From Symbols to Knowledge Systems: A. Newell and H. A. Simon's Contribution to Symbolic AI

Luis M. AUGUSTO*

Editor-in-Chief

Journal of Knowledge Structures & Systems

July 2021

Vol.: 2 Issue: 1 Pages: 29-62

Abstract

A. Newell and H. A. Simon were two of the most influential scientists in the emerging field of artificial intelligence (AI) in the late 1950s through to the early 1990s. This paper reviews their crucial contribution to this field, namely to symbolic AI. This contribution was constituted mostly by their quest for the implementation of general intelligence and (commonsense) knowledge in artificial thinking or reasoning artifacts, a project they shared with many other scientists but that in their case was theoretically based on the idiosyncratic notions of symbol systems and the representational abilities they give rise to, in particular with respect to knowledge. While focusing on the period 1956-1982, this review cites both earlier and later literature and it attempts to make visible their potential relevance to today's greatest unifying AI challenge, to wit, the design of wholly autonomous artificial agents (a.k.a. robots) that are not only rational and ethical, but also self-conscious.

Key words: Symbols and Symbol Structures; (Physical) Symbol Systems; Representation; The Knowledge Level; Knowledge Systems

1 Introduction

The present article reviews the contribution of the seminal work of A. Newell and H. A. Simon to symbolic AI, in particular to its subfield of knowledge representation and reasoning (KRR). This academic subfield is said to have been kick-started by McCarthy (1959)–see, e.g., Brachman & Levesque (1985)–but these two scientists, more than any other, focused their joint work on the nature of representation and

*✉ luis.ml.augusto@gmail.com

knowledge. They did not start by addressing these notions directly; their work in symbols and knowledge representation was a natural evolution of their initial focus on problem solving and general intelligence. This, too, was already revolutionary work; as M. Veloso recently put it in an interview when asked what researcher had inspired her in her research:

> For me, the work of Herb Simon and Allen Newell on problem solving in the mid 50s was fascinating. Before that, Computer Science was about very specific areas such as cryptography or algorithms. Newell and Simon defined, in general, how a computer can solve any problem. The task of the program is to end up in a state in which the objective is satisfied. Topics that are currently very popular such as planning, search, learning from experience, and advice all started from there. I still think in this way in my current research. (Veloso, 2016)

The salience of knowledge and representation in their joint work was best expressed by A. Newell in the Presidential Address of the American Association for Artificial Intelligence delivered at Stanford University in 1980 on the occasion of the First National Conference on Artificial Intelligence (AAAI-80):

> Only two foci are really possible for a presidential address: the state of the society or the state of the science. I believe the latter to be the correct focus. ... [O]ur main business is to help AI become a science–albeit a science with a strong engineering flavor. Thus, though a president's address cannot be narrow or highly technical, it can certainly address a substantive issue. That is what I propose to do. I wish to address the question of knowledge and representation. (Newell, 1982)

Despite the fact that the eclectic cluster known today as knowledge technologies–a product of the recent industrialization of knowledge (see Augusto, 2020c)–is substantially different from the knowledge systems envisaged by A. Newell and H. A. Simon, current trends in knowledge-based AI can greatly benefit from their AI project. This is especially so for the ongoing megaproject in artificial agency in real-world environments. In my view, one–perhaps the most important–of the few intrinsic characteristics of thinking beings with respect to the environment is representation, namely of the symbolic kind (e.g., Augusto, 2014). If we manage to capture, in a formal way, how thinking beings, and in particular humans, represent not only the environment but also their presence and acting in it in such a way that this representation equates with knowledge of the environment and the self, then the main challenge in KRR is halfway won; the other half is to find the way(s) to implement this intrinsic ability in artificial agents.[1] For this effort, generally known since Harnad (1990) as *the grounding problem*, Newell and Simon's contribution may be of import.

This accounts for the circumscription of this review to their conception of *symbols* and *(physical) symbol systems* (see Section 3) and to the foundational notion in AI

[1] I trust I am not alone in holding this belief; for instance, the words "represent(s)" and "representation(s)" occur 35 and 114 times, respectively, in Chella et al. (2019), a recent collection of texts on robot consciousness.

of *the knowledge level* (Section 4).[2] In reviewing their contribution to symbolic AI, I begin by elaborating on aspects that are, in my view, essential to contextualize Newell and Simon's AI work, which largely aimed at replicating in machines the cognitive behavior of humans from the viewpoint of symbol manipulations and knowledge structures. Two articles are central in this review, to wit, Newell (1980, 1982), but their analysis requires both earlier and later work by both authors to be also more or less deeply invoked.[3] In particular, Newell (1990), in which this author reviews and reflects on his joint work with H. A. Simon, is a major source for this review. As is usual in a review article, I end with a few Conclusions, but leave some homework for the reader.

1.1 General Intelligence and (Commonsense) Knowledge in the AIs

One effort that can be said to unify the large field of AI is that of replicating–or just simulating–*general intelligence* in artificial agents or computing machines (machines, for short). In other words, this is the problem of conceiving some machine M that performs task, or solves problem, P_0 by finding solutions to (sub-)tasks/(sub-)problems $P_1, P_2, ..., P_k$, where k is a natural number as large as possible, so that $\mathcal{PS}_M = \bigcup_{i=0}^{k} P_i$ for $\mathcal{PS}_M$ denoting the *problem space* that can be effectively processed by M. This, in turn, requires that M apply a wide variety of cognitive capabilities. In particular, the large variety of skills a higher vertebrate typically is capable of applying in a constantly changing environment when acting in it with its survival in view can be said to be the necessary basis of "true intelligence" (e.g., Brooks, 1991).

For instance, a hunting hawk needs to be able to maintain the representation of its prey–say, a rabbit–relatively invariant from detection at high altitude, through downward acceleration to approximation, and finally to capture of the rabbit that in the meantime usually has detected the danger from above and is rapidly changing position in space and direction in motion, an apparently effortless feat that took millions of years to perfect. Several intelligent features are here at play: The classification of the rabbit as prey remains invariant in spite of changes in altitude and motion and the hawk reacts adequately to the static or dynamic properties of the rabbit. Moreover, if the rabbit hides in a bush, the hawk will reason that it is seeking escape by hiding there and will try to make it leave the bush, as entering it would, so it likely reasons, cause more or less serious physical injury to itself. This example shows that for a hunting animal to solve the problem of how to catch its prey–what Brooks (1991) calls "the essence of being and reacting"–there are several (sub-)problems to be solved that require different intelligent skills.

[2] While these aspects can certainly be discussed from the viewpoint of cognitive architectures, and of production systems in particular, both prototypically embodied in Soar, these are more immediately connected to the concept of *control structures* than to *knowledge structures and systems* (cf. Newell, 1990), reason why they are essentially left out in this review. Yet another reason for this omission is that H. A. Simon's contribution to Soar appears to have been rather "implicit," as A. Newell states in the Preface to Newell (1990); according to this author, his collaborator was not particulary interested in cognitive architectures, being rather skeptical as to their scientific importance. See Laird & Newell (1983) for the original presentation of Soar and see Laird et al. (1987) for early progress; Newell (1990), a major source for this review, has a comprehensive elaboration on Soar, which has however been superseded by Laird's (2012) "definitive presentation."

[3] Below, the anecdotal reasons why these articles were not co-authored by H. A. Simon are given.

In this illustration, the hawk can be said to *know* that this particular prey behaves differently from other prey; the hawk does not expect rabbits to fly, so that plausibly it *knows* that rabbits, contrarily to pigeons, do not fly. (Further examples of knowledge can be easily extracted from the illustration.) If it is true that intelligence and knowledge are different things, they are linked in many and various ways. In particular, intelligence is a property that should be applied *only* to, and to *all*, knowledge-level systems. In fact, as phrased in Newell (1990):

> A system is *intelligent* to the degree that it approximates a knowledge-level system.

From this perspective, the hawk can be seen as a *natural* knowledge-level system.

Arguably, AI's highest aim is to create *artificial* knowledge-level systems, i.e. artifacts (man-made machines) exhibiting general intelligence and possessing (commonsense) knowledge. If these are identified as mental properties, then the effort of replicating them in machines constitutes what can be called *strong AI*, it being meant by this that machines capable of general intelligence and (commonsense) knowledge would actually possess a mind, whereas their simulation alone is called *weak AI*–a distinction that has been the object of multiple interpretations since its original conception (Searle, 1980). Be it as it may, these two perspectives on AI actually define two potentially diverging *AIs*. Although H. A. Simon and A. Newell are typically not invoked in the debate over the two AIs their work was developed in the theoretical framework of strong AI, an aspect that, albeit not emphasized here, constitutes the *fil rouge* of this review.

1.2 Task Environments, Search Spaces, and Representations

The key word in the hawk illustration above is *representation*. But this word is key in a sense other than just *visual* representation (which it is, too): Firstly, the hawk detecting a rabbit in an environment can be said to represent that this is a kind of prey suitable to its feeding needs and habits; secondly, from targeting the rabbit in its initial position to capturing it in accelerated escape, the hawk can be said to represent both the permanence of its prey in time and its likely moving in space. Moreover, the hawk can generalize these representations to prey with such different habitats and escape or camouflage manners as a pigeon, a chameleon, or even a small fox.

A human navigating the many food sections of a supermarket–let us call this the *search space*–has the same objective of the hunting hawk and exhibits the same skills, though chasing the food items is not a routine in most societies today: When shopping for food in a supermarket (the *task environment*), the human has representations with respect to the persistence in time and the permanence in space of the food items available there, the need to walk around the sections, the need to extend the arm to collect items from the shelves and lower it to put them in the shopping cart, etc. Because the food items in a supermarket are not fleeing this human can analyze them at length for cost, ingredients, aesthetics, etc., in order to make the decision whether or not to put them–perhaps temporarily–in the shopping cart.

In either case, both the human and the hawk are here trying to solve a crucial problem (obtaining food). To play on the safe side, we limit our lexicon to *representations* in the case of a hawk; in the case of a human, we would invoke *(justified true)*

beliefs and speak of *knowledge representations* (Augusto, 2020b). In any case, systems exhibiting such wide-ranging capabilities as a hawk or a human can be characterized "in terms of their having *knowledge* and behaving in light of it" (Newell, 1990).

1.3 Today's AI Unifying Challenge

A final introductory remark: The illustrations above featuring hunting hawks and shopping humans are not Newell and Simon's, but they could very well be, given their assumption that complexity of behavior is not so much to be attributed to an agent's intrinsic abilities but to the environment, or problem space, in which it acts.[4] Newell and Simon's work concentrated in large measure on specifically human cognitive task environments, such as theorem proving, which do not require motor abilities. But *the times, they have a'changed*, and the design of autonomous artificial agents, a.k.a. robots, that not only act both rationally and ethically but have also some form of self-consciousness, is possibly today's most challenging unifying AI project (e.g., Chella et al., 2019; Lemaignan et al., 2017).[5] This effort requires that thinking and reasoning organisms other than humans be studied with the objective of replicating or simulating their intelligent behavior, an aspect that at least H. A. Simon had realized implicitly, if not explicitly so. Tellingly, compare the following passage of Simon (1996)–my emphasis–with the original passage in Simon (1969), given below in a footnote:[6]

> [There is] evidence that there are only a few "intrinsic" characteristics of the inner environment of thinking *beings* that limit the adaptation of thought to the shape of the problem environment. All else in thinking and problem-solving behavior is artificial–is learned and is subject to improvement through the invention of improved designs and their storage in memory.

2 Antecedents and Early Work: Classical Computation and Human Problem Solving

The idea that can be said to characterize Newell and Simon's symbolic, or classical, AI project can be summarized in the following way: A machine can be said to have knowledge of X if it has the (abstract) set of all the possibly derivable symbolic expressions (for short: expressions) related to X; for instance, machine M can be said to know (how to play) chess if, from any position represented by means of expressions,

[4] See, for example, H. A. Simon's (1969) example of an ant exploring its way on a beach surface. Also, in Newell (1980), an animal circling its prey is given explicitly as an example, together with "a person in conversation with another, a player choosing a chess move, a student solving a physics exercise, a shopper bargaining with a seller," of purposeful behavior as a function of the environment. This sanctions my choice of real-world illustrations already given above and those to be given below.

[5] See also Floridi (2005) and compare the "Floridi Test" with the Turing Test (see below).

[6] "[There is] evidence that there are only a few 'intrinsic' characteristics of the inner environment of thinking *man* that limit the adaptation of *his* thought to the shape of the problem environment. All else in *his* thinking and problem-solving behavior is artificial–is learned and is subject to improvement through the invention of improved designs." If the reader did not notice, I call their attention to the fact that "man" was replaced by "beings" in Simon (1996) and the third person pronoun "his" was removed (actually, already so in the 1981 second edition). Incidentally, this book is now on its 4th edition (Simon, 2019), an updating of the 3rd edition with a new introduction by J. E. Laird.

M can derive other positions also represented by means of expressions to its advantage as a player. Intelligence is then the skill to derive all these possible expressions (see Newell, 1980). This can be summed up as the *Turing paradigm*, whose core concepts are the universal Turing machine and the Turing test.[7] But Simon and Newell's belief was that theirs, just like all AI quests, was one of empirical inquiry, namely one of cognitive psychology. This explains their project of simulating, or actually realizing, human cognitive skills, in particular general intelligence and (commonsense) knowledge, in machines taken as physical symbol systems. I next briefly analyze these central aspects to their own work.

2.1 The Turing Machine: The Mid 1930s

A. Turing's (1936-7) famous machine, as avowed by Newell & Simon (1976), lay at the root of their project of creating (universal) symbol systems:

> Our present understanding of symbol systems grew ... through a sequence of stages. Formal logic familiarized us with symbols, treated syntactically, as the raw material of thought, and with the idea of manipulating them according to carefully defined formal processes. The Turing machine made the syntactic processing of symbols truly machine-like, and affirmed the potential universality of strictly defined symbol systems. The stored-program concept for computers reaffirmed the interpretability of symbols, already implicit in the Turing machine. List processing brought to the forefront the denotational capacities of symbols, and defined symbol processing in ways that allowed independence from the fixed structure of the underlying physical machine. By 1956 all of these concepts were available, together with hardware to implement them. The study of the intelligence of symbol systems, the subject of artificial intelligence, could begin.

How do a human computer and a computing machine solve a problem, say, $2 + 3 =$? For the symbolic paradigm, given an input of strings of symbols, called expressions, both a human and a machine output expressions; between the input and the output there is a well-defined sequence of actions or steps called an *algorithm*, by means of which the input expressions undergo a series of manipulations aiming at obtaining output expressions, where for some input expression w and some output expression z, we have either $w = z$ or $w \neq z$, but in any case we have $w \vdash^* z$, for $\vdash$ denoting a step in the algorithm and * denoting zero or more (steps). In order to illustrate the statement in the quotation above that the Turing machine made the processing of symbols truly machine-like I focus on some of its basic aspects.

As is well known, a Turing machine is a 7-tuple

$$M_T = (Q, \Gamma, \#, \Sigma, q_0, A, \delta)$$

where $Q = \{q_0, q_1, ..., q_k\}$ is a finite set of states, Γ is the tape alphabet, $\# \in \Gamma$ is the blank symbol, $\Sigma \subseteq \Gamma$ is the input alphabet, $A \subseteq Q$ is the set of accepting (or halting) states, and $\delta : (Q \times \Gamma) \longrightarrow (Q \times \Gamma \times \{L, R\})$, where L, R denote direction

[7] In Augusto (2020a), I explain why the Turing paradigm can be said to be classical.

(left and right, respectively), is the transition function.[8] The *computer model* for a Turing machine M_T consists simply of (i) a control box programmed by δ equipped with (ii) a read-and-write head. This head reads an input string starting on the left of an input tape that is infinite to the right (or to both sides; cf. Fig. 1). When the machine reads a symbol a while in state q, it switches into state p, replaces symbol $a \in \Gamma$ by symbol $b \in \Gamma$, and moves one tape cell in direction $D = \{L, R\}$, so that we have

$$\delta\left(q, a\right) = \left(p, b, D\right).$$

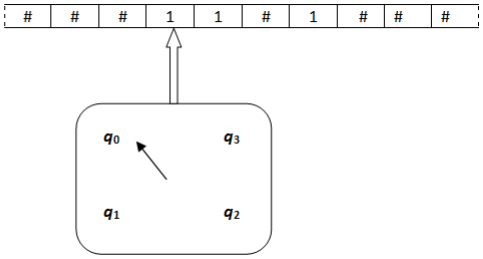

Figure 1: Computer model for a Turing machine. (Source: Augusto, 2020a.)

Given an input string w, the complete sequence for $q_0 w \vdash_T^* q_i z$, where q_0 is an initial state and q_i is an accepting state (or a state with no move to another one), is called a *computation*. For example, $M_T = \left(\{q_0, q_1, q_2, q_3\}, \{1, \#\}, \#, \{1\}, q_0, \{q_3\}, \delta\right)$ adds two positive integers by performing the following computation over the input 11#111##...:[9]

$$q_0\underline{1}1\#111\# \vdash q_0 1\underline{1}\#111\# \vdash q_0 11\underline{\#}111\# \vdash q_1 111\underline{1}11\# \vdash q_1 1111\underline{1}1\#$$
$$\vdash q_1 11111\underline{1}\# \vdash q_1 111111\underline{\#} \vdash q_2 11111\underline{1}\# \vdash q_3 11111\#\underline{\#}$$
$$\equiv q_0 11\#111\# \vdash^8 q_3 11111\#$$

The behavior of the machine is as follows: M_T can only read one symbol (denoted by $\underline{\sigma}$ for some $\sigma \in \Gamma$ being read) and take one step at a time. Given the input 11#111##..., which corresponds to the two integers 2 and 3, in order to add both strings M_T has firstly, after reading the two initial 1s, to delete the blank space between them, replace it with 1, and then keep moving right until it finds the first empty cell (denoted by #); this done, M_T goes back to the last 1, deletes it, and halts after moving the read-and-write head to the right. The output string is 11111 followed by infinitely many blank symbols. Figure 2 shows the state diagram for this M_T.

In performing the computation above, it can then be said that machine M_T *defines* the function $f\left(m, n\right) = m + n$ for an arbitrary input $1^m\#1^n\#\#...$ if

$$q_0 1^m\#1^n\# \vdash_T^{(m+n)+3} q_3 1^m 1^n\#.$$

[8] See Augusto (2020a) for further discussion of the Turing machine. For a comprehensive analysis of Turing (1936-7), see Petzold (2008); see also Copeland (2004). See Augusto (2019) for a shorter discussion of Turing's seminal article.

[9] The integers are represented in unary notation $1^n = \underbrace{111...1}_{n}$; e.g. $1^3 = 111 = 3$. Subscript T is omitted in $\vdash$.

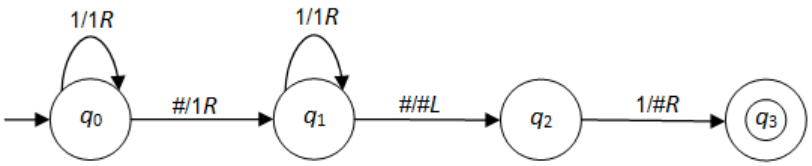

Figure 2: A Turing machine that computes the function $f(m,n) = m + n$ for $m, n \in \mathbb{Z}^+$.

When this is generalized to any computable function, we speak of the *Turing Thesis*, according to which any effectively computable function is computable by a Turing machine. Universality of this thesis is attained via the *universal Turing machine* (UTM), a Turing machine that, given as input a string $\xi = \langle M_T, w \rangle$ (where $\langle \cdot \rangle$ denotes an encoding, paradigmatically in binary code) of both an arbitrary Turing machine M_T and a string w, (i) accepts ξ if and only if M_T accepts w and (ii) outputs $\langle y \rangle$ if M_T accepts w and outputs y. In other words, a UTM computes the functions of all the Turing machines, showing that a symbol system is capable of universality.

2.2 Computing Machinery and the Turing Test: Mid-Century

Although the computation of a Turing machine solving the problem $11+111 = 11111$ and that of a human for $2+3 = 5$ are very different, the two problems are informationally equivalent and both the human brain and the Turing machine solve the problem effectively, namely by a sequence of discrete steps or operations, constituting discrete states or configurations, over discrete symbols and strings thereof. Also, it should be noted that solving a mathematical problem is considered a higher problem than, say, collecting some food item from a supermarket shelf; in particular, "higher," more recent areas of the human brain are believed to be recruited for the former. But can a Turing machine, in particular a UTM–or as Turing called it, a ULCM (Universal Logical Computing Machine)–actually have a (human-like) mind?

In 1950, A. Turing devised a test, now known as the Turing Test, that would provide us with a definite positive answer: Given a computer (i.e. a UTM), a human, and a human interrogator, can the latter identify which is which (computer or human) simply by the answers on a screen given by them? If not, then it is because the computer *imitates* the human so well that the interrogator is fooled into concluding that it is a human.

The point of the Turing Test is that if some machine (say, a computer) *behaves* as intelligently as a human, then it is because it *thinks* like a human–Turing's "*criterion* for 'thinking'" (Turing, 1950)–, and no ontological difference between "human intelligence" and "machine intelligence" can be postulated. In particular, one could say that the computer can "imitate a brain."

Turing (1950) speaks explicitly of a computer, but it cannot be concluded that it is the digital computer that he is speaking of, as already in Turing (1948) by "computing machinery" he actually meant what would today be called artificial neural networks, or connectionist models. Despite this caveat, the Turing Test soon became associated

with symbolic AI (vs. connectionist AI).[10] This was a propeller for what would come to be spoken of as *good old-fashioned AI* (abbr.: GOFAI), an expression originating in Haugeland (1985). Although this acronym is often used disparagingly, in fact it designates an interesting and highly productive alliance between studies in human psychology and AI in which the linking bridge is to be found in *information processing*: Both humans–as postulated by mid-1960s emergent cognitive psychology and early 20th-century Gestalt psychology–and computers are information-processing systems. In terms of the Turing Test, one can say then that a computer stands the test if and only if it processes information like a human–brain or mind, depending on how precise one wants to put this commonality.

2.3 Information Processing in Psychology and AI: The 1960s

The acknowledgment of this commonality was stated from both sides: U. Neisser, the father of cognitive psychology–whose main achievement was to make talk of *mind* acceptable in scientific circles by postulating both internal (i.e. *mental*) entities and systems like beliefs, memory, etc. and internal processes on them–wrote in his extremely influential book entitled precisely *Cognitive Psychology*:

> Although information measurement may be of little value to the cognitive psychologist, another branch of the information sciences, computer *programming*, has much more to offer. A program is not a device for measuring information, but a recipe for selecting, storing, recovering, combining, outputting, and generally manipulating it. As pointed out by Newell, Shaw, and Simon (1958), this means that programs have much in common with theories of cognition. Both are descriptions of the vicissitudes of input information. (Neisser, 1967)

Neisser referred to this as the "program analogy" and considered it to be theoretically useful in the sense that borrowing, even if only loosely, notions such as *parallel processing, feature extraction*, etc. from programmers could provide a means to test some psychological theories; but not to simulate human thinking, as he briefly rants against the Logic Theorist (see Neisser, 1967, pp. 8-9). From their AI perspective, Simon and Newell (1971) wrote:

> Information-processing explanations refer frequently to processes that go on inside the head–in the mind, if you like–and to specific properties of human memory: its speed and capacity, its organization. These references are not intended to be in the least vague. What distinguishes the information-processing theories of thinking and problem solving described here from earlier discussion of mind is that terms like "memory" and "symbol structure" are now pinned down and defined in sufficient detail to embody their referents in precisely stated programs and data structures.

The achievement mentioned above of cognitive psychology was actually a paradigm shift, in the Kuhnian sense (Kuhn, 1962), in psychology: Against the black-box model

[10] See Smolensky (1987) for a critical, albeit one-sided, comparative analysis.

of behaviorism mediating between the stimulus and the response, abbreviated as S-R, cognitive psychology postulated in fact the existence of *mental representations*, whether stimulus-based or other.[11] In AI, too, the notion *representation*–"a symbol structure with definite properties on which well-defined processes can operate to retrieve specific kinds of information" (Simon & Newell, 1971)–was readily adopted and adapted, albeit solely in symbolic AI.

In this, it can be postulated that both human brains and computers of the von Neumann type can process information by means of data types (namely, lists and arrays) and operate on these by means of associative operations such as $find\,(X, Y)$ (e.g., "Find (the name of) the father of John.") in such a way that representations in both media–human brains and computers–are informationally, even if not computationally, equivalent. This presupposes that there is an *information-processing language* whose set of processes or operations is common to both humans and computers, it being the case that from an input expression in this language to an output expression thereof there is a sequence of (transformations or compositions of) expressions–a computation–that can be traced in detail.[12]

This fundamental commonality of both human and machine computations assumed by Simon and Newell had already been empirically established via the General Problem Solver (GPS), a computer program that both accomplished the same tasks as humans and simulated the processes used by these when accomplishing the tasks (Newell & Simon, 1961).[13] As a matter of fact, this had already been shown, at a more incipient level, in Newell & Simon (1956): The Logic Theorist (LT) was an information-processing system that was able to discover, using heuristic methods, proofs for theorems in symbolic logic; actually, it could prove in sequence most of the 60 odd theorems in Chapter 2 of Russell and Whitehead's celebrated *Principia mathematica* (Whitehead & Russell, 1910). And importantly, both GPS and LT were implemented in IPL, an abbreviation for Information Processing Language (see, e.g., Newell et al., 1961); besides the performance of arithmetic operations and other input/output operations, IPL provided symbolic means for operations such as finding and erasing attributes or locating the next symbol in what would soon become a cen-

[11] "As used here, the term 'cognition' refers to all the processes by which the sensory input is transformed, reduced, elaborated, stored, recovered, and used. It is concerned with these processes even when they operate in the absence of relevant stimulation, as in images and hallucinations. Such terms as *sensation*, *perception*, *imagery*, *retention*, *recall*, *problem-solving*, and *thinking*, among many others, refer to hypothetical stages or aspects of cognition." (Neisser, 1967)

[12] See the brief discussion of the Turing machine above.

[13] GPS (Newell & Simon, 1961; Newell et al., 1959, 1960) dealt with task environments constituted by objects that could be transformed by means of operators; it detected differences between objects and organized the information into goals, collections of information (i) defining what goal attainment was, (ii) making available the information relevant to the attainment of the goal, and (iii) relating the information to other goals. These, in turn, came in three types: (1) transform object A into object B; (2) reduce difference D between object A and object B; (3) apply operator Q to object A. In Newell & Simon (1961), the task accomplished by GPS and in whose solution it simulated human thinking (a student in engineering) was constituted by logical expressions upon which rules of inference (operators) were carried out in order to, say, change a connective (difference). More specifically, GPS proved the deduction of $\neg\,(\neg Q \wedge P)$ from $(R \rightarrow \neg P) \wedge (\neg R \rightarrow Q)$ by means of the application of rules of inference of a given calculus. In effect, the task accomplished was

$$(R \rightarrow \neg P) \wedge (\neg R \rightarrow Q) \vdash^{14}_{GPS} \neg\,(\neg Q \wedge P)$$

to use the notation for computations defined above in Section 2.1.

tral component of programming languages, to wit, lists. In Simon & Newell (1962), the authors wrote:[14]

> It is no longer necessary to argue that computers can be used to simulate human thinking or to explain in general terms how such simulation can be carried out. A dozen or more computer programs have been written and tested that perform some of the interesting symbol-manipulating, problem-solving tasks that human beings can perform and that do so in a manner which simulates, at least in some general respects, the way in which humans do these tasks. Computer programs now play chess and checkers, find proofs for theorems in geometry and logic, compose music, balance assembly lines, design electric motors and generators, memorize nonsense syllables, form concepts, and learn to read.

2.4 The Representational Theory of Mind: The Mid 1970s

According to the short introduction above, an information-processing system works on–i.e. produces, transforms, stores, and retrieves–symbol structures, so representations must be information-bearing symbol structures. Beliefs, say, are information-bearing structures:[15] For instance, if John believes that it is raining, he will very likely take an umbrella with him when going out to work. This entails that concepts such as RAIN and UMBRELLA are also information-bearing structures, and are operative (i.e. behaviorally causal) regardless of whether John perceived that it is raining when looking out from his bedroom window or just nurtures the unjustified belief that it is raining; this means that both percepts and hallucinations are also information-bearing structures. This illustrated analysis could be extended to a plethora of other so-called *mental representations* (e.g., ideas, desires, emotions, schemas, ...), and we would end up with the same conclusion: All mental representations are information-bearing structures. This–the *representational theory of mind* (RTM), which is almost as old as philosophy, going back to Aristotle (see, e.g., Augusto, 2006)–is highly controversial in philosophical and cognitive-science circles, and its adoption by AI cannot but be even more troublesome.[16] So what is the pay-off of adopting (or adapting) it, as Newell and Simon did?

Before trying to answer this question, it will be useful to remark that representations come in many forms, of which some are equivalent and some are not. In Simon (1978), two kinds of equivalence for representations are proposed:

1. Two representations R_1 and R_2 are said to be *informationally equivalent* if the transformation of R_i into R_j, $i, j = 1, 2$, $i \neq j$, entails no loss of information or, what is the same, if each can be constructed from the other. (E.g.; (R_{1a}) "Distance equals average velocity times time" and (R_{2a}) "$s = vt$" are informationally equivalent; a two-dimensional depiction of the Necker Cube (R_{1b}) and the three-dimensional information stored about its vertices, edges, and faces such that the front and back faces are fixed (R_{2b}) are not informationally equivalent.[17])

[14] The contribution of J. C. Shaw to the creation of IPL should be noted.

[15] In a more philosophical lingo, "information-bearing" can be replaced by the term "intentional."

[16] Fodor (1975) is a cornerstone of RTM. See Stich (1992) for a state of the debate in the early 1990s.

[17] Looking at the Necker Cube, the subject normally alternates between the front and back faces.

2. Two representations R_1 and R_2 are said to be *computationally equivalent (in the extended sense)* if the information extracted from both with the same amount of computation up to a fixed factor of proportionality is the same. (E.g., a geometrical problem can be represented as a theorem to be proven from a set of propositions and axioms (R_{1c}), a pair of simultaneous equations (R_{2c}), or as a drawing on a sheet of paper (R_{3c}). None of these is computationally equivalent, though they are all informationally equivalent; for instance, humans would likely find R_{1c} the most difficult way to solve the problem, and R_{3c} the easiest.)

It should now be remarked that R_{1c} and R_{3c} are believed to be represented in different brain hemispheres, with R_{1c} being represented on the left hemisphere and R_{3c} on the right hemisphere. As there have not been found any tissue differences between the two brain hemispheres, and as the representations "typical" of one hemisphere can be taken over by the other (in case of, say, lesion), it is safe to conclude that representations that are not computationally equivalent can be processed by a single physical system (in this case, the brain). On the other hand, any of the representations above can be employed by information-processing systems as physically different as human brains and many kinds of computers of the von Neumann type. The conclusion follows: "There is no close relation between a system's 'hardware' and the representations it may employ" (Simon, 1978).

Importantly, although the tasks solved by GPS were limited to a specific class, to wit, tasks that had a clearly defined goal and whose solution required multiple steps, this commonality was taken in two directions: Firstly, it sanctioned the designing of a large project aiming at explaining how particular human intelligent behavior came about and what the mechanisms are that enable this behavior with a view to the improvement of human performance, in particular of learning and decision making. Thus, this project continued the previous work of this duo having for subject the psychology of human thinking (e.g., Simon & Newell, 1962). Secondly, it aimed to use this empirical evidence to design machine programs that can better *approximate* human intelligent behavior, with the practical objective of designing programs that can solve problems that are difficult for humans. The point to retain from this bidirectional approach is that the AI explanations of human problem-solving behavior were truly so only to the extent that the processes used by programs like LT and GPS were the same as those used by humans. And these are *heuristic* processes. These, in turn, are reducible to search processes in what can be seen as search trees whose nodes are *states of knowledge*, so that ultimately a program that approximates human intelligence or problem-solving abilities is a program whose core component is to be found at the *knowledge level*. This is the level at which an agent has mental representations like beliefs, desires, etc., but below these there is the level at which they are constituted as such: The *symbol level*.

3 Symbol Systems

Symbols in general were well known in the 1960s. As a matter of fact, a purely symbolical language–what we today call a formal language (see Augusto, 2020a)–had been envisaged by G. W. von Leibniz as early as in the 17th century: He called it *lingua universalis*, a universal conceptual language based on a general theory of

MEMORY	Structures that contain symbol tokens Independently modifiable at some grain size
SYMBOLS	Patterns that provide access to distal structures A symbol token is the occurrence of a pattern in a structure
SYMBOL OPERATIONS	Processes that take symbol structures as input and produce symbol structures as output
SYMBOLIC INTERPRETATION	Processes that take symbol structures as input and execute operations
CAPACITIES	Sufficiency (enough memory and symbols) Completeness of composability and of interpretability

Figure 3: A symbol system. (Adapted from Newell, 1990.)

symbols (the *characteristica universalis*) that could serve as a medium for a wholly mechanical calculus (the *calculus ratiocinator*). Most importantly, and as avowed in Newell & Simon (1961), A. Turing had shown that, given a suitable device such as a Turing machine, symbols can be copied and transformed into expressions in such a way that symbol patterns can be given all the essential characteristics of linguistic symbols. Thus, one may ask where exactly the novelty lay in Newell and Simon's work with respect to symbols. Firstly, they made it clearer what the relation between symbols and representation is, a topic of obvious centrality for symbolic AI; secondly, they gave symbols such an importance that they actually saw them as correlated with not only the notion of general intelligence, but also the concept of mind. Both aspects were elaborated on based on their conception of a symbol system (SS).

3.1 Symbol Systems are Transformers of Patterns

The lesson that Newell and Simon drew from both programs, LT and GPS, which can be considered to be two of the earliest AI programs, was that computers are not just manipulators of numbers but are first and foremost general manipulators of symbols and, as such, transformers of patterns. While today's readers may be unimpressed, given the contemporary profusion of not only automated provers at hand for a plethora of classes of logics but also "intelligent" technology systems of other types, it should be remarked that before LT and GPS computations were carried out mostly on numbers, namely by the so-called calculators, though other task environments such as chess had already started to be explored.

In essence, a SS transforms a set of symbols in a memory through time, requiring for this end a memory where symbols are stored; this is what constitutes its *sufficiency* (cf. Fig. 3). Additionally, a SS requires a set of operations over, and an

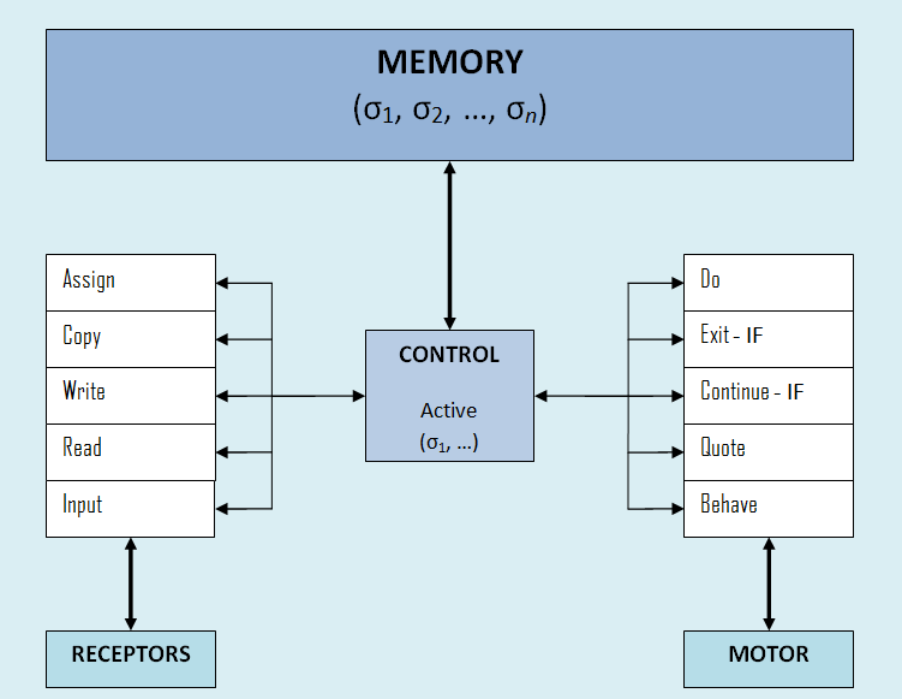

Figure 4: A paradigmatic SS. The σ_i denote abstract symbols. (Adapted from Newell, 1980.)

interpretation of, symbols such that it is *complete* both in terms of composition (any legal symbol structure can be constructed in the SS) and interpretability (input legal symbol structures will output operations). Responsible for the completeness of the SS is a *program*, or a collection of programs, sets of symbol structures that are to be serially interpreted with completion of a given goal in view. Figure 4 shows a paradigmatic SS in some detail. As can be seen in this schema (Fig. 4), a SS is composed of a *memory*, a *control*, an *input*, an *output*, and a set of *operators* (*Assign*, *Copy*, etc.). (See Newell, 1980, for the detailed description of these components and how they interact in a paradigmatic SS, namely by means of a program.)

3.2 Designation and Interpretation

But what are symbols per se? As it is, symbols are extremely simple entities: "physical patterns that can occur as components of another type of entity called an expression (or symbol structure) ... composed of a number of instances (or tokens) of symbols related in some physical way (such as a token being next to another)" (Newell & Simon, 1976). This physical nature of symbols gives them a concrete existence in the world, but this does not remove from them the quasi-transcendental character of this existence. In effect, symbols exist in a *wider* world than that of symbols and symbol structures. Without this *widerness*, symbols might well exist as solely physical patterns, but their function, that which defines them, would not exist. This function is one of providing *distal access*, taken generally, to this *widerness* that is in fact an

otherness. Put informally, a symbol is an entity that stands for–better: takes the place of–some entity other than itself. This "standing for," or "taking the place of," is formally spoken of as *designation* and *interpretation*.

With respect to the former, an expression is said to *designate* "an object if, given the expression, the system can either affect the object itself or behave in ways dependent on the object" (Newell & Simon, 1976). More formally, "an entity X designates an entity Y relative to a process P, if, when P takes X as input, its behavior depends on Y" (Newell, 1980). For instance, the expression "the cat (X_1) is on the mat (X_2)" affects the object *cat* (Y_1) by placing it on yet another object, the *mat* (Y_2), which, in turn, makes the holder of the expression (as a belief) not step on the mat (Y_2), so as not to hurt (P) the cat (Y_1).[18] This shows that the expression "the cat is on the mat" provides *access* to the object(s), and this is the very essence of designation: Access to objects *in the real world* is obtained via symbolic expressions in what can be seen as a phenomenon of *action at a distance* (Newell, 1980). The relation between *access* and *designation* in a SS can be summed up as follows:

> The great magic comes because this limited capability for accessing supports a general capability for designation. The set of processes must be expanded to include programs, and their inputs must be taken to include expressions. Then, for any entity (whether in the external world or in the memory), if an expression can be created at time T that is dependent on the entity in some way, processes can exist in the symbol system that, at some later time T', take that expression as input and, behaving according to the recorded structure, behave in a way dependent on the entity. Hence these expressions designate the entity. (Newell, 1980)

Designation accounts for universality in the following way: A machine can only simulate another arbitrary machine if it has (the) symbols that designate *it*.[19] Importantly, designation attains here its highest reach, to wit, to allow a machine to behave as something other than what it is–and this is the ultimate reach of symbols.

As for interpretation, we say that a SS can *interpret* an expression if (a) the expression designates a process, and (b) the system can carry out the process. More strictly taken, in the sense it is used in computer science, interpretation is the "act of accepting as input an expression that designates a process and then performing that process" (Newell, 1980). Going on with the real-world illustration above, a human having as a goal "to walk to the other side of the mat without hurting the cat" (the process) will avoid stepping on the cat by walking over it (the carrying-out the process). This, interpretation, does not require a homunculus in the system that links the symbolic input to the performance of a process: It is the symbolic input per se that, if accepted (for instance, it is a legal expression), triggers the performance of the process.

A few additional features can be attributed to symbols and symbol expressions in a SS: Symbols are arbitrary in the sense that it is not prescribed a priori what expressions they can designate; for every process the system is capable of carrying out

[18] This time, I am loosely adapting an illustration to be found in Newell (1990). Actually, "the cat is on the mat" is an illustration highly resorted to in cognitive science, and especially so when symbols and symbol systems are discussed. See, e.g., Harnad (1992).

[19] See the brief discussion of the UTM in Section 2.1 above.

there is a corresponding expression in the system; there are processes for creating any expressions and for modifying any expression in arbitrary ways; expressions are stable in the sense that, once created, they will continue to exist until they are explicitly modified or deleted; and there is no upper bound on the number of expressions a SS can hold.

The above features guarantee that a SS is capable of general intelligence, or, to put it more technically, a SS is a *general-purpose* system. For this generality of purpose, or of intelligence, to be realized there are two requirements: Firstly, the system must be able to *represent the symbols and the symbol structures*; secondly, there must be a *physical realization* for this ability. I next address the former and then discuss the latter subjects as conceived by A. Newell and H. A. Simon.

3.3 Intelligence and Representation

One of the most specific characteristics of intelligent behavior is *heuristic search*, it being meant by this the generation and progressive modification of symbol structures until a solution to a problem has been produced in the form of a structure. This led our authors to formulate a law of qualitative structure that sums up the discussion above (Allen & Newell, 1976):

> *Heuristic Search Hypothesis:* The solutions to problems are represented as symbol structures.

In other words, in face of the task of stating a problem a SS designates, firstly, a *test* for a class of symbol structures, and then a *generator* of symbol structures that are potential solutions, so that to solve a problem is to generate a structure, by using this generator, that satisfies the test. This, however, first and foremost means that SSs are resource-limited systems: Indeed, having only a finite number of steps that they can take in order to find the solution to a problem and a finite interval of time to do so, there is only a finite number of processes that they can carry out.[20] Given this, faced with a problem space, i.e. a space of symbol structures in which problem situations (including the initial and goal situations) can be represented, a SS needs to be able not only to represent it, but also to possess generators appropriate to the diverse situations such that they can generate a space of symbol structures that has some degree of order and pattern (*Condition 1*). This pattern, in turn, must be detectable (*Condition 2*) and the generator of potential solutions must be able, depending on this pattern, to behave in a differential way (*Condition 3*). In terms of information processing, this simply means that the problem space must contain information, and the SS must be capable of using it. The example Newell & Simon (1976) give is extremely simple, yet it illustrates perfectly what is meant by the wording above: Given the problem "$ax + b = cx + d$", a SS must be able to generate any expression e that satisfies the structure "$ae + b = ce + d$".

[20] This resource-limitation feature is obviously to be attributed only to *physical* SSs, a topic I discuss below. As a purely abstract device, a Turing machine, just like any other abstract machine or automaton, is not constrained by any resource limitations, even if it is the basis for the classical theories of computational complexity of physically (non-)realizable computations. For a foundational approach to this subject, see, for example, Blum (1967), where the famous *Blum axioms* were originally formulated.

Thus, and in light of the resource-limited nature of SSs, intelligence appears now more clearly defined as the need to make wise choices of what to do next. And this has actually got a form: The form of a search tree. This is a fundamental remark, as anyone working in computing knows that search trees can be associated to an exponential explosion of search. Intelligence is thus the ability to avoid this, i.e. the ability to "generate only structures that show promise of being solutions or of being along the path toward solutions" (Newell & Simon, 1976), and thus decrease the rate of branching of the trees. Another, more controversial way to put this, is to say that intelligence is the ability to represent solely the space of potential solutions.

The term "representation" has a bad reputation in cognitive science, to a large extent due to the many inconclusive philosophical debates it has motivated since at least Aristotle (see Augusto, 2006). This historically bad reputation is mostly associated to the expression "mental representation" (Augusto, 2005), but "symbolic representation" is also not altogether innocent, though it is clearly not as old as the former. As it is, the ill repute of the latter extends to the field of AI, in which it is associated with inefficient implementations, from a practical viewpoint (e.g., Brooks, 1991), as well as, now from a more theoretical perspective, with the so-called *(symbol) grounding problem*, or the problem of how the symbol-based interpretation of a SS can "be made intrinsic to the system, rather than just parasitic on the meanings in our heads" (Harnad, 1990). These constitute two of the major challenges to the alliance "symbolic representation" in AI, and the fact that they were thus clearly posed in the early 1990s only means that they had been "cooking" for quite a while before appearing in press in the guise of Harnad (1990) and Brooks (1991).

If the reader goes back to the Introduction, they will see that I began by emphasizing R. Brooks conception of what "true intelligence" is. If the reader is literate in the history of AI, namely of AI robotics, they will know that he was one of the most vociferous antagonists of SSs, claiming that "explicit representations and models of the world simply get in the way" when one is trying to mimic or simulate very simple levels of intelligence, and hypothesizing that "[r]epresentation is the wrong unit of abstraction in building the bulkiest parts of intelligent systems" (Brooks, 1991). Summarily, and in his own words, it is "better to use the world as its own model" (Brooks, 1991), a statement that stands for an advocacy of what is now called *the reactive paradigm* in AI robotics (e.g., Murphy, 2000). Greatly based on the behaviorist theory of stimulus-response, this *SENSE-ACT* paradigm emerged in the early 1990s as a reaction against the *hierarchical paradigm*, which had an intermediary primitive component, to wit, *PLAN*, that required an explicit symbol-based representation known as a *world model*, essentially a knowledge base whose facts were expressed in a first-order predicate logical language.

A. Newell and H. A. Simon were not working explicitly in designing robots, but theirs and Brooks was one and the same problem: How to replicate general intelligence in a machine or device. The difference between theirs and Brooks' work lay precisely on the conception of intelligence. The *SENSE-ACT* pairings, or the aspects of perception and motor skills that were all-too-often abstracted away from the general endeavor of AI, according to Brooks (1991), constituted in fact the *hard problems* solved by intelligent systems; in particular, Brooks did not consider "human intelligence" as the focus of AI.[21] Newell and Simon, in contrary, not only saw AI as a

[21] Brooks was very clear with respect to this subject: "I, and others, believe that human level

means of informing cognitive science with respect to human psychology (see Section 2.3), but clearly stated that *solely* processes that resemble those by humans in the solution of problems, and in particular problems that humans find difficult, should be contemplated in the large project of explaining human intelligent behavior (Simon & Newell, 1971). In effect, the ultimate aim of this project was that of designing artificial SSs that could approximate human behavior, namely insofar as problem solving and reasoning were concerned. And for them this entailed (mental) representations.

In Section 2.4, it was settled that both human brains and some computing devices have the ability to acquire, store, and manipulate symbols and symbol structures, i.e. they share one and the same language (class) in the sense that representations in both media can be informationally equivalent, albeit computationally distinct. I now address the very nature of representations and their central role in the definition of human intelligence. I resort again to the human shopping for food in a supermarket. Let us suppose that this agent lost their shopping list and is now trying to recollect what needed to be bought. "I need a can of tomatoes" they think while in the stationery section, so that the can of tomatoes–the *distal stimulus*–is not available for direct perception. This thought contains in fact two representations, to wit, (N) the agent needs something, and (T) this something is a can of tomatoes, but I shall firstly concentrate only on the latter.

Representation T is conveniently expressed in the form of a proposition, a string of symbol tokens that are *combined* in such a way–i.e. arranged–that it means something for the agent. More precisely, the symbol tokens "I", "need", "a", "can", "of", and "tomatoes" constitute the expression, or symbol structure,

(T) I need a can of tomatoes.

Note that the six symbol tokens that compose T could have been arranged in $6! = 720$ distinct ways (!), but only the arrangement or permutation constituting T has meaning for the agent. This combinatorial feat being achieved, which in itself is already an instance of a representation,[22] let us now focus on the particular expression "a can of tomatoes." This contains two concepts, CAN and TOMATO, but I shall isolate the latter, so that "tomato" is the *symbol* in the *symbol structure* "a can of tomatoes." In other words–more precisely, in the terminology of Newell (1990)–, "tomato" is the symbol token that occurs as part of the representation medium that gets processed as a region within the expression or structure "a can of tomatoes."[23] By now, the reader should benefit from taking a look at Figure 5; in this, the structure "a can of tomatoes" should be placed on the left, in the shape contained within "PROCESS", so that "tomato" is the symbol within the structure in consideration. When the agent represents the symbol token "tomato" within this specific structure, they are actually *accessing* a body of *further* symbols and symbol structures associated

intelligence is too complex and little understood to be correctly decomposed into the right subpieces at the moment and that even if we knew the subpieces we still wouldn't know the right interfaces between them. Furthermore, we will never understand how to decompose human level intelligence until we've had a lot of practice with simpler level intelligences." (Brooks, 1991).

[22] "Why is it the case that when we create representations we so often create arrangements that include tokens?" (Newell, 1990).

[23] The medium here is verbal language, in which the symbol tokens are not necessarily letter strings, but can be sound strings. For example, the letter string "tomato" can be equivalently represented as the sound string /təˈmeɪtoʊ/.

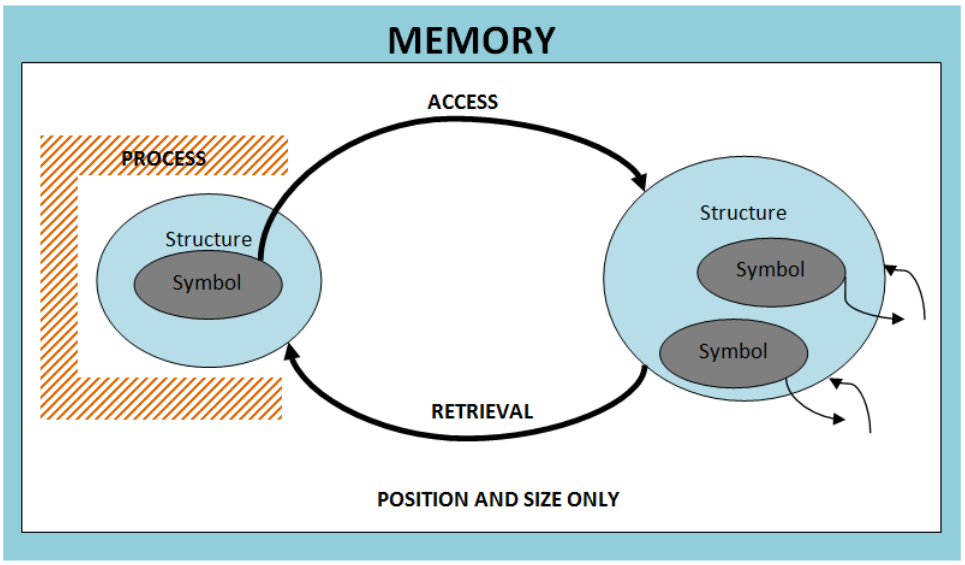

Figure 5: Symbols and distal access. (Adapted from Newell, 1990.)

with "tomato" (e.g., "is a fruit or a vegetable," "is an ingredient in salads," "is a healthy food," "can be processed as ketchup," etc.), from which they *retrieve* the representation that *tomato* is being referred to. This is so because the structure of the token "tomatoes" does not contain all the knowledge associated with it; the needed knowledge "is elsewhere and must be accessed and retrieved" (Newell, 1990).

With respect to representation N, "representations must be at the service of other activities of the organism, and hence must be evocable at appropriate times and places" (Newell, 1990). This entails two things: Firstly, a *memory*, or a structure where symbol structures are stored in a structured way, and *control* over that structure in the sense that it can be accessed at will and searched in a targeted way. This, as seen above, is the constitution of a SS (cf. Fig. 4).

But a SS can be purely abstract–say, a Turing machine. Human agents can in principle compute whatever it is a Turing machine computes, and they do it by using their brains, so that we have to leave the realm of purely abstract systems and enter that of physical ones.

3.4 Physical Symbol Systems

The above Sections 3.1-3 suffice, together with the Turing machine (cf. Section 2.1), for a theory of symbolic representation, namely as this accounts for intelligent behavior. For instance, LT and GPS are, just like the Turing machine, first and foremost programs that can be conceived at a purely abstract level as, say, transition tables or directed graphs in which a given symbol σ_i , $1 \leq i \leq n$, in a symbol structure $\sigma_1, \sigma_2, ..., \sigma_n$ determines a change (of state) in the SS that is solving a problem (compare Fig.s 1-2 with Fig. 4). In order to implement this theory in real machines we need the notion of physical realizability of symbolic representations, i.e. we need a

conception of physical entities that embody SSs. Newell and Simon formulated a hypothesis that may be of import for today's AI challenge of designing wholly autonomous artificial agents.[24] Before I give their hypothesis, a few preliminaries are called for.

Let us call a physically realized/realizable SS a *Physical Symbol System* (PSS); then, it can be argued that humans are PSSs, as they are obviously physical organisms and they exhibit symbolic behavior. As a matter of fact, humans *define* the class of PSSs, in the sense that any physical organism or device that is to be classified as a PSS must exhibit the same symbolic behavior as that of humans. Hence, PSSs obey the laws of physics and are realizable by engineered systems made of engineered components (Newell & Simon, 1976).[25]

As seen above, symbolic behavior in humans equates with intelligent behavior. We can now state a general scientific hypothesis, a law of qualitative structure for SSs, it being the case that by "general intelligent action" the scope of intelligence seen in human action is indicated, and where by this it is meant "that in any real situation behavior appropriate to the ends of the system and adaptive to the demands of the environment can occur within some limits of speed and complexity" (Newell & Simon, 1976):

> *The Physical Symbol System Hypothesis* (PSSH): A PSS has the necessary and sufficient means for general intelligent action.

The PSSH, which is a law of qualitative structure inasmuch as it specifies a general class of systems, and is an empirical hypothesis insofar as this class is believed to account for a set of phenomena found in the real world, states in fact that a PSS is an instance of a universal machine. This, in turn, tells us that intelligence is realized by a universal computer. But the PSSH goes beyond Turing's thesis, according to which any computable function is computable by a UTM: The PSSH asserts *explicitly* (while the Turing Thesis does so implicitly) that this UTM is a symbol system, thus making an architectural assertion about the very nature of intelligent systems.

In these architectural terms, two stages of development can be seen: Firstly, the use of formal logic showed that a machine could be run from a symbolic description. Let us call this the first stage of the realization of the *principle of interpretation*. Secondly, the stored program concept embodied the second stage of this realization, in the sense that it showed that a system's own data can be interpreted. However, the stored program concept realized in the digital computer does not yet contain the notion of the physical relation that underlies meaning, namely via designation. According to Newell & Simon (1976), this was attained by means of the discovery of *list processing*:

> The contents of the data structures were now [in 1956] symbols, in the sense of our Physical Symbol System: patterns that designated, that had referents. Lists held addresses which permitted access to other lists–thus

[24] According to Newell (1980), this hypothesis was the most fundamental contribution to cognitive science as this is conceived as the joint enterprise of AI and computer science. It is important to remark that this article was originally delivered at the very first meeting of the Cognitive Science Society held in 1979 at UCSD.

[25] Humans, like any other biological systems, can be seen as systems engineered by evolution, i.e. constituted by components that were evolutionarily engineered with a view to survival of the species.

> the notion of list structures. ... List processing produced a model of designation, thus defining symbol manipulation in the sense in which we use this concept in computer science today.

It is thus that in the mid 1950s the concept of the *designating symbol* and *symbol manipulation* emerged, being soon later on completely abstracted into a new formal system of symbolic expressions equivalent to the UTM–as well as to all other equivalent universal schemas of computation–with the creation of languages such as IPL and, greatly influenced by this, LISP (McCarthy, 1960). This final development made it clear what in the PSSH respects sufficiency and what in it respects necessity: With respect to the former, the PSSH postulates that a PSS is sufficient for producing intelligence; in other words, any PSS can be organized further in such a way that it exhibits general intelligent action. As for the latter, whenever general intelligence is exhibited in a physical system it is necessary that this system be a PSS, i.e. there will be symbols and expressions to be found in it, as well as the processes that can be identified as being of general intelligent action.

But the PSSH has a reach farther beyond this: Because the symbols in any PSS are just the same kind of symbols, the PSSH states implicitly that (the phenomenon of) *mind* is a feature of the physical universe. This, in turn, "sets the terms on which we search for a *scientific* theory of mind" (Newell, 1980; my italics). This can be further clarified:

> What we all seek are the further specifications of physical symbol systems that constitute the human mind or that constitute systems of powerful and efficient intelligence. The physical symbol system is to our enterprise what the theory of evolution is to all biology, the cell doctrine to cellular biology, the notion of germs to the scientific concept of disease, the notion of tectonic plates to structural geology. (Newell, 1980)

Importantly, this does not equate with a defense of the *computer metaphor*, or the stance in the philosophy of mind known as *computationalism*; in the case of the joint work of H. A. Simon and A. Newell, this is rather a *theory of information processing* that approaches all PSSs on the same footing. A theory of mind elaborated in this framework actually imposes several constraints on mind, thirteen, to be precise, as listed in Newell (1980). Any phenomenon that is to be classified as *mental* must:

1. Behave as an (almost) arbitrary function of the environment (universality).
2. Operate in real time.
3. Exhibit rational, i.e., effective adaptive behavior.
4. Use vast amounts of knowledge about the environment.
5. Behave robustly in the face of error, the unexpected, and the unknown.
6. Use symbols (and abstractions).
7. Use (natural) language.
8. Exhibit self-awareness and a sense of self.
9. Learn from its environment.
10. Acquire its capabilities through development.
11. Arise through evolution.

12. Be realizable within the brain as a physical system.
13. Be realizable as a physical system.

It is not evident how mental phenomena might be satisfied by such disparate constraints as, for instance, the use of natural language and learning from the environment, so that the best strategy is that of having different scientists tackling the distinct constraints on the list. But the generator for the class of SSs that may satisfy these constraints must first and foremost have two features, to wit, *universality* (with respect to the class) and *symbolic behavior*. In effect, the definition here is that *SSs* are the same as *universal machines* (Newell, 1980).[26]

As seen above, the PSSH addresses a few of these constraints in combination (6, 12, and 13). Yet another constraint that can be addressed from within this framework is 4, which I next discuss.[27]

4 Knowledge-Level Systems

In Section 3.3, it was seen that search was considered to be the distinguishing feature of intelligence. However, there is not only one search, but two *separate searches*, as observed in Newell (1990): There is (i) the *problem search*, and there is (ii) the *knowledge search*, which actually guides or controls the former. In fact, if there is enough knowledge available for an agent, this need not search at all, moving through the problem space directly to the solution. All that it takes in this case is *recognition*, i.e. the ability to, given information and the right access thereto, propose actions that are appropriate to the features the system is faced with. These actions should be appropriate to the *patterns of facts*–to be found in very large numbers if the intelligence scope is large–that can be linked to the problem space. This entails that SSs, and in particular PSSs,

> are collections of patterns and processes, the latter being capable of producing, destroying, and modifying the former. The most important properties of patterns is that they can designate objects, processes, or other patterns, and that, when they designate processes, they can be interpreted. Interpreting means carrying out the designated processes. The two most significant classes of symbol systems with which we are acquainted are human beings and computers. (Newell & Simon, 1976)

When carrying out the designated processes results in rational action, i.e. action that is appropriate to the patterns of facts, then we speak of knowledge. This is a much debated notion, namely in epistemology, but what is needed in cognitive science is a concept of knowledge that allows us both to describe and predict the response functions of an SS. When we have found this concept, then we have a *knowledge*

[26] Universality is thus a *relative* notion here. For instance, the UTM is a universal machine firstly with respect to the class of Turing machines (see Section 2.1 above). But the interesting discovery in computability theory was that several classes of (abstract) machines known as *effective computable procedures* are actually equivalent with respect to the universe of the computable functions. This equivalence entails that, secondly, the UTM is universal with respect to all these machines. See Newell (1980) for a detailed description of a UTM simulating a specific Turing machine.

[27] Much as I would like to, I cannot analyze these 13 constraints in detail here; I refer the reader to Anderson & Lebiere (2003) for such an analysis.

system. In other words, SSs *realize* knowledge systems (Newell, 1990). This leads us now directly to the next topic of this survey, that of knowledge systems and the knowledge level as they were conceived by A. Newell and H. A. Simon.[28]

4.1 Representing Knowledge

Any computing machine, or PSS, can be described at different hierarchically organized levels in a bottom-up way. Because, as seen, these include also humans, I skip the most basic levels of description–a neural architecture, in the case of humans–and address directly two levels higher in the hierarchy that are immediately connected: The *symbol level* and, directly above it, the *knowledge level*. The symbol level in a computing device can summarily be seen to be constituted by programs, which require a language, which in turn requires symbols. The relevant feature of programs here is not that they are sequential symbolic instructions–which they are–but that they represent in a farther-reaching way than the one discussed above, namely in Section 3.3. The terms in which this reach is referred to, to wit, *knowledge* and *access*, have already been used above in the discussion on symbols and SSs, but now the point is to discuss their import immediately above the symbol level. At this level–the knowledge level–representation is characterized as follows:

> It is a competence-like notion whose nature can be indicated by the slogan formula:
>
> $$\text{Representation} = \text{Knowledge} + \text{Access}$$
>
> Given a representation, making use of it requires processing to produce other symbolic expressions (or behavior). Although it is possible for a symbolic structure to yield only a small finite number of new expressions, in general there can be an unbounded number. Consider what can be obtained from a chess position, or from the axioms of group theory, or from a visual scene. Further, to obtain most of these new expressions requires varying amounts of processing. Thus, it is theoretically useful to separate analytically the set of potential expressions that a representation can yield from the process of extracting them, i.e., the access to them. Knowledge is this abstract set of all possible derived expressions. (Newell, 1980)

The equation above gives us a definition of knowledge as representation without access, and this is what is meant by the "abstract set of all possible derived expressions" in this passage. By "abstract," it is simply meant that the expressions are not connected to the real world in the sense that they do not represent, or what is the same, do not designate objects in the world. In effect, I retake the concept *designation*, which

[28] For this discussion, I draw essentially on Newell (1980, 1982, 1990), which, I believe, reflect much of his collaborative work with H. A. Simon. Newell (1980) was a single-authored article only for the reason that H. A. Simon was presenting his own paper at the same event (the first meeting of the Cognitive Science Society held in 1979) and Newell (1982) was so because it was A. Newell's presidential address of the American Association for Artificial Intelligence (AAAI)–the first address, for that matter. As usually in single-authored papers by any of these two authors, the other author of this duo is profusely mentioned in these articles.

was seen above as definitional of the notion of symbol, now from the viewpoint of its relation to knowledge. To begin with, representation can be taken as just another way to refer to a structure that designates, a symbolic structure in the case at hand:

> *X represents Y* if *X* designates aspects of *Y*, i.e., if there exist symbol processes that can take *X* as input and behave as if they had access to some aspects of *Y*. (Newell, 1980)

With this, the conclusion can be drawn that representation is tightly connected to symbol systems. How is this conclusion connected to the equation above, one might now ask. Simply, it informs us that knowledge and access thereto are what actually constitutes a representation, in the sense elaborated on above; this is crucial to distinguishing knowledge from its particular uses. Indeed, if we manipulate the equation above, we end up with

$$\text{Representation} - \text{Access} = \text{Knowledge}$$

which gives us a "definition" of knowledge as that which a representation has.

It can in effect be considered that the *essence* of representation is the ability "to go from something to something else by a different path when the originals are not available" (Newell, 1990), which can have a formal formulation as

$$(K^*) \qquad decode\left[encode\left(T\right)\left(encode\left(X\right)\right)\right] = T\left(X\right)$$

where X denotes the original external situation and T does so for the external transformation. K^* is actually a general law, *the representation law* that explains why the internal situation represents the external situation, i.e. why $T\left(X\right)$ is knowledge of X. But, as there are many ways to encode in any situation, $T\left(X\right)$ must be seen from a combinatorial perspective; this, in turn, requires control (see Fig. 4), otherwise the system can face a combinatorial explosion. It can then be considered that a system can represent if it can *compose* functions and if it can do it internally under its own control (Newell, 1990). Because the systems that provide composability of transformations are the *computational systems*, knowledge must then be approached from this viewpoint. But they must be approached from a level higher than that of symbols, namely as a *specialization* of this level in the sense that this only *approximates* the level of interest, to wit, the knowledge level.

4.2 The Knowledge Level

It is customary today to represent schematically a knowledge system as a set of modules comprising at least a knowledge base and an inference engine interconnected by bidirectional arrows (e.g., Poole et al., 1998). More complex representations, comprising the environment, the goals, and the actions to be carried out in the environment are also available (e.g., Augusto, 2020b). However, at the knowledge level a knowledge system, designated as *knowledge-level system* (KLS), can–and retaking the discussion immediately above–be summarily represented as in Figure 6.

As it can be seen from Newell's (1990) depiction of a KLS, this is constituted by three main *computational* components: The *representations* about the environment and the associated goals, the *composability* of the responses, and the *instructions*

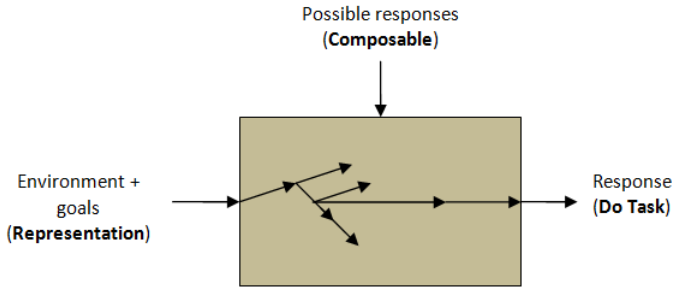

Figure 6: A knowledge-level system. (Adapted from Newell, 1990.)

with respect to the tasks to be performed as responses. The juice to be extracted from Figure 6 is that symbols provide the computational means to access the distal stimuli, but they are not the distal stimuli: They are but *surrogates* of these. But if the response is to be rational–as it is expected from a KLS–, then they provide to the system exactly the properties of the stimuli that are relevant for the attainment of the goals.[29] Importantly, as representation is in fact the input for the composition of functions, it can be concluded that a KLS is a *representation system*. In other words, if we abbreviate the class of the latter systems as RS, we have the inclusion relation

$$\mathcal{KLS} \subseteq \mathcal{RS}.$$

Recall the illustration above, in Section 2, of a chess-playing machine M: M can be said to have knowledge if it has the set of all the derivable symbolic expressions representing both all the possible positions on the chess board and the moves M can make to win the game. By "all the possible positions on the chess board" we mean now the environment, and by "the moves to win the game" we mean the actions (moves) to be taken towards the goal (winning the game). If we know *how* M represents both the environment and the possible actions to take towards the goal, then we know about M's symbol level. If we know *what* M will do (i.e. if we can predict M's behavior) in light of these representations–or even wholly abstracting from them–, then we know about M's knowledge level. This is largely an independent level, even if it can be reduced to the symbol level (see Fig. 7).

KNOWLEDGE LEVEL	SYMBOL LEVEL
Agent	Total SS
Actions	SSs with transducers
Knowledge	Symbol structure + its processes
Goals	(Knowledge of goals)
Principle of rationality	Total problem-solving process

Figure 7: Reduction of the knowledge level to the symbol level. (Adapted from Newell, 1982.)

[29] One way to put it is to speak of *analogical* representations (Newell, 1990).

I elaborate on this in due detail. Already in Simon & Newell (1971), knowledge had been identified as the crucial element in the approximation of a program to human intelligence:

> Each node in a problem space may be thought of as a possible state of knowledge to which the problem solver may attain. A state of knowledge is simply what the problem solver knows about the problem at a particular moment of time–knows in the sense that the information is available to him and can be retrieved in a fraction of a second. ... The problem solver's search for a solution is an odyssey through the problem space, from one knowledge state to another, until his current knowledge state includes the problem solution–that is, until he knows the answer.

Above, the knowledge level was defined as the level at which one can predict the behavior of an agent if one knows (1) what the agent knows about the environment and (2) what this agent's goals–i.e. what the environment should be–are. However, in order to attribute knowledge to this agent a third factor is required, to wit, that (3) this agent be governed by the *principle of rationality*. Indeed, this agent, despite (1) and (2), may not be interested in actually winning the game, in which case all their knowledge will be useless. Inversely, by considering these aspects, we can design a KS to behave in the desired way. This specification can be made before any other specifications of the system at the inferior levels. This actually calls for a formal formulation (in Newell, 1982):

> *The Knowledge-Level Hypothesis:* There exists a distinct computer systems level, lying immediately above the symbol level, which is characterized by knowledge as the medium and the principle of rationality as the law of behavior.

This hypothesis allows for the formulation of a few features concerning the *nature* of knowledge at this level, most of which I discussed–or just touched upon–above:

> (A) Knowledge is intimately linked with rationality. In fact, a system of which it can be said that behaves rationally is a KS.
>
> (B) Knowledge is a competence-like notion, a potential for generating action.
>
> (C) The knowledge level is an approximation, imperfect in both degree and scope.
>
> (D) The body of knowledge at the knowledge level is realized by data structures and processes, systems whose function is to represent.
>
> (E) Knowledge is the specification of what a symbol structure should be able to do.
>
> (F) Logics are one class of representations among many, though they are uniquely fitted to the analysis of knowledge and representation.

From this list of well-formulated items about knowledge and representation, one of which I did not have the opportunity to discuss here as it would take me a long

way,[30] one should actually extract what Brooks (1991) called "the essence of being and reacting" and which he wanted to divest SSs of, i.e. the awareness that a KLS system "will do whatever is within its power to attain its goals, in so far as its knowledge indicates" (Newell, 1982). And the illustrations given by Newell (1982) are, I trust, clear from this viewpoint of being and reacting in the case of human agents:

> (α) "She knows where this restaurant is and said she'd meet me here. I don't know why she hasn't arrived."
> (β) "Sure, he'll fix it. He knows about cars."
> (γ) "If you know that $2 + 2 = 4$, why did you write 5?"

5 Conclusions

The discussion above on the joint contribution of H. A. Simon and A. Newell to symbolic AI was focused on the time window 1956-1982 (even though later works were included). In the early 1980s, the industrialization of knowledge that characterizes today's so-called knowledge technologies (e.g., Augusto, 2020b-c) was yet wholly unknown, and many of the knowledge subfields and communities that today thrive on notions such as *representation* had not even emerged: For instance, ontologies were first conceived as a means of representing and sharing knowledge in the early 1990s (e.g., Gruber, 1993; Guarino, 1995), and the Semantic Web, which now aims at recruiting all these fields, emerged itself as an idea right after the turn of the millennium (Berners-Lee et al., 2001). The impact of Newell and Simon's work can be seen in particular in Davis et al. (1993), a highly cited paper on knowledge representation. In this paper, Davis and colleagues answer the question of what a knowledge representation (KR) is by discussing five roles thereof:

> (1) A KR is a surrogate.
> (2) A KR is a set of ontological commitments.
> (3) A KR is a fragmentary theory of intelligent reasoning expressed generally in inferencing.
> (4) A KR is a medium for pragmatically efficient computation.
> (5) A KR is a language.

Although their names are only fleetingly mentioned in Davis et al. (1993) we can see that A. Newell and H. A. Simon contributed to the five roles above that today can

[30] I mean item (F). A few words, however, are called for. By the early 1980s, logic had a bad reputation in AI, namely because "Uniform proof techniques have been proven grossly inadequate; the failure of resolution theorem proving implicates logic generally; logic is permeated with a static view, and logic does not permit control. Any doubts about the reality of this residual reaction can be stilled by reading Pat Hayes' attempt to counteract it in his ... (Hayes, 1977)" (Newell, 1982). However, A. Newell saw logic as *the* language of SSs: "Existing work, mostly stemming from the analysis of formal logic, confirms that the class of systems described here (i.e., universal symbol systems) is also the class of systems with general powers of representation or (equivalently) description. The representational range of all first order predicate calculi is the same and corresponds to universal systems (when one asks what functions can be described in the logic)" (Newell, 1980). In Newell (1982), too, logic is clearly indicated as the language for the representation of knowledge, and the well-known problem of logical omniscience is dismissed with the justification that this is the reason why the symbol level is merely an approximation, i.e. logical closure of a set of logical expressions is deemed too high a requirement. Contrast this stance with, for example, Minsky (1974).

be said to define a KR. Much as I would like to discuss at length each of the above roles with relation to Simon and Newell's contribution to them, this article is already too long (for the preferences of both reviewers and readers) and I trust the reader is capable of doing this by themselves. To inspire them for this task, I leave the words of Newell (1990):

> The entire field of artificial intelligence is, in a sense, devoted to discovering the symbol-level mechanisms that permit a close approximation to the knowledge level.

Today, we see that most of the thirteen features listed by Newell in 1980 for PSSs (cf. Secction 3.3 above), as are or adapted, are considered essential *desiderata* for the field of AI robotics. Features 1-3 and 5 are important for all the paradigms in this field. Features 9 and 10, and particularly 7 and 8 together with 13, are more recently seen as desiderata for (humanoid) robots (e.g., Chella et al., 2019). Desiderata 4 and 6, together, clearly segregate the so-called hierarchical paradigm of AI robotics from the reactive one: Whereas in the former the robot relies on a world model expressed in a logical language (typically, first-order predicate logic), in the latter it relies on couplings of perceptual and motor schemas that simulate innate releasing mechanisms in a large diversity of animals, from insects to higher vertebrates. For instance, a frog reacts to the stimulus of a fly in the air (the perceptual schema) by trying to capture it without any planning or the help of a world model (the motor schema).[31] It is only interesting to remark that from a chronological viewpoint the work of Newell and Simon falls on the hierarchical paradigm period (see Murphy, 2000). Is this a re-emerging paradigm? I finish this review with, again, the words of M. Veloso, an AI scientist working in robotics who was directly acquainted with H. A. Simon and A. Newell:

> In the coming years, everything that is going on inside a robot (probabilities, decisions, etc.) will have to be translated into language. (Veloso, 2016)

This meets directly item (5) above, which is arguably the most crucial feature of a knowledge representation. In particular, if a robot can make statements like (α) and (β), and answer questions like (γ) about itself or other robots, or even humans, this seems to me to be a criterion of intelligence–possibly mind–higher than that set by the Turing Test. It is thus perhaps high time to start talking of the *Newell & Simon Test.*[32]

Acknowledgments

I wish to thank the reviewers John E. Laird and Jörg Siekmann for engaging in constructive discussion with me, giving suggestions and making corrections that greatly improved the final manuscript. The peer-review process was a double-blind one.

[31] See Arbib's (1987) *rana computatrix* for a good illustration of this behavior implemented in a robot model.

[32] Anderson & Lebiere (2003) speak of the *Newell Test* with respect to the 13 criteria listed above in Section 3.3. Because it is hardly possible to separate A. Newell's from H. A. Simon's work in symbols and knowledge systems I rename it here as Newell & Simon Test.

References

Anderson, J. R. & Lebiere, C. (2003). The Newell Test for a theory of cognition. *Behavioral and Brain Sciences*, *26*, 587-640.

Arbib, M. A. (1987). Levels of modeling of mechanisms of visually guided behavior. *Behavioral and Brain Sciences*, *10*, 407-465.

Augusto, L. M. (2005). *Who's afraid of idealism? Epistemological idealism from the Kantian and Nietzschean viewpoints*. Lanham, etc.: University Press of America.

Augusto, L. M. (2006). A little idealism is idealism enough: A study on idealism in Aristotle's epistemology. *Idealistic Studies*, *36*, 61-73.

Augusto, L. M. (2014). Unconscious representations 2: Towards an integrated cognitive architecture. *Axiomathes*, *24*, 19-43.

Augusto, L. M. (2019). *Formal logic: Classical problems and proofs*. College Publications: London.

Augusto, L. M. (2020a). *Languages, machines, and classical computation*. 2nd ed. London: College Publications.

Augusto, L. M. (2020b). Toward a general theory of knowledge. *Journal of Knowledge Structures & Systems, 1*(1), 63-97.

Augusto, L. M. (2020c). Editorial. *Journal of Knowledge Structures & Systems, 1*(1), 1-2.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34-43.

Blum, M. (1967). A machine-independent theory of the complexity of recursive functions. *Journal of the Association for Computing Machinery*, *14*, 322-336.

Brachman, R. J. & Levesque, H. J. (eds.). (1985). *Readings in knowledge representation*. Los Altos, CA: Morgan Kaufmann.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, *47*, 139-149.

Chella, A., Cangelosi, A., Metta, G., & Bringsjord, S. (eds.). (2019). *Consciousness in humanoid robots*. Lausanne: Frontiers Media.

Copeland, B. J. (2004). Computable numbers: A guide. In B. J. Copeland (ed.), *The essential Turing* (pp. 5-57). Oxford: Oxford University Press.

Davis, R., Shrobe, H., & Szvolovits, P. (1993). What is a knowledge representation? *AI Magazine*, *14*(1), 17-33.

Floridi, L. (2005). Consciousness, agents and the knowledge game. *Minds and Machines*, *15*(3-4), 415-444.

Fodor, J. A. (1975). *The language of thought*. Sussex: Harvester Press.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199-220.

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, *43*(5-6), 625-640.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, *42*, 335-346.

Harnad, S. (1992). Connecting object to symbol in modeling cognition. In A. Clark & R. Lutz (eds.), *Connectionism in context* (pp. 75-90). London: Springer.

Haugeland, J. (1985). *Artificial Intelligence: The very idea*. Cambridge, MA: MIT Press.

Hayes, P. (1977). In defence of logic. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Pittsburgh, PA.

Kuhn, T. (1962). *The structure of scientific revolutions*. Chicago: University of Chicago Press.

Laird, J. E. (2012). *The Soar cognitive architecture*. Cambridge, MA & London, UK: MIT Press.

Laird, J. E. & Newell, A. (1983). A universal weak method: Summary of results. *IJCAI'83: Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, *2*, 771-773.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, *33*(1), 1-64.

Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., & Alami, R. (2017). Artificial cognition for social human-robot interaction: An implementation. *Artificial Intelligence*, *247*, 45-69.

McCarthy, J. (1959). Programs with common sense. *Proceedings of the Symposium on Mechanization of Thought Processes*, 77-84.

McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine. *Communications of the ACM, 3*(4), 184-195.

Minsky, M. (1974). A framework for representing knowledge. Report AIM, 306, Artificial Intelligence Laboratory, MIT.

Murphy, R. R. (2000). *An introduction to AI robotics*. Cambridge, MA & London, UK: MIT Press.

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Newell, A. (1980). Physical symbol systems. *Cognitive Science*, *4*, 135-183.

Newell, A. (1982). The knowledge level. *Artificial Intelligence*, *18*(1), 87-127.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA & London, UK: Harvard University Press.

Newell, A. & Simon, H. A. (1956). The Logic Theory Machine: A complex information processing system. The Rand Corporation, P-868.

Newell, A. & Simon, H. A. (1961). GPS, a program that simulates human thought. In H. Billing (ed.), *Lernende Automaten* (pp. 109-124). Munich: R. Oldenbourg.

Newell, A. & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, *19*(3), 113-126.

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review*, *65*, 151-166.

Newell, A., Shaw, J. C., & Simon, H. A. (1959). Report on a General Problem Solving program. *Proceedings of the International Conference on Information Processing.* UNESCO.

Newell, A., Shaw, J. C., & Simon, H. A. (1960). A variety of intelligent learning in a General Problem Solver. In M. C. Yovits & S. Cameron (eds.), *Self-organizing systems* (pp. 153-189). Oxford, etc.: Pergamon Press.

Newell, A., Tonge, F. M., Feigenbaum, E. A., Green Jr., B. F., & Mealy, G. H. (1961). *Information Processing Language-V manual*. Englewood Cliffs, NJ: Prentice-Hall.

Petzold, C. (2008). *The annotated Turing: A guided tour through Alan Turing's historic paper on computability and the Turing machine*. Indianapolis: Wiley.

Poole, D., Mackworth, A., & Goebel, R. (1998). *Computational intelligence: A logical approach*. New York & Oxford: Oxford University Press.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, *3*(3), 417-457.

Simon, H. A. (1969). *The sciences of the artificial*. 1st ed. Cambridge, MA: MIT Press.

Simon, H. A. (1978). On the forms of mental representation. In C. W. Savage (ed.), *Perception and cognition: Issues in the foundations of psychology* (pp. 3-18). Minneapolis: University of Minnesota Press.

Simon, H. A. (1981). *The sciences of the artificial*. 2nd ed. Cambridge, MA: MIT Press.

Simon, H. A. (1996). *The sciences of the artificial*. 3rd ed. Cambridge, MA: MIT Press.

Simon, H. A. (2019). *The sciences of the artificial*. 4th ed. Cambridge, MA: MIT Press.

Simon, H.A. & Newell, A. (1962). Computer simulation of human thinking and problem solving. *Monographs of the Society for Research in Child Development*, *27*(2), 137-150.

Simon, H. A. & Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist*, *26*(2), 145-159.

Smolensky, P. (1987). Connectionist AI, symbolic AI, and the brain. *Artificial Intelligence Review*, *1*, 95-109.

Stich, S. (1992). What is a theory of mental representation? *Mind*, *101*(402), 243-261.

Turing, A. (1936-7). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Series 2*, *41*, 230-265.

Turing, A. M. (1948). Intelligent machinery. In B. J. Copeland (ed.), *The essential Turing* (pp. 410-432). Oxford: Oxford University Press. 2004.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*, 433-460.

Veloso, M. (2016). Interview with ... Manuela Veloso. Interviewer: M. van Zee. *Benelux A.I. Newsletter*, *30*(2), 3-5.

Whitehead, A. N. & Russell, B. (1910). *Principia mathematica*. Vol. 1. Cambridge: Cambridge University Press.

Cite this article as:

Augusto, L. M. (2021). From symbols to knowledge systems: A. Newell and H. A. Simon's contribution to symbolic AI. *Journal of Knowledge Structures & Systems*, *2*(1), 29-62.

EDITORIAL INFORMATION

Editor-in-chief: Luis M. Augusto
Handling editor: Farshad Badie
Peer reviewers:[a]
John E. Laird (*University of Michigan*)
Jörg Siekmann (*Saarland University*)
Submitted: April 14, 2021; **Revised**: June 13, 2021; **Accepted**: June 23, 2021

[a]Double-blind review.

Reviewer Commentary

As a reviewer who reported positively and recommended publication of this article, it is needless to say that in my mind it gives a very good presentation and philosophical reflection upon the period of research in AI in the 70s of the previous century. It traces the development from symbols (in computer science) up to their role in knowledge representation, the "knowledge symbols," by focussing on the contributions of Alan Newell and Herb Simon at a time when this was the dominant research paradigm, nicknamed as "GoFAI, good old-fashioned AI."

*It deserves mentioning however that today another approach based on a simulation of what neuroscientists observe in a brain–called neural computing and deep learning–is more dominant. See for a recent account the Turing lecture "Deep Learning for AI" by the three authors Y. Bengio, Y. Lecun and G. Hinton (*Communications of the ACM*, July 2021, vol 64, No7) who recently received–just like Newell and Simon previously–the Turing Award.*

For instance, the hunting hawk example in the beginning of the article serves certainly well to demonstrate the general idea the author wants to convey, but as a matter of fact it would actually be modelled and computed by a trained neural net. To rescue the symbolic approach many researchers, including myself, might argue that a Neural Net Learning algorithm would be used to do the computational job for the recognition and then giving it a symbolic representation named "hawk." See http://www.neural-symbolic.org/ for the various approaches in the "International Workshop series on Neural-Symbolic Learning and Reasoning" whose goal is to combine neural networks' robust learning mechanisms with symbolic knowledge representation and reasoning capability in ways that retain the strengths of each paradigm. But the three authors mentioned above would not.

While the debate between scientists from either of these two paradigms may be more or less typical for any established scientific area with different schools, the argument about consciousness however is very different. For example Roger Penrose argues in Shadows of the Mind *that consciousness is not computational. This debate is far more basic, fascinating and controversial where philosophy, neurosciences, cognitive science and psychology, Buddhist theories of the mind and quantum physics argue about a new cognitive model.*

Jörg Siekmann

Author Reply

I, too, think that artificial autonomous agents mimicking animal behavior in solving problems such as capturing prey or fleeing predators can–and perhaps must–make use of the classification abilities of neural networks and other machine learning models. All it takes, after all, for a robot to seize some X (an object or an animal) is to classify that thing as X, though the action in itself is substantially dependent on both the motor skills of the robot and how the motor schema is coupled with the perceptual schema, to use jargon that is perhaps already outdated.

In this framework, it will–in time–be a piece of cake to replicate in an artificial autonomous agent the behavior of a flying predator spotting and chasing its prey. The difficult problems will be presented by *motivation*. For instance, a hungry hawk will

not give up when faced with a rabbit that sought escape in a bush; this frustration fuels its reasoning skills, and the hungrier the hawk is, the more it will persevere in its attempt to dislodge the rabbit from its shelter. Hunger, and hence survival, is here the motivation.

It is not impossible to implement motivations such as reward and punishment in neural networks; in effect, it has been done, and, as I discuss in the article "Transitions versus dissociations: A paradigm shift in unconscious cognition" (*Axiomathes*, *28*, 269-291; 2018), further psychological mechanisms associated to motivation, such as repression, might be next to be implemented in neural-based computational models. As I see it, if the ultimate aim of AI is to create robots that behave like humans–i.e. potentially rationally and ethically–, we will need to be able to implement in these machines three emotions that work for us as basic motivations, to wit, *fear*, *guilt*, and *shame*. Fear will be a motivation for self-preservation and preservation of cooperating agents, guilt will prevent uncontrolled agency, and shame will be a motor for self-improvement.

Interestingly enough, these are *the* core concepts of psychoanalysis (see my *Freud, Jung, Lacan: Sobre o inconsciente*, University of Porto, 2013), which is all about how we (fail to) manage to live with each other in specific places and times; in AI terms, the *environment* and the *goals*. This, in the case of humans, both generates and recruits knowledge (see items (α)-(γ) above). It will take more than "+" (for reward) and "−" (for punishment) in neural-based models for an artificial agent to represent these emotions and thus be capable of some form of (self-)consciousness. This will not only take special high-level languages, but also special compilation techniques to "translate" them into the (possibly many and very different) machine languages of the artificial agents that will predictably be out and about around us in a not-so-far future. This is what A. Newell called "the great magic"; in this effort, his joint work with H. A. Simon in symbols and knowledge representation might provide us with an abundance of clues.

Luis M. Augusto