The Cogitoid: A Computational Model of Cognitive Behaviour. (Revised Version)

Wiedermann, Jiří
1998

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

The Cogitoid: A Computational Model of Cognitive Behaviour (Revised Version)

Jiří Wiedermann

Technical report No. 743

March, 1998

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+4202) 66 05 35 20   fax: (+4202) 85 85 789
e-mail: wieder@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

The Cogitoid: A Computational Model of Cognitive Behaviour (Revised Version)

Jiří Wiedermann [1]

Technical report No. 743
March, 1998

Abstract

A new interactive computational device with learning abilities — a so–called *cogitoid* — is proposed. Formally it is represented by a lattice of concepts with two basic operations — abstraction and concretization. Concepts are used for representing knowledge gained from cogitoid's sensory inputs as well as for mediating its actions to its effectors. Its computational behaviour supports a formation of new concepts and of both excitatory and inhibitory associations among them. It is proved that cogitoids are able to realize behaviour elicited by the presentation of specific stimulus–response patterns, such as retrieval by causality, learning of sequences, learning of composed concepts from partial ones, or similarity based behaviour. Also instances of Pavlovian conditioning can be acquired. Final examples give the plausible algorithmic explanation of the so–called operant conditioning that is determined by the positive or negative stimuli which occur after the responses. The case of delayed reinforcement is also handled. From the point of view of their computational power cogitoids equipped with the respective number of tapes are in a certain reasonable way equivalent to multitape interactive Turing machines.

Keywords

Cognitive computing; Computational models of the mind and brain; Turing machines; Learning

[1] This research was partially supported by GA ČR Grant No. 201/98/0717, EU Grant No. INCO–COOP 96–0195 'ALTEC–KIT' and Ministry of Education grant OK–304

1 Introduction

The idea of thinking about the brain as being a computational device that can be mathematically formalized and explained, has attracted a lot of attention in artificial intelligence, cognitive sciences, and in computer science. Within the latter science we are nowadays witnessing rapid development especially in the related field of neurocomputing. Besides various kinds of neural nets designed for specific learning tasks, computational models of the brain, or mind, based on paradigms of neurocomputing, have also emerged (for a recent overview cf. [1] or [6]). Along these lines, the *memory surface model* by Goldschlager [5], and the *neural tabula rasa* by Valiant [7] seem to belong among the most elaborated models (for a brief overview of these, and other computational models of the brain, cf. [8], [10]). Unfortunately, so far none of these or related models appears to be able to formally, mathematically treat more complicated cognitive tasks, for various reasons. In the case of memory surface this is because the Goldschlager's model is not precise enough in details that are necessary in any formal, or semi–formal reasoning. Also, the mechanism of inhibition that seems to be necessary to explain some cases of operant conditioning is missing in his model. On the other hand, this model offers a good conceptual framework for the explanation of basic cognitive processes, such as concept and association formation, and even offers a plausible "high–level" explanation of some higher brain functions. Contrary to this, Valiant's approach is very precise in details. It offers the notion of the so–called neuroid, a kind of a neuron that can be programmed to fullfil various atomic cognitive tasks out of which more complicated tasks ought to be assembled. At the same time this seems to introduce some limits to the potential of this model: to explain, or model more complicated cognitive tasks at the level of actions of individual neuroids, moreover in a probabilistic setting, is not easy. A similar objection holds for other "bottom–up" approaches starting at neuronal level.

In this circumstance a sufficiently high–level model of the brain that would concentrate onto the global aspects of mental processes and would be sufficiently formal to eventually allow an exact mathematical reasoning, but at the same time would abstract from "implementation details" at the neuronal level, could be useful.

In the paper at hand a candidate for such a model — a so–called cogitoid, is presented. The basic entities which the model deals with are concepts. Formally, the cogitoid is represented by a lattice of concepts with operation of abstraction and concretization. Moreover, in the course of cogitoid's computation associations among concepts keep developing. The model is not programmable in the standard sense, but can be trained to perform certain cognitive tasks by providing it with the right sequence of inputs.

The paper is organized as follows. In section 2, mathematical and technical background needed to cogitoid's definition, and the definition itself, is given.

Next, in section 3 it is shown that the model as defined in the previous section is sound, in a certain pragmatic sense: it presents not only a suitable framework for the description and formalization of certain cognitive tasks, as described in various experiments with animals, but is able to also offer an algorithmic explanation of the respective phenomena, with sufficient accuracy. Its minimality is supported by the fact that the proofs of the respective theorems concerning the cognitive power of cogitoids make use of all of model's features.

The above mentioned properties of cogitoids are first demonstrated on simple cognitive tasks such as the acquisition of a behaviour elicited by the repeated presentation of specific stimulus–response patterns, acquisition of sequences, the emergence of composed concepts from simpler ones, or the behaviour elicited by similar circumstances. Next, examples of standard textbook instances of Pavlovian conditioning are shown. This kind of behaviour refers to the fact that animals can be conditioned in such a way that they will tend to activate a concept that is apparently unrelated to the stimulus at hand. Final examples also give the plausible algorithmic explanation of the so–called operant conditioning. This is the behaviour that is acquired by exposing an animal to circumstances in which a certain stimuli–response behaviour is consistently rewarded or punished *after* the response is obtained. The case of delayed reinforcement is considered as well.

In Section 4 the computational power of cogitoids is investigated. It is shown that any cogitoid equipped with Turing machine tapes can be trained to computationally behave, for some time, as any other given Turing machine.

Finally, section 5 contains some open problems and conclusions.

2 The Cogitoid

2.1 Lattices of Concepts

Any cogitoid can be seen as central part of a finite interactive computational device that interacts with its environment with the help of its sensors and effectors. The respective information flowing from sensors into a cogitoid and from a cogitoid to its effectors is represented by concepts. In the simplest case concepts can be seen as Boolean vectors of fixed size. Each Boolean value in such a vector reports presence or absence of some feature in the respective concept. This was an earlier framework investigated in a technical report [9]. However, in the present approach we will present a more general view in which concepts are basic entities, with no known internal structure. Each concept simply represents some "event" as perceived by a cogitoid. Over a set of concepts certain operations are defined in such a way that the resulting algebraical structure forms a lattice (cf. [12] where lattices of concepts are also used for knowledge representation).

A *lattice* $\mathcal{L}(\Sigma)$ over the set Σ is the set Σ that is partially ordered by the relation $\leq$, that has the least element ℓ and the greatest element g, and any two elements a and b of Σ have a supremum $a \vee b$ called *join* and an infimum $a \wedge b$ called *meet.*

For any $\Pi \subseteq \Sigma$ the algebra $\mathcal{L}(\Pi)$ is a *sublattice* of $\mathcal{L}(\Sigma)$ iff $\mathcal{L}(\Pi)$ itself is a lattice. Note than for any $a \in \Sigma$ and for $\Pi = \{b \in \Sigma | b \leq a\}$ the system $\mathcal{L}(\Pi)$ is a sublattice (in fact, it is a principal ideal) denoted as $\mathcal{L}_a$. Its greatest element is a while its least element is ℓ. For more details cf. [2].

A *lattice of concepts* is a lattice whose elements are called *concepts*. For any two elements a and b of such a lattice, with $a \leq b$, we say that a is an *abstraction* of b, while b is a *concretization* of a. Then, a supremum of any two of its elements is the smallest concretization of these elements, while their infimum is the largest abstraction of these elements. We shall say that two concepts are *non–meeting* iff their largest abstraction is equal to the least element of the respective lattice.

2.2 Computational Mechanisms of Cogitoids

Basically, any cogitoid is a lattice of concepts over which certain additional relations, called associations, evolve during the course of cogitoid's computation. At the same time some additional information about concepts and associations is kept and updated in order to discover, and remember certain specific patterns in the data processed so far. Based on these information and on the current input to cogitoid some concepts are activated. By the virtue of cogitoid's computational laws these active concepts give rise in the next computational step to another set of active concepts. The latter set serves then as the output of the cogitoid. This output takes place simultaneously with the reading of the next input.

Thus, at any time t each concept in a cogitoid $\mathcal{C}$ can be in either of the two possible states: in an *active*, or in a *passive*, state. Moreover, at any time all concepts are present with a various strengths. The strength of each concept is a non–negative integer. Initially, at the beginning of cogitoid's computation, the strength of all concepts is set to zero and all concepts are in a passive state. Then, when the computation starts, some concepts gets activated and their strength is increased by a small amount. The concepts with a positive strength are called *initialized concepts.*

Within the set of all concepts in any cogitoid there are two distinguished disjoint subsets of non–meeting concepts, called positive or negative *affects*, or *operant concepts.* They correspond to pleasurable or painful feelings of animals.

The affects determine the so–called *quality* of concepts. The quality of a concept at time t is an element from the set $\{+1, -1, \pm 1\}$ and is defined only for concepts active at time t. Positive affects have always positive quality, while negative affects have always negative quality. From affects the quality propagates to larger concepts in the respective partial lattice order. Should a concept obtain in this way two opposite signs it gets the quality ± 1. By similar rules also the concepts that did not acquire quality by the previous process obtain their quality: they will simply inherit their quality from the smallest concepts that encompass them. If there are no affects active at time t then all active concepts obtain a positive quality at that time.

Concepts may be activated either directly from cogitoid's input, or indirectly, via so–called associations, from concepts activated at previous step.

Any ordered pair of form (a, b) of concepts is called an *association* of a with b (or between a and b), written also $a \rightarrow b$. We also say that a and b are then correlated by *causality.* The associations appear in two forms: as excitatory ones, denoted as

$(a,b)^+$, and inhibitory ones, denoted as $(a,b)^-$.

Similarly as to concepts the strength is also assigned to each association. Between any pair of concepts there are associations of both types. Their strength is initially set to zero and when the respective concepts are activated the associations are strengthened, subject to special rules to be described in the sequel. Any association with a positive strength is called an *initialized association.*

If $a \wedge b = c \neq \ell$, then we shall say that the concept a resembles the concept b (or a is similar to b) in c. Otherwise, when $c = \ell$, the concepts are considered not to be similar[2]. When a is similar to b we also say that the former two concepts are correlated by *similarity*, or by *resemblance* via the concept c and we write $a \approx b$ (or $b \approx a$ since this is a symmetric relation). In a cogitoid the previous relation will be also represented by two associations: $a \rightarrow b$ and $b \rightarrow a$.

The rules for concept activations are as follows:

- some concepts are activated directly from the input (by external stimuli);
- other concepts may be activated indirectly (by internal stimuli), in the following way:
  - by the virtue of abstraction: the activation of some concept a causes simultaneous activations of all its abstractions;
  - by simultaneous occurrence: if two concepts a and b are activated simultaneously (what shall be sometimes called as the appearance of $\{a,b\}$), they will activate also their single concretization $a \vee b$ (which in turn will activate all its abstractions);
  - via associations: when A is the set of all active concepts at time t and B is the set of all passive concepts at that time that are correlated to the concepts from A by causality, then the maximally excited concept $b \in B$ will get activated.

The excitation of each concept $b \in B$ is proportional to the sum of strength of all (initialized) excitatory associations from currently active concepts decreased by the sum of strengths of all (initialized) inhibitory associations from currently active concepts. The respective mechanism that selects and activates the maximally excited concept is called the *selection mechanism.* If there are more concepts $a_1, a_2, \ldots, a_k$ equally excited satisfying the maximality condition, then the selection mechanism selects all of them. In such a case their concretization $a = a_1 \vee a_2 \ldots \vee a_k$ gets activated. Note that until that time a need not be initialized.

Note that via associations not only concepts related by causality, but also by resemblance may be activated.

[2] Depending on the size of c we could introduce resemblance relations of a various degree of similarity; for simplicity reasons we abstain from such an idea. This is why c will not be mentioned in the sequel in the respective similarity relation.

Besides concepts reinforcement also associations strengthening takes place in a cogitoid. An association $a \rightarrow b$ will be strengthened at two occasions:

- whenever a is activated, b is initialized and $a \approx b$;
- whenever the concept b is activated immediately after a was activated in the previous step. When both concepts at hand have been activated from the input the amount of the respective association strengthening is grater than in other cases. Moreover, when a was an affect then the amount of strengthening is still greater than in the latter case.

What type of associations — whether an excitatory or inhibitory one — will be reinforced depends on the quality of the concept a at that time. If this quality was positive (negative) then excitatory (inhibitory) associations are reinforced; otherwise both types of associations are strengthened.

2.3 Computation of a Cogitoid

For any finite lattice $\mathcal{L}(\Sigma)$ the respective cogitoid $\mathcal{C}$ can be seen as a *transducer* that translates an infinite input string σ over the finite alphabet 2^Σ into an infinite output string γ, also over 2^Σ. Thus, the elements of both σ and γ are sets of concepts. The elements of σ are sometimes also called *contexts* while those of γ *behaviors*, or *actions*.

A *configuration* c_t of $\mathcal{C}$ at time t is a complete list of all initialized concepts and associations in $\mathcal{C}$ at time t, inclusively their strengths, and in case of concepts also inclusively their qualities and states.

The computation of a cogitoid consists of an infinite sequence of computational steps.

Let the set $I_t \in \sigma$ be the set of concepts at the input of $\mathcal{C}$ at time t, for some $t \geq 0$, let $i_t = \vee_{a \in I_t} a$.

Let c_t be the configuration of $\mathcal{C}$ at the beginning of the t–th computational step, prior to reading new inputs.

Let O_t be the set of all active concepts in c_t that, as we shall see, also present an output from $\mathcal{C}$ at that time.

We shall show that $\mathcal{C}$ maintains the following invariant relation that holds between I_{t-1}, O_{t-1} and O_t at the beginning of t–th computational step:

- O_t is the union of concepts from two ideals of $\mathcal{L}(\Sigma)$. The first ideal is $\mathcal{L}_{i_{t-1}}$ and the second one $\mathcal{L}_a$, where $a \in O_t$ is the maximally excited concept from among all concepts that are correlated at time $t-1$ to O_{t-1} via associations.

At time $t = 0$ no concepts and no associations are present in $\mathcal{C}$. Thus, $O_t = \emptyset$ and the invariant holds vacuously.

For any $t > 0$, assume that the cogitoid finds itself in a configuration c_t, with O_t being the set of active concepts that fulfills the previous invariant.

We shall show that after performing the t–th computational step the above invariant will be restored.

The t-th computational step consists of 6 phases:

Phase 1: *Input/Output:*

1. *The Input:* The t–th symbol I_t from the input string σ is read. Subsequently, all concepts from I_t are activated by external stimuli. This gives rise to the activation of the concretization $i_t = \vee_{a \in I_t} a$ by the virtue of simultaneous occurrence. In parallel, all concepts in $\mathcal{L}_{i_t}$ are also activated by the rule of abstraction activation.

2. *The Output:* The t–th symbol of the output string γ corresponding to the behaviour O_t is produced as the output at time t.

Phase 2: *Activation of New Concepts by Internal Stimuli:* Based on the set O_t of active concepts the *selection mechanism* mentioned in Section 2.2 will activate the mostly excited concept a from the set P_t. Here P_t denotes the set of those initialized passive concepts in c_t that are correlated with the set O_t via associations. Simultaneously, by the rule of abstraction activation all abstractions of a are activated. Hence, the entire ideal $\mathcal{L}_a$ gets activated.

Phase 3: *Determining the Quality of Concepts:* Let $A_t = \mathcal{L}_{i_t} \cup O_t \cup \mathcal{L}_a$ be the set of all active concepts at this time. The quality of each concept in A_t is determined according to the rules described in Section 2.2.

Phase 4: *Long–Term Memorization:*

1. *Strengthening of Concepts:* The strength of all concepts in A_t is increased by some fixed positive constant.

2. *Strengthening of Associations:* The strength of associations between all concepts similar to currently active concepts, and all subsequently occurring different active concepts is increased, according to the strengthening rules described in Section 2.2. In the first case, the associations leading to, and from, the concepts similar to concepts in O_t are strengthened. In the latter case, the first concept in the pair is selected from the set O_t (since the respective concepts have been activated by the end of $(t-1)$–st step) while the second one in the pair is chosen from the set $\mathcal{L}_{i_t} \cup \mathcal{L}_a - O_t$ (since the respective concepts have been activated in the present, t–th step).

Phase 5: *Gradual Forgetting:* The strength of all initialized concepts that are not active at this very moment, and the strength of all association among them, is decreased by some fixed amount.

Phase 6: *Deactivation:* The concepts in $O_t - \mathcal{L}_{i_t} - \mathcal{L}_a$ are deactivated. The set of currently active concepts remains only $\mathcal{L}_{i_t} \cup \mathcal{L}_a$, and this is going to be the set O_{t+1} of all active concepts at the beginning of the next step.

The previous six phases uniquely determine the next configuration c_{t+1} of $\mathcal{C}$. Note that in Phase 6 the invariant has been restored.

From the previous description it is seen that a cogitoid is completely specified by the respective lattice, affects, and the constants by which concepts and associations are reinforced or by which the forgetting is modeled. The computational mechanism of all cogitoids is the same. By varying the respective constants one can influent the speed of cogitoid's learning and forgetting, its sensitivity to affects or to external stimuli.

3 Simple Cognitive Tasks

3.1 Behavioristic Learning

The previous principles are enough to ensure the elicitation of any desired behaviour of a cogitoid solely by purposefully externally applied stimulus–response pattern via the activation of the respective concepts. Namely, by repeating the required stimulus–response pattern enough times, strong concepts representing stimuli and the respective responses are formed with strong associations among them. Then, any later activation of a stimulating concept will invoke the respective response. This can be done with many different patterns, and patterns can be presented to a cogitoid in a random order, and some of them also in parallel, grouped arbitrarily.

The details are given in the next theorem:

Theorem 3.1 *Let $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$, $k > 0$, be two disjoint sets of non–meeting concepts. We shall say that a pair $\{x_i, y_i\}$, for any $i = 1, 2, \ldots, k$, is presented sequentially to a cogitoid $\mathcal{C}$ at time t iff x_i is at the input of $\mathcal{C}$ at time t, y_i is at the input at time $t+1$ and ℓ (the least element of the respective lattice) is at the input at time $t+2$. Let pairs be presented randomly to a cogitoid $\mathcal{C}$, in any order, with the possibility of presenting a few randomly chosen pairs at the same time.*

Then, later on, after each pair has been presented to $\mathcal{C}$ several times, whenever x_i is presented to $\mathcal{C}$ at some step, y_i will be activated in the next step, for any $1 \leq i \leq k$.

Sketch of the proof: First assume that at time t the pair $\{x_i, y_i\}$ alone is presented sequentially to $\mathcal{C}$ for the first time. Thus, at time t the concept x_i is activated from the input (in Phase 1.1, in Section 2) and as a member of A_t it will remain active also at the beginning of time $t+1$ when y_i will be activated from the input. Then, according to Phase 4.2 the association $x_i \rightarrow y_i$ will be established by setting its strength to the value c_3. This association will be in the future strengthened at any similar occasion. Thus, when after a few repetitions of the above process the successor association $x_i \rightarrow y_i$ will become the strongest one among all associations invoked by x_i, whenever x_i will be activated, y_i will get in turn activated by the virtue of Phase 2. The above arguments hold for any $1 \leq i \leq k$.

Note that when more randomly chosen pairs will be presented to $\mathcal{C}$ at the same time, then concepts joining all concepts appearing simultaneously will also emerge. Nevertheless, the probability that such concepts will reappear in the near future is small and therefore they will tend to be forgotten due to Phase 5.

□

In a similar manner as above also sequences of concepts can be acquired:

Theorem 3.2 *Let $(x_1, x_2, \ldots, x_k)$, $k > 1$, be a sequence of non–meeting concepts in which each concept occurs exactly once. Let the pairs $\{x_i, x_{i+1}\}$ be sequentially presented to a cogitoid $\mathcal{C}$, in any random order, with the possibility of presenting a few randomly chosen pairs at the same time, for any $k > i > 0$.*

Then, later on, after each pair has been presented to $\mathcal{C}$ several times, whenever x_i alone is presented to $\mathcal{C}$ at some step, for any $1 \leq i < k$, concepts $x_{i+1}, \ldots, x_k$ will be subsequently activated in $k - i$ next steps, one concept at each step.

Sketch of the proof: Under the assumptions of the theorem, similarly as in the previous proof, associations of form $x_i \to x_{i+1}$ start to emerge in $\mathcal{C}$, for any $k > i > 0$. After these associations start to be strong enough to activate x_{i+1} whenever x_i gets activated, the cogitoid will start to behave as predicted in the theorem.

□

The next theorem shows that in a cogitoid more complex, compound concepts are spontaneously built from partial ones that resemble each other:

Theorem 3.3 *Let $\mathcal{C}$ be any cogitoid, let a, b be any concepts that resemble each other in some $c = a \wedge b \neq \ell$. Let a or b be repeatedly activated at some randomly chosen times. Then the later activation of c may activate the joint concept $a \vee b$ even though $a \vee b$ has never been initialized before.*

Sketch of the proof: Since $a \approx b$, $a \approx c$, and $b \approx c$, after both a and b have already been activated a few times strong associations between a, b, and c, in either direction, will be established. Then some later activation of c alone will equally excite both a and b. When no other concepts are more excited the selection mechanism will eventually in Phase 2 of cogitoid's computation activate $a \vee b$ (see the description of selection mechanism in Section 2.2 for the case when more concepts satisfy the condition of maximal excitation).

□

The last theorem in this section is concerned with the "similarity based" behaviour that forms the basis of more complex behaviors [11]:

Theorem 3.4 *Let $\mathcal{C}$ be any cogitoid, let $X = \{x_1, x_2, \ldots, x_k, y\}$ be a set of concepts, let $x_i \to y$ be strong associations established between the respective concepts, and let there be no other associations that could excite y.*

Then, later on, whenever any concept $x \notin X$, with $x \wedge y = \ell$ and x so far uncorrelated to any x_i, is presented to $\mathcal{C}$ at time t, then

- *when there exist at least one i such that $x \approx x_i$ for some $1 \leq i \leq k$, then y will be activated at time $t+2$;*
- *when x does not resemble any x_i for all $1 \leq i \leq k$, y will not be activated at time $t+2$.*

Sketch of the proof: When some $x \notin X$ will appear at the input at time t, by virtue of similarity it will in turn invoke x_i. This in turn will invoke y at time $t+2$. The latter concept cannot be invoked sooner due to our assumptions on y and x.

It is obvious that when x does not resemble any x_i, the activation of x will have no effect on y since due to our assumptions x can excite neither any x_i, nor any abstraction of it, nor y itself.

□

3.2 Pavlovian Conditioning

Translated to our terminology, the Pavlovian conditioning is a phenomenon in which an animal can be conditioned to activate a concept as a response to an apparently unrelated stimulating concept (cf. [7], p. 217).

This behaviour and its variations can be precised, modeled, and explained in the framework of cogitoids. The following theorem describes in depth various cases of Pavlovian conditioning that are frequently treated in books on animal psychology:

Theorem 3.5 *Let there be a cogitoid $\mathcal{C}$ containing non–meeting concepts a, b, r, s in which such a strong association $s \longrightarrow r$ has been established that s, a stimulus, alone, and only s, elicits r, a response. Let $\mathcal{C}$ undergo the* basic training process *that consists of repeating the following two steps a few times:*

- *at the beginning of a computational cycle, s appears simultaneously with another concept a, as the set $\{a, s\}$, at the input of $\mathcal{C}$. The concept a has so far no particular associations to r;*
- *at the beginning of the next computational cycle, there in "no input" to $\mathcal{C}$ — i.e., the minimal abstraction ℓ corresponding to the empty input is activated;*

Then:

1. *during the basic training process $\mathcal{C}$ starts to activate r at every second computational step;*
2. *when after the end of the basic training process the input a alone is presented to $\mathcal{C}$ in the first step, and ℓ in the second step, $\mathcal{C}$ will activate r in the third step, for some time;*

3. (extinction): *after some time, instead of answering r in the third step, $\mathcal{C}$ will start to answer ℓ;*

4. (inhibition): *moreover, when after the end of the basic training process $\mathcal{C}$ will undergo an additional training consisting of repeatedly performing the next two steps:*

   - *in the first step, an input chosen randomly from the set $\{\{a,s\},\{a,b\}\}$ is presented to $\mathcal{C}$, where the concept b is a stimulus that does not elicit r by itself;*
   - *in the next step, ℓ is presented to $\mathcal{C}$,*

   then, after some time, a alone, presented in the first step, will continue to activate r in the third step, while $\{a,b\}$ presented simultaneously will activate ℓ in the third step.

Sketch of the proof: To see the first claim, note that the simultaneous occurrence of $\{a,s\}$ at the input in the first step will activate and strengthen $a \vee s$ and both a and s, according to Phase 1.1 and 4.1 in Section 2.3. In the next step, s will activate r since, according to our assumption, there is a strong association $s \to r$. Moreover, since ℓ appears at the input, in this step the associations $a \to \ell$, $s \to \ell$, and $a \vee s \to \ell$ also start to establish themselves very strongly, as associations between input concepts, according to the rules in Phase 4.2. Nevertheless, for some time, thanks to the strong initial association $s \to r$, the response r will be the most excited concept and hence the cogitoid keeps activating r in every second step due to the rule in Phase 2.

As far as the second claim is concerned, observe that after some time, when the concept $a \vee s$ becomes strong enough, being repeatedly at the input, the appearance of a alone, at some time, will activate, in the second step, $a \vee s$ by the relation of resemblance (Phase 2) and, in turn, still in the same step, also a and s as the abstractions of $a \vee s$. Finally, in the third step, s will activate r. Here we assume that thanks to this initially strong association the excitation of r is for some time greater than that of ℓ (coming from a, s, and $a \vee s$) in this step.

The extinction of r's activation will come into power when the lastly mentioned activation of r will be overruled by the activation of ℓ, since the respective associations from a, s and $a \vee s$ to ℓ are repeatedly strengthened when empty input in the second step is read by $\mathcal{C}$.

The inhibition can be explained by similar mechanisms as before. Namely, during the additional training, concepts $a \vee s$ and $a \vee b$ start to establish themselves, similarly as the associations of all combinations (joins) of concepts a, b, and s to ℓ.

When a alone appears at the input, in the next step, a activates $a \vee s$ by similarity rather than $a \vee b$ since due to the basic training the resemblance association $s \to a \vee s$ is very strong. This is followed by an immediate activation of s and a (the last rule in

Phase 2). In the third step the selection mechanism of $\mathcal{C}$ eventually activates r, via the initially strongly established association $s \rightarrow r$. Here we have to assume that via this association r is excited stronger then ℓ is via the previously mentioned associations, and stronger than $a \vee b$ is excited by resemblance .

However, when $\{a, b\}$ appears at the input, a, b, and $a \vee b$ are activated in the first step from the external stimuli. In the second step, not only $a \vee s$ is activated again, but a, b, and $a \vee b$ remain active also, as members of A_t. In the third step, ℓ, rather than r, will be the most excited concept (by the joint effort of all combinations of a,b, and s) and therefore will be activated. All other concepts will be deactivated in Phase 5 of this step. □

Note that in order to explain the Pavlovian conditioning no use of negative operant concepts and of the related inhibitory associations was needed. Also observe that in the case of extinction and inhibition the cogitoid keeps activating some concepts in its second and third step, but others than r.

3.3 Operant Conditioning

Now we show that our model is also able to realize so–called *operant behaviour*. This is a behaviour acquired, shaped, and maintained by stimuli occurring *after* the responses rather than before. Thus, the invocation of a certain response concept r is confirmed as a "good one" (by invoking the positive operant concept p) or "bad one" (the negative operant concept n) only after r has been invoked. It is the reward (p), or punishment (n) that act to enhance the likelihood of r being re–invoked under the similar circumstances as before.

The real problem here is hidden in the last statement which says that r should be re–invoked (or not re–invoked) only under similar circumstances as before. Thus, inhibition, or excitation of r must not depend on s alone: in some contexts, r should be inhibited, and in others, excited.

The implementation of such a behaviour is described in the following theorem. In this theorem the context in which the conditioning should occur is called *operant context*; it is represented by a concept that appears invariantly as the part of the input of a cogitoid during the circumstances at hand.

Theorem 3.6 *Let $\mathcal{C}$ be any cogitoid containing the non–meeting concepts a, n, r, p, s and z. Let $t_1 < t_2 < \ldots$ be the distinct times, with $t_{i+1} > t_i + 3$. Let a strong association $s \rightarrow r$ be established in $\mathcal{C}$ prior to time t_1. Let at times $t_1 < t_2 < \ldots$ $\mathcal{C}$ finds itself in the operant context z for four subsequent steps (i.e., in times t_i, $t_i + 1$, $t_i + 2$, and $t_i + 3$, for $i = 1, 2, \ldots$).*

Let $\mathcal{C}$ undergo either of following two trainings:

1. Negative Conditioning: *Let s, a stimulus, appear at the input at time $t_i + 1$ and let the activation of r, a response, at time $t_i + 2$ will be punished by the negative operant concept n appearing at the input at time $t_i + 3$.*

Then there exists a constant $k > 0$ such that for $i = 1, 2, \ldots, k$, s elicits r at time $t_i + 2$, whereas for $i = k+1, \ldots$, s will not elicit r at time $t_i + 2$.

2. Positive Conditioning: *Let s appear at the input at time $t_i + 1$ and let the activation of r at time $t_i + 2$ will be rewarded by the positive operant concept p appearing at the input at time $t_i + 3$.*

   Then r will be activated at time $t_i + 2$, for $i = 1, 2, \ldots$

Whenever s is presented to $\mathcal{C}$ and z is not active then r still will be activated in the next step.

Prior to giving the proof of this theorem, a few words to explain its essence are in order.

Observe that the assumptions of the theorem are fairly general. To take a lesson from behaviour conditioned by an operant concept, no particularly intensive, continuous, long term training is needed. It is enough when a similar circumstance, modeled by the concept s and the operant context z will occur a few times (i.e., at times $t_1 < t_2 < \ldots$), possibly with long intervals between the successive occurrences.

The "similarity" of circumstances is modeled by the requirement that the operant context z remains the same at all pertinent occasions. This is fulfilled e.g. in cases when, at all times, $\mathcal{C}$ finds itself in configurations that share a subset of sufficiently strong concepts that alone uniquely characterize the event (cf. also Theorem 3.4).

When the subsequently elicited action r will be always punished, the negative conditioning causes that after only a few punishments the cogitoid ceases to invoke r at these occasions. The positive reward will cause a significant increase of strength of the respective excitatory associations that will then significantly contribute to the excitation of r at the given circumstance. This in turn will have the effect that under similar circumstance, and also when r will compete with other candidates for activation, r will be activated.

Sketch of the proof: The idea of the proof is as follows: we have to achieve that whenever at each time t_i after the time t_1 the concepts z and s appear simultaneously at the input, $\mathcal{C}$ has to "recall" what it did at a similar occasion in the past, i.e., at time t_{i-1} and shorty afterwards. Then it has to accept the decision that was either approved by p (i.e., reinforce the excitation of r), or rejected by n (inhibit the excitation of r) at time $t_{i-1} + 3$. This is achieved by "coupling" the operant context z with the respective affect $a \in \{p, n\}$ and by building the association $z \vee a \rightarrow r$. Then, whenever at any future similar occasion, $z \vee a$ will be active jointly with s, depending on the quality of a, $z \vee a \rightarrow r$ will reinforce, or inhibit, the activation of r.

In a more detail this idea works as follows. Consider the state of affairs in our cogitoid at time t_1.

At this time, $\mathcal{C}$ finds itself in the operant context z and therefore z appears at the input for the first time.

At time $t_1 + 1$, z keeps appearing at the input, with s also appearing simultaneously. This will activate $s \vee z$ and reactivate z and s by the rules from Phase 1.1 in Section 2.3. Subsequently, in the next step, r will be activated via $s \rightarrow r$, and the association $z \rightarrow r$ and $z \vee s \ \rightarrow r$ will emerge.

In turn, at time $t_1 + 3$, external activation of an operant concept a will follow, according to our assumptions. Thanks to our assumption on z, this concept will still be at the input, and thanks to its simultaneous occurrence with a, a composed concept $z \vee a$ will thus appear.

Next time, when z appears at the input, both $z \vee s$ and $z \vee a$ will be reminded (activated) by similarity in the next step, since both are equally excited (see the selection mechanism from Phase 2, Section 2). Hence, a joint concept $z \vee s \vee a$ will be established. In the next step, the association $z \vee s \vee a \rightarrow r$ will emerge for the first time.

This scenario is repeated a few times, with the latter association becoming repeatedly stronger. At some point, say at time t_k, it becomes so strong that its influence to the excitation of r is no longer negligible.

Thus, in the case of negative conditioning, when $a = n$, from this time on this association will prevent r from being activated, since it inherits the negative strength from a. However, in the case of positive conditioning this association will reinforce the activation of r.

Assume now that z is not active, and s appears at the input. In the next step, the activation of r by causality competes with the activation of $z \vee s \vee a$ by similarity. In this case we may safely suppose that the former activation will be selected by the selection mechanism. This is because we have assumed that the strength of the association $s \rightarrow r$ was great initially and therefore the excitation of r is greater than that of $z \vee s \vee a$.

□

From the previous theorem and from its proof the following interesting fact follows. When, at time $t_1 + 2$, the response r is activated and it is a correct one, confirmed by p, the cogitoid behaves like having already acquired the right behavior. But from the proof it is seen that this is not yet the case, since r is activated by the stimulus s alone, not yet taking p into account. In the case of an incorrect answer the cogitoid takes a proper lesson very fast and after a few exercises behaves "correctly". Note, however, that due to Phase 5, in the long run the cogitoid tends to forget what it has learned in this case, since the inhibitory connections are not strengthened any longer, because r is not invoked.

In practice this means that behaviour acknowledged by positive rewards looks like it is being acquired faster than the one acknowledged by negative rewards. Moreover, in the long run, if not rewarded, the inhibited behaviour tends to be forgotten. This seems to correspond well to our everyday experience — reinforcing good habits is easier than suppressing wrong ones.

3.4 Delayed Reinforcement

It appears that by a similar mechanism that ties a certain operant concept to some temporarily prevailing operant context one can also explain a more complicated case of the so–called *delayed reinforcing* when the reinforcing stimulus — a punishment or a reward — does not necessarily appear immediately after the step to be reinforced.

Thus, we shall deal with a problem when a punishment appears only after the cogitoid activates a previously acquired sequence of concepts in a way described in Theorem 3.2.

Theorem 3.7 *Let a sequence of non–meeting concepts $(x_1, x_2, \ldots, x_k)$, $k > 1$, in which each concept occurs exactly once, be acquired by a cogitoid $\mathcal{C}$. Let i be fixed, with $1 \leq i < k$, let z be an operant context that is active only during the activities of $x_{i+1}, \ldots, x_k$.*

If each activation of x_k subsequently invoked by the initial activation of x_1 (cf. Theorem 3.2) will be punished by activating a negative operant concept n, then, after this happens a few times, activation of x_1 will eventually cease to active x_k. Then, later on, when continuing in activating x_1, this activation will cease to activate both x_k and x_{k-1}, etc., until finally the activation of x_1 will activate only $x_2, \ldots, x_i$, but none of x_j, for $i < j \leq k$.

Sketch of the proof: The similar mechanism as in the Theorem 3.6, viz. the repeated punishment of x_k by n in the context z, will give rise to the negatively reinforcing association $z \vee n \vee x_{k-1} \rightarrow x_k$. Thus, under the circumstances at hand after several repetitions of the previous experience $\mathcal{C}$ ceases to activate x_k, but will still activate x_{k-1} (when $i < k - 1$). Nevertheless, the latter activation will then by resemblance activate the concept $z \vee n \vee x_{k-1}$ that has emerged previously. This concept will act like a punishment for the activation of x_{k-1}, and hence the negatively reinforcing association $z \vee n \vee x_{k-2} \rightarrow x_{k-1}$ will eventually emerge. After some time, it will start to inhibit the activation of x_{k-1}. The similar process repeats itself for x_{k-3}, etc., until it reaches x_i. Then, clearly, in accordance with Theorem 3.2 activation of x_1 will still activate the concepts $x_2, \ldots, x_i$. Nevertheless, unlike the preceeding ones, the latter concept, x_i, will be active simultaneously with the operant context z whose association with the negative operant concept n will cause that none of x_j, for $i < j \leq k$, will be activated. □

Based on the proof of the previous theorem it appears that cogitoids are also able to realize a strategy that is similar to the so–called *backtracking* technique used in searching for the accepting solutions in decision trees. In this case, at each potential branching point cogitoid must find itself in a specific operant context. When some sequence of concept activation presents a "blind alley" (i.e., leads to a punishment) in this context, this operant context will get associated with the negative operant concept much in the same way as in the proof of theorem 3.6. This will subsequently enable the cogitoid to return to the branching point and, eventually, to take an other path. However, we will abstain from stating the respective theorem since its proper formulation requires a lot of technical assumptions.

4 The Computational Power of Cogitoids

It is clear that in principle, not taking the efficiency into account, any cogitoid can be simulated by a Turing machine. We shall prove that also the reverse simulation is possible.

W.l.o.g. assume that the deterministic Turing machine to be simulated is a single tape machine. Its behaviour is governed by the transition function δ whose transitions take the form $(s_1, q_1) \rightarrow (s_2, q_2, d)$. Here s_1 is the symbol read by the Turing machine in state q_1, s_2 is the new symbol by which s_1 is rewritten, q_2 is the new state entered afterwards, and d is a direction of move of the machine head.

If a cogitoid has to simulate such a machine the role of its environment will be played by a similar tape as the machine has. The Turing machine head will fulfill the task of cogitoids sensors (reading the input) and effectors (writing the output and moving the head). The role of finite state control mechanism will be played by the cogitoid itself. For such a purpose the cogitoid has to learn "what to do" in what circumstance. This is described by the transition function δ. Thus, a simple learning scenario is to sequentially present all pairs $\{(s_1, q_1), (s_2, q_2, d)\}$ to the cogitoid. These pairs are seen as pairs of concepts, similarly as described in Theorem 3.1. This will give rise to associations of type as needed for cogitoid's behaviour we are after. The respective actions over the tape will then realize the transitions from one machine configuration to the next one. The fact that the respective concepts need not necessarily be non–meeting will cause the emergence of associations between similar concepts in the above mentioned pairs. However, this, at least for some time, will not be harmful to the intended simulation since the associations between input concepts are strengthened more than the other associations.

Thus, the simulation will work for a certain number of inputs until some rarely used transitions will get forgotten thanks to the mechanism of gradual forgetting. Another circumstance when the simulation could be lead astray will occur after infrequently used associations will get weakened to such an extent that the "parasitic" resemblance associations will take over. However, by a more sofisticated training that will aim at continuous strengthening of right associations (e.g. by recalling, and thus strengthening, all of them over and over again after realizing any simulation step) we can achieve a more stable simulation. Possible mechanisms for building such a habit are described in the forthcoming paper [11]

An other idea for simulation is to teach the cogitoid to simulate a universal Turing machine. Then the cogitoid equipped with a sufficient number of tapes and the respective sensors and effectors will be able to "trace" the computation of any Turing machine on any input providing that it gets both the input and the machine description to its disposal.

It is amusing to see the last results as the extension of Turing's original idea that has lead to the design of his model of computation machine. Namely, his machine can be seen as an abstraction of a knowledgeable, for a certain specific purpose trained, or programmed mathematician that does certain computations but is incapable to change his program. The role of this man is played by the machine's finite state control. Using a cogitoid in place of this finite state control device one can get a more flexible interactive machine that be be taught and whose behaviour thus can mimic more closely the cognitive behaviour of animals or men.

5 Conclusions and Open Problems

The contribution of this paper is twofold. First, a new formal finite model of mind has been introduced. It presents a deviation from similar models that are based on paradigms of neurocomputing. This is because it abstracts from brain–like implementations and instead focuses on the substance of mental processes — viz. the concept and association calculus. Secondly, the viability of this model has been demonstrated by its capability to exactly formulate the instances of Pavlovian and operant conditioning and to give their plausible algorithmic explanation. This seems to be the first occasion when a computational justification of the related mental phenomena, in a form usual in computer science, is given.

The result on "behavioral" equivalence of Turing machines and cogitoids equipped with the external memory is of independent interest.

Of course, the underlying emerging theory of cogitoids has a lot of open ends. First of all, the versatility of the model must by further verified on other cognitive tasks described in experimental psychology. This will probably lead to further tuning of the model. But already now, in its present form the model seems to offer an interesting and promising alternative for studying the computational aspects of mental processes. It opens ways for its possible computer simulation. It appears that with the help of this model algorithmic explanation of higher brain functions (such as the emergence of habits, concept of self, language acquisition and generation, thinking, consciousness emergence, etc.) could be also possible. Such possibilities are subject of the author's current research in this field [11]. The work by [4] or [5] can serve as a vital source of inspiration along these lines.

Bibliography

[1] Arbib, M. A. (Editor): The Handbook of Brain Theory and Neural Networks. The MIT Press, Cambridge — Massachusetts, London, England, 1995, 1118 p.

[2] Birkhoff, G.: Lattice Theory. American Mathematical Society, New York, 1948

[3] Casti, J.L.: Paradigms Lost. Avon Books, New York, 1989, 565 p.

[4] Dennet, D.C.: Consciousness Explained. Penguin Books, 1991, 511 p.

[5] Goldschlager, L.G.: A Computational Theory of Higher Brain Function. Technical Report 233, April 1984, Basser Department of Computer Science, The University of Sydney, Australia, ISBN 0 909798 91 5

[6] Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, Inc., New York, 1994, 696 p.

[7] Valiant, L.G.: Circuits of the Mind. Oxford University Press, New York, Oxford, 1994, 237 p., ISBN 0–19–508936–X

[8] Wiedermann, J.: Towards Computational Models of the Brain: Getting Started. Neural Network World, No.1, 1997, pp. 89–120

[9] Wiedermann, J.: The Cogitoid: A Computational Model of Mind. Technical Report No. V–685, Institute of Computer Science AS ČR, Prague, September 1996

[10] Wiedermann, J.: Towards Machines That Can Think (Invited Talk). In: Proceeding of the 24–th Seminar on Current Trends in Theory and Practice of Informatics SOFSEM'97, LNCS Vol. 1338, Springer Verlag, Berlin, 1997, pp.12–141

[11] Towards Algorithmic explanation of Mind Evolution and Functioning. Technical report in preparation, 1998

[12] Wille, R.: Concept Lattices and Conceptual Knowledge Systems. *Computers Math. Applic.* Vol. 23, No. 6–9, 1992, pp. 493–515