PHILOSOPHY 101

FALL 2016

PHIL101.COLINMCLEAR.NET

TABLE OF CONTENTS

CAN MACHINES THINK?

TWO QUESTIONS

1. Can a physical system capable of performing certain functions think?
2. Can a sufficiently sophisticated computer program think?
    - Is the mind related to the brain like software is to hardware?

Could a sufficiently advanced computer qualify as a thinking being?

> *A. Yes*
> *B. No*

STRONG & WEAK AI

Strong AI:

thinking is constituted by the manipulation of formal symbols, such as occurs in a computer program

Weak AI:

thinking may be modeled by formal symbol systems, such as computer programs

THE IMITATION GAME

- Can you guess, using a series of questions, which of two conversation partners is a machine and which a human?
- Questions may be of all kinds:
  - what's your name
  - what's your favorite color?
  - what does the smell of freshly cut grass remind you of?

THE TURING TEST

> *I believe that in about fifty years' time it will be possible to programme computers…to make them play the imitation game so well that an average interrogator will not have more than 70 percent chance of making the right identification after five minutes of questioning…I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted. (Alan Turing)*

1. For some arbitrary time period, there may be no discernible difference between the linguistic behavior of a person and that of a machine
2. If there is no discernible difference in linguistic behavior between man and machine, then there is no reason to think that there is any underlying difference in the causes of that behavior
3. $\therefore$ If we are willing to say that it is intelligent thought that is the cause of the linguistic behavior in the person we should be willing to say the same thing about the machine

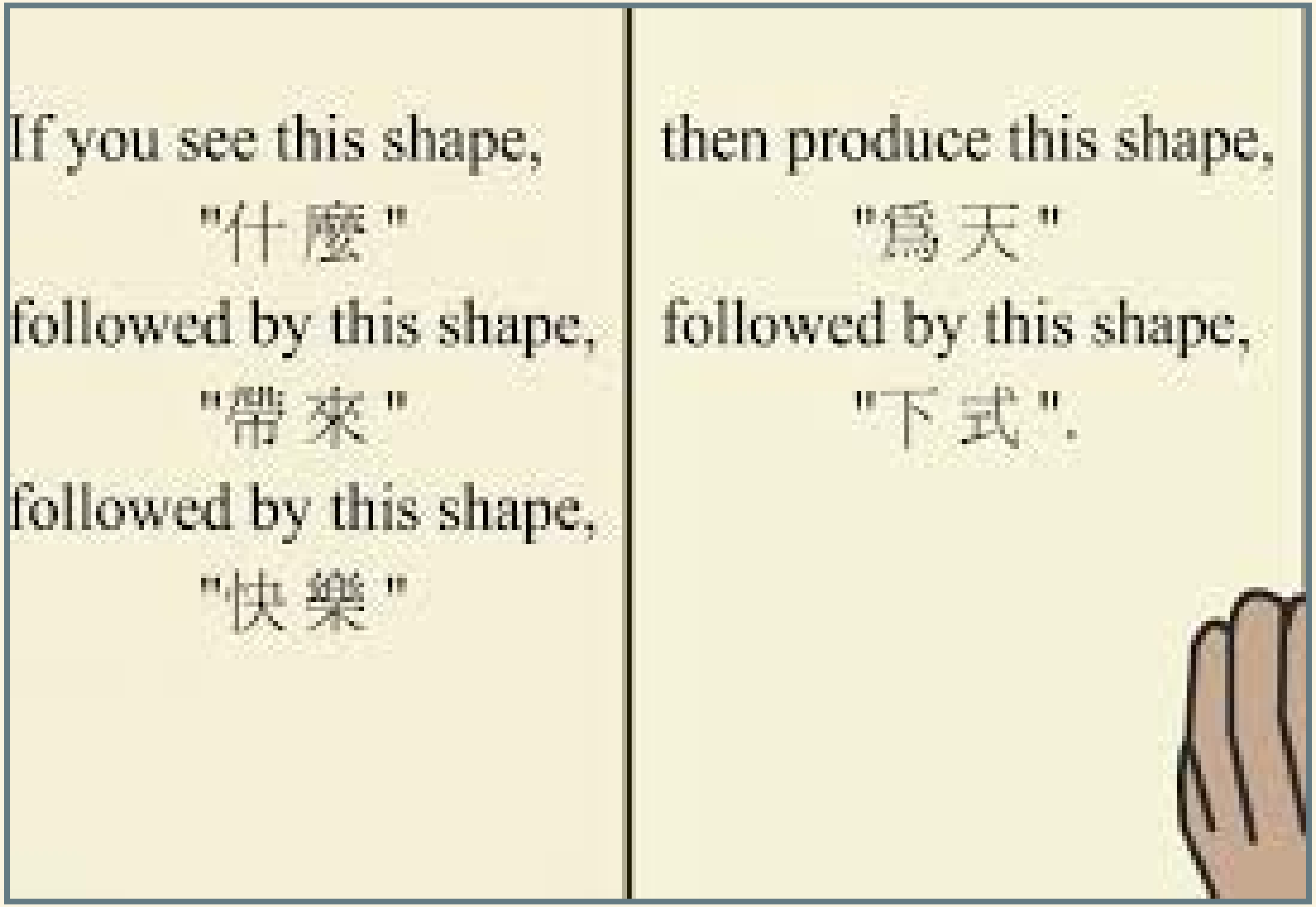
STRONG AI & THE TURING TEST

- Any computer that can pass the Turing Test for arbitrarily long periods of time will, according to strong AI, qualify as a thinking machine

THE CHINESE ROOM ARGUMENT

> *suppose I am placed in a room containlng baskets full of Chinese symbols. Suppose also that I am given a rule book in English for matching Chinese symbols with other Chinese symbols. The rules identify the symbols entirely by their shapes and do not require that I understand any of them. The rules might say such things as, "Take a...sign from basket number one and put it next to a...sign from basket number two." Imagine that people outside the room who understand Chinese hand in small bunches of symbols and that in response I manipulate the symbols according to the rule book and hand back more small bunches of symbols.*

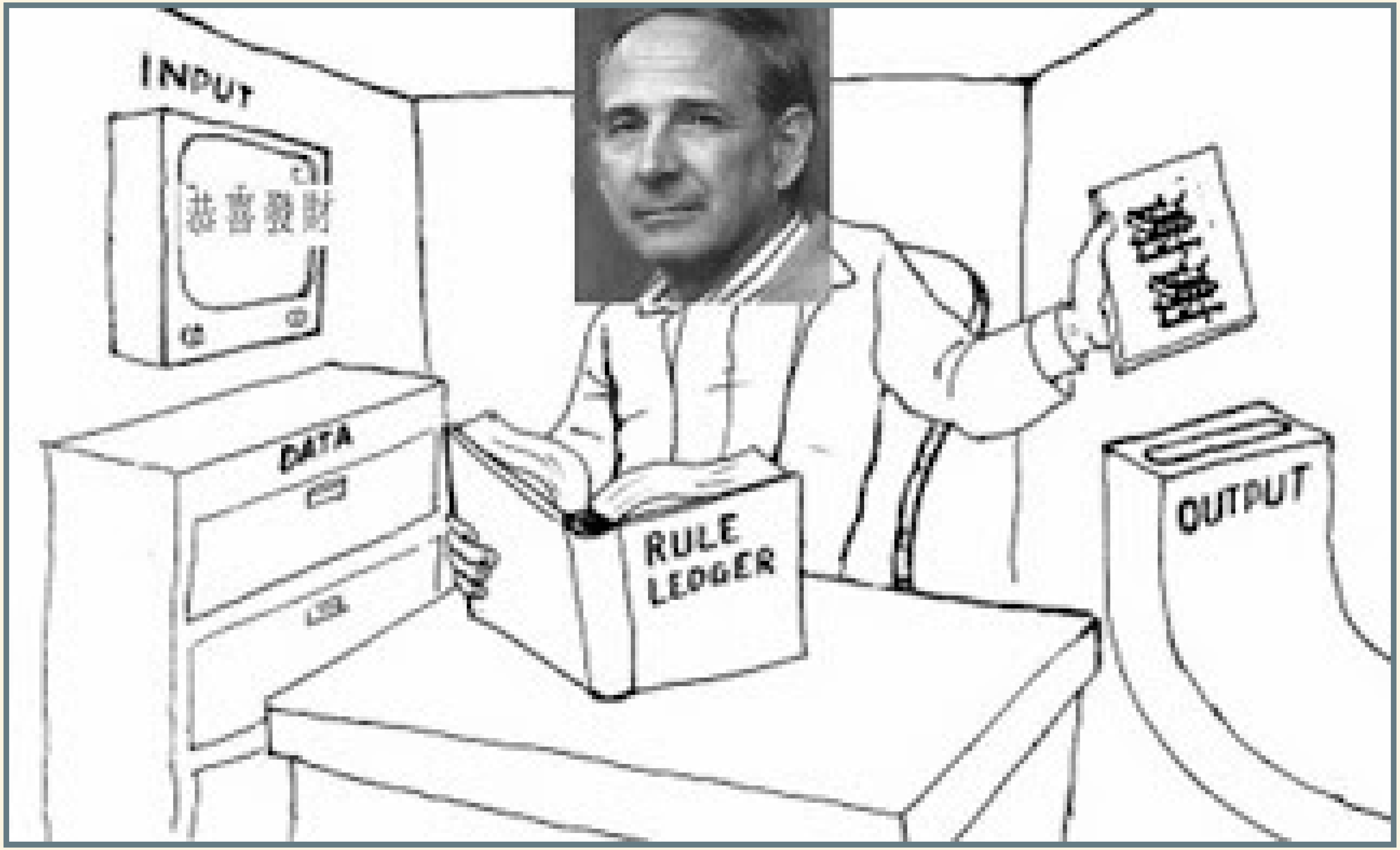
> *Now, the rule book is the "computer program." The people who wrote it are "programmers," and I am the "computer." The baskets full of symbols are the "data base," the small bunches that are handed in to me are "questions" and the bunches I then hand out are "answers."*



If you see this shape,
"什麼"
followed by this shape,
"帶來"
followed by this shape,
"快樂"

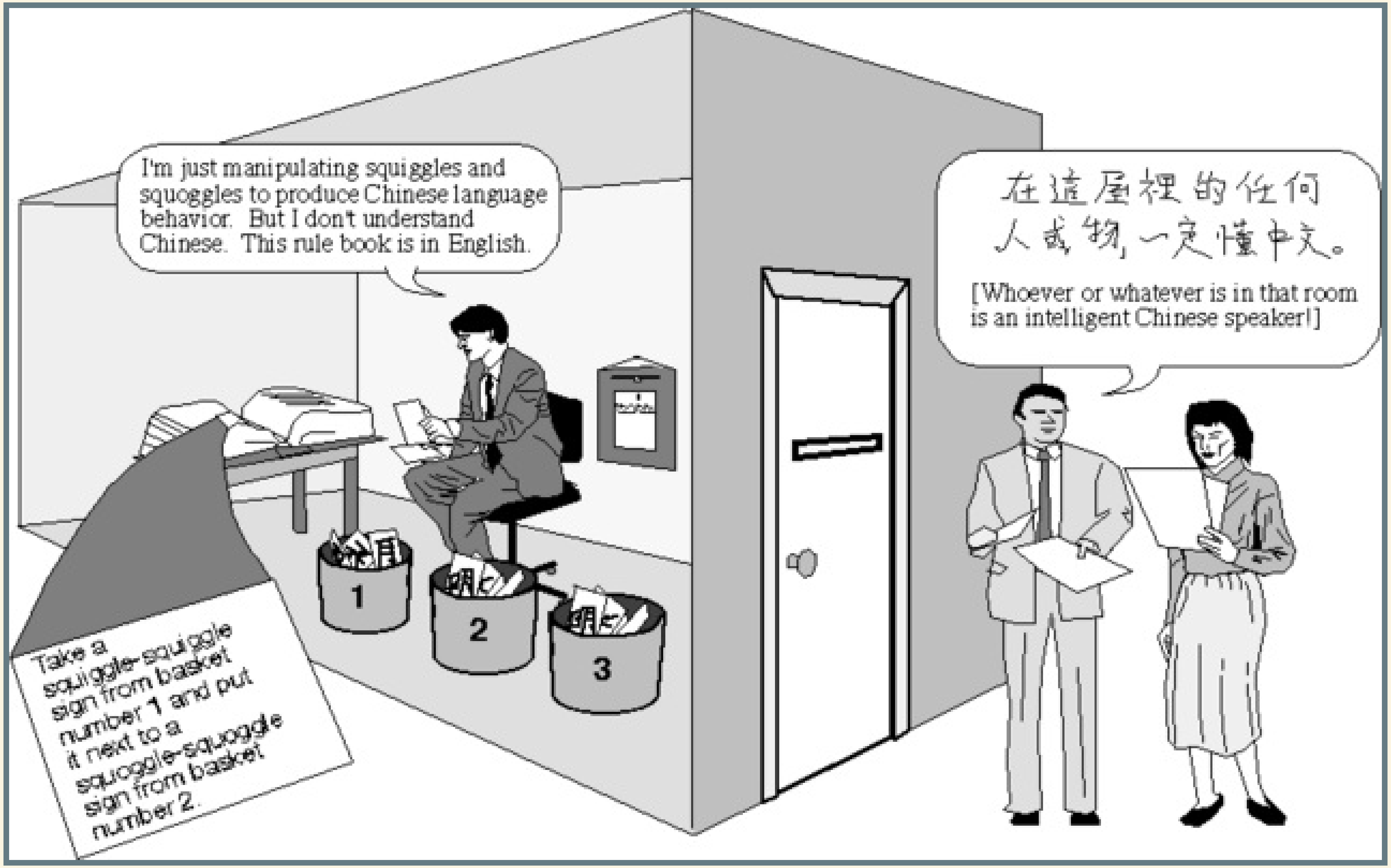
then produce this shape,
"爲天"
followed by this shape,
"下式".

The rulebook



The Chinese room

> *Now suppose that the rule book is written in such a way that my "answers" to the "questions" are indistinguishable from those of a native Chinese speaker. For example, the people outside might hand me some symbols that unknown to me mean, "What's your favorite color?" and I might after going through the rules give back symbols that, also unknown to me, mean, "My favorite is blue, but I also like green a lot." I satisfy the Turing test for understanding Chinese. All the same, I am totally ignorant of Chinese. And there is no way I could come to understand Chinese in the system as described, since there is no way that I can learn the meanings of any of the symbols. Like a computer, I manipulate symbols, but I attach no meaning to the symbols. (Searle, 26)*


I'm just manipulating squiggles and squoggles to produce Chinese language behavior. But I don't understand Chinese. This rule book is in English.
在這屋裡的任何人或物，一定懂中文。
[Whoever or whatever is in that room is an intelligent Chinese speaker!]
Take a squiggle-squiggle sign from basket number 1 and put it next to a squoggle-squoggle sign from basket number 2.
1
2
3

The Chinese room

SYNTAX & SEMANTICS

Syntax:

the formal or structural features of a symbol system which determine which expressions are legitimate members of the system and which are not

- The syntax of English (its grammar) requires that all complete sentences have a noun phrase and a verb phrase
  - 'John goes to school' vs. 'school John to goes'

SYNTAX & SEMANTICS

Semantics:

The system of meanings assigned to a symbol system, given by determining the referents of the symbols and the truth conditions of symbol strings

- 'Schnee' refers to snow
- 'weiß' refers to the property of being white
- 'Schnee ist weiß' is true just in case snow is white

THE ARGUMENT CLARIFIED

1. Programs are purely formal (syntactic)
2. Human minds have mental contents (semantics)
3. Syntax by itself is neither constitutive of, nor sufficient for, semantic content
4. $\therefore$ Programs by themselves are not constitutive of nor sufficient for minds

WHAT DOES THE ARGUMENT INTEND TO PROVE?

- You can't get semantic content from syntax alone
- A system must have more than purely syntactic properties in order to possess intentional states

OBJECTIONS TO THE CHINESE ROOM ARGUMENT

TWO OBJECTIONS

1. The 'Systems' Objection
2. The 'Implementation' Objection

THE 'SYSTEMS' OBJECTION

- Perhaps the person *in* the Chinese room does not understand Chinese but the *Chinese Room itself* understands Chinese
- Since the Chinese room is the proper analogue to the computer program, and not the person *in* the Chinese room, Searle's example proves nothing

SEARLE'S REPLY

> *My response to the systems theory is quite simple: let the individual internalize all of these elements of the system. He memorizes the rules in the ledger and the data banks of Chinese symbols, and he does all the calculations in his head. The individual then incorporates the entire system. There isn't anything at all to the system that he does not encompass. We can even get rid of the room and suppose he works outdoors. All the same, he understands nothing of the Chinese, and a fortiori neither does the system, because there isn't anything in the system that isn't in him. If he doesn't understand, then there is no way that the system could understand because the system is just a part of him.*

PRYOR'S REBUTTAL (I) SEARLE'S ARGUMENT IS INVALID

> *Searle: "[The man in the room] understands nothing of the Chinese, and [therefore] neither does the system, because there isn't anything in the system that isn't in him"*

- This is a bad inference—compare:

> *Searle doesn't weigh 3 pounds, and therefore neither does his heart, because there is nothing in his heart that isn't in him*

- the form of inference Searle uses here doesn't generalize to other inferences with the same kind of form
    - leaves open the possibility that the particular argument Searle makes here is true

PRYOR'S REBUTTAL (II) INTERNALIZATION IS IRRELEVANT

> *Searle: "If he doesn't understand, then there is no way that the system could understand because the system is just a part of him."*

- Consider a software emulator
  - allows one operating system to run 'on top of' another using the same hardware
    - Mac computers can emulate the Windows OS

- Assume a Mac runnning its OS *plus* an emulation of Windows OS

    1. The Windows OS is integrated or incorporated into the Mac OS
    2. Nevertheless, the states of the 'incorporated' Windows OS are in many ways independent of the Mac OS and its states
        - Windows may crash and become unresponsive, while the Mac software (including the emulator) keeps running
        - Windows might be treating Internet Explorer as the frontmost, active program; but–if you don't have the emulator software active in your Mac–the Mac software could be treating Safari as its frontmost, active program

> *when Jack memorizes all the instructions in the Chinese book, he becomes like the Mac software, and the Chinese room software becomes like the emulated Windows software. Jack fully incorporates the Chinese room software. That does not mean that Jack shares all the states of the Chinese room software, nor that it shares all of his states. If the Chinese room software crashes, Jack may keep going fine. If the Chinese room software is in a state of believing that China was at its cultural peak during the Han dynasty, that does not mean that Jack is also in that state. And so on. In particular, for the Chinese room software to understand some Chinese symbol, it is not required that Jack also understand that symbol.*

- Problem 2: 'Internalizing' the Chinese room program is irrelevant
  - two programs running on the same hardware need not share all of the same (or any of the same) states

SUMMARY OF PRYOR'S REBUTTALS:

1. Searle's argument is invalid
    - the form of inference Searle uses here doesn't generalize to other inferences with the same kind of form in a way that preserves truth
2. 'Internalization' is irrelevant
    - two programs running on the same hardware need not share all of the same (or any of the same) states

THE IMPLEMENTATION OBJECTION

PROGRAMS VS. IMPLEMENTATIONS

> 1. *Programs are purely formal (syntactic)*
> 2. *Human minds have mental contents (semantics)*
> 3. *Syntax by itself is neither constitutive of, nor sufficient for, semantic content*
> 4. *∴ Programs by themselves are not constitutive of nor sufficient for minds*

- We need to distinguish between a *program* and an *implementation of the program*

> *Programs are abstract computational objects and are purely syntactic. Certainly, no mere program is a candidate for possession of a mind. Implementations of programs, on the other hand, are concrete systems with causal dynamics, and are not purely syntactic. An implementation has causal heft in the real world, and it is in virtue of this causal heft that consciousness and intentionality arise. It is the program that is syntactic; it is the implementation that has semantic content. (Chalmers 1996, 327)*

CHALMERS'S PARODY:

1. Recipes are syntactic.
2. Syntax is not sufficient for crumbliness.
3. Cakes are crumbly.
4. ∴ Implementing a recipe is insufficient for a cake.

> *A recipe implicitly specifies a class of physical systems that qualify as implementations of the recipe, and it is these systems that have such features as crumbliness. Similarly, a program implicitly specifies a class of physical systems that qualify as implementations of the program, and it is these systems that give rise to such features as minds. (Chalmers, 327)*

Created by Colin McLear.