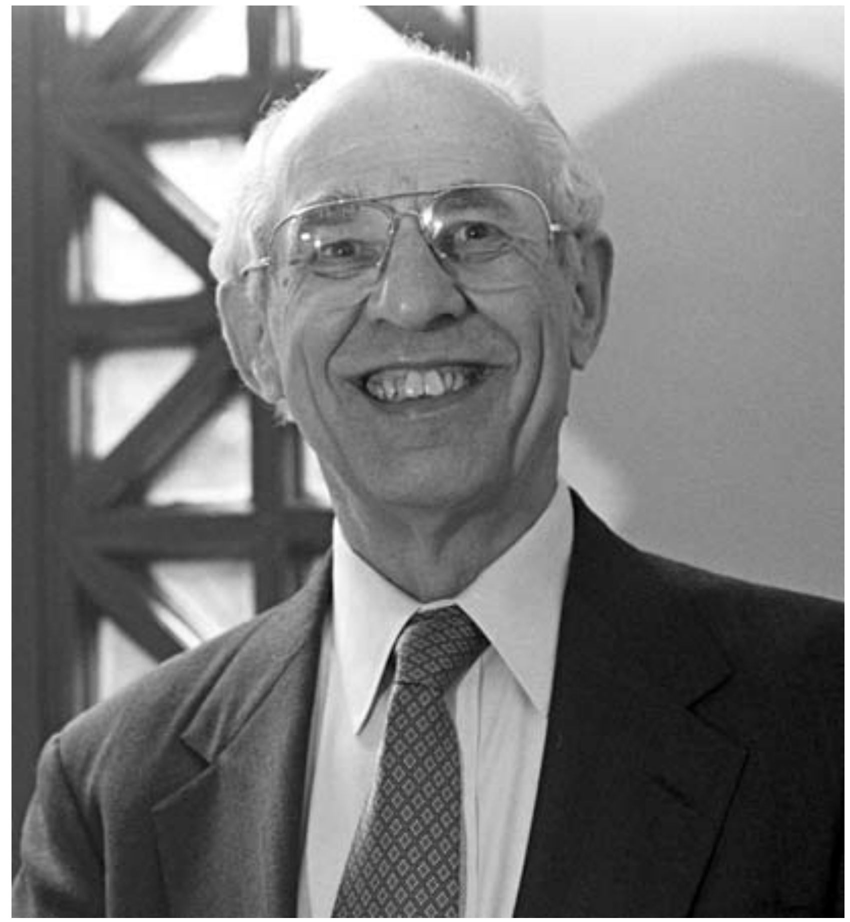
Alan Turing
1912-1954



Hilary Putnam
1926-2016

Jerry Fodor
1935-2017

Functionalism I

References

- A. Turing, “On computable numbers, with an application to the Decision Problem” (1936– 7).
- Putnam (1967) « The mental life of some machines » (first delivered as a talk in 1962, reprinted in his collected work vol 2, p. 408sq).
- Putnam (1967), « The nature of mental sates ».
- Fodor (1965), « Explanations in psychology ».
- Fodor (1968), « The appeal to tacit knowledge in psychological explanation ».
- David Marr (1982), *Vision*, chap. 1.

Plan

- Mental states as causal roles
- Two varieties of functionalism
    - Turing-machines functionalism
    - Psycho-functionalism
- A top-down functional analysis of psychological capacity: David Marr on human vision

I The core idea

Mental states as causal roles

- Functionalism: what makes something a belief, a desire, a pain… isn't its internal physical constitution, but its **function**, or the **causal role** it plays in the cognitive system of which it is a part.
- The pain role: pain in an internal state, typically caused by bodily damages, which typically causes pain behaviors (crying, frowning, whimpering …), as well as other mental states: the belief that one is in pain, the desire for pain to stop, emotions like anger, fear, etc.
- A function F is realized by a state P in a system X iff P plays the F-role in X.
- Functions might be multi-realizable:
    - A function F is multi-realizable iff its causal role can be realized by different physical structures in different systems.

- Multi-realizability is an important argument for functionalism according to Putnam, Fodor, and Block.
- Block 1980: functionalism provides a more inclusive, a less “(species-) chauvinistic” theory of the mental than the identity theory.
    - For instance: even the silicon-based states of hypothetical martians could realize the pain role, if their psychology was sufficiently similar to ours.
- Widely different physical structures may realize the same functional properties. Consider the following exemple, due to Ned Block:

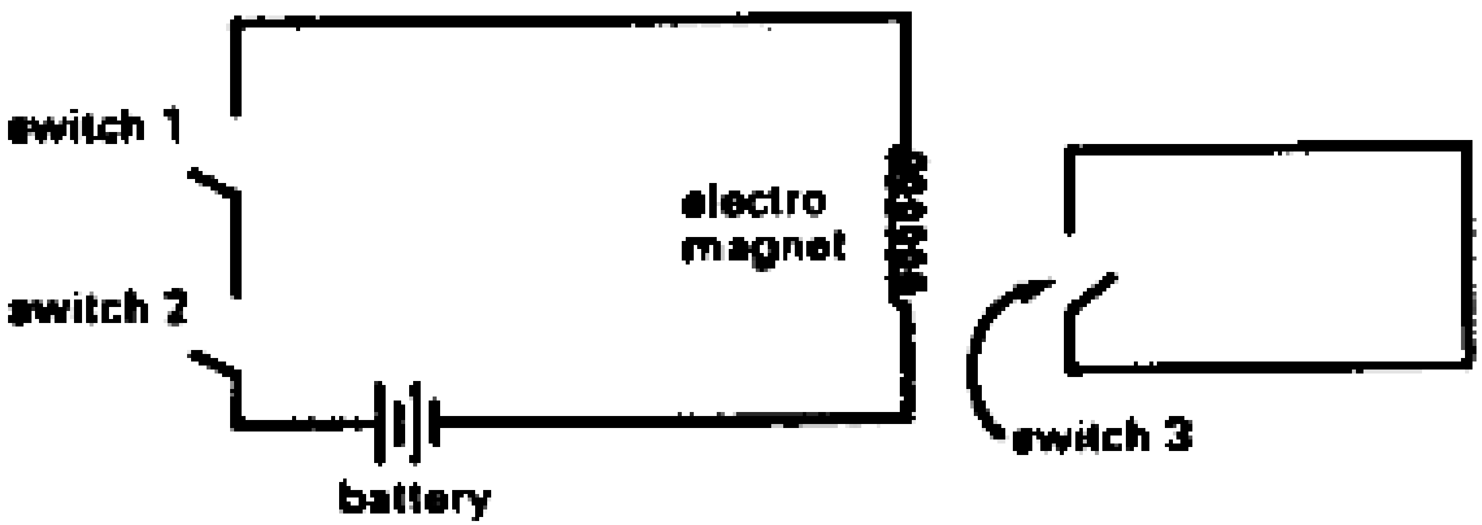


Figure 4: Electrical AND gate. Open = 0, closed = 1

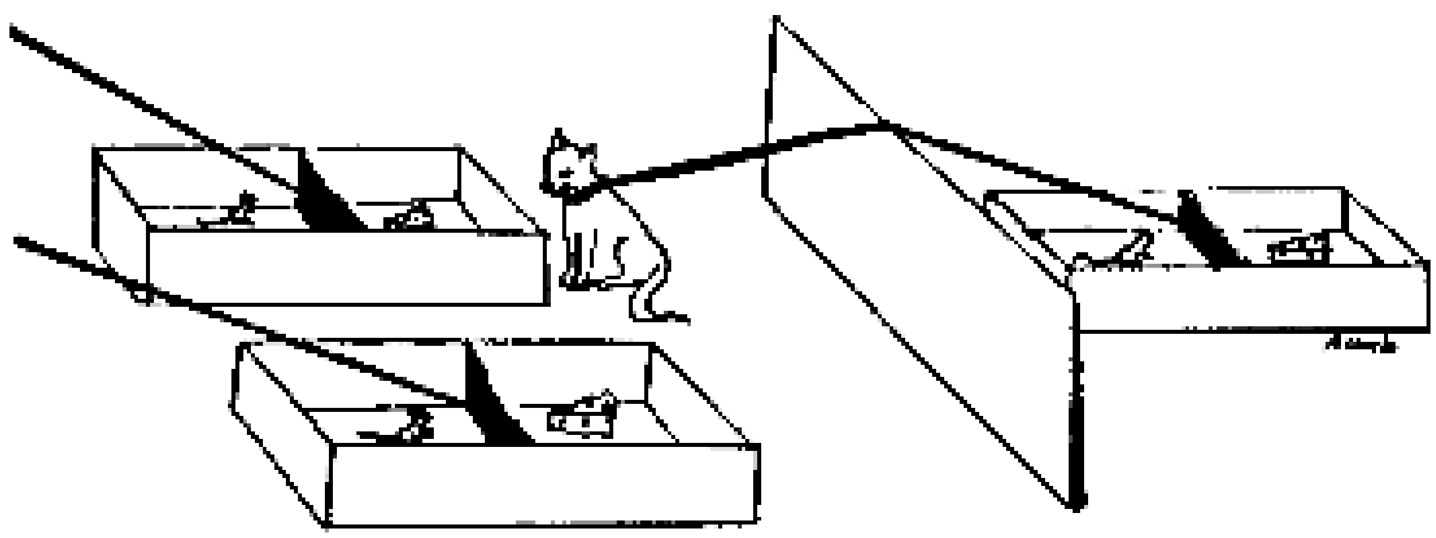


Figure 5: Cat and mouse AND gate. Hungry mouse = 0, mouse fed = 1

The metaphysics of functionalism

- **Role functionalism** (Putnam, Fodor): mental states are higher-order properties, like the property of having some state that plays causal role R. For instance, « being in pain » = the second-order property of having a state that plays the causal role of pain.

- **Realizer functionalism** (Lewis, Armstrong, Kim): mental states are first-order states that are characterized, or described, as playing causal role. For instance, pain for humans is C fibers-firing, because for humans C-fibers firing is the state that plays the causal role of pain.

II Two kinds of functionalism

Turing machines

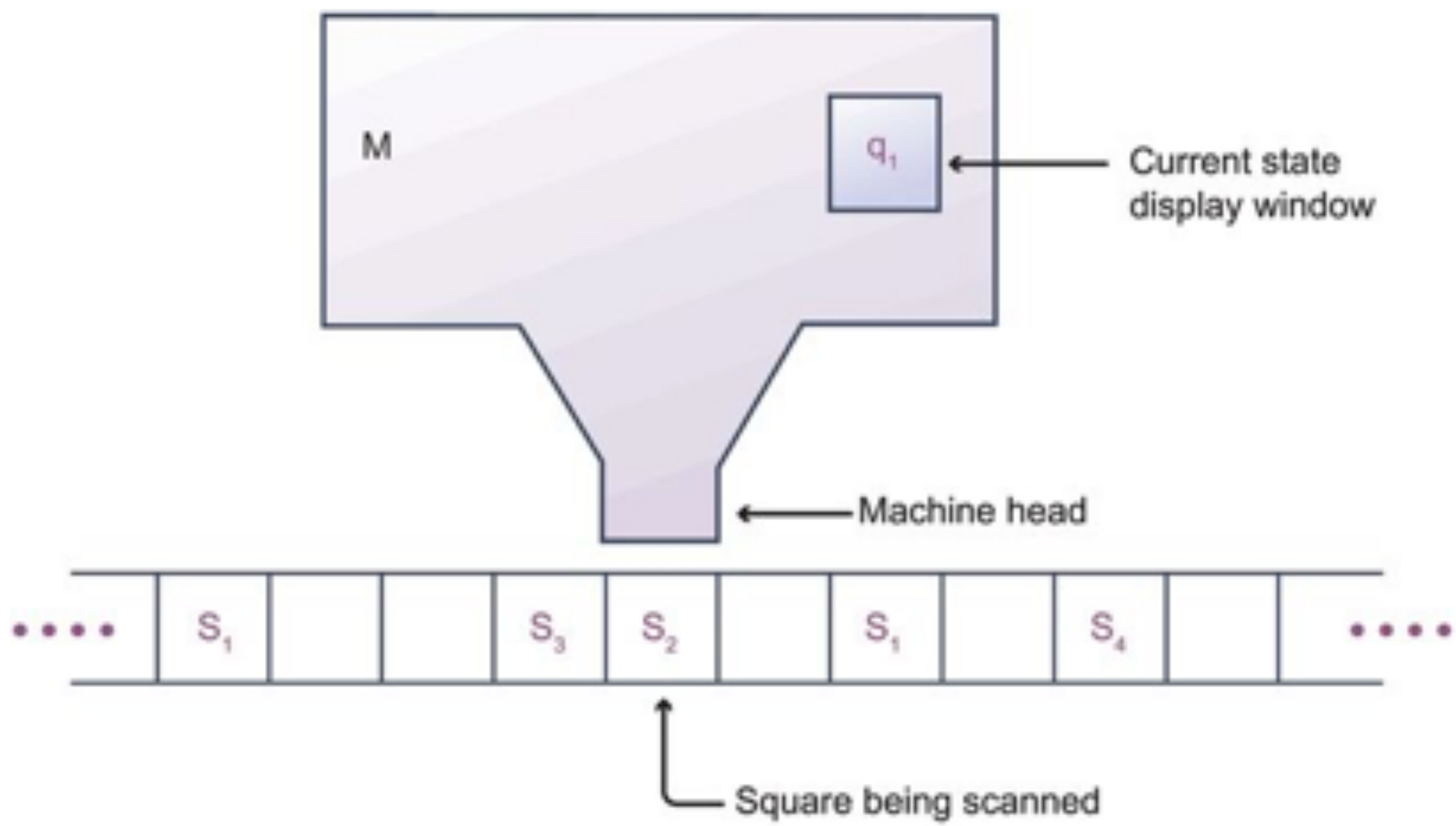
- Alan Turing : “On computable numbers, with an application to the Decision Problem” (1936– 7).

- Turing’s aim was to find a purely mechanical solution to the **halting problem**: the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running or continue to run forever.

- In 1936 Turing proves that a general algorithm to solve the halting problem for all possible program-input pairs cannot exist.

- To this effect, he has to give a precise mathematical definition of an algorithm. That’s the motivation for the introduction of a Turing machine.

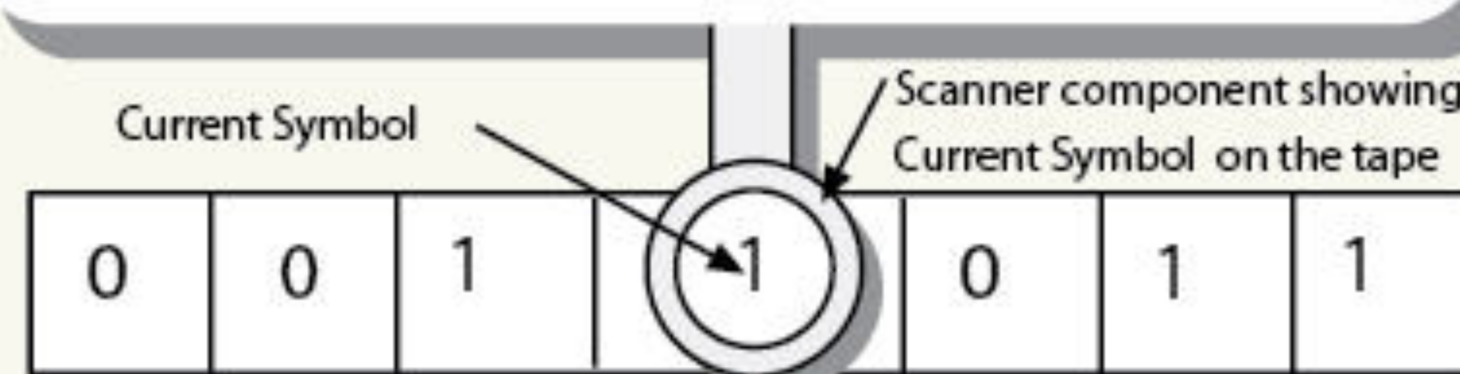
- A Turing machine is like a computer with an infinitely large hard disk: it has an infinite tape, comparable to an infinite working memory.
- Each of the cells of the Turing tape can be either blank or contain a single symbol.
- The Turing machine contains a machine head. The tape runs through the machine head, with a single cell under the head at a given moment.
- The head can read the symbols the cells contain, and carry out the following operations:
    - delete the symbol in the cell
    - write a new symbol in the cell
    - move the tape one cell to the left
    - move the tape one cell to the right
- The machine also has a finite set of instructions, its program.
    - Each instruction has this form: f the machine is in state Si, and receives input Ij, it will go into state Sk and produce output Ol (for a finite number of states, inputs and outputs).
- At each instant, it can be in one of a finite number of internal states.



M
q_1
Current state display window
Machine head
S_1
S_3
S_2
S_1
S_4
Square being scanned

IF the Current State is ...	AND IF the Current Symbol reads ...	THEN do the following to the symbol on the tape and/or halt ...	AND move the tape as follows ...	AND THEN change the Current State to ...
1	0	Do nothing	Right	1
1	1	Do nothing	Right	2
2	0	Erase & write 1	Right	3
2	1	Do nothing	Right	2
3	0	Do nothing	Left	4
3	1	Do nothing	Right	3
4	0	Erase & write 0	No move	4
4	1	Halt	No move	4

Current State 1



Universal Turing Machine

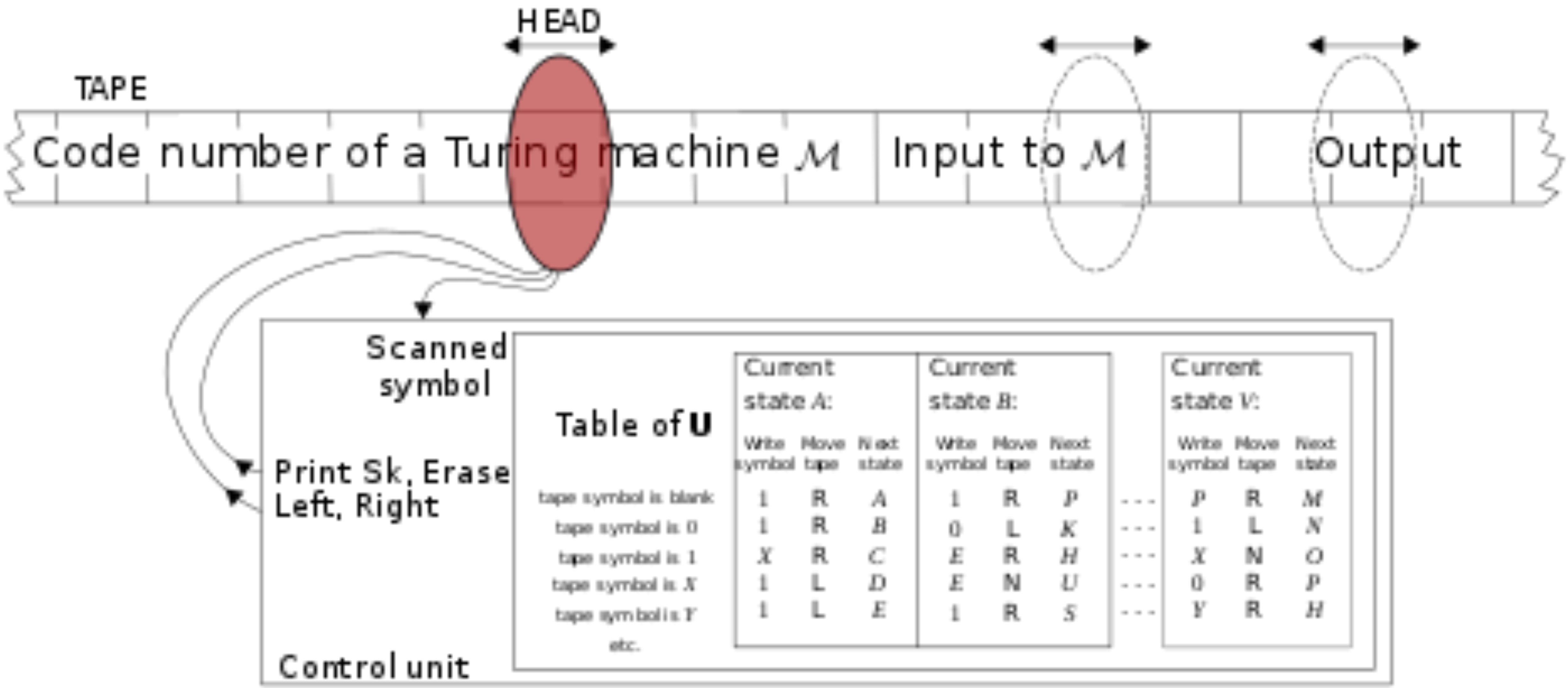
- A Universal Turing Machine is a Turing machine that can perform all the calculations of any other Turing machine.

- To emulate a given machine, the Universal Turing Machine is programmed with a special tape that fully describes the emulated machine's table.

- In 1936, Turing describes how such a machine can always be constructed from the description of any particular Turing Machine:

    - « It is possible to invent a single machine which can be used to compute any computable sequence. If this machine U is supplied with a tape on the beginning of which is written the S.D ["standard description" of an action table] of some computing machine M, then U will compute the same sequence as M. »

- This idea influenced John von Neumann's conception of the first American discrete-symbol (as opposed to analog) computer.


HEAD
TAPE
Code number of a Turing machine M
Input to M
Output
Scanned symbol
Print Sk, Erase
Left, Right
Table of U
Current state A:
Current state B:
Current state V:
Write symbol
Move tape
Next state
tape symbol is blank
tape symbol is 0
tape symbol is 1
tape symbol is X
tape symbol is Y
etc.
1 R A
1 R B
X R C
1 L D
1 L E
1 R P
0 L K
E R H
E N U
1 R S
P R M
1 L N
X N O
0 R P
Y R H
Control unit

Machine functionalism

- First defended in Putnam (1967) « The mental life of some machines » (first delivered as a talk in 1962, reprinted in his collected work vol 2, p. 408sq).

  - « As applied to Turing Machines, the functional organization is given by the machine table. A description of the functional organization of a human being might well be something quite different and more complicated. But the important thing is that descriptions of the func- tional organization of a system are logically different in kind either from descriptions of its physical–chemical composition or from descriptions of its actual and potential behavior. »

Machine functionalism vs behaviorism

- Behaviorism: Mental states are essentially dispositions to behave in certain ways, given sensory inputs; the mind is similar to a **black box** between sensory stimulations and behaviors.

- Machine Functionalism: Mental states are essentially dispositions to behave in certain ways, given sensory inputs and other mental states, corresponding to the machine's inner states.

  - Putnam 1960, « Minds and machines » : there is an 'analogy between man and Turing machine' (Putnam, 1960, p. 159). According to Putnam, mental states are indeed analogous to the internal states of a TM as described by its abstract machine table, whereas brain states are analogous to the physical states of a **hardware realization** of a TM.

Machine functionalism and multiple realizability

- Putnam : « The functional organization (problem solving, thinking) of the human being or machine can be described in terms of the sequences of mental or logical states respectively (and the accompanying verbalizations), without reference to the nature of the 'physical realization' of these states. » (Ibid., p. 149)

  - The multiple-realizations argument is echoed in the metaphor of the mind as a « software » implemented in the brain.

Fodor’s psycho-functionalism

- According to Fodor’s (1965) « Explanations in psychology », psychological theories are developed in two logically distinct phases: **functional analyses**, and **mechanical analyses**.

- **Functional analyses**: internal states of organisms are characterized « only in the way they function in the production of behaviors ». At this stage, no information is given about the underlying mechanisms. Explanations at this stage are « compatible with indefinitely many hypotheses about the physiology ».

- **Mechanical analyses**: directed towards « determining the nature of the mechanisms whose functional characteristics one specifies ».

- This view should be « **distinguished from all varieties of reductionism** ».

« to put it succinctly a complete psychological explanation requires more than an account of what the neural circuitry is; it requires also an account of what such circuitry does »

Fodor’s synthesis

- TMs are individuated by a finite number of internal state types and the program specifying what transitions occur under what conditions.

- Functional analyses on the other hand are specifications of mechanism types, consisting of different component types and their assigned functions without specifying the precise state types and state transitions that must occur within the analyzed system.

- As soon as 1968 (« The appeal to … ») however, Fodor claims that « A completed psychological theory must provide **systems of instructions** to account for the forms of behavior available to an organism »

- He thus suggests that functional analysis must be complemented by a (computational) specification of programs:

« Or, to vary the image, every box name in a computer flow chart is the name of a problem. If the computer is to simulate behavior, every box name will be the name of a psychological problem. The problem is to specify a sequence of instructions which

(a) will convert the input to the box into the output from the box, and

(b) can be written in a way that mentions only operations that are elementary for the organism whose behavior we are trying to simulate. » (Fodor, 1968)

The « elementary operations » here are operations that could be computed in the brain.

The autonomy of psychology

- Following Fodor, many cognitive scientists have argued since 1968 that **cognitive processes can be studied independently of their physical realization.**

- Just as we can understand a program without knowing anything about the hardware on which it runs, so we can understand cognitive processes without knowing anything about the neural machinery that runs them.

- Cognitive science as **reverse engineering**: the main aim of cognitive science isn't to describe the detail of neural mechanisms, but **to extract the software** from an examination of brain circuitry.

David Marr (1945-1980)

David Marr - as a schoolboy - 1960

David Marr - in La Jolla, CA - 1974

III A case study: David Marr's theory of vision

Three levels

- Marr distinguishes three different levels for analyzing cognitive systems, which roughly correspond to Fodor's three levels (functional analysis, computational analysis, mechanisms), although the terminology differs:

Computational theory	Representation and algorithm	Hardware implementation
What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?	How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?	How can the representation and algorithm be realized physically?

Figure 1–4. The three levels at which any machine carrying out an information-processing task must be understood.

An interdisciplinary approach to vision

- Marr's views about the computational level were influenced by research into brain-damaged patients.
    - He quotes Elisabeth Warrington's work on patients with damages to the left and right parietal cortex.
- Patients with right parietal lesions are able to name objects when they are presented in a familiar way, but not when they are presented under an unconventional perspective.

Figure 2.11 The image on the left is a familiar or conventional view of a bucket. The image on the right is an unfamiliar or unconventional view of a bucket. (From Warrington and Taylor 1973)

- Patients with left parietal lesions, on the other hand, are unable to name objects, but can match a conventional and an unconventional view of the same object, which suggests that they have access to viewpoint independent representations of 3-D objects.

- Marr’s conclusion: « Elizabeth Warrington had put her finger on what was somehow the quintessential fact about human vision – that it tells us about shape and space and spatial arrangement. **Here lay a way to formulate its purpose – building a description of the shapes and positions of things from images**.»

Computational theory

- At this level, cognitive scientists analyze in very general terms the particular type of task that the system performs.

- They identify vision's input and it's output:

  - Input: the pair of 2-D images projected from the environment to the viewpoint of the observer's eyes.

    - The image that strikes the retina is continuous, but it's registration by retinal receptors is discrete;

    - Receptors are much more densely packed in the fovea that in the periphery, but a computational theory of vision is going to idealize the retinal representation as a homogeneous 2-D array of receptors.

  - Output: a representation of the three-dimensional shapes and spatial arrangement of objects in a form that will allow them to be recognized.

Representation and algorithm

- At this level, scientists are looking for an algorithm, or a set of algorithms, that could transform the input into the output.
- In Marr's theory the output representation is derived from a series of increasingly complex and sophisticated representations, which he terms the **primal sketch**, the 2**.5D sketch**, and the **3D sketch**.
- Theses stages correspond to what is still described in the computational vision field as an **image-based stage**, a **surface-based stage**, and an **object-based stage**.

The image based stage

- It includes image processing such as detecting edges, lines, curves, linking local edges, defining 2-D regions in the image;

- Crucial point: the primitive representational elements represent information about the 2-D structure of luminance image (edges and lines defined by differences in light intensity) rather than information about physical object.

The surface-based stage

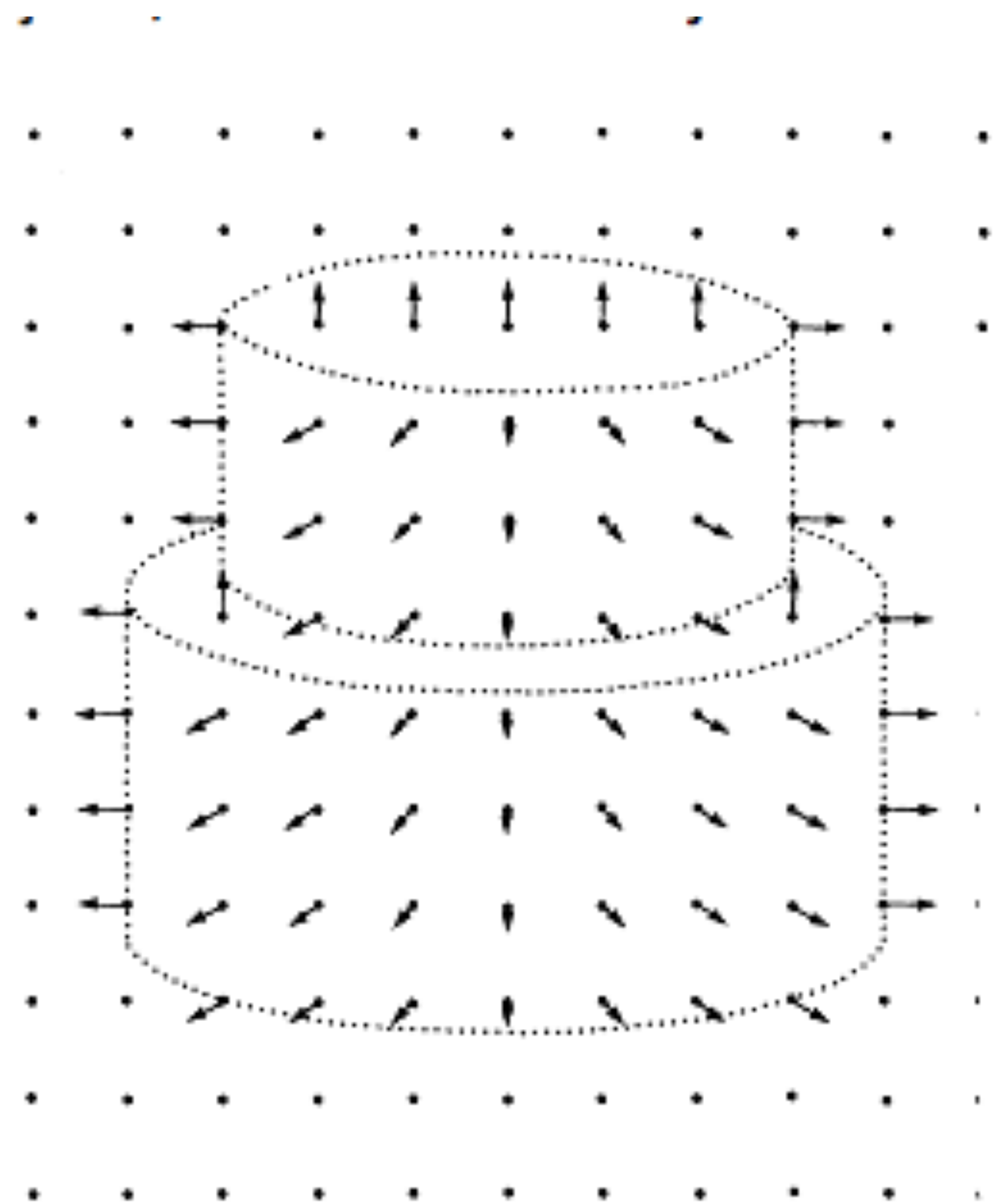
- At this stage, intrinsic properties of visible surfaces in the external world that might have produced the image-based representation are recovered: so the spatial layout of 3-D visible surfaces is represented.

- The concept of a spatial layout — that is, of a spatial distribution of **surfaces (rather than objects)** in the 3-D environment — is first due to James J. Gibson, and it was also advocated by Marr, who implemented it in computer simulations (1978).

- Marr named this representation 2.5-D sketch because it lies between image-based and object-based representations.

- This is a **viewer-dependent, perspectival, representation**: it contains only information about the surfaces that are visible from the agent's viewpoint.

- Most current theories treat such surfaces as sets of small, locally flat, oriented pieces:



The object-based stage

- Unseen surfaces are included in the representation, which means that it's a truly object-based, 3-D representation.
- Objects are represented as three-dimensional, viewer-independent, entities — as arrangements of some set of primate 3-D shapes.
    - Volumetric primitives: descriptions of three-D volumes, including information about unseen surfaces.
    - 3-D geometry.
    - Object-based reference frame (vs viewer-centered reference frame at the 2.5-D stage).

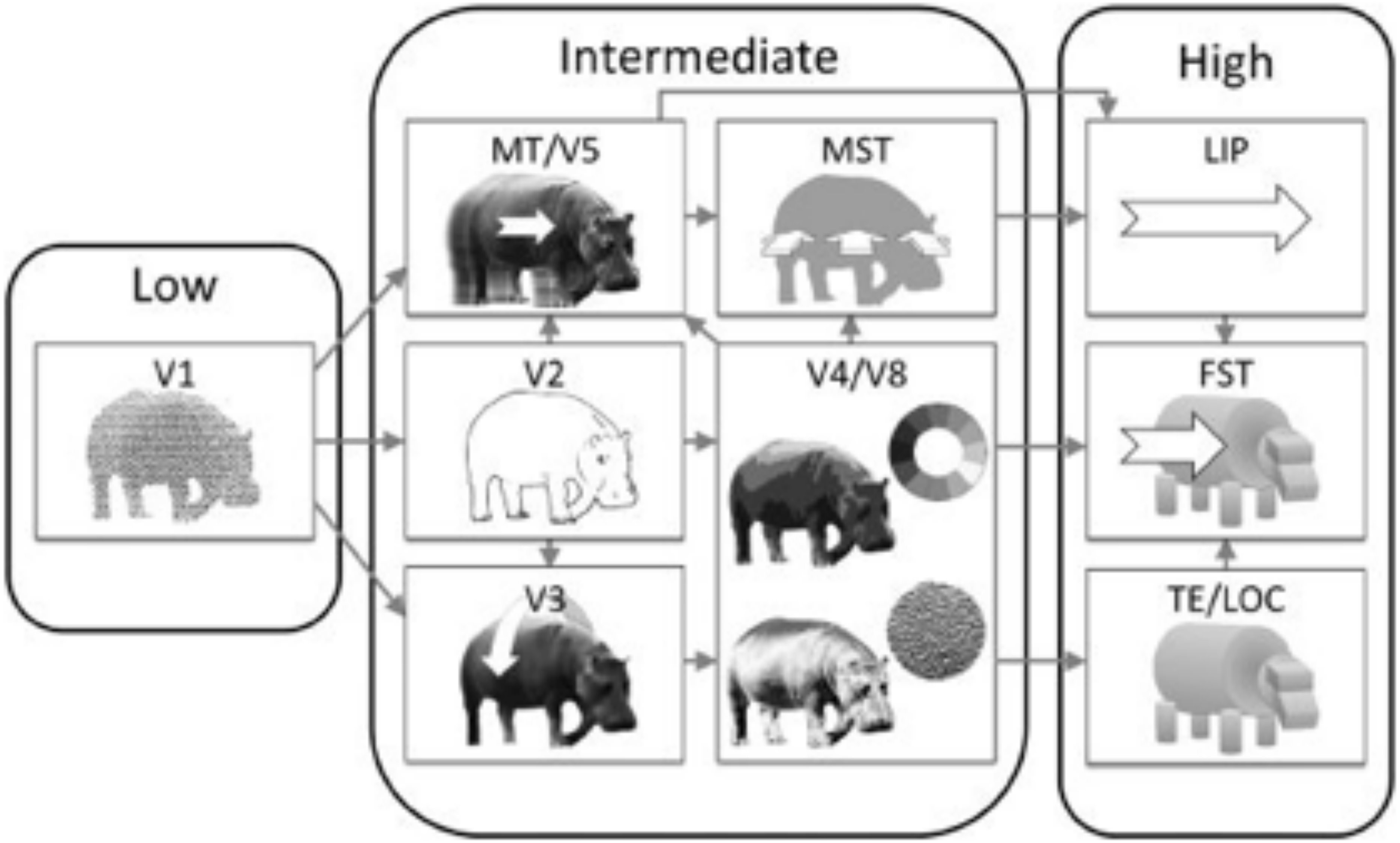


FIGURE 3 A simplified diagram of the visual system.