Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. IV (2009), No. 2, pp. 167-177

Hierarchical Distributed Reasoning System for Geometric Image Generation

Nicolae Ţăndăreanu, Mihaela Verona Ghindeanu, Sergiu Andrei Nicolescu

University of Craiova, Romania
Department of Mathematics and Computer Science
A.I. Cuza St, No. 13, 200585
E-mail: ntand@rdslink.ro, mghindeanu@yahoo.com

Abstract: The concept of hierarchical reasoning system was introduced in [5], where an intuitive method to build such systems based on their inputs is given. In this paper we formalize several concepts which open a possible research line concerning the use of these structures. A hierarchical reasoning system H is a directed graph organized on several levels such that each node of the level j is a hyper-schema of order j. As a mathematical structure, H is an abstract one and a special kind of *formal computation* is introduced. As a result of this computation we obtain a set $\mathcal{F}(H)$ of formulas. We explain what we understand by an *interpretation* of H and define its corresponding *semantical computation*. By means of an interpretation $\mathcal{I}(H)$ for H and applying the rules of the semantical computation, each element of $w \in \mathcal{F}(H)$ becomes some object $\mathcal{I}(w)$ of a given space. We exemplify these concepts and we show that for two distinct interpretations $\mathcal{I}_1(H)$ and $\mathcal{I}_2(H)$ for the same system H, a given formula $w \in \mathcal{F}(H)$ is transformed into a sentence $\mathcal{I}_1(w)$ of a natural language whereas $\mathcal{I}_2(w)$ is a geometric image. A short description of a Java implementation of a hierarchical system generating images is also given in a separate section. By examples we show that the mechanism introduced in this paper allows us to model the distributed knowledge. Finally several open problems are specified.
Keywords: semantic schema, interpretation, hyper-schema, distributed reasoning system, geometrical image generation

1 Introduction

Various kinds of mechanisms for image synthesis were presented and implemented on computer. The panel of the mathematical models for this subject includes the *rewriting systems* and *graph-based models*. Picture-processing grammars ([2]), picture grammars ([3]), stochastic grammars ([14]) and L-systems are some of the rewriting systems used to process images. The L-systems are a class of string rewriting mechanism originally developed by a biologist, A. Lindenmayer, in 1968 ([7]). The original emphases were on plant topology - spatial relations between cells or larger plant modules. The L-systems are a practical tool for generating fractal forms. Today these models are applied in architecture, physiology ([1]) and music. In order to interpret the L-system as music, LMUSe system ([9]) maps any of the turtle's 3D movement, orientation directions (forward, up, and left), its drawing line length, and thickness into musical pitches, note durations and volume.

A great number of research works and practical implementations have confirmed the interest of mathematicians and computer scientists in developing and applying the methods of graph theory. These methods were applied to obtain new knowledge representation models and to process images. A very productive notion with large applications in knowledge representation is that of *conceptual graph*, a notion introduced in literature by J.F.Sowa ([8],[10]). We can find several applications of the graph-based methods in [6] (low-level processing of digital images, learning algorithms for high-level computer vision and pattern recognition).

The concept of semantic schema was introduced in [11] as an extension of semantic networks. This structure is obtained by means of a labeled graph and a Peano algebra built over the edge labels. Since then many applications of this structure were presented (new semantics in logic programming, knowledge representation for intelligent dialog systems etc).

In [12] we defined a new mechanism for generating images similar with the edge rewriting in the way that both approaches can be used to define complex images based on some simple other images. In the mentioned paper the concept of Hierarchical Distributed Reasoning System was introduced. Each leaf of the system is given by a semantic schema. The other nodes are hyper-schemas ([12]). We presented an *intuitive method* to obtain geometrical images. The leaves represent the input of the system in semantic schemas and, by appending proper interpretations, they obtain the graphical illustrations of the received inputs. In this manner the leaves obtains the initial images. Then, at the upper levels, these images are combined by hyper-schemas to obtain complex images. We obtained a *bottom-up method* to obtain images from initiators.

In this paper we obtain the following results:

- Starting with the concept of Hierarchical Distributed and Reasoning System (HGR system) introduced in [12] in Section 3 we define a formal computation in such a structure. As a result of this computation we obtain a set $\mathcal{F}(H)$ of formulas for an arbitrary HDR system H. This is the *formal computation* in an HDR system.

- An HDR system H is an abstract structure. In Section 4 we introduce the concept of *interpretation* for H. By means of an interpretation $\mathcal{I}(H)$ for H each element of $\mathcal{F}(H)$ becomes some object of a given space. This gives the *semantical computation*. Both the formal and semantical computations are exemplified. We show that for two distinct interpretations $\mathcal{I}_1(H)$ and $\mathcal{I}_2(H)$ for the same system H we can generate sentences in a natural language giving the reasoning conclusions and geometrical images respectively.

- A short description of a Java implementation of an HDR system is also given in Section 5.

- By examples we show that the mechanism introduced in this paper allows us to model the distributed knowledge.

- The last section contains the conclusions and future works. Several open problems are specified in this section.

2 Basic concepts

Consider a symbol θ of arity 2. A θ-semantic schema ([11]) or shortly, θ-schema is a system $\mathcal{S} = (X, A_0, A, R)$, where:

- X is a finite non-empty set of symbols named object 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 of relations which fulfills the following conditions:

  1. $(x, \theta(u, v), y) \in R \Rightarrow \exists z \in X : (x, u, z) \in R, (z, v, y) \in R$
  2. $\theta(u, v) \in A, (x, u, z) \in R, (z, v, y) \in R \Rightarrow (x, \theta(u, v), y) \in R$

3. $\{\alpha \mid \exists(x, \alpha, y) \in R\} = A$

An element from $R \cap (X \times A_0 \times X)$ is a *regular arc* of $\mathcal{S}$.

We denote by $Ded(\mathcal{S})$ the least set satisfying the following properties ([13]):

• If $(x, a, y) \in R_0$ then $([x, y], a) \in Ded(\mathcal{S})$

• If $([x_i, \ldots, x_k], u) \in Ded(\mathcal{S})$ and $([x_k, \ldots, x_r], v) \in Ded(\mathcal{S})$, $i < k < r$ and $\theta(u, v) \in A$ then $([x_i, \ldots, x_r], \theta(u, v)) \in Ded(\mathcal{S})$.

An element of $Ded(\mathcal{S})$ is a **deductive path** of $\mathcal{S}$.

Let us consider the schemas $\mathcal{S}_1 = (X_1, A_{01}, A_1, R_1)$ and $\mathcal{S}_2 = (X_2, A_{02}, A_2, R_2)$. In the remainder of this section we describe a new structure which relieves a special kind of cooperation between $\mathcal{S}_1$ and $\mathcal{S}_2$.

If $d_1 = ([x, \ldots, y], u) \in Ded(\mathcal{S}_i)$ and $d_2 = ([y, \ldots, z], v) \in Ded(\mathcal{S}_{3-i})$, where $i \in \{1, 2\}$, then we say that d_1 is **connected to right** by d_2 or d_2 is **connected to left** by d_1. We say that d_1 is **connected** by d_2 if d_1 is connected to right or to left by d_2.

We consider the sets of deductive paths $L_1 \subseteq Ded(\mathcal{S}_1)$ and $L_2 \subseteq Ded(\mathcal{S}_2)$. We say that $L_1 \cup L_2$ is a **pairwise connected set of deductive paths** if every deductive path of L_i is connected by some deductive path of L_{3-i}.

For each $i \in \{1, 2\}$ we consider a set V_i of symbols such that $V_i \cap (A_1 \cup A_2) = \emptyset$. We consider also a set E_i such that $E_i \subseteq X_i \times V_i \times X_i$, $Card(E_i) = Card(L_i)$ and $E_1 \cap E_2 = \emptyset$. Consider also a bijective mapping $g_i : L_i \longrightarrow E_i$ such that $g_i(d) = (x, e, y)$, where $d = ([x, \ldots, y], \theta(u, v)) \in L_i$. This mapping transforms each deductive path $([x, \ldots, y], \theta(u, v))$ from L_i into a regular arc (x, e, y). Shortly, we say that the path $([x, \ldots, y], \theta(u, v))$ **is designated** by (x, e, y). We can define now a cooperating structure of hyper-schemas.

A **hyper-schema of order zero** is a semantic schema. Consider the hyper-schemas $\mathcal{S}_1$ and $\mathcal{S}_2$ of order zero. A **hyper-schema of order one** over $\mathcal{S}_1$ and $\mathcal{S}_2$ obtained by means of L_1 and L_2 is a θ-schema $\mathcal{S}$ which includes the regular arcs obtained from L_1 and L_2 ([12]). We denote by $Hyp_1(\{\mathcal{S}_1, \mathcal{S}_2\})$ the set of all hyper-schemas of first order over $\mathcal{S}_1$ and $\mathcal{S}_2$. In general we write $S \in Hyp_k(\{S_1, \mathcal{S}_2\})$ and we name S a **hyper-schema of order** k if S_1 and S_2 are hyper-schemas of order $j \leq k-1$ and at least one of them has the order $k - 1$.

An HDR system ([12]) is the tuple $H = (Q_1, Q_2, \ldots, Q_k)$ where $k \geq 2$ and

- $Q_1 = \{\mathcal{S}_1, \ldots, \mathcal{S}_{n_1}\}$, $n_1 > 1$, constitutes the first level of the system. The entities $\{\mathcal{S}_1, \ldots, \mathcal{S}_{n_1}\}$ are hyper-schemas of order zero. The set Q_1 gives the leaves of H.

- $Q_2 = \{\mathcal{S}_{n_1+1}, \ldots, \mathcal{S}_{n_2}\}$, $n_2 \geq n_1 + 1$, gives the second level of the system and $\mathcal{S}_{n_1+1}, \ldots, \mathcal{S}_{n_2}$ are hyper-schemas of order 1. More precisely, for every $m \in \{n_1 + 1, \ldots, n_2\}$ there are $m_1, m_2 \in \{1, \ldots, n_1\}$, $m_1 \neq m_2$ such that $\mathcal{S}_m \in Hyp_1(\{\mathcal{S}_{m_1}, \mathcal{S}_{m_2}\})$.

- For $j \in \{3, \ldots, k\}$, $Q_j = \{\mathcal{S}_{n_{j-1}+1}, \ldots, \mathcal{S}_{n_j}\}$ represents the j-th level of the system, where $n_j \geq n_{j-1} + 1$. For every $m \in \{n_{j-1} + 1, \ldots, n_j\}$ there is $m_1 \in \{n_{j-2}, \ldots, n_{j-1}\}$ and there is $m_2 \in \{1, \ldots, n_{j-1}\}$ such that $\mathcal{S}_m \in Hyp_{j-1}(\{\mathcal{S}_{m_1}, \mathcal{S}_{m_2}\})$.

3 Formal computations in HDR Systems

Suppose that $H = (Q_1, Q_2, \ldots, Q_k)$ is an HDR system. The components of H are the hyper-schemas $\mathcal{S}_1, \ldots, \mathcal{S}_{n_k}$. We can visualize H as a graph structure. In order to obtain this structure we represent each hyper-schema by a node and we draw two directed arcs from $\mathcal{S}_r$ to $\mathcal{S}_j$ and to $\mathcal{S}_m$ if $\mathcal{S}_r \in Hyp_p(\{S_j, S_m\})$ for some p. The structure obtained in this manner is not a tree. This can be observed in Figure 1: there are two distinct paths from $\mathcal{S}_7$ to $\mathcal{S}_2$ and there is no root of this structure.

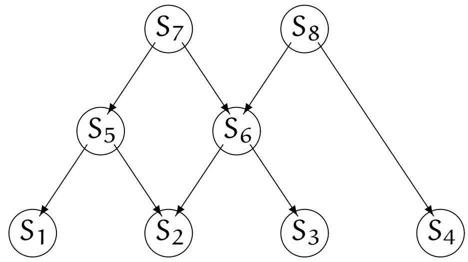


Figure 1: The graph structure of H

For each $i \in \{1, \ldots, n_k\}$ we consider that $\mathcal{S}_i$ is given by the tuple $\mathcal{S}_i = (X_i, A_{0i}, A_i, R_i)$ and we denote $R_{0i} = R_i \cap (X_i \times A_{0i} \times X_i)$. For each $r \in \{n_1 + 1, \ldots, n_k\}$ such that $\mathcal{S}_r$ is a hyper-schema over $\mathcal{S}_j$ and $\mathcal{S}_m$ in H we consider:

• the connected sets $L_{j,r} \subseteq Ded(\mathcal{S}_j)$ and $L_{m,r} \subseteq Ded(\mathcal{S}_m)$;

• the sets $E_{j,r}$, $E_{m,r}$ and the transformational mappings $g_{j,r} : L_{j,r} \longrightarrow E_{j,r}$, $g_{m,r} : L_{m,r} \longrightarrow E_{m,r}$.

By the assumptions of the previous section we have $R_{0r} \supseteq E_{j,r} \cup E_{m,r}$. We denote $N_{0r} = E_{j,r} \cup E_{m,r}$. Obviously we have the following property:

Proposition 1. $N_{0i} = \emptyset$ if and only if $\mathcal{S}_i$ is a leaf of H.

For a symbol h of arity 1 we consider the set:

$$M = \bigcup_{i=1}^{n_k} \{ h([x, y], a) \mid (x, a, y) \in R_{0i} \setminus N_{0i}\}$$

where we used the notation $h([x, y], a)$ instead of $h(([x, y], a))$.

We consider the symbols $\sigma_1, \ldots, \sigma_{n_k}$ of arity 2 and denote by $\mathcal{H}_H$ *the Peano* $\{\sigma_1, \ldots, \sigma_{n_k}\}$*-algebra generated by* M.

We consider the alphabet Z including the symbols σ_i, the elements of X_i, the elements of A_i, the left and right parentheses, the square brackets [and], the symbol h and comma. As in the theory of formal languages, the set Z^* defines all the words over Z. Because a hyper-schema is a semantic schema we have the following property:

Proposition 2. If $\mathcal{S}_i$ is a hyper-schema of H and $([x_1, \ldots, x_{k+1}], \theta(u, v)) \in Ded(\mathcal{S}_i)$ then there is r uniquely determined such that $([x_1, \ldots, x_{r+1}], u) \in Ded(\mathcal{S}_i)$ and $([x_{r+1}, \ldots, x_{k+1}], v) \in Ded(\mathcal{S}_i)$.

Definition 1. Let be $w_1, w_2 \in Z^*$. We define the following binary relation on Z^*, denoted by $\Rightarrow_H$:

• For $i \in \{1, \ldots, n_k\}$, if $(x, e, y) \in R_{0i} \setminus N_{0i}$ then $w_1([x, y], e)w_2 \Rightarrow_H w_1 h([x, y], e)w_2$;

• For $i \in \{1, \ldots, n_k\}$, if $(x, e, y) \in N_{0i}$ then $w_1([x, y], e)w_2 \Rightarrow_H w_1 d w_2$, where d is the deductive path designated by (x, e, y);

• Suppose that $([x_1, \ldots, x_{k+1}], \theta(u, v)) \in Ded(\mathcal{S}_i)$, $i \in \{1, \ldots, n(H)\}$, $([x_1, \ldots, x_{r+1}], u) \in Ded(\mathcal{S}_i)$ and $([x_{r+1}, \ldots, x_{k+1}], v) \in Ded(\mathcal{S}_i)$ then:

$w_1([x_1, \ldots, x_{k+1}], \theta(u, v))w_2 \Rightarrow_H w_1 \sigma_i(([x_1, \ldots, x_{r+1}], u), ([x_{r+1}, \ldots, x_{k+1}], v))w_2$

The reflexive and transitive closure of $\Rightarrow_H$ is denoted by $\Rightarrow_H^*$. We denote $\mathcal{F}(\mathcal{S}_i) = \{w \in \mathcal{H}(H) \mid \exists d \in Ded(\mathcal{S}_i) : d \Rightarrow_H^* w\}$ and $\mathcal{F}(H) = \bigcup_{i=1}^{n_k} \mathcal{F}(\mathcal{S}_i)$.

Let us exemplify this computation. We consider the hyper-schemas $\mathcal{S}_1$ and $\mathcal{S}_2$ of order zero from Figure 2 and the hyper-schema of order 1 from Figure 3.

If we take

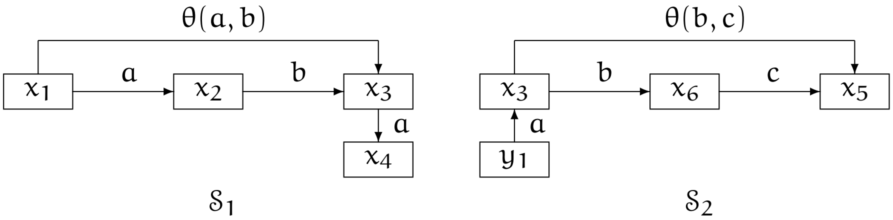


Figure 2: Semantic schemas $\mathcal{S}_1$ and $\mathcal{S}_2$ of order zero

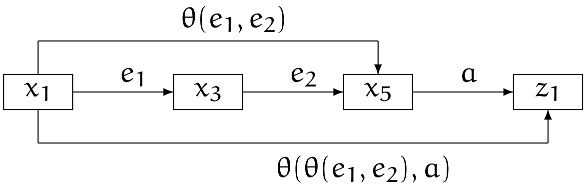


Figure 3: Hyper-schema $\mathcal{S}_3 \in Hyp_1(\{\mathcal{S}_1, \mathcal{S}_2\})$

- $L_{1,3} = \{([x_1, x_2, x_3], \theta(a, b))\}$, $L_{2,3} = \{([x_3, x_6, x_5], \theta(b, c))\}$
- $E_{1,3} = \{(x_1, e_1, x_3)\}$, $E_{2,3} = \{(x_3, e_2, x_5)\}$
- $g_{1,3}([x_1, x_2, x_3], \theta(a, b)) = (x_1, e_1, x_3)$, $g_{2,3}([x_3, x_6, x_5], \theta(b, c)) = (x_3, e_2, x_5)$

then we obtain the following computations:

- $([x_1, x_3, x_5], \theta(e_1, e_2)) \Rightarrow_H \sigma_3(([x_1, x_3], e_1), ([x_3, x_5], e_2))$
- $([x_1, x_3], e_1) \Rightarrow_H ([x_1, x_2, x_3], \theta(a, b)) \Rightarrow_H \sigma_1(([x_1, x_2], a), ([x_2, x_3], b)) \Rightarrow_H^*$ $\sigma_1(h([x_1, x_2], a), h([x_2, x_3], b)) \in \mathcal{F}(\mathcal{S}_1)$
- $([x_3, x_5], e_2) \Rightarrow_H ([x_3, x_6, x_5], \theta(b, c)) \Rightarrow_H \sigma_2(([x_3, x_6], b), ([x_6, x_5], c)) \Rightarrow_H^*$ $\sigma_2(h([x_3, x_6], b), h([x_6, x_5], c)) \in \mathcal{F}(\mathcal{S}_2)$

In conclusion,
$([x_1, x_3, x_5], \theta(e_1, e_2)) \Rightarrow_H^* \sigma_3(\sigma_1(h([x_1, x_2], a), h([x_2, x_3], b)), \sigma_2(h([x_3, x_6], b), h([x_6, x_5], c)))$
and the last formula is an element of $\mathcal{F}(H)$, where $H = (Q_1, Q_2)$, $Q_1 = \{\mathcal{S}_1, \mathcal{S}_2\}$ and $Q_2 = \{\mathcal{S}_3\}$.

4 Semantical computations in HDR systems

The semantical computation in an HDR system H transforms every formula of $\mathcal{F}(H)$ into an object of some space. In this section we describe this transformational process.
Let us consider the HDR system $H = (Q_1, Q_2, \dots, Q_k)$ and an element $w \in \mathcal{F}(H)$. If $d = ([x_1, \dots, x_k], \theta(u, v)) \in Ded(S_i$ and $d \Rightarrow_H^* w$ then we write $sort(w) = \theta(u, v)$.

Definition 2. An **interpretation** for H is a system $\mathcal{I} = (Ob, ob, ALG)$:
- Ob is a set of objects;
- $ob : X \longrightarrow Ob$, where $X = \bigcup_{i=1}^{n_k} X_i$, is a mapping that "interprets" each node as an object;
- $ALG = \bigcup_{i=1}^{n_k} \{Alg_u^i\}_{u \in A_i}$, where Alg_u^i is an algorithm with two input arguments and one output argument such that if $g_{j,k}([x, \dots, y], \theta(u, v)) = (x, e, y)$ then $Alg_e^k = Alg_{\theta(u,v)}^j$.

Definition 3. The **valuation mapping** Val_H of the HDR system H is defined as follows:

- If $w = h([x, y], a) \in \mathcal{F}(H)$ then $Val_H(w) = \bigcup_{i=1}^{n_k} \{Alg_a^i(ob(x), ob(y))\}$.

- If $w = \sigma_j(w_1, w_2) \in \mathcal{F}(H)$, $w_1 \in \mathcal{F}(H)$, $w_2 \in \mathcal{F}(H)$ and $sort(w) = \alpha$ then

$$Val_H(w) = \{Alg^j_\alpha(o_1, o_2) \mid o_k \in Val_H(w_k), k = 1, 2\}$$

In order to exemplify the computations we consider again the HDR system H from Section 3. We define an interpretation of H by means of some *sentential forms*. Such a structure is a sentence containing two variables. If we substitute each variable by an object then a sentential form becomes a sentence in a natural language. We shall consider the following sentential forms:

$p_1(x, y)$="x is the father of y"; $p_2(x, y)$="x is the mother of y";
$p_3(x, y)$="x is the brother of a y"; $p_4(x, y)$="x likes to eat y";
$q_1(x, y)$="x is the grandmother of y"; $q_2(x, y)$="a brother of x likes to eat y";
$r(x, y)$="A nephew of x likes to eat y";

We consider the following algorithms:

Algorithm $Alg^1_a(o_1, o_2)$ { return $p_1(o_1, o_2)$}; **Algorithm** $Alg^1_b(o_1, o_2)$ { return $p_2(o_1, o_2)$};
Algorithm $Alg^2_b(o_1, o_2)$ { return $p_3(o_1, o_2)$}; **Algorithm** $Alg^2_c(o_1, o_2)$ { return $p_4(o_1, o_2)$};
Algorithm $Alg^1_{\theta(a,b)}(o_1, o_2)$ { if $o_1 = p_1(t_1, t_2)$, $o_2 = p_2(t_2, t_3)$ then return $q_1(t_1, t_3)$}
Algorithm $Alg^2_{\theta(b,c)}(o_1, o_2)$ { if $o_1 = p_3(t_1, t_2)$, $o_2 = p_4(t_2, t_3)$ then return $q_2(t_1, t_3)$}
Algorithm $Alg^1_{e_1}(o_1, o_2)$ { return $q_1(o_1, o_2)$};
Algorithm $Alg^1_{e_2}(o_1, o_2)$ { return $q_2(o_1, o_2)$};
Algorithm $Alg^1_b(o_1, o_2)$ { return $p_2(o_1, o_2)$};
Algorithm $Alg^3_{\theta(e_1,e_2)}(o_1, o_2)$ { if $o_1 = q_1(t_1, t_2)$, $o_2 = q_2(t_2, t_3)$ then return $r(t_1, t_3)$}

Consider the interpretation $\mathcal{I}_1 = (Ob_1, ob_1, ALG_1)$ of the system H, where we specify only the useful entities allowing to exemplify the computation:

- $Ob_1 = \{Peter, Helen, John, Sorin, pizza\}$
- $ob_1(x_1) = Peter, ob_1(x_2) = Helen, ob_1(x_3) = John, ob_1(x_6) = Sorin, ob_1(x_5) = pizza$
- $ALG_1 = \{Alg^1_a, Alg^1_b, Alg^2_b, Alg^2_c, Alg^1_{\theta(a,b)}, Alg^2_{\theta(b,c)}, Alg^3_{e_1}, Alg^3_{e_2}, Alg^3_{\theta(e_1,e_2)}\}$

where $Alg^3_{e_1} = Alg^1_{\theta(a,b)}, Alg^3_{e_2} = Alg^2_{\theta(b,c)}$.

It is not difficult to observe that for the formula

$w = \sigma_3(\sigma_1(h([x_1, x_2], a), h([x_2, x_3], b)), \sigma_2(h([x_3, x_6], b), h([x_6, x_5], c))) = \sigma_3(\alpha, \beta)$

from the last part of the previous section we obtain the following computations:

$Val_H(\alpha) = \{Alg^1_{e_1}(o_3, o_4) \mid o_3 \in Val_H(h([x_1, x_2], a)), o_4 \in Val_H(h([x_2, x_3], b))\}$

$Val_H(h([x_1, x_2], a)) = \{Alg^1_a(Peter, Helen)\} = \{p_1(Peter, Helen)\}$

$Val_H(h([x_2, x_3], b)) = \{Alg^1_b(Helen, John), Alg^2_b(Helen, John)\} = \{p_2(Helen, John), p_3(Helen, John)\}$

therefore $Val_H(\alpha) = \{Alg^1_{e_1}(p_1(Peter, Helen), p_2(Helen, John)), Alg^1_{e_1}(p_1(Peter, Helen), p_3(Helen, John))\} = \{q_1(Peter, John)\}$

$Val_H(\beta) = \{Alg^2_{e_2}(o_5, o_6) \mid o_5 \in Val_H(h([x_3, x_6], b)), o_6 \in Val_H(h([x_6, x_5], c))\}$

$Val_H(h([x_3, x_6], b)) = \{Alg^1_b(John, Sorin), Alg^2_b(John, Sorin)\} = \{p_2(John, Sorin), p_3(John, Sorin)\}$

$Val_H(h([x_6, x_5], c)) = Alg^2_c(Sorin, pizza)\} = \{p_4(Sorin, pizza)\}$

therefore $Val_H(\beta) = \{Alg^2_{e_2}(p_2(John, Sorin), p_4(Sorin, pizza)), Alg^2_{e_2}(p_3(John, Sorin), p_4(Sorin, pizza))\} = \{q_2(John, pizza)\}$

Finally, from $Val_H(\alpha)$ and $Val_H(\beta)$ we deduce

$Val_H(w) = \{Alg^3_{\theta(e_1,e_2)}(q_1(Peter, John), q_2(John, pizza))\} = \{A\ nephew\ of\ Peter\ likes\ to\ eat\ pizza\}$

We observe that the conclusion obtained by H can not be obtained neither by $\mathcal{S}_1$, neither by $\mathcal{S}_2$. This explains why H is named a *distributed system*.

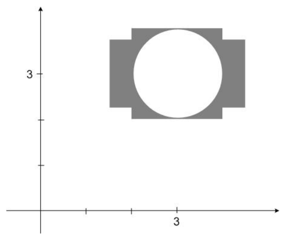


Figure 4: The image generated by $\mathcal{I}_2$

We give now a short description of another interpretation $\mathcal{I}_2$ for the same system H. As a result we obtain geometrical images.

• $Ob_2 = \{1, (3,3), (3,1.5)\}$

• $ob_2(x_1) = 1, ob_2(x_2) = ob_2(x_6) = (3,3), ob_2(x_3) = 1, ob_2(x_5) = (3,1.5)$

• $Alg^1_a(p,q)$\{*Return the interior of circle with radius p and center q*\}

• $Alg^1_b(p,q)$\{*Return the interior of the square centered in p and the sides of length 2*q parallel with coordinate axes* \}

• $Alg^1_{\theta(a,b)}(\alpha,\beta)$\{ If $\alpha = Alg^1_a(p,q)$ and $\beta = Alg^1_b(q,r)$ then return $\beta \setminus \alpha$ \}

• $Alg^2_b(p,q)$\{*Return the exterior of circle with radius p and center q*\}

• $Alg^2_c(p,q)$\{*Return the interior of the rectangle centered in p and the sides of lengths specified by q, parallel with coordinate axes* \}

• $Alg^2_{\theta(b,c)}(\alpha,\beta)$\{ If $\alpha = Alg^1_{\theta(a,b)}(p,q)$ and $\beta = Alg^2_{\theta(b,c)}(q,r)$ then return $\beta \cap \alpha$ \}

• $Alg^3_{\theta(e_1,e_2)}(\alpha,\beta)$\{ If $\alpha = Alg^2_a(p,q)$ and $\beta = Alg^2_c(q,r)$ then return $\beta \cup \alpha$ \}

For the same formula $w \in \mathcal{F}(H)$ as in the previous computation, the object $Val_H(w)$ given by $\mathcal{I}_2$ is shown in Figure 4.

5 A Java implementation

If we note by $\mathcal{A}$ the set consisting of some geometrical objects names then each system's input is an word $w = a_1 \dots a_k$ over the alphabet $V = \mathcal{A} \cup \{+,-\}$ having the following properties:

• $\mathbf{a_i} = +/-$ means a left/right rotation with a specific angle, denoted by δ and to draw a line on the current direction

• $\mathbf{a_i} = \mathbf{O_j}$ means to draw the graphical illustration of the object O_j such that its entry direction is on the current direction. In our implementation, each geometrical object used in the generation method is an instance of the a class named *Object*. Graphically, it is a representation of a figure inside a square. Every instance of this class can have one of the following types: **circle**, **triangle**, **star** and **square** corresponding to the figure it consists of. Other members of this class are the **entry direction** and the **exit direction** related to some corner of the object. The corner corresponding to the entry direction becomes the **entry point** of the object. Similar for the **exit point**. The main routine of the Algorithm is **createHDRS** (Algorithm 2). The construction of the system starts by defining the schemas of the agents (steps $1 \div 4$). The hyper-schemas of order one corresponding to the managers of the second level are constructed using the steps $7 \div 14$. The condition for existing a hyper-schema over two schemas is that their maximal paths are connected deductive paths. This property is verified using the routine **connectedPaths** (Algorithm 3). If the second level of the system was successfully defined (**If** condition of step 15) then the process of creating new levels in HDRS continues using the **While** loop of step 17. The hyper-schemas of orders greater than 2 are created using the routine **createHypSchs** (Algorithm 4).

The geometrical objects that are used for the image generation process are introduced using the first

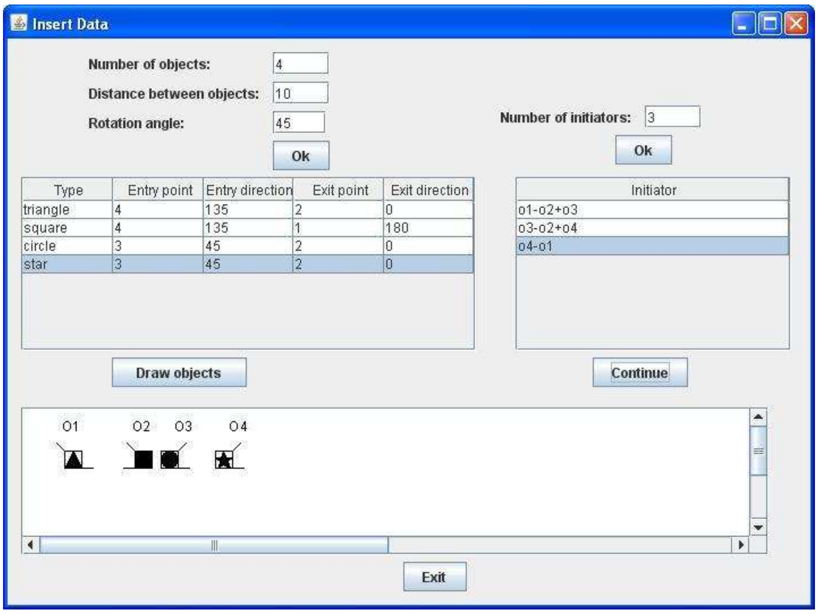
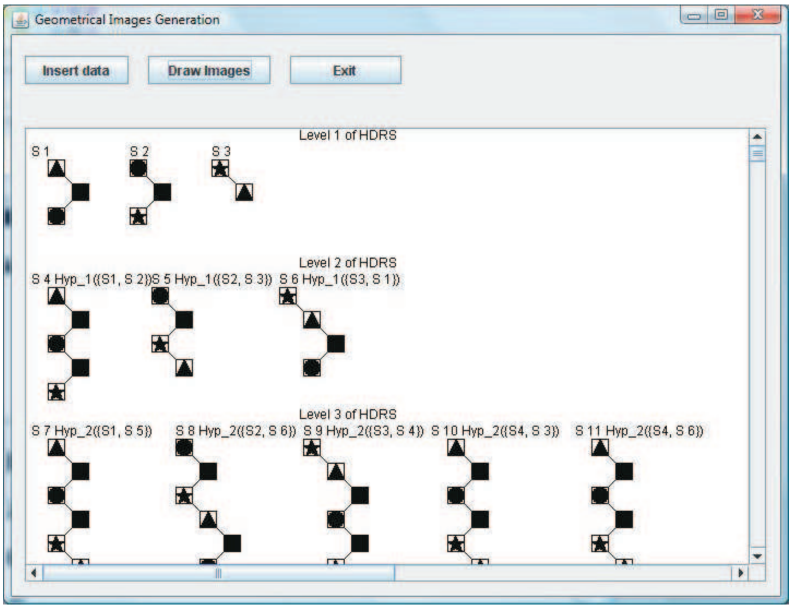


Figure 5: First window of the application

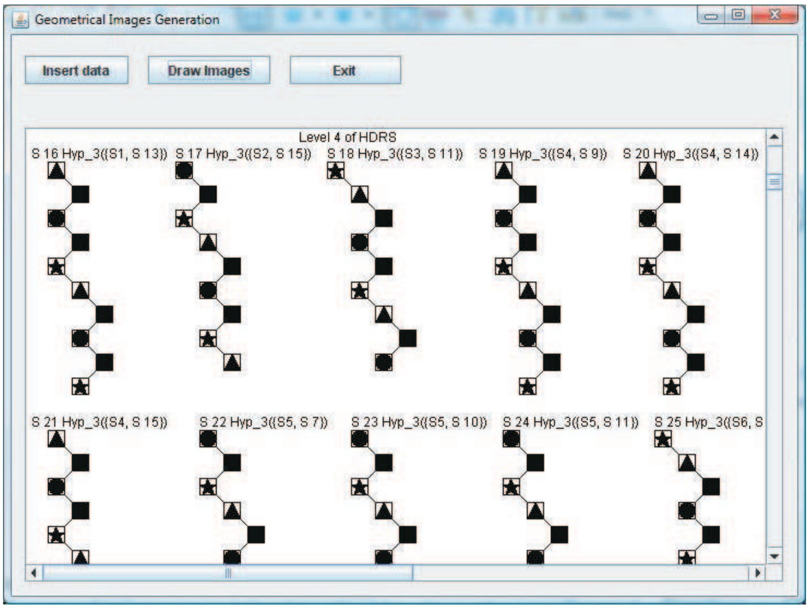
window of the application. For each object the user must specify the type, the entry and exit points (the corners are numbered starting from the down-left) and related to them the entry and the exit direction. Also, using the controls of the first window, the input descriptions can be edited(see Figure 5). The second window of the application gives the outputs provided by the system's reasoning components (see Figure 6). It consists of three buttons and a panel. The application can draw maximum 1000 images with maximum 50 geometrical objects per image.

6 Conclusions and future works

In this paper we formalized the syntactical and semantical computations in an HDR system. We exemplified these computations and for some HDR system H we gave two interpretations: one interpretation generates phrases and the other generates geometrical images. This examples give an idea concerning the generative power of our mechanism. We relieved also by these examples the fact that the distributed reasoning can be modeled by an HDR system. A short description of a Java implementation for an HDR system generating images is also given. We intend to develop the applications of an HDR system. First, we intend to use the mobile agents to process such systems ([4]). Second, we intend to use the HDR systems in e-learning. The basic idea comes from the fact that a link in an HTML document gives a reference to another document of the similar structure.



(a) The initiators and some images obtained by the managers of Q_2 and Q_3 levels



(b) Images obtained at the 4th level in the system

Figure 6: Second window of the application

Algorithm 2 Procedure createHDRS

```
Procedure createHDRS
    For i ← 1, noCmd
      call create_schema(commands[i], schema[i], agent[i])
      maximalPath[i] ← schema[i].getMaximalPath()
    EndFor
    noAg ← noComd
    noKM ← noAg + 1
    For i, j ← 1, noAg; j ≠ i
        If connectedPaths(maximalPath[i], maximalPath[j])
            call create_hyperSch(hypSch[noKM], schema[i], schema[j])
              hypSch[noKM].order ← 1
              maximalPath[noKM] ← hypSch[noKM].getMaximalPath()
              noKM ← noKM + 1
          EndIf
     EndFor
     If noKM > noAg + 1
          order ← 2
          While createHypSchs(order)
              order ← order + 1
          EndWhile
     EndIf
EndProcedure
```

Algorithm 3 Function connectedPaths

```
Function connectedPathsPath1, Path2
    If Path1.lastNode=Path2.firstNode
        return true
    EndIf
    If Path1.firstNode=Path2.lastNode
        return true
    EndIf
    return false
EndFunction
```

Algorithm 4 Function createHypSchs

```
FunctioncreateHypSchsorder
    newHypSch ← false
    For i ← noKM − 1, noAg
        If hypSch[i].order ≠ order − 1
            continue
        EndIf
        For j ← 1, noKM − 1; j ≠ i
            If connectedPaths(maximalPath[i], maximalPath[j])
                newHypSch ← true
                If j ≤ noAg
                    call create_hyperSch(hypSch[noKM], hypSch[i], schema[j])
                 Else
                    call create_hyperSch(hypSch[noKM], hypSch[i], hypSch[j])
                 EndIf
                 hypSch[noKM].order ← order
                 maximalPath[noKM] ← hypSch[noKM].getMaximalPath()
                 noKM ← noKM + 1
            EndIf
        EndFor
     EndFor
     return newHypSch
EndFunction
```

Bibliography

[1] Allen M., Prusinkiewicz P., DeJong T. (2004) *Systems for Modeling the Architecture and Physiology of Growing Trees: The L-PEACH Model*, Proceedings of the 4thInternational Workshop on Functional-Structural Plant Models, pp. 220-225

[2] Chang Shi-Kuo (1970) *The analysis of two-dimensional patterns using picture processing grammars*, Annual ACM Symposium on Theory of Computing archive, Proceedings of the second annual ACM symposium on Theory of computing, p. 206-216

[3] Drewes F., Ewert S., Klempien-Hinrichs R., Kreowsky H.J. (2003) *Computing raster images from grid picture grammars*, Journal of Automata, Languages and Combinatorics, Vol.8, Issue 3, p. 499-519

[4] Dzitac I., Bărbat B. E. (2009) Artificial Intelligence + Distributed Systems = Agents, *Int. J. of Computers, Communications* & *Control*, ISSN 1841-9836, E-ISSN 1841-9844, vol. IV, no. 1, pp. 17-26

[5] Ghindeanu M. (2008) Constructing Architectures for an Hierarchical Distributed Reasoning System Based on its Inputs, *International Multi-Conference on Engineering and Technological Innovation*, USA, p. 231-234

[6] Kandel A., Bunke H., Last M. (eds) (2007) Applied Graph Theory in Computer Vision and Pattern Recognition, Springer, Studies in Computational Intelligence 52

[7] Lindenmayer A. (1968) *Mathematical models for cellular interaction in development*, Parts I and II, Journal of Theoretical Biology (18), p. 280-315.

[8] Priss U., Corbett D., Angelova G. (Eds.) (2002) Conceptual Structures: Integration and Interfaces, *10th Int. Conf. on Conceptual Structures, ICCS 2002*

[9] Sharp D. (1998) *LMUSe version 0.7b,* http://www.geocities.com/Athens/Academy/8764/ lmuse/l-musetxt.html

[10] Sowa J.F. (1984) Conceptual structures- Information Processing in Mind and Machine, Addison-Wesley

[11] Ţăndăreanu N. (2004). *Semantic schemas and applications in logical representation of knowledge*, Proceedings of the 10th International Conference on Cybernetics and Information Technologies, Systems and Applications, USA, Vol.III, p. 82-87

[12] Ţăndăreanu N., Ghindeanu M. (2008) Hierarchical Semantic Structures Applied in Automatic Image Generation, *Proceedings of 11th IASTED International Conference on Intelligent Systems and Control*, ISBN: 978-0-88986-777-2

[13] Ţăndăreanu N., Ghindeanu M. (2008) Path-based Reasoning in Semantic Schemas, *Annals of University of Craiova, Mathematics and Computer Science Series*, Vol.35, p.171-181

[14] Zu Song-Chun , Mumford D. (2006) *A stochastic grammar of images*, Foundations and Trends in Computer Graphics and Vision, Vol. 2, Issue 4, p. 259-362