Int. J. Open Problems Compt. Math., Vol. 4, No. 2, June 2011
ISSN 1998-6262; Copyright ©ICSRS Publication, 2011
www.i-csrs.org

Algebraically Coherent Languages for Efficient Algorithm Construction. Universal Languages and Optimal Principle.

Juan-Esteban Palomar Tarancón

Dep. Math. Inst. Jaume I, 12530-Burriana-(Castellón), Spain.
e-mail: jepalomar.tarancon@gmail.com

Abstract

The author uses some methods of categorical algebra in order to state the properties that optimal languages must satisfy for allowing efficient algorithm construction. Likewise, the universal-language concept is introduced. By virtue of its properties, any universal language can work fine as a bridge between each couple of natural languages in translation algorithms. It is worth pointing out, that this is only an introductory paper suggesting a wide class of open problems, and the same nature of these problems prevent us from solving them in a single paper.

Keywords: *Universal language, optimal principle, efficient algorithms* [1].

1 Introduction

Consider the roman numeral system Rom $= \{I, V, X, L, C, D, M\}$ to denote integer sets. Using Rom-notation in order to compute sums, products, roots, etc. involves an unnecessary complexity. Indeed, it is a good idea to perform some subroutine translating every roman integer to a binary one, and then computing algorithms can work with binary digits. This translation allows to perform more efficient algorithms. This circumstance arises from the existence of a morphism family $f_1 : \mathbb{Z} \to \mathbb{Z}_2$, $f_2 : \mathbb{Z} \to \mathbb{Z}_4 \ldots f_n : \mathbb{Z} \to \mathbb{Z}_{2^n}$ from any binary integer into the number represented by their tail. For instance, the

[1] *Mathematical Subject classification: 94A99,18B35*

following equivalences

$$
\begin{aligned}
1001100101 &\equiv 101(\mod 2)\\
1001100101 &\equiv 101(\mod 2^2)\\
1001100101 &\equiv 101(\mod 2^3)
\end{aligned}
$$

induced by these kind of morphisms, can be obtained removing the first digits in the word 1001100101. Indeed, in order to build algorithms, not only programming languages matter, but also the structure of those languages representing the involved real world objects and actions are also essential in efficient algorithm construction. Computer science has been enriched by means of different programming languages together with several languages denoting document structures like XML. However, an efficient language to denote real world objects and actions only for members of numeric sets and logical connectives are available, for instance the binary numeral system. This paper aims to investigate the algebraic structure of efficient languages, in order to denote real world objects and actions, by means of which efficient algorithms can be defined, in particular, translation algorithms between natural languages.

Since algorithms work over words of some alphabet, it is more effective to deal with a language such that each word equivalence corresponds to a logical one between the represented objects, that is to say, the map μ sending each word into the corresponding denoted object, must preserve the involved algebraic language-structures as much as possible. Unfortunately, natural languages do not fit into this model, because they are the result of arbitrary conventions.

It is easy to see, why the underlying language in any algorithm matters. For instance, the English syntactic rule imposing a gerund after a preposition can be deduced by observing the phrases “while reading” and “of knowing”, whenever the meaning of these phrases are known, because there is a conceptual equivalence between “while” and “of”, since both are prepositions. However, there is no algebraic equivalence between both words, both being regarded as members of the free-monoid generated by the involved alphabet. By contrast, suppose that there is a language $\mathcal{P}$ in which the prepositions “while” and “of” are denoted by the words “xxp” and “xsp”, respectively, and the gerunds “reading” and “knowing” by the words “xxvg” and “xwvg”, besides, the concept “gerund” is denoted in $\mathcal{P}$ as “vg”. There is the word-equivalence between “xxp” and “xsp” consisting of having the same last letter “p”; and there is also a similar equivalence between “xxvg” and “xwvg”. In addition, the concept of “gerund” containing both words “xxvg” and “xwvg”, as particular cases, is denoted by the letters “vg” determining such an equivalence. In general, to know that both English words “of” and “while“ are prepositions, it is required to know their meanings, however an algorithm knows nothing about the meaning of the words it deals with. By contrast, in the artificial language $\mathcal{P}$, an algorithm can classify both words "xsp" and "xxp" as prepositions, without knowing their meanings, observing the only final letter "p". This is why efficient algorithms can be only performed using artificial languages such that the map sending words and sentences into their meanings preserves the underlying algebraic structures. In next sections, this kind of languages are defined accurately and described as coherent languages. In addition, basic concepts and methods are introduced too. To this end, a conceptual system $\Omega_{\mathfrak{P}}$ will be defined, this being structured by means of a transitive-reflexive relation denoted as $\preceq$. For every couple of concepts X and Y, the relation $X \preceq Y$ means Y to be an abstraction of X. Analogously, a language algebra $\mathcal{L}$ equipped with another transitive-reflexive relation $\sqsubseteq$ will be defined too. For every couple of words w_1 and w_2 in $\mathcal{L}$, the relation $w_1 \sqsubseteq w_2$ holds provided that w_2 is the tail of w_1. The underlying set of $\mathcal{L}$ consists of a set of words $\mathrm{Dic}(\mathcal{L})$ together with a collection $\mathrm{Sen}(\mathcal{L})$ of word-sequences, that is to say, sentences.

Morphisms between both algebraic constructions $(\mathcal{L}, \sqsubseteq)$ and $(\Omega_{\mathfrak{P}}, \preceq)$ are preserving-structure maps. These structures lead to the definition of the concepts of applied dictionary and applied language as follows. A 5-tuple $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq, \Omega_{\mathfrak{P}}, \preceq, \mathcal{M})$ is an applied dictionary, provided that $(\Omega_{\mathfrak{P}} \preceq)$ is a conceptual system and $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq)$ is a language algebra. In addition, $\mathcal{M} \subseteq \mathcal{L} \times \Omega_{\mathfrak{P}}$ is a binary relation satisfying the following property. For every word $w \in \mathcal{L}$ and each concept $O \in \Omega_{\mathfrak{P}}$, the relation $(w, O) \in \mathcal{M}$ is true if and only if w denotes O. Thus, if $\mathcal{M}$ is a mapping, then it sends each word w of $\mathcal{L}$ into its meaning. In this case, the map-image uniqueness implies $\mathcal{L}$ to be unambiguous. The concept of applied language is obtained by extension, that is to say, extending the mapping $\mathcal{M}$ to word-sequences or sentences. An unambiguous applied language is coherent provided that $\mathcal{M}$ is a morphism. As we shall show, coherent applied languages are the adequate devices for efficient algorithm construction.

In the last sections, we investigate the problem of efficient translation algorithms between couples of natural and artificial languages by means of an intermediate one $\mathcal{L}_0$. To this end, it is a good method each language $\mathcal{L}$ to be equipped with a translating algorithm between $\mathcal{L}_0$ and $\mathcal{L}$, and then $\mathcal{L}_0$ works as a bridge between each couple of natural languages. Such a strategy is widely accepted among programming languages, like Java, in which it is used the intermediate one commonly referred to as Java bytecode, in order to translate applications among different operating systems.

One of our main results will show, that if $\mathfrak{T}$ denotes a functor sending each object $\mathcal{L}_P$, consisting of word-sequences and patterns of a language $\mathcal{L}$, into the collection $\mathfrak{T}(\mathcal{L}_P)$ of its sentences, then there is a language $\mathcal{L}_0$, determined by means of a structured arrow $\mathcal{L}_0 \xrightarrow{i_{\mathcal{L}_0}} \mathfrak{T}(\mathcal{L}_{0,P})$ satisfying some universal property. Such a language $\mathcal{L}_0$ is unique up to isomorphisms. Since $\mathcal{L}_0$ is determined by a universal property, then it is justified to describe it as universal language. As a consequence, any universal language, defined in a wide class, is syntax-free and can be built in a coherent way. In addition, $\mathcal{L}_0$ satisfies each property of any other language lying in the same category, whenever it is preserved under morphisms. It is just by virtue of these properties that a universal language is a good choice to work as intermediate among natural languages and among artificial ones too. We will detail some language optimization method by means of which it is possible to discern what are the structure properties that the best language must satisfy for each purpose. The same method shows that optimal languages are universal too.

Finally, it is worth mentioning, that this paper is intended as an introductory one suggesting new open problems to be solved, which will be proposed in the last section.

2 Conceptual algebra

Languages we are dealing with are partial free monoids each member of which is equipped with a meaning. Since the meaning of a word w is nothing but the concept c denoted by w, there is a map f sending w into c. We say a language to be coherent provided that f preserves the involved algebraic structure. To define preserving-structure maps between languages and conceptual systems, it is required a conceptual algebra. In [8] it is introduced a conceptual system and algebra of analogies, the definition of each of its objects consists of predicates lying in a class $\mathfrak{P}$ being stable under conjunctions and disjunctions. Morphisms between two concepts X and Y are transitive and reflexive relations, denoted as $\overset{\mathfrak{N}}{\preceq}$; where $\mathfrak{N}$ is a subclass of $\mathfrak{P}$. However, for our purposes it is sufficient a particular case, consisting of the subcategory such that, for every $\overset{\mathfrak{N}}{\preceq}$, the class $\mathfrak{N}$ coincides with $\mathfrak{P}$. Thus, since there is no confusion, the transitive-reflexive relation $\overset{\mathfrak{N}}{\preceq}$, will be denoted, simply, as $\preceq$, which it is defined in the following paragraphs, hence this paper is self-contained.

Let $\mathfrak{P}$ denote a class of predicates being stable under conjunctions and disjunctions. Assume, as an axiom, that $\mathfrak{P}$ contains at least one tautology. With these assumptions, say X to be a $\mathfrak{P}$-defined object, provided that there is a predicate $\mathrm{Def}_X(O) \in \mathfrak{P}$ specifying X, that is to say, for every object Y, the relation $\mathrm{Def}_X(Y) = True$, implies $X = Y$. Of course, $\mathrm{Def}_X(X) = True$. To denote this fact, we shall say $\mathrm{Def}_X(O)$ to be a deductive definition for X.

Definition 2.1 *For every couple of $\mathfrak{P}$-defined objects X and Y, the object Y is an abstraction of X, and X a concretion of Y, if and only if there is a predicate $p(O) \in \mathfrak{P}$ satisfying the following relation*

$$\mathrm{Def}_X(O) \Leftrightarrow (\mathrm{Def}_Y(O) \wedge p(O))$$

for every $\mathfrak{P}$-defined object O.

For example, consider the defined objects "triangle" and "polygon" together with the predicate

$$p(O) = \text{“}O \text{ contains three and only three angles”}$$

Indeed,

$$\mathrm{Def}_{\text{“triangle”}}(O) \Leftrightarrow (\mathrm{Def}_{\text{“polygon”}}(O) \wedge p(O))$$

therefore, the concept of “polygon” is an abstraction of the “triangle” notion.

Henceforth, to denote Y to be an abstraction of X write $X \preceq Y$, therefore in the preceding example we can write “triangle” $\preceq$ “polygon”.

Lemma 2.2 *The relation $\preceq$ is reflexive and transitive.*

Proof. On the one hand, since $p(X)$ can be a tautology, then for every defined object X it is true that $X \preceq X$, hence the relation $\preceq$ is reflexive. On the other hand, both relations $X \preceq Y$ and $Y \preceq Z$ imply the existence of two predicates $p_1(O)$ and $p_2(O)$ such that, for every $\mathfrak{P}$-defined object O,

$$\mathrm{Def}_X(O) \Leftrightarrow \mathrm{Def}_Y(O) \wedge p_1(O) \tag{1}$$

$$\mathrm{Def}_Y(O) \Leftrightarrow \mathrm{Def}_Z(O) \wedge p_2(O) \tag{2}$$

therefore,

$$\mathrm{Def}_X(O) \Leftrightarrow \mathrm{Def}_Z(O) \wedge (p_1(O) \wedge p_2(O)) \tag{3}$$

Since by assumption $\mathfrak{P}$ is stable under conjunctions, then $(p_1(O) \wedge p_2(O))$ belongs to $\mathfrak{P}$; hence, $X \preceq Z$ and the lemma follows.

Now, denoting as $\Omega_{\mathfrak{P}}$ the class of all $\mathfrak{P}$-defined objects, the pair $(\Omega_{\mathfrak{P}}, \preceq)$ is a category such that for every couple of objects X and Y the corresponding hom-set $\hom(X, Y)$ either is empty or it is the singleton $\{X \preceq Y\}$, that is to say, $X \to Y$ if and only if $X \preceq Y$.

Let $\Omega_{\mathfrak{P}}(X)$ denote the predicate “X is a $\mathfrak{P}$-defined object”. From now on, assume $\Omega_{\mathfrak{P}}(X) \in \mathfrak{P}$, hence the concept of $\mathfrak{P}$-defined object is also a $\mathfrak{P}$-defined one. Henceforth, denote such an object as $\Omega_{\mathfrak{P}}^{\curlyvee}$. Since, by definition, every $\mathfrak{P}$-defined object satisfies the predicate $\Omega_{\mathfrak{P}}(X)$, then for every $X \in \Omega_{\mathfrak{P}}$ the relation

$$\mathrm{Def}_X(X) \Leftrightarrow \big(\mathrm{Def}_{\Omega_{\mathfrak{P}}}(X) \wedge \mathrm{Def}_X(X)\big) = \Omega_{\mathfrak{P}}(X) \wedge \mathrm{Def}_X(X)$$

holds; therefore

$$\forall X \in \Omega_{\mathfrak{P}} : \quad X \preceq \Omega_{\mathfrak{P}}^{\curlyvee} \tag{4}$$

It is a known fact [1], that in any category associated to a preordered set $(\Omega_{\mathfrak{P}}, \preceq)$, for every set-family $\mathbf{X} = \{X_i \mid i = 1, 2 \ldots n\}$ there is the co-product

$$\mathbf{X}^{\curlyvee} = \coprod_{0<i\leq n} X_i$$

whenever there is an element $\Omega_{\mathfrak{P}}^{\curlyvee}$ satisfying (4). From now on, denote such a co-product by means of the binary operator $\curlyvee$, that is to say,

$$\mathbf{X}^{\curlyvee} = \coprod_{0<i\leq n} X_i = X_1 \curlyvee X_2 \curlyvee \cdots \curlyvee X_n$$

Since by definition, for every $i \leq n$ there is the relation $X_i \preceq \mathbf{X}^{\curlyvee}$, say $\mathbf{X}^{\curlyvee}$ to be the abstraction of the set $\mathbf{X}$. Likewise, say the X_i to be concretions of $\mathbf{X}^{\curlyvee}$.

When two $\mathfrak{P}$-defined objects X and Y satisfy some predicate $p(x)$, this fact can be regarded as an analogy between them. If $p(x)$ is equivalent to the definition $\mathrm{Def}_Z(x)$ of some object Z, obviously both relations $X \preceq Z$ and $Y \preceq Z$ hold. Thus, the existence of the co-product $X \curlyvee Y$ implies an analogy between X and Y. By the nature of co-product concept, this is just, the strongest analogy that can be found between both objects.

Using a lighter notation we can state the following. Let X, Y and Z be three $\mathfrak{P}$-defined objects and $p(O)$ a predicate in $\mathfrak{P}$ being equivalent to the definition $\mathrm{Def}_Z(O)$ of Z. If both relations $X \preceq Z$ and $Y \preceq Z$ hold, we shall denote this circumstance writing $X \overset{p(x)}{=} Y$.

It is not difficult to see, $\overset{p(x)}{=}$ to be an equivalence relation, because $\preceq$ is reflexive and transitive. The symmetry of $\overset{p(x)}{=}$ is consequence of the required logical conjunction $(X \preceq Z) \wedge (Y \preceq Z)$ which is also a symmetric formula. Such an equivalence denotes some analogy between X and Z. Since, by assumption, $p(x)$ is equivalent to $\mathrm{Def}_Z(x)$, both notations $X \overset{p(x)}{=} Y$ and $X \xlongequal{\mathrm{Def}_Z(x)} Y$ are adequate.

Abstractions, concretions together with analogies are the elements structuring the conceptual algebra we are dealing with. Since both natural and artificial languages are used to denote concepts, language algebra will be investigated in the next section.

3 Language algebra

Languages we are dealing with are structured sets consisting of word sequences. Thus, the members of a language $\mathcal{L}$ are words of a partial free-monoid generated by any finite alphabet A. Henceforth, assume as an axiom every language to contain the empty word $\varnothing$. The set of all words of any language $\mathcal{L}$ will be denoted as $\mathrm{Dic}\,(\mathcal{L})$, and it will be termed as the dictionary of $\mathcal{L}$. In any dictionary $\mathrm{Dic}\,(\mathcal{L})$ we define a reflexive and transitive relation $\sqsubseteq$ as follows.

Definition 3.1 *For every pair of non-empty words $w = a_1a_2 \dots a_n$ and $b = b_1b_2 \dots b_m$ of a language $\mathcal{L}$, the relation $w \sqsubseteq v$ is true if and only if $n \geq m$, besides, $a_{j+1}a_{j+2} \dots a_{j+m} = b_1b_2 \dots b_m$; where $j = n - m$. In addition, if $\varnothing$ denotes the empty-word, then $\forall w \in$
$mathrmDic(\mathcal{L}) : w \sqsubseteq \varnothing$.*

Lemma 3.2 *The relation $\sqsubseteq$ is reflexive and transitive.*

Proof. By definition $\varnothing \sqsubseteq \varnothing$. If $w = a_1a_2 \dots a_n$ is any nonempty word, since $\forall i \leq n : a_i = a_i$ then the relation $w \sqsubseteq w$ holds; therefore $\sqsubseteq$ is reflexive. To see that $\sqsubseteq$ is also transitive, consider three nonempty words $w = a_1a_2 \dots a_n$, $v = b_1b_2 \dots b_m$ and $u = c_1c_2 \dots c_k$ and suppose that $w \sqsubseteq v$ and $v \sqsubseteq u$. By definition, we have that

$$a_{j+1}a_{j+2} \dots a_{j+m} = b_1b_2 \dots b_m \tag{5}$$
$$b_{r+1}b_{r+2} \dots b_{r+k} = c_1c_2 \dots c_k \tag{6}$$

where $j = n - m$ and $r = m - k$. From the preceding relations it follows that $a_{t+1}a_{t+2} \dots a_{t+m} = c_1c_2 \dots c_k$ for $t = n - k$, hence $w \sqsubseteq u$, and the lemma follows.

According to the preceding lemma, the pair $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq)$ is a category such that for every couple of words w and v, the set $\hom(w, v)$ either is empty or it is the singleton $\{w \sqsubseteq v\}$. From now on, to avoid exceptions we shall consider that every word ends with the empty letter sequence $\varnothing$; consequently, for every word w the relation $\mathbf{w} \sqsubseteq \varnothing$ holds. As in the conceptual systems defined above, for every couple of words $\mathbf{w}$ and $\mathbf{v}$, both relations $w \sqsubseteq u$ and $v \sqsubseteq u$ can be interpreted as an analogy between w and v, for any word u. Henceforth, denote such an analogy as $w \overset{u}{=} v$.

Definition 3.3 *An applied dictionary is a 5-tuple $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq, \Omega_{\mathfrak{P}}, \preceq, \mathcal{M})$, such that $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq)$ is a dictionary in a language $\mathcal{L}$, $(\Omega_{\mathfrak{P}}, \preceq)$ is a conceptual system and $\mathcal{M} \subseteq \mathrm{Dic}(\mathcal{L}) \times \Omega_{\mathfrak{P}}$ a binary relation satisfying the following property. If w is a word of $\mathrm{Dic}(\mathcal{L})$ and $O \in \Omega_{\mathfrak{P}}$ a $\mathfrak{P}$-defined object, then $(w, O) \in \mathcal{M}$ if and only if the meaning assigned to w is $O \in \Omega_{\mathfrak{P}}$*

Definition 3.4 *An applied dictionary $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq, \Omega_{\mathfrak{P}}, \preceq, \mathcal{M})$ is unambiguous, provided that $\mathcal{M}$ is a mapping.*

It is worth noticing, that according to the former definition, in any unambiguous dictionary each word has only a meaning.

Definition 3.5 *A dictionary $(\mathrm{Dic}(\mathcal{L}), \sqsubseteq, \Omega_{\mathfrak{P}}, \preceq, \mathcal{M})$ is algebraically coherent provided that $\mathcal{M}$ preserves both structures of $(\mathcal{L}, \sqsubseteq)$ and $(\Omega_{\mathfrak{P}}, \preceq)$.*

Disambiguation of any ambiguous applied dictionary can be performed enriching each word with attributes. Attributes can be adjectives or keywords restricting the meaning of any word, together with those terms denoting the context under which each word occurs. For instance, "English", "Hamlet", "preposition", "polite", "verb", "ironic", "buzzword", "jargon", "future","plural", etc. can be word attributes. Denote word attributes adding symbols to any word and using the symbol "@" to separate each word from its attributes. For example, the word "staff" can be equipped with both attributes "noun" and "verb", hence "staff@noun" and "staff@verb" are disambiguations of this word. From now on, we are supposing that all applied dictionaries we are dealing with, are unambiguous.

Dictionaries form a category **Dic** each object of which is an applied dictionary and morphisms between two objects $(\mathrm{Dic}\,(\mathcal{L}_\infty)\,, \sqsubseteq, \Omega_{\mathfrak{P}}, \preceq, \mathcal{M}_1)$ and $(\mathrm{Dic}\,(\mathcal{L}_\in)\,, \sqsubseteq, \Omega_{\mathfrak{P}}, \preceq, \mathcal{M}_2)$ are all maps $f : \mathcal{L}_1 \to \mathcal{L}_2$ preserving meanings, that is to say, f is a morphism provided that

$$\forall w \in \mathcal{L}_1 : \quad \mathcal{M}_1(w) = \mathcal{M}_2\,(f(w)) \tag{7}$$

From now on, assume that any dictionary only contains words in singular and infinitive tense; consequently plurals and other verb tenses will be denoted by means of attributes. In fact, dictionaries of natural languages are performed in this way, and attributes are contained in definitions implicitly.

Lemma 3.6 *For every* **Dic***-morphism* $f : \mathcal{L}_1 \to \mathcal{L}_2$ *and for each pair* (w_1, w_2) *lying in* $\mathcal{L}_1 \times \mathcal{L}_1$ *the following statements hold.*

1. *If* $\mathcal{M}\,(f(w_1)) \curlyvee \mathcal{M}\,(f(w_2))$, *then this is equal to* $\mathcal{M}(w_1) \curlyvee \mathcal{M}(w_2)$.
2. *The relation* $\mathcal{M}(w_1) \preceq \mathcal{M}(w_2)$ *holds if and only if* $\mathcal{M}(f(w_1)) \preceq \mathcal{M}(f(w_2))$.
3. *For every predicate* $p(x)$, *there is the analogy* $\mathcal{M}(w_1) \overset{p(x)}{=\!=} \mathcal{M}(w_2)$ *if and only if* $\mathcal{M}\,(f(w_1)) \overset{p(x)}{=\!=} \mathcal{M}\,(f(w_2))$.

Proof. By definition, each of the relations $\preceq$, $\curlyvee$ and $\overset{p(x)}{=\!=}$ depends only upon the meaning of the involved symbols, and morphisms preserve meanings.

3.1 Efficient algorithms

Consider a family $\mathbf{F}$ of maps the domain and co-domain of each of which is a set X. Suppose that there is a family of abstractions

$$\mathbf{A} = \{X \preceq A_{ij} \mid i \in I, j \in J_i\}$$

where both J_i and I are countable index sets, for every index i. Since each analogy $\stackrel{\mathrm{Def}(A_{ij})}{=\!=}$ is an equivalence relation, then there are the corresponding quotient sets. If for short we denote each $\stackrel{\mathrm{Def}(A_{ij})}{=\!=}$ as $\mathcal{R}_{ij}$, then the corresponding quotient set family is $\mathbf{Q} = \{X/\mathcal{R}_{ij} \mid (i,j) \in I \times J_i\}$. These assumptions lead to state the following axiom.

Axiom 3.7 *For every $i \in I$, there is a mapping $\rho_i : X \to J_i$ such that*

$$x = \bigcap_{i \in I} [x]_{\mathcal{R}_{i\rho_i(x)}} \tag{8}$$

where $[x]_{\mathcal{R}_{i\rho_i(x)}}$ denotes the $\mathcal{R}_{ij}$-equivalence class containing x.

Lemma 3.8 *If a set X satisfies Axiom 3.7 with respect to a family of analogies $\mathcal{R}_{ij}$, then there is a coherent language to denote the members of X.*

Proof. By hypothesis, for every $i \in I$ there is the map $\rho_i : X \to J_i$ satisfying (8). If for each $i \in I$, A_i is an alphabet equipollent to J_i, then there is a bijection $\mathcal{M} : J_i \to A_i$ and each member α of A_i denotes a member $\mathcal{M}(\alpha)$ of J_i. Likewise, if $I = \{1, 2 \ldots k\}$, then for every $i \in I$ each member $\alpha_{i\rho_i(x)}$ of A_i can denote the equivalence-class $[x]_{\mathcal{R}_{i\rho_i(x)}}$; therefore, taking into account (8), the word $\alpha_{k\rho_k x}\alpha_{(k-1)\rho_{(k-1)}}(x) \ldots \alpha_{1\rho_1(x)}$ can denote x in a coherent way, because each letter represents an equivalence class, and by virtue of (8),

$$x = \mathcal{M}(\alpha_{k\rho_k(x)}) \cap \mathcal{M}(\alpha_{(k-1)\rho_{(k-1)}(x)}) \cap \cdots \mathcal{M}(\alpha_{1\rho_1(x)})$$

Thus, using the alphabet $A = \cup_{i \in I} A_i$ the members of X can be denoted by k-letter words. Likewise, every n-letter word $\alpha_{nj_n}\alpha_{(n-1)j_{n-1}} \ldots \alpha_{1j_1}$, with $n \leq k$, denotes the equivalence class intersection $\mathcal{M}(\alpha_{nj_n}) \cap \mathcal{M}(\alpha_{(n-1)j_{(n-1)}}) \cap \cdots \mathcal{M}(\alpha_{1j_1})$, Accordingly, for every pair of words $\mathbf{w}_1 = \alpha_{kj_k}\alpha_{(k-1)j_{k-1}} \cdots \alpha_{1j_1}$ and $\mathbf{w}_2 = \beta_{kj_k}\beta_{(k-1)j_{k-1}} \ldots \beta_{1j_1}$, if for some $n \leq k$ it is true that

$$\mathbf{u} = \alpha_{nj_n}\alpha_{(n-1)j_{n-1}} \ldots \alpha_{1j_1} = \beta_{nj_n}\beta_{(n-1)j_{n-1}} \ldots \beta_{1j_1}$$

the following relations are true

$$\mathbf{w}_1 \sqsubseteq \mathbf{u} \tag{9}$$
$$\mathbf{w}_2 \sqsubseteq \mathbf{u} \tag{10}$$
$$\mathbf{w}_1 \stackrel{\mathbf{u}}{=} \mathbf{w}_2 \tag{11}$$
$$\tag{12}$$

and, analogously, let $Z = \mathcal{M}(\alpha_{nj_n}) \cap \mathcal{M}(\alpha_{(n-1)j_{(n-1)}}) \cap \cdots \mathcal{M}(\alpha_{1j_1})$,

$$\mathcal{M}(\mathbf{w}_1) \preceq Z \tag{13}$$
$$\mathcal{M}(\mathbf{w}_2) \preceq Z \tag{14}$$
$$\mathcal{M}(\mathbf{w}_1) \stackrel{\mathrm{Def}_Z(x)}{=\!=} \mathcal{M}(\mathbf{w}_2) \tag{15}$$

therefore the map $\mathcal{M}$ preserves both structures.

In the following examples we describe some instances of coherent languages based upon some well-known equivalences of number theory.

Example. Consider the set of natural numbers $X = \{0, 1, 2 \dots 29\}$. Let $p_1 = 2$, $p_2 = 3$ and $p_3 = 5$ the three first primes, and let $J_i = \{1, 2 \dots p_i | i \in I\}$ for $I = \{1, 2, 3\}$. Consider the family of predicates

$$\mathbf{P} = \{p_{ij}(x) = (x \equiv j \mod (p_i)) \, | i \in I, j < i\}$$

Let $J_1 = \{0_2, 1_2\}$ be the alphabet each member of which j_2 denotes the equivalence class $j_2 = \{x \in X | x \equiv j \mod (2)\}$; likewise each member j_3 of $J_2 = \{0_3, 1_3, 2_3\}$ denotes the equivalence class $j_3 = \{x \in X | x \equiv j \mod (3)\}$, and each member j_5 of the alphabet $\{0_5, 1_5, 2_5, 3_5, 4_5\}$ denotes the equivalence class $j_5 = \{x \in X | x \equiv j \mod (5)\}$. With these assumptions each three-letter word denotes a member of X, for example, denoting as $\mathcal{M}$ the map sending each word into its meaning, we have that

$$3 = \mathcal{M}(3_5 0_3 1_2) \tag{16}$$

$$4 = \mathcal{M}(4_5 1_3 0_2) \tag{17}$$

$$13 = \mathcal{M}(3_5 1_3 1_2) \tag{18}$$

because $3 \equiv 1(\mod 2)$, $3 \equiv 0(\mod 3)$, $3 \equiv 3(\mod 5)$ etc.

Now, let $A = J_1 \cup J_2 \cup J_3$, it is not difficult to see, that the applied dictionary $\mathrm{Dic}(\mathcal{L}) = (A, \sqsubseteq, X \subset \Omega_{\mathfrak{P}}, \preceq, \mathcal{M})$ is coherent, because each word denotes an equivalence-class. In addition, it does not matter the order in which each letter occurs in each word. To see this fact, let z stand for the concept of member of X being divisible by both 2 and 3, that is to say, $z = \mathcal{M}(0_3 0_2)$. The definition $\mathrm{Def}(z)$ is equivalent to $p_{10}(x) \wedge p_{20}$. Indeed, for every $j_5 \in J_3$, the relation $\mathcal{M}(j_5 0_3 0_2) \preceq \mathcal{M}(0_3 0_2)$ holds, because the definition of $j_5 0_3 0_2$ is equivalent to $p_{5j}(x) \wedge p_{30}(x) \wedge p_{20}(x)$ and

$$p_{5j}(x) \wedge p_{30}(x) \wedge p_{20}(x) \Leftrightarrow \mathrm{Def}_z(x) \wedge p_{5j}(x) = (p_{30}(x) \wedge p_{20}(x)) \wedge p_{5j}(x)$$

The proof for any other pair of words is similar.

Now, say a map $f : X \to X$ in $\mathbf{F}$ to be compatible with an applied dictionary $(A, \sqsubseteq, X \subset \Omega_{\mathfrak{P}}, \preceq, \mathcal{M})$, provided that it preserves the equivalence classes denoted by its words; consequently there is a family of maps $F_f = \{f_i : A_i \to A_i \, | \, i \in I\}$ such that for every $x \in X$, if $w_1 w_2 \dots w_k$ is the word denoting x then

$$f(x) = \bigcap_{i \in I} \mathcal{M}\left(f_i(w_i)\right)$$

The language defined in the example above, allows to construct efficient algorithms in order to evaluate polynomials in integer rings. The following example illustrates this fact.

Example. Consider the set X in the preceding example and the map $f(x) = x^2 + 4$. Since $f(x)$ preserves the equivalence classes "$\equiv \mod (p)$", the corresponding quotient maps $f_1 : \mathbb{Z}_2 \to \mathbb{Z}_2$, $f_2 : \mathbb{Z}_3 \to \mathbb{Z}_3$ and $f_3 : \mathbb{Z}_5 \to \mathbb{Z}_5$ can be defined also by means of the polynomials $f_1(x) = x^2$, $f_2(x) = x^2 + 1_3$ and $f_3(x) = x^2 + 4_5$, respectively. To compute $f(3)$ we can evaluate each of the maps $f_1(1_2) = 1_2$, $f_2(0_3) = 1_3$ and $f_3(3_5) = 3_5$ obtaining $f(3) = \mathcal{M}(3_5 1_3 1_2) = 13$, consequently $f(3) = 13$.

An algorithm to compute a map can consists of a hash-table and a searching algorithm by means of which each value x of its argument can be found in the first column, and in the same row, but in the second column the corresponding value of $f(x)$ can be obtained. Now, a table to compute a map $f(x)$ being compatible with the applied dictionary defined in the former example, must contain 30 rows. By contrast, a table to compute $f_1 : \mathbb{Z}_2 \to \mathbb{Z}_2$ is a two-row one; a table to compute f_2 can consist of 3 rows, and f_3 can be computed through a table containing 5 rows. Thus, the three maps f_1, f_2 and f_3 can be computed with a table of $2 + 3 + 5 = 10$ rows, while to compute f it is required a table of size 30. Likewise, each integer n can be represented as a word $w_p \ldots w_5 w_3 w_2$ such that $(w_p, \ldots w_5, w_3, w_2) \in \mathbb{Z}_p \times \cdots \mathbb{Z}_5 \times \mathbb{Z}_3 \times \mathbb{Z}_2$. To represent integers such a language is algebraically coherent; however, in order to represent real number sets it is not, because the structure of $\mathbb{R}$ involves the concept of limit and the standard topology.

In general, algorithms can be split into maps, therefore using coherent languages it is possible to perform more efficient algorithms for those maps being compatible with the analogy relations of their domains. To this end, each algorithm must contain a translation one to denote each defined object by means of a coherent language. Accordingly, efficient translation systems must be investigated, which can be also applied to perform translations among natural languages. Next sections are devoted to this topic.

4 Applied languages

An applied language is a 6-tuple $\mathcal{L} = (\mathrm{Dic}(\mathcal{L}), \mathrm{Sen}(\mathcal{L}), \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu)$; where $\mathrm{Dic}(\mathcal{L})$ is an applied dictionary; $\mathrm{Sen}(\mathcal{L})$ is a set of sentences, that is to say, finite word sequences in $\mathrm{Dic}(\mathcal{L})$ having a meaning; both $\preceq$ and $\sqsubseteq$ are the relations defined above; and $\mu : \mathrm{Sen}(\mathcal{L}) \to \Omega_{\mathfrak{P}}$ is a map sending each sentence into its meaning. From now on, we are assuming that each word is enriched with attributes in order to avoid any ambiguity. Likewise, since each word of an applied dictionary has a meaning, we assume as an axiom that each word is also a one-word sentence; consequently $\mathrm{Dic}(\mathcal{L}) \subseteq \mathrm{Sen}(\mathcal{L})$, for any language.

The class of all applied languages form a category **AppLng** the morphisms of which are all maps between sentence sets preserving both meanings and attributes. Thus, for every couple of applied languages $\mathcal{L}_1$ and $\mathcal{L}_2$, a map

$f : \mathrm{Sen}(\mathcal{L}_1) \to \mathrm{Sen}(\mathcal{L}_2)$ is a morphism, provided that each sentence $s \in \mathrm{Sen}(\mathcal{L}_1)$ has the same meaning as its image $f(s) \in \mathrm{Sen}(\mathcal{L}_2)$. In addition, the attribute family of s is a subset of the attributes of $f(s)$. Since we are assuming the relation $\mathrm{Dic}\,(\mathcal{L}) \subseteq \mathrm{Sen}\,(\mathcal{L})$, then $\mathcal{L}_0 = (\mathrm{Dic}\,(\mathcal{L})\,, \mathrm{Dic}\,(\mathcal{L})\,, \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu_0)$ is a sub-language of $\mathcal{L} = (\mathrm{Dic}\,(\mathcal{L})\,, \mathrm{Sen}\,(\mathcal{L})\,, \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu)$ and the inclusion map $i_{\mathcal{L}_0} : \mathcal{L}_0 \to \mathcal{L}$ a morphism; where $\mu_0 = \mu|_{\mathrm{Dic}(\mathcal{L})}$ denotes the restriction of μ to $\mathrm{Dic}\,(\mathcal{L})$.

Disregarding idioms, the meaning of each sentence $w_1 w_2 \dots w_n$, is built from the meanings of the words in a subset $\{w_{r_1}, w_{r_2} \dots w_{r_k}\}$ of $\{w_1, w_2 \dots w_n\}$, with $k \leq n$, by means of some procedure P. For instance, in the English sentence $s =$"he does not write", the meaning depends only on the words "he" and "write", while the words "does" and "not" form the structure of this sentence. To see this fact, consider the sentence "she does not read". The later has the same structure as s, but the meaning depends on the words "she" and "read". In fact, the sentence structure "$f(x, y) = x$ does not y" works as a map sending the values of x and y into the meaning of the underlying sentence. Of course, this map $f(x, y)$ is nothing but an English pattern. From this viewpoint, a language is a collection of words and maps, that is to say, patterns, and each pattern determines a procedure by means of which the meaning of each sentence in its image must be built. Henceforth, say such a procedure to be a meaning-constructor.

The concept of continuity can be applied to patterns as follows. A pattern $f(x_1, x_2 \dots x_n)$ is continuous, provided that for every couple of n-tuples $(x_1, x_2 \dots x_n)$ and $(y_1, y_2 \dots y_n)$, if both $f(x_1, x_2 \dots x_n)$ and $f(y_1, y_2 \dots y_n)$ are sentences, then for every n-tuple $(z_1, z_2 \dots z_n)$ the statement

$$\forall i \leq n : \quad x_i \preceq z_i \preceq y_i$$

implies $f(z_1, z_2 \dots z_n)$ to be a sentence too. In natural languages, continuity is widely assumed, because substituting a word w in a sentence s by another one v with an analogous meaning $\mu(v) \overset{p(x)}{=} \mu(w)$, in general, it is supposed the result to be a sentence s' such that $\mu(s) \overset{p(x)}{=} \mu(s')$. Roughly speaking, substituting w in s by a word v having a similar meaning, it is obtained a new sentence having a similar meaning too.

Definition 4.1 *Let* $\mathcal{L} = (\mathrm{Dic}(\mathcal{L}), \mathrm{Sen}(\mathcal{L}), \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu)$ *be an applied language. Two sentences* $s_1 = w_1 w_2 \dots w_n$ *and* $s_2 = v_1 v_2 \dots v_n$ *in* $\mathrm{Sen}(\mathcal{L})$ *are affine, provided that the following statements hold.*

1. *There is a nonempty subset* $I \subseteq \{1, 2 \dots n\}$ *such that the meaning of* s_1 *can be obtained, by means of some procedure* P*, from the meaning-set* $\{\mu(w_i) | i \in I\}$ *of all words with index in* I*.*

2. *The meaning of s_2 can be obtained by means of the same procedure P from the meaning-set $\{\mu(v_i)|i \in I\}$, besides, for every $i = 1, 2 \dots n$, the relation $i \notin I$ implies $w_i = v_i$.*

In the former definition, the words with index in I determine the meaning of each sentence, and this is why we shall term them as essential words in each affinity. Likewise, we shall term each non-essential word of any affinity as a structure one.

Lemma 4.2 *Affinity is an equivalence relation.*

Proof. The former definition remains unaltered interchanging s_1 and s_s; therefore both properties symmetry and reflexivity follow from this fact. To show transitivity, consider that, according to the former definition, two affine sentences are of the same length. In addition, if s_1 and s_2 are affine, and so are s_2 and s_3, then the index I of the essential words in s_1 is the same as the one for s_2, and likewise for s_3; consequently s_1 and s_3 are affine too.

Henceforth, affinity will be denoted by the symbol $\approxeq$, and for every sentence $\mathbf{w}$, the affine-equivalence-class containing it will be denoted as $[\mathbf{w}]_{\approxeq}$.

Definition 4.3 *A word sequence $\mathbf{u} = u_{r_1} u_{r_2} \dots u_{r_k}$ is a sub-concretion for a sentence $\mathbf{w} = w_1 w_2 \dots w_n$ provided that, if $\{w_{r_1}, w_{r_2} \dots w_{r_k}\}$ is the set of essential words in $\mathbf{w}$, then for each integer m such that $0 < m \leq k$ the relation $u_{r_m} \preceq w_{r_m}$ is true.*

Notation. Let $I = \{1, 2 \dots n\}$ be a finite index set and $J = \{r_1, r_2 \dots r_k\}$ a subset of I. Let $\mathbf{w} = w_1 w_2 \dots w_n$ be a sentence and $\mathbf{u} = u_{r_1} u_{r_2} \dots u_{r_k}$ a sub-concretion for $\mathbf{w} = w_1 w_2 \dots w_n$. Henceforth, denote as

$$w_1 w_2 \dots w_n \hookleftarrow u_{r_1} u_{r_2} \dots u_{r_k}$$

the word sequence obtained by substituting each word w_{r_m} in $\mathbf{w}$ by the corresponding u_{r_m} in $\mathbf{u}$.

Remark. When in an expression occur some words, say "triangle" and "pentagon", it is interpreted that each of them is denoting its meaning. However, under some contexts they can be considered, simply, as symbol-sequences disregarding any meaning. Nevertheless, since both symbols $\preceq$ and $\curlyvee$ are defined for objects or concepts, in expressions like "triangle" $\curlyvee$ "pentagon" it must be understood that each word stands for its meaning without using a map μ denoting it. Since there is no confusion, to improve the readability, in this section the same symbols $\mathbf{w}_1$, $\mathbf{w}_2 \dots$ will be used in order to denote words and their meanings when the law $\curlyvee$ occurs among them.

The following example illustrates these ideas.

Example. Consider the English sentences "it depends on height", "it depends on length". Both sentences are affine; and so are $s_1 =$"speak out" and s_2="write out", but "speak out" and "go out" are not, because the meanings of both sentences s_1 and s_2 are built from the meanings of "speak" and "write", respectively, enriching them with the notion of "clearness", while the meaning of "go out" is built by restricting the meaning of the verb "to go" to the only direction from the interior to the exterior. In the affinity "speak out" $\approxeq$ "write out" the only structure word is "out"; and the only essential word in "speak out" is "speak", while the essential one in "write out" is "write". It is worth noticing, the following relations: "speak out" $=$ "write out"$\leftarrow\!\!\prec$"speak" and "write out" $=$ "speak out"$\leftarrow\!\!\prec$"write".

Let I be a countable index-set of cardinality greater than 1, and $\mathbf{A} = \{w_{i,1}w_{i,2}\dots w_{i,n} \mid i \in I\}$ a set of pairwise affine sentences in an applied language

$$\mathcal{L} = (\mathrm{Dic}\,(\mathcal{L})\,, \mathrm{Sen}\,(\mathcal{L})\,, \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu)$$

Let $J = \{r_1, r_2 \dots r_k\}$ the largest subset of $\{1, 2 \dots n\}$ such that for every $m \in J$ the word $w_{i,m}$ is essential in each sentence lying in $\mathbf{A}$; hence for each $m \in \{1, 2 \dots n\}$, the relation $m \notin J$ implies $w_{i,m}$ to be a structure word in each sentence lying in $\mathbf{A}$. These assumptions lead to state the following definition.

Definition 4.4 *A word sequence $\mathbf{p} = u_1u_2\dots u_n$ is the pattern generated by $\mathbf{A}$ provided, that for every $r_m \in J$,*

$$u_{r_m} = \coprod_{i\in I} w_{i,r_m} = w_{1,r_m} \curlyvee w_{2,r_m} \curlyvee \cdots$$

and for every $m \in I \setminus J$ and each $i \in I$, $u_m = w_{i,m}$.

It is worth pointing out, that by virtue of (4), for every j in I and each $r_m \in J$, the following relation holds

$$w_{j,r_m} \preceq u_{r_m} = \coprod_{i\in I} w_{i,r_m} \tag{19}$$

Of course, translation patterns are obtained comparing sentences having the same meaning, but lying in two different languages, for instance see [5]. By contrast, our definition of pattern involves a unique language. However, these kind of patterns are the same as those obtained comparing two languages. Once two pattern sets are obtained from two different languages, say Spanish and English, it is easy to state a correspondence among them assigning each Spanish pattern to the corresponding English one. Such a correspondence can be built because affinities are defined by means of meaning constructors, while the correspondence between two patterns of different languages is implied by the coincidence of their meanings. Thus, the coincidence of meaning-constructors implies the meaning-equality.

Notice, that we have defined the concept of pattern in affinity classes of cardinality greater than 1; therefore for every sentence $\mathbf{w}$ lying in an affinity class $[\mathbf{w}]_{\approxeq}$ of cardinality greater than 1 there is a sub-concretion $\mathbf{u}$ for the pattern $\mathbf{p}$ determined by $[\mathbf{w}]_{\approxeq}$ such that $\mathbf{w} = \mathbf{p} \longleftarrow\!\!\prec \mathbf{u}$.

Definition 4.5 *Let w be a sentence of more than one word, if the affinity class $[w]_{\approxeq}$ is a singleton, then w is called an idiom.*

The following lemma is a straightforward consequence of the previous definition.

Lemma 4.6 *For every applied language $\mathcal{L}$ and every sentence $\mathbf{w} \in \mathrm{Sen}(\mathcal{L})$, one of the following statement holds.*

1. *The sentence $\mathbf{w}$ is an idiom.*
2. *The sentence $\mathbf{w}$ consists only of one word.*
3. *There is a pattern $\mathbf{p}$ and a sub-concretion $\mathbf{u}$ such that $\mathbf{w} = \mathbf{p} \longleftarrow\!\!\prec \mathbf{u}$.*

Proof. The affinity class $[\mathbf{w}]_{\approxeq}$ containing $\mathbf{w}$ either is a singleton of contains more than one element. In the first case, either $\mathbf{w}$ is an idiom or it contains only a word. If $[\mathbf{w}]_{\approxeq}$ is not a singleton, then there is another sentence $\mathbf{v}$ in $[\mathbf{w}]_{\approxeq}$ and the co-product $\mathbf{w} \curlyvee \mathbf{v}$ is a pattern $\mathbf{p}$. If $\mathbf{u} = w_1 w_2 \ldots w_n$ is the essential word sequence in $\mathbf{w}$, then by definition, $\mathbf{w} = \mathbf{p} \longleftarrow\!\!\prec \mathbf{u}$.

Henceforth, to avoid any loss of generality, for every idiom $\mathbf{w}$ we shall extend the definition of the binary law $\longleftarrow\!\!\prec$ writing $\mathbf{w} = \mathbf{w} \longleftarrow\!\!\prec \mathbf{u}$ for every word sequence $\mathbf{u}$. From this view-point, every idiom is an absorbent element for he law $\longleftarrow\!\!\prec$.

It is worth noticing, that one-word sentences can generate patterns induced by attributes and generic concepts. For instance, both words

“triangle@geometry” and “pentagon@geometry”

are concretions of “polygon@geometry”, therefore both equalities

$$\text{“triangle@geometry”} = \text{“polygon@geometry”} \longleftarrow\!\!\prec \text{“triangle@geometry”}$$

and

$$\text{“pentagon@geometry”} = \text{“polygon@geometry”} \longleftarrow\!\!\prec \text{“pentagon@geometry”}$$

hold.

Notation. For every applied language $\mathcal{L}$, denote as $\mathbf{Ptt}(\mathcal{L})$ the collection of all patterns and idioms of $\mathcal{L}$.

Example. Consider the English sentences $\mathbf{s}_1$ = "The area of a triangle" and $\mathbf{s}_2$ = "The area of a pentagon". The pattern generated by these sentences is $\mathbf{p}$ = "The area of a polygon", because "triangle" $\curlyvee$ "pentagon" = "polygon". Since there is no ambiguity, attributes need not be considered; nevertheless the words of both sentences can be enriched adding the attributes "English" and "geometry".

It is worth noticing, that both sentences can be obtained from the pattern "The area of a polygon" by two sub-concretions, as follows.

$$\text{“The area of a triangle”} = \text{“The area of a polygon”} \longleftarrow \text{“triangle”}$$

$$\text{“The area of a pentagon”} = \text{“The area of a polygon”} \longleftarrow \text{“pentagon”}$$

Notation. For every pattern or idiom $\mathbf{w} \in \mathbf{Ptt}(\mathcal{L})$ denote as $\mathrm{subC}(\mathbf{w})$ the family of all sub-concretions for $\mathbf{w}$. Likewise, denote as $\mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L})]$ the family of all sub-concretions for a all members of $\mathbf{Ptt}(\mathcal{L})$.

Definition 4.7 *Let* **PttSC** *denote the category the object-class of which is*

$$\{\mathbf{Ptt}(\mathcal{L}) \times \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L})]\,|\mathcal{L} \in \mathbf{AppLng}\}$$

and morphisms between two objects

$$\mathbf{Ptt}(\mathcal{L}_1) \times \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L}_1)]$$

and

$$\mathbf{Ptt}(\mathcal{L}_2) \times \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L}_2)]$$

are all map-pairs (f, g) such that $g : \mathrm{Dic}(\mathcal{L}_1) \to \mathrm{Dic}(\mathcal{L}_2)$ preserves meanings; in addition, $f : \mathbf{Ptt}(\mathcal{L}_1) \to \mathbf{Ptt}(\mathcal{L}_2)$ satisfies the following condition. For every pattern $\mathbf{w}$, and each sub-concretion $u_1u_2 \ldots u_k \in \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L}_1)]$ both sentences $\mathbf{w} \longleftarrow u_1u_2 \ldots u_k$ and $f(\mathbf{w}) \longleftarrow g(u_1)g(u_2) \ldots g(u_k)$ have the same meaning.

Of course, for every **PttSC**-object the corresponding identity is $\mathrm{id} \times \mathrm{id}$.

4.1 Universal languages

Consider the functor $\mathfrak{T} : \mathbf{PttSC} \to \mathbf{AppLng}$ defined as follows. The image of every object $\mathbf{Ptt}(\mathcal{L}) \times \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L})]$ under the functor $\mathfrak{T}$ is the applied language $\mathcal{L}$. In addition, the functor $\mathfrak{T}$ sends every **PttSC**-morphism

$$(f, g) : \mathbf{Ptt}(\mathcal{L}_1) \times \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L}_1)] \to \mathbf{Ptt}(\mathcal{L}_2) \times \mathrm{subC}\,[\mathbf{Ptt}(\mathcal{L}_2)]$$

into the **AppLng**-arrow $\mathcal{L}_1 \xrightarrow{\mathfrak{T}(f,g)} \mathcal{L}_2$, the underlying map of which

$$\mathfrak{T}(f,g) : \mathrm{Sen}(\mathcal{L}_1) \to \mathrm{Sen}(\mathcal{L}_2)$$

is defined as follows. According to Lemma 4.6, for every $\mathbf{w} \in \mathrm{Sen}(\mathcal{L}_1)$ there are a pattern $\mathbf{v}$ and a sub-concretion $\mathbf{u} = u_1u_2\dots u_k$ for it such that $\mathbf{w} = \mathbf{v} \leftarrowtail \mathbf{u}$, accordingly $\mathfrak{T}(f,g)$ sends $\mathbf{w} = \mathbf{v} \leftarrowtail u_1u_2\dots u_k$ into

$$\mathfrak{T}(f,g)(\mathbf{w}) = f(\mathbf{v}) \leftarrowtail g(u_1)g(u_2)\dots g(u_k)$$

Recall that by virtue of Definition 4.7, both sentences $\mathbf{w}$ and $\mathfrak{T}(f,g)(\mathbf{w})$ have the same meaning.

Definition 4.8 *Let $\mathbf{C}_1$ and $\mathbf{C}_2$ two categories and $\mathfrak{T} : \mathbf{C}_1 \to \mathbf{C}_2$ a functor. A structured arrow $\sigma : X \to \mathfrak{T}(Y)$ is $\uparrow$-universal provided, that for every $\mathbf{C}_2$-morphism $f : X \to \mathfrak{T}(Z)$, there is a unique $\mathbf{C}_1$-morphism $f^* : Z \to Y$ such that the following triangle commutes.*

$$\begin{tikzcd} X \arrow[r, "\sigma"] \arrow[dr, "f"'] & \mathfrak{T}(Y) \\ & \mathfrak{T}(Z) \arrow[u, "\mathfrak{T}(f^*)"'] \end{tikzcd} \tag{20}$$

Definition 4.9 *Let* $\mathbf{L}$ *be a full subcategory of* **AppLng**. *An object*

$$\mathcal{L} = (\mathrm{Dic}(\mathcal{L}), \mathrm{Sen}(\mathcal{L}), \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu)$$

of $\mathbf{L}$ *is a* universal language *in* $\mathbf{L}$ *provided that the inclusion-map*

$$i_{\mathcal{L}_0} : \mathcal{L}_0 \to \mathfrak{T}\left(\mathbf{Ptt}(\mathcal{L}) \times \mathrm{subC}\left[\mathbf{Ptt}(\mathcal{L})\right]\right)$$

is $\uparrow$-universal; where

$$\mathcal{L}_0 = (\mathrm{Dic}(\mathcal{L}), \mathrm{Dic}(\mathcal{L}), \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu|_{\mathrm{Dic}(\mathcal{L})})$$

The concept of universal language is closely related to the ordinary concept of universal element of a structured arrow. In the following theorem we show the main properties.

Theorem 4.10 *If $\mathcal{L}$ is a universal language in a family $\mathbf{L}$, then the following statements are true.*

1. *Every universal language in $\mathbf{L}$ is isomorphic to $\mathcal{L}$, that is to say, universal languages are unique up to isomorphisms.*

2. *Denoting as "syntax" the ordering in which essential words occur in a pattern, if the family* $\mathbf{L}$ *contains sufficient languages in order to contain every possible syntax, then* $\mathcal{L}$ *must be a syntax-free language.*

3. $\mathcal{L}$ *satisfies every property of any natural or artificial language in* $\mathbf{L}$ *that can be preserved under* $\mathbf{PttSC}$*-morphisms.*

Proof.

1. Suppose that there are two $\uparrow$-universal arrows

$$i_{\mathcal{L}_0} : \mathcal{L}_0 \to \mathfrak{T}\left(\mathbf{Ptt}(\mathcal{L}) \times \mathrm{subC}\left[\mathbf{Ptt}(\mathcal{L})\right]\right)$$

and

$$j_{\mathcal{L}_0} : \mathcal{L}_0 \to \mathfrak{T}\left(\mathbf{Ptt}(\mathcal{L}') \times \mathrm{subC}\left[\mathbf{Ptt}(\mathcal{L}')\right]\right)$$

By definition, we have that there are also two morphisms (f_1, g_1) and (f_2, g_2) such that

$$i_{\mathcal{L}_0} = \mathfrak{T}(f_1, g_1) \circ j_{\mathcal{L}_0} \tag{21}$$

$$j_{\mathcal{L}_0} = \mathfrak{T}(f_2, g_2) \circ i_{\mathcal{L}_0} \tag{22}$$

hence, $i_{\mathcal{L}_0} = \mathfrak{T}\left((f_1, g_1) \circ (f_2, g_2) \circ\right) i_{\mathcal{L}_0}$, and by virtue of the assumed uniqueness, $(f_1, g_1) \circ (f_2, g_2) = \mathrm{id} \times \mathrm{id}$, and statement 1) follows.

2. Let $\mathbf{w} = w_1 w_2 \ldots w_n$ be a pattern in any language $\mathcal{L}_1$ lying in $\mathbf{L}$ and $w_{r_1} w_{r_2} \ldots w_{r_k}$ the sequence of essential words in $\mathbf{w}$. Let $f : \mathcal{L}_0 \to \mathcal{L}_1$ be a map sending each word w in $\mathcal{L}_0$ into a word $f(w) \in \mathrm{Dic}(\mathcal{L}_1)$ having the same meaning; where $\mathcal{L}_0 = \big(\mathrm{Dic}(\mathcal{L}), \mathrm{Dic}(\mathcal{L}), \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu|_{\mathrm{Dic}(\mathcal{L})}\big)$. Since $\mathcal{L}$ is assumed to be universal, then there is a unique morphism

$$(f^*, g^*) : \left(\mathbf{Ptt}(\mathcal{L}_1) \times \mathrm{subC}\left[\mathbf{Ptt}(\mathcal{L}_1)\right]\right) \to \left(\mathbf{Ptt}(\mathcal{L}) \times \mathrm{subC}\left[\mathbf{Ptt}(\mathcal{L})\right]\right)$$

such that the following triangle commutes.

$$\begin{tikzcd} \mathcal{L}_0 \arrow[r, "i_{\mathcal{L}_0}"] \arrow[dr, "f"'] & \mathfrak{T}\left(\mathbf{Ptt}(\mathcal{L}) \times \mathrm{SubC}[\mathbf{Ptt}(\mathcal{L})]\right) \\ & \mathfrak{T}\left(\mathbf{Ptt}(\mathcal{L}_1) \times \mathrm{SubC}[\mathbf{Ptt}(\mathcal{L}_1)]\right) \arrow[u, "\mathfrak{T}(f^*, g^*)"'] \end{tikzcd} \tag{23}$$

Thus, if $\mathbf{w}$ is a pattern in a language $\mathcal{L}_1$ the essential word-sequence of which is $w_{r_1} w_{r_2} \ldots w_{r_k}$, then there is also a pattern $f(\mathbf{w})$ in $\mathcal{L}$ such that $g^*(w_{r_1}) g^*(w_{r_2}) \ldots g^*(w_{r_k})$ is the corresponding essential word sequence. Now, by hypothesis, there are all possible syntax in the languages of $\mathbf{L}$; accordingly, if S_k is the symmetric group of degree k, then for every

$\sigma \in S_k$ there is at least one language $\mathcal{L}_\sigma$ such that the pattern $\mathbf{w}_\sigma$, being equivalent to $\mathbf{w}$, that is to say, determining the same meaning as both $\mathbf{w}$ and $f(\mathbf{w})$, the corresponding essential-word sequence of which is $\sigma(g^*(w_{r_1}))\sigma(g^*(w_{r_2}))\ldots\sigma(g^*(w_{r_k}))$; consequently it does not matter the order in which each word occurs in $f^*(\mathbf{w}) \longleftarrow g^*(w_{r_1})g^*(w_{r_2})\ldots g^*(w_{r_k})$, hence $\mathcal{L}$ must be a syntax-free language.

3. With the same assumptions as in the preceding statement, if P is a property of a pattern $\mathbf{w}$ and the corresponding essential word sequence $w_{r_1}w_{r_2}\ldots w_{r_k}$ of any language $\mathcal{L}_1 \in \mathbf{L}$ being preserved under (f^*, g^*), then the pattern $f^*(\mathbf{w})$ together with $g^*(w_{r_1})g^*(w_{r_2})\ldots g^*(w_{r_k})$ satisfy also such a property P. Thus, since $\mathcal{L}_1$ is an arbitrary language in $\mathbf{L}$, then the universal language $\mathcal{L}$ satisfies every property P of any language in $\mathbf{L}$ being preserved under **PttSC**-morphisms.

4.1.1 Structure of universal languages

As we have just seen above, the most noticeable property of a universal language is the lack of any syntax. Thus, if a language $\mathcal{L}$ is universal in a family $\mathbf{L}$, the essential words $w_{r_1}w_{r_2}\ldots w_{r_k}$ in a sentence $\mathbf{w} = w_1w_2\ldots w_n$ in $\mathcal{L}$ can occur in any ordering; hence $w_{r_1}w_{r_2}\ldots w_{r_k}$ can be regarded as a word-set instead of a word-sequence. Thus, each sentence $\mathbf{w}$ in $\mathcal{L}$ can consist of a set of words $\mathbf{s} = \{w_1, w_2 \ldots w_k\}$ together with a subsequence $\mathbf{u}$ denoting the procedure by means of which the meaning of $\mathbf{w}$ must be built from the meanings of the members of $\mathbf{s}$. Thus, the sub-sentence $\mathbf{u}$ determines also a pattern $\mathbf{u}^* = u_1u_2\ldots u_n$, which can be regarded as a map $\mathbf{u}^\natural(u_1, u_2 \ldots u_n)$ sending the n-tuple $(u_1, u_2 \ldots u_n)$ into the meaning of $\mathbf{w}$; accordingly,

$$\mathbf{w} = \mathbf{u}\,\mathbf{s} = \mathbf{u}^* \longleftarrow \mathbf{s} = \mathbf{u}^\natural(w_1, w_2 \ldots w_n)$$

After these ideas, it is clear, that sentences in universal languages are couples consisting of a word set $\mathbf{s}$ together with a sentence $\mathbf{u}$ denoting only a procedure by means of which the meaning of $\mathbf{u}\,\mathbf{s}$ must be built from the meanings of the words of $\mathbf{s}$. Thus, a universal language $\mathcal{L}$ must contain a sub-language $\mathrm{Aff}(\mathcal{L})$ each sentence of which denotes a meaning-constructor; therefore any algorithm to determine affinities can be built easily in universal languages. For instance, in $\mathbf{w} = \mathbf{u}\,\mathbf{s}$, the sentence $\mathbf{u}$ belongs to the language $\mathrm{Aff}(\mathcal{L})$, therefore two sentences $\mathbf{w}_1 = \mathbf{u}_1\,\mathbf{s}_1$ and $\mathbf{w}_2 = \mathbf{u}_2\,\mathbf{s}_2$ are affine, whenever $\mathbf{u}_1 = \mathbf{u}_2$ and affinities can be obtained easily in universal languages.

In general, it is possible to assign a meaning-constructor set to $\mathbf{s}$ of cardinality greater than $n!$, being n the size of $\mathbf{s}$; accordingly universal languages are more powerful than non-universal ones lying in the same class $\mathbf{L}$. The construction of an efficient sub-language $\mathrm{Aff}(\mathcal{L})$ together with the investigation of its properties are open problems.

Perhaps the property of being syntax-free, is the most powerful feature in order to use any universal language as the intermediate device between two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ lying in the same class $\mathbf{L}$. To see this fact, consider that to translate a sentence $\mathbf{w}$ from $\mathcal{L}_1$ to $\mathcal{L}_2$, in any translation algorithm the syntactic rules of both $\mathcal{L}_1$ and $\mathcal{L}_2$ must be considered. By contrast, to translate a sentence $\mathbf{w}$ from $\mathcal{L}_1$ to $\mathbf{w}' \in \mathcal{L}$ the only syntax to be considered is the underlying one of $\mathcal{L}_1$; and to translate $\mathbf{w}'$ from $\mathcal{L}$ to the corresponding sentence $\mathbf{w}''$ of $\mathcal{L}_2$ an algorithm need only deal with the syntax of $\mathcal{L}_2$.

In addition, since universal languages are artificial, then they can be built in a coherent way; therefore, in order to obtain abstractions and concretions, not only affinities, but algorithms for this aim can be also performed in an efficient way. Finally, taking into account, statement 3) in the former theorem, universal languages satisfy every property of any language in $\mathbf{L}$ that are preserved under morphisms, hence each of these properties remains unaltered under translations from any language to a universal one. By virtue of this features, universal languages are the more adequate intermediate for natural and artificial language translations. If nowadays it is an unthinkable task to built mathematical algorithms using the roman numeral system in order to denote real numbers, it is a very plausible hypothesis, that in the future the use of at least one universal language equipped with a coherent dictionary will be a necessary device in computer science and artificial intelligence research.

Since universal languages are unique up to isomorphisms, they can be investigated and developed by different researcher groups and the same language structure will be obtained. This feature cannot be found in any natural language, and this is why the adjective "universal" is adequate beyond the scope of algebra.

5 Coherent language construction

The construction of coherent languages requires to handle the meaning of each word in order to determine the relation $\preceq$ between the denoted concepts. However, it is possible to build algorithms working with sentences of a language by means of which the relation $\preceq$ can be determined knowing no meaning. To see this fact, consider any applied language

$$\mathcal{L} = \left(\mathrm{Dic}\left(\mathcal{L}\right), \mathrm{Sen}\left(\mathcal{L}\right), \preceq, \Omega_{\mathfrak{P}}, \sqsubseteq, \mu\right)$$

and a sentence $\mathbf{w} = w_1 w_2 \dots w_j \dots w_n \in \mathrm{Sen}\left(\mathcal{L}\right)$. Suppose that there is a set of words $\mathbf{W} = \{u_1 u_2 \dots u_k\}$ containing w_j such that substituting w_j in $\mathbf{w}$ by any member of $\mathbf{W}$ the obtained word sequence has a meaning, that is, it is again a sentence, but any other substitution gives rise to a nonsense result. With these assumptions, if there is a predicate $p(x)$ defining the set $\mathbf{W}$, that is to say, if

$\mathbf{W} = \{x \mid p(x)\}$, then a sentence $w_1 w_2 \ldots w_j \ldots w_n \in \text{Sen}(\mathcal{L})$ is meaningful, provided that the word occurring in the j-th place belongs to $\mathbf{W}$, accordingly its meaning satisfies the predicate $p(x)$. Thus, if in some text occurs a sentence $w_1 w_2 \ldots w_j \ldots w_n \in \text{Sen}(\mathcal{L})$ this fact allows us to know that $w_j \in \mathbf{W}$, besides, $p(\mu(w_j))$. Accordingly, taking into account, that

$$\mathbf{W}^{\curlyvee} = \mu(u_1) \curlyvee \mu(u_2) \curlyvee \cdots \curlyvee \mu(u_k)$$

is the concept defined by the predicate $p(x)$, then for every word $w \in \mathbf{W}$ the denoted object $\mu(w)$ satisfies the relation $\mu(w) \preceq \mathbf{W}^{\curlyvee}$. Accordingly, it is possible to build algorithms in order to determine the relation $\preceq$ among the represented objects dealing only with words and sentences. To this end it is only necessary to observe word occurrences in sets of sentences.

Example. Consider the sentences $\mathbf{s}_1 =$ "teachers know how to write texts" and $\mathbf{s}_2 =$ "doctors know how to write texts". Indeed, each word occurring in the first position in these sentences denotes a person being able to write texts. Denoting as "x" any human being that knows how to write any text, then the following relations are true: $\mu(\text{“teachers”}) \preceq x$, $\mu(\text{“doctors”}) \preceq x$, because both concepts denoted by "teacher" and "doctor" are particular cases of persons knowing how to write texts.

Of course, the more restrictive a sentence is, the more concrete is the meaning of any involved word. Thus, from a sentence collection, selected from any language $\mathcal{L}$, some classification algorithms can be built in order to arrange his words in an increasing abstraction level without knowing their meanings, and by means of such a classification new coherent languages can be built. The construction of such a kind of algorithms is an open problem.

6 Optimal principle

Consider a set E of algebraic expressions in the ordinary formal language. A natural equivalence $\mathcal{R} \subseteq E \times E$ can be stated for those expressions having the same value; for example, both expressions $e_1 = x^2x^3y$ and $e_2 = x^5y$ have the same value, therefore $(e_1, e_2) \in \mathcal{R}$. Now, if an expression $e \in E$ contains n symbols, consider the mapping ν sending e into $\frac{1}{n}$. Thus, $\nu : E \to (0, 1]$ is the function sending each expression into the inverse of the number of symbols it contains; for example, $\nu(x^2x^3y) = \frac{1}{5}$ and $\nu(x^5y) = \frac{1}{3}$. The mapping ν defines a measure of the simplicity of any expression in E. By means of the map ν two binary relations can be defined. The first one $\leq_{\mathcal{S}}$ compares the simplicity of two expressions; therefore $e_1 \leq_{\mathcal{S}} e_2$ if and only if $\nu(e_1) \leq \nu(e_2)$. The second relation $\mathcal{S} \subseteq E \times E$ is the equivalence defined as follows: $(e_1, e_2) \in \mathcal{S}$ if and only if $\nu(e_1) = \nu(e_2)$. It is easy to see, that both relations $e_1 \leq_{\mathcal{S}} e_2$ and $e_2 \leq_{\mathcal{S}} e_1$ imply $(e_1, e_2) \in \mathcal{S}$, for instance, let $e_1 = x^2y$ and $e_2 = yx^2$ both expressions are $\mathcal{S}$-equivalent but they are different words. Thus, from the view-point of the ordinary formal language, both relations $e_1 \leq_{\mathcal{S}} e_2$ and $e_2 \leq_{\mathcal{S}} e_1$ need not imply $e_1 = e_2$, but they imply $(e_1, e_2) \in \mathcal{S}$. This circumstance occurs because of being $\leq_{\mathcal{S}}$ reflexive and transitive, it need not be an antisymmetric relation; consequently, the result of any simplification need not be unique. Thus, if e_1 cannot be simplified the relation $e_1 \leq_{\mathcal{S}} e_2$ implies $(e_1, e_2) \in \mathcal{S}$, however, in general, the equality $e_1 = e_2$ does not hold; therefore e_1 is not a $\leq_{\mathcal{S}}$-maximal element, but it can be regarded as an "optimal" simplification in the $\mathcal{R}$-equivalence class containing it. After these considerations, we introduce the concept of "optimal" element, with respect to any transitive and reflexive relation, as a generalization of maximal concept; hence, with this convention, "maximal" $\preceq$ "optimal".

Let E be a set, $\mathcal{R} \subseteq E \times E$ and $\mathcal{S} \subseteq E \times E$ two equivalence relations and $\leq_{\mathcal{S}} \subseteq E \times E$ a reflexive and transitive one satisfying the following axioms.

Axiom 6.1 *For every couple x and y of members of E, either $x \leq_{\mathcal{S}} y$ or $y \leq_{\mathcal{S}} x$ if and only if $(x, y) \in \mathcal{R}$.*

Axiom 6.2 *For every couple x and y in E, if both relations $x \leq_{\mathcal{S}} y$ and $y \leq_{\mathcal{S}} x$ hold, then $(x, y) \in \mathcal{S}$.*

Definition 6.3 *Let E be a set, $\mathcal{R} \subseteq E \times E$, $\mathcal{S} \subseteq E \times E$ two equivalence relations, and $\leq_{\mathcal{S}} \subseteq E \times E$ a transitive and reflexive one satisfying both Axiom 6.1 and Axiom 6.2. A member x of E is $\leq_{\mathcal{S}}$-optimal provided that, for every $y \in E$, the relation $x \leq_{\mathcal{S}} y$, implies that $y \leq_{\mathcal{S}} x$.*

It is worth pointing out, that by virtue of Axiom 6.2, if x is $\leq_{\mathcal{S}}$-optimal, then the relation $x \leq_{\mathcal{S}} y$, implies that $(x, y) \in \mathcal{S}$.

Definition 6.4 *Let X and Y be two sets, $\mathcal{R}$ and $\mathcal{S}$ two equivalence relations, and $\leq_{\mathcal{S}}$ a transitive and reflexive one, defined in $X \cup Y$, and satisfying both Axiom 6.1 and Axiom 6.2. The set Y is a $\leq_{\mathcal{S}}$-optimization of X, provided that for each $(x, y) \in X \times Y$, the relation $(x, y) \in \mathcal{R}$ implies $x \leq_{\mathcal{S}} y$.*

We shall denote this property writing $X \leq^*_{\mathcal{S}} Y$. Likewise, in any category of structured sets, we shall say an object X to be $\leq^*_{\mathcal{S}}$-optimal provided, that for every object Y, the relation $X \leq^*_{\mathcal{S}} Y$ implies $Y \leq^*_{\mathcal{S}} X$.

Definition 6.5 *Let* $\mathbf{C}$ *be a concrete category over* **Set**, *and $\mathcal{R}$, $\mathcal{S}$, and $\leq_{\mathcal{S}}$ three binary relations satisfying both Axiom 6.1 and Axiom 6.2, being defined in the union of all objects of* $\mathbf{C}$. *A subcategory* $\mathbf{K}$ *of* $\mathbf{C}$ *is directed by $\leq_{\mathcal{S}}$ provided that each* hom-*set* $\hom_{\mathbf{K}}(X, Y)$ *is nonempty if and only if Y is a $\leq_{\mathcal{S}}$-optimization of X.*

Theorem 6.6 (Optimal principle) *Let* $\mathbf{C}$ *be a concrete category over* $\mathbf{Set}$*, and* $\mathcal{R}$*,* $\mathcal{S}$*, and* $\leq_{\mathcal{S}}$ *three binary relations defined in*

$$\bigcup_{O \in Obj(\mathbf{C})} O$$

and satisfying both Axiom 6.1 and Axiom 6.2. Let $\mathbf{C}_1$ *be a subcategory of* $\mathbf{C}$ *directed by* $\leq_{\mathcal{S}}$*, and* $\mathfrak{T} : \mathbf{C}_1 \to \mathbf{C}_2$ *a functor. With these assumptions, for every* $\uparrow$*-universal arrow* $O \xrightarrow{\sigma} \mathfrak{T}(X)$ *the object* X *is* $\leq^*_{\mathcal{S}}$*-optimal.*

Proof. Suppose that there is a $\mathbf{C}_1$-object Y such that

$$X \leq^*_{\mathcal{S}} Y \tag{24}$$

Now, by Definition 6.5 the set $\hom_{\mathbf{C}_1}(X, Y)$ is nonempty; so then there is at least one morphism $g : X \to Y$, and the composition $f = \mathfrak{T}(g) \circ \sigma$ belongs to $\hom_{\mathbf{C}_2}(O, \mathfrak{T}(Y))$. Since σ is assumed to be $\uparrow$-universal, the existence of f implies that there is a unique $f^* : Y \to X$ such that $\sigma = \mathfrak{T}(f^*) \circ f$, and because $\mathbf{C}_1$ is assumed to be $\leq_{\mathcal{S}}$-directed, then $Y \leq^*_{\mathcal{S}} X$. Finally, taking into account (24), the object X is $\leq^*_{\mathcal{S}}$-optimal.

In addition to the universal language properties described in Section 4.1.1, the former theorem enriches the universal language concept with those improvements denoted by any transitive and reflexive relation $\leq_{\mathcal{S}}$ directing the category to which it belongs. In particular, $\mathcal{R}$ can be the meaning equivalence, while both $\leq_{\mathcal{S}}$ and $\mathcal{S}$ can denote suitable improvements for each purpose. Accordingly, for every family of languages $\mathbf{L}$ and each purpose, Theorem 6.6 allows us to determine the properties with respect to which a universal language is optimal. Thus, the frequently asked questions: "what is the best programming language for some purpose" or "what is the best language to build efficient algorithms for some object-structure", in general, can be answered with the help of the preceding theorem, whenever the considered language category is directed by a transitive and reflexive relation adequate for the desired aim. Indeed, optimality can be proved with the apodicticity of algebra. Nevertheless, the research of universal languages can provide us a wide class of open problems in order to improve algorithm efficiency together with a deeper knowledge of the algebraic structure of natural intelligence, because human thought always runs through some language.

7 Open Problems

For a small category $\mathfrak{L}$ of natural and artificial languages, the object-class of which is of cardinal n, to perform translations among them it is required a family of $\binom{n}{2}$ translating algorithms. However, choosing a language $\mathcal{L} \in \mathfrak{L}$

working as intermediate, n of them are sufficient. Now, the question consists of determining those properties that $\mathcal{L}$ must satisfy, in order to be the better choice. Both Theorem 4.10 and Theorem 6.6 lead the choice towards any universal language in $\mathfrak{L}$. In any case, to obtain efficient algorithms, algebraic coherence must be required too.

We also consider auxiliary analogical representations, since in order to denote concrete concepts, coherent analogical languages are more adequate than abstract ones. For example, to denote the triangle concept in a coherent way the symbol $\triangle$ works better than any word. It is worth pointing out, that we term as analogical languages those representation systems consisting of bidimensional pictures preserving some structure properties of the denoted concepts. Musical notation and chemical formulas can be regarded as instances of our analogical language concept. Likewise, a didactic video can be regarded as an analogical explanation of some subject; therefore the involved images form an analogical language.

By contrast, by virtue of the prefix-suffix machinery, together with the possibility of performing arbitrary associations, to denote abstract concepts words and sentences are more efficient. Recall that the relation $\sqsubseteq$ is defined for words and their tails, and the tail of a word can be regarded as a suffix.

After these considerations, we analyze the translation problem among a category of languages $\mathfrak{L}$ using an intermediate one $\mathcal{L}_0$. As we have shown above, the better choice can be an algebraically coherent language $\mathcal{L}_0$ being universal in $\mathfrak{L}$. Thus, the construction of $\mathcal{L}_0$ would be the first step towards a universal translation project, hence the main problem consists of determining its properties. On the one hand, the words and alphabet of $\mathcal{L}_0$ can be decided via conventions among researchers; whenever word-structures are defined in a coherent way. On the other hand, by Theorem 4.10 we know that $\mathcal{L}_0$ must satisfy the following properties.

1. Every universal language is unique up to isomorphisms, besides, it is syntax-free.

2. Every language $\mathcal{L}_0$ being universal in a category $\mathfrak{L}$ satisfies each property of every language in $\mathfrak{L}$ which is preserved under morphisms.

The last property arises from Theorem 4.10, but intuitively was adopted by L. L. Zamenhof in his invention, the international auxiliary language Esperanto. The Zamenhof's invention is an easy-to-learn and politically neutral language containing the most noticeable features of several natural languages. Nevertheless, Theorem 4.10 shows the convenience of this requirement from an algebraic view-point. These considerations lead to the following problems.

7.1 Universal Language Problems

As we have just seen, a universal language must be syntax-free and coherent. On the one hand, disambiguation of a syntax-free language can be performed by means of a declension-system, this being stronger than the underlying one in Latin. For instance, consider the phrase "interesting books and old pictures". The word order matters. However with some suffixes linking nouns and adjectives, this inconvenient vanishes. To this end, consider the suffixes "&a", "&b". Writing "interesting&a books&a and old&b pictures&b" the word order does not matter, since the same suffix joins every adjective with the word to which it is applied.

On the other hand, coherence requires to handle word-analogies. Recall, that as it is shown in Section 5, word-analogies can be found by observing word occurrences in sentences of texts written in some language $\mathcal{L}$. These facts suggest the following problems.

Problem 7.1 *For a given natural language, find an algorithm in order to extract all patterns lying in a sufficient large text.*

Problem 7.2 *For a given natural language L, find an algorithm A such that processing texts written in L the algorithm A can build a coherent language L^* together with the corresponding dictionary or hash-table $L \rightleftarrows L^*$.*

Both problems above are very helpful in translating algorithm construction; but the main problem we are dealing with is the following one.

Problem 7.3 *Does there exist any translating algorithm between a universal language $\mathcal{L}_0$ and any natural one $\mathcal{L}$?*

This problem requires some explanation. On the one hand, a universal language is unique, therefore in order to be determined no complementary conditions are required. On the other hand, translations must be syntactically correct and coherent. Nowadays, there are several translating algorithms between couples of natural languages which build incoherent sentences. When we "say translating algorithm", it must be understood any procedure as efficient as a human. In general, a human is able to discern inflections and contexts. Natural languages are not context-free ones. The former problem requires the existence of algorithms determining emphasis and context from the observed phrases. Recall that disambiguations are performed by means of attributes, therefore attributes in any phrase must be determined too.

Notice, that a human can deduce the meaning of ambiguous sentences, even incorrect ones. For example, a human can interpret the phrase "a tree-angled polygon" supposing to be a misspelled instance of the phrase "a three-angled polygon", simply, looking for the smallest modification having a meaning. Frequently, actual translating algorithms build incorrect sentences. For instance, in an algebraic context, a popular English-to-Spanish translating algorithm returns "el juego tiene un miembro" introducing a sentence containing the word sequence "the set contains one member". In fact, the Spanish phrase "El juego tiene un miembro" means " the game has a member". The correct translation is "el conjunto posee un elemento". This is why the following problem also matters.

Problem 7.4 *For a given a natural language $\mathcal{L}_1$, does there exist an algorithm discerning whether or not a sentence is correct and coherent?*

It is an author's conjecture, with respect to the former problem, that working with auxiliary-analogical languages the existence of the required algorithm can be proved. By contrast, such an algorithm cannot exist for natural languages which consist of arbitrary conventions, without the help of analogical representations. Indeed, to know whether or not a sentence is correct it is sufficient to be found out in a reliable text. However, such a method is an endless one, because combining words one can build infinite sentence-sets. This is why the required algorithm must work with a family of rules. In general, to apply a rule it is required to know the meaning of the involved sentences.

For instance, consider an adjective u which cannot be applied to the concept O denoted by a sentence $s = w_1 w_2 \dots w_n$, unless O satisfies some property P. Thus, to know whether or not a sentence $u\ s = u\ w_1 w_2 \dots w_n$ containing s is correct, it is required to know whether O satisfies P. Consequently, the algorithm $\mathcal{A}$ must be able to determine some property family of O containing P. Since analogical languages we have defined above preserve some properties of the denoted objects, then, involving an analogical language $\mathfrak{A}$, such an algorithm could be built. To this end, $\mathcal{A}$ must assign a member of $\mathfrak{A}$ to s satisfying P.

Perhaps, this problem can be solved using multi-threading languages, that is to say, a kind of complex languages combining both analogical and abstract ones in a similar way to musical notation. This can be the aim of further researches. In fact, human beings work with such a kind of languages. For instance, in a didactic video equipped with a narrating voice, the visual information is analogical, while the narration is performed in some natural language. Likewise, a figure sequence with captions can be regarded as a two-threaded language. In any case, it is the author's opinion that restricting our notations to a partial free-monoid, the problem cannot be solved.

References

[1] AJ. Adámek, H. Herrlich and G. E. Strecker , *Abstract and Concrete Categories. The joy of Cats*, Willey-Intersciencie Publication, (1990).

[2] Al-Maitah, M., "About One way of Encoding Alphanumeric and Symbolic Information", *Int. J. Open Probl. Comput. Sci. Math.*, (2010).

[3] Borceux, F., *Handbook of Categorical Algebra, Basic Category Theory*, Encyclopedia of Mathematics and its Applications, **50**, (1994).

[4] Jurafsky, D. and Martin, J. H., Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, *Upper Saddle River, NJ: Prentice Hall (Prentice Hall series in artificial intelligence*, (2000).

[5] McTait, K. and Trujillo, A., "A Language-Neutral Sparse-Data Algorithm for Extracting Translation Patterns", *Proceedings of 8th International Conference on Theoretical and Methodological Issues in Machine Translation*, Chester, UK, (1999).

[6] Manning, C. D. and Schütze, H., *Foundations of Statistical Natural Language Processing*,Cambridge, MA: The MIT Press, (1999).

[7] Ouyang, C. and Dumas, M. and ter Hofstede A. H.M., and van der Aalst, W. M.P., "Pattern-based Translation of BPMN Process Models to BPEL Web Services", *OUTePrints*, (2007).

[8] Palomar Tarancón, J. E., "Algebra of analogies and abstractions. Applications in natural intelligence research. (Beyond Turing machines)", submitted to *Surv. Math. Appl.*, (2010).

[9] Premalatha, K. and Natarajan, A. M., "Hybrid PSO and GA for Global Maximization", *Int. J. Open Probl. Comput. Sci. Math.* (2009).