

# 컴퓨팅(기계계산)의 세계

이광근

서울대학교

컴퓨터공학부

[kwangkeunyi@snu.ac.kr](mailto:kwangkeunyi@snu.ac.kr)

서울대 예비대학 특강

# 컴퓨팅 교육의 목표

- ▶ 컴퓨팅은 모든 공부의 언어, 보편학문의 성격을 가짐
  - ▶ 모든 전공의 성과를 이끌고 (학술 도구)
  - ▶ 모든 전공의 성과위에 자라고
- ▶ 우리를 둘러싼 컴퓨팅을 바라보는 시각을 형성시켜주기
  - ▶ 과학교육이 우리를 둘러싼 자연을 바라보는 시각을 형성해주듯이

# 컴퓨팅 교육의 목표

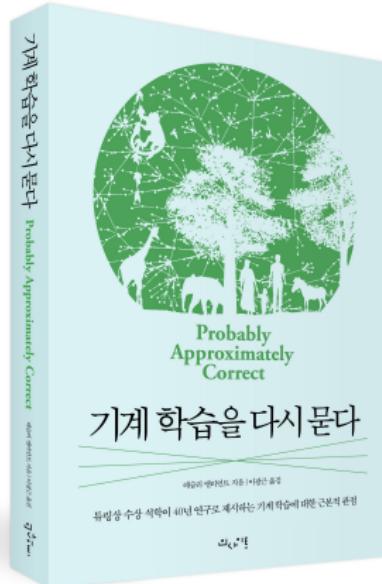
- ▶ 컴퓨팅은 모든 공부의 언어, 보편학문의 성격을 가짐
  - ▶ 모든 전공의 성과를 이끌고 (학술 도구)
  - ▶ 모든 전공의 성과위에 자라고
- ▶ 우리를 둘러싼 컴퓨팅을 바라보는 시각을 형성시켜주기
  - ▶ 과학교육이 우리를 둘러싼 자연을 바라보는 시각을 형성해주듯이

컴퓨터/SW가 그런 자격이 될까? 네.

- ▶ 우리가 벗어날 수 없다
- ▶ 우리가 이용할 방법이 무궁무진
- ▶ 우리능력을 확장시키는 범위가 인류역사에 유래없음
- ▶ 알면 안심할 수 있다

# 가져가실 내용

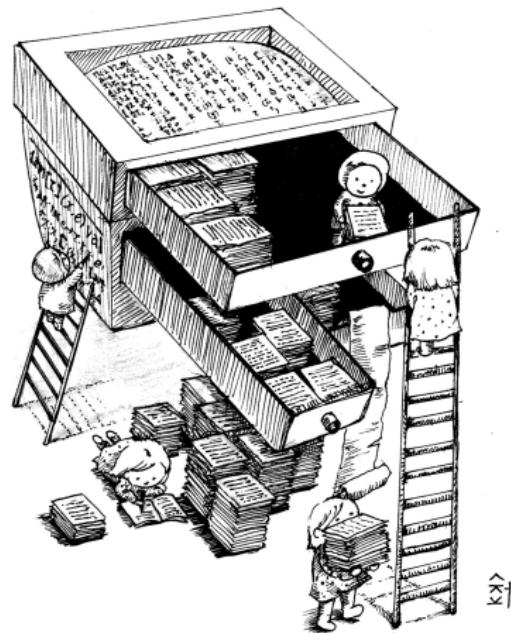
1. 컴퓨터라는 도구의 정의
2. 컴퓨터가 할 수 있는 일의 세계
3. 컴퓨터에게 일 시키는 방법, 두 가지
4. 인공지능시대 컴퓨팅 공부법



# I. 컴퓨터의 정의

## 컴퓨터, 마음의 도구, 만능 기계

컴퓨터 = “기계적인 일”을 해주는 도구



# “기계적인 일”?

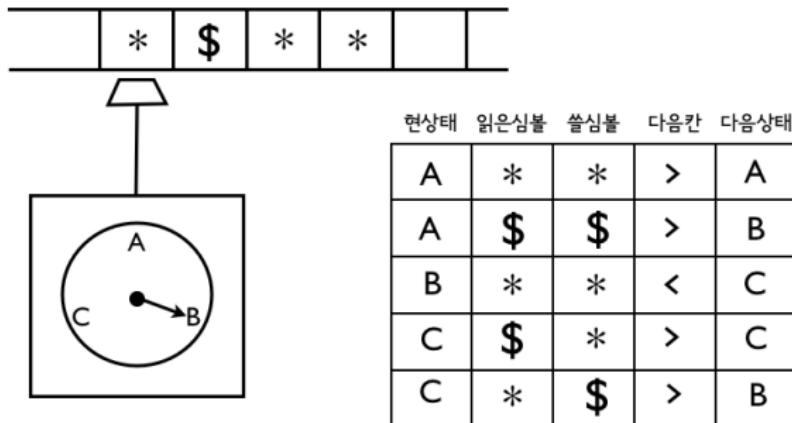
- ▶ 반복적으로 할 수 있는 일?
  - ▶ 반복하는 알맹이 일이 뭐냐고요...
- ▶ 별생각없이 할 수 있는 일?
  - ▶ 그 일이 뭐냐고요...
- ▶ 규칙대로만 하는 일?
  - ▶ 그 일이 뭐냐고요... 어떤 규칙들이냐고요...

애매함 없이 정확하게 정의할 수 있었어요. 그래서  
컴퓨터는 뭐든 기계적인 일이라면 다 해줘,  
라고 말할 수 있게되었답니다.

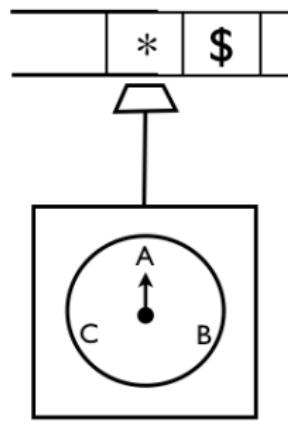
# “기계적인 일”의 그런 정의 (아직 깨지지 않은 파라다임)

튜링기계를 만들어 돌릴 수 있는 일

- ▶ 튜링기계 = 아래의 간단명확한 작동을 하는 네 부품으로 만든 기계
- ▶ 예)

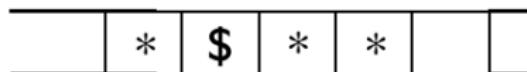
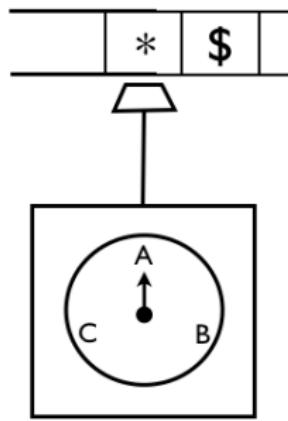


# 어떤 튜링기계의 작동 예시



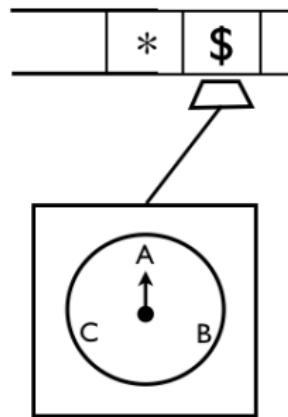
| 현상태 | 읽은심볼 | 쓸심볼 | 다음칸 | 다음상태 |
|-----|------|-----|-----|------|
| A   | *    | *   | >   | A    |
| A   | \$   | \$  | >   | B    |
| B   | *    | *   | <   | C    |
| C   | \$   | *   | >   | C    |
| C   | *    | \$  | >   | B    |

# 어떤 튜링기계의 작동 예시



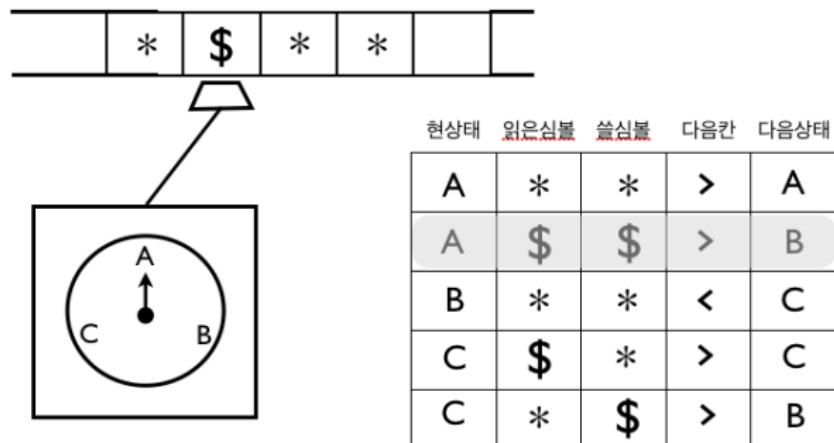
| 현상태 | 읽은심볼 | 쓸심볼 | 다음칸 | 다음상태 |
|-----|------|-----|-----|------|
| A   | *    | *   | >   | A    |
| A   | \$   | \$  | >   | B    |
| B   | *    | *   | <   | C    |
| C   | \$   | *   | >   | C    |
| C   | *    | \$  | >   | B    |

# 어떤 튜링기계의 작동 예시

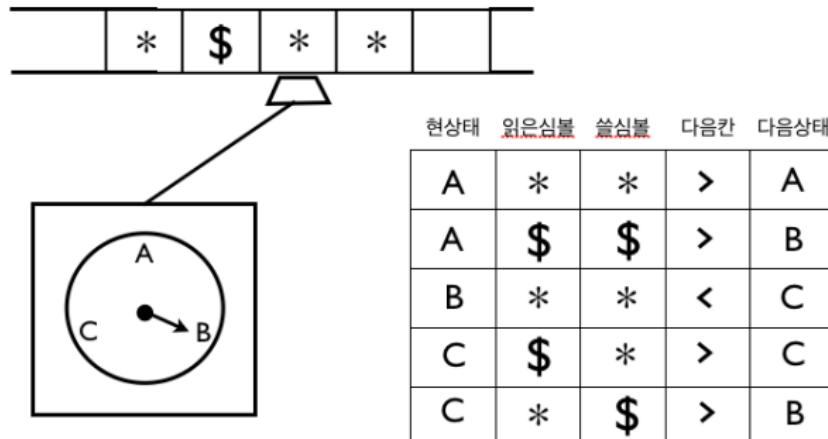


| 현상태 | 읽은심볼 | 쓸심볼 | 다음칸 | 다음상태 |
|-----|------|-----|-----|------|
| A   | *    | *   | >   | A    |
| A   | \$   | \$  | >   | B    |
| B   | *    | *   | <   | C    |
| C   | \$   | *   | >   | C    |
| C   | *    | \$  | >   | B    |

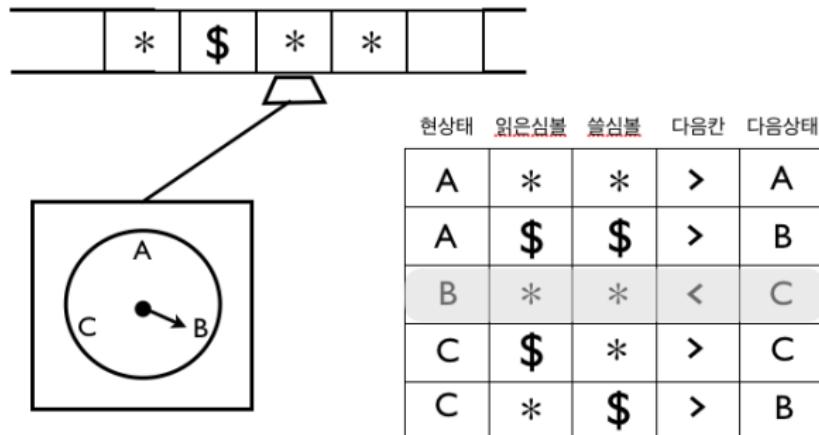
# 어떤 튜링기계의 작동 예시



# 어떤 튜링기계의 작동 예시

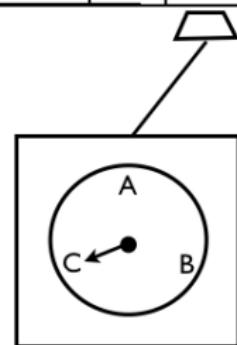


# 어떤 튜링기계의 작동 예시



# 어떤 튜링기계의 작동 예시

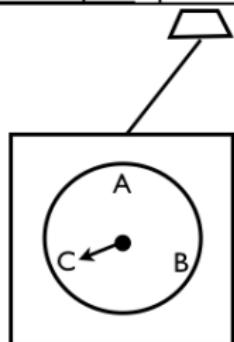
\_\_\_\_\_ | \* | \$ | \* | \* | \_\_\_\_\_



| 현상태 | 읽은심볼 | 쓸심볼 | 다음칸 | 다음상태 |
|-----|------|-----|-----|------|
| A   | *    | *   | >   | A    |
| A   | \$   | \$  | >   | B    |
| B   | *    | *   | <   | C    |
| C   | \$   | *   | >   | C    |
| C   | *    | \$  | >   | B    |

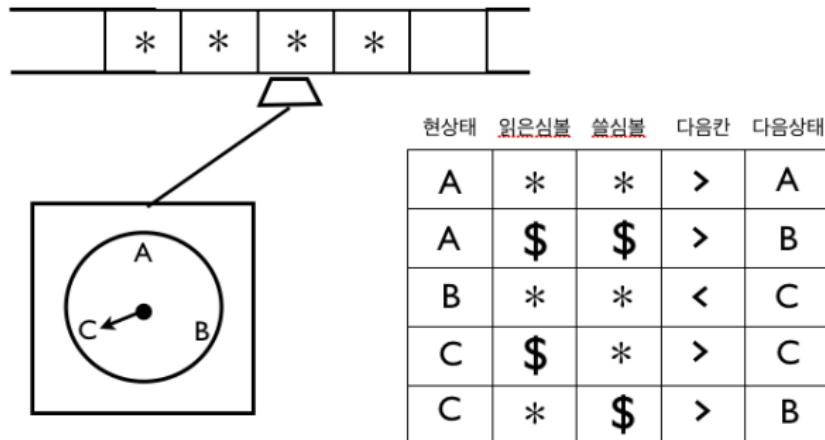
# 어떤 튜링기계의 작동 예시

\_\_\_\_\_ | \* | \$ | \* | \* | \_\_\_\_\_

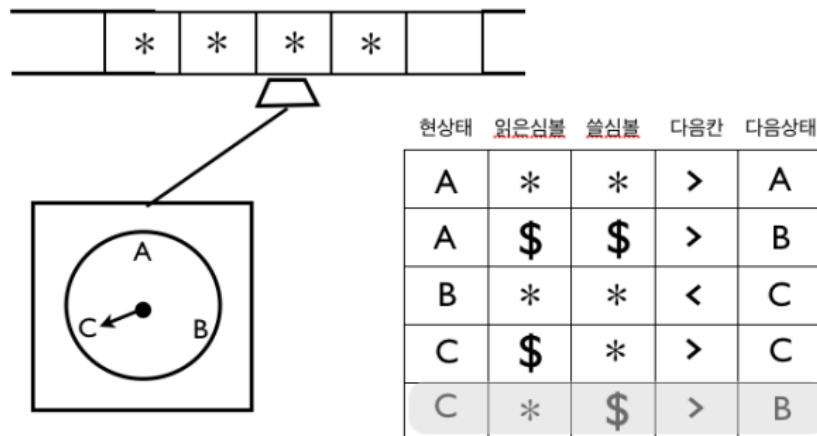


| 현상태 | 읽은심볼 | 쓸심볼 | 다음칸 | 다음상태 |
|-----|------|-----|-----|------|
| A   | *    | *   | >   | A    |
| A   | \$   | \$  | >   | B    |
| B   | *    | *   | <   | C    |
| C   | \$   | *   | >   | C    |
| C   | *    | \$  | >   | B    |

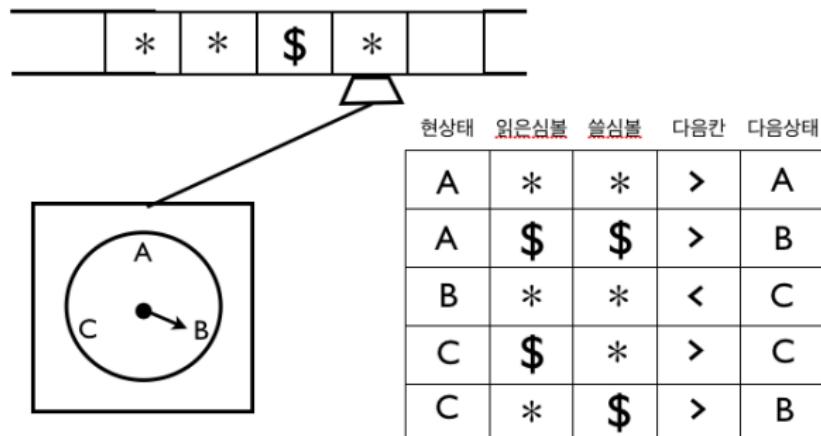
# 어떤 튜링기계의 작동 예시



# 어떤 튜링기계의 작동 예시

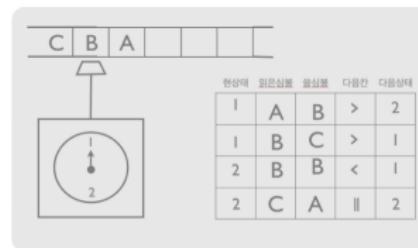
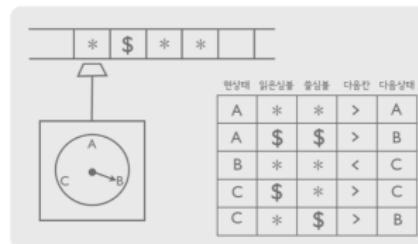


# 어떤 튜링기계의 작동 예시



# 튜링기계 하나 = 기계적인 일 하나

더하기 튜링기계, 카톡 튜링기계, 유튜브 튜링기계, 등등



# 특이한 튜링기계, 만능 튜링기계 = 컴퓨터

만능 튜링기계 *universal turing machine*

어떤 튜링기계, 그러나 “만능”

- ▶ 입력: **튜링기계를** 글로 표현해서 테잎에 받는다
- ▶ 출력: 그 튜링기계의 작동을 그대로 **따라한다**

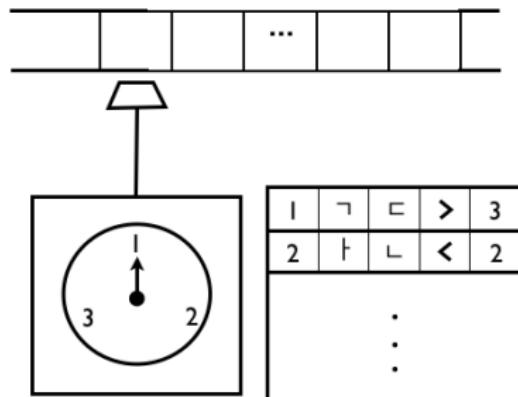
왜 가능하냐고요?

튜링기계가 너무나 간단명확하기 때문

# 특이한 튜링기계, 만능 튜링기계 = 컴퓨터

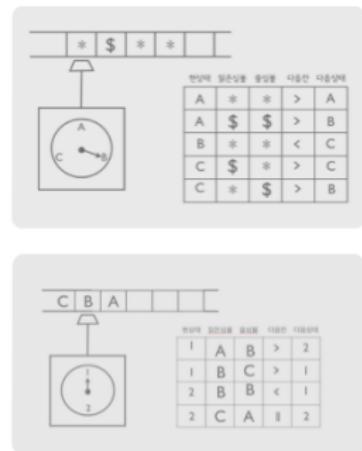
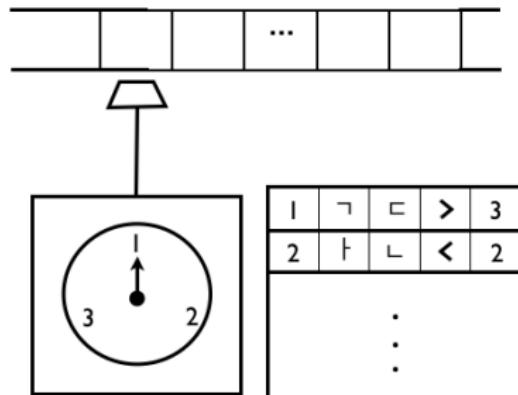
만능 튜링기계 *universal turing machine*

어떤 튜링기계, 그러나 “만능”



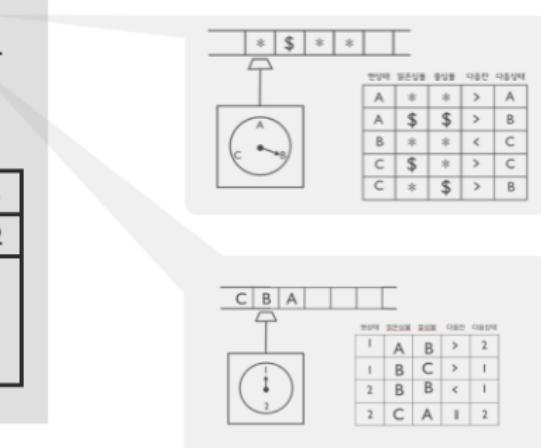
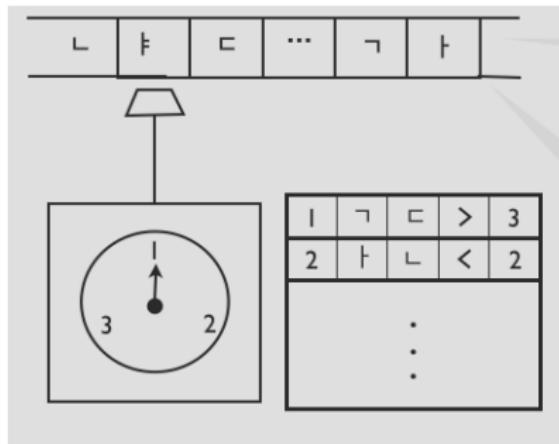
# 특이한 튜링기계, 만능 튜링기계 = 컴퓨터

만능 튜링기계 *universal turing machine*  
어떤 튜링기계, 그러나 “만능”



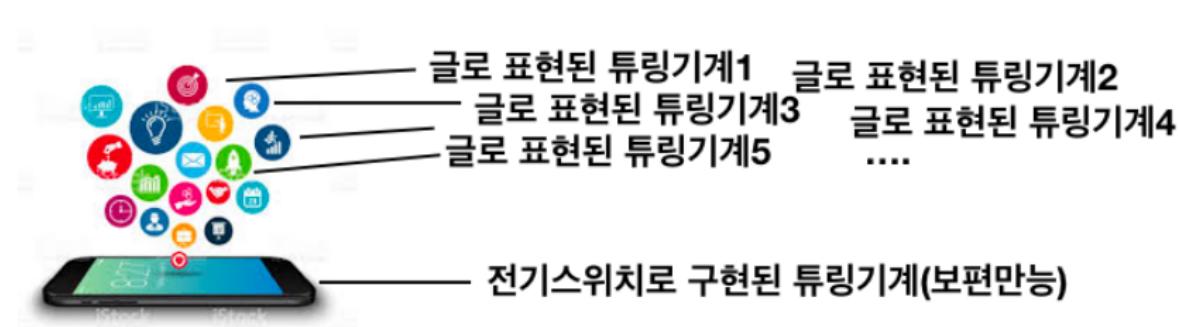
# 컴퓨터 = 만능 튜링기계 universal turing machine

- ▶ 사용법: 임의의 튜링기계(기계적인 일, SW) 테잎에 싣기
- ▶ 자동작동: 테잎에 있는 튜링기계(기계적인 일, SW) 그대로 따라하기(**실행**)



# 컴퓨터 = 만능 튜링기계 universal turing machine

- ▶ 사용법: 임의의 튜링기계(기계적인 일, SW) 테잎에 싣기
- ▶ 자동작동: 테잎에 있는 튜링기계(기계적인 일, SW) 그대로 따라하기(**실행**)



(사진출처: Google)

# 정리 I: 컴퓨터와 SW의 정의

- ▶ 컴퓨터 = 기계적인 일이면 뭐든 실행해주는 도구
  - ▶ SW = 기계적인 일
- ▶ 기계적인 일 = “튜링기계”로 만들어 돌릴 수 있는 일
  - ▶ 이 정의는 아직 깨지지않음
  - ▶ 애매함 1도 없음
  - ▶ HW와 SW가 독립적으로 전속력 발전할 수 있는 토대
- ▶ AI: 어떤 SW(기계적인 일, 튜링기계)

## II. SW의 세계 & AI의 위치

# SW(기계적인 일)들의 세계

- ▶ 기계적인 일은 무수히 많다 ( $|\mathbb{N}|$ )

# SW(기계적인 일)들의 세계

- ▶ 기계적인 일은 무수히 많다 ( $|\mathbb{N}|$ )
  - ▶ 기계적으로 정답내기 불가능한 일도 무수히 많다 ( $|2^{\mathbb{N}}| - |\mathbb{N}|$ )

# SW(기계적인 일)들의 세계

- ▶ 기계적인 일은 무수히 많다 ( $|\mathbb{N}|$ )
  - ▶ 기계적으로 정답내기 불가능한 일도 무수히 많다 ( $|2^{\mathbb{N}}| - |\mathbb{N}|$ )
- ▶ 기계적이지만 정답내기가 하세월인 일도 많다
  - ▶ 예) 인수분해, 최적 여행코스, 배우 캐스팅
  - ▶ 기계적으로 적당한답을 내는 기술 = 어림잡기, 통밥 등등

# SW(기계적인 일)들의 세계

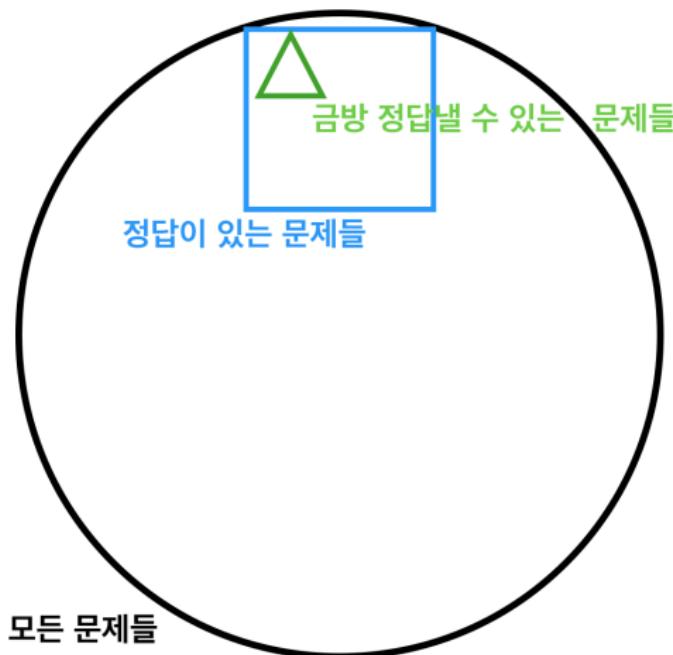
- ▶ 기계적인 일은 무수히 많다 ( $|\mathbb{N}|$ )
  - ▶ 기계적으로 정답내기 불가능한 일도 무수히 많다 ( $|2^{\mathbb{N}}| - |\mathbb{N}|$ )
- ▶ 기계적이지만 정답내기가 하세월인 일도 많다
  - ▶ 예) 인수분해, 최적 여행코스, 배우 캐스팅
  - ▶ 기계적으로 적당한답을 내는 기술 = 어림잡기, 통밥 등등
- ▶ 정답이 아예 없는 일도 무수히 많다.
  - ▶ 예) 미더운 정도, 좋은 번역/요약/비평/광고문구/아이디어 등등
  - ▶ 기계적으로 얼추거의 맞는 답을 내는 기술 ↳ 기계학습(AI)

# 아예 기계적인 방법을 모르거나 썬 방법을 못찾은 경우

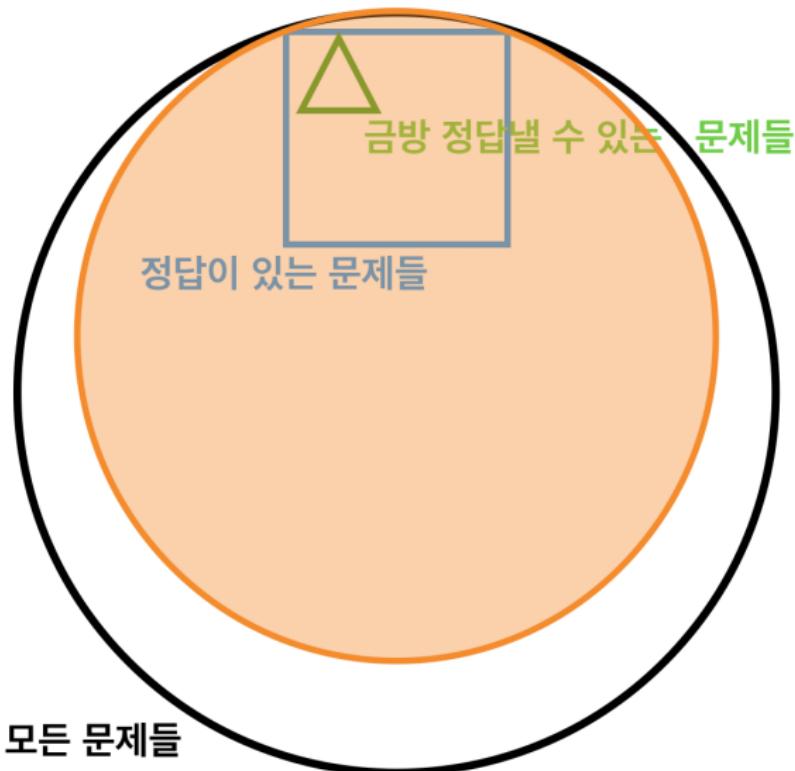
그래도 컴퓨터로 돌리고 싶다면

- ▶ 모든 입력에 정답이 아니어도 좋다면
  - ▶ “모든 입력” 대신 “흔한 입력”
  - ▶ “정답” 대신 “적당한 답” + “얼추거의 맞는 답”
  - ▶ 예) 길 안내, 진로 선택, 배우 캐스팅, 번역, 요약, 그림그리기 등등
- ▶ 사용하는 기법
  - ▶ 어림잡기 approximation, 통밥 heuristic, 기계학습 machine learning

## 정리 II: SW 세계, 컴퓨터(기계적인 방식)로 푸는 문제들



## 정리 II: SW 세계, 컴퓨터(기계적인 방식)로 푸는 문제들



컴퓨터로 푸는 문제들: 정답 + 적당한답

### III. SW 만들기 두 가지 방법

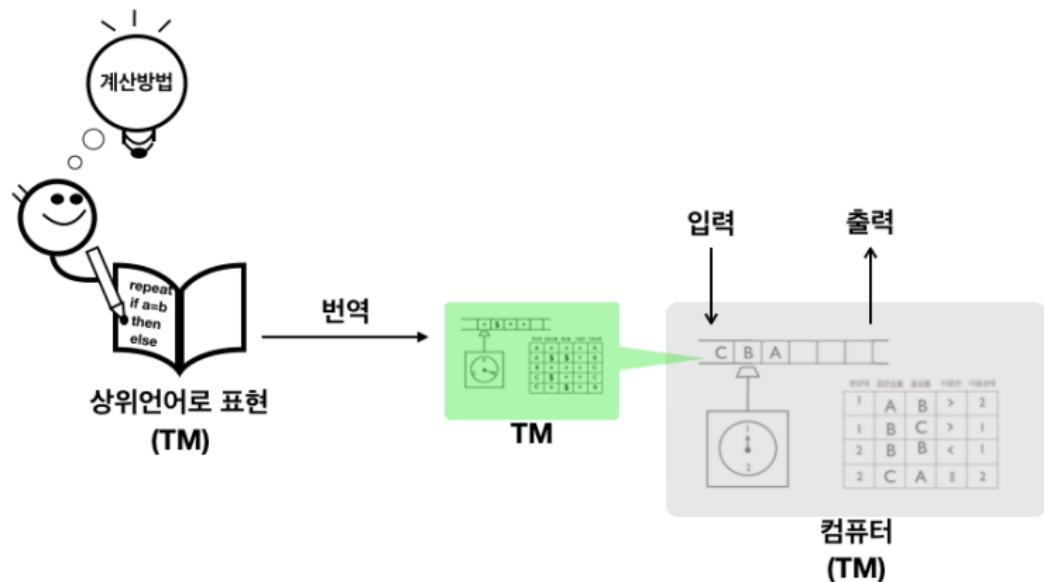
# SW 만들기 I: 언어와 논리로

언어와 논리로 우리가 직접 짠다 (AI도움도 받고)

- ▶ 계산방법을 꼼꼼히 상위의 언어로
- ▶ 그 언어와 논리는 나이가 상위 레벨로 올라감
- ▶ 컴퓨터가 많은 빈틈을 메꿔줌
  - ▶ 번역, 최적화, 지루한 부분, 초벌 스케치, 등등

프로그래밍/코딩 programming, 알고리즘 algorithm

# SW 만들기 I: 언어와 논리로



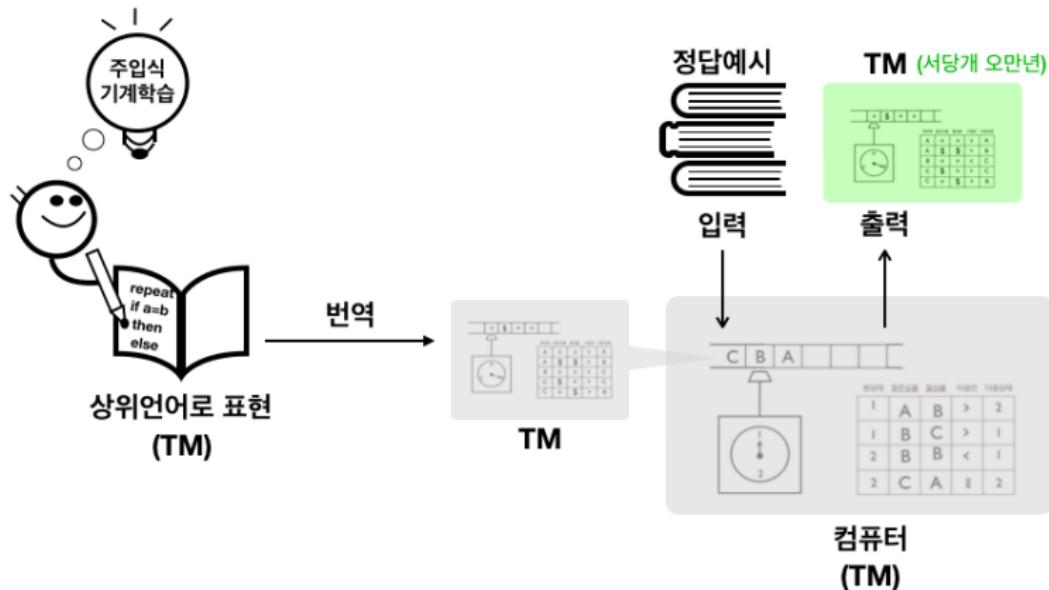
## SW 만들기 II: 언어와 논리로 표현못할 경우

컴퓨터가 짜도록 한다

- ▶ 주입식 기계학습으로
  - ▶ “보고 따라하기” “서당개 5만년” 훈련
- ▶ 결과 SW(TM)(예, 신경망 neural net)를 컴퓨터에 싣는다
  - ▶ 매우 하위언어로 짜여진 기계적인 일
- ▶ 기계학습으로 만들어 지는 SW(AI)의 특성
  - ▶ 얼추 거의 맞는 방법임
  - ▶ 늘 정답을 내는건 아님

기계학습 machine learning, 알고리즘 algorithm

## SW 만들기 II: 언어와 논리로 표현못할 경우



## IV. 인공지능시대 컴퓨팅 공부법

# 팀웍: 우리지능 x SW지능

우리지능과 SW지능이 한 팀

- ▶ 우리지능: 4백만년(진화) + 1백년(교육) 동안 학습한 지능
  - ▶ 논리/언어/시각지능, 청각/촉각지능
  - ▶ 공감/부끄럼/양보/상식지능  
(인/의/예/지, 측은/수오/사양/시비지심)
- ▶ SW지능: 컴퓨터 기술로 만드는 지능
  - ▶ 알고리즘(algorithm): 정답을 끄는 SW
  - ▶ 에코리즘(ecorithm): 주입식 기계학습한 얼추거의 맞는 SW(AI)

# 팀웍 양상: 컴퓨팅 공부법

- ▶ 미래 지능적인 일 ≠ 현재 지능적인 일
  - ▶ ≠ 현재 전공하기, 현재 코딩하기, 현재 수학하기, 현재 글쓰기, 등등…
- ▶ 미래 지능적인 일
  - ▶ 없던 개념 만들기 ∈ 지능(공부)의 최고 수준
    - ▶ 과거 예시) “인권”(루소), “질량”(뉴턴), “인내천”(최제우), “진화”(다윈), “기계계산”(튜링)
  - ▶ 주변문제 초대있게 풀어가기 (문제선택 × 인의예지)
    - ▶ 조합가능한 문제는 무진장; 그 중 의미있는 문제를 선택할 수 있어야

# 팀웍 양상: 컴퓨팅 공부법

- ▶ 그리고 최고 팀웍(우리지능 × SW지능)을 이루고 싶다면
  - ▶ 상상하는 힘: SF소설읽기
  - ▶ 묻는 힘: 철학공부하기
  - ▶ 핵심을 따지는 힘: 수학공부하기
  - ▶ 실현하는 힘: 컴퓨팅공부하기
  - ▶ 주변문제 준대있게 푸는 힘: 쉬운말로공부하기(모국어의 심연)

