서 라이브러리를 만들고, 우리가 위에서 '마커'라는 걸 새롭게 도입하면 거의 30줄이 넘는걸 7줄로 줄일 수 있는 것처럼 압축시킬 수 있는 거겠지. 그리고 사실 가장 low한, 우리가 방금 푼 상황에 대해서 우리가 한 방법보다 훨씬 효율적인 방법들이 있을 수도 있어. 나중에 이렇게 또 '비용 문제'와 같은 다른 문제들을 발생시키는 거지. 우리가 문제를 푸는 건 둘째치고, 이게 정말 현실적으로 짧은 시간 안에 풀 수 있는지도 생각을 해야하니까 말이야. 이런 거에 관심있으면 p-np문제나 빅 O 표기법 같은걸 찾아봐. 컴공에서 맨날하는 게 이거지. 최적화! 근데 우리는 이런 것보다는 컴퓨터의 기본 원

리에 집중하고 있으니까 오늘은 넘어가자.

동훈 : 좋아. 이제 두 가지 버전에서 다 해봤으니까, 우리만의 그 마커 라이브러리를 만들 수 있지 않을까?

건무 : 그치? 일단 난 가장 중요한 건, 라이브러리의 역할을 명확히 정의하는 거라고 생각해. 무슨 말이나면, 마커의 역할을 명확히 정의한다는 거야. ‘마커로 읽는다’는 행위는, 결국 ‘마커 옆의 숫자를 헤더가 읽는다’는 행위와 완전히 같다는 걸 알아야해. 이런 것들을 정의해보면, 자연스럽게 마커 라이브러리를 만들 수 있을 것 같은데?

동훈 : 오 좋네. 그러면 구체적인 코드를 하나 보면서, 어떻게 일반적으로 확장시킬지도 생각해보면서 새로운 규칙을 만들면 될 것 같아.

b _ 1 r b 1 r

이 코드로 한번 해보자.

건무 : 좋아. 일단 ‘일반성’을 띠기 위해서는, 변하는 모든 수마다 새로운 상태들을 만들면 할 수 있을 것 같거든? 그렇게 한번 해볼게. 위 규칙에서는 변하는 것들이 ‘현재 상태 / 현재 심볼 / 다음 심볼 / 헤더 방향 / 다음 상태 / 마커 읽기 / 마커 이동’이니까 상태는 거의 7개가 하나의 규칙마다 생겨야하는 느낌이네. 일단 ‘b _ 1 r b 1 r’ 규칙으로 예를 들고 있으니까, 현재 상태가 b라고 해보자. b인 상태에서 헤더가 마커가 있는 쪽까지 갔다가 다시 오려면, 일단 현재 읽은 빈칸을 _에서 \$같은, 안접치는 심볼로 바꿔야할거야. 어디까지 와야하는지를 알아야하니까. 그리고 바꾼 다음에는 다른 심볼들은 전부 무시하고 오직 #만 찾아서 가야하는 상태로 돌입해야해. 그런데 이 상태 후에 # 옆에 뭐가 있는지에 따라 또 이

후 상태들이 달라져야하니까, 다른거 다 무시하고 가는, #만 찾아서 가는 상태에서 우리는 분화를 시작해야해. 그리고 이걸 헤더가 읽은 값과, 현재 상태에 대해 달라져야하니까, 나는 'b-move_'으로 상태를 명명할게. 여기서 b하고, _은 읽은 값과 현 상태에 대해서 계속 달라지는 거지!

이 'b-move_' 상태는 다른 것들은 다 무시하고 #만 찾아가게 규칙을 세우는거야. 이제 그것쯤은 할 수 있겠지? 그리고 #에 도달하면, 상태가 또 바뀌어야만해. 'b-move_'상태에서는 0, 1과 같은 다른 심볼들은 무시했으니까 말야. 이 상태를 'b-move_*'이라고 할게. 이 상태에서 # 옆에 있는 심볼을 읽는 순간, 또 그 심볼에 따라서 규칙이 달라져야해. 읽어들인 심볼에 따라서 뭐가 써질지도 달라지니깐 말이야! 이 규칙에서는 1을 쓰니까, 상태를 'b-move_-copy1'으로 바꿀게. 남은 건 2개뿐이야. 마커를 이동하고, 헤더로 돌아와서 다음 심볼을 쓰면 돼. 어차피 헤더 방향과 다음 상태는 마지막에 다음 심볼을 씌고 동시에 설정하면 되니까 고려할 필요가 없어.

여기서 마커 이동은 3가지 경우밖에 없으니, 각각의 경우에 따라 상황을 설정해주면 될 것같은데? 마커를 오른쪽으로 옮긴다면 'b-move_-copy1-markermoveR'이라는 상태를 만들고(마치 1-1 상태처럼), 오른쪽 가다가 빈칸 닿으면 #을 쓰고 그 다음엔 다시 'b-move_-copy1' 상태를 만드는 거지. 이제 끝났어. 이러면 'b-move_-copy1'상태에서 다른 심볼들은 다 무시하다가 \$를 만났을 때 '다음 심볼'에 해당하는 심볼을 적고, 헤더 방향과 다음 상태를 설정해주면 끝나!

동훈 : 생각보다 쉬운걸? 그니까 이 라이브러리는 결국에

'b-move_copy1-markermoveR'이 4개의 상태 가지만 설정해주면 되는 거구나. 'b, _, 1, R' 이 4가지에 대한 상태와 그에 대한 규칙들만 조금 바꾸면 되는거네.

건무 : 맞아 정확해. 하다보니까 하나를 발견했어. 라이브러리를 만들 때, 우리가 규칙을 확인하는 순서 그대로 가면 되네. 규칙을 볼 때 '현재 상태 -> 현재 심볼 -> 마커 읽기 -> 마커 이동 -> 다음 심볼 -> 헤더 방향 -> 다음상태' 이런 순서로 우리가 읽으니까 말야. 라이브러리도 이 순서대로 짜주면 되는거지! 이러면 다른 라이브러리들도 다 짤 수 있겠는 걸?

동훈 : 그러네. 어쨌든 상태를 남발해서 결국 어떻게든 만들수는 있는거구나. 그런 상태를 어떻게 더 잘 정리하고 이해하기 쉽게 써놓냐의 문제 같네. 변수에 따라서 상태만 계속 바꿔주면 되는 느낌이야!

건무 : 좋아. 그러면 이 마커 라이브러리를 진짜 총 정리해보자. 예시 자체는 아까의 예시인 'b _ 1 r b 1 r'로 들게. 즉, 이 하나의 코드하고 완전 똑같은 코드를 만들고 있는 거야. 다만, 변할 수 있는 부분은 밑줄 표시로 알아보기 쉽게 해놓을게!

```
b _ $ 1 b-move_  
; 헤더가 돌아올 위치 설정  
  
b-move_ _ _ 1 b-move_  
b-move_ 1 1 1 b-move_  
b-move_ 0 0 1 b-move_  
b-move_ # # 1 b-move_*
```

; #빼고 다른 건 다 무시

```
b-move_* 1 1 r b-move_copy1  
b-move_copy1 # _ r b-move_copy1-markermoveR  
; 마커 옮길 준비 및 입력받은 값에 따라 달라지기
```

```
b-move_copy1-markermoveR (1 or 0) (1 or 0) r  
b-move_copy1-markermoveR  
b-move_copy1-markermoveR _ # r b-move_copy1  
; 마커 옮기기(오른쪽으로 옮기는 경우)
```

```
b-move_copy1 (1 or 0 or _) (1 or 0 or _) r  
b-move_copy1  
b-move_copy1 $ 1 r b  
; copy 마무리
```

괄호로 묶어놓은 건 복부호동순 비슷한 역할이라고 생각하면 될 것 같아. 전부 쓰기 귀찮아서 그냥 저렇게 쓴거야. 위 코드에서, 밑줄 친 부분만 바뀌면 코드를 완전히 똑같이 바꿀 수 있어. 진짜 ‘라이브러리’가 되어버린 거지! 예를들어 ‘b _ 0 r b 0 r’ 이런 규칙을 하고 싶다면, 4, 5번째 밑줄인 1과 copy1, 그리고 마무리 1 r b만을 건들어주면 되는거야. 그리고 위 코드는 너무 기존과 비슷해서 별로 티가 나지 않기도 하고.

동훈 : 와우. 이게 되긴 되네! 뿌듯하다.

건무 : 그치. 결국, 우린 이걸 해보면서 튜링머신을 어떻게 ‘확장’

할 수 있는지에 대해 알아본 것 같은데? 어떻게 생각해?

동훈 : 맞아. 나도 그렇게 생각해. 솔직히 아직까지도 튜링머신이 어떻게 모든 기계적인 일들을 할 수 있다는 건지는 ‘증명’은 할 수 없으니까 완벽히 이해가 되지는 않거든? 근데 뭔가 이렇게 점점 계층을 쌓아가는게 컴퓨터 언어의 어떤 기본 원리라는 거는 알게된 것 같아. 그리고 이렇게 확장해보면서 튜링머신으로 실제로 엄청나게 많은 것들을 할 수 있다는 건 알 것 같네.

건무 : 정확해. 컴퓨터 언어도 ‘기계어’라는 기본 언어에서부터 이렇게 하나씩 하나씩 대응시켜서 ‘번역’을 하는 거야. 나중에 관심있으면 그것도 찾아보면 좋을 것 같아.

동훈 : 좋아! 난 이정도면 만족한다! 아직 모르는 부분들은 많지만, 그래도 기본은 된거같은 느낌? 이제는 내가 혼자 공부해볼게! 무지한 날 가르치느라 고생 많았다 ... 맛있는거 사줄테니까 시험끝나고 연락해!

건무 : ㅋㅋㅋㅋㅋ오케이! 바로 연락해야징~~

<에필로그>

(빔스에 간 동훈과 건무)

동훈 : 아니 그렇다고 빔스를 사달라고 하냐?;;;

건무 : ㅎㅎㅎ 고생한게 있는데 그럼 비싼거 먹어야지!

동훈 : 에잉... 내가 배운게 있으니까 사주긴 한다. 쯤든 내가 너랑 좀 얘기해본 다음에 그 '컴퓨터 과학이 여는 세계'라는 책을 사서 읽어봤거든?

건무 : 영 근데? 괜찮지? 너가 궁금했던 내용 좀 많이 나오지 않

아?

동훈 : 응 그렇더라고. 근데 거기서도 또 의문이 생기는 내용들이
좀 있었어ㅋㅋㅋㅋ 이번엔 회로다! 긴장하라구~~

건무 : ,,!? 어쩐지 밍스를 순순히 사준다 했다...

#시놉시스

무한한 규칙표에 의해 통제되어 온 25 년 동안의 삶. 세주는 알지 못한다. 평소 기계 같은 성격이라는 말을 들었지만, 대수롭지 않게 여겼다. 본인도 자신이 이성적이고 계산적인 성격을 가졌다고 생각했기 때문이다. “결정장애”, “회색분자”와 같은 말은 이해하지 못했다. 모두가 자신의 직업을 고민할 때, 세주는 생명공학 연구원이 되기로 결정했다.

여느 때와 같이 정시에 퇴근하던 세주는 어떤 사람을 보게 된다. 술 또는 약에 취한 듯 비틀거리던 그를 부축하려 했던 세주는 그에게서 알약 한 통을 받는다. 약 냄새를 맡자 세주는 인생에서 한 번도 느끼지 못한 혼란을 겪는다. 일상의 규칙이 전부 깨지는 듯한 느낌. 처음 느껴보는 삶의 무게.

그는 자신의 유일한 인간관계인 애인에게 겪은 일을 얘기하는데, 그는 낯선 사람처럼 허탈하고 분노 섞인 목소리로 이해할 수 없는 말을 건넨다. “넌 기계야. 기계적인 사람이지. 사람 같은 기계이고, 기계가 기계적이지 않을 수는 없어.” 이해되지 않는 말들이었지만, 그 상황을 벗어나니 너무나 계획대로 흘러간 자신의 삶을 의심하게 된다. 다음 날, 회사에서 자신을 잡으려 하는 경비원들로부터 도망친다. 그때 진실을 알고 있는 누군가 등장하고 세주를 도와주는데... 더 큰 진실을 알기 위해서는 회사에 다시 들어가야만 한다. 규칙표에 고려되지 않은 상태를 마주한 세주. 그의 테이프에 그려질 다음 심벌은 무엇이 될까?

#1 연구실

벽에 걸린 시계가 밤 11시를 가리킨다. 시계를 응시한 후 세주는 쓰던 기구를 정리하고 연구실에서 짐을 챙긴다. 창밖은 추운 겨울.

#2 회사 앞

“IDEA(이데아)”라는 큰 간판이 붙어있는 높은 건물 앞. [IDEA, 독보적인 뇌 과학 기술로 알츠하이머 치료에 큰 기여.]와 [IDEA, 집요하고 섬세한 연구로 ADHD 치료제 개발.] 라고 쓰인 플래카드도 눈에 띈다. 세주는 추위에 몸을 웅크리며 계단을 내려온다. 그때, 허름한 옷차림의 누군가가 비틀비틀 걸다가 쓰러진다.

세주: (경계하지만 도와줘야 함을 아는 듯이 다가가며, 낮은 목소리로) 저기요, 저기 괜찮으세요..?

낯선 이(호윤):(쓰러진 상태로 앓는 소리를 낸다)

세주: (놀랐지만 짐작한 목소리로) 저기요, 119 부를까요?

낯선 이(호윤):(천천히 일어나며, 슬프고 반짝이는 눈으로 세주를 쳐다보며) 감사합니다..

호윤이 낡은 가방에서 술병을 꺼내 마시려다 떨어뜨린다. 술병이 깨진다.

세주: (이상한 향을 맡고 극심한 두통을 느끼며 머리를 부여잡는다.) 으

낯선 이(호윤): (약통을 건네주며, 작지만 비장한 목소리로) 알고 싶으면 이 약을 먹어. 알아야 돼. 넌 알아야 돼.

세주: (머리를 부여잡으며, 혼란스러운 표정)

세주는 얇게 쌓인 눈길 위에 쓰러진다. 흐릿하게 보이는 낯선 이(호윤)의 뒷모습.

#3 세주의 꿈-어릴 적 기억들

세주는 무의식 상태에서 어릴 적 기억을 떠올린다. 텔레비전이 지지직하는 장면, 아이들이 뛰어다니는 장면, 교복점에서 옷을 맞추는 장면 등이 빠르게 지나가다가 고등학교 때로 멈춘다. 자기 꿈을 발표를 하는 장면에서 멈춘다.

학생 1: (당차게) 안녕하세요, 저는 의사가 되고 싶은 임승주입니다! (학생들이 박수친다) 제가 의사가 되고 싶은 이유는.....

학생 2: (수줍어 하며)안녕하세요, 저는 신푸름이고, 선생님이 되고 싶습니다. 제가 초등학교 때 도움을 많이 주셨던 선생님이 계신데.....

어린 세주: (차분하게) 안녕하세요, 강세주입니다. 저는 뇌과학 연구원이 되고 싶습니다. (정적)

선생님: 음 세주는 뇌과학? 연구원이 되고 싶은 이유가 있을까?

어린 세주: (멍하니 바닥을 보다가) 음... 잘 모르겠어요... 하지만 나중에 제가 뇌과학 연구원을 하고 있을 것 같습니다. 뇌과학은 현재 많은 분야 중에서도.....

#4 다음날, 승현의 집

세주는 천천히 눈을 뜬다. 익숙한 듯 낯선 승현의 방 풍경을 둘러본다. 애인 승현이 걱정스러운 눈빛으로 쳐다본다.

승현: (걱정 가득한 목소리로) 세주야, 괜찮아?

세주: (낯선 사람을 바라보는 눈빛으로) 승현아, 너가 어떻게 여기 있어?

승현: (약간 놀랐지만 티 나지 않는 표정으로) 뭘 그렇게 경계해. 어제 야근하고 퇴근하는데 너 길에 쓰러져 있었어. 병원에 가려고 했는데, 네가 우리집으로 가자고 해서 여기로 온 거야. 기억 안 나?

세주: (혼란스럽고 지친 표정으로) 아.. 그랬나, 기억이 잘 안 나네.

승현: 우리 만나면서 너 이렇게 쓰러진 적 종종 있었으니까.. (세주를 안으며) 건강이 제일 중요한 거 알지?

세주: (얇은 미소)

승현이 세주 곁에서 떨어져 일어난다.

승현: 뭐 먹을까?

세주: (갑자기 극심한 두통을 느끼며) 으. 머리가 너무 아파.

승현: 어? 괜찮아? 약 먹을까?

세주: (얼굴을 찌푸리며) 모르겠어.. 뭔가 다 이상해.

승현: (표정이 약간 굳으며) 다 이상하다고?

세주: 응.. 지금까지 나는 고민을 해본 적이 없는데, 의심도 해본 적이 없는데, 지금 내가 보고 있는 게 다 처음 보는 것 같아. 너도 이상하게 느껴지고.

승현: (차갑고 썩한 표정으로) 세주야, 너 무슨 일 있었어?

세주: (놀라 굳은 표정으로) 아니. 아무 일 없었어.

승현: (어이없다는 듯이) 왜 거짓말을 해. 아무 일 없었는데 네가 지금 이렇게 됐다고?

세주: (약간 겁먹은 표정으로) 내가 지금 어떻게 됐는데..?

승현: (화가 난 듯이) 고민을 한다며, 고민을! 나를 의심하고, 세상을 의심하기 시작했잖아. 너 누구 만났어? 누구 만났냐고!

세주: (놀란 표정으로) 너 미쳤어? 왜 갑자기 소리를 질러.

승현: (숨을 내쉬고 배신당했다는 듯 쳐다보며) 넌 기계야. 기계적인 사람이야. 사람 같은 기계이고, 기계가 기계적이지 않을 수는 없어.

세주는 곧바로 짐을 챙겨 승현의 집을 나간다. 나가는 도중 떨어진 알약 통. 세주는 망설이듯 알약 통을 집어 주머니에 넣는다. 누군가에게 통화를

거는 듯한 승현의 뒷모습.

#5 세주의 집 앞

세주가 택시에서 내려 집으로 빠른 걸음으로 들어가는 도중 익숙한 사람을 만난다.

호윤: 안녕하세요, 강세주씨.

세주: (놀랐지만, 기력 없는 모습으로 쳐다보며) 누구세요?

호윤: 저는 진호윤입니다. 세주씨가 지금 일하고 있는 IDEA(이데아)를 만든 사람입니다. 아, 그러니까 지금 대표인 정군태와 같이 만들었어요. 혹시 시간 되시면 이야기 좀 나눌 수 있을까요?

세주: (피곤한 기색이 역력한 표정으로) 저를 왜 찾아오셨는지 모르겠지만, 제가 지금 많이 피곤해서요. 죄송합니다.

호윤: (난처한 표정으로 미소를 지으며) 아.. 그럼 여기 제 연락처입니다. 편하신 시간에 다시 연락 주세요.

멀어져가는 호윤의 뒷모습. 기시감을 느낀 세주의 표정.

#6 다음날, 회사.

세주는 IDEA(이데아)에 출근한다. 로비에서 경비원들이 세주를 예의주시한다. 세주, 피곤한 모습으로 생각에 몰두하고 엘리베이터에 탄다. 39층까지의 버튼 중 14층을 누르고 기다린다. 엘리베이터가 멈추고 들어오는 Security 복장의 경비원들. 경비원들은 세주가 14층에서 내리는 것을 막는다. 세주가 위험에 처하자 호윤이 주사기로 경비원들을 잠들게 한다. 곧장 다른 경비원들이 세주와 호윤을 쫓아온다.

경비원1: 잡아!

호윤의 주도로 둘은 함께 도망간다. 막다른 길을 맞이한다.

세주: 이쪽으로! (사원증으로 다른 통로로 호윤을 안내한다.)

비상계단으로 주차장에 도달한 호윤과 세주. 주차장에 들어서자 세주는 다리에 힘이 풀려 주저앉는다. 두통에 몸부림치는 세주를 업어 차에 태우는 호윤.

#7 차 안

호윤: 이제 시간이 얼마 없어. 예전으로 돌아가기 싫으면 약 먹어.

흔들리는 세주의 눈동자. 불안한 표정으로 약을 서둘러 삼키고 두통을 느끼고 스르르 눈을 감는다. 쫓아오는 경비원들. 차는 무사히 주차장을 빠져 나간다.

#8 “IDEA” 39층 대표 사무실

후드타를 입고, 까치집 모양의 머리를 한 사람이 컴퓨터를 두들기고 있다. 그때 울리는 벨소리.

균태: 네~ 여 보 세 요 !

전화기 너머의 목소리: 죄송합니다. 놓쳤습니다.

균태: (장난기 어린 목소리로) 예? 뭐라구요?

전화기 너머의 목소리: 그.. 강세주씨를 놓쳤습니다. 어떤 사람이 갑자기 나타나서...

균태: 아, 음, 듣기 싫은 소리라 그런가 잘 안 들리네요! 수고 많으셨어요!

균태가 전화를 끊는다.

균태: (의자 뒤로 한없이 몸을 눕히고, 초점 없는 눈으로) 아! 세주야. 왜 그래.

#9 외진 곳에 위치한 호운의 집

호운: 여기는 안전하니까 걱정 마.

세주: (두통을 느끼고 얼굴을 찌푸린다) 도대체 이게 다 무슨 일이에요? 당신은 알고 있는 거죠?

호운: (엄지손가락으로 검지손가락을 문지르며 고민한다)

세주: 저번에 회사 앞에서도 당신이 이상한 약 준 거 맞죠? 나한테 왜 이러는 거예요?

호운: (차분하게) 내가 IDEA(이데아)를 만든 사람이라고 얘기했었지? 균태랑 같이.

세주: (고개를 끄덕인다)

호운: 그 회사를 세우기 전에 균태랑 내가 만든 게 바로 너야.

세주: (두통을 느껴 얼굴을 찌푸리고 어이없다는 듯 실소를 터뜨리며)
그게 무슨 헛소리야... 말도 안 되는 소리 하지 마세요.

호윤: 그러니까, 지금 네 뇌에는 칩이 하나 심어져 있어. 네가 8살 때 이식된 거야.

세주: 내 뇌에 무슨 칩이 있다는 거야..

호윤: 네가 지금 연구하고 있는 분야도 그쪽 아니야? 그러니까 너가 알
다시피 네 뇌신경을 우리가 정해놓은 방식으로 작용하도록 통제하는 칩이
심어져 있어.

세주: 말도 안 돼..

호윤: 네가 그렇게 단순하게 살아온 게 사실은 그 칩이 너의 뇌신경을
조절해서야. 우리는 규칙표를 만들고, 그 다음에 네 뇌에 칩을 심고, 칩과
네 뇌가 충돌하지 않도록 정해진 환경 속에만 너를 가두고 똑같은 행동
양식을 반복하도록 했어. 그러니까 우리가 만든 규칙표에 맞게 너의 몸이
익숙하도록 뇌를 직접 통제하기도 했고, 환경도 같이 통제한 거지. 그리고
데이터를 학습하는 모델도 칩에 같이 넣어서 규칙표에 없더라도 비슷한
유형의 자극은 적절히 행동할 수 있게끔 한 거야. 이 아이디어를 처음 낸
게 정균태야.

세주: 그러니까... 내 뇌에 있는 칩이 있는데 그게 25년 전부터 만든 거
라고요?

호윤: 정균태랑 나는 대학교 때 만났어. 서로 생각하는 것도 비슷했고,
소위 당하는 사람들이었고, 그래서 친구가 됐지. 균태가 얘기하더라고, 세
상을 좀 바꿔보자고.

처음에는 게임을 만들자고 시작한 거였어. 사람들이 자기가 살고 싶은
대로 선택할 수 있는 게임을 만들자고, 그러려면 튜링 머신처럼 정해진 상
태에서 어떤 행동을 할지, 어떤 상태를 마주할지 정해야 했어. 그렇게 규

칙표를 만들고 규칙표가 또 규칙표를 만들고, 행동 반경이 정해지면 완전히 통제된 인생을 살 수 있게 할 만큼의 규칙표가 거의 완성될 때쯤, 군태가 나보고 얘기했었다. “호윤아. 우리가 이렇게 살았으면 참 좋았겠지?”

(호윤과 군태의 목소리가 겹치고, 회상씬으로 넘어간다.)

#10 과거 회상, 공대 창고 안에 있는 외딴 방

군태: (무한히 생성되고 있는 규칙표를 보며) 호윤아. 우리가 이렇게 살았으면 참 좋았겠지? 흐.

호윤: (장난스럽게 웃으며) 아이씨.. 우리가 이런 골방에서 둘이 이려고 있지는 않았겠다. 남들처럼 평범하게 살았겠지? (웃는 얼굴로 군태를 툴치며) 너랑 나는 왜 이려고 있냐~

군태: (쓱쓱한 미소를 머금고 침묵한다)

호윤: (군태를 보며, 비슷한 표정을 짓다가) 아이 됐고, 빨리 에러날 곳 없는지 확인이나 계속 하자.

군태: (호윤을 똑바로 쳐다보며) 호윤아. 우리 진짜 그렇게 해볼까?

호윤: (이해되지 않는 듯이 쳐다보며) 뭐? 뭐를 해봐...

군태: 진짜 사람이 이 규칙표 그대로 살게 해보자. 이 규칙표에 맞게 신경을 통제하고, 환경을 통제하면 사람이 필요한 행동만 하고, 불필요한 고통은 안 느낄 수 있어..

호윤: 그건 그렇지.. (실소를 터뜨리며) 근데 그게 말이 되냐? 뇌신경에 대해서 알기는 하고?

군태: 우리 같은 사람들이 생각보다 이 세상에 많아. 머리는 좋은데 상처가 큰 사람들. 하고 싶은 거 참으면서 사는 사람들. 그 사람들이랑 같이 세상을 바꿀 만한 일, 사람을 구원하는 일을 해보자. 난 하고 싶어.

#11 다시 현재. 외진 곳에 위치한 호운의 집.

세주: 왜 그런 짓을 한 거야?

호운: 옛날에는 개도, 나도 불쌍하니까, 상처 있는 사람들끼리 멍치게 되더라고. 그리고 사실 개 옆에 있으면 내가 덜 불쌍해지는 것 같았어. 도움을 주는 입장이 되니까. 그래서 정말 열심히 살았어. 그 꿈에, 그 일에 몰두하고 있으니까 잘못된 줄 몰랐어. 그냥 우리가 천재가 돼서 전능한 일을 하고 있는 줄로만 알았지. 지금 와서 생각해보면, 그냥 둘 다 미친놈이었지. 지금도 그렇고. (씩씩하게 웃는다.)

세주: (말없이 호운을 바라본다)

호운: 여기까지가 내 이야기야. 그 이후로는 정균태나 회사에 접근할 수가 없었어. 제대로 문제를 알고 해결하려면, 그 회사에 다시 들어가서 정균태를 만나야 돼. 시스템도 멈춰야 되고, 네 머리에 이식된 소프트웨어가 더 이상 널 조종할 수 없게 해야 돼.

세주: 당신도 똑같은 사람이네..? 당신이랑 정균태가 나한테서 나를 없애고, 이상한 규칙표를 심어 놓은 거잖아.

호운: 그래 너가 날 비난하는 거, 내가 없어지기를 바라는 거 이해해. 내가 생각해도 난 그렇게 돼야 마땅해. 근데 나는 널 돕고 싶다. 부모 없는 너를 처음 봤을 때, 날 닮았다고 생각했어. 나도 엄마 아빠 없이 자랐거든. 근데 어쩔 수가 없었어. 이미 그 칩을 개발하는 데 7년이라는 시간을 썼고, 그때는 나도 이식이 성공하기만을 바랐거든. 그런데 칩을 이식한 후에 하루 종일 데이터를 보고, 주입 받는 너를 보면서 내가 얼마나 잘못된 건지 생각하기 시작했어. 네가 들으면 웃기겠지만, 죄책감에 잠을 못 잤다. 시스템을 중단하자고 말하려고 했지만, 자기 신념에 이미 미쳐있는 균태를 내가 설득할 수는 없었어. 그래서 정균태와 인연을 끊고 네가 너를 찾을 수 있도록 하는 장치를 만들기 시작했어. 그게 바로 너한테 준 약이

야. 술병을 깨뜨렸을 때 퍼진 성분이 약통에 들어있는 약이랑 비슷해. 이 약을 개발하는 데 오랜 시간이 걸렸어. 네 뇌에 들어있는 칩이랑 바깥 시스템 사이의 교류를 끊는 거야. 하지만 약은 일시적인 방편이지, 진짜로 네가 해방되기 위해서는 아까 말한 것처럼 시스템을 멈춰야 돼.

세주: 당신이 나를 돕고 싶다는 말, 제가 그 말을 어떻게 믿어요?

호윤: (웃는다) 평생을 의심 없이 기계처럼 살아온 네가 그런 질문을 하니까 신기하네. 어떻게 대가 없는 희생이 가능한지 묻는 거야?

세주: (끄덕인다)

호윤: 대가 없는 희생을 가능하게 하는 동기는 글썄.. 잘 모르겠네. 그냥 인간이라? 그런데 내가 너를 돕는 이유는 확실해. 내가 나를 구원하려고. 죄책감을, 그리고 불쌍했던 내 과거가 너를 구하면 정화될 것 같아서. 그래서 그래. 너의 존재가 지금의 나를 살리고 있는 거야.

세주: (처음으로 중요한 무언가를 알게 된 듯한 표정으로 호윤을 쳐다 본다) 그럼.. 이제 어떻게 해야 하죠? 오늘 봤다시피 이미 회사에는 나를 잡으려는 사람들이 너무 많아요. 정균태가 이미 수를 써놓았을 수도 있고요...

호윤이 계획을 설명한다.

호윤: (세주를 쳐다보며) 계획 다 알아 들었지?

세주: (고개를 끄덕인다) 그런데 실패하면 어찌죠?

호윤: 실패하지 않아. 내가 그렇게 만들 거야. 걱정하지 말고, 그날 전까지는 편하게 지내봐. 이곳저곳 다니면서 세상이 어떤 곳인지 네가 온전히 느꼈으면 좋겠다.

세주와 호윤은 세주가 이제껏 해보지 못한 다양한 경험을 해본다. 그림 그리기, 문학 책 읽기, 낚시 해보기 등등.

#12 D-day, 차 안

호윤: 긴장했냐?

세주: (긴장한 얼굴로 호윤을 쳐다보며) 긴장 안 되세요?

호윤: (장난스럽게 웃으며) 어차피 성공할 거 뺏히러 긴장을 해~

세주: (미소를 머금고) 그럼 긴장 안 해요.

호윤: 그래. 얘기했던 대로만 하자. 그렇게 안 돼도 괜찮으니까 걱정 말고, 약 먹었지?

세주: (회사를 쳐다보며) 네. 이제 가요.

#13 회사

차에서 내려 호윤과 세주는 회사에 잠입하고 경비원들, 미로 같은 건물 구조에서 중요한 단서를 찾아 나선다. 하지만, 호윤과 떨어진 사이 세주가 붙잡히게 된다.

#14 군태의 39층 사무실

군태가 세주를 반갑게 맞이한다. 최신의 장비로 가득한 사무실. 고층의 사무실은 통 유리창으로 둘러싸여 있다.

군태: (아이처럼 미소 짓고 숙스러워 하며) 안녕. 음.. 안녕! 어떡해. 너무 떨리네. 아 일단 편하게 앉아. 내 의자에 앉아. 이거 좋은 의자야. 엄청 편해. (세주를 자기 의자 쪽으로 안내한다)

세주: (경계하며 천천히 앉는다)

군태: 어때 편하지 않아?

세주: (경계하고 두려워하는 표정을 짓는다)

군태: (굳은 표정으로) 안 편한가? 으.. 다른 데 앉을래?

세주: 아니요, 괜찮아요.

균태: (웃는 표정으로) 그래! 아.. 진짜 오랜만이다...

세주: 오랜만이요..? 지금 처음 만나는 거 아닌가요?

균태: (재밌다는 듯이 키득대며) 아! 너는 기억 못하지, 맞다. 혼자 웃어서 미안해. 그게 말이야, 완벽한 너니까... 혹시 결함은 없나 확인 차 몇 번 나랑 대화도 나누고, 검사도 받고 그랬어... 또 새로운 환경에 맞게 반응도 업데이트도 하고... 건강검진 같은 거지! 건강검진!

세주: 그리고 내가 기억 못하는 상태로 만든 거고요?

균태: (눈치 보듯이) 응.. 그렇지.. 그래야 완전한 기계로 남으니까. 정호 윤한테 다 들었나보네. 충격 받는 표정이 아닌 거 보니까. 그래 뭐 지금은 상태여도 상관없지. 이렇게 된 거 네가 얼마나 완전했는지 보여줄까? (리모컨으로 뿜 하고 영상을 재생한다.)

여러 개로 분할된 화면에 어린 시절 세주의 모습이 나타난다.

균태: 보여? 너는 진짜 대단했어. 진짜 사랑스러웠다구. 데이터를 입력하는 족족 다 제대로 받아들이고 행동했어. 내가 만든 규칙표대로, 내가 계획한 대로 행동하고 느끼는 그런 기계였어. 너만큼 완벽한 기계는 없었어... 내가 틀리지 않았다는 증거, 내가 쏟아 부은 시간이 헛되지 않았다는 증거. 그게 너야, 세주야. 어차피 너는 나랑 이렇게 나는 대화도 이제 곧 잊어버릴 테니까.. 네가 듣고 싶었던 얘기 해줄게. 내가 너를 왜 만들었는지.

나는..... 괜찮다는 말을 질리도록 했어. 사람들 틈에 있으면 그 말을 해야 그나마 숨 좀 쉬어지거든. 그렇지 않고, 내가... 내가 좋아하는 거 얘기 하잖아? 그럼 나를 잠깐 이상하게 쳐다보고는 자기들끼리 눈빛을 막 주고 받아. 누가 너를 이상하게 보는 거.... 넌 모르겠지? 내가 너를 그렇게 안 만들었으니까. 절대로 하대 받을 수 없는 조건으로 네 인생을 설계했으니까. 나는 하대 받는 삶을 살았어. 돈이 없어서, 못생겨서, 사회성이 부족해

서, 키도 작고 눈도 못 마주치고, 아무것도 안 해도 무시당하고, 욕을 먹었어. 나는 머리가 좋고 똑똑한데도, 아무도 관심 갖지 않았어. 관심 갖는 건 내가 자기들보다 못한 부분들이었어. 내가 얼마나 나를 구하고 싶었는지 알아? 이미 늦었었지. 태어날 때부터 난 이렇게 태어난 건데 어찌겠어. 그래서 내가 나를 구하지는 못해도, 단 한 명이라도, 이 세상에 존재하는 단 한 명만이라도 이 더러운 세상이... 더러운 건지 모르게 하고 싶었어. 네가 그 어린 나이에 집이 있고, 차가 있고, 상처받은 적 없고, 힘들지 않은 건... 규칙표 안에서 살아와서 그런 거야. 내 덕분이라고, 고민? 진짜 감정? 그런 건 삶을 흔들기만 해. 도움이 안 된다니까? 경험해보지 않아도 될 것들이야. 왜 이제 그런 게 궁금해지는 거야 진짜. 완벽한 상태로 남아 세주야. 괜히 흠집 내려고 하지 말고.

세주: (균태를 노려보며) 네 욕망 하나 펼치려고 내 뇌에 칩을 심고, 날 통제했는데.. 그게 나를 위한 일이라고 지껄인거야?

균태: (눈을 동그랗게 뜨며) 세주야, 너 분노도 느끼는 거야? 진짜 안 되겠다. 이제 그만 원래 세주로 돌아가자. 정호윤 그 새끼가 애를 망쳐놨어. 그 약도 다 없애고 다른 약 효과도 안 통하게 칩도 바꿔야겠다.

세주: 정호윤은... 날 도와준 거야. 네가 나를 망쳐놓은 거지.

균태: 세주가 지금 말을 잘 못 알아 듣는구나? 정호윤은 너랑 나, 우리를 배신한 놈이야. 엄마 아빠 없이 힘들게 살아갈 뻔한 너한테 새 삶을 선물해준 건 나고, 물론 정호윤도 초반에는 도왔지만 결국에 지금 다 망쳐놓고 있잖아? 아, 이제 그만하자. 세주야 나는 너를 무척 아껴. 지금 네가 정말 화가 나게 행동하고 있지만, 그래도 그건 약 때문이지, 원래의 네가 아니야. 여기 뷰 좀 볼래? 진짜 좋아.

균태가 앞장서서 창가 쪽으로 가자 세주도 일어나 균태 뒤를 따라간다. 그리고는 조심스럽게 자켓 주머니에 넣은 주사기를 꺼내든다. 세주가 주사

기로 균태를 찌르려고 하자 균태의 비서가 세주에게 미리 준비해놓은 마취제를 발사한다. 세주가 천천히 의식을 잃고 쓰러지고 균태가 비서에게 말한다.

균태: (슬픈 얼굴로) 이제 세주 옮기고 기억 지워주세요. 손 봐야 할 부분은 내일 직원들 소집해서 얘기할 테니까 간단히 하고 집에 가시구요.

비서: 네. 알겠습니다. (경비원들을 향해) 40층으로 올라가요.

경비원들이 세주를 이동침대에 눕히고, 데려간다. 비상 엘리베이터에 타 40층을 누른다.

#15 다시 균태의 39층 사무실

몸 이곳저곳에 상처를 입은 호운이 균태를 찾아왔다.

균태: 이번에는 용케 여기까지 왔다? 몇 번을 쥐새끼처럼 여기에 들어오려고 하던지. 지겹지도 않니? 호운아.

호운: 그럼 너는 세주 괴롭히는 거 지겹지도 않냐? 25년이면 이제 그만 할 때도 됐잖아.

균태: (의아하다는 표정으로) 지겹긴 뭐가 지겨워. 매번 새롭고 경이로운데? 세상은 항상 바뀌고 세주도 그에 맞게 바뀌어야지. 그러려면 내가 필요해.

호운: (약간 화를 참으며) 내 말은 이제 죄책감 느낄 때가 되지 않았냐는 거잖아.

균태: (호운을 똑바로 쳐다보며) 응 안 느껴. 나는 세주를 구해줬는데 내가 왜? 죄책감은 네가 느껴야지.

호운: 몇 년만에 네 얼굴 보는지 모르겠는데 여전히 변한 거 없이 짜증 나는구나.

균태: 너도 마찬가지로야. 반갑지도 않아.

호윤: 세주를 위해서 오늘 너랑 나랑 둘 다 사라져야 해.

균태: 음 아니지. 나는 아니고 너 혼자 사라지면 딱이겠다. 어차피 여기 바깥에도 경비원분들도 계시고, 너는 빠져나갈 수가 없어. 그냥 약 어떻게 만들었는지만 설명하고 내 눈앞에서 사라져. 앞으로 나타나지도 말고, 나는 폭력 싫어, 죽이기도 싫고.

호윤: 폭력 싫다는 애가 오늘은 나 때리라고 엄청 얘기했나보다? 내 꼴 좀 봐.

균태: 네가 세주를 계속 괴롭히니까 그렇지.

호윤: 너는 진짜... 죽어서도 잘못된 거를 모를 것 같다.

호윤이 주사기를 꺼내 균태한테 달려간다. 둘이서 서투게 하는 몸싸움. 끝내 호윤의 몸에 균태가 주사를 찔러 넣는다.

#16 40층 건물 복도

세주가 실패실로 옮겨지던 도중 진주가 동료들과 세주를 구하러 온다. 진주가 새 약을 넣자 세주가 눈을 뜬다. 벽에 붙어있는 40F 표시.

세주: (힘겹게 몸을 일으켜 세우며) 누..구세요?

진주: 저는 호윤이 친구예요. 여기는 제 동료들이고, 세주씨 정신 차리게 하는 약 만든 사람들이예요.

세주: (잇은 걸 떠올렸다는 표정으로) 아...

진주: (약간 다급하게) 지금 호윤이가 정균태랑 같이 있어요. 오늘 원래 정균태를 만나러 온 거예요. 둘이 같이 사라진다고... 호윤이는 저희가 데리러 갈게요. 세주씨는 여기서 잠시 쉬고 있어요.

세주: (깜짝 놀라며) 저도 갈게요. 가고 싶어요.

동료들이 세주를 부축해 침대에서 내려주고, 함께 뛰어간다.

#17 군태의 39층 사무실

엉망인 상태로 쓰러져 있는 호운. 그 옆에 군태도 쓰러져있다. 세주와 진주 일행이 오자 호운이 힘겹게 입을 뗀다.

호운: 아 진짜 오지 말라니까. 내가 이런 꼴을 보여야겠냐..

진주: 어떻게 안 오냐. 네가 이려고 있을 거 뻔히 아는데. 세주씨랑 인사는 해야될 거 아니야..

세주: 괜찮으세요? 누가 이렇게....

호운: (쿨럭할 때 입에서 피가 울컥 나온다) 아이.. 또 이러네. 이제 진짜 얼마 안 남았어. 나는 괜찮아. 좀만 더 누워 있다가 갈 거니까 먼저 가서 그 방을 찾아. 정군태는 이제 아무 기억도 못 할 거니까.

세주: 그 방 어디인지 알아요. 아까 끌려가던 복도에 40층이라고 되어 있었어요. 39층이 정군태 사무실이고 엘리베이터에도 없는 층수인 거 보니까 아무래도 거기 있는 방들 중 하나 같아요. 그러니까 얼른 같이 가요.

호운: 영화를 많이 본 보람이 있네! 탐정 같다 세주야. 그런데 나는 아직 걸기가 힘들어.. 먼저 가 있어라. (진주의 팔을 붙잡고) 진주야, 세주 좀 데려가줘. 나는 괜찮으니까. 얼른. 진짜 고맙웠다.

진주: (눈시울이 약간 붉어지며) 응.. 그래.. 나도 고맙웠어.

세주: (이상함을 느끼는 표정으로) 오는 거 맞죠?

호운: (미소를 띠며) 당연히 가야지. 또 낚시하러 가자.

세주: (미소를 띠며) 그래요. 좀 이따 배요.

#18 40층 연구실

진주 일행과 함께 비상 엘리베이터로 40층에 다시 온 세주. 여러 방을 돌아다니다 마지막 가장 구석진 곳에 있는 연구실 앞에 서 있다.

진주: 우리는 앞을 지키고 있을 테니까 세주씨 혼자 들어가 보세요.

세주가 천천히 문을 열고 들어간다. 커다란 연구실에 세워진 여러 대형 기둥 안에 하나씩 전시되어있는 칩과 그에 대한 설명. 눈앞에 보이는 커다란 스크린. 우리나라 지도 위로 수백 개의 빨간 불빛들이 반짝인다. 컴퓨터의 화면에서 불빛을 터치하자 사람의 사진과 성격에 대한 설명, 칩 버전, 만들어진 연도가 나타난다. 자신 외에도 수많은 기계적 인간이 있음을 알게 된 세주. 작동 중지 버튼을 공허한 표정으로 뚫어져라 쳐다보다 결심한 듯한 표정을 하고 손을 얹는다. 천천히 버튼을 누른다. 쓰러지는 세주. 지도 위로 사라지는 빨간 불빛들. 진주 일행이 울려 퍼지는 비상 경보음 소리에 연구실 안으로 들어온다. 쓰러진 세주를 보고 놀라며 세주를 업고 건물 밖으로 빠져나간다.

#19 건물 밖

진주: (떨리는 목소리로) 호윤아, 우리 나왔어. 카메라 확인해보니까 경비원들도 다 따라 나온 것 같고 건물 비어있어. 자료는 다 챙겼으니까 이제 해도 될 것 같아.

호윤: (숨을 깊게 내쉬며) 정군태도 바깥에 잘 버려뒀지?

진주: (살짝 웃으며) 응. 잘 버려뒀어.

호윤: (쿨럭거린 후에) 고생 많았어 진주야. 네가 만들어준 약 없었으면 이렇게 할 수 없었을 거야.

진주: (감정에 북받쳐) 네가 약 임상실험에 몇 년을 바쳤는데... 네 몸 그렇게 망가진 것도 다 그거 때문이잖아. 세주씨 알기는 해..?

호윤: (얇게 웃으며) 어떻게 알겠어. 얘기하지마. 세주한테 관한 죄책감
심어주는 일 절대 하지마 진주야.

진주: (망설이다가) 알겠어. 알겠어... 너 생각 안 바꿀 거야?

호윤: (힘겹게 한 마디 한 마디 내뱉으며) 안 바뀐. 지금 나 한계야 진
주야. 너무 힘들다. 세주 잘 부탁해. 고마워.

진주: (전화를 내리고 운다)

호윤이 가지고 있던 기폭장치 버튼을 누르며 건물이 폭발한다. 진주 일
행과 세주가 탄 헬기는 유유히 비행한다.

에필로그

겨울날 낚시터. 머리 근처에 수술자국이 있는 세주가 낚시를 하고 있다.
그 옆에는 비슷한 곳에 수술자국이 있는 어떤 이가 함께 앉아 있다.

-

주제: 사회의 편견과 차별을 겪으며 일그러진 인간이 일그러뜨린 한 인
간의 방향과, 책임감 있는 어른의 도움으로 그가 주체성을 갖춰가는 과정.

시 사냥

강승윤

400년의 축적

#상승 -샤를 보들레르

#길 -신경림

그 도구의 실현

#Trip Switch -Nothing But Theives

소프트웨어, 지혜로 짓는 세계

#그릇 -오세영

#낮과 밤의 경계 -다양성

#The Times They Are A-Changin' -Bob Dylan

#나는 물고기에게 말한다 -정호승

#거울 -이상

그 도구의 응용¹⁾

#출발 -김동률

#철길 -용혜원

#네가 황제다 -알렉산드로 푸쉬킨

못을 넘어, 골짜기를 넘어,
산을, 숲을, 구름을, 바다를 넘어,
태양을 지나, 에테르를 지나,
별 박힌 천구의 경계를 지나,

내 정신아, 너는 날렵하게 움직여,
물결 속에서 நீ를 잃는 수영선수처럼,
형언할 수 없고 씩씩한 기쁨에 겨워
그윽한 무한대를 쾌활하게 누빈다.

이 병든 장기에서 멀리 날아가,
드높은 대기 속에서 너를 맑게 씻고,
청명한 공간을 가득 채운 저 밝은 불을
순결하고 신성한 술처럼 마셔라.

안개 낀 삶을 무겁게 짓누르는
권태와 망망한 근심걱정에 등 돌리고,
복되도다. 빛나고 청명한 별판을 향해
힘찬 날개로 날아갈 수 있는 자,

생각이 종달새처럼, 하늘을 향해
아침마다 자유 비상을 하는 자,

- 삶 위로 날며, 꽃들과 말 없는 것들의 말을
애쓰지 않고 알아듣는 자 복되도다!

-

문학사에 새로운 도정을 연 샤를 보들레르와 인공지능과 컴퓨터 과학의 새로운 막을 연 앨런 튜링은 많이 닮아있다. 튜링이 컴퓨터 과학계에 던진 질문, 그리고 보들레르가 당대 사회상에 던진 질문이 얼마나 파격적이었는가. 나도, 당신도 질문하고, 상승하자. 우리의 생각이 종달새처럼, 매일 아침 하늘을 향해 자유 비상하기를 바라며 한 마디를 마친다.

사람들은 자기들이 길을 만든 줄 알지만
길은 순순히 사람들의 뜻을 좇지는 않는다
사람을 끌고 가다가 문득
벼랑 앞에 세워 낭패시키는가 하면
큰물에 우정 제 허리를 동강 내어
사람이 부득이 저를 버리게 만들기도 한다
사람들은 이것이 다 사람이 만든 길이
거꾸로 사람들에게 세상 사는
슬기를 가르치는 거라고 말한다.
길이 사람을 밖으로 불러내어
온갖 곳 온갖 사람살이를 구경시키는 것도
세상 사는 이치를 가르치기 위해서라고 말한다
그래서 길의 뜻이 거기 있는 줄로만 알지
길이 사람을 밖에서 안으로 끌고 들어가
스스로를 깊이 들여다보게 한다는 것을 모른다.
길이 밖으로가 아니라 안으로 나 있다는 것을
아는 사람에게만 길은 고분고분해서
꽃으로 제 몸을 수놓아 향기를 더하기도 하고
그늘을 드리워 사람들이 땀을 식히게도 한다
그것을 알고 나서야 사람들은 비로소
자기들이 길을 만들었다고 말하지 않는다

-

튜링은 어떤 길을 걸었을까? 그는 바깥으로 향하는 길이 아니라 자신의 물음에 끊임없이 답하고 또 질문하며 자신의 안으로 향하는 길을 걸었으리라. 질문에 대한 답을 찾기 힘들 때에는 좌절하는 낭패를 겪기도 했을 것이다. 그러나 걷다보니 길이 생기고, 자신의 물음을 살피다 보니 길은 그에게 꽃을 피워주지 않았을까? 우리도 내면을 살피고, 스스로에게 귀 기울이다 보면 고단하기만 했던 길에 꽃이 줄지어 피어 있지 않을까. 안으로 나있는 길을 모두가 발견하기를 바라며 한 마디를 마친다.

그 도구의 실현

Trip Switch

아티스트: Nothing But Thieves

앨범, 장르: Nothing But Thieves, 얼터너티브 락

Sharing secrets with another world
Rubbing shoulders with some unknown lovers
Making waves through the universe
Starting wars with anonymous brothers
Trip switch, trip switch
Make a wish and I'll count to three
Press the button and we'll both be happy
Sending signals is a dirty trick
I get my love in a digital package
Trip switch, trip switch

What we do when the power's out
What do we do when the lights go down?
What we do when the power's out
What do we do when the lights go?
Down, down, down, down, down, when the lights go
Down, down, down, down, down, lights go

Yeah, gimme data 'cause I need a hit
How I love stratospheric numbers

Information, I've been craving it
Gimme yours, then gimme all the others
Trip switch, trip switch

What we do when the power's out
What do we do when the lights go down?
What we do when the power's out
What do we do when the lights go?
Down, down, down, down, down, when the lights go
Down, down, down, down, down, when the lights go
They put out the lights, they put out the lights
They put out the lights

-

튜링이 정의한 ‘기계적인 방식’은 단순한 다섯 종류의 기계 부품으로 만든 기계로 작동하는 방식이다. 그 기계를 현실에서 구현할 때 가장 핵심적인 장치는 ‘스위치’이다. 스위치가 직렬 또는 병렬로 연결되고, 열리거나 닫히며 조지 부울(George Boole)의 논리를 구현할 수 있다.

이 노래의 가사는 Trip Switch를 놀리 디지털 세상에서 벗어나는 상황에 대해 담고 있다. Trip switch는 안전상의 목적으로 전류를 끄는 자동 스위치이다. 원활한 기계 작동과 안전을 위해 스위치는 적절히 개폐가 이루어져야 한다. 우리의 마음 속 스위치도 마찬가지다. 우리 마음속에 스스로를 지킬 수 있는 Trip Switch가 있는지 생각해보기를 바라며 한 마디를 마친다.

깨진 그릇은
칼날이 된다.
절제와 균형의 중심에서
빛나간 힘,
부서진 원은 모를 세우고
이성의 차가운
눈을 뜨게 한다.

맹목(盲目)의 사랑을 노리는
사금파리여,
지금 나는 맨발이다.
베어지기를 기다리는
살이다.
상처 깊숙이서 성숙하는 혼(魂)

깨진 그릇은
칼날이 된다.
무엇이나 깨진 것은
칼이 된다.

-

소프트웨어를 구현하는 두 가지 핵심 요소는 알고리즘과 언어이다. 이때 알고리즘은 소프트웨어가 만든 문제를 푸는 방법이고, 언어는 소프트웨어를 표현하는 방식이다. 먼저 알고리즘의 그릇 역할을 적절하게 표현해낸 시를 소개하고, 이후에 고삐 역할을 하는 언어에 관한 시를 살펴보고자 한다.

좋은 알고리즘이란, 적은 비용으로, 보다 많은 문제를 풀 수 있어야 한다. 문제는 모든 문제를 적은 비용으로 풀 수 있는 알고리즘을 구현하는 것 자체가 비현실적이라는 점이다. 따라서 현실에서 알고리즘은 많이 나오는 문제에 대해 정확한 값이 아닌 근사치를 답으로 제시하고 있다.

알고리즘이라는 그릇 안에 모든 문제를 담아내겠다고 생각하면, 그 그릇은 깨지고 말 것이다. 깨진 그릇은 칼날이 된다. 절제와 균형은 얼마나 중요한가. 그릇이 깨지지 않으려면 그릇에 강요되는 무언가를 비워내야 하지 않을까? 우리 일상에서도, 또 컴퓨터 과학에서도 ‘비워냄’의 미학은 똑같이 적용되는 것만 같다. 마음의 그릇이 좁다면, 안에 담긴 감정을 비워내는 법을 배우자. 알고리즘 그릇이 좁다면, 안에 담은 문제를 비워내자. 깨진 그릇은 칼날이 되므로, 작은 그릇에 과도하게 우겨넣지 말자. 우리 모두 그릇을 조금씩 넓혀가는 방법을 고민해보기를 바라며 한 마디를 마친다.

소프트웨어, 지혜로 짓는 세계

낮과 밤의 경계

아티스트: 다양성

앨범, 장르: 종이비행기와 풍선으로 가득 찬 세상, 록/메탈

뭐가 겁이 나서 그렇게 망설이고 있는 거야
시간이 없어
풀리질 않는 문제들을 이리저리 물고 늘어진대서
해결되진 않더라고 무슨 말이 필요해
일단 가서 부딪치고 저지르고 생각해
무엇이든 잘될 것만 같은 그런 느낌이 있지
일단 가서 부딪치고 저지르고 생각해
늦어버리기 전에
지금 시작해볼래
불어오는 바람에 머리칼을 내맡기고 달려가
새로운 나를 위해
살아있다는 걸 보여줄 거야
여기 낮과 밤의 경계에 서서
경계에 서서
경계에 서서
어제보다 오늘
더 화려해지고 자유로워졌어
지나간 일들과 불안한 미래를 걱정하진 않을래
너무 늦어버리기 전에
지금 시작해볼래

불어오는 바람에 머리칼을 내맡기고 달려가
새로운 나를 위해
살아있다는 걸 보여줄 거야
여기 낮과 밤의 경계에 서서
너무 늦기 전에
좀 더 높은 곳으로
좀 더 멀리 바라볼 거야
조금 더 화려하게
난 달라질 거야
여기 낮과 밤의 경계에 서서
경계에 서서
경계에 서서

-

알고리즘으로 모든 문제를 쉽게 풀 수 있을까? 그렇지 않다. 문제 중에는 쉽게 풀리는 P문제, 어려운 문제, 그리고 그 둘의 경계에 있는 NP문제가 있다. 하지만 NP문제의 존재는 수학적으로 증명된 바가 없다.

NP문제를 현실적으로 풀 수 있는 방법은 없을까? 모든 입력에 대한 완벽한 답을 내기에는 무리다. 가사처럼 풀리지 않는 문제들을 이리저리 물고 늘어진대서 해결되진 않는다. 그래서 사용하는 현실 지혜는 ‘통밥’과 ‘무작위’이다. 통밥은 직관에 기대어 적당한 답을 내고, 무작위는 아무렇게나 답을 내고 유연하게 대처한다. 너무 늦어버리기 전에 시작해보는, 이 두 가지 방식은 현실에서 꽤나 유용하다. 우리가 일상에서 마주하는 NP문제들에 망설이지 말고, 일단 가서 부딪쳐보기를 바라며 한 마디를 마친다.

소프트웨어, 지혜로 짓는 세계

The Times They Are A-Changin'

아티스트: Bob Dylan

앨범, 장르: The Times They Are A-Changin', 포크

Come gather 'round people
Wherever you roam
And admit that the waters
Around you have grown
And accept it that soon
You'll be drenched to the bone
If your time to you is worth savin'
And you better start swimmin'
Or you'll sink like a stone
For the times they are a-changin'

Come writers and critics
Who prophesize with your pen
And keep your eyes wide
The chance won't come again
And don't speak too soon
For the wheel's still in spin
And there's no tellin' who
That it's namin'
For the loser now

Will be later to win
For the times they are a-changin'

Come senators, congressmen
Please heed the call
Don't stand in the doorway
Don't block up the hall
For he that gets hurt
Will be he who has stalled
The battle outside ragin'
Will soon shake your windows
And rattle your walls
For the times they are a-changin'

Come mothers and fathers
Throughout the land
And don't criticize
What you can't understand
Your sons and your daughters
Are beyond your command
Your old road is rapidly agin'
Please get out of the new one
If you can't lend your hand
For the times they are a-changin'

The line it is drawn
The curse it is cast
The slow one now
Will later be fast
As the present now
Will later be past
The order is rapidly fadin'
And the first one now
Will later be last
For the times they are a-changin'

-

밥 딜런은 최초로 노벨 문학상을 받은 싱어송라이터이다. 노벨 문학상 수여를 맡고 있는 스웨덴 아카데미에 따르면, “딜런은 위대한 미국의 가요의 전통 속에 새로운 시적인 표현들을 창조해냈다”고 선정 이유를 밝혔다.

그는 이 가사에서 사회는 언제나 변혁될 수 있음을 시적으로 표현하고 있다. 그는 사회 비판과 계몽의 의도로 이 가사를 창작했을 수 있지만, 그가 사용한 “the times they are a-changin'”은 학문에도 적용될 수 있다. 시간이 지남에 따라 모든 학문은 변화하기 마련이다. 컴퓨터 과학의 세계가 천천히 확장되고 있듯이. 따라서 어떤 분야든, 학문에 몸담고 있는 학자들에 있어, 시간의 변화에 따라 유연히 생각하는 태도는 자신의 주장에 갖는 자신감만큼이나 중요하지 않을까?

-

알고리즘의 현실적인 비용을 줄여준 신기한 알고리즘이 등장했다. 바로 양자 알고리즘이다. ‘어벤저스: 엔드게임’에서 시간여행을 가능하게 해준 그 양자역학 이론에서 파생한 아이디어이다. 물론 영화 속 양자역학을 통한 시간 여행은 판타지이지만, 컴퓨터에 이용되는 양자 알고리즘은 현실이다.

컴퓨터에서 이용하는 세 가지 양자 현상은 ‘겹쳐있기’, ‘얽혀있기’와 ‘확률진폭’이다. 각각은 양자 안에 여러 숫자가 중첩되어 있음, 두 양자는 서로 연결되어 있음, 하나의 확률을 두 개의 확률 진폭으로 나타낼 수 있음을 의미한다. 다음과 같은 현상을 활용한 두 가지 알고리즘이 존재하는데, 양자 인수분해 알고리즘과 양자 탐색 알고리즘이다. 이를 바탕으로 양자 컴퓨터로는 동시에 여러 데이터를 처리 및 확인할 수 있으므로, 기존의 컴퓨터보다 훨씬 빠른 속도로 작동할 수 있다. 아직까지 상용화되지 않고 실험 단계에 있지만, 머지않은 미래에 양자 이론은 인류에게 새로운 컴퓨터 과학의 세계를 열어줄 것이다.

아인슈타인은 ‘신은 주사위 놀이를 하지 않는다.’고 말했다. 하지만, for the times they are a-changin’! (시대는 변하고 있다!)

소프트웨어, 지혜로 짓는 세계

나는 물고기에게 말한다

정호승

그래도 너를 사랑한다고 말하고 싶을 때
그래도 너를 사랑하지 않는다고 말하고 싶을 때
그래도 떠날 때는 내 돈을 모두 너에게 주고 싶다고 말하고 싶을 때
그래도 너에게 단 한푼도 줄 수 없다고 말하고 싶을 때
나는 촛불을 들고 강가로 나가 물고기에게 말한다
물고기는 조용히 지느러미를 흔들며 내 말을 듣고만 있을 뿐
아무에게도 아무 말도 하지 않으므로

내 산을 모두 밭으로 만들어 너에게 주고 싶다고 말하고 싶을 때
네 밭을 모두 산으로 만들어 내가 가지고 싶다고 말하고 싶을 때
아무에게도 들키지 않고 이제는 인간이 되고 싶지 않을 때
기어이 인간을 버리고 혼자 울고 싶을 때
나는 강가로 나가 물고기의 허리를 껴안고 운다
침묵만이 그들의 언어이므로
침묵 외에는 그 어떠한 말도 하지 않으므로

-

언어는 소프트웨어를 표현하는 방식이다. 적절한 언어가 없으면, 소프트웨어를 만들 수 없다. 따라서 컴퓨터의 언어는 중요하다. 보통 소프트웨어를 제작하는 일을 프로그래밍이라고 부르기 때문에, 컴퓨터의 언어를 프로그래밍 언어 또는 C 언어라고 부르기도 한다. 이중 가장 원시적인 형태는 규칙표이다. 하지만, 이는 인간이 이해하기에는 한계가 있기 때문에, 각기 다른 기능에 초점을 둔 컴퓨터 언어가 발전해왔다. 컴퓨터와 인간 사이에 존재하는 거리를 좁히기 위해 다양한 프로그래밍 언어는 사슬 관계를 맺고 있다. 인간이 이해하기 쉬운 상위 언어가 점점 밑 단계의 언어로 번역이 되면서 컴퓨터에 도달한다. 즉, '언어 사슬'과 '번역'이 프로그래밍 언어의 가장 주요한 개념이다.

인간이 하고 싶은 말을 프로그래밍 언어로 입력하면 언어 사슬을 타고 내려가며 번역이 이뤄진다. 컴퓨터에 시키면, 컴퓨터는 실행한다. 하지만 물고기에 말을 하면, 물고기는 침묵한다. 침묵만이 그들의 언어이므로, 어떤 사람들은 하고 싶은 말을 꼭 눌러 답는 게 습관이다. 그러다 해야 하는 말마저 꼭 눌러 답아 울고 싶은 순간을 마주한다. 다른 사람한테 말할 수도 없지만 홀로 가두고 있기에는 벽찬 순간을 마주한다. 위로와 회복의 시간이 필요하다. 그렇게 물고기한테 간다. 물고기가 지느러미 달린 물고기가 아닐 수 있다. 침묵만이 언어인 나만의 물고기에게 간다. 아무에게도 들키지 않고, 이제는 인간이 되고 싶지 않기 때문에. 우리가 누군가에게 물고기가 될 수 있기를 바라며 한 마디를 마친다.

거울속에는소리가없오
저렇게까지조용한세상은참없을것이오
거울속에도내게귀가있오
내말을못알아듣는딱한귀가두개나있오
거울속의나는왼손잡이오
내악수(握手)를받을줄모르는-악수(握手)를모르는왼손잡이오
거울때문에나는거울속의나를만져보지를못하는구료마는
거울아니었던들내가어찌거울속의나를만나보기만이라도했겠오
나는지금(至今)거울을안가졌오마는거울속에는늘거울속의내가있오
잘은모르지만외로된사업(事業)에골몰할께요
거울속의나는참나와는반대(反對)요마는또꽤답았오
나는거울속의나를근심하고진찰(診察)할수없으니떡섭섭하오

-

알론조 처치는 튜링과는 다른 방식으로 “기계적으로 계산 가능한 것”을 정의했다. 바로 램다 계산법이다. 이는 튜링 프로그래밍 언어와는 다른 시각으로 문제에 접근한다. 튜링은 기계의 상태 변화를 가능하게 하는 도구를 언어라고 생각했다면, 램다 계산법은 특정한 램다식이 계산되고 결과가 나오는 논리를 언어이다.

램다 계산법의 관점에서, 논리는 언어의 거울인 셈이다. 논리가 언어이고, 언어가 논리이다. 이렇게 두 대상이 동일하거나 비슷한 속성을 공유할 때, 우리는 흔히 ‘거울’에 빗대어 표현한다. 그렇기에 거울은 내면을 돌아보는 중요한 매개체이기도 하다. 사람이 자신을 시각적으로 인지할 수 있는 도구이기 때문이다. 나의 표정과 모습은 내 상태를 돌아보게 만든다. 윤동주의 시 ‘참회록’과 ‘자화상’을 읽어보기를 바라며 한 마디를 마친다.

그 도구의 응용

출발

가수: 김동률

앨범, 장르: Monologue, 발라드

아주 멀리까지 가보고 싶어
그 곳에선 누구를 만날 수가 있을지
아주 높이까지 오르고 싶어
얼마나 더 먼 곳을 바라볼 수 있을지

작은 물병 하나 먼지 낀 카메라
때 묻은 지도 가방 안에 넣고서

언덕을 넘어 숲길을 헤치고
가벼운 발걸음 닿는 대로
끝없이 이어진 길을 천천히 걸어가네

멍하니 앉아서 쉬기도 하고
가끔 길을 잃어도 서두르지 않는 법
언젠가는 나도 알게 되겠지
이 길이 곧 나에게 가르쳐 줄 테니까

촉촉한 땅바닥 앞서간 발자국
처음 보는 하늘 그래도 낮익은 길

언덕을 넘어 숲길을 헤치고
가벼운 발걸음 닿는 대로
끝없이 이어진 길을 천천히 걸어가네

새로운 풍경에 가슴이 뛰고
별것 아닌 일에도 호들갑을 떨면서
나는 걸어가네 휘파람 불며
때로는 넘어져도 내 길을 걸어가네

작은 물병 하나 먼지 낀 카메라
때 묻은 지도 가방 안에 넣고서

언덕을 넘어 숲 길을 헤치고
가벼운 발걸음 닿는 대로
끝없이 이어진 길을 천천히 걸어가네

내가 자라고 정든 이 거리를
난 가끔 그리워 하겠지만
이렇게 나는 떠나네 더 넓은 세상으로

-

컴퓨터는 다른 도구와는 다른 ‘보편만능’이라는 성질로 인간의 능력에 커다란 변화를 가져왔다. 인간의 지능, 본능, 현실이 확장되었다.

지능의 측면에서 인간은 컴퓨터와 협력하고 있다. 새로운 지식 표현 방식이 생겨났고, 이제까지 만들기 어려웠던 지식이 생성되고 있다. 뿐만 아니라 대량의 정보를 검색할 수 있고, 여러 사람들과 함께 하는 연구가 가능해져 문제 해결이 더욱 신속해지기도 했다. 이러한 지능의 확장은 또 컴퓨터의 발전으로 이어지며 인간과 컴퓨터가 상호작용함을 알 수 있다.

본능의 측면에서 놀이 본능과 소통 본능이 확장되었다. 넓은 세계를 바탕으로 하는 다양한 게임이 개발되었고, 소셜 미디어는 관심 있는 사람들과 소통하는 아주 중요한 통로가 되었다. 이러한 발달은 인간의 놀이 본능과 소통 본능을 실현시키고, 확장시켰다.

현실의 측면에서 컴퓨터는 시공간의 한계를 극복하는 데 기여했다. 과거에는 정보의 공유가 시공간의 공유를 전제로 했다면, 현재 컴퓨터가 발달하며 정보를 전달하는 제 3자의 역할이 생겨났다. 이때 제 3자가 그 정보를 알 수 없도록 암호를 사용하고, 다양한 암호화 방식이 개발되고 있다.

-

인간은 아주 멀리까지 가보고 싶고, 아주 높이까지 오르고 싶어 한다. 이러한 욕망과 본능은 인간을 끝없이 이어진 길을 따라 더 넓은 세상으로 이끌고, 컴퓨터는 인간의 출발을 돕는 조력자다. 발전의 여정을 가는 도중 길을 잃기도 하고, 넘어지기도 하겠지만 인간과 컴퓨터가 무사히 일어서기를 바라며 한 마디를 마친다.

친구야, 생각해보게나.
철길 말일세.
두 개의 선이 나란히 가고 있지
가끔씩 받침대를 두고 말일세.
다정한 연인들 같다고나 할까?
수많은 돌들은 그들이 남긴 이야기고 말일세.
그 철길 위로 열심히 달리는 기차를
생각해보게나
두 선로는 만날 수 없네.
그러나 가는 길은 똑같지.
어느 쪽도 기울어져서는 안되지.
거리 간격이 언제나 똑같지 않았나.
언제나 자리를 지켜주는 것을 보게나.

친구야!
우리의 우정은 철로일세.
물론 자네가 열차가 되고 싶다면
할 수 없네. 그러나 열차는 한 번 지나가지만
철길은 언제나 남는 것이 아닌가?
열차가 떠나면 언제나 아쉬움만 남지.

친구야, 우리의 길을 가세.

철길이 놓이는 곳에는 길이 열리지 않나.

-

컴퓨터가 소통의 통로로 자리매김 한 데에는 클로드 새넌의 ‘정보이론’이 결정적이었다. 통신의 전체적인 틀을 만들고, 통신의 한계가 물리(하드웨어)에 있지 않고 정보량(소프트웨어)에 있다는 새로운 시각을 제시했다. 이를 통해 정돈되지 않았던 통신의 개념과 문제가 명료해지고, 통신 기술이 정보를 중심으로 발전하기 시작했다.

시대의 변화 속 중요한 업적을 세운 새넌과 튜링은 1943년 제 2차 세계대전 중 만남을 가졌다. 각자 암호 해독에 관한 프로젝트에 몸담고 있으며 그에 관한 내용은 배제하고, 순수한 학문적 교류를 했다고 전해진다. 그 당시만큼은 서로 철길 같은 존재이지 않았을까. 간격을 두고 있지만, 같은 곳을 바라보고 있었다. 비슷한 관심사를 공유하며 그들은 그들의 길을 갔고, 디지털 문명이라는 새로운 길을 열었다. 글을 읽으며 당신과 같은 곳을 바라보는 철길 같은 사람이 떠올랐기를 바라며 한 마디를 마친다.

그 도구의 응용

네가 황제다

알렉산드로 푸쉬킨

너의 자유로운 혼이 가고 싶은 대로
너의 자유로운 길을 가라.
너의 소중한 생각의 열매들을 실현하라.
그리고 너의 고귀한 행동에 대한 아무런
보상도 요구하지 말아라.
보상은 바로 제 자신에게 있는 것이다.
네 자신이 너의 최고 심판관이다.
다른 누구보다도 엄격하게 너는 제 자신의
작품을 심판할 수 있다.
너는 네 작품에 만족하는가?
의욕 많은 예술가여!
네가 황제다. 고독하게 살아라

-

마지막으로 전하고픈 말이다.

우리는 자유로운 존재이고, 소중한 생각을 하는 사람들이다. 사유와 실천이라는 고귀한 행동은 그 자체만으로 보상이다. 그 누구도 우리를 마음대로 재단할 수 없지만, 우리는 스스로 엄격하게 평가내리기도 해야 한다. 스스로에게 만족하는 일은 중요하기 때문이다. 의욕이 적던, 많던 고독하게 자유로운 길을 걷는 사람이다.

이번 챕터를 끝까지 읽어주셔서 감사한 마음을 전하며 한 마디를 마친다!

완벽한 하인

이전형

오늘날 컴퓨터는 우리 몸과 점점 가까워지고 있다. 스마트폰과 시계는 이미 손 안의 훌륭한 컴퓨터로 자리매김했다. 언제 어디서나 사용할 수 있는 컴퓨터 덕분에 우리의 삶은 획기적으로 변할 수 있었고, 지금 이 순간에도 하루가 다르게 변화하고 있다.

그렇다면 그 다음 순서는 무엇일까? 최근 크게 관심을 끌고 있는 분야는 스마트 글라스, 이른바 눈 위에 쓰는 컴퓨터이다. 이미 구글, 페이스북, 샤오미, 삼성전자 등 국내외 수많은 IT 회사들이 개발에 나서고 있다. 과연 스마트 글라스는 우리의 삶을 또 어떻게 바꿀 수 있을까? 여기서는 곧 다가올 스마트 글라스의 모습을 상상해보자.

자율주행차 역시 빠르게 발전하고 있는 기술인데, 이는 곧 이동하는 컴퓨터가 등장하게 된 것이라고 할 수 있다. 자율주행차가 가져올 자동차와 사람의 관계, 그리고 이동방식의 변화에 대해 예상해보도록 하자.

세상을 보는 눈

우리는 눈을 통해 수많은 정보들을 받아들인다. 시각은 인간의 여러 감각 중에서도 가장 중요하다고 여겨지며, 눈은 바로 이 시각을 담당하는 기관이다. 기술이 발전함에 따라 TV, 컴퓨터, 스마트폰 등 다양한 전자기기들이 등장하였는데, 이 덕분에 우리의 눈은 갈수록 혹사당하고 있다. 그렇다면 스마트 글라스는 과연 어떨까?

인간의 눈은 다양한 기능만큼이나 한계를 가진다. 우리는 밝은 곳에서는 명확하게 볼 수 있지만 어두운 곳에서는 아주 흐릿하게 볼 수 있다. 우리는 가까이 있는 물체를 볼 수 있지만 더 가까이 있는 물체는 볼 수 없고, 멀리 있는 물체를 볼 수 있지만 더 멀리 있는 물체는 볼 수 없다. 우리는 지금, 바로 앞을 볼 수 있지만 얼마 전의, 혹은 옆이나 뒤는 볼 수 없다. 우리는 당장 보이는 것만 볼 수 있고, 그 이상의 것들에 대해서는 볼 수도, 알 수도 없다. 스마트 글라스는 이러한 인간의 시각을 획기적으로 보조하고, 확장할 수 있다.

21세기 천리안, 스마트 글라스

스마트 글라스는 우리의 눈과 함께 본다. 단순히 보는 것만이 아니라 볼 수 없는 것까지를 볼 수 있게 해준다. 멀리 있는 것을 가까이 있는 것처럼, 가까이 있는 것을 더 크고 자세하게, 어두운 곳에서도 마치 환한 것처럼, 옆이나 뒤와 같이 잘 보이지 않는 곳까지도 속속들이 볼 수 있게 해준다. 이렇듯 시각의 기능을 전반적으로 확장시켜주기 때문에 안경은 이제 단순히 기존에 시력이 좋지 않아서 쓰던 보조기구에서 벗어나, 첨단기구로써 자리매김한다. 천리안,

투시와 같이 꿈만 꾸던 일들도 이제는 더 이상 상상이 아닐 수 있다.

스마트 글라스는 다양한 형태로도 만들어질 수 있다. 물론 아직까지는 그 명칭이 뜻하는 바와 같이 글라스, 즉 안경의 형태이지만, 굳이 안경의 형태에만 고정되어 있을 필요가 없다. 앞으로는 안경 이외에도 렌즈의 형태를 띠는 스마트 렌즈나 수술을 통해 영구적으로 눈에 칩을 이식하는 방법 등으로 다양하게 사용될 수 있다. 우리들의 신체와 주변을 느낄 수 있는 여러 가지 감각들과 상호작용할 수만 있다면 스마트 글라스, 스마트 워치 등 인간과 함께하는 컴퓨터가 얼마든지 만들어질 수 있다.

일상을 기록하고 기억하는 안경



그림 9 Ray-Ban Stories

Meta 에서는 최근 Ray-Ban Stories²⁾ 라는 스마트 글라스 제품을 만들어 공개하였다. Ray-Ban Stories 를 사용하면 전화통화를 할 수 있으며 사진을 찍거나 음악을 듣는 활동이 가능하다. 이처럼 스

마트 글라스는 봄과 동시에 기록할 수 있다. 한 번 보고 지나쳐버리는 우리의 눈이 가지지 못했던 완전히 새로운 기능이다. 우리의 눈이 뇌에 일시적으로 정보를 전달한다면, 스마트 글라스는 지속적으로 정보를 수집하고 기록한다. 우리는 언제든지 이렇게 기록된 우리의 생활을 되돌아볼 수 있다. 다시 살펴보기를 원하는 시간대만 선택하면, 스마트 글라스가 알아서 자신이 기록했던 정보를 눈앞에 생생하게 비추어준다. 이 과정을 통해 우리는 특정 장면을 생생하게 다시 볼 수 있다. 추억은, 이제 되감아보기로 탈바꿈한다.

스마트 글라스에 탑재된 소프트웨어의 도움을 받는다면 너와 나의 삶을 보다 쉽게 공유할 수 있다. 이미 우리는 페이스북, 유튜브, 인스타그램 등 각종 SNS 매체를 통해 각자의 삶을 공유하고 있다. 하지만 스마트 글라스를 통해서는 훨씬 더 자유롭게 행동하며 모두 각자의 시점을 완벽하게 그대로 재현하여 공유할 수 있다는 것이 특별하다. 이제 모든 사람들이 서로의 현실을 보다 쉽고 간편하게 공유할 수 있으며, 이를 더욱 실감나게 엿보면서 즐길 수 있다.

눈 위의 컴퓨터, 눈 위의 프로그램

스마트 글라스는 현실과 가상의 경계를 허무는, 증강현실의 세상을 열 수 있을 것으로 기대되는 제품이다. 스마트 글라스는 단순히 물리적인 안경만을 뜻하는 것이 아니다. 이와 연동되는 다양한 종류의 소프트웨어 서비스까지를 포함하는 개념이라고 할 수 있다. 대표적으로는 스마트 글라스를 통해 다양한 정보들을 시각적으로 보여줄 수 있다. 정보제공은 기존의 컴퓨터나 스마트폰이 제공하던 정보들을 시각적인 형태로 바꾸어 눈에 직접적으로 노출시키는 것이다. 현재 위치에 대한 정보, 목적지까지의 경로에 대한 정보, 근

치 건물들에 대한 정보, 바로 앞 매장에 대한 정보, 현재 보고 있는 물건에 대한 정보 등 기존에 우리가 검색을 통해 찾아야만 했던 정보들을 스마트 글라스를 통해서도 자동으로, 시각 정보와 함께 실시간으로 확인할 수 있다는 것이 가장 큰 효과이다. 자동번역은 또 다른 예로, 여러 가지 언어로 표시된 글자를 사용자가 설정한 언어에 맞추어서 번역하여 우리 눈으로 보여줄 수 있다. 이 기능을 통해 우리는 전 세계 어디에서라도 언어에 대한 장벽 없이 생활할 수 있다.

스마트 글라스가 가진 다양한 기술을 효과적으로 활용하기 위해서는 사람들의 개인정보를 안전하게 처리할 수 있는 기술이 반드시 마련되어야 한다. 우선 스마트 글라스를 통해 우리는 각자 서로의 모습을 자연스럽게 인식하게 되므로 이를 특정하지 않아야 한다. 만약 이러한 기술이 없다면 어느 누구나 상대방 모르게 촬영하는 등 신상이 무분별하게 노출될 수 있다. 또한 스마트 글라스를 통해 사용자 개인의 생활이 밀접하게 저장될 수 있으므로 사용자의 정보 또한 철저히 관리되어야 한다. 수많은 개인정보를 저장하고 보관하면서도 이를 사용자가 원하는 경우에는 언제든지 꺼내어 활용할 수 있어야 한다는 점에서 지금보다도 더 발전된 형태의 보안 기술이 필요하다. 오늘날 개발 중인 동형암호 기술이 더 발전하여 상용화된다면 스마트글라스에서 개인정보를 직접적으로 다루어 활용할 수 있는 가능성도 더욱 넓어질 것이다.

스스로 운전하는 자동차, 그 너머

자동차는 가장 중요한 이동수단 중 하나다. 비교적 가까운 거리를

빠르게 이동하는 수단으로써 자동차는 필수적인 역할을 담당한다. 처음에는 자동으로 움직인다는 것만으로도 충분히 매력적이었겠지만, 이제는 시간이 충분히 흘렀다. 이제는 단순히 이동한다는 것만으로는 조금 심심할 따름이다. 우리는 생각한다. 이동하는 것은 기계에게 맡기고, 그 시간 동안 사람은 조금 더 재미있는 것을 할 수 있지 않을까?

그리고 드디어 자율주행차가 등장함으로써 이러한 기대를 충족시켜 줄 수 있을 것으로 보인다. 하지만 이미 언급했듯이 자율주행차가 등장하는 것이 단순히 인간에게 더 편해진다고만 생각하는 것은 지나치게 단순한 생각이다. 운전해서 이동하는 능력은 이동수단으로서 갖추어야 할 본질적이면서도 가장 핵심적인 기능이다. 지금까지 이 기능을 인간이 일부 담당했던 것이고, 이제 이 기능을 온전히 기계에게 넘기려 하는 것이다. 자동차가 스스로 운전하는 능력까지 갖춘다는 것은, 자동차의 본질이 단순한 이동수단에서 완전히 다른 무엇인가로 변화한다는 것을 의미한다.

그렇다면 과연 우리는 자동차가 인간에게 앞으로 어떤 존재이기를 원하는가?

이동하는 컴퓨터

자율주행차는 단순히 자동차만을 의미하는 것이 아니다. 자율주행차에는 인간에게 다양한 편의기능을 제공하는 자율주행차만의 소프트웨어, 자율주행차만의 서비스들이 포함된다. 기존의 컴퓨터, 스마트폰 등과는 달리 자율주행차이기 때문에 가능한 새로운 형태의 서비스가 만들어질 것이다. 자율주행차 역시나 같은 컴퓨터이지만, 중

류가 사뭇 다르다.

일단 기본적으로 자율주행차는 지금까지 인간의 주행을 완전하고 완벽하게 대신한다. 여기서 완전하다는 것은 모든 경우의 수를 포함한다는 의미이고, 완벽하다는 것은 완전한 모든 것들에 대해 오류가 없다는 뜻이다.

자율주행차가 많아질수록 자율주행차들마다 서로의 정보를 공유할 수 있게 되어 일종의 거대한 지도가 실시간으로 만들어진다고 생각할 수 있다. 이를 바탕으로 지금보다도 훨씬 더 상세한 지도가 자동적으로 만들어지고 유지될 수 있으며, 도로와 교통 정보 등으로부터 훨씬 효율적인 이동이 가능해진다. 이는 기존의 내비게이션과 지도 정보의 확장이라고 볼 수 있다.

드디어 컴퓨터의 도움을 받아 길고도 길었던 운전으로부터 인류는 해방되었다. 그렇다면, 그 다음은?

바퀴 달린 집

자율주행차 시대의 자동차는 이동수단이 아닌 하나의 공간이라는 의미가 더욱 부각될 것이다. 즉 자율주행차 자체가 하나의 플랫폼이 되어 우리들이 그 플랫폼 안에서 다양한 콘텐츠를 소비할 수 있게 될 것이다.

기본적으로 출발지와 목적지를 입력하면 경로를 분석하여 다양한 정보를 제공해주며, 이를 실제 주행상황과 적절히 결합하여 더욱 효과적으로 사람들에게 전달할 수 있다. 예를 들어 현재 자율주행

차가 움직이고 있는 공간에 관한 여러 가지 정보들을 실시간으로 찾아볼 수 있다.

이동하는 시간 동안에는 내부 공간의 디스플레이를 활용하여 다양한 활동이 가능하다. 각종 영상이나 영화를 볼 수도 있고, 게임을 하는 등의 다양한 콘텐츠를 즐길 수 있다. 이 모든 일들은 인간이 운전으로부터 해방되어 이른바 남은 시간을 적극적으로 활용할 수 있게 됨으로부터 가능해진 것이다.

이렇듯 자율주행차는 다양한 편의기능들을 작동시키며 움직이는 컴퓨터가 될 것이며 이를 통해 우리는 단순히 이동 시간을 흘러가던 것에서 벗어나 자율주행차 안에서 다양한 활동을 통해 시간을 소비할 수 있게 된다. 이러한 맥락에서 자율주행차는 일종의 주거공간이 될 수도 있다.



3)

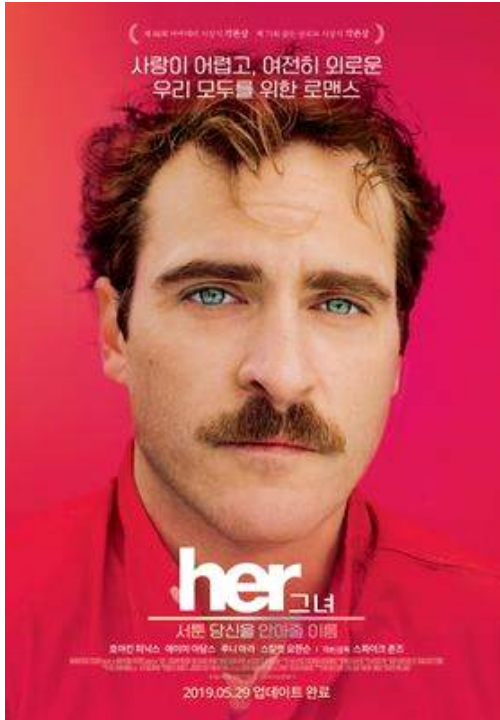
달려왔던 길, 달려나갈 길

오늘날의 도로는 인간이 주행하기 위한 길이다. 따라서 인간이 운전을 하는 데에 도움을 주기 위한 각종 표지판과 차선 등의 표시를 통해 다양한 종류의 정보를 전달하고 있다. 하지만 자율주행차가 대중화된다면 지금과 똑같은 방식의 도로가 더 이상 필요하지 않을 수도 있다. 만약 달라진다면 어떠한 것들이 달라질 수 있을까?

일단 기존의 도로에서 가장 핵심적으로 정보를 전달하는 수단은 빛이다. 빨강, 노랑, 초록 색깔의 3가지 종류의 빛 신호를 통해 인간은 멈추어야 하는지, 곧 멈추어야 하는지, 아니면 앞으로 나아가도 되는지를 판단한다. 하지만 날씨가 좋지 않거나 바로 앞에 커다란 화물차가 있는 등 여러 상황에 따라서는 신호등의 빛이 잘 보이지 않을 수도 있다. 이미 인간은 여러 가지 종류의 신호들을 사용하여 다양한 기술을 개발하여 사용하고 있는 만큼 만약 자율주행차가 도입된다면 굳이 빛의 색깔을 이용하여 신호를 주고받는 상황이 더 이상 필요하지 않을 수도 있다. 예를 들어 레이저와 같은 신호를 이용할 수 있다. 따라서 지금의 도로가 운전자와 보행자라는 양쪽의 인간을 모두 고려하여 설계한 결과물이라면 앞으로 자율주행시대의 도로는 보행자를 더욱 고려하여 만들어질 수 있을 것이다. 차도에서는 자율주행차 내부의 컴퓨터가 더욱 인식하기에 편리한 신호를 사용하고, 도보는 인간이 더 편리하고 안전하게 자동차로부터 분리되어 걸어다닐 수 있게 설계하면 된다.

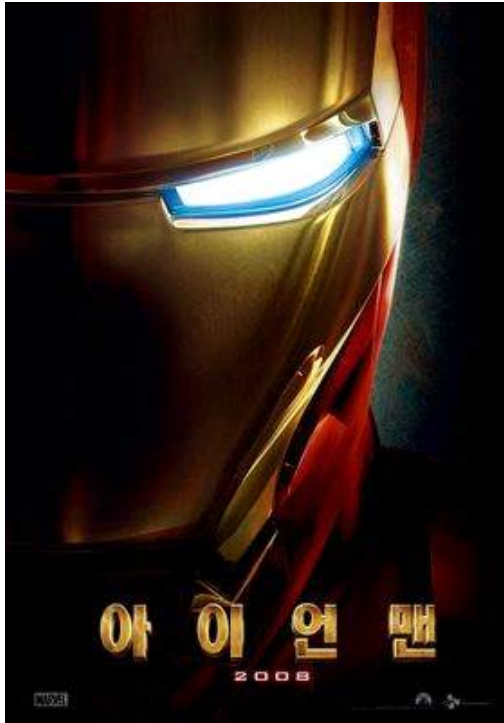
컴퓨터가 만들어지고 사용된 것은 불과 100년이 채 되지 않았다. 인간이 사용하는 다른 여러 도구들에 비교한다면 그 역사가 굉장히 짧다고 할 수 있다. 그러나 짧은 역사에 비해 컴퓨터는 인류에게 아주 많은 분야에서 아주 커다란 영향을 주었다. 이제는 컴퓨터과학을 연구하는 학자가 아닌 일반인들도 컴퓨터 없이는 일상생활마저도 거의 불가능할 정도가 되었다. 그렇다면 앞으로 컴퓨터는 또 어떤 방식으로 우리들과 함께 발전해나갈 수 있을까?

미래를 예측하는 것은 어렵지만 앞으로 일어날 일을 미리 상상해본다는 것은 그만큼 흥미진진하다. 미래를 상상하는 여러 방법들 중 가장 실감나는 방법 중 하나가 바로 영화를 만들어 실제와 같은 느낌을 불러넣는 것이다. 여기서의 여러 영화들에 나타나는 컴퓨터과 학이 여는 세계의 모습들에 대해서 알아보기로 한다. 구체적으로 살펴볼 영화들은 Her, Iron man, 마이너리티 리포트, 레드 플레이어 어 윈, 매트릭스 이다. 우선 각 영화에서 나타나는 미래 사회의 모습이 어떠한지를 간략하게 알아볼 것이다. 이를 통해 컴퓨터과학이 앞으로 어떻게 발전할 것이며, 그 결과 미래에 우리 인간에게 어떠한 변화를 가져다 줄 것인지 하나씩 분석해본다. 또 각 영화들이 그리는 미래가 어떠한지를 서로 비교해보고, 이를 바탕으로 미래의 모습은 어떠한 것인지에 대해서 한번 예측해보자.



4)

영화 Her (2013) 과 Iron man (2008) 에서는 주인공과 함께하는 인공지능 시스템이 등장한다. 우선 Her 에서의 인공지능은 사용자의 비서이자 한 걸음 더 나아가 동료와 같은 역할을 한다. 기본적으로 사용자가 하루 동안 해야 할 일들을 체계적으로 관리해주며 사용자의 요청에 따라 각종 필요한 정보를 제공해준다. 일상에서 마치 실제 사람과도 같이 맥락이 있는 대화를 주고받는 것도 가능하다. Iron man 에서의 시스템은 여기에 더해 주인공 Iron man이 만든 로봇 시스템을 대신해서 관리해준다. 시스템을 관리하는 시스템인 셈이다. 이런 AI 시스템 덕분에 Iron man이 슈퍼히어로가 될 수 있었다고 보아야 할 것이다.



5)

아직까지 컴퓨터에게 명령을 내릴 수 있는 방법은 대부분 키보드를 이용하는 것이다. 우리는 손가락으로 키보드를 내리쳐 한 글자씩 컴퓨터에 입력한다. 그러나 이 방법은 물리적인 동작을 통해 문자를 입력해야 한다는 번거로움이 존재한다. 키보드 배열을 암기해야 하는 것 또한 초기 진입장벽이다. 때문에 최근에는 음성인식기술을 활용하여 이러한 불편함을 극복하고, 사람과 컴퓨터의 소통방식을 한 단계 더 발전시키기 위해 노력하고 있다. Her 와 Iron man에서 보여주는 미래는 컴퓨터가 어떻게 개인의 삶을 더 풍성하게 만들어줄 수 있는지를 잘 보여주고 있다. 우리는 더욱 긴밀하고 효과적으로 컴퓨터와 소통하게 될 것이고 이를 잘 활용하는 능력이 중요해질 것이다. 비록 아직은 문자를 입력하여 컴퓨터에게 명령을 내

리는 것만이 가능하지만 앞으로는 음성이나 움직임 등 다양한 방법을 통해 컴퓨터가 인식할 수 있는 신호를 전달할 수 있게 될 것이다. 컴퓨터 또한 인간이 인식할 수 있는 다양한 종류의 신호를 되돌려주게 될 것이다. 이러한 기술의 발달에 힘입어 인간과 컴퓨터는 보다 더 자연스럽게 편리하게 의사소통할 수 있게 될 것이다.



6)

개인적인 차원을 넘어 사회적인 차원에서도 컴퓨터는 충분히 활용될 수 있다. 이미 컴퓨터를 각종 문제를 해결하는 데 사용하는 분야가 여럿 존재한다. 예를 들어 기상청에서는 이른바 슈퍼컴퓨터를 활용하여 내일, 모레 등 앞으로 다가올 날들의 날씨가 어떠한 지를

미리 예측하여 사람들에게 알려주는 역할을 수행하고 있다. 수많은 데이터를 바탕으로 계산하고 그 통계를 이용하여 결과를 예측하는 것은 이미 대부분의 영역에서 컴퓨터가 인간을 대신하고 있다.

영화 마이너리티 리포트 (2002) 에서는 미래 우리 사회에서 유용하게 사용되는 범죄 예측 시스템이 등장한다. 정확한 원리와 작동방식 등에 대해서는 나오지 않지만 기본적으로 컴퓨터가 알고리즘을 통해 특정 지역에서 범죄가 발생할 것인지를 계산하여 범인과 같은 범죄와 관련된 다양한 정보를 사전에 경찰에게 알려주는 방식이다. 경찰은 이를 바탕으로 쉽고 정확하게, 범죄가 미처 발생하기도 전에 해당 범죄에 대응할 수 있게 된다. 컴퓨터의 도움을 받아 사회의 치안을 보다 더 효율적으로 안전하게 유지할 수 있게 된 것이다.

마이너리티 리포트에서는 경찰들이 컴퓨터의 범죄 예측을 상당히 절대적인 지표로써 활용한다. 하지만 만약 앞으로 더욱 정교한 범죄 예측 시스템이 만들어진다고 하더라도 실제로 그 예측을 완전하게 믿을 수 있을지는 더 두고 보아야 할 일이다. 만약 그 예측이 확률을 기반으로 한다면 아무리 높은 정확도를 가지는 시스템이라고 하더라도 과연 무조건적으로 그 결과를 수용할 수 있을까? 더군다나 불완전하고 기계적이지 않은 인간에 대한 예측이라면 신뢰하기에 더욱 어려울 수 있다.

결국 컴퓨터는 인간이 지시한 일을 수행할 뿐이기에, 컴퓨터가 알려주는 결과를 어떤 방식으로 이용하는지는 온전히 인간의 손에 달려있다. 컴퓨터를 완전무결한 기계로 믿고 의존할 것인지, 아니면 그 결함을 충분히 인지하여 조금 더 신중하게 판단할 것인지는 조금 더 섬세한 관찰과 사회적인 합의가 필요한 부분이다. 결국 사람

이 만든 결과물의 하나로서 컴퓨터가 완벽하지 않다는 것은 사실이지만, 그렇다고 해서 완벽하게 기계적인 계산을 수행하는 컴퓨터를 믿지 않는 것은 인간의 논리 체계의 근간을 신뢰하지 않는다는 것과 같으므로 적당한 수준에서 받아들이는 것이 필요할 것이다.



7)

영화 레디 플레이어 원 (2018) 에서는 가상현실에서 벌어지는 각종 모험을 다루고 있다. 레디 플레이어 원은 사람들이 활동하고 즐길 수 있는 현실과도 같은, 그러나 현실과는 또 다른 하나의 현실이 우리와 함께하는 세상을 그린다. 가상현실에서는 사람들이 자신의 아바타로 대체되며, 각자가 모두 주인공이 되며 각각은 다양한 모습과 개성을 지닌 게임 캐릭터이다. 여러 플레이어들은 자신들이 원하는 게임을 실재감 있게 즐길 수 있

며 게임을 끝내고 난 후에는 다시 현실로 빠져나오게 된다. 이러한 레디 플레이어 원의 세상은 가상현실이 발달함에 따라 이 세상이 여러 개 존재하는 것과 같이 변화한 모습을 그리고 있다. 마치 다중우주 세계관과 비슷하다. 차이점이라면 가상현실은 인간이 직접 만들어낸 또 하나의 우주라는 점이다. 레디 플레이어 원은 오늘날 meta, roblox, microsoft 등의 회사들이 주축이 되어 개발하고 있는 메타버스 혹은 가상현실 등의 기술이 발전한 미래를 사실감 있게 그려낸다.

가상현실은 그 가능성으로 인해 많은 주목을 받는다. 아직까지 우리가 접할 수 있는 가상의 정보는 2차원 평면을 통해 영상으로 전달되는 것이 대부분이다. 지금까지는 2차원에서 보다 더 현실적인 느낌을 전달하기 위한 방법으로 화면의 크기가 점점 커지고 각 부분들을 더 섬세하게 표현하는 방향으로 기술이 발전하였다. 하지만 이 방법만으로는 결국 3차원의 느낌을 주는 것은 불가능하다. 컴퓨터 기술을 현실과 조합하여 보안한 것이 가상의 실체와 현실을 결합한 증강현실이다. 증강현실에서도 현실에서 인간이 가지는 감각과 인식의 한계를 훌륭하게 극복하여 여러 가지 편의를 가져다 줄 수 있다. 가상현실은 이에 한걸음 더 나아간 것으로 인간이 스스로 새로운 세상을 하나 더 만들어낸 것과도 같다. 가상현실의 도입으로 각광받는 부분은 영화 레디 플레이어 원에서와 같이 새로운 인간의 놀이공간으로 활용하는 것이다. 가상현실의 실재감은 인간의 몰입을 현실에서와 가장 유사한 수준으로 끌어올린다. 그러면서도 정말 현실이 아니기 때문에 사용자들은 훨씬 더 자유롭게 행동하는 것이 가능하고 마치 게임을 하는 것과 같이 현실을 즐길 수 있게 되는 것이다.

영화 매트릭스 (1999) 는 AI가 지배하는 먼 미래를 그리고 있다. 지금은 인간이 AI를 활용하여 각종 문제를 해결하는 도구로 사용

을 하는 것에 비해 매트릭스의 세상에서는 AI가 인간을 그들의 생존을 위해 사용하는 도구로 그려진다. 인간과 AI의 관계가 완벽하게 역전된 것이다. 이는 컴퓨터 과학이 발전함에 따라 AI가 인간의 능력을 뛰어넘게 되고 그 과정에서도 컴퓨터과학이 가장 위험한 방향으로 발전하게 되는 결과를 가정한 것이다. 매트릭스의 세계에서는 더 이상 인간의 역사가 아닌 컴퓨터 과학의 역사에서 인간이 일종의 부품으로서 존재할 뿐이다. 인간은 AI가 만들어낸 가상현실을 마치 현실인 것처럼 인지하면서 살아가는데, 이는 영화 레디 플레이어 원에서 제시하는 가상현실과는 완전히 다른, 인간이 수동적으로 주입되는 가상의 공간이라는 점에서 그 맥락이 다르다.



8)

오늘날 컴퓨터 기술의 발전으로 인간이 드디어 기계학습, AI 등의 기술을 활용하여 각종 문제를 해결할 수 있게 되었다. 이전까지는 충분한 기술이 뒷받침되지 않아 이론적으로는 가능했으나 실제로는 구현하여 실행하기에는 현실적으로 불가능한 기술이었다. 이제는 AI의 도움을 받아 이전보다도 더 넓은 분야에 컴퓨터를 활용하여 문제를 해결할 수 있을 것으로 기대하고 있다.

컴퓨터 기술의 발전과 그로 인해 인간이 얻는 편리함만큼이나 그 부작용에 대한 사람들의 걱정 또한 항상 큰 주목을 받는다. 사람들의 걱정은 컴퓨터 기술이 거꾸로 인간에게 해를 끼치는 방향으로 발전하지는 않을지에 대한 것이다. 간단히 생각해보면 단순히 인간의 명령에 따라 기계적인 계산을 아주 잘 수행하는 역할을 담당하는 컴퓨터가 그 자체로 인간에게 해를 끼치는 것은 불가능하다. 인간의 명령이 온전하다는 가정에서는 그렇다. 컴퓨터가 가지는 가능성은 결국 명령을 내리는 인간의 선택에 달려있는 것이다.

똑같은 도구를 사용해도 사람마다 어떤 방식으로 사용하는지는 다를 수 있다. 지금껏 인류의 역사에서 수많은 기술들이 출현했고 그 기술들은 각기 다른 방향으로 인간에게 영향을 끼쳤다. 대표적으로 아인슈타인이 발표한 질량과 에너지는 동등하다는 이론은 오늘날 원자력발전의 기본 원리로 활용되지만, 원자폭탄을 만들어 수많은 사람들을 죽이는 데에도 사용되었다. 결국 단순히 컴퓨터 기술을 발전시키는 것에만 집중하는 것이 아닌, 그 기술을 어떻게 활용하고 관리할 것인지에 대한 사회적인 합의를 만들어낸 것도 매우 중요하다고 할 수 있다. 이러한 공감의 과정을 충분히 거치고 난 후에 인간과 컴퓨터과학이 함께 미래를 만들어나갈 수 있을 것이다.

암호 기술로 책을 마치며
강승윤
이건무
이전형

1. 컴퓨터 암호의 기본적인 원리

새님이 발견한 정보이론을 통해 데이터 손실이 없이 온전히 정보를 전달할 수 있음이 알려진 이후, 이제는 시공간을 넘어서서 우리의 삶을 컴퓨터로 공유하는 것이 가능해졌다. 아이러니하게도 이러한 시공간의 공유가 가능해진 이유는 컴퓨터의 ‘한계’와 관련이 있다. NP 문제처럼, 컴퓨터가 완벽하게 풀 수 없는 문제들을 이용해서 사람들 간의 통신에 대한 신뢰도를 확보할 수 있다는 것이다. 예를 들어, 컴퓨터가 ‘현실적인 비용으로 풀 수 없는’ 대표적인 문제인 소인수 분해로 이를 생각해보자. 현재 내가 뉴욕에 사는 누군가와 가위바위보를 하고 싶은 상황이다. 일반적으로는 가위 / 바위 / 보를 입력하는 시간이 두 사람이 완벽히 같을 수가 없기 때문에, 서로 간에 ‘늦게 냈지?’라는 불신이 생길 수밖에 없는 것이다. 이러한 것을 소인수 분해로 해결할 수 있다. 일의 자리가 1인 소수 중 하나를 입력하면 가위, 일의 자리가 3인 소수 중 하나를 입력하면 바위, 일의 자리가 7인 소수 중 하나를 입력하면 보... 이런 식으로 컴퓨터가 인식하도록 알고리즘을 짠다고 해보자. 가위 / 바위 / 보 중 하나가 입력이 되고 나면, 컴퓨터에서 랜덤으로 하나의 소수를 곱해서 상대방에게 전달한다. 그리고 이는, 상대방이 입력한 것이 나에게 전달될 때 비로소 곱해진 소수가 공개되며, 처음 입력한 소

수, 즉 가위 / 바위 / 보 중 하나를 확인할 수 있다는 것이다. 통신 상의 속도 차이로 먼저 소수가 전달된다고 하더라도, 거대한 수에 대한 소인수분해는 현실적으로 하기가 쉽지 않으므로 속도 차이가 발생해도 상관이 없어진다는 것이다.

컴퓨터 암호에서도 이러한 소인수분해와 같은 NP 문제가 잘 사용된다. 예를 들어 내가 전달하고픈 말을 적은 뒤 암호화시키고 싶다고 가정하면, '1'은 소수1에, '2'는 소수2에 이런 식으로 대응이 된다고 규칙표를 만든 뒤, 하나하나의 낱말에다 미리 정해놓은 거대한 소수를 곱해서 암호화를 시킬 수 있는 것이다. 어차피 중간에 가로채진다고 하더라도 소인수분해를 할 수 있는 제대로 된 방법이 없기 때문에. 그러나, 이러한 암호 전달에 대해서 한가지 문제점이 있음을 조금만 생각해보면 금방 확인할 수 있다. 암호를 받은 사람은 이러한 소인수분해과정을 거치지 않아야 암호를 다시 복호화할 수 있다. 즉, 암호를 받은 사람은 이미 어떤 'key'를 알고 있어야 그냥 바로 전달받은 수에서 말하고자하는 바를 캐치할 수 있는 것이다. 이러한 'key'를 전달할 수 있는 방법이 무엇인가. 비밀스런 통신이 필요한 것 아닌가? 'Key'를 전달하려면 비밀스러운 암호 통신이 필요하다는 이야기가 된다는 것이다. 암호통신을 위해서는 'key'가 필요하고, 'key'를 위해서는 암호통신이 필요하다. 어느 것이 과연 먼저일까? 이러한 역설적인 상황을 해결할 수 있을까?

답은 '공개키 암호기술'에 있다. 이 또한 NP 문제를 활용하는 것이다. 아이러니하게도, key 설정을 '공개적'으로 하겠다는 것이 바로 이 공개키 암호기술이다. 나는 서울에, 친구는 부산에 있는 멀리 떨어진 상황이라고 해보자. 나와 친구가 각각 하나씩 숫자를 고르고(x, y), 하나의 공개된 숫자(P)는 공개키 암호 설정을 위해 필

요한 상황이다. 우리에게 필요한 건 어떤 ‘연산’이다. 어떤 연산을 이용해서, 나는 x 와 P 를 계산하여 친구에게, 친구는 y 와 P 를 계산하여 나에게 보낸다. 그 후, 나에게 전달된 계산된 y 와 P 를, 내가 가진 x 와 다시 계산한다. 친구는 그 반대과정을 수행하면 된다. 이렇게 되면, 결과적으로 나와 친구는 모두 x, y, P 를 동일한 연산에 대해 수행하는 것이 되고, 하나의 계산 결과가 공통적으로 나오게 될 것이다. 이것이 우리 둘 사이의 ‘key’가 되도록 하면 된다.

그렇다면 이때 중요한 점이 무엇인가? 바로 ‘어떤 연산’을 정의하는 것이다. 이 연산은 마치 소수 2개의 곱을 하는 듯한, 그런 효과를 가지고 있어야 한다. 무슨 뜻이냐면, x 와 P 를 계산한 결과를 보고서는 절대 다시 x 와 P 를 도출할 수 없도록 해야한다는 것이다. 그래야만이 우리는 공개적으로 암호를 설정하는데도 우리가 말하고자하는 바를 유출하지 않으면서 안전하게 암호화를 시킬 수 있다는 것이다. 따라서 이러한 연산을 정의하는 것이 매우 중요하고, 위 성질을 만족하는 연산의 대표적인 예시가 바로 $(2^x * \text{mod } P)$ 이다. 이를 이용하면 x, y 사이의 교환법칙도 성립함과 동시에 x 와 P 를 결과값에서 분리해낼 수 없기 때문이다.

2. 암호로 구현한 ‘책을 마치며’..

컴퓨터로 암호를 만드는 방식에는 비즈네르 암호, DES, RSA 등등 ... 상당히 다양한 방법들이 존재한다. 앞에서 말했던 공개키 암호 알고리즘은 이들 중 하나로, RSA에 이에 속한다. 이러한 암호 방식들은 비밀키를 이용하여 암호화와 복호화가 전부 가능한 방식들이다. 반면, 오직 암호화만 가능한 방식들도 있다. MD5,

SHA-512 등의 방식이 그런 것들이다. 이런 것을 ‘해쉬 함수를 이용한 암호화’라고 하는데, 어떤 문자를 입력하든 간에 128비트, 즉 32자로 그것을 바꿔준다. 오직 암호화만이 가능하기 때문에 ‘단방향 암호’라고도 부르며, 원칙적으로는 복호화 할 수 없다. 그러나 해쉬 ‘함수’에서 알 수 있듯, 같은 입력값에 대해서는 항상 동일한 출력 값이 나오기 때문에, 이를 이용해서 데이터베이스를 구축해놓을 수는 있다. 즉, ‘hi’라는 것이

‘49f68a5c8493ec2c0bf489821c21fc3b’와 대응된다면, 이를 직접적으로 복호화할 수는 없지만, 여러가지 문자들을 대입해보다가 ‘hi’라는게 ‘49f68a5c8493ec2c0bf489821c21fc3b’와 대응되는 것을 알 수 있는 있다는 것이다.

이러한 것을 이용해서, 우리만의 암호를 만들어보고자 한다. 특히, 암호에 뜻을 넣기 위해, 특별하게 우리가 이 책을 적으면서 느꼈던 점들, 소감들을 암호로 담아보고자 한다. 암호화 방식은 MD5를 사용했으며, 이 책을 읽는 독자들이 내용에 대해 궁금하다면 ‘MD5 복호화 사이트’라고 검색한 뒤 복호화를 해서 우리의 소감을 읽을 수 있을 것이다.

1) 이진문의 소감

```
60acec636777b72e6bd9507b25ca5a81
9e925e9341b490bfd3b4c4ca3b0c1ef2
4eb35c16de4db5fa4cc0725e0d0d7ef0
dd7536794b63bf90eccfd37f9b147d7f
54c84b40e9ff5a31472904a0cd2f0a17
```

0d149b90e7394297301c90191ae775f0
a77b3598941cb803eac0fcdafe44fac9
db82206b1d49042d1a710e9c88c21d36
01b6e20344b68835c5ed1ddedf20d531
af9dc3cd6ce29d560a1e4077694254d9
8fc42c6ddf9966db3b09e84365034357
0fc6de89aa1fcca53e6407f6dcdc2255
8bf8854bebe108183caeb845c7676ae4
524164822d03894ee68052e183e7ea36
21582c6c30be1217322cdb9aebaf4a59
dd7536794b63bf90eccfd37f9b147d7f
a77b3598941cb803eac0fcdafe44fac9
821ae63dbe0c573eff8b69d451fb21bc
2976c4aa9a0c897a24076ef23debb34e
efeb369cccbd560588a756610865664c
2abdb04c9ee61b22a719544a61452fe6
dd7536794b63bf90eccfd37f9b147d7f
c04cd38aeb30f3ad1f8ab4e64a0ded7b
b807023f87e63b8ada92f79f546ff9cc
a3e0f6576a754059784babff593d798a
be5d5d37542d75f93a87094459f76678
a3e0f6576a754059784babff593d798a
8bf8854bebe108183caeb845c7676ae4
0bf83831ea4a1fb853e6d38ff5d43737
97bc592b27a9ada2d9a4bb418ed0ebed

21582c6c30be1217322cdb9aebaf4a59
dd7536794b63bf90eccfd37f9b147d7f
b42dad5453b2128a32f6612b13ea5d9f
a5d7ceb2c59b8cee46c2953fec9abc19
8fc42c6ddf9966db3b09e84365034357
3b2fa9c8faf83bc0b6e18243f066dc6a
f970e2767d0cfe75876ea857f92e319b
dd7536794b63bf90eccfd37f9b147d7f
b42dad5453b2128a32f6612b13ea5d9f
8c00c19e66a7ba9d80bded9460cb1b41
c0db7d4e3cbad2b18c767c659f448a11
01b6e20344b68835c5ed1ddedf20d531
67b0d7c16929369c0ce38e0d9b65c0cc
0d149b90e7394297301c90191ae775f0
addec426932e71323700afa1911f8f1c
9010f5324a95075ef15d7428d9d13c1a
901889f4f34f8ca18ac2f53d1fed346e
1cea5345d284f36245a94301b114b27c
8fc42c6ddf9966db3b09e84365034357
821f03288846297c2cf43c34766a38f7
5cb6f3513617cb22d539f1c4f4a109ea
d529e941509eb9e9b9cfaeae1fe7ca23
8134b84030cca5285ed0e0b31ba06f10
25a9ac406aceb47a0c6cade972bc26fa
8fc42c6ddf9966db3b09e84365034357

664409c6be3cc40a472af29607c8e4d5
d581de187f7b3a4157fc879033c96b26
639bae9ac6b3e1a84cebb7b403297b79
01b6e20344b68835c5ed1ddedf20d531
8fc42c6ddf9966db3b09e84365034357
5cb620c7ff5c90e08734a392ede5af0d
be5d5d37542d75f93a87094459f76678
b04f3782a8cbc2c9ffb3d6d62969b9f2
c5ba2735fc10d446f9d1911d8c254197
53d670af9bb16ea82e7ef41ee23ec6df
91a51175128716cb88eb83b38017c377
8fc42c6ddf9966db3b09e84365034357
4a2c767ca92e768999bdd142c9f75895

2) 이전형의 소감

27baf79af932bd1592a6f4d5d61e96a9
0cc175b9c0f1b6a831c399e269772661
df53ca268240ca76670c8566ee54568a
5d554bc5f3d2cd182cdd0952b1fb87ca
f1425da40a9f2d21ab702a1c7feae026
dd7536794b63bf90eccfd37f9b147d7f
f0441366488bd123be0e5fb7d6c03c2b
d6e047923b2b0f63a7a03c66074a0151
b71e59a3962a4af02e9a7dc3b88ed3ea
37b19816109a32106d109e83bbb3c97d

8bf8854bebe108183caeb845c7676ae4
e329b27943961ccab9dc7a8ac2edb666
8bf8854bebe108183caeb845c7676ae4
df53ca268240ca76670c8566ee54568a
fb5c7f9bb4b32ce2f3bff4662f1ab27b
a181a603769c1f98ad927e7367c7aa51
3b759a9ca80234563d87672350659b2b
8fc42c6ddf9966db3b09e84365034357
256c956c2bc0dae7d8231a70d479b66c
06a209c1f2f572d1ecfaf94e0a73d13b
db88a0257c220dbfdd2e40f6152d6a8d
bbe51039d025b265f9790490781cb201
e4a1704178fcbf5a24d946324af8e66c
65b50b04a6af50bb2f174db30a8c6dad
1cba77c39b4d0a81024a7aada3655a28
ed2b5c0139cec8ad2873829dc1117d50
1c8a2a7c1511c01ee5158ec2ed99b749
14754f13e5280c5d49d2ae536c2d57e2
a77b3598941cb803eac0fcdafe44fac9
8fc42c6ddf9966db3b09e84365034357
f7c0a09108cdf26287c1bc5af2ed1f93
4e70d36d9adc0527adf39bdbe23509e6
60bbfb5a9316723dab1c1d453998fb2c
ff007f8e8939ad6a68b45452a53bd4f1
b8a9f715dbb64fd5c56e7783c6820a61

29e4b66fa8076de4d7a26c727b8dbdfa
c5f6f584b79463f58c223f18fef206ef
8bf8854bebe108183caeb845c7676ae4
7e73d06707f5fb75ec77cc5f2bd9bb92
2be6e0dc940b0a4f282916ddb39af6b
8fc42c6ddf9966db3b09e84365034357
c5fd94950c06088e2163a923ba8c1535
14754f13e5280c5d49d2ae536c2d57e2
be5d5d37542d75f93a87094459f76678
8fc42c6ddf9966db3b09e84365034357
945f3fc449518a73b9f5f32868db466c
c2d630959c5f84ac3484bdd350ae5307
a77b3598941cb803eac0fcdafe44fac9
07ccfe360dce69b84595428e2ec1c1cc
e8a7286d11b01115dd44659e52c40c8c
dd7536794b63bf90eccfd37f9b147d7f
f4540ba5efaeb8411aa4f0ccc1649313
be5d5d37542d75f93a87094459f76678
f016441d00c16c9b912d05e9d81d894d
ef56b0b0ddb93c2885892c06be830c68
8c4291f6956da81515a5c0caec2976d0
965dbaac085fc891bfbbd4f9d145bbc8
01b6e20344b68835c5ed1ddedf20d531
1e8e42b87a65326b98ced7d3af717a72
8fc42c6ddf9966db3b09e84365034357

8518008e8835d42ddb7c83810b5cede0
da907a1b8f74e6922d93b025eecfb852
23a58bf9274bedb19375e527a0744fa9
df53ca268240ca76670c8566ee54568a
be5d5d37542d75f93a87094459f76678
df53ca268240ca76670c8566ee54568a
fb5c7f9bb4b32ce2f3bff4662f1ab27b

3) 강승윤의 소감

c0c5c886246d1d539c668f5a0bebdb45
9e925e9341b490bfd3b4c4ca3b0c1ef2
369480a6c5de07d997c9a69a6c68a8ed
deca2c2b6024183a6b5151ef8586b1d1
86280ac8a7f8881407ba060c76d72d6a
0cc175b9c0f1b6a831c399e269772661
8b0a44048f58988b486bdd0d245b22a8
53d670af9bb16ea82e7ef41ee23ec6df
2c61ebff5a7f675451467527df66788d
1e8e42b87a65326b98ced7d3af717a72
8fc42c6ddf9966db3b09e84365034357
7d793037a0760186574b0282f2f435e7
13b5bfe96f3e2fe411c9f66f4a582adf
f016441d00c16c9b912d05e9d81d894d
29e4b66fa8076de4d7a26c727b8dbdfa
78ee54aa8f813885fe2fe20d232518b9

8bf8854bebe108183caeb845c7676ae4
8a10406b9741d86dc36888de6929534f
dd7536794b63bf90eccfd37f9b147d7f
f0441366488bd123be0e5fb7d6c03c2b
67b0d7c16929369c0ce38e0d9b65c0cc
db88a0257c220dbfdd2e40f6152d6a8d
df53ca268240ca76670c8566ee54568a
3309a7a7941818e131b4dfb9a6349914
e4a1704178fcbf5a24d946324af8e66c
be5d5d37542d75f93a87094459f76678
8fc42c6ddf9966db3b09e84365034357
c83b72dd001482ce10f0b106c7a0ed0e
0d149b90e7394297301c90191ae775f0
c0cef1755aa8e8aa9269079bc6ff043c
8fc42c6ddf9966db3b09e84365034357
256c956c2bc0dae7d8231a70d479b66c
f79af963741bf9ef659952aa0098e16d
dd7536794b63bf90eccfd37f9b147d7f
f0441366488bd123be0e5fb7d6c03c2b
b42dad5453b2128a32f6612b13ea5d9f
0cc175b9c0f1b6a831c399e269772661
ddfe7fc16d097dabb5f57cc0144ef9ba
8aaab33740413527c3d6f2be39c6ed2c
01b6e20344b68835c5ed1ddedf20d531
efb2a684e4afb7d55e6147fbe5a.332ee

0cc175b9c0f1b6a831c399e269772661
821f03288846297c2cf43c34766a38f7
a66db021d84e617bd7667ee300aa29c8
ed2b5c0139cec8ad2873829dc1117d50
8fc42c6ddf9966db3b09e84365034357
851adbefaca9932ef2861a5288d5ceea
be5d5d37542d75f93a87094459f76678
a542e9b744bedcfd874129ab0f98c4ff
d98a07f84921b24ee30f86fd8cd85c3c
8fc42c6ddf9966db3b09e84365034357
19f41045ab92d3898f92cf6b93b9f4da
2d1406762ebaf860e2d9e7bd1c26d855
b807023f87e63b8ada92f79f546ff9cc
f6bc81b1fa9e98a5f645cf8e05202f6f
01b6e20344b68835c5ed1ddedf20d531
c14606b159a391c7c8181af939dcca0
7b34fdbd72fdec596f0c583dd483a0f
be5d5d37542d75f93a87094459f76678
7801ef7e930d2e49cf6e071f5fc43c63
53d670af9bb16ea82e7ef41ee23ec6df
cc23a5af73453c27c71b49531c5d8900
8d966b2253a917086c8604959e152243
90b884933d23c4d57ca01dbe2898d405
be5d5d37542d75f93a87094459f76678
9010f5324a95075ef15d7428d9d13c1a

61e415f686ac4441d253997a3baf8d5a
b43b3777e949a30229c2121fb22a8285
b5b6116573b76baa722edeb026bf5cf7
856822c59edf43fe8223aa585210bace

참고자료

- 1) 이광근, “컴퓨터 과학이 여는 세계”, 인사이트, 2017.2.28.
- 2) Ray-Ban Stories
<https://tech.fb.com/ray-ban-and-facebook-introduce-ray-ban-stories-first-generation-smart-glasses/>
- 3) Business models will drive the future of autonomous vehicles
<https://techcrunch.com/2017/08/25/business-models-will-drive-the-future-of-autonomous-vehicles/>
- 4) Her
<https://pedia.watcha.com/ko-KR/contents/mWyzmNO>
- 5) Iron Man
<https://pedia.watcha.com/ko-KR/contents/m5XMeBW>
- 6) Minority Report
<https://pedia.watcha.com/ko-KR/contents/mJ01n8d>
- 7) Ready Player One
<https://pedia.watcha.com/ko-KR/contents/mOVvJ68>
- 8) The Matrix
<https://pedia.watcha.com/ko-KR/contents/mJ5xwgd>