Annals of University of Craiova, Math. Comp. Sci. Ser.
Volume 36(2), 2009, Pages 97–108
ISSN: 1223-6934

From Text Description to Semantic Schema

Cristina Zamfir

Abstract. The concept of semantic schema was introduced in [5]. This structure is an abstract concept S allowing to model an inference process in knowledge representation. In order to represent some knowledge piece KP, an appropriate interpretation I of S for KP is used. The abstract inference from S is interpreted by I and the conclusion in KP is obtained. The problem of finding a semantic schema S for a knowledge piece KP is not yet an automated process. In this paper we inaugurate a possible research line for this process. Starting from a text description of KP given in a natural language, after various processes described in this paper, we obtain a semantic schema that can model KP. Several examples are presented to exemplify these processes. Finally we describe the future work.

2000 Mathematics Subject Classification. Primary 68T30, Secondary 68T35, 68T50.
Key words and phrases. Recursive Transition Network, Semantic Schema.

1. Introduction

There are three methods based on graph theory for knowledge representation: conceptual graphs ([2], [3]), labeled stratified graphs ([7]), semantic schemas ([5]). The papers [4] and [6] treat the following problem: reconstitute an image from a text description. The formalism used in these papers is based on labeled stratified graphs. The text description is given in natural language and the papers [8] and [9] are strongly connected by this subject. In this paper we treat a similar problem, but the mechanism is offered by semantic schemas instead of labeled stratified graphs.

In the philosophy of language, a natural language (also known as ordinary language) is a language that is spoken, signed, or written by humans for general-purpose communication. The natural language is distinguished from formal languages such as computer-programming languages (C++, Java, ...) or the "languages" used in the study of formal logic, especially mathematical logic and from constructed languages.

The theory of universal grammar proposes that all natural languages have certain underlying rules which constrain the structure of the specific grammar for any given language.

There are two types of languages: artificial languages and natural languages. The artificial language is defined by creating the correct set of rules that applied to the defined vocabulary will form the phrases of the language.

A phrase in a natural language is a finite sequence of words over some alphabet. The combination of words must respect certain rules that form the language syntax and it must be coherent thus respecting the rules of the language semantics.

Let's consider the following phrase:
Peter is a doctor. He works in a hospital. Maria is a teacher and she works in a school. Maria is the wife of Peter. Ann is their daughter.

A human can perfectly understand the phrase and its implications and therefore can realize the connections and extract the fact that:

- `The wife of Peter works in a school`
- `Peter is the father of Ann`
- `Maria is the mother of Ann`
- `The father of Ann works in a hospital`

In order for a program to obtain the results listed above the following stages must be realized:

(1) the transformation of the phrase P into a semantic schema; in other words, we extract from P several entities which allow to build a semantic schema S;
(2) the construction of an interpretation I for S;
(3) the analysis of the schema S; as a result of this stage we conclude that by means of I we can retrieve just the semantics given by P;
(4) the construction of the formal computations in S;
(5) the interpretations of the final entities obtained by the formal computations and the certification of I as a good interpretation for P.

The focus of this paper is the first phase: the transformation of a phrase into a semantic schema.

2. Semantic schemas

The concept of θ-semantic schema (or semantic schema) was introduced in [5] and was developed in [10], [11], [12] and [13]. Various cooperating systems based on semantic schemas were developed in [14], [15] and [16]. Several extensions of semantic schemas were presented in [17], [18] and [19].

The term "semantic schema" is very important for this paper and will be encountered many times in the pages to come. It is important to recall the definition of a semantic schema:

Definition 2.1. *A θ-**schema** is a system $\mathcal{S} = (X, A_0, A, R)$, where*

- *X is a finite non-empty set of symbols;*
- *A_0 is a finite non-empty set of elements named label symbols and $A_0 \subseteq A \subseteq \overline{A}_0$, where $\overline{A}_0$ is the Peano θ-algebra generated by A_0;*
- *$R \subseteq X \times A \times X$ is a non-empty set which fulfills the following conditions:*
  (1) $(x, \theta(u, v), y) \in R \Longrightarrow \exists z \in X : (x, u, z) \in R, (z, v, y) \in R$
  (2) $\theta(u, v) \in A, (x, u, z) \in R, (z, v, y) \in R \Longrightarrow (x, \theta(u, v), y) \in R$
  (3) $u \in A \Longleftrightarrow \exists (x, u, y) \in R$

We recall that θ is an operator symbol of arity 2 and the Peano θ-algebra generated by A_0 is the set

$$\overline{A}_0 = \bigcup_{n \geq 0} A_n$$

where A_n are defined recursively as follows:

$$A_{n+1} = A_n \cup \{ \ \theta(u, v) \ \ | \ \ u, v \in A_n\}, \quad n \geq 0$$

Let $\mathcal{S} = (X, A_0, A, R)$ be a semantic schema. We consider a symbol h of arity 1 and take the set:

$$M = \{h(x, a, y) \quad | \quad (x, a, y) \in R \cap (X \times A_0 \times X)\}$$

where we use the notation $h(x, a, y)$ instead of $h((x, a, y))$. We consider a symbol σ of arity 2 and denote by $\mathcal{H}$ *the Peano σ-algebra generated by M.*

We denote by Z the alphabet including the symbol σ, the elements of X, the elements of A, the left and right parentheses, the symbol h and comma. We denote by Z^* the set of all words over Z. We define the following binary relation on Z^*:

Definition 2.2. *([5]) Let be $w_1, w_2 \in Z^*$.*

- *If $a \in A_0$ and $(x, a, y) \in R$ then*
  $$w_1(x, a, y)w_2 \Rightarrow w_1 h(x, a, y)w_2$$
- *Let be $(x, \theta(u, v), y) \in R$. If $(x, u, z) \in R$ and $(z, v, y) \in R$ then*
  $w_1(x, \theta(u, v), y)w_2 \Rightarrow w_1\sigma((x, u, z), (z, v, y))w_2$

We denote by $\Rightarrow^*$ the reflexive and transitive closure of the relation $\Rightarrow$.

We can define now the mapping generated by a semantic schema:

Definition 2.3. *The* **mapping generated** *by $\mathcal{S}$ is the mapping*

$$\mathcal{G}_{\mathcal{S}} : R \longrightarrow 2^{\mathcal{H}}$$

defined as follows:

- $\mathcal{G}_{\mathcal{S}}(x, a, y) = \{h(x, a, y)\}$ *for* $a \in A_0$
- $\mathcal{G}_{\mathcal{S}}(x, \theta(u, v), y) = \{w \in \mathcal{H} \mid (x, \theta(u, v), y) \Rightarrow^* w\}$

The set $\mathcal{H}$ is an infinite one. We extract from $\mathcal{H}$ those elements which can be derived from R and we denote this set by $\mathcal{F}_{comp}(S)$. In other words,

$$\mathcal{F}_{comp}(S) = \{w \in \mathcal{H} \mid \exists (x, u, y) \in R : (x, u, y) \Rightarrow^* w\}$$

Definition 2.4. *An* **interpretation** *$\mathcal{I}$ of $\mathcal{S}$ is a system $\mathcal{I} = (Ob, ob\{Alg_u\}_{u \in A})$ where*

- *Ob is a finite set of elements which are called the* **objects** *of the interpretation*
- *$ob : X \rightarrow Ob$ is a bijective function.*
- *For each $u \in A$ the algorithm Alg_u has two input arguments.*

Let be $R_0 = (X \times A_0 \times X) \cap R$.

Definition 2.5. *For an interpretation $\mathcal{I}$ we define the* **output space** *$Y(\mathcal{I})$ as follows:*

$$Y(\mathcal{I}) = \bigcup_{u \in A} Y_u$$

where

$Y_a = \{Alg_a(ob(x), ob(y)) | (x, a, y) \in R_0\}$ *for* $a \in A_0$

$Y_{\theta(u,v)} = \{Alg_{\theta(u,v)}(o_1, o_2) | o_1 \in Y_u, o_2 \in Y_v\}$

We define recursively the valuation mapping

$$Val_{\mathcal{I}} : \mathcal{F}_{comp}(\mathcal{S}) \longrightarrow Y(\mathcal{I})$$

as follows:

- $Val_{\mathcal{I}}(h(x, a, y)) = Alg_a(ob(x), ob(y))$
- $Val_{\mathcal{I}}(\sigma(\alpha, \beta)) = Alg_{\theta(u,v)}(Val_{\mathcal{I}}(\alpha), Val_{\mathcal{I}}(\beta))$ if $sort(\sigma(\alpha, \beta)) = \theta(u, v)$.

3. RTN and a context free grammar

Another very important term for the phase we are considering is the grammar and the recursive transition network on which the algorithm of the phase is constructed.

A recursive transition network (shortly, RTN) is a graph theoretical schema used to represent the rules of a context free grammar. RTNs have application to programming languages, natural language and lexical analysis ([1], [20]). Any sentence that is constructed according to the rules of a RTN is said to be "well-formed." The structural

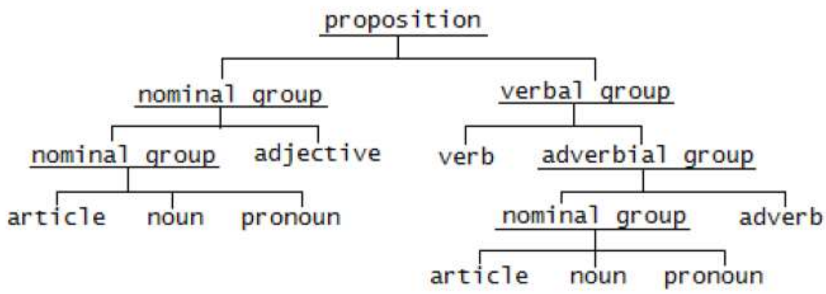

FIGURE 1. Communication user-VoSys

elements of a well-formed sentence may also be well-formed sentences by themselves, or they may be simpler structures. This is why RTNs are described as recursive.

We recall the concept of context free grammar defined in Chomsky's classification, that can approximate a natural language.

Definition 3.1. *A **context free grammar** is a system $G = (V_N, V_T, S, P)$, where*
- *V_N is a finite non-empty set of symbols named nonterminal symbols;*
- *V_T is a finite non-empty set of elements named terminal symbols;*
- *$S \in V_N$ is the initial symbol of the grammar;*
- *$P \subset V_N \times (V_N \bigcup V_T)^*$ is a non-empty set of elements named productions;*

In what follows we consider that a phrase is formed by sentences bounded together by conjunctions. A sentence can be formed by a verbal group or a nominal group and a verbal group. A nominal group can be formed by a nominal group and an adjective or by a noun or by a noun and an article. A verbal group can be formed by a verb and by an adverbial group. An adverbial group is formed by a nominal group and/or by an adverb. In the schema below are descried all the relations.

Following the above mentioned structure of the natural language we can form the below context free grammar $G = (V_N, V_T, S, P)$, where
- V_N = {S, Ng, Vg, Ag}, Ng represents a nominal group, Vg represents a verbal group and Ag represents an adverbial group;
- V_T = {art, noun, pro, vb, conj, adj, adv} where *art* represents an article, *pro* is a pronoun, *vb* represents a verb, *conj* represents a conjunction, *adj* represents an adjective and *adv* represents an adverb.
- $S \in V_N$ is the initial symbol of the grammar;
- $P \subset V_N \times (V_N \bigcup V_T)^*$ and P contains the following productions:
  1. $S \rightarrow < Vg >$
  2. $S \rightarrow < Ng >< Vg >$
  3. $Ng \rightarrow < Ng >< adj >$
  4. $Ng \rightarrow < noun >$
  5. $Ng \rightarrow < art >< noun >$
  6. $Ng \rightarrow < pro >$
  7. $Vg \rightarrow < vb >$
  8. $Vg \rightarrow < vb >< adv >$
  9. $Vg \rightarrow < vb >< Ag >$

(10) $Ag \rightarrow < Ng >< adv >$
(11) $Ag \rightarrow < Ng >$

In order to give a simple example of the use of this grammar the productions are restricted to the objects of the P1 sentence, where P1= "Peter is a doctor":

$S \longrightarrow_{(2)} < Ng >< Vg > \longrightarrow_{(4)} < noun >< Vg > \longrightarrow_{(9)} < noun >< vb >< Ag > \longrightarrow_{(5)} < noun >< vb >< art >< noun >$

The following productions characteristic to this sentence are added:
(12) $< noun > \rightarrow Peter$
(13) $< noun > \rightarrow doctor$
(14) $< verb > \rightarrow is$
(15) $< art > \rightarrow a$

Therefore by considering the last productions added, the sentence 1 can be obtained: $< noun >< vb >< art >< noun > \longrightarrow_{(12)} Peter < vb >< art >< noun > \longrightarrow_{(14)} Peter\ is < art >< noun > \longrightarrow_{(15)} Peter\ is\ a < noun > \longrightarrow_{(13)} Peter\ is\ a\ doctor$

As you can imagine, this was an exaggerated example of the process that is performed when working with a grammar. In an actual application, the last productions added will be different because the application should be connected to an enormous data base of words with different characteristics. Never the less, the process described above is at the base of the application an can be used to exemplify how different phrases and sentences can be formed or parsed using the grammar.

When communicating by writing or voice a person/application plays two different roles: the part of constructing a message and sending it and the part of receiving the message and decoding the message. In the example above it was shown how the grammar is used to construct a message - that is the first role of communicating. In the next section an algorithm used to decode a message will be constructed.

4. The message decoding algorithm

In the process of communication, the sender and the receiver of the message must use the same "signs" that have the same or similar meanings and the same code to encrypt and decode the message. With this in mind we conclude that for an effective communication to take place, the same grammar that was used to encrypt a message must be used also when decoding the message. This means that we have to test the message to see if it corresponds to the chosen grammar. We start by dividing the phrase into sentences and words. If the phrase is grammatically correct(from the point of view of the grammar that is described above) then a semantic schema must be constructed. We will use the constructed schema to accomplish a new interpretation of the phrase thus giving an answer to the phrase.

4.1. Step 1: Dividing the phrase into propositions. In order to obtain a semantic schema for the entire phrase we must first create the schemas for each proposition and then unite them all into one.

When one of the following punctuation signs is met **. ; ! ? ...** all the characters that stand before the sign are consider to form a proposition.

If we consider the phrase $\Phi =$ "Peter is a doctor. He works in a hospital. Maria is a teacher and she works in a school. Maria is the wife of Peter. Ann is their daughter." then by dividing it into sentences after the punctuation signs we will obtain the following propositions:
(1) $P_1 =$ "Peter is a doctor"

(2) P_2 = "He works in a hospital"
(3) P_3 = "Maria is a teacher and she works in a school"
(4) P_4 = "Maria is the wife of Peter"
(5) P_5 = "Ann is their daughter"

In the given example there was only one of the types of punctuation signs: the full stop or dot, but in real life conversations the question mark, exclamation mark, dots, semicolon appear just as often.

The set Υ_{signs} contains all the punctuation signs after which a phrase can be divided. The objects of this phrase depend on the grammar.

In a natural language $\Upsilon_{signs} = \{.,!,?,;,...\}$ where the sign "," is used in the set to separate the objects and must not be considered as a separator. Not all the signs present in the set are found in the example we gave, thus the set $\Upsilon_{signs} = \{.\}$.

Considering all of the above, we conclude that a phrase Φ can be divided into propositions by the objects of the set Υ_{signs}. After dividing Φ we can construct the set Φ_P that will contain all the sentences of the phrase; $\Phi_P = \{P_1, P_2, ..., P_n\}$ where n is the number of punctuation signs (objects of the Υ_{signs} set) found in the phrase.

For the above example $\Phi_P = \{P_1, P_2, P_3, P_4, P_5\}$, $n = 5$.

Remark 4.1. *It is very important to observe that the component propositions do not contain the separating punctuation signs.*

4.2. Step 2: Evaluating a proposition. The entire step 2 is to be repeated for all the propositions of Φ_P.

4.2.1. *Step 2.a: Parsing - Dividing the proposition into words.* When one of the following punctuation signs is met **, :** and/or the blanc character, all the characters that stand before the signs are consider to form a word.

If we consider the proposition P_1 = {Peter is a doctor} then by dividing it into words after the punctuation signs and after the blanc character we will obtain the following words:
(1) w_1 = "Peter"
(2) w_2 = "is"
(3) w_3 = "a"
(4) w_4 = "doctor"

The set υ_{signs} contains all the punctuation signs after which a sentence can be divided. The objects of this set depend on the grammar.

In a natural language $\upsilon_{signs} = \{,:\}$ and the blanc character, the sign "," is not used in the set to separate the objects and must be considered as a separator. Not all the signs present in the set are found in the example, thus the set υ_{signs} is formed by the blanc character.

Remark 4.2. *A proposition P_i can be divided into words by the objects of the set υ_{signs}. After dividing P_i, we can construct the set P_i^w that will contain all the sentences of the phrase; $P_i^w = \{w_1, ..., w_m\}$ where $i \leq n$, n is the number of punctuation signs (objects of the υ_{signs} set) found in the phrase at the above step of the algorithm and m is the number of blanc characters incremented by one.*

For the above example $P_1^w = \{w_1, w_2, w_3, w_4\}$ we have $m = 4$.

Remark 4.3. *It is very important to notice that the words do not contain the blanc character or any punctuation signs.*

4.2.2. *Step 2.b: Semantical and grammatical evaluation.* In order to analyze and work with a text we must first determine if the text is grammatically correct (from our defined context free grammar point of view).

Each word of Φ must be evaluated in relation with the parts of speech (verb, noun...). This will be done by searching the words in an online dictionary.

(1) P_1 ="Peter is a doctor"
- Peter - noun (proper noun, sg, M)
- is - verb
- a - article
- doctor - noun (common noun, sg)

"Peter is a doctor" is grammatically correct.

(2) P_2 = "He works in a hospital"
- he - pronoun (II person, sg, M)
- works - verb
- in -adverb
- a - article
- hospital - noun (common noun, sg)

"He works in a hospital" is grammatically correct.

(3) P_3 = "Maria is a teacher and she works in a school"
- Maria - noun (proper noun, sg, F)
- is - verb
- a - article
- teacher - noun (common noun, sg)
- and - conjunction
- she - pronoun (II person, sg, F)
- works - verb
- in - adverb
- a - article
- school - noun (common noun, sg)

"Maria is a teacher and she works in a school" is grammatically correct.

(4) P_4 = "Maria is the wife of Peter"
- Maria - noun (proper noun, sg, F)
- is - verb
- the - article
- wife - noun (common noun, sg)
- of - article
- Peter - noun (proper noun, sg, M)

"Maria is the wife of Peter" is grammatically correct.

(5) P_5 = "Ann is their daughter"
- Ann - noun (proper noun, sg, F)
- is - verb
- their - pronoun
- daughter - noun (common noun, sg)

"Ann it their daughter" is grammatically correct.

Although the phrase we analyzed was grammatically correct, it is very important to remember that while grammarians, writers of dictionaries, and language policy-makers all have a certain influence on the evolution of language, their ability to influence what people *think* and what they *ought* to say is distinct from what people actually *say*. This means that we could find a phrase that is not grammatically correct from the program's point of view but it is indeed a phrase that is used in the natural language.

4.2.3. *Step 2.c: Constructing the object set and the relation.* This is a very important step because we define the set of objects that we consider an output of a semantic schema.

Definition 4.1. *Let P_i be a proposition of Φ_P, $i \leq n$. We construct the set Ob_{P_i} of the elements of P_i^w that are nouns, adjectives and pronouns that are not in relation to the verb.*

This step is calculated by the computer through an algorithm that contains many rules. However this algorithm is not the subject of this article.
For P_1 we will have Ob_{P_1} = {Peter, doctor}.
For P_2 we will have Ob_{P_2} = {He, hospital}.
For P_3 we will have Ob_{P_3} = {Maria, teacher, she, school}.
For P_4 we will have Ob_{P_4} = {Maria, Peter}.
For P_5 we will have Ob_{P_5} = {Ann, their}.

4.2.4. *Step 2.d: Dividing into relations.* By looking at a proposition and at the corresponding object set it is easy to see that the verb and the other parts of the proposition form a bound, a relation between the two objects.

Definition 4.2. *Let P_i be a proposition of Φ_P, $i \leq n$ and Ob_{P_i} the set of objects defined above. The set Rel_{P_i} is formed by the verbs and the other parts of speech that are in relation to the verb.*

For P_1 we will have Rel_{P_1} = {is a}.
For P_2 we will have Rel_{P_2} = {works in a}.
For P_3 we will have Rel_{P_3} = {is a, works in a}.
For P_4 we will have Rel_{P_4} = {is wife of}.
For P_5 we will have Rel_{P_5} = {is daughter}.

4.2.5. *Step 2.e: Applying Simb.*

Definition 4.3. *Simb is a bijective function, Simb: $Ob_{P_i} \rightarrow X$ where X is a finite non-empty set of symbols from the definition of the semantic schema.*

Consider P_1, then Simb(Peter)=x_1 and Simb(doctor)=x_2, where $x_1, x_2 \in X_{P_1}$
$X_{P_1} = \{x_1, x_2\}$

4.2.6. *Step 2.f: Applying Label.*

Definition 4.4. *Label is a bijective function, Label: $Rel_{P_i} \rightarrow X$ where A_0 is a finite non-empty set of symbols from the definition of the semantic schema.*

Consider P_1, then Label(is a)=a, where $a \in A_0^{P_1}$
$A_0^{P_1}$ = {a}

4.2.7. *Step 2.g: Constructing the relations and the set R.*

Definition 4.5. *Let ObRel be a bijective function, ObRel:* $Ob_{P_i} \times Rel_{P_i} \times Ob_{P_i} \to X_{P_i} \times A_0^{P_i} \times X_{P_i}$*, ObRel* $= \{(x,a,y)$ *where* $x,y \in X_{P_i}$ *and* $a \in A_0^{P_i}$ $\}$. X_{P_i} *and* $\in A_0^{P_i}$ *are the finite non-empty sets of symbols from the definition of the semantic schema and (x,a,y) forms the* R_{P_i} *set.*

Consider P_1, then ObRel(Peter, is a, doctor)=(x_1, a, x_2), and $R_{P_1} = \{(x_1, a, x_2)\}$

We now have all the sets described in the semantic schema definition. We can define $S_{P_i} = (X_{P_i}, A_0^{P_i}, A_{P_i}, R_{P_i})$.

For P_1 we can define $S_{P_1} = (X_{P_1}, A_0^{P_1}, A_{P_1}, R_{P_1})$. The graphical representation is given in Figure 2.

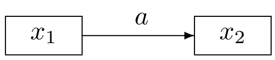

FIGURE 2. The graphical representation

5. Step 3: Constructing the semantic schema of the phrase

If we consider the phrase as a whole and not formed by several propositions then we must obtain one and only one semantic schema. We will obtain this by constructing the sets X, A_0, A and R that consist of the reunion of the sets X_{P_i}, $A_0^{P_i}$, A_{P_i}, R_{P_i} defined above.

Looking at this phrase, a human can immediately realize that the objects "he" and "she" refer to the subjects of the propositions that stand before them, in this case, "he" refers to "Peter" and "she" refers to "Maria".

We can put this new finding in a rule for the program to use in all the phrases from now on:

Rule: All the pronouns refer to a noun (usually a proper noun) that can be found in the set of the objects of the phrase - Ob_Φ - and that was already processed.

Example: Let's consider Ob_Φ **= {Peter, doctor, he, ...**. When we reach the "he" object, "Peter" is the only proper noun in the set and it is obviously chosen to replace "he".

This means that the nouns are written or spoken before the pronoun. In most cases the noun is also the subject of the proposition. The gender of the pronoun must be taken into account when searching for the appropriate noun. Also the singular or plural must be considered.

In the "she" object case, Ob_Φ **= {Peter, doctor, he, hospital, Maria, teacher, she,...**, the object "Maria" is the only proper noun that refers to a woman. If we look in the phrase (**... Maria is a teacher and she works in a school ...**) we can se that the object is found in the same proposition that contains the "she" object. Thus, the object "she" can be substituted with the object "Maria".

Also in this case a human can see that the pronoun "their" refers to "Peter" and "Maria". Still this case is a little different because the object "Ann" will now be bind to "Peter" and "Maria" through the relation "is daughter".

After this last modification of the phrase we can start creating the semantic schema.

We have the set Φ_F="Peter is a doctor. He works in a hospital. Maria is a teacher and she works in a school. Maria is the wife of Peter. Ann is their daughter."

Constructing the set Ob

Ob_{P_1} = {Peter, doctor}

Ob_{P_2} = {He, hospital} = {Peter, hospital}

Ob_{P_3} = {Maria, teacher, she, school} = {Maria, teacher, school}

Ob_{P_4} = {Maria, Peter}

Ob_{P_5} = {Ann, Maria, Peter}

$Ob_{\Phi} = Ob_1 \bigcup Ob_2 \bigcup Ob_3 \bigcup Ob_4 \bigcup Ob_5$ = {Peter, doctor, hospital, Maria, teacher, school, Ann}

Constructing the set Rel

Rel_{Φ} = {is a, works in a, is wife of, is daughter}

Constructing the X set

Simb(Peter) = x_1;

Simb(doctor) = x_2;

Simb(hospital) = x_3;

Simb(Maria) = x_4;

Simb(teacher) = x_5;

Simb(school) = x_6;

Simb(Ann) = x_7;

X = {x_1,..., x_7}

Constructing the set A_0

Label(is a) = a_1

Label(works in) = a_2

Label(wife of) = a_3

Label(is daughter of) = a_4

$A_0 = \{a_1, a_2, a_3, a_4\}$

Constructing the set R

ObRel(Peter, is a, doctor)=(x_1, a_1, x_2)

ObRel(Peter, works in a, hospital)=(x_1, a_2, x_3)

ObRel(Maria, is a, teacher)=(x_4, a_1, x_5)

ObRel(Maria, works in a, school)=(x_4, a_2, x_6)

ObRel(Maria, is wife of, Peter)=(x_4, a_3, x_1)

ObRel(Ann, is daughter of, Peter)=(x_7, a_4, x_1)

ObRel(Ann, is daughter of, Maria)=(x_7, a_4, x_4)

R = $\{(x_1, a_1, x_2),\ (x_1, a_2, x_3),\ (x_4, a_1, x_5),\ (x_4, a_2, x_6),\ (x_4, a_3, x_1),\ (x_7, a_4, x_1),\ (x_7, a_4, x_4)\}$

We have now constructed the schema $S = (X, A_0, A, R)$. The graphical representation is in Figure 3.

6. Working with the defined semantic schema

Now that we have constructed a final semantic schema for the entire phrase we can use this schema to construct other phrases and to extract knowledge from the phrase. This is done through the normal process of schema interpretation.

In a forthcoming paper the process of extended interpretation will be discussed.

References

[1] **L.Bolc** (Ed.):- The Design of Interpreters, Compilers,and Editors for Augmented Transition Networks, Springer-Verlag, Symbolic Computation, 1983

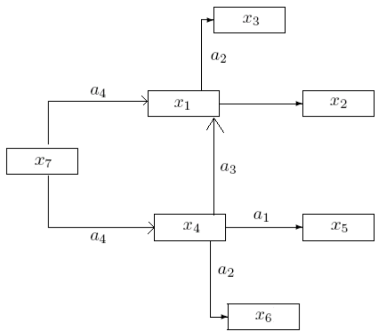

FIGURE 3. The semantic schema obtained by the algorithm

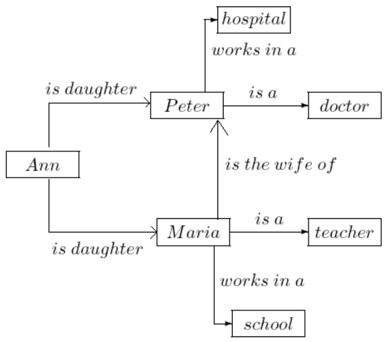

FIGURE 4. The meaning of the objects and relations

[2] **J.F. Sowa**- Conceptual structures- Information Processing in Mind and Machine, Addison-Wesley, 1984

[3] **J.F. Sowa** - Knowledge Representation: Logical, Philosophical and Computational Foundations, *Brooks Cole Publishing Co., Pacific Grove, CA.*, 2000.

[4] **Nicolae Tandareanu, Mihaela Ghindeanu** - Image Synthesis from Natural Language Description, Research Notes in Artificial Intelligence and Digital Communications, Vol.103, 3nd Romanian Conference on Artificial Intelligence and Digital Communications, Craiova, June 2003, p.82-96

[5] **N. Ţăndăreanu** - Semantic Schemas and Applications in Logical Representation of Knowledge, *Proceedings of the 10th Int. Conf. on CITSA*, Vol. III, July 21-25, Orlando, USA, 2004, 82-87.

[6] **N. Tandareanu, M. Ghindeanu** - Hierarchical Reasoning Based on Stratified Graphs. Application in Image Synthesis., 15th International Workshop on Database and Expert Systems Applications, Proceedings of DEXA2004, Zaragoza, Spania, IEEE Computer Society, Los Alamitos California, p.498-502, 2004

[7] **N. Ţăndăreanu** - Distinguished Representatives for Equivalent Labelled Stratified Graphs and Applications, *Discrete Applied Mathematics*, 144/1-2, 183-208, 2004.

[8] **Claudiu Poprlan** - A Java Implementation of Modeling Results About Stratified Graphs, Research Notes in Artificial Intelligence and Digital Communications, Vol.104, 4-nd Romanian Conference on Artificial Intelligence and Digital Communications, June 2004, Craiova, Romania, p.31-38, ISBN 973-671-014-9

[9] **Mihaela Ghindeanu, Claudiu Poprlan** - A Natural Language Processing System using Java-Prolog Technology, Vol.105, 5-nd Romanian Conference on Artificial Intelligence and Digital Communications, September 2005, Craiova, Romania, p.63-68, ISBN 973-671-055-6

[10] **N. Ţăndăreanu, M. Ghindeanu** - A Three-Level Distributed Knowledge System Based on Semantic Schemas, *16th Int. Workshop on Database and Expert Systems, DEXA'05, Copenhagen*, 2005, 423-427.

[11] **N. Ţăndăreanu**- Transfer of knowledge via semantic schemas, 9th World Multi-Conference on Systemics, Cybernetics and Informatics, July 10-13, Vol. IV, p.70-75, 2005

[12] **N. Ţăndăreanu** and **M.Ghindeanu**- Properties of derivations in a Semantic Schema, Annals of University of Craiova, Math. Comp. Sci. Ser.,Vol.33, p.147-153, 2006

[13] **N. Ţăndăreanu**- Semantic Schemas: The Least Upper Bound of Two Interpretations, 10th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, USA, July 16-19, Vol. III, p.150-155, 2006

[14] **N. Tandareanu** - Master-Slave Systems of Semantic Schemas and Applications, The 10th IASTED International Conference on Intelligent Systems and Control (ISC 2007), November 19-21, 2007, Cambridge, Massachusetts, USA, p.150-155, ISBN 978-0-88986-707-9

[15] **N. Tandareanu** - Cooperating Systems Based on Maximal Graphs in Semantic Schemas, Proceedings of the 11th WSEAS International Multiconference CSCC (Circuits, Systems, Communications, Computers), Vol. 4, p.517-522, Crete Island, Greece, July 23-28, 2007, ISSN: 1790-5117, ISBN: 978-960-8457-92-8

[16] **N. Tandareanu** - Cross-Deduction Based on Maximal Elements in Semantic Schemas, 11th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2007), Orlando, USA, July 8-11, Vol.I, p. 97-102, 2007

[17] **N. Tandareanu** - An Extension of Semantic Schemas to Represent Multiple Meanings, International Multi-Conference on Engineering and Technological Innovation (IMETI 2008), June 29th- July 2nd 2008,Orlando, USA, Vol.I, p.143-148, ISBN 978-1-934272-46-6

[18] **N. Tandareanu** - Generalized Semantic Schemas and Arguments to Use These Structures, 12th World Multiconference on Systemics, Cybernetics and Informatics (WMSCI 2008), Orlando, USA, June 29th- July 2nd, 2008, Vol.I, p. 56-61, ISBN 978-1-934272-31-2

[19] **N. Tandareanu, M. Ghindeanu**: - Hierarchical Semantic Structures Applied in Automatic Image Generation, The 11th IASTED International Conference on Intelligent Systems and Control (ISC 2008)

[20] **Woods William A.**:- Transition Network Grammars for Natural Language Analysis, Comm. ACM, 13, No. 10, October, 1970. Reprinted in Yoh-Han Pao and George W. Ernest (eds.), Tutorial: Context-Directed Pattern Recognition and Machine Intelligence Techniques for Information Processing. IEEE Computer Society Press, Silver Spring, MD, 1982. Also reprinted in Barbara Grosz, Karen Sparck Jones and Bonnie Webber (eds.), Readings in Natural Language Processing, San Mateo: Morgan Kaufmann, 1986

(Cristina Zamfir) Department of Informatics, University of Craiova, Al.I. Cuza Street, No. 13, Craiova RO-200585, Romania, Tel. & Fax: 40-251412673
E-mail address: cristina.zamfir@star-storage.ro