Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 356029, 27 pages
doi:10.1155/2010/356029

Research Article

Associative Models for Storing and Retrieving Concept Lattices

María Elena Acevedo,[1] Cornelio Yáñez-Márquez,[2] and Marco Antonio Acevedo[1]

[1] *Department of Communications and Electronic Engineering at Superior School of Mechanical and Electrical Engineering, National Polytechnic Institute, Avenue IPN s/n, Col. Lindavista, C.P. 07738 Mexico City, Mexico*
[2] *Artificial Intelligence Laboratory, Computation Research Center, National Polytechnic Institute, Avenue Juan de Dios Bátiz s/n, C.P. 07738 Mexico City, Mexico*

Correspondence should be addressed to María Elena Acevedo, eacevedo@ipn.mx

Received 17 March 2010; Accepted 21 June 2010

Academic Editor: Wei-Chiang Hong

Alpha-beta bidirectional associative memories are implemented for storing concept lattices. We use Lindig's algorithm to construct a concept lattice of a particular context; this structure is stored into an associative memory just as a human being does, namely, associating patterns. Bidirectionality and perfect recall of Alpha-Beta associative model make it a great tool to store a concept lattice. In the learning phase, objects and attributes obtained from Lindig's algorithm are associated by Alpha-Beta bidirectional associative memory; in this phase the data is stored. In the recalling phase, the associative model allows to retrieve objects from attributes or vice versa. Our model assures the recalling of every learnt concept.

1. Introduction

Concept Lattices is the common name for a specialized form of Hasse diagrams [1] that is used in conceptual data processing. Concept Lattices are a principled way of representing and visualizing the structure of symbolic data that emerged from Rudolf Wille efforts to restructure lattice and order theory in the 1980s. Conceptual data processing, also known as Formal Concept Analysis, has become a standard technique in data and knowledge processing that has given rise to applications in data visualization, data mining, information retrieval (using ontologies), and knowledge management. Organization of discovered concepts in the form of a lattice-structure has many advantages from the perspective of knowledge discovery. It facilitates insights into dependencies among different concepts mined from a dataset. Lattices of concepts have been implemented with a number of different algorithms [2–7]. Any of them can generate a very large number of concepts; therefore, a suitable method is required for an efficient storage and retrieval of parts of the lattice. The task of efficiently organizing and retrieving various nodes of a lattice is the focus of this work. A concept is a pair that consists of a set of objects and a particular set of attribute values shared by the objects. From an initial table, with rows representing the objects and columns representing the attributes, a concept lattice can be obtained. From this structure, we can retrieve an object from the attribute or vice versa, and these pairs form a concept.

The main goal of an Associative Memory is to associate pairs of patterns for recalling one pattern presenting its corresponding pattern; the recalling is done in one direction only. In the particular case of Bidirectional Associative Memories (BAM), we can recall any of the two patterns belonging to a pair just presenting one of them; therefore, the recalling is in both directions. This behavior allows BAM to be a suitable tool for storing and retrieving concepts which form a particular lattice concept. The first step for achieving this task is to apply any of the existing algorithms to obtain the lattice concept; in this work, we use the Linding's algorithm [5]; then we store each node (concept) associating the objects and attributes forming that concept. Once we stored all concepts, we are able to retrieve them by presenting an object or an attribute. The model of BAM used here is the Alpha-Beta Bidirectional Associative Memory [8]. The main reason for using this model is because it presents perfect recall of the training set; this means that it can recall every pair of patterns that it associated, no matter the size of the patterns or the number of these. This advantage is not presented by other BAM models which present stability and convergence problems or limit their use for a particular number of patterns or to the nature of them, such as, Hamming distance or linear dependency [9–19].

In Section 2, we present a brief discussion on Formal Context Analysis. In Section 3, we introduce the basic concepts of Associative Models, in particular the Alpha-Beta Model, because it is the base of Alpha-Beta BAM. Then, we present the theoretical foundations of our associative model which assure the perfect recall of the training set of patterns with no limits in the number or nature of patterns. We describe the software that implements our algorithm in Section 4 and we show an example.

2. Formal Concept Analysis

Formal Concept Analysis (FCA) was first proposed by Wille in 1982 [20] as a mathematical framework for performing data analysis. It provides a conceptual analytical tool for investigating and processing given information explicitly [21]. Such data is structured into units, which are formal abstractions of "*concepts*" of human thought allowing meaningful and comprehensible interpretation. FCA models the world as being composed of *objects* and *attributes*. It is assumed that an incident relation connects objects to attributes. The choice of what is an object and what is an attribute is dependent on the domain in which FCA is applied. Information about a domain is captured in a "formal context". A formal context is merely a formalization that encodes only a small portion of what is usually referred to as a "*context*". The following definition is crucial to the theory of FCA.

Definition 2.1. A formal context $K = (G, M, I)$ is a triplet consisting of two sets G (set of objects) and M (set of attributes) and a relation I between G and M.

Definition 2.2. A formal concept in a formal context is a pair (A, B) of sets $A \subseteq G$ and $B \subseteq M$ such that $A \uparrow= B$ and $B \downarrow= A$ (completeness constraint), where $A \uparrow= \{m \in M \mid g\,\mathrm{Im}$ for all $g \in A\}$ (i.e., the set of attributes common to all the objects in A), and $B \downarrow= \{g \in G \mid g\,\mathrm{Im}$ for all $m \in B\}$ (i.e., the set of objects that have all attributes in B). By $g\,\mathrm{Im}$ we denote the fact that object g has attribute m.

The set of all concepts of a context (G, M, I) is denoted by $\mathbf{B}(G, M, I)$. This consists of all pairs $(A^c \psi B)$ such that $A_\uparrow \longleftarrow \mathfrak{I} \longleftarrow B$ and $B_\downarrow \longleftarrow \mathfrak{I} \longleftarrow A$, where $A \subseteq G$ and $B \subseteq M$.

Definition 2.3. Specificity-generality order relationship. If (A_1, B_1) and (A_2, B_2) are concepts of a context, then (A_1, B_1) is called a subconcept of (A_2, B_2) if $A_1 \subseteq A_2$ (or equivalently $B_1 \supseteq B_2$). This sub-super concept relation is written as $(A_1, B_1) \leq (A_2, B_2)$. According to this definition, a subconcept always contains fewer objects and greater attributes than any of its super concepts.

2.1. Concept Lattice

A set of all concepts of the context (G, M, I) (denoted by $\mathbf{B}(G, M, I)$) when ordered with the order relation $\leq$ (a subsumption relation) defined above forms a *concept lattice* of the context and is denoted by $\mathbf{B}(G, M, I)$.

A lattice is an ordered set V with an order relation in which for any given two elements x and y, the *supremum* and the *infimum* elements always exist in V. Furthermore, such a lattice is called a *complete lattice* if supremum and infimum elements exist for any subset X of V. The fundamental theorem of FCA states that the set of formal concepts of a formal context forms a complete lattice.

This complete lattice, which is composed by *formal concepts*, is called a *concept lattice*.

A Concept lattice can be visualized as a graph with nodes and edges/links. The concepts at the nodes from which two or more lines run up are called *meet* concepts (i.e., nodes with more than one parent) and the concepts at the nodes from which two or more lines run down are called *join* concepts (i.e., nodes with more than one child).

A *join concept* groups objects which share the same attributes and a *meet concept* separates out objects that have combined attributes from different parents (groups of objects). Each of these join and meet concepts creates a new sub- or super-category or class of a concept.

3. Alpha-Beta Bidirectional Associative Memories

In this section, the Alpha-Beta Bidirectional Associative Memory is presented. However, since it is based on the Alpha-Beta autoassociative memories, a summary of this model will be given before presenting our model of BAM.

3.1. Basic Concepts

Basic concepts about associative memories were established three decades ago in [22–24]; nonetheless, here we use the concepts, results, and notation introduced in [25]. An associative memory $\mathbf{M}$ is a system that relates input patterns and outputs patterns, as follows: $\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y}$ with $\mathbf{x}$ and $\mathbf{y}$ the input and output pattern vectors, respectively. Each input vector forms an

Table 1: Alpha operator. $\alpha : A \times A \to B$.

x	y	$\alpha(x,y)$
0	0	1
0	1	0
1	0	2
1	1	1

Table 2: Beta operator. $\beta : B \times A \to A$.

x	y	$\beta(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

association with a corresponding output vector. For k integer and positive, the corresponding association will be denoted as (x^k, y^k). Associative memory $\mathbf{M}$ is represented by a matrix whose ijth component is m_{ij}. Memory $\mathbf{M}$ is generated from an a priori finite set of known associations, known as the fundamental set of associations.

If μ is an index, the fundamental set is represented as $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \ldots, p\}$ with p being the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu$, for all $\mu \in \{1, 2, \ldots, p\}$, $\mathbf{M}$ is *autoassociative*, otherwise it is *heteroassociative*; in this case, it is possible to establish that $\exists \mu \in \{1, 2, \ldots, p\}$ for which $x^\mu \neq y^\mu$. A distorted version of a pattern x^k to be recovered will be denoted as $\tilde{x}^k$. If when feeding a distorted version of x^ϖ with $\varpi = \{1, 2, \ldots, p\}$ to an associative memory $\mathbf{M}$, it happens that the output corresponds exactly to the associated pattern y^ϖ, we say that recall is perfect.

3.2. Alpha-Beta Associative Memories

Among the variety of associative memory models described in the scientific literature, there are two models that, because of their relevance, it is important to emphasize: morphological associative memories which were introduced by Ritter et al. [18] and Alpha-Beta associative memories. Because of their excellent characteristics, which allow them to be superior in many aspects to other models for associative memories, morphological associative memories served as a starter point for the creation and development of the Alpha-Beta associative memories.

The Alpha-Beta associative memories [25] are of two kinds and are able to operate in two different modes. The operator α is useful at the learning phase, and the operator β is the basis for the pattern recall phase. The heart of the mathematical tools used in the Alpha-Beta model is two binary operators designed specifically for this model. These operators are defined as follows: first, we have the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$; then the operators α and β are defined in Tables 1 and 2, respectively.

The sets A and B, the α and β operators, along with the usual $\wedge$ (minimum) and $\vee$ (maximum) operators, form the algebraic system $(A, B, \alpha, \beta, \wedge, \vee)$ which is the mathematical basis for the Alpha-Beta associative memories. Below are shown some characteristics of Alpha-Beta autoassociative memories.

(1) The fundamental set takes the form $\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, \ldots, p\}$.

(2) Both input and output fundamental patterns are of the same dimension, denoted by n.

(3) The memory is a square matrix, for both modes, $\mathbf{V}$ and $\mathbf{\Lambda}$. If $\mathbf{x}^\mu \in A^n$, then

$$\begin{aligned} v_{ij} &= \bigvee_{\mu=1}^{p} \alpha\left(x_i^\mu, x_j^\mu\right), \\ \lambda_{ij} &= \bigwedge_{\mu=1}^{p} \alpha\left(x_i^\mu, x_j^\mu\right) \end{aligned} \tag{3.1}$$

And according to $\alpha : A \times A \rightarrow B$, we have that v_{ij} and $\lambda_{ij} \in B$, for all $i \in \{1, 2, \ldots, n\}$ and for all $j \in \{1, 2, \ldots, n\}$.

In the recall phase, when a pattern $\mathbf{x}^\mu$ is presented to memories $\mathbf{V}$ and $\mathbf{\Lambda}$, the ith components of recalled patterns are

$$\begin{aligned} \left(\mathbf{V}\Delta_\beta \mathbf{x}^\omega\right)_i &= \bigwedge_{j=1}^{n} \beta\left(v_{ij}, x_j^\omega\right), \\ \left(\mathbf{\Lambda}\nabla_\beta \mathbf{x}^\omega\right)_i &= \bigvee_{j=1}^{n} \beta\left(\lambda_{ij}, x_j^\omega\right). \end{aligned} \tag{3.2}$$

The next two theorems show that Alfa-Beta autoassociative memories max and min are immune to certain amount of additive and subtractive noise, respectively. These theorems have the original numbering presented in [25] and are an important part of the mathematical foundations for Alfa-Beta BAM theory.

Theorem 3.1. *Let $\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, \ldots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type $\vee$ represented by V, and let $\widetilde{\mathbf{x}} \in A^n$ be a pattern altered with additive noise with respect to some fundamental pattern $\mathbf{x}^\omega$, with $\omega \in \{1, 2, \ldots, p\}$. If $\widetilde{\mathbf{x}}$ is presented to V as input, and also for every $i \in \{1, \ldots, n\}$ it holds that $\exists j = j_0 \in \{1, \ldots, n\}$, which is dependent on ω and i such that $v_{ij_0} \leq \alpha(x^\omega, \widetilde{x}_{j_0})$, then recall $\mathbf{V}\Delta_\beta\widetilde{\mathbf{x}}$ is perfect; that is, to say that $\mathbf{V}\Delta_\beta\widetilde{\mathbf{x}} = \mathbf{x}^\omega$.*

Theorem 3.2. *Let $\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, \ldots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type $\wedge$ represented by Λ, and let $\widetilde{\mathbf{x}} \in A^n$ be a pattern altered with subtractive noise with respect to some fundamental pattern $\mathbf{x}^\omega$, with $\omega \in \{1, 2, \ldots, p\}$. If $\widetilde{\mathbf{x}}$ is presented to memory Λ as input, and also for every $i \in \{1, \ldots, n\}$ it holds that $\exists j = j_0 \in \{1, \ldots, n\}$, which is dependent on ω and i, such that $\lambda_{ij_0} \leq \alpha(x^\omega, \widetilde{x}_{j_0})$, then recall $\Lambda\nabla_\beta\widetilde{\mathbf{x}}$ is perfect; that is, to say that $\Lambda\nabla_\beta\widetilde{\mathbf{x}} = \mathbf{x}^\omega$.*

With these bases we proceed to describe Alfa-Beta BAM model.

Figure 1: General scheme of a Bidirectional Associative Memory.

3.3. Alpha-Beta Bidirectional Associative Memories

Usually, any bidirectional associative memory model appearing in current scientific literature has the following scheme showed in Figure 1.

A BAM is a "black box" operating in the following way: given a pattern $\mathbf{x}$, associated pattern $\mathbf{y}$ is obtained, and given the pattern $\mathbf{y}$, associated pattern $\mathbf{x}$ is recalled. Besides, if we assume that $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are noisy versions of $\mathbf{x}$ and $\mathbf{y}$, respectively, it is expected that BAM could recover all corresponding free noise patterns $\mathbf{x}$ and $\mathbf{y}$.

The first bidirectional associative memory (BAM), introduced by Kosko [26], was the base of many models presented later. Some of these models substituted the learning rule for an exponential rule [9–11]; others used the method of multiple training and dummy addition in order to reach a greater number of stable states [12], trying to eliminate spurious states. With the same purpose, linear programming techniques [13] and the descending gradient method [14, 15] have been used, besides genetic algorithms [16] and BAM with delays [17, 27]. Other models of noniterative bidirectional associative memories exist, such as, morphological BAM [18] and Feedforward BAM [19]. All these models have arisen to solve the problem of low pattern recall capacity shown by the BAM of Kosko; however, none has been able to recall all the trained patterns. Also, these models demand the fulfillment of some specific conditions, such as a certain Hamming distance between patterns, solvability by linear programming, orthogonality between patterns, among other.

The model of bidirectional associative memory presented in this paper is Alpha-Beta BAM [28] and is based on the Alpha-Beta associative memories [25]; it is not an iterative process and does not present stability problems. Pattern recall capacity of the Alpha-Beta BAM is maximal, being $2^{\min(n,m)}$, where m and n are the input and output patterns dimension, respectively. Also, it always shows perfect pattern recall without imposing any condition.

The model used in this paper has been named Alpha-Beta BAM since Alpha-Beta associative memories, both max and min, play a central role in the model design. However, before going into detail over the processing of an Alpha-Beta BAM, we will define the following.

In this work we will assume that Alpha-Beta associative memories have a fundamental set denoted by $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \ldots, p\}$, $\mathbf{x}^\mu \in A^n$ and $\mathbf{y}^\mu \in A^m$, with $A = \{0, 1\}$, $n \in \mathbf{Z}^+$, $p \in \mathbf{Z}^+$, $m \in \mathbf{Z}^+$ and $1 < p \leq \min(2^n, 2^m)$. Also, it holds that all input patterns are different; M that is, $\mathbf{x}^\mu = \mathbf{x}^\xi$ if and only if $\mu = \xi$. If for all $\mu \in \{1, 2, \ldots, p\}$ it holds that $\mathbf{x}^\mu = \mathbf{y}^\mu$, the Alpha-Beta memory will be *autoassociative*; if on the contrary, the former affirmation is negative, that is, $\exists \mu \in \{1, 2, \ldots, p\}$ for which it holds that $\mathbf{x}^\mu \neq \mathbf{y}^\mu$, then the Alpha-Beta memory will be *heteroassociative*.

Definition 3.3 (One-Hot). Let the set Abe $A = \{0, 1\}$ and $p \in \mathbf{Z}^+$, $p > 1$, $k \in \mathbf{Z}^+$, such that $1 \leq k \leq p$. The kth one-hot vector of p bits is defined as vector $h^k \in A^p$ for which it holds that the kth component is $h_k^k = 1$ and the set of the components are $h_j^k = 0$, for all $j \neq k$, $1 \leq j \leq p$.

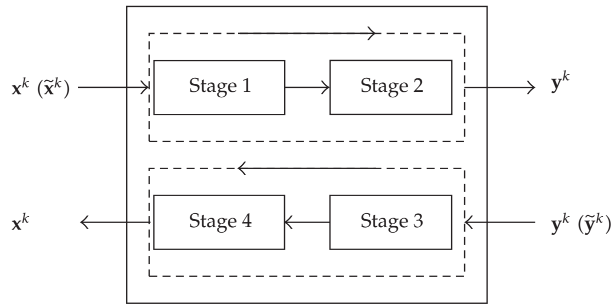

Figure 2: Alpha-Beta BAM model scheme.

Remark 3.4. In this definition, the value $p = 1$ is excluded since a one-hot vector of dimension 1, given its essence, has no reason to be.

Definition 3.5 (Zero-Hot). Let the set A be $A = \{0, 1\}$ and $p \in \mathbf{Z}^+$, $p > 1$, $k \in \mathbf{Z}^+$, such that $1 \leq k \leq p$. The kth zero-hot vector of p bits is defined as vector $\overline{\mathbf{h}}^k \in A^p$ for which it holds that the kth component is $h_k^k = 0$ and the set of the components are $h_j^k = 1$, for all $j \neq k$, $1 \leq j \leq p$.

Remark 3.6. In this definition, the value $p = 1$ is excluded since a zero-hot vector of dimension 1, given its essence, has no reason to be.

Definition 3.7 (Expansion vectorial transform). Let the set A be $A = \{0, 1\}$ and $n \in \mathbf{Z}^+$, $ym \in \mathbf{Z}^+$. Given two arbitrary vectors $\mathbf{x} \in A^n$ and $\mathbf{e} \in A^m$, the expansion vectorial transform of order m, $\tau^e : A^n \to A^{n+m}$, is defined as $\tau^e(\mathbf{x}, \mathbf{e}) = \mathbf{X} \in A^{n+m}$, a vector whose components are $X_i = x_i$ for $1 \leq i \leq n$ and $X_i = e_i$ for $n + 1 \leq i \leq n + m$.

Definition 3.8 (Contraction vectorial transform). Let the set A be $A = \{0, 1\}$ and $n \in \mathbf{Z}^+$, $ym \in \mathbf{Z}^+$ such that $1 \leq m < n$. Given one arbitrary vector $\mathbf{X} \in A^{n+m}$, the contraction vectorial transform of order m, $\tau^c : A^{n+m} \to A^m$, is defined as $\tau^c(\mathbf{X}, m) = \mathbf{c} \in A^m$, a vector whose components are $c_i = X_{i+n}$ *for* $1 \leq i < m$.

In both directions, the model is made up by two stages, as shown in Figure 2.

For simplicity, first will be described the process necessary in one direction, in order to later present the complementary direction which will give bidirectionality to the model (see Figure 3).

The function of Stage 2 is to offer a $\mathbf{y}^k$ as output $(k = 1, \dots, p)$ given an $\mathbf{x}^k$ as input.

Now we assume that as input to Stage 2 we have one element of a set of p orthonormal vectors. Recall that the *Linear Associator* has perfect recall when it works with orthonormal vectors. In this work, we use a variation of the *Linear Associator in order to obtain* $\mathbf{y}^k$, parting from a *one-hot* vector $\mathbf{h}^k$ in its kth coordinate.

For the construction of the modified Linear Associator, its learning phase is skipped and a matrix $\mathbf{M}$ representing the memory is built. Each column in this matrix corresponds to each output pattern $\mathbf{y}^\mu$. In this way, when matrix $\mathbf{M}$ is operated with a one-hot vector $\mathbf{h}^k$, the corresponding $\mathbf{y}^k$ will always be recalled.

The function of Stage 2 is to offer a $\mathbf{y}^k$ as output $(k = 1, \dots, p)$ given an $\mathbf{x}^k$ as input.

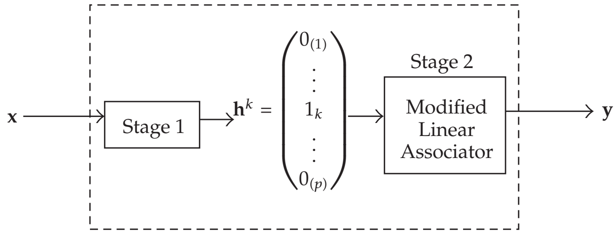

Figure 3: Schematics of the process done in the direction from $\mathbf{x}$ to $\mathbf{y}$. Here, only Stage 1 and Stage 2 are shown. Notice that $h_k{}^k = 1$, $v_i{}^k = 1$ for all $i \neq k$, $1 \leq i \leq p$, $1 \leq k \leq p$.

3.3.1. *Theoretical Foundation of Stages 1 and 3*

Below are presented 5 theorems and 9 lemmas with their respective proofs, as well as an illustrative example of each one. This mathematical foundation is the basis for the steps required by the complete algorithm, which is presented in Section 3.3.2. These theorems and lemmas numbering corresponds to the numeration used in [23].

By convention, the symbol □ will be used to indicate the end of a proof.

Theorem 3.9. *Let $\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1, 2, \ldots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by $\mathbf{V}$, and let $\widetilde{\mathbf{x}} \in A^n$ be a pattern altered with additive noise with respect to some fundamental pattern $\mathbf{x}^\omega$ with $\omega \in \{1, 2, \ldots, p\}$. Let us assume that during the recalling phase, $\widetilde{\mathbf{x}}$ is presented to memory $\mathbf{V}$ as input, and let us consider an index $k \in \{1, \ldots, n\}$. The kth component recalled $(\mathbf{V}\Delta_\beta \widetilde{\mathbf{x}})_k$ is precisely x_k^ω if and only if it holds that $\exists r \in \{1, \ldots, n\}$, dependant on ω and k, such that $\nu_{kr} \leq \alpha(x_k^\omega, \widetilde{x}_r)$.*

Proof. $\Rightarrow$) By hypothesis we assume that $(\mathbf{V}\Delta_\beta \widetilde{\mathbf{x}})_k = x_k^\omega$. By contradiction, now suppose false that $\exists r \in \{1, \ldots, n\}$ such that $\nu_{kr} \leq \alpha(x_k^\omega, \widetilde{x}_r)$. The former is equivalent to stating that for all $r \in \{1, \ldots, n\}$, $\nu_{kr} > \alpha(x_k^\omega, \widetilde{x}_r)$, which is the same to saying that for all $r \in \{1, \ldots, n\}, \beta(\nu_{kr}, \widetilde{x}_r) > \beta[\alpha(x_k^\omega, \widetilde{x}_r), \widetilde{x}_r] = x_k^\omega$. When we take minimums at both sides of the inequality with respect to index r, we have

$$\bigwedge_{r=1}^{n} \beta(\nu_{kr}, \widetilde{x}_r) > \bigwedge_{r=1}^{n} x_k^\omega = x_k^\omega \tag{3.3}$$

and this means that $(\mathbf{V}\Delta_\beta \widetilde{\mathbf{x}})_k = \bigwedge_{r=1}^{n} \beta(\nu_{kr}, \widetilde{x}_r) > x_k^\omega$, which contradicts the hypothesis.

$\Leftarrow$) Since the conditions of Theorem 3.1 hold for every $i \in \{1, \ldots, n\}$, we have that $\mathbf{V}\Delta_\beta \widetilde{\mathbf{x}} = \mathbf{x}^\omega$; that is, it holds that $(\mathbf{V}\Delta_\beta \widetilde{\mathbf{x}})_i = x_i^\omega$, for all $i \in \{1, \ldots, n\}$. When we fix indexes i and j_0 such that $i = k y j_0 = r$ (which depends on ω and k), we obtain the desired result: $(\mathbf{V}\Delta_\beta \widetilde{\mathbf{x}})_k = x_k^\omega$. □

Lemma 3.10. *Let $\{(\mathbf{X}^k, \mathbf{X}^k) \mid k = 1, \ldots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by $\mathbf{V}$, with $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ for $k = 1, \ldots, p$, and let $\mathbf{F} = \tau^e(\mathbf{x}^k, \mathbf{u}) \in A^{n+p}$ be a version of a specific pattern $\mathbf{X}^k$, altered with additive noise, being $\mathbf{u} \in A^p$ the vector defined as $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$. If during the recalling phase $\mathbf{F}$is presented to memory $\mathbf{V}$, then component X_{n+k}^kwill be recalled in a perfect manner; that is, $(\mathbf{V}\Delta_\beta \mathbf{F})_{n+k} = X_{n+k}^k = 1$.*

Proof. This proof will be done for two mutually exclusive cases.

Case 1. Pattern $\mathbf{F}$ has one component with value 0. This means that $\exists j \in \{1,\ldots,n+p\}$ such that $F_j = 0$; also, due to the way vector $\mathbf{X}^k$ is built, it is clear that $X^k_{n+k} = 1$. Then $\alpha(X^k_{n+k}, F_j) = \alpha(1,0) = 2$, and since the maximum allowed value for a component of memory $\mathbf{V}$ is 2, we have $\nu_{(n+k)j} \leq \alpha(X^k_{n+k}, F_j)$. According to Theorem 3.9, X^k_{n+k} is perfectly recalled.

Case 2. Pattern $\mathbf{F}$ does not contain a component with value 0. That is, $F_j = 1$ for all $j \in \{1,\ldots,n+p\}$. This means that it is not possible to guarantee the existence of a value $j \in \{1,\ldots,n+p\}$ such that $\nu_{(n+k)j} \leq \alpha(X^k_{n+k}, F_j)$, and therefore Theorem 3.9 cannot be applied. However, we will show the impossibility of $(\mathbf{V}\Delta_\beta\mathbf{F})_{n+k} = 0$. The recalling phase of the autoassociative Alpha-Beta memory of type max $\mathbf{V}$, when having vector $\mathbf{F}$ as input, takes the following form for the $(n+k)$th recalled component:

$$(\mathbf{V}\Delta_\beta\mathbf{F})_{n+k} = \bigwedge_{j=1}^{n} \beta\left(\nu_{(n+k)j}, F_j\right) = \bigwedge_{j=1}^{n} \beta\left\{\left[\bigvee_{\mu=1}^{p} \alpha\left(X^\mu_{n+k}, X^\mu_j\right)\right], F_j\right\}. \tag{3.4}$$

Due to the way vector $\mathbf{X}^k$ is built, besides $X^k_{n+k} = 1$, it is important to notice that $X^\mu_{n+k} \neq 1$, for all $\mu \neq k$, and from here we can establish that the following

$$\bigvee_{\mu=1}^{p} \alpha\left(X^\mu_{n+k}, X^\mu_j\right) = \alpha\left(X^k_{n+k}, X^k_j\right) = \alpha\left(1, X^k_j\right) \tag{3.5}$$

is different from zero regardless of the value of X^k_j. According to $F_j = 1$ for all $j \in \{1,\ldots,n+p\}$, we can conclude the impossibility of

$$(\mathbf{V}\Delta_\beta\mathbf{F})_{n+k} = \bigwedge_{j=1}^{n} \beta\left(\alpha\left(1, X^k_j\right), 1\right) \tag{3.6}$$

being zero. That is, $(\mathbf{V}\Delta_\beta\mathbf{F})_{n+k} = 1 = X^k_{n+k}$. □

Theorem 3.11. *Let $\{(\mathbf{X}^k, \mathbf{X}^k) \mid k = 1,\ldots,p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by $\mathbf{V}$, with $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ for $k = 1,\ldots,p$, and let $\mathbf{F} = \tau^e(\mathbf{x}^k, \mathbf{u}) \in A^{n+p}$ be a pattern altered with additive noise with respect to some specific pattern $\mathbf{X}^k$, with $\mathbf{u} \in A^p$ being the vector defined as $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$. Let us assume that during the recalling phase, $\mathbf{F}$ is presented to memory $\mathbf{V}$ as input, and the pattern $\mathbf{R} = \mathbf{V}\boldsymbol{\Delta}_\beta\mathbf{F} \in A^{n+p}$ is obtained. If when taking vector $\mathbf{R}$ as argument, the contraction vectorial transform $\mathbf{r} = \tau^c(\mathbf{R}, n) \in A^p$ is done, the resulting vector $\mathbf{r}$ has two mutually exclusive possibilities: $\exists k \in \{1,\ldots,p\}$ such that $\mathbf{r} = \mathbf{h}^k$, or $\mathbf{r}$ is not a one-hot vector.*

Proof. From the definition of contraction vectorial transform, we have that $r_i = R_{i+n} = (\mathbf{V}\boldsymbol{\Delta}_\beta\mathbf{F})_{i+n}$ for $1 \leq i \leq p$, and in particular, by making $i = k$ we have $r_k = R_{k+n} = (\mathbf{V}\boldsymbol{\Delta}_\beta\mathbf{F})_{k+n}$. However, by Lemma 3.10 $(\mathbf{V}\Delta_\beta\mathbf{F})_{n+k} = X^k_{n+k}$, and since $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$, the value X^k_{n+k} is equal to the value of component $h^k_k = 1$. That is, $r_k = 1$. When considering that $r_k = 1$, vector $\mathbf{r}$ has two mutually exclusive possibilities: it can be that $r_j = 0$ for all $j \neq k$ in which case $\mathbf{r} = \mathbf{h}^k$, or happens that $\exists j \in \{1,\ldots,p\}$, $j \neq k$ for which $r_j = 1$, in which case it is not possible that $\mathbf{r}$ is a one-hot vector, given Definition 3.3. □

Theorem 3.12. *Let $\{(\mathbf{x}^\mu, \mathbf{x}^\mu) \mid \mu = 1,2,\ldots,p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by $\boldsymbol{\Lambda}$, and let $\tilde{\mathbf{x}} \in A^n$ be a pattern altered with subtractive noise with respect to some fundamental pattern x^ω with $\omega \in \{1,2,\ldots,p\}$. Let us assume that during the recalling phase, x^ω is presented to memory $\boldsymbol{\Lambda}$ as input, and consider an index $k \in \{1,\ldots,n\}$. The kth recalled component $(\boldsymbol{\Lambda}\nabla_\beta\tilde{\mathbf{x}})_k$ is precisely x_k^ω if and only if it holds that $\exists r \in \{1,\ldots,n\}$, dependant on $\omega$and k, such that $\lambda_{kr} \geq \alpha(x_k^\omega, \tilde{x}_r)$.*

Proof. ⇒) By hypothesis, it is assumed that $(\boldsymbol{\Lambda}\nabla_\beta\tilde{\mathbf{x}})_k = x_k^\omega$. By contradiction, now let suppose it is false that $\exists r \in \{1,\ldots,n\}$ such that $\lambda_{kr} \geq \alpha(x_k^\omega, \tilde{x}_r)$. That is to say that for all $r \in \{1,\ldots,n\}$, $\lambda_{kr} < \alpha(x_k^\omega, \tilde{x}_r)$, which is in turn equivalent to for all $r \in \{1,\ldots,n\}$, $\beta(\lambda_{kr}, \tilde{x}_r) < \beta[\alpha(x_k^\omega, \tilde{x}_r), \tilde{x}_r] = x_k^\omega$. When taking the maximums at both sides of the inequality, with respect to index r, we have

$$\bigvee_{r=1}^{n} \beta(\lambda_{kr}, \tilde{x}_r) < \bigvee_{r=1}^{n} x_k^\omega = x_k^\omega \tag{3.7}$$

and this means that $(\boldsymbol{\Lambda}\nabla_\beta\tilde{\mathbf{x}})_k = \bigvee_{r=1}^{n}\beta(\lambda_{kr}, \tilde{x}_r) < x_k^\omega$, an affirmation which contradicts the hypothesis.

⇐) When conditions for Theorem 3.2 [19] are met for every $i \in \{1,\ldots,n\}$, we have $\boldsymbol{\Lambda}\nabla_\beta\tilde{\mathbf{x}} = \mathbf{x}^\omega$. That is, it holds that $(\boldsymbol{\Lambda}\nabla_\beta\tilde{\mathbf{x}})_i = x_i^\omega$ for all $i \in \{1,\ldots,n\}$. When indexes i and j_0 are fixed such that $i = k$ and $j_0 = r$, depending on ω and k, we obtain the desired result $(\boldsymbol{\Lambda}\nabla_\beta\tilde{\mathbf{x}})_k = x_k^\omega$. □

Lemma 3.13. *Let $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\ldots,p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by $\boldsymbol{\Lambda}$, with $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$ for $k = 1,\ldots,p$, and let $\mathbf{G} = \tau^e(\mathbf{x}^k, \mathbf{w}) \in A^{n+p}$ be a pattern altered with subtractive noise with respect to some specific pattern $\mathbf{X}^k$, being $\mathbf{w} \in A^p$a vector whose components have values $w_i = u_i - 1$, and $\mathbf{u} \in A^p$ the vector defined as $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$. If during the recalling phase, $\mathbf{G}$ is presented to memory $\boldsymbol{\Lambda}$, then component $\overline{X}_{n+k}^k$ is recalled in a perfect manner. That is, $(\boldsymbol{\Lambda}\nabla_\beta\mathbf{G})_{n+k} = \overline{X}_{n+k}^k = 0$.*

Proof. This proof will be done for two mutually exclusive cases.

Case 1. Pattern $\mathbf{G}$ has one component with value 1. This means that $\exists j \in \{1,\ldots,n+p\}$ such that $G_j = 1$. Also, due to the way vector $\overline{\mathbf{X}}^k$ is built, it is clear that $\overline{X}_{n+k}^k = 0$. Because of this, $\alpha(\overline{X}_{n+k}^k, G_j) = \alpha(0,1) = 0$ and, since the minimum allowed value for a component of memory $\boldsymbol{\Lambda}$ is 0, we have $\lambda_{(n+k)j} \geq \alpha(\overline{X}_{n+k}^k, G_j)$. According to Theorem 3.12, $\overline{X}_{n+k}^k$ is perfectly recalled.

Case 2. Pattern $\mathbf{G}$ has no component with value 1; that is, $G_j = 0$ for all $j \in \{1,\ldots,n+p\}$. This means that it is not possible to guarantee the existence of a value $j \in \{1,\ldots,n+p\}$ such that $\lambda_{(n+k)\ j} \geq \alpha(\overline{X}_{n+k}^k, G_j)$, and therefore Theorem 3.12 cannot be applied. However, let us show the impossibility of $(\boldsymbol{\Lambda}\nabla_\beta\mathbf{G})_{n+k} = 1$. Recalling the phase of the autoassociative Alpha-Beta memory of type min $\mathbf{\Lambda}$ with vector $\mathbf{G}$ as input takes the following form for the $(n+k)$th recalled component:

$$(\mathbf{\Lambda}\nabla_{\beta}\mathbf{G})_{n+k} = \bigvee_{j=1}^{n} \beta(\lambda_{(n+k)j}, G_j) = \bigvee_{j=1}^{n} \beta\left\{ \left[\bigwedge_{\mu=1}^{p} \alpha\left(\overline{X}^{\mu}_{n+k}, \overline{X}^{\mu}_{j}\right) \right], G_j \right\}. \tag{3.8}$$

Due to the way vector $\overline{\mathbf{X}}^{k}$ is built, besides that $\overline{X}^{k}_{n+k} = 0$, it is important to notice that $\overline{X}^{\mu}_{n+k} \neq 0$, for all $\mu \neq k$, and from here we can state that

$$\bigwedge_{\mu=1}^{p} \alpha\left(\overline{X}^{\mu}_{n+k}, \overline{X}^{\mu}_{j}\right) = \alpha\left(\overline{X}^{k}_{n+k}, \overline{X}^{k}_{j}\right) = \alpha\left(0, \overline{X}^{k}_{j}\right) \tag{3.9}$$

is different from 2 regardless of the value of $\overline{X}^{k}_{j}$. Taking into account that $G_j = 0$ for all $j \in \{1, \dots, n+p\}$, we can conclude that it is impossible for

$$(\mathbf{\Lambda}\nabla_{\beta}\mathbf{G})_{n+k} = \bigvee_{j=1}^{n} \beta\left(\alpha\left(0, \overline{X}^{k}_{j}\right), 0\right) \tag{3.10}$$

to be equal to 1. That is, $(\mathbf{\Lambda}\nabla_{\beta}\mathbf{G})_{n+k} = 0 = \overline{X}^{k}_{n+k}$. □

Theorem 3.14. *Let $\{(\overline{\mathbf{X}}^{k}, \overline{\mathbf{X}}^{k}) \mid k = 1, \dots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by $\mathbf{\Lambda}$, with $\overline{\mathbf{X}}^{k} = \tau^{e}(\mathbf{x}^{k}, \overline{\mathbf{h}}^{k})$ for $k = 1, \dots, p$, and let $\mathbf{G} = \tau^{e}(\mathbf{x}^{k}, \mathbf{w}) \in A^{n+p}$ be a pattern altered with subtractive noise with respect to some specific pattern $\mathbf{X}^{k}$, with $\mathbf{w} \in A^{p}$ being a vector whose components have values $w_i = u_i - 1$, and $\mathbf{u} \in A^{p}$ the vector defined as $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^{i}$. Let us assume that during the recalling phase, $\mathbf{G}$ is presented to memory $\mathbf{\Lambda}$ as input, and the pattern $\mathbf{S} = \mathbf{\Lambda}\nabla_{\beta}\mathbf{G} \in A^{n+p}$is obtained as output. If when taking vector $\mathbf{S}$ as argument, the contraction vectorial transform $\mathbf{s} = \tau^{c}(\mathbf{S}, n) \in A^{p}$ is done, the resulting vector $\mathbf{s}$ has two mutually exclusive possibilities: $\exists k \in \{1, \dots, p\}$ such that $\mathbf{s} = \overline{\mathbf{h}}^{k}$, or $\mathbf{s}$ is not a one-hot vector.*

Proof. From the definition of contraction vectorial transform, we have that $s_i = S_{i+n} = (\mathbf{\Lambda}\nabla_{\beta}\mathbf{G})_{i+n}$ for $1 \leq i \leq p$, and in particular, by making $i = k$ we have $s_k = S_{k+n} = (\mathbf{\Lambda}\nabla_{\beta}\mathbf{G})_{k+n}$. However, by Lemma 3.13 $(\mathbf{\Lambda}\nabla_{\beta}\mathbf{G})_{n+k} = \overline{X}^{k}_{n+k}$, and since $\overline{\mathbf{X}}^{k} = \tau^{e}(\mathbf{x}^{k}, \overline{\mathbf{h}}^{k})$, the value $\overline{X}^{k}_{n+k}$ is equal to the value of component $\overline{h}^{k}_{k} = 0$. That is, $s_k = 0$. When considering that $s_k = 0$, vector $\mathbf{s}$ has two mutually exclusive possibilities: it can be that $s_j = 1$ for all $j \neq k$ in which case $\mathbf{s} = \overline{\mathbf{h}}^{k}$; or it holds that $\exists j \in \{1, \dots, p\}$, $j \neq k$ for which $s_j = 0$, in which case it is not possible for $\mathbf{s}$ to be a zero-hot vector, given Definition 3.5. □

Lemma 3.15. *Let $\{(\overline{\mathbf{X}}^{k}, \overline{\mathbf{X}}^{k}) \mid k = 1, \dots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by $\mathbf{V}$, with $\mathbf{X}^{k} = \tau^{e}(\mathbf{x}^{k}, \mathbf{h}^{k}) \in A^{n+p}$ for all $k \in \{1, \dots, p\}$. If t is an index such that $n + 1 \leq t \leq n + p$, then $\nu_{ij} \neq 0$ for all $j \in \{1, \dots, n+p\}$.*

Proof. In order to establish that $\nu_{ij} \neq 0$ for all $j \in \{1, \ldots, n+p\}$, given the definition of α, it is enough to find, for each for all $t \in \{n+1, \ldots, n+p\}$, an index μ for which $X_t^\mu = 1$ in the expression that produces the tjth component of memory $\mathbf{V}$, which is $\nu_{tj} = \bigvee_{\mu=1}^{p} \alpha(X_t^\mu, X_j^\mu)$. Due to the way each vector $\mathbf{X}^\mu = \tau^e(\mathbf{x}^\mu, \mathbf{h}^\mu)$ for $\mu = 1, \ldots, p$ is built, and given the domain of index $t \in \{n+1, \ldots, n+p\}$, for each t exists $s \in \{1, \ldots, p\}$ such that $t = n+s$. This is why two useful values to determine the result are $\mu = s$ and $t = n+s$, because $X_{n+s}^s = 1$. Then, $\nu_{tj} = \vee_{\mu=1}^{p} \alpha(X_t^\mu, X_j^\mu) = \alpha(X_{n+s}^s, X_j^s) = \alpha(1, X_j^s)$, a value which is different from 0. That is, $\nu_{ij} \neq 0$ for all $j \in \{1, \ldots, n+p\}$. □

Lemma 3.16. *Let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \ldots, p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by* $\mathbf{V}$, *with* $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ *for* $k = 1, \ldots, p$, *and let* $\mathbf{F} = \tau^e(\mathbf{x}^k, \mathbf{u}) \in A^{n+p}$ *be an altered version, by additive noise, of a specific pattern* $\mathbf{X}^k$, *with* $\mathbf{u} \in A^p$ *being the vector defined as* $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$. *Let us assume that during the recalling phase,* $\mathbf{F}$ *is presented to memory* $\mathbf{\Lambda}$ *as input. Given a fixed index* $t \in \{n+1, \ldots, n+p\}$ *such that* $t \neq n+k$, *it holds that* $(\mathbf{V}\Delta_\beta \mathbf{F})_t = 1$ *if and only if the following logic proposition is true: for all* $j \in \{1, \ldots, n+p\}$, $(F_j = 0 \rightarrow \nu_{tj} = 2)$.

Proof. Due to the way vectors $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ and $\mathbf{F} = \tau^e(\mathbf{x}^k, \mathbf{u})$ are built, we have that $F_t = 1$ is the component with additive noise with respect to component $X_t^k = 0$.

⇒) There are two possible cases.

Case 1. Pattern $\mathbf{F}$ does not contain components with value 0. That is, $F_j = 1$, $j \in \{1, \ldots, n+p\}$. This means that the antecedent of proposition $F_j = 0 \rightarrow \nu_{tj} = 2$ is false, and therefore, regardless of the truth value of consequence $\nu_{tj} = 2$, the expression for all $j \in \{1, \ldots, n+p\}$ $(F_j = 0 \rightarrow \nu_{tj} = 2)$ is true.

Case 2. Pattern $\mathbf{F}$ contains at least one component with value 0. That is, $\exists r \in \{1, \ldots, p\}$ such that $F_r = 0$. By hypothesis, $(\mathbf{V}\Delta_\beta \mathbf{F})_t = 1$, which means that the condition for a perfect recall of $X_t^k = 0$ is not met. In other words, according to Theorem 3.9 expression $\neg[\exists j \in \{1, \ldots, n+p\}$ such that $\nu_{tj} \leq \alpha(X_t^k, F_j)]$ is true, which is equivalent to

$$\forall j \in \{1, \ldots, n+p\} \text{ it holds that } \nu_{tj} \leq \alpha\left(X_t^k, F_j\right). \tag{3.11}$$

In particular, for $j = r$, and taking into account that $X_t^k = 0$, this inequality ends up like this: $\nu_{tr} > \alpha(X_t^k, F_r) = \alpha(0, 0) = 1$. That is, $\nu_{tr} = 2$, and therefore the expression for all $j \in \{1, \ldots, n+p\}$ $(F_j = 0 \rightarrow \nu_{tj} = 2)$ is true.

⇐) Assuming the following expression is true for all $j \in \{1, \ldots, n+p\}$ $(F_j = 0 \rightarrow \nu_{tj} = 2)$, there are two possible cases.

Case 1. Pattern $\mathbf{F}$ does not contain components with value 0. That is, $F_j = 1$ for all $j \in \{1, \ldots, n+p\}$. When considering that $(\mathbf{V}\Delta_\beta \mathbf{F})_t = \bigwedge_{j=1}^{n+p} \beta(\nu_{tj}, F_j)$, according to the definition of β, it is enough to show that for all $j \in \{1, \ldots, n+p\}$, $\nu_{tj} \neq 0$, which is guaranteed by Lemma 3.15. Then, it has been proven that $(\mathbf{V}\Delta_\beta \mathbf{F})_t = \bigwedge_{j=1}^{n+p} \beta(\nu_{tj}, F_j) = \bigwedge_{j=1}^{n+p} \beta(\nu_{tj}, 1) = 1$.

Case 2. Pattern $\mathbf{F}$ contains at least one component with value 0. That is, $\exists r \in \{1, \ldots, p\}$ such that $F_r = 0$. By hypothesis we have that for all $j \in \{1, \ldots, n+p\}$, $(F_j = 0 \rightarrow \nu_{tj} = 2)$ and, in particular, for $j = r$ and $\nu_{tr} = 2$, which means that $(\mathbf{V}\Delta_\beta \mathbf{F})_t = \bigwedge_{j=1}^{n+p} \beta(\nu_{tj}, F_j) = \beta(\nu_{tr}, F_r) = \beta(2, 1) = 1$. □

Corollary 3.17. *Let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \ldots, p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by* $\mathbf{V}$*, with* $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ *for* $k = 1, \ldots, p$*, and let* $\mathbf{F} = \tau^e(\mathbf{x}^k, \mathbf{u}) \in A^{n+p}$ *be an altered version, by additive noise, of a specific pattern* $\mathbf{X}^k$*, with* $\mathbf{u} \in A^p$ *being the vector defined as* $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$*. Let us assume that during the recalling phase,* $\mathbf{F}$ *is presented to memory* $\mathbf{\Lambda}$ *as input. Given a fixed index* $t \in \{n+1, \ldots, n+p\}$ *such that* $t \neq n+k$*, it holds that* $(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0$ *if and only if the following logic proposition is true: for all* $j \in \{1, \ldots, n+p\}$*,* $(F_j = 0$ *AND* $\nu_{tj} \neq 2)$*.*

Proof. In general, given two logical propositions P and Q, the proposition (P if and only if Q) is equivalent to proposition ($\neg P$ if and only if $\neg Q$). If P is identified with equality $(\mathbf{V}\Delta_\beta \mathbf{F})_t = 1$ and Q with expression for all $j \in \{1, \ldots, n+p\}$ $(F_j = 0 \to \nu_{tj} = 2)$, by Lemma 3.16 the following proposition is true: $\{\neg[(\mathbf{V}\Delta_\beta \mathbf{F})_t = 1]$ if and only if $\neg$[for all $j \in \{1, \ldots, n+p\}$ $(F_j = 0 \to \nu_{tj=2})]\}$. This expression transforms into the following equivalent propositions:

$$\begin{gathered}
\left\{(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0 \text{ iff } \exists j \in \{1, \ldots, n+p\} \text{ such that } \neg(F_j = 0 \to \nu_{tj} = 2)\right\}, \\
\left\{(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0 \text{ iff } \exists j \in \{1, \ldots, n+p\} \text{ such that } \neg\left[\neg(F_j = 0) \text{ OR } \nu_{tj} = 2\right]\right\}, \\
\left\{(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0 \text{ iff } \exists j \in \{1, \ldots, n+p\} \text{ such that } \left[\neg\left[\neg(F_j = 0)\right] \text{ AND} \neg\ (\nu_{tj} = 2)\right]\right\}, \\
\left\{(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0 \text{ iff } \exists j \in \{1, \ldots, n+p\} \text{ such that } \left[(F_j = 0) \text{ AND } \nu_{tj} \neq 2\right]\right\}.
\end{gathered} \tag{3.12}$$

□

Lemma 3.18. *Let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \ldots, p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by* $\mathbf{\Lambda}$*, with* $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k) \in A^{n+p}$ *for all* $k \in \{1, \ldots, p\}$*. If* t *is an index such that* $n+1 \le t \le n+p$*, then* $\lambda_{tj} \neq 2$ *for all* $j \in \{1, \ldots, n+p\}$*.*

Proof. In order to establish that $\lambda_{tj} \neq 2$ for all $j \in \{1, \ldots, n+p\}$, given the definition of α, it is enough to find, for each $t \in \{n+1, \ldots, n+p\}$, an index μ for which $\overline{X}_t^\mu = 0$ in the expression leading to obtaining the tjth component of memory $\mathbf{\Lambda}$, which is $\lambda_{tj} = \bigwedge_{\mu=1}^{p} \alpha(\overline{X}_t^\mu, \overline{X}_j^\mu)$. In fact, due to the way each vector $\overline{X}^k = \tau^e(x^k, \overline{h}^k)$ for $\mu = 1, \ldots, p$ is built, and given the domain of index $t \in \{n+1, \ldots, n+p\}$, for each t exists $s \in \{1, \ldots, p\}$ such that $t = n+s$; therefore two values useful to determine the result are $\mu = s$ and $t = n+s$, because $\overline{X}_{n+s}^s = 0$, then $\lambda_{tj} = \bigwedge_{\mu=1}^{p} \alpha(\overline{X}_t^\mu, \overline{X}_j^\mu) = \alpha(\overline{X}_{n+s}^s, \overline{X}_j^s) = \alpha(0, X_j^\mu)$, a value different from 2. That is, $\lambda_{tj} \neq 2$ for all $j \in \{1, \ldots, n+p\}$. □

Lemma 3.19. *Let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \ldots, p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by* $\mathbf{\Lambda}$*, with* $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$*for* $k = 1, \ldots, p$*, and let* $\mathbf{G} = \tau^e(\mathbf{x}^k, \mathbf{w}) \in A^{n+p}$ *be an altered version, by subtractive noise, of a specific pattern* $\mathbf{X}^k$*, with* $\mathbf{w} \in A^p$ *being a vector whose components have values* $w_i = u_i - 1$*, and* $\mathbf{u} \in A^p$ *the vector defined as* $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$*. Let us assume that during the recalling phase,* $\mathbf{G}$ *is presented to memory* $\mathbf{\Lambda}$ *as input. Given a fixed index* $t \in \{n+1, \ldots, n+p\}$ *such that* $t \neq n+k$*, it holds that* $(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 0$*, if and only if the following logical proposition is true for all* $j \in \{1, \ldots, n+p\}$ $(G_j = 1 \to \lambda_{tj} = 0)$*.*

Proof. Due to the way vectors $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$ and $\mathbf{G} = \tau^e(\mathbf{x}^k, \mathbf{w})$ are built, we have that $G_t = 1$ is the component with subtractive noise with respect to component $\overline{X}_t^k = 1$.

⇒) There are two possible cases.

Case 1. Pattern $\mathbf{G}$ does not contain components with value 1. That is, $G_j = 0$ for all $j \in \{1, \dots, n+p\}$. This means that the antecedent of logical proposition $G_j = 1 \rightarrow \lambda_{tj} = 0$ is false and therefore, regardless of the truth value of consequent $\lambda_{tj} = 0$, the expression for all $j \in \{1, \dots, n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)$ is true.

Case 2. Pattern $\mathbf{G}$ contains at least one component with value 1. That is, $\exists r \in \{1, \dots, n+p\}$ such that $G_r = 1$. By hypothesis, $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = 0$, which means that the perfect recall condition of $\overline{X}_t^k = 1$ is not met. In other words, according to Theorem 3.12, expression $\neg[\exists j \in \{1, \dots, n+p\}$ such that $\lambda_{tj} \geq \alpha(\overline{X}_t^k, G_j)]$ is true, which in turn is equivalent to

$$\forall j \in \{1, \dots, n+p\} \text{ it holds that } \lambda_{tj} < \alpha\left(\overline{X}_t^k, G_j\right). \tag{3.13}$$

In particular, for $j = r$ and considering that $\overline{X}_t^k = 1$, this inequality yields $\lambda_{tr} < \alpha(\overline{X}_t^k, G_r) = \alpha(1,1) = 1$. That is, $\lambda_{tr} = 0$, and therefore the expression for all $j \in \{1, \dots, n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)$ is true.

⇐) Assuming the following expression to be true, for all $j \in \{1, \dots, n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)$, there are two possible cases.

Case 1. Pattern $\mathbf{G}$ does not contain components with value 1. That is, $G_j = 0$ for all $j \in \{1, \dots, n+p\}$. When considering that $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = \bigvee_{j=1}^{n+p} \beta(\lambda_{tj}, G_j)$, according to the β definition, it is enough to show that for all $j \in \{1, \dots, n+p\}$, $\lambda_{tj} \neq 2$, which is guaranteed by Lemma 3.18. Then, it is proven that $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = \bigvee_{j=1}^{n+p} \beta(\lambda_{tj}, G_j) = \bigvee_{j=1}^{n+p} \beta(\lambda_{tj}, 0) = 0$.

Case 2. Pattern $\mathbf{G}$ contains at least one component with value 1. That is, $\exists r \in \{1, \dots, n+p\}$ such that $G_r = 1$. By hypothesis we have that for all $j \in \{1, \dots, n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)$ and, in particular, for $j = r$ and $\lambda_{tr} = 0$, which means that $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = \bigvee_{j=1}^{n+p} \beta(\lambda_{tj}, G_j) = \beta(\lambda_{tr}, G_r) = \beta(0,0) = 0$. □

Corollary 3.20. *Let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \dots, p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by* $\mathbf{\Lambda}$, *with* $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$ *for* $k = 1, \dots, p$, *and let* $\mathbf{G} = \tau^e(\mathbf{x}^k, \mathbf{w}) \in A^{n+p}$ *be an altered version, by substractive noise, of a specific pattern* $\mathbf{X}^k$, *with* $\mathbf{w} \in A^p$ *being a vector whose components have values* $w_i = u_i - 1$, *and* $\mathbf{u}$ *the vector defined as* $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$. *Let us assume that during the recalling phase,* $\mathbf{G}$ *is presented to memory* $\mathbf{\Lambda}$ *as input. Given a fixed index* $t \in \{n+1, \dots, n+p\}$ *such that* $t \neq n+k$, *it holds that* $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = 1$ *if and only if the following logic proposition is true:* $\exists j \in \{1, \dots, n+p\}$ $(G_j = 1\, AND\ \lambda_{tj} \neq 0)$.

Proof. In general, given two logical propositions P and Q, the proposition (P if and only if Q) is equivalent to proposition ($P\neg$ if and only if $\neg Q$). If P is identified with equality $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = 0$ and Q with expression for all $j \in \{1, \dots, n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)$, by Lemma 3.19, the following proposition is true: $\{\neg[(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 0]$ if and only if $\neg[$for all $j \in \{1,\ldots,n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)]\}$. This expression transforms into the following equivalent propositions:

$$
\begin{gathered}
\left\{(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 1 \text{ iff } \exists j \in \{1,\ldots,n+p\} \text{ such that } \neg\left[(G_j = 1 \rightarrow \lambda_{tj} = 0)\right]\right\}, \\
\left\{(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 1 \text{ iff } \exists j \in \{1,\ldots,n+p\} \text{ such that } \neg\left[\neg(G_j = 1) \rightarrow \text{ OR } \lambda_{tj} = 0\right]\right\}, \\
\left\{(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 1 \text{ iff } \exists j \in \{1,\ldots,n+p\} \text{ such that } \left[\neg\left[\neg(G_j = 1)\right] \text{ AND } (\lambda_{tj} = 0)\right]\right\}, \\
\left\{(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 1 \text{ iff } \exists j \in \{1,\ldots,n+p\} \text{ such that } \left[G^j = 1 \text{ AND } \lambda_{tj} \neq 0\right]\right\}.
\end{gathered}
\tag{3.14}
$$

□

Lemma 3.21. *Let* $\{(\mathbf{X}^k, \mathbf{X}^k) \mid k = 1,\ldots,p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by* $\mathbf{V}$*, with* $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ *for* $k = 1,\ldots,p$*, and let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\ldots,p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by* $\mathbf{\Lambda}$*, with* $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$*, for all* $k \in \{1,\ldots,p\}$*. Then, for each* $i \in \{n+1,\ldots,n+p\}$ *such that* $i = n+r$*, with* $r_i \in \{1,\ldots,p\}$*, it holds that* $\nu_{ij} = \alpha(1, X_j^{r_i})$ *and* $\lambda_{ij} = \alpha(0, \overline{X}_j^{r_i})$ *for all* $j \in \{1,\ldots,n+p\}$*.*

Proof. Due to the way vectors $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ and $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$ are built, we have that $X_i^{r_i} = 1$ and $\overline{X}_i^{r_i} = 0$, besides $X_i^\mu = 0$ and $\overline{X}_i^\mu = 1$ for all $\mu \neq r_i$ such that $\mu \in \{1,\ldots,p\}$. Because of this, and using the definition of α, $\alpha(X_i^{r_i}, X_j^{r_i}) = \alpha(1, X_j^{r_i})$ and $\alpha(X_i^\mu, X_j^\mu) = \alpha(0, X_j^\mu)$, which implies that, regardless of the values of $X_j^{r_i}$ and X_j^μ, it holds that $\alpha(X_i^{r_i}, X_j^{r_i}) \geq \alpha(X_i^\mu, X_j^\mu)$, from whence

$$
\nu_{ij} = \bigvee_{\mu=1}^{p} \alpha\left(X_i^\mu, X_j^\mu\right) = \alpha\left(X_i^{r_i}, X_j^{r_i}\right) = \alpha\left(1, X_j^{r_i}\right). \tag{3.15}
$$

We also have $\alpha(\overline{X}_i^{r_i}, \overline{X}_j^{r_i}) = \alpha(0, \overline{X}_j^{r_i})$ and $\alpha(\overline{X}_i^\mu, \overline{X}_j^\mu) = \alpha(1, \overline{X}_j^\mu)$, which implies that, regardless of the values of $\overline{X}_j^{r_i}$ and $\overline{X}_j^\mu$, it holds that $\alpha(\overline{X}_i^{r_i}, \overline{X}_j^{r_i}) \leq \alpha(\overline{X}_i^\mu, \overline{X}_j^\mu)$, from whence

$$
\lambda_{ij} = \bigwedge_{\mu=1}^{p} \alpha\left(\overline{X}_i^\mu, \overline{X}_j^\mu\right) = \alpha\left(\overline{X}_i^{r_i}, \overline{X}_j^{r_i}\right) = \alpha\left(0, \overline{X}_j^{r_i}\right) \tag{3.16}
$$

$\mu \in \{1,\ldots,p\}$, for all $j \in \{1,\ldots,n+p\}$. □

Corollary 3.22. *Let* $\{(\mathbf{X}^k, \mathbf{X}^k) \mid k = 1,\ldots,p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type* max *represented by* $\mathbf{V}$*, with* $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ *for all* $k \in \{1,\ldots,p\}$*, and let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\ldots,p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type* min *represented by* $\mathbf{\Lambda}$*, with* $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$*, for all* $k \in \{1,\ldots,p\}$*. Then,* $\nu_{ij} = \lambda_{ij} + 1$*, for all* $i \in \{n+1,\ldots, n+p\}$*,* $i = n + r_i$*, with* $r_i \in \{1,\ldots,p\}$ *and for all* $j \in \{1,\ldots,n\}$*.*

Proof. Let $i \in \{n+1,\ldots,n+p\}$ and $j \in \{1,\ldots,n\}$ be two indexes arbitrarily selected. By Lemma 3.21, the expressions used to calculate the ijth components of memories $\mathbf{V}$ and $\mathbf{\Lambda}$ take the following values:

$$\nu_{ij} = \alpha\left(1, X_j^{r_i}\right), \qquad \lambda_{ij} = \alpha\left(0, \overline{X}_j^{r_i}\right). \tag{3.17}$$

Considering that for all $j \in \{1,\ldots,n\}$, $X_j^{r_i} = \overline{X}_j^{r_i}$, there are two possible cases.

Case 1 ($X_j^{r_i} = 0 = \overline{X}_j^{r_i}$). We have the following values: $\nu_{ij} = \alpha(1,0) = 2$ and $\lambda_{ij} = \alpha(0,0) = 1$, therefore $\nu_{ij} = \lambda_{ij} + 1$.

Case 2 ($X_j^{r_i} = 1 = \overline{X}_j^{r_i}$). We have the following values: $\nu_{ij} = \alpha(1,1) = 1$ and $\lambda_{ij} = \alpha(0,1) = 0$, therefore $\nu_{ij} = \lambda_{ij} + 1$.

Since both indexes i and j were arbitrarily chosen inside their respective domains, the result $\nu_{ij} = \lambda_{ij} + 1$ is valid for all $i \in \{n+1,\ldots,n+p\}$ and for all $j \in \{1,\ldots,n\}$. □

Lemma 3.23. *Let $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\ldots,p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by $\mathbf{V}$, with $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ for all $k \in \{1,\ldots,p\}$, and let $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\ldots,p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by $\mathbf{\Lambda}$, with $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$, for all $k \in \{1,\ldots,p\}$. Also, if we define vector $\mathbf{u} \in A^p$ as $\mathbf{u} = \sum_{i=1}^p \mathbf{h}^i$, and take a fixed index for all $r \in \{1,\ldots,p\}$, let us consider two noisy versions of pattern $X^r \in A^{n+p}$: vector $\mathbf{F} = \tau^e(\mathbf{x}^r, \mathbf{u}) \in A^{n+p}$ which is an additive noise altered version of pattern X^r, and vector $\mathbf{G} = \tau^e(\mathbf{x}^r, \mathbf{w}) \in A^{n+p}$, which is a substractive noise altered version of pattern $\overline{\mathbf{X}}^r$, with $\mathbf{w} \in A^p$ being a vector whose components take the values $w_i = u_i - 1$ for all $i \in \{1,\ldots,p\}$. If during the recalling phase, $\mathbf{G}$ is presented as input to memory $\mathbf{\Lambda}$ and $\mathbf{F}$ is presented as input to memory $\mathbf{V}$, and if also it holds that $(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 0$ for an index $t \in \{n+1,\ldots,n+p\}$, being fixed such that $t \neq n+r$, then $(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0$.*

Proof. Due to the way vectors $\mathbf{X}^r$, $\mathbf{F}$ and $\mathbf{G}$ are built, we have that $F_t = 1$ is the component in the vector with additive noise corresponding to component X_t^r, and $G_t = 0$ is the component in the vector with subtractive noise corresponding to component $\overline{X}_t^r$. Also, since $t \neq n+r$, we can see that $X_t^r \neq 1$, that is, $X_t^r = 0$ and $\overline{X}_t^r = 1$. There are two possible cases.

Case 1. Pattern F does not contain any component with value 0. That is, $F_j = 1$ for all $j \in \{1,\ldots,n+p\}$. By Lemma 3.15 $\nu_{tj} \neq 0$ for all $j \in \{1,\ldots,n+p\}$, then $\beta(\nu_{tj}, F_j)$ for all $j \in \{1,\ldots,n+p\}$, which means that $(\mathbf{V}\Delta_\beta \mathbf{F})_t = \bigwedge_{j=1}^{n+p} \beta(\nu_{tj}, F_j) = 1$. In other words, expression $(\mathbf{V}\Delta_\beta \mathbf{F})_t = 0$ is false. The only possibility for the theorem to hold is for expression $(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 0$ to be false too. That is, we need to show that $(\mathbf{\Lambda}\nabla_\beta \mathbf{G})_t = 1$. According to Corollary 3.20, the latter is true if for every $t \in \{n+1,\ldots,n+p\}$ with $t \neq n+r$, exists $j \in \{1,\ldots,n+p\}$ such hat ($G_j = 1$ AND $\lambda_{tj} \neq 0$). Now, $t \neq n+r$ indicates that $\exists s \in \{1,\ldots,p\}$, $s \neq r$ such that $t = n+s$, and by Lemma 3.21 $\alpha(\overline{X}_t^s, \overline{X}_j^s) \leq \alpha(\overline{X}_t^\mu, \overline{X}_j^\mu)$ for all $\mu \in \{1,\ldots,p\}$, for all $j \in \{1,\ldots,n+p\}$, from where we have $\lambda_{tj} = \bigwedge_{j=1}^p \alpha(\overline{X}_t^\mu, \overline{X}_j^\mu) = \alpha(\overline{X}_t^s, \overline{X}_j^s)$, and by noting the equality $\overline{X}_t^s = \overline{X}_{n+s}^s = 0$, it holds that

$$\lambda_{tj} = \alpha\left(0, \overline{X}_j^s\right) \quad \forall j \in \{1,\ldots,n+p\}. \tag{3.18}$$

On the other side, for all $i \in \{1,\dots,n\}$ the following equalities hold: $\overline{X}_i^r = x_i^r = 1$ and $\overline{X}_i^s = x_i^s$ and also, taking into account that $\mathbf{x}^r \neq \mathbf{x}^s$, it is clear that $\exists h \in \{1,\dots,p\}$ such that $x_h^s \neq x_h^r$, meaning $x_h^s = 0 = X_h^s$ and therefore,

$$\lambda_{th} = \alpha(0,0) = 1. \tag{3.19}$$

Finally, since for all $i \in \{1,\dots,n\}$ it holds that $G_i = \overline{X}_i^r = x_i^r = 1$, in particular $G_h = 1$, then we have proven that for every $t \in \{n+1,\dots,n+p\}$ with $t \neq n+r$, exists $j \in \{1,\dots,n+p\}$ such that ($G_j = 1$ and $\lambda_{tj} \neq 0$), and by Corollary 3.20 it holds that $(\mathbf{\Lambda}\nabla_\beta\mathbf{G})_t = 1$, thus making expression $(\mathbf{\Lambda}\nabla_\beta\mathbf{G})_t = 1$ be false.

Case 2. Pattern F contains, besides the components with value of 1, at least one component with value 0. That is, $\exists h \in \{1,\dots,n+p\}$ such that $F_h = 0$. Due to the way vectors $\mathbf{G}$ and $\mathbf{F}$ are built for all $i \in \{1,\dots,n\}$, $G_i = F_i$ and, also, necessarily $1 \leq h \leq n$ and thus $F_h = G_h = 0$. By hypothesis, $\exists t \in \{n+1,\dots,n+p\}$ being fixed such that $t \neq n+r$ and $(\mathbf{\Lambda}\nabla_\beta\mathbf{G})_t = 0$, and by Lemma 3.19 for all $j \in \{1,\dots,n+p\}$ $(G_j = 1 \rightarrow \lambda_{tj} = 0)$. Given the way vector $\mathbf{G}$ is built we have that for all $j \in \{n+1,\dots,n+p\}$, $G_j = 0$, thus making the former expression like this: for all $j \in \{1,\dots,n\}(G_j = 1 \rightarrow \lambda_{tj} = 0)$. Let J be a set, proper subset of $\{1,\dots,n\}$, defined like this: $J = \{j \in \{1,\dots,n\} \mid G_j = 1\}$. The fact that J is a proper subset of $\{1,\dots,n\}$ is guaranteed by the existence of $G_h = 0$. Now, $t \neq n+r$ indicates that $\exists s \in \{1,\dots,p\}$, $s \neq r$ such that $t = n+s$, and by Lemma 3.21 $\nu_{tj} = \alpha(1, X_j^s)$ and $\lambda_{tj} = \alpha(0, \overline{X}_j^s)$ for all $j \in \{1,\dots,n+p\}$, from where we have that for all $j \in J$, $\overline{X}_j^s = 1$, because if this was not the case, $\lambda_{tj} \neq 0$. This means that for each $j \in J$, $\overline{X}_j^s = 1 = G_j$ which in turn means that patterns $\mathbf{X}^r$ and $\mathbf{X}^s$ coincide with value 1 in all components with index $j \in J$. Let us now consider the complement of set J, which is defined as $J^c = \{j \in \{1,\dots,n\} \mid G_j = 0\}$. The existence of at least one value $j_0 \in J^c$ for which $G_{j_0} = 0$ and $\overline{X}_{j_0}^s = 1$ is guaranteed by the known fact that $\mathbf{x}^r \neq \mathbf{x}^s$. Let us see, if $\overline{X}_j^s = 0$ for all $j \in J^c$ then for all $j \in \{1,\dots,n\}$ it holds that $\overline{X}_j^s = G_j$, which would mean that $\mathbf{x}^r = \mathbf{x}^s$. Since $\exists j_0 \in J^c$ for which $G_{j_0} = 0$ and $\overline{X}_{j_0}^s = 1$, this means that $\exists j_0 \in J^c$ for which $F_{j_0} = 0$ and $X_{j_0}^s = 1$. Now, $\beta(\nu_{tj_0}, F_{j_0}) = \beta(\alpha(1, X_{j_0}^s), 0) = \beta(\alpha(1,1), 0) = \beta(1,0) = 0$, and finally

$$\left(\mathbf{V}\Delta_\beta\mathbf{F}\right)_t = \bigwedge_{j=1}^{n+p} \beta\left(\nu_{tj}, F_j\right) = \beta\left(\nu_{tj_0}, F_{j_0}\right) = 0. \tag{3.20}$$

□

Lemma 3.24. *Let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\dots,p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by* $\mathbf{V}$, *with* $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ *for all* $k \in \{1,\dots,p\}$, *and let* $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1,\dots,p\}$ *be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by* $\mathbf{\Lambda}$, *with* $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$, *for all* $k \in \{1,\dots,p\}$. *Also, if we define vector* $\mathbf{u} \in A^p$ *as* $\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i$, *and take a fixed index for all* $r \in \{1,\dots,p\}$, *let us consider two noisy versions of pattern* $X^r \in A^{n+p}$: *vector* $\mathbf{F} = \tau^e(\mathbf{x}^r, \mathbf{u}) \in A^{n+p}$ *which is an additive noise altered version of pattern* X^r, *and vector* $\mathbf{G} = \tau^e(\mathbf{x}^r, \mathbf{w}) \in A^{n+p}$, *which is a subtractive noise altered version of pattern* $\overline{\mathbf{X}}^r$, *with* $\mathbf{w} \in A^p$ *being a vector whose components take the values* $w_i = u_i - 1$ *for all* $i \in \{1,\dots,p\}$. *If during the recalling phase,* $\mathbf{G}$ *is presented as input to memory* $\mathbf{\Lambda}$ *and* $\mathbf{F}$ *is presented as input to memory* $\mathbf{V}$, *and if also it holds that* $(\mathbf{V}\Delta_\beta\mathbf{F})_t = 1$ *for an index* $t \in \{n+1,\dots,n+p\}$, *being fixed such that* $t \neq n+r$, *then* $(\mathbf{\Lambda}\nabla_\beta\mathbf{G})_t = 1$.

Proof. Due to the way vectors $\mathbf{X}^r$, $\mathbf{F}$ and $\mathbf{G}$ are built, we have that $F_t = 1$ is the component in the vector with additive noise corresponding to component X_t^r, and $G_t = 0$ is the component in the vector with subtractive noise corresponding to component $\overline{X}_t^r$. Also, since $t \neq n + r$, we can see that $X_t^r \neq 1$, that is, $X_t^r = 0$ and $\overline{X}_t^r = 1$. There are two possible cases.

Case 1. Pattern $\mathbf{G}$ does not contain any component with value 1. That is, $G_j = 0$ for all $j \in \{1, \ldots, n + p\}$. By Lemma 3.18 $\lambda_{tj} \neq 2$ for all $j \in \{1, \ldots, n + p\}$; thus $\beta(\lambda_{tj}, G_j) = 0$ for all $j \in \{1, \ldots, n + p\}$, which means that $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = \bigvee_{j=1}^{n+p} \beta(\lambda_{tj}, G_j) = 0$. In other words, expression $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = 1$ is false. The only possibility for the theorem to hold is for expression $(\mathbf{V} \Delta_\beta \mathbf{F})_t = 1$ to be false too. That is, we need to show that $(\mathbf{V} \Delta_\beta \mathbf{F})_t = 0$. According to Corollary 3.17, the latter is true if for every $t \in \{n + 1, \ldots, n + p\}$ with $t \neq n + r$, exists $j \in \{1, \ldots, n + p\}$ such that ($F_j = 0$ AND $\nu_{tj} \neq 2$). Now, $t \neq n + r$ indicates that $\exists s \in \{1, \ldots, p\}$, $s \neq r$ such that $t = n + s$, and by Lemma 3.19 $\alpha(X_t^s, X_j^s) \geq \alpha(X_t^\mu, X_j^\mu)$ for all $\mu \in \{1, \ldots, p\}$, for all $j \in \{1, \ldots, n + p\}$, from where we have $\nu_{tj} = \bigvee_{\mu=1}^{p} \alpha(X_t^\mu, X_j^\mu) = \alpha(X_t^s, X_j^s)$, and by noting the equality $X_t^s = X_{n+s}^s = 1$, it holds that

$$\nu_{tj} = \alpha\left(1, X_j^s\right) \quad \forall j \in \{1, \ldots, n + p\}. \tag{3.21}$$

On the other side, for all $i \in \{1, \ldots, n\}$ the following equalities hold: $X_i^r = x_i^r = 0$ and $X_i^s = x_i^s$ and also, taking into account that $\mathbf{x}^r \neq \mathbf{x}^s$, it is clear that $\exists h \in \{1, \ldots, p\}$ such that $x_h^s \neq x_h^r$, meaning $x_h^s = 1 = X_h^s$ and therefore,

$$\nu_{th} = \alpha(1, X_h^s) = \alpha(1, 1) = 1. \tag{3.22}$$

Finally, since for all $i \in \{1, \ldots, n\}$ it holds that $F_i = X_i^r = x_i^r = 0$, in particular $F_h = 0$, then we have proven that for every $t \in \{n+1, \ldots, n+p\}$ with $t \neq n+r$, exists $j \in \{1, \ldots, n+p\}$ such that ($F_j = 0$ AND $\nu_{tj} \neq 2$), and by Corollary 3.17 it holds that $(\mathbf{V} \Delta_\beta \mathbf{F})_t = 0$, thus making expression $(\mathbf{V} \Delta_\beta \mathbf{F})_t = 1$ false.

Case 2. Pattern $\mathbf{G}$ contains, besides the components with value of 0, at least one component with value 1. That is, $\exists h \in \{1, \ldots, n + p\}$ such that $G_h = 1$. Due to the way vectors $\mathbf{G}$ and $\mathbf{F}$ are built for all $i \in \{1, \ldots, n\}$, $G_i = F_i$ and, also, necessarily $1 \leq h \leq n$ and thus $F_h = G_h = 0$. By hypothesis $\exists t \in \{n + 1, \ldots, n + p\}$, being fixed such that $t \neq n + r$ and $(\mathbf{V} \Delta_\beta \mathbf{F})_t = 1$, and by Lemma 3.16 for all $j \in \{1, \ldots, n + p\}$ ($F_j = 0 \rightarrow \nu_{tj} = 2$). Given the way vector $\mathbf{F}$ is built, we have that for all $j \in \{n + 1, \ldots, n + p\}$, $G_j = 1$, thus making the former expression like this: for all $j \in \{1, \ldots, n+p\}$ ($F_j = 0 \rightarrow \nu_{tj} = 2$). Let J be a set, a proper subset of $\{1, \ldots, n\}$, defined like this: $J = \{j \in \{1, \ldots, n\} \mid F_j = 0\}$. The fact that J is a proper subset of $\{1, \ldots, n\}$ is guaranteed by the existence of $G_h = 1$. Now, $t \neq n + r$ indicates that $\exists s \in \{1, \ldots, p\}$, $s \neq r$ such that $t = n + s$, and by Lemma 3.21 $\nu_{tj} = \alpha(1, X_j^s)$ and $\lambda_{tj} = \alpha(0, \overline{X}_j^s)$ for all $j \in \{1, \ldots, n + p\}$, from where we have that for all $j \in J$, $X_j^s = 0$, because if this was not the case, $\nu_{tj} \neq 0$. This means that for each $j \in J$, $X_j^s = 0 = F_j$ which in turn means that patterns $\mathbf{X}^r$ and $\mathbf{X}^s$ coincide with value 0 in all components with index $j \in J$. Let us now consider the complement of set J, which is defined as $J^c = \{j \in \{1, \ldots, n\} \mid F_j = 1\}$. The existence of at least one value $j_0 \in J^c$ for which $F_{j_0} = 1$

and $X_{j_0}^s = 0$ is guaranteed by the known fact that $\mathbf{x}^r \neq \mathbf{x}^s$. Let us see, if $X_j^s = 1$ for all $j \in J^c$ then for all $j \in \{1, \ldots, n\}$ it holds that $X_j^s = F_j$, which would mean that $\mathbf{x}^r = \mathbf{x}^s$. Since $\exists j_0 \in J^c$ for which $F_{j_0} = 1$ and $X_{j_0}^s = 0$, this means that $\exists j_0 \in J^c$ for which $G_{j_0} = 1$ and $\overline{X}_{j_0}^s = 0$. Now, $\beta(\lambda_{tj_0}, G_{j_0}) = \beta(\alpha(0, \overline{X}_{j_0}^s), 1) = \beta(\alpha(0,0), 1) = \beta(1,1) = 1$, and finally

$$(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = \bigvee_{j=1}^{n+p} \beta(\lambda_{tj}, G_j) = \beta(\lambda_{tj_0}, G_{j_0}) = 1. \tag{3.23}$$

□

Theorem 3.25 (Main Theorem). *Let $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \ldots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type max represented by $\mathbf{V}$, with $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$ for all $k \in \{1, \ldots, p\}$, and let $\{(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k) \mid k = 1, \ldots, p\}$ be the fundamental set of an autoassociative Alpha-Beta memory of type min represented by $\mathbf{\Lambda}$, with $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$, for all $k \in \{1, \ldots, p\}$. Also, if we define vector $\mathbf{u} \in A^p$ as $\mathbf{u} = \sum_{i=1}^p \mathbf{h}^i$, and take a fixed index $r \in \{1, \ldots, p\}$, let us consider two noisy versions of pattern $X^r \in A^{n+p}$: vector $\mathbf{F} = \tau^e(\mathbf{x}^r, \mathbf{u}) \in A^{n+p}$, which is an additive noise altered version of pattern X^r, and vector $\mathbf{G} = \tau^e(\mathbf{x}^r, \mathbf{w}) \in A^{n+p}$, which is a subtractive noise altered version of pattern $\overline{\mathbf{X}}^r$, with $\mathbf{w} \in A^p$ being a vector whose components take the values $w_i = u_i - 1$ for all $i \in \{1, \ldots, p\}$. Now, let us assume that during the recalling phase, $\mathbf{G}$ is presented as input to memory $\mathbf{\Lambda}$ and $\mathbf{F}$ is presented as input to memory $\mathbf{V}$, and patterns $S = \mathbf{\Lambda} \nabla_\beta \mathbf{G} \in A^{n+p}$ and $\mathbf{R} = \mathbf{V} \Delta_\beta \mathbf{F} \in A^{n+p}$ are obtained. If when taking vector $\mathbf{R}$ as argument the contraction vectorial transform $\mathbf{r} = \tau^c(\mathbf{R}, n) \in A^p$ is done, and when taking vector $\mathbf{S}$ as argument the contraction vectorial transform $\mathbf{s} = \tau^c(\mathbf{S}, n) \in A^p$ is done, then $\mathbf{H} = (\mathbf{r}$ AND $\overline{\mathbf{s}})$ will be the kth one-hot vector of p bits, where $\overline{\mathbf{s}}$ is the negated from of $\mathbf{s}$.*

Proof. From the definition of contraction vectorial transform, we have that $\mathbf{r}_i = \mathbf{R}_{i+n} = (\mathbf{V} \Delta_\beta \mathbf{F})_{i+n}$ *and* $s_i = S_{i+n} = (\mathbf{\Lambda} \nabla_\beta \mathbf{G})_{i+n}$ for $1 \leq i \leq p$, and in particular, by making $i = k$ we have $r_k = R_{k+n} = (\mathbf{V} \Delta_\beta \mathbf{F})_{k+n}$ *and* $s_k = S_{k+n} = (\mathbf{\Lambda} \nabla_\beta \mathbf{G})_{k+n}$. By Lemmas 3.10 and 3.13 we have $(\mathbf{V} \Delta_\beta \mathbf{F})_{n+k} = X_{n+k}^k = 1$ and $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_{n+k} = \overline{X}_{n+k}^k = 0$, and thus: $H_k = r_k$ AND $\overline{s}_k = 1$ AND $\neg 0 = 1$ AND $1 = 1$.

Now, by Lemma 3.23 we know that if $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = 0$ such that $t = i + n$ is a fixed index with $t \neq n + k$, then $(\mathbf{V} \Delta_\beta \mathbf{F})_t = 0$; thus,

$$H_i = r_i \text{ AND } \overline{s}_i = (\mathbf{V} \Delta_\beta \mathbf{F})_t \text{ AND } \neg (\mathbf{\Lambda} \nabla_\beta \mathbf{G})_t = 0 \neg 0 = 0\ 1 = 0. \tag{3.24}$$

On the other side, by Lemma 3.24 it is known that if $(\mathbf{V} \Delta_\beta \mathbf{F})_q = 1$ for a fixed index $q = i + n$ such that $q \neq n + k$, then $(\mathbf{\Lambda} \nabla_\beta \mathbf{G})_q = 1$. According to the latter,

$$H_i = r_i \text{ AND } \neg s_i = (\mathbf{V} \Delta_\beta \mathbf{F})_q \text{ AND } \neg (\mathbf{\Lambda} \nabla_\beta \mathbf{G})_q = 1 \text{ AND } \neg 1 = 1\ 0 = 0. \tag{3.25}$$

Then $H_i = 1$ for $i = k$ and $H_i = 0$ for $i \neq k$. Therefore, and according to Definition 3.3, H will be the kth one-hot vector of p bits. □

3.3.2. Theoretical Foundation of Stages 2 and 4

In this section is presented the theoretical foundation which serves as the basis for the design and operation of Stages 2 and 4, whose main element is an original variation of the *Linear Associator*.

Let $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, \ldots, p\}$ with $A = \{0, 1\}$, $\mathbf{x}^\mu \in A^n$ and $\mathbf{y}^\mu \in A^m$ be the fundamental set of the *Linear Associator*. The Learning Phase consists of two stages.

(i) For each of the p associations $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ find matrix $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$ of dimensions $m \times n$.

(ii) The p matrices are added together to obtain the memory

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \tag{3.26}$$

in such way that the ijth component of memory M is expressed as

$$m_{ij} = \sum_{\mu=1}^{p} y_i^\mu x_j^\mu. \tag{3.27}$$

The Recalling Phase consists of presenting an input pattern $\mathbf{x}^\omega$ to the memory, where $\omega \in \{1, 2, \ldots, p\}$ and doing operation

$$\mathbf{M} \cdot \mathbf{x}^\omega = \left[\sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t \right] \cdot \mathbf{x}^\omega. \tag{3.28}$$

The following form of expression allows us to investigate the conditions that must be met in order for the proposed recalling method to give perfect outputs as results:

$$\mathbf{M} \cdot \mathbf{x}^\omega = \mathbf{y}^\omega \cdot \left[(\mathbf{x}^\omega)^t \cdot \mathbf{x}^\omega \right] + \sum_{\mu \neq \omega} \mathbf{y}^\mu \cdot \left[(\mathbf{x}^\mu)^t \cdot \mathbf{x}^\omega \right]. \tag{3.29}$$

For the latter expression to give pattern $\mathbf{y}^\omega$ as a result, it is necessary that two equalities hold:

(i) $[(\mathbf{x}^\omega)^t \cdot \mathbf{x}^\omega] = 1$;

(ii) $[(\mathbf{x}^\mu)^t \cdot \mathbf{x}^\omega] = 0$ as long as $\mu \neq \omega$.

This means that, in order to have perfect recall, vectors $\mathbf{x}^{\mu}$ must be orthonormal to each other. If that happens, then, for $\mu = 1, 2, \ldots, p$, we have

$$\mathbf{y}^1 \cdot \left(\mathbf{x}^1\right)^t = \begin{pmatrix} y_1^1 \\ y_2^1 \\ \vdots \\ y_m^1 \end{pmatrix} \cdot \left(x_1^1, x_2^1, \ldots, x_n^1\right) = \begin{pmatrix} y_1^1 & 0 & 0 & \cdots & 0_{(n)} \\ y_2^1 & 0 & 0 & \cdots & 0_{(n)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_m^1 & 0 & 0 & \cdots & 0_{(n)} \end{pmatrix},$$

$$\mathbf{y}^2 \cdot \left(\mathbf{x}^2\right)^t = \begin{pmatrix} y_1^2 \\ y_2^2 \\ \vdots \\ y_m^2 \end{pmatrix} \cdot \left(x_1^2, x_2^2, \ldots, x_n^2\right) = \begin{pmatrix} 0 & y_1^2 & 0 & \cdots & 0_{(n)} \\ 0 & y_2^2 & 0 & \cdots & 0_{(n)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & y_m^2 & 0 & \cdots & 0_{(n)} \end{pmatrix}, \tag{3.30}$$

$$\mathbf{y}^p \cdot \left(\mathbf{x}^p\right)^t = \begin{pmatrix} y_1^p \\ y_2^p \\ \vdots \\ y_m^p \end{pmatrix} \cdot \left(x_1^p, x_2^p, \ldots, x_n^p\right) = \begin{pmatrix} 0 & 0 & 0 & \cdots & y_{1\,(n)}^p \\ 0 & 0 & 0 & \cdots & y_{2\,(n)}^p \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & y_{m\,(n)}^p \end{pmatrix}.$$

Therefore,

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^{\mu} \cdot \left(\mathbf{x}^{\mu}\right)^t = \begin{pmatrix} y_1^1 & y_1^2 & y_1^3 & \cdots & y_n^p \\ y_2^1 & y_2^2 & y_2^3 & \cdots & y_2^p \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ y_m^1 & y_m^2 & y_m^3 & \cdots & y_m^p \end{pmatrix}. \tag{3.31}$$

Taking advantage of the characteristic shown by the *Linear Asssociator* when the input patterns are orthonormal, and given that, by Definition 3.3, one-hot vectors $\mathbf{v}^k$ with $k = 1, \ldots, p$ are orthonormal, we can obviate the learning phase by avoiding the vectorial operations done by the *Linear Associator*, and simply put the vectors in order, to form the *Linear Associator*.

Stages 2 and 4 correspond to two modified *Linear Associators*, built with vectors $\mathbf{y}$ and $\mathbf{x}$, respectively, of the fundamental set.

3.3.3. Algorithm

In this section we describe, step by step, the processes required by the Alpha-Beta BAM, in the Learning Phase as well as in the Recalling Phase (by convention only) in the direction $\mathbf{x} \to \mathbf{y}$, the algorithm for Stages 1 and 2.

The following algorithm describes the steps needed by the Alpha-Beta bidirectional associative memory for the learning and recalling phases to happen, in the direction $\mathbf{x} \to \mathbf{y}$.

Learning Phase

(1) For each index $k \in \{1, \ldots, p\}$, do expansion: $\mathbf{X}^k = \tau^e(\mathbf{x}^k, \mathbf{h}^k)$.

(2) Create an Alpha-Beta autoassociative memory of type max $\mathbf{V}$ with the fundamental set

$$\left\{\left(\mathbf{X}^k, \mathbf{X}^k\right) \mid k = 1, \ldots, p\right\}. \tag{3.32}$$

(3) For each index $k \in \{1, \ldots, p\}$, do expansion: $\overline{\mathbf{X}}^k = \tau^e(\mathbf{x}^k, \overline{\mathbf{h}}^k)$.

(4) Create an Alpha-Beta autoassociative memory of type min $\mathbf{\Lambda}$ with the fundamental set

$$\left\{\left(\overline{\mathbf{X}}^k, \overline{\mathbf{X}}^k\right) \mid k = 1, \ldots, p\right\}. \tag{3.33}$$

(5) Create a matrix consisting of a modified *Linear Associator* with patterns $\mathbf{y}^k$

$$\mathbf{LAy} = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^p \\ y_2^1 & y_2^2 & \cdots & y_2^p \\ \vdots & \vdots & \cdots & \vdots \\ y_n^1 & y_n^2 & \cdots & y_n^p \end{bmatrix}. \tag{3.34}$$

Recalling Phase

(1) Present, as input to Stage 1, a vector of the fundamental set $\mathbf{x}^\mu \in A^n$ for some index $\mu \in \{1, \ldots, p\}$.

(2) Build vector $\mathbf{u} \in A^p$ in the following manner:

$$\mathbf{u} = \sum_{i=1}^{p} \mathbf{h}^i. \tag{3.35}$$

(3) Do expansion: $\mathbf{F} = \tau^e(\mathbf{x}^\mu, \mathbf{u}) \in A^{n+p}$.

(4) Operate the Alpha-Beta autoassociative memory max $\mathbf{V}$ with $\mathbf{F}$, in order to obtain a vector $\mathbf{R}$ of dimension $n + p$

$$\mathbf{R} = \mathbf{V}\Delta_\beta \mathbf{F} \in A^{n+p}. \tag{3.36}$$

(5) Do contraction $\mathbf{r} = \tau^c(\mathbf{R}, n) \in A^p$.

(6) **If** ($\exists k \in \{1, \ldots, p\}$ such that $\mathbf{h}^k = \mathbf{r}$), it is assured that $k = \mu$ (based on Theorem 3.11), and the result is $\mathbf{h}^\mu$. Thus, operation LAy · r is done, resulting in the corresponding $\mathbf{y}^\mu$. STOP. **Else** {

(7) Build vector $\mathbf{w} \in A^p$ in such way that $w_i = u_i - 1$, for all $i \in \{1, \ldots, p\}$.

(8) Do expansion: $\mathbf{G} = \tau^e(\mathbf{x}^\mu, \mathbf{w}) \in A^{n+p}$.

(9) Operate the Alpha-Beta autoassociative memory min $\mathbf{\Lambda}$ with $\mathbf{G}$, in order to obtain a vector $\mathbf{S}$ of dimension $n + p$

$$\mathbf{S} = \mathbf{\Lambda} \nabla_\beta \mathbf{G} \in A^{n+p}. \tag{3.37}$$

(10) Do contraction $\mathbf{s} = \tau^c(\mathbf{S}^\mu, n) \in A^p$.

(11) **If** ($\exists k \in \{1, \ldots, p\}$ such that $\overline{\mathbf{h}}^k = \mathbf{s}$), it is assured that $k = \mu$ (based on Theorem 3.14), and the result is $\mathbf{h}^\mu$. Thus, operation $\mathbf{LAy} \cdot \overline{\mathbf{s}}$ is done, resulting in the corresponding $\mathbf{y}^\mu$. STOP. **Else** {.

Do operation $\mathbf{t} = \mathbf{r} \wedge \overline{\mathbf{s}}$, where $\mathbf{\Lambda}$ is the symbol of the logical AND. The result of this operation is $\mathbf{h}^\mu$ (based on Theorem 3.25). Operation $\mathbf{LAy} \cdot \mathbf{t}$ is done, in order to obtain the corresponding $\mathbf{y}^\mu$. STOP.}}.The process in the contrary direction, which is presenting pattern $\mathbf{y}^k$ $(k = 1, \ldots, p)$ as input to the Alpha-Beta BAM and obtaining its corresponding $\mathbf{x}^k$, is very similar to the one described above. The task of Stage 3 is to obtain a one-hot vector $\mathbf{h}^k$ given a $\mathbf{y}^k$. Stage 4 is a modified Linear Associator built in similar fashion to the one in Stage 2.All this theoretical foundations assure every training pattern to be recalled without imposing any condition in the nature of patterns, such as linear dependency, Hamming distance, orthogonality, nor the number of patterns to be trained. The algorithm shows that the method is not iterative but rather a one-shot algorithm, which is an advantage because our model does not have stability problems.

4. Experiments and Results

We first use Lindig's algorithm [5] to generate a concept lattice from a set of objects and attributes. In the learning phase, each object or subset of objects is associated with their corresponding attribute or subset of attributes, that is, concepts are stored in a BAM. The bidirectionality of this associative model enables the system to retrieve a concept from objects or attributes information. We present an illustrative example to explain our proposal in a simple way.

Suppose the context of the planets showed in Table 3. From the table, it can be observed that there are 9 objects corresponding to the number of planets, and there are 7 attributes. Based on this table a concept lattice is derived from Lindig's algorithm, which is shown in Figure 4.

In both forward and reverse directions, the number of objects becomes the number of elements of vectors $\mathbf{x}$ and the number of attributes is the number of elements of vectors $\mathbf{y}$, as Figure 5 shows.

A software implementation of the Alpha-Beta BAM to store concept lattices derived from Lindig's algorithm was developed. The software was programmed with Visual C++6.0. A result related to the example showed in Figure 4 can be observed in Figure 6.

The first step is to generate the concept lattice. Information about objects and attributes is read from a *txt* file, then Lindig's algorithm is applied and the concept lattice is built. We have the option to choose objects or attributes; these are selected and the corresponding concept is showed.

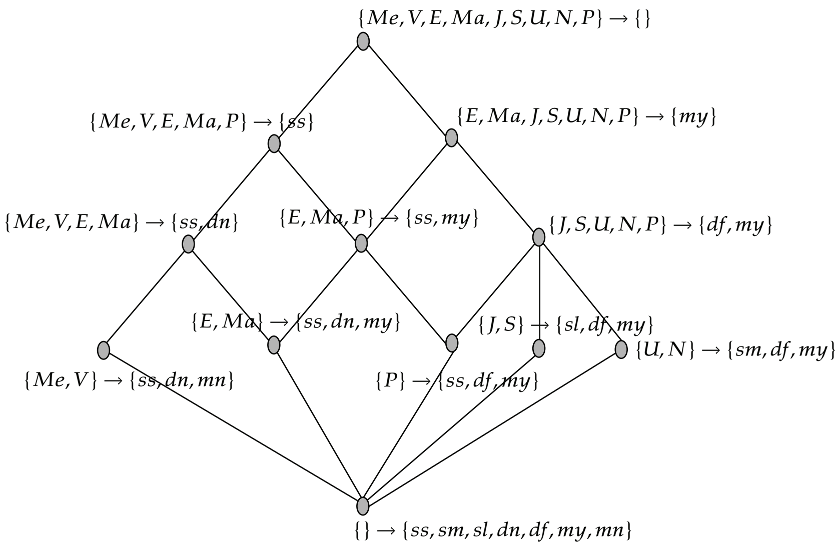

Figure 4: Concept lattice for the context of the planets given in Table 3.

$$\mathbf{x} = \begin{bmatrix} Me \\ V \\ E \\ Ma \\ J \\ S \\ U \\ N \\ P \end{bmatrix} \longrightarrow \mathbf{y} = \begin{bmatrix} ss \\ sm \\ sl \\ dn \\ df \\ my \\ mn \end{bmatrix} \qquad \boxed{\begin{matrix}\text{Example:} \\ \{Me, V, E, Ma\} \longrightarrow \{ss, dn\}\end{matrix}} \quad \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \longrightarrow \mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 5: In the learning phase, concepts are stored as associations of objects and attributes.

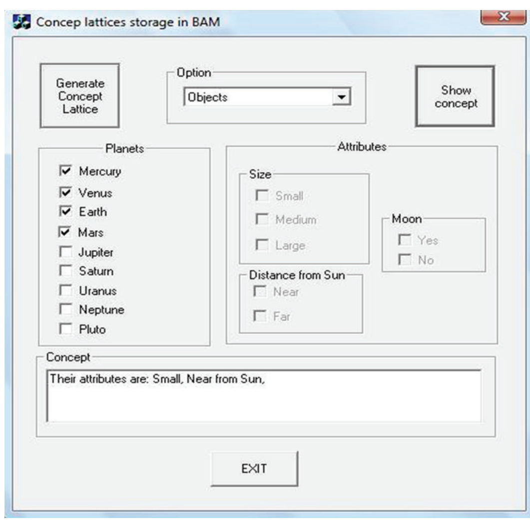

Figure 6: Example screen of software developed. Object information is provided and attributes are recalled.

Table 3: Context of planets.

Planet	Size			Distance from Sun		Moon	
	Small (ss)	Medium (sm)	Large (sl)	Near (dn)	Far (df)	Yes (my)	No (mn)
Mercury (Me)	—	—	—	—	—	—	—
Venus (V)	—	—	—	—	—	—	—
Earth (E)	—	—	—	—	—	—	—
Mars (Ma)	—	—	—	—	—	—	—
Jupiter (J)	—	—	—	—	—	—	—
Saturn (S)	—	—	—	—	—	—	—
Uranus (U)	—	—	—	—	—	—	—
Neptune (N)	—	—	—	—	—	—	—
Pluto (P)	—	—	—	—	—	—	—

We tested the system introducing each set of objects from each concept, and associated attributes were recalled. In the same way, sets of attributes from the concept lattice were presented to the Alpha-Beta BAM and associated objects were recalled. In this manner, we had perfect recall in the experiment.

In this case, the context has few elements. However, the context could have any number of objects and attributes and Alpha-Beta BAM will recall every association, that is, it will show perfect recall. This can be concluded because Alpha-Beta BAM has a mathematical foundation assuring perfect recall without imposing any condition.

5. Conclusions

Formal Concept Analysis is a tool to represent the way human beings conceptualize the real world by giving the bases to analyze contexts and to obtain formal concepts. Lindig's algorithm takes these formal concepts and builds a concept lattice to represent human thought. Therefore, this step provides a way to model mental level. On the other hand, Alpha-Beta Bidirectional Associative Memories are a great tool to store concept lattices because the model is a one-shot algorithm and all patterns are recalled perfectly in both directions. The process for storing formal concepts consists in associating every object with its corresponding attribute using Alpha-Beta operators during the learning phase. The recalling phase of Alpha-Beta BAM allows retrieving a formal concept by presenting an object or attribute. The main feature of Alpha-Beta BAM is that it exhibits perfect recall on all fundamental patterns without any condition in number of patterns or the nature of patterns. This makes our algorithm an adequate tool to store concepts. Besides, it provides a way to model brain level.

Our proposed model stores and retrieves concepts from a concept lattice in a suitable way. This model can be used to create knowledge databases, which is our next goal.

Acknowledgment

The authors would like to thank the Instituto Politécnico Nacional (COFAA and SIP) and SNI for their financial support to develop this work.

References

[1] C. Carpineto and G. Romano, *Concept Data Analysis: Theory and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

[2] J.-P. Bordat, “Calcul pratique du treillis de Galois d’une correspondance,” *Mathématiques et Sciences Humaines*, no. 96, pp. 31–47, 1986.

[3] M. I. Zabezhailo, V. G. Ivashko, S. O. Kuznetsov, M. A. Mikheenkova, K. P. Khazanovskii, and O. M. Anshakov, “Algorithms and programs of the JSM-method of automatic hypothesis generation,” *Automatic Documentation and Mathematical Linguistics*, vol. 21, no. 5, pp. 1–14, 1987.

[4] B. Ganter, “Two basic algorithms in concept analysis,” Tech. Rep. 831, Darmstadt University, Darmstadt, Germany, 1984.

[5] C. Lindig, *Algorithmen zur Begriffsanalyse und ihre Anwendung bei Softwarebibliotheken*, Technische Universität Braunschweig, Braunschweig, Germany, 1999.

[6] M. Chein, “Algorithme de recherche des sou-matrices premières d’une matrice,” *Bulletin Mathématique de la Société des Sciences Mathématiques de Roumanie*, vol. 13, no. 1, pp. 21–25, 1969.

[7] L. Nourine and O. Raynaud, “A fast algorithm for building lattices,” *Information Processing Letters*, vol. 71, no. 5-6, pp. 199–204, 1999.

[8] M. E. Acevedo-Mosqueda, C. Yáñez-Márquez, and I. López-Yáñez, “Alpha-beta bidirectional associative memories: theory and applications,” *Neural Processing Letters*, vol. 26, no. 1, pp. 1–40, 2007.

[9] Y.-J. Jeng, C.-C. Yeh, and T. D. Chiueh, “Exponential bidirectional associative memories,” *Electronics Letters*, vol. 26, no. 11, pp. 717–718, 1990.

[10] W.-J. Wang and D.-L. Lee, “Modified exponential bidirectional associative memories,” *Electronics Letters*, vol. 28, no. 9, pp. 888–890, 1992.

[11] S. Chen, H. Gao, and W. Yan, “Improved exponential bidirectional associative memory,” *Electronics Letters*, vol. 33, no. 3, pp. 223–224, 1997.

[12] Y.-F. Wang, J. B. Cruz Jr., and J. H. Mulligan Jr., “Two coding strategies for bidirectional associative memory,” *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 81–92, 1990.

[13] Y.-F. Wang, J. B. Cruz Jr., and J. H. Mulligan Jr., “Guaranteed recall of all training pairs for bidirectional associative memory,” *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 559–567, 1991.

[14] R. Perfetti, “Optimal gradient descent learning for bidirectional associative memories,” *Electronics Letters*, vol. 29, no. 17, pp. 1556–1557, 1993.

[15] G. Zheng, S. N. Givigi, and W. Zheng, *A New Strategy for Designing Bidirectional Associative Memories*, vol. 3496 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 2005.

[16] D. Shen and J. B. Cruz Jr., “Encoding strategy for maximum noise tolerance bidirectional associative memory,” *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 293–300, 2005.

[17] S. Arik, “Global asymptotic stability analysis of bidirectional associative memory neural networks with time delays,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 580–586, 2005.

[18] G. X. Ritter, J.L. Diaz-de-Leon, and P. Sussner, “Morphological bidirectional associative memories,” *Neural Networks*, vol. 12, no. 6, pp. 851–867, 1999.

[19] Y. Wu and D. A. Pados, “A feedforward bidirectional associative memory,” *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 859–866, 2000.

[20] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Germany, 1999.

[21] R. K. Rajapakse and M. Denham, “Information retrieval model using concepts lattices for content representation,” in *Proceedings of the FCA KDD Workshop of the 15th European Conference on Artificial Intelligence (ECAI '02)*, Lyon, France, July 2002.

[22] Kohonen, “Correlation matrix memories,” *IEEE Transactions on Computers*, vol. 21, no. 4, pp. 353–359, 1972.

[23] G. X. Ritter, P. Sussner, and J.L. Diaz-de-Leon, “Morphological associative memories,” *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 281–293, 1998.

[24] C. Yáñez-Márquez and J. L. Díaz de León-Santiago, “Memorias asociativas basadas en relaciones de orden y operaciones binarias,” *Computación y Sistemas*, vol. 6, no. 4, pp. 300–311, 2003.

[25] C. Yáñez-Márquez, *Associative Memories Based on Order Relations and Binary Operators (In Spanish)*, Ph.D. thesis, Center for Computing Research, México, 2002.

[26] B. Kosko, “Bidirectional associative memories,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[27] J. H. Park, “Robust stability of bidirectional associative memory neural networks with time delays,” *Physics Letters A*, vol. 349, no. 6, pp. 494–499, 2006.

[28] M. E. Acevedo, *Alpha-beta bidirectional associative memories (In Spanish)*, Ph.D. thesis, Center for Computing Research, México, 2006.