# C

## COGNITIVE SYSTEMS AND COGNITIVE ARCHITECTURES

### INTRODUCTION

Cognitive systems refer to computational models and systems that are in some way inspired by human (or animal) cognition as we understand it, which is a broad class of systems, not always well defined or clearly delineated. There are a variety of forms of cognitive systems. They have been developed for a variety of different purposes and in a variety of different ways. We will describe two broad categories below.

In general, computational cognitive modeling explores the essence of cognition through developing computational models of mechanisms (including representations) and processes of cognition, thereby producing realistic cognitive systems. In this enterprise, a cognitive architecture is a domain-generic and comprehensive computational cognitive model that may be used for a wide range of analysis of behavior. It embodies generic descriptions of cognition in computer algorithms and programs. Its function is to provide a general framework to facilitate more detailed computatonal modeling and understanding of various components and processes of the mind. Cognitive architectures occupy a particularly important place among all kinds of cognitive systems, as they aim to capture all basic structures and processes of the mind, and therefore are essential for broad, multiple-level, multiple-domain analyses of behavior. Developing cognitive architectures has been a difficult task. In this article, the importance of developing cognitive architectures, among other cognitive systems, will be discussed, and examples of cognitive architectures will be given.

Another common approach toward developing cognitive systems is the logic-based approach. From the logical point of view, a cognitive system is first and foremost a system that, through time, adopts and manages certain attitudes toward propositions, and reasons over these propositions, to perform the actions that will secure certain desired ends. The most important propositional attitudes are *believes that* and *knows that.* (Our focus herein will be on the latter. Other propositional attitudes include *wants that* and *hopes that*.) A propositional attitude is simply a relationship holding between an agent (or system) and one or more propositions, where propositions are declarative statements.

We can think of a cognitive system's life as being a cycle of sensing, reasoning, acting; sensing, reasoning, acting; ..., and so on. In a cognitive system, this cycle repeats *ad infinitum*, presumably with goal after goal achieved along the way. In a logic-based cognitive system, the knowledge at the heart of this cycle is represented as formulas in one or more logics, and the reasoning in question is also regimented by these logics.

The eventual objective of cognitive systems research is to construct physically instantiated cognitive systems that can perceive, understand, and interact with their environment, and evolve and learn to achieve human-like performance in complex activities (often requiring context-specific knowledge). The readers may look into Refs. 1–4 for further information.

### COGNITIVE ARCHITECTURES

In this section, we describe cognitive architectures. First, the question of what a cognitive architecture is is answered. Next, the importance of cognitive architectures is addressed. Then an example cognitive architecture is presented.

#### What is a Cognitive Architecture?

As mentioned earlier, a cognitive *architecture* is a comprehensive computational cognitive model, which is aimed to capture the essential structure and process of the mind, and can be used for a broad, multiple-level, multiple-domain analysis of behavior (5,6).

Let us explore this notion of architecture with an analogy. The architecture for a building consists of its overall framework and its overall design, as well as roofs, foundations, walls, windows, floors, and so on. Furniture and appliances can be easily rearranged and/or replaced and therefore they are not part of the architecture. By the same token, a cognitive architecture includes overall structures, essential divisions of modules, essential relations between modules, basic representations and algorithms within modules, and a variety of other aspects (2,7). In general, an architecture includes those aspects of a system that are relatively invariant across time, domains, and individuals. It deals with componential processes of cognition in a structurally and mechanistically well-defined way.

In relation to understanding the human mind (i.e., in relation to cognitive science), a cognitive architecture provides a concrete framework for more detailed computational modeling of cognitive phenomena. Research in computational cognitive modeling explores the essence of cognition and various cognitive functionalities through developing detailed, process-based understanding by specifying corresponding computational models of mechanisms and processes. It embodies descriptions of cognition in concrete computer algorithms and programs. Therefore, it produces runnable computational models of cognitive processes. Detailed simulations are then conducted based on the computational models. In this enterprise, a cognitive architecture may be used for broad, multiple-level, multiple-domain analyses of cognition.

In relation to building intelligent systems, a cognitive architecture specifies the underlying infrastructure for intelligent systems, which includes a variety of capabilities, modules, and subsystems. On that basis, application

systems may be more easily developed. A cognitive architecture also carries with it theories of cognition and understanding of intelligence gained from studying human cognition. Therefore, the development of intelligent systems can be more cognitively grounded, which may be advantageous in many circumstances (1,2).

Existing cognitive architectures include Soar (8), ACT-R (9), CLARION (6), and many others.

For further (generic) information about cognitive architectures, the readers may turn to the following websites:

http://www.cogsci.rpi.edu/~rsun/arch.html

http://books.nap.edu/openbook.php?isbn=0309060966

as well as the following websites for specific individual cognitive architectures (Soar, ACT-R, and CLARION):

http://www.cogsci.rpi.edu/~rsun/clarion.html

http://act-r.psy.cmu.edu/

http://sitemaker.umich.edu/soar/home

### Why are Cognitive Architectures Important?

For cognitive science, the importance of cognitive architectures lies in the fact that they are beneficial to understanding the human mind. In understanding cognitive phenomena, the use of computational simulation on the basis of cognitive architectures forces one to think in terms of process and in terms of detail. Instead of using vague, purely conceptual theories, cognitive architectures force theoreticians to think clearly. They are, therefore, critical tools in the study of the mind. Researchers who use cognitive architectures must specify a cognitive mechanism in sufficient detail to allow the resulting models to be implemented on computers and run as simulations. This approach requires that important elements of the models be spelled out explicitly, thus aiding in developing better, conceptually clearer theories. It is certainly true that more specialized, narrowly scoped models may also serve this purpose, but they are not as generic and as comprehensive and thus they are not as useful (1).

An architecture serves as an initial set of assumptions to be used for further computational modeling of cognition. These assumptions, in reality, may be based on either available scientific data (for example, psychological or biological data), philosophical thoughts and arguments, or ad hoc working hypotheses (including computationally inspired such hypotheses). An architecture is useful and important precisely because it provides a comprehensive initial framework for further modeling in a variety of task domains. Different cognitive architectures, such as Soar, ACT-R, or CLARION, embody different sets of assumptions (see an example later).

Cognitive architectures also provide a deeper level of explanation. Instead of a model specifically designed for a specific task (often in an ad hoc way), using a cognitive architecture forces modelers to think in terms of the mechanisms and processes available within a generic cognitive architecture that are not specifically designed for a particular task, and thereby to generate explanations of the task that is not centered on superficial, high level features of a task (as often happens with specialized, narrowly scoped models), that is, to generate explanations of a deeper kind. To describe a task in terms of available mechanisms and processes of a cognitive architecture is to generate explanations centered on primitives of cognition as envisioned in the cognitive architecture (e.g., ACT-R or CLARION), and therefore such explanations are deeper explanations. Because of the nature of such deeper explanations, this style of theorizing is also more likely to lead to unified explanations for a large variety of data and/or phenomena, because potentially a large variety of tasks, data, and phenomena can be explained on the basis of the same set of primitives provided by the same cognitive architecture. Therefore, using cognitive architectures leads to comprehensive theories of the mind (5,6,9), unlike using more specialized, narrowly scoped models.

Although the importance of being able to reproduce the nuances of empirical data from specific psychological experiments is evident, broad functionality in cognitive architectures is also important (9), as the human mind needs to deal with the full cycle that includes all of the following: transducing signals, processing them, storing them, representing them, manipulating them, and generating motor actions based on them. There is clearly a need to develop generic models of cognition that are capable of a wide range of functionalities to avoid the myopia often resulting from narrowly-scoped research (in psychology in particular).

In all, cognitive architectures are believed to be essential in advancing the understanding of the mind (5,6,9). Therefore, developing cognitive architectures is an important enterprise in cognitive science.

On the other hand, for the fields of artificial intelligence and computational intelligence (AI/CI), the importance of cognitive architectures lies in the fact that they support the central goal of AI/CI—building artificial systems that are as capable as human beings. Cognitive architectures help us to reverse engineer the best existing intelligent system—the human mind. They constitute a solid basis for building intelligent systems, because they are well motivated by, and properly grounded in, existing cognitive research. The use of cognitive architectures in building intelligent systems may also facilitate the interaction between humans and artificially intelligent systems because of the similarity between humans and cognitively based intelligent systems.

It is also worth noting that cognitive architectures are the antithesis of "expert systems": Instead of focusing on capturing performance in narrow domains, they are aimed to provide broad coverage of a wide variety of domains (2). Business and industrial applications of intelligent systems increasingly require broad systems that are capable of a wide range of intelligent behaviors, not just isolated systems of narrow functionalities. For example, one application may require the inclusion of capabilities for raw image processing, pattern recognition, categorization, reasoning, decision making, and natural language communications. It may even require planning, control of robotic devices, and interactions with other systems and devices. Such requirements accentuate the importance of research on broadly scoped cognitive architectures that perform a wide range of

cognitive functionalities across a variety of task domains (as opposed to more specialized systems).

## An Example of a Cognitive Architecture

**An Overview.** As an example, we will describe a cognitive architecture CLARION. It has been described extensively in a series of previous papers, including Refs. 6,10–12. The readers are referred to these publications for further details.

Those who wish to know more about other cognitive architectures in existence (such as ACT-R or Soar) may want to see Refs. 8 and 9.

CLARION is an integrative architecture, consisting of a number of distinct subsystems, with a dual representational structure in each subsystem (i.e., implicit versus explicit representations; more later). Its subsystems include the action-centered subsystem (the ACS), the nonaction-centered subsystem (the NACS), the motivational subsystem (the MS), and the meta-cognitive subsystem (the MCS). The role of the action-centered subsystem is to control actions, regardless of whether the actions are for external physical movements or for internal mental operations. The role of the nonaction-centered subsystem is to maintain general knowledge (either implicit or explicit). The role of the motivational subsystem is to provide underlying motivations for actions in terms of providing impetus and feedback (e.g., indicating whether outcomes are satisfactory). The role of the meta-cognitive subsystem is to monitor, direct, and modify the operations of the action-centered subsystem dynamically as well as the operations of all the other subsystems.

Each of these interacting subsystems consists of two "levels" of representation (i.e., a dual representational structure): Generally, in each subsystem, the top level encodes explicit knowledge and the bottom level encodes implicit knowledge. The distinction of implicit and explicit knowledge has been amply argued for before (6,13–15). The two levels interact, for example, by cooperating in actions, through a combination of the action recommendations from the two levels respectively, as well as by cooperating in learning through a bottom-up and a top-down process (to be discussed below). See Fig. 1.

It has been intended that this cognitive architecture satisfy some basic requirements as follows. It should be able to learn with or without a priori domain-specific knowledge to begin with (unlike most other existing cognitive architectures) (11,13). It also has to learn continuously from ongoing experience in the world (as indicated by Refs. 16 and 17, and others, human learning is often gradual and ongoing). As suggested by Refs. 13 and 14, and others, there are clearly different types of knowledge involved in human learning. Moreover, different types of learning processes are involved in acquiring different types of knowledge (9,11,18). Furthermore, it should include both situated actions/reactions and cognitive deliberations (6). It should be able to handle complex situations that are not amenable to simple rules. Finally, unlike other existing cognitive architectures, it should more fully incorporate emotional and motivational processes as well as meta-cognitive processes. Based on the above considerations, CLARION was developed.

**Some Details.** ***The Action-Centered Subsystem.*** First, let us look into the action-centered subsystem (the ACS) of CLARION. The overall operation of the action-centered subsystem may be described as follows:

1. Observe the current state $x$.
2. Compute in the bottom level the Q-values of $x$ associated with each of all the possible actions $a_i$'s: $Q(x, a_1)$, $Q(x, a_2)$, ....., $Q(x, a_n)$.

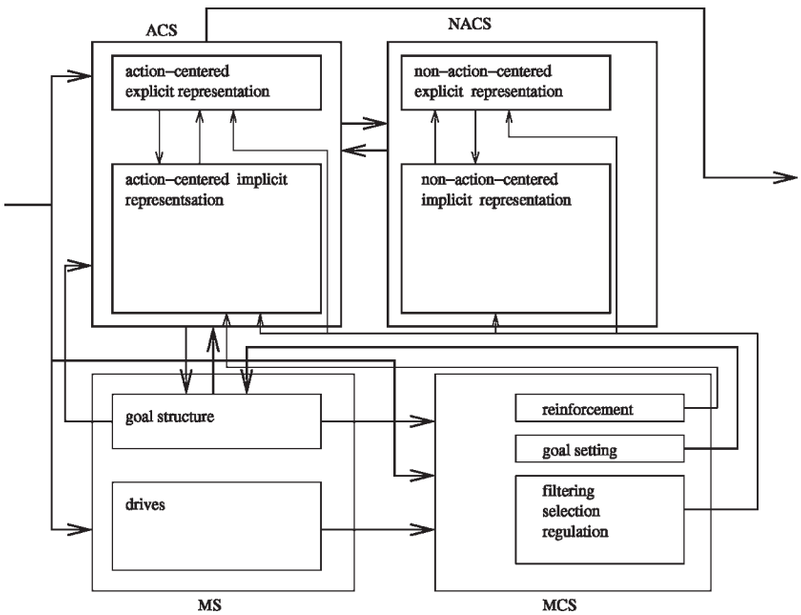


**Figure 1.** The CLARION architecture.

3. Find out all the possible actions ($b_1, b_2, \ldots, b_m$) at the top level, based on the input $x$ (sent up from the bottom level) and the rules in place.
4. Compare or combine the values of the selected $a_i$s with those of $b_j$s (sent down from the top level), and choose an appropriate action $b$.
5. Perform the action $b$, and observe the next state $y$ and (possibly) the reinforcement $r$.
6. Update Q-values at the bottom level in accordance with the *Q-Learning-Backpropagation* algorithm
7. Update the rule network at the top level using the *Rule-Extraction-Refinement* algorithm.
8. Go back to Step 1.

In the bottom level of the action-centered subsystem, implicit reactive routines are learned: A Q-value is an evaluation of the "quality" of an action in a given state: $Q(x, a)$ indicates how desirable action $a$ is in state $x$ (which consists of some sensory input). An action may be chosen in any state based on Q-values in that state. To acquire the Q-values, the *Q-learning* algorithm (19) may be used, which is a reinforcement learning algorithm (see the articles on learning algorithms in this encyclopedia). It basically compares the values of successive actions and adjusts an evaluation function on that basis. It thereby develops reactive sequential behaviors or reactive routines (such as navigating through a body of water or handling daily activities, in a reactive way (6,12). Reinforcement learning is implemented in modular (multiple) neural networks. Due to such networks, CLARION is able to handle very complex situations that are not amenable to simple rules.

In the top level of the action-centered subsystem, explicit symbolic conceptual knowledge is captured in the form of explicit symbolic rules; see Ref. 12 for details. There are many ways in which explicit knowledge may be learned, including independent hypothesis-testing learning and "bottom-up learning" as discussed below.

Humans are generally able to learn implicit knowledge through trial and error, without necessarily using a priori knowledge. On top of that, explicit knowledge can be acquired also from ongoing experience in the world, possibly through the mediation of implicit knowledge (i.e., bottom-up learning; see Refs. 6,18, and 20). The basic process of bottom-up learning (which is generally missing from other existing cognitive architectures and which distinguishes CLARION from others) is as follows: If an action implicitly decided by the bottom level is successful, then the agent extracts an explicit rule that corresponds to the action selected by the bottom level and adds the rule to the top level. Then, in subsequent interaction with the world, the agent verifies the extracted rule by considering the outcome of applying the rule: If the outcome is not successful, then the rule should be made more specific and exclusive of the current case; if the outcome is successful, the agent may try to generalize the rule to make it more universal (21).[1] After explicit rules have been learned, a variety of explicit reasoning methods may be used. Learning explicit conceptual representation at the top level can also be useful in enhancing learning of implicit reactive routines at the bottom level (11).

Although CLARION can learn even when no a priori or externally provided explicit knowledge is available, it can make use of it when such knowledge is available (9,22). To deal with instructed learning, externally provided knowledge, in the forms of explicit conceptual structures such as rules, plans, categories, and so on, can 1) be combined with existent conceptual structures at the top level, and 2) be assimilated into implicit reactive routines at the bottom level. This process is known as top-down learning (12).

***The Non-action-Centered Subsystem.*** The nonaction-centered subsystem (NACS) may be used for representing general knowledge about the world (23), for performing various kinds of memory retrievals and inferences. The nonaction-centered subsystem is under the control of the action-centered subsystem (through its actions).

At the bottom level, "associative memory" networks encode nonaction-centered implicit knowledge. Associations are formed by mapping an input to an output (such as mapping "2 + 3" to "5"). For example, the regular backpropagation learning algorithm can be used to establish such associations between pairs of inputs and outputs (24).

On the other hand, at the top level of the nonaction-centered subsystem, a general knowledge store encodes explicit nonaction-centered knowledge (25). In this network, chunks are specified through dimensional values (features).[2] A node is set up in the top level to represent a chunk. The chunk node connects to its corresponding features (represented as individual nodes) in the bottom level of the nonaction-centered subsystem (25). Additionally, links between chunks encode explicit associations between pairs of chunks, known as associative rules. Explicit associative rules may be formed (i.e., learned) in a variety of ways (12).

Different from most other existing cognitive architectures, during reasoning, in addition to applying associative rules, similarity-based reasoning may be employed in the nonaction-centered subsystem. During reasoning, a known (given or inferred) chunk may be automatically compared with another chunk. If the similarity between them is sufficiently high, then the latter chunk is inferred (12,25).

As in the action-centered subsystem, top-down or bottom-up learning may take place in the nonaction-centered subsystem, either to extract explicit knowledge in the top level from the implicit knowledge in the bottom level or to assimilate explicit knowledge of the top level into implicit knowledge in the bottom level.

***The Motivational and the Meta-Cognitive Subsystem.*** The motivational subsystem (the MS) is concerned with why an

[1] The detail of the bottom-up learning algorithm can be found in Ref. 10.

[2] The basic form of a chunk is as follows: *chunk-id*$_i$: ($dim_{i_1}$, $val_{i_1}$)($dim_{i_2}$, $val_{i_1}$) . . . . . ($dim_{i_n}$, $val_{i_n}$), where *dim* denotes a particular state/output dimension and *val* specifies its corresponding value. For example, *table-1:* (*size*, *large*) (*color*, *white*) (*number-of-legs*, *four*) specifies a large, four-legged, white table.

agent does what it does. Simply saying that an agent chooses actions to maximizes gains, rewards, reinforcements, or payoffs leaves open the question of what determines these things. The relevance of the motivational subsystem to the action-centered subsystem lies primarily in the fact that it provides the context in which the goal and the reinforcement of the action-centered subsystem are set. It thereby influences the working of the action-centered subsystem, and by extension, the working of the nonaction-centered subsystem.

A dual motivational representation is in place in CLARION. The explicit goals (such as "finding food") of an agent (which is tied to the working of the action-centered subsystem) may be generated based on internal drive states (for example, "being hungry"; see Ref. 12 for details).

Beyond low level drives (concerning physiological needs),[3] there are also higher level drives. Some of them are primary, in the sense of being "hard-wired".[4] Although primary drives are built-in and relatively unalterable, there are also "derived" drives, which are secondary, changeable, and acquired mostly in the process of satisfying primary drives.

The meta-cognitive subsystem (the MCS) is closely tied to the motivational subsystem. The meta-cognitive subsystem monitors, controls, and regulates action-centered and nonaction-centered processes for the sake of improving performance (26,27). Control and regulation may be in the forms of setting goals for the action-centered subsystem, setting essential parameters of the action-centered subsystem and the nonaction-centered subsystem, interrupting and changing ongoing processes in the action-centered subsystem and the nonaction-centered subsystem, and so on. Control and regulation can also be carried out through setting reinforcement functions for the action-centered subsystem. All of the above can be done on the basis of drive states in the motivational subsystem. The meta-cognitive subsystem is also made up of two levels: the top level (explicit) and the bottom level (implicit).

**Accounting for Cognitive Data.** Like some other cognitive architectures (ACT-R in particular), CLARION has been successful in accounting for and explaining a variety of psychological data. For example, a number of well-known psychological tasks have been simulated using CLARION that span the spectrum ranging from simple reactive skills to complex cognitive skills. The simulated tasks include serial reaction time tasks, artificial grammar learning tasks, process control tasks, categorical inference tasks, alphabetical arithmetic tasks, and the Tower of Hanoi task (6). Among them, serial reaction time and process control tasks are typical implicit learning tasks (mainly involving implicit reactive routines), whereas Tower of Hanoi and alphabetic arithmetic are high level cognitive skill acquisition tasks (with a significant presence of explicit processes). In addition, extensive work has been done on a complex minefield navigation task, which involves complex sequential decision making (10,11). Work has also been done on an organizational decision task (28), and other social simulation tasks, as well as meta-cognitive tasks. While accounting for various psychological data, CLARION provides explanations that shed new light on cognitive phenomena.

In all of these cases of simulations, the use of the CLARION cognitive architecture forces one to think in terms of process, and in terms of details, as envisaged in CLARION. The use of CLARION also provides a deeper level of explanations. It is deeper because the explanations were centered on lower level mechanisms and processes (1,6). Due to the nature of such deeper explanations, this approach is also likely to lead to unified explanations, unifying a large variety of data and/or phenomena. For example, all the afore-mentioned tasks have been explained computationally in a unified way in CLARION.

[3] Low level drives include, for example, *need for food, need for water, need to avoid danger, need to avoid boredom,* and so on (12).

[4] A few high level drives include: *desire for domination, desire for social approval, desire for following social norms, desire for reciprocation, desire for imitation* (of certain other people), and so on (12).

## LOGIC-BASED COGNITIVE SYSTEMS

We now give an account of logic-based cognitive systems, mentioned in broad strokes earlier.

### Logic-Based Cognitive Systems in General

At any time $t$ during its existence, the cognitive state of a cognitive system $S$ consists in what the system knows at that time, denoted by $\Phi_S^t$. (To ease exposition, we leave aside the distinction between what $S$ knows versus what it merely believes.) We assume that as $S$ moves through time, what it knows at any moment is determined, in general, by two sources: information coming directly from the external environment in which $S$ lives, through the transducers in $S$'s sensors that turn raw sense data into propositional content, and from reasoning carried out by $S$ over its knowledge.

For example, suppose you learn that Alvin loves Bill, and that everyone loves anyone who loves someone. Your goal is to determine whether or not everyone loves Bill, and whether or not Katherine loves Dave. The reasoning needs to be provided in the form of an explicit series of inferences (which serves to guarantee that the reasoning in question is "surveyable").

Your knowledge (or *knowledge base*) now includes that Alvin loves Bill. (It also includes 'Everyone loves anyone who loves someone'.) You know this because information impinging upon your sensors has been transduced into propositional content added to your knowledge base. We can summarize the situation at this point as follows:

$$\Phi_S^{t_{n+1}} = \Phi_S^{t_n} \cup \{\text{Loves}(\text{alvin}, \text{bill})\}$$

Generalizing, we can define a ternary function *env* from timepoint-indexed knowledge bases, and formulas generated by *trans* applied to raw information hitting sensors, to a new, augmented knowledge base at the next timepoint. So we have:

$$\Phi_S^{t_{n+1}} = env(\Phi_S^{t_n}, trans(raw))$$

where $trans(raw) =$ `Loves(alvin,bill).`

Now consider the second source of new knowledge, viz., reasoning. On the basis of reasoning over the proposition that Alvin loves Bill, we know that someone loves Bill, that someone loves someone, that someone whose name starts with 'A' loves Bill, and so on. These additional propositions can be directly deduced from the single one about Alvin and Bill; each of them can be safely added to your knowledge base.

Let $\mathcal{R}[\Phi]$ denote an augmentation of $\Phi$ via some mode of reasoning $\mathcal{R}$. Then your knowledge at the next timepoint, $t_{n+2}$, is given by

$$\Phi_S^{t_{n+2}} = \mathcal{R}[env(\Phi_S^{t_n}, trans(raw))]$$

As time flows on, the environment's updating, followed by reasoning, followed by changes the cognitive system makes to the environment (the system's actions), define the cognitive life of $S$.

But what is $\mathcal{R}$, and what is the structure of propositions returned by *trans*? This point is where logic enters the stage. In a logic-based cognitive system, propositions are represented by formulas in a logic, and a logic provides precise machinery for carrying out reasoning.

### Knowledge Representation in Elementary Logic

In general, when it comes to any logic-based system, three main components are required: one is syntactic, one is semantic, and one is metatheoretical in nature.

The syntactic component includes specification of the alphabet of a given logical system, the grammar for building well-formed formulas (wffs) from this alphabet, and, more importantly, a proof theory that precisely describes how and when one formula can be inferred from a set of formulas. The semantic component includes a precise account of the conditions under which a formula in a given system is true or false. The metatheoretical component includes theorems, conjectures, and hypotheses concerning the syntactic component, the semantic component, and connections between them.

The simplest logics to build logic-based cognitive systems are the propositional calculus and the predicate calculus (or first-order logic, or just FOL).

The alphabet for propositional logic is an infinite list

$$p_1, p_2, \ldots, p_n, p_{n+1}, \ldots$$

of propositional variables and the five familiar truth-functional connectives $\neg$, $\rightarrow$, $\leftrightarrow$, $\wedge$, $\vee$. (The connectives can at least provisionally be read, respectively, as 'not,' 'implies' (or 'if then'), 'if and only if,' 'and,' and 'or.') To say that 'if Alvin loves Bill, then Bill loves Alvin, and so does Katherine,' we could write

$$a_l \rightarrow (b_1 \wedge k_l)$$

where $b_l$ and $k_l$ are the propositional variables.

We move up to first-order logic when we allow the quantifiers $\exists x$ ('there exists at least one thing $x$ such that ...') and $\forall x$ ('for all $x$ ...'); the first is known as the *existential* quantifier, and the second is known as the *universal*. We also allow a supply of variables, constants, relations, and function symbols. Using this representation, the proposition that 'Everyone loves anyone who loves someone' is represented as

$$\forall x \forall y(\exists z Loves(y,z) \rightarrow Loves(x,y))$$

### Deductive Reasoning

The hallmark of deductive reasoning is that if the premises are true, then that which is deduced from them must be true as well. In logic, deduction is formalized in a *proof theory*. Such theories (versions of which were first invented and presented by Aristotle) are often designed not to model the reasoning of logically untrained humans, but rather to express ideal, normatively correct human deductive reasoning targeted by the logically trained. To canvass other proof theories explicitly designed to model the deductive reasoning of logically untrained humans, interested readers may consult Ref. 29.

A number of proof theories are possible (for either of the propositional or predicate calculi). When the goal is to imitate human reasoning and to be understood by humans, the proof theory of choice is *natural deduction* rather than *resolution*. The latter approach to reasoning (whose one and only rule of inference, in the end, is that from $\varphi \vee \psi$ and $\neg \varphi$ one can infer $\psi$), while used by a number of automated theorem provers (e.g., Otter, which, along with resolution, is presented in Ref. 30), is generally impenetrable to humans.

On the other hand, suppositional reasoning is at the heart of natural deduction. For example, one such common suppositional technique is to assume the opposite of what one wishes to establish, to show that from this assumption some contradiction (i.e., an absurdity) follows, and to then conclude that the assumption must be false. The technique in question is known as *reductio ad absurdum,* or indirect proof, or proof by contradiction. Another natural rule is that to establish that some conditional of the form $\varphi \rightarrow \psi$ (where $\varphi$ and $\psi$ are any formulas in a logic $L$), it suffices to suppose $\varphi$ and derive $\psi$ based on this supposition. With this derivation accomplished, the supposition can be discharged, and the conditional $\varphi \rightarrow \psi$ established. The needed conclusion from the previous example (i.e., whether or not everyone loves Bill, and whether or not Katherine loves Dave) follows readily from such reasoning. (For an introduction to natural deduction, replete with proof-construction and proof-checking software, see Ref. 31.)

### Nonmonotonic Reasoning

Deductive reasoning is monotonic. That is, if $\varphi$ can be deduced from some knowledge base $\Phi$ of formulas (written $\Phi \vdash_D \phi$), then for any formula $\psi \notin \Phi$, it remains true that $\Phi \cup \{\psi\} \vdash_D \phi$ In other words, when $\mathcal{R}$ is deductive in nature, new knowledge never invalidates prior reasoning.

This process is not how human cognition works in real life. For example, at present, I know that my house is standing. But if, later in the day, while away from my home and working at RPI, I learn that a vicious tornado

passed over RPI, and touched down in the town of Brunswick where my house is located, I have new information that probably leads me to at least suspend judgment as to whether or not my house still stands. Or to take the much-used example from AI, if I know that Tweety is a bird, I will probably deduce that Tweety can fly, on the strength of a general principle saying that birds can fly. But if I learn that Tweety is a penguin, the situation must be revised: that Tweety can fly should now not be in my knowledge base. Nonmonotonic reasoning is the form of reasoning designed to model, formally, this kind of *defeasible* inference.

There are many different logic-based approaches that have been designed to model defeasible reasoning—default logic, circumscription, argument-based defeasible reasoning, and so on. (The *locus classicus* of a survey can be found in Ref. 32. In the limited space available in the present chapter, we can only briefly explain one of these approaches—argument-based defeasible reasoning, because it seems to accord best with what humans do as they adjust their knowledge through time.

Returning to the tornado example, what is the argument that supports the belief that the house stands (while one sits within it)? Here is Argument 1:

(1) I perceive that my house is still standing.
(2) If I perceive $\phi$, $\phi$ holds.
$\therefore$(3) My house is still standing.

Later on, we learned that the tornado had touched down in Brunswick, and devastating damage to some homes has come to pass. At this point ($t_2$), if one was pressed to articulate the current position on (3), one might offer something like this (Argument 2):

(4) A tornado has just (i.e., at some time between $t_1$ and $t_2$) touched down in Brunswick, and destroyed some houses there.
(5) My house is located in Brunswick.
(6) I have no evidence that my house was *not* struck to smithereens by a tornado that recently passed through the town in which my house is located.
(7) If a tornado has just destroyed some houses in town $T$, and house $h$ is located in $T$, and one has no evidence that $h$ is not among the houses destroyed by the tornado, then one ought not to believe that $h$ was not destroyed.
$\therefore$(8) I ought not to believe that my house is still standing (i.e., I ought not to believe (3).

The challenge is to devise formalisms and mechanisms that model this kind of mental activity through time. The argument-based approach to nonmonotonic reasoning does this. Although the details of the approach must be left to outside reading (33), it should be easy enough to see that the main point is to allow one argument to shoot down another (and one argument to shoot down an argument that shoots down an argument, which revives the original, etc.), and to keep a running tab on which propositions should be believed at any particular time.

Argument 2 above rather obviously shoots down Argument 1. Should one then learn that only two houses in Brunswick were leveled, and that they are both located on the other side of the town, Argument 2 would be defeated by a third argument, because this third argument would overthrow (6). With Argument 2 defeated, (3) would be reinstated, and back in my knowledge base. Notice that this ebb and flow in argument-versus-argument activity is far more than just straight deductive reasoning. (Logic can be used to model nondeductive reasoning that is not only nonmonotonic, but also inductive, abductive, probabilistic, model-based, and analogical, but coverage of these modes of inference is beyond the scope of the present entry). For coverage of the inductive and probabilistic modes of reasoning, see Ref. 34. For coverage of model-based reasoning, which is not based solely on purely linguistic formulas, but rather on models, which are analogous to states of affairs or situations on which linguistic formulas are true or false (or probable, indeterminate, etc.), see Ref. 35.

### Modal Logics

Logics can be used to represent knowledge, but advanced logics can also be used to represent knowledge about knowledge, and reasoning about knowledge about knowledge. Modeling such knowledge and reasoning is important for capturing human cognition, and in light of the fact that heretofore the emphasis in psychology of reasoning has been on modeling simpler reasoning that does not involve modals, the level of importance only grows. Consider the Wise Man Puzzle below as an illustration of modal reasoning to be captured:

> Suppose there are three wise men who are told by their king that at least one of them has a white spot on his forehead; actually, all three have white spots on their foreheads. We assume that each wise man can see the others' foreheads but not his own, and thus each knows whether the others have white spots. Suppose we are told that the first wise man says, "I do not know whether I have a white spot," and that the second wise man then says, "I also do not know whether I have a white spot." Now we would like to ask you to attempt to answer the following questions:
>
> 1. Does the third wise man now know whether or not he has a white spot?
> 2. If so, what does he know, that he has one or doesn't have one?
> 3. And, if so, that is, if the third wise man does know one way or the other, provide a detailed account (showing all work, all notes, etc.; use scrap paper as necessary) of the reasoning that produces his knowledge.

The logic able to answer these questions is a modal propositional epistemic logic; we refer to it simply as $\mathcal{L}_{KT}$. This logic is produced by adding to the propositional calculus the modal operators $\Box$ (traditionally interpreted as 'necessarily') and $\Diamond$ (traditionally interpreted as 'possibly'), with subscripts on these operators to refer to cognitive systems. Because we are here concerned with what cognitive systems believe and know, we will focus on the box, and will

rewrite $\Box_\alpha$ as $\mathbf{K}_\alpha$ [i.e., cognitive system $\alpha$ knows (something)]. So, to represent that 'Wise man A knows he doesn't have a white spot on his forehead,' we can write $\mathbf{K}_A$ (¬White(A)). Here's the grammar for $\mathcal{L}_{KT}$.

1. All wffs in the propositional calculus are wffs.
2. If $\phi$ is a closed wff, and $\alpha$ is a constant, then $\Box_\alpha\phi$ is a wff. Since we are here concerned with doxastic matters, that is, matters involving believing and knowing, we say that $\mathbf{B}_\alpha\phi$ is a wff, or, if we are concerned with 'knows' rather than 'believes,' that $\mathbf{K}_\alpha\phi$ is a wff.
3. If $\phi$ and $\psi$ are wffs, then so are any strings that can be constructed from $\phi$ and $\psi$ by the usual propositional connectives (e.g., $\rightarrow$,$\Lambda$,...).

Next, here are some key axioms and rules of inference:

**K** $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$

**T** $\Box\phi \rightarrow \phi$

**LO** ("logical omniscience") Where $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$, from $\Phi \vdash_D \psi$ and $\mathbf{K}_\alpha\phi_1$, $\mathbf{K}_\alpha\phi_2$,... infer $\mathbf{K}_\alpha\psi$

The first rule says that if one knows a conditional, then if one knows the antecedent of the conditional, one knows the consequent. The second says that if one knows some proposition, that proposition is true. The inference rule LO says that the agent $\alpha$ knows that which can be deduced from what she knows. This rule of inference, without restrictions placed on it, implies that if $a$ knows, say, the axioms of set theory (which are known to be sufficient for deductively deriving all of classic mathematics from them), $\alpha$ knows all of classic mathematics, which is not cognitively plausible. Fortunately, LO allows for the introduction of parameters that more closely match the human case. For example $\text{LO}^n$ would be the rule of inference according to which $\alpha$ knows the consequences of what she knows, as long as the length of the derivations (in some fixed proof theory) of the consequences does not exceed $n$ steps.

To ease exposition, we restrict the solution to the two-wise man version. In this version, the key information consists in these three facts:

1. A knows that if A does not have a white spot, B will know that A does not have a white spot.
2. A knows that B knows that either A or B has a white spot.
3. A knows that B does not know whether or not B has a white spot.

Here is a proof in $\mathcal{L}_{KT}$ that solves this problem:

1. $\mathbf{K}_A(\neg\text{White(A)} \rightarrow \text{K}_\text{B}(\neg\text{White(A)}))$ (first fact)
2. $\mathbf{K}_A(\mathbf{K}_B(\neg\text{White(A)} \rightarrow \text{White(B)}))$ (second fact)
3. $\mathbf{K}_A(\neg\mathbf{K}_B(\text{White(B)}))$ (third fact)
4. $\neg\text{White(A)} \rightarrow \mathbf{K}_B(\neg\text{White(A)})$ 1, T
5. $\mathbf{K}_B(\neg\text{White(A)} \rightarrow \text{White(B)})$ 2, T
6. $\mathbf{K}_B(\neg\text{White(A)}) \rightarrow \mathbf{K}_B(\text{White(B)})$ 5, K
7. $\neg\text{White(A)} \rightarrow \mathbf{K}_B(\text{White(B)})$ 4, 6
8. $\neg\mathbf{K}_B(\text{White(B)}) \rightarrow \text{White(A)}$ 7
9. $\mathbf{K}_A(\neg\mathbf{K}_B(\text{White(B)}) \rightarrow \text{White(A)})$ 4–8, 1, LO
10. $\mathbf{K}_A(\neg\mathbf{K}_B(\text{White(B)})) \rightarrow \mathbf{K}_A(\text{White(A)})$ 9, K
11. $\mathbf{K}_A(\text{White(A)})$ 3, 10

The foregoing solution closely follows that, provided by Ref. 32; this solution lacks a formal semantics for the inference rules in question. For a fuller version of a solution to the arbitrarily iterated $n$-wise man version of the problem, replete with a formal semantics for the proof theory used, and a real-life implementation that produces a logic-based cognitive system, running in real time, that solves this problem; see Ref. 36.

### Examples of Logic-Based Cognitive Systems

There are many logic-based cognitive systems that have been engineered. It is important to know that they can be physically embodied, have to deal with rapid-fire interaction with the physical environment, and still run efficiently.

For example, Amir and Maynard-Reid (37) built a logic-based robot able to carry out clerical functions in an office environment; similar engineering has been carried out in Ref. (38). For a set of recent examples of readily understood, small-scale logic-based cognitive systems doing various things that humans do; see Ref. 39.

There is insufficient space to put on display an actual logic-based cognitive system of a realistic size here. So see the afore-mentioned references for further details.

## CONCLUDING REMARKS

In recent decades, the research on cognitive systems has progressed to the extent that we can start to build computational systems that mimic the human mind to some degree, although there is a long way to go before we can fully understand the architecture of the human mind and thereby develop computational cognitive systems that replicate its full capabilities.

Some example cognitive systems have been presented here. Yet, it is still necessary to explore more fully the space of possible cognitive systems (40,41), to further advance the state of the art in cognitive systems, in cognitive modeling, and in cognitive science in general. It will also be necessary to enhance the functionalities of cognitive systems so that they can be capable of the full range of intelligent behaviors. Many challenges and issues need to be addressed (1,2). We can expect that the field of cognitive systems will have a significant and meaningful impact on cognitive science and on computer science both in terms of understanding cognition and in terms of developing artificially intelligent systems. The eventual objective of constructing embodied systems that can perceive, understand, and interact with their environment to achieve human-like performance in various activities drives this field forward.

## BIBLIOGRAPHY

1. R. Sun, The importance of cognitive architectures: An analysis based on CLARION, *J. Experimen. Theoret. Artif. Intell.*, In press.

2. P. Langley, J. Laird, and S. Rogers, Cognitive architectures: Research issues and challenges, *Cog. Sys. Res.*, In press.
3. R. W. Pew and A. S. Mavor (eds), *Modeling Human and Organizational Behavior: Application to Military Simulations*. Washington, D.C.: National Academy Press, 1998.
4. F. Ritter, N. Shadbolt, D. Elliman, R. Young, F. Gobet, and G. Baxter, *Techniques for Modeling Human Performance in Synthetic Environments: A Supplementary Review*. Dayton, OH: Human Systems Information Analysis Center, Wright-Patterson Air Force Base, 2003.
5. A. Newell, *Unified Theories of Cognition*, Cambridge, MA: Harvard University Press, 1990.
6. R. Sun, *Duality of the Mind*, Mahwah, N.J.: Lawrence Erlbaum Associates, 2002.
7. R. Sun, Desiderata for cognitive architectures, *Philosoph. Psych.*, **17** (3): 341–373, 2004.
8. P. Rosenbloom, J. Laird, and A. Newell, *The SOAR Papers: Research on Integrated Intelligence*. Cambridge, MA: MIT Press, 1993.
9. J. Anderson and C. Lebiere, *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates, 1998.
10. R. Sun and T. Peterson, Autonomous learning of sequential tasks: experiments and analyses, *IEEE Trans. Neural Networks*, **9** (6): 1217–1234, 1998.
11. R. Sun, E. Merrill, and T. Peterson, From implicit skills to explicit knowledge: A bottom-up model of skill learning, *Cog. Sci.*, **25** (2): 203–244, 2001.
12. R. Sun, *A Tutorial on CLARION*. Technical report, Cognitive Science Department, Rens-selaer Polytechnic Institute. Available: http://www.cogsci.rpi.edu/~rsun/sun.tutorial.pdf.
13. A. Reber, Implicit learning and tacit knowledge, *J. Experimen. Psych.: General*, **118** (3): 219–235, 1989.
14. C. Seger, Implicit learning, *Psycholog. Bull.*, **115** (2): 163–196, 1994.
15. A. Cleeremans, A. Destrebecqz and M. Boyer, Implicit learning: News from the front. *Trends in Cog. Sci.*, **2** (10): 406–416, 1998.
16. D. Medin, W. Wattenmaker, and R. Michalski, Constraints and preferences in inductive learning: An experimental study of human and machine performance, *Cog. Sci.*, **11**: 299–339, 1987.
17. R. Nosofsky, T. Palmeri, and S. McKinley, Rule-plus-exception model of classification learning, *Psycholo. Rev.*, **101** (1): 53–79, 1994.
18. A. Karmiloff-Smith, From meta-processes to conscious access: Evidence from children's metalinguistic and repair data, *Cognition*, **23**: 95–147, 1986.
19. C. Watkins, *Learning with Delayed Rewards*. Ph.D Thesis, Cambridge, UK: Cambridge University, 1989.
20. W. Stanley, R. Mathews, R. Buss, and S. Kotler-Cope, Insight without awareness: On the interaction of verbalization, instruction and practice in a simulated process control task, *Quart. J. Experimen. Psych.*, **41A** (3): 553–577, 1989.
21. R. Michalski, A theory and methodology of inductive learning, *Artif. Intell.*, **20**: 111–161, 1983.
22. W. Schneider and W. Oliver, An instructable connectionist/control architecture, in K. VanLehn (ed.), *Architectures for Intelligence*, Hillsdale, NJ: Erlbaum, 1991.
23. M. R. Quillian, Semantic memory, in M. Minsky (ed.), *Semantic Information Processing*. Cambridge, MA: MIT Press, 1968, pp. 227–270.
24. D. Rumelhart, J. McClelland and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Micro structures of Cognition*. Cambridge, MA: MIT Press, 1986.
25. R. Sun, Robust reasoning: Integrating rule-based and similarity-based reasoning. *Artif. Intell.*, **75** (2): 241–296, 1995.
26. T. Nelson, (ed.) *Metacognition: Core Readings*. Allyn and Bacon, 1993.
27. J. D. Smith, W. E. Shields, and D. A. Washburn, The comparative psychology of uncertainty monitoring and metacognition, *Behav. Brain Sci.*, **26** (3): 317–339, 2003.
28. R. Sun and I. Naveh, Simulating organizational decision making with a cognitive architecture CLARION, *J. Artif. Soc. Social Simulat.*, **7** (3): 2004.
29. L. Rips, *The Psychology of Proof*. Cambridge, MA: MIT Press, 1994.
30. L. Wos, R. Overbeek, E. Lusk, and J. Boyle, *Automated Reasoning: Introduction and Applications*. New York: McGraw Hill, 1992.
31. J. Barwise and J. Etchemendy, *Language, Proof and Logic*, New York: Seven Bridges, 1999.
32. M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*. Los Altos, CA: Morgan Kaufmann, 1987.
33. J. L. Pollock, How to reason defeasibly, *Artif. Intell.*, **57** (1): 1–42, 1992.
34. B. Skyrms, *Choice and Chance: An Introduction to Inductive Logic*. Belmont, CA: Wadsworth, 1999.
35. P. Johnson-Laird, *Mental Models*. Harvard, MA: Harvard University Press, 1983.
36. K. Arkoudas and S. Bringsjord, Metareasoning for multi-agent epistemic logics. *Fifth International Conference on Computational Logic In Multi-Agent Systems (CLIMA 2004)*, Lecture Notes in Artificial Intelligence (LNAI), **3487**: 111–125, 2005.
37. E. Amir and P. Maynard-Reid, LiSA: A robot driven by logical subsumption, *Proc. of the Fifth Symposium on the Logical Formalization of Commonsense Reasoning*, AAAI Press, 2001.
38. S. Bringsjord, S. Khemlani, K. Arkoudas, C. McEvoy, M. Destefano, and M. Daigle, Advanced Synthetic Characters, Evil, and E, in M. Al-Akaidi and A. El Rhalibi (eds.), *Game-On 2005, 6th International Conference on Intelligent Games and Simulation*, Ghent-Zwijnaarde, Belgium: European Simulation Society, 2005, pp 31–39.
39. E. Mueller, *Commonsense Reasoning*. San Francisco, CA: Morgan Kaufmann, 2006.
40. A. Sloman and R. Chrisley, More things than are dreamt of in your biology: Information processing in biologically-inspired robots. *Cog. Sys. Res.*, **6** (2): 145–174, 2005.
41. R. Sun and C. Ling, Computational cognitive modeling, the source of power and other related issues. *AI Magazine*, **19** (2): 113–120, 1998.

## FURTHER READING

A. Clark and A. Karmiloff-Smith, The cognizer's innards: A psychological and philosophical perspective on the development of thought. *Mind and Language*, **8** (4): 487–519, 1993.

A. Maslow, *Motivation and Personality*, 3rd Edition. New York: Harper and Row, 1987.

A. Newell and H. Simon, Computer science as empirical inquiry: Symbols and search. *Commun. of ACM*, **19**: 113–126, 1976.

R. Sun and T. Peterson, Multi-agent reinforcement learning: Weighting and partitioning. *Neural Networks*, **12** (4–5): 127–153, 1999.

R. Sun, P. Slusarz, and C. Terry, The interaction of the explicit and the implicit in skill learning: A dual-process approach, *Psycholog. Rev.*, **112** (1): 159–192, 2005.

R. Sun and X. Zhang, Accessibility versus action-centeredness in the representation of cognitive skills, *Proc. of the Fifth International Conference on Cognitive Modeling*, Bamberg, Germany, 2003, pp. 195–200.

RON SUN,
SELMER BRINGSJORD
Rensselaer Polytechnic
Institute, Troy, New York