Random Graph Languages for Distorted and Ambiguous Patterns: Single Layer Model

Marek R. Ogiela
AGH University of Science and Technology
30 Mickiewicza, 30-059
Krakow, Poland
mogiela@agh.edu.pl

Marcin Piekarczyk
Institute of Computer Science
Pedagogical University of Cracow
2 Podchorazych, Krakow, Poland
marp@up.krakow.pl

***Abstract* — The work introduces a linguistic based model designed for distorted or ambiguous patterns where a graph based approach is used for structure representation. The knowledge about unevenness is usually created on the basis of finite number of patterns treated as positive samples of unknown language. The IE graphs are used as the base. Single pattern can be represented using deterministic IE graph. Subsequently, the collection of patterns, represented by deterministic graph is transformed into equivalent random graph language. Utilization of the grammatical inference mechanisms gives the possibility to perform this process in automatic way. Using the IE graphs and imposing some simple limitations on graph structures allows to obtain a polynomial complexity of knowledge inference. In the work it is described how to use the proposed model for collecting the knowledge in handwritten signatures recognition and analysis systems. Information about graphemes (solid fragment of handwritten signature) variability is stored in the form of random IE graphs and stochastic ETPL(k) graph grammars. Instead of an ordinary the IE graph, an attributed one is used in order to increase a descriptive power of the proposed schema. The parametrical data embedded in the graph carries some additional semantic information associated with the structure of pattern. The work presents discussion about inference scheme and computational complexity of the proposed linguistic representation scheme. Described methodology can be especially suited for creating the knowledge representation of the handwritten signatures, signs and ideograms (e.g. kanji) in offline recognition systems.** *(Abstract)*

Keywords – graph grammar, ETPL(k), attributed random IE graph, knowledge-based systems, IE graph, ambiguous patterns, attribute-controlled graph grammar, grammatical inference, heterogenous parsing, random languages.

I. INTRODUCTION

In pattern recognition there is often necessity to analyze structures with some sort of variability. It may result either from the natural features of the recognized pattern as it is the case with handwritten signatures (*it is impossible to obtain two identical representations of the signature*) or from the technical reasons that cause some disorders in the analyzed scenes. Application of neural networks or some statistical methods based on HMM, DTW or SVN approaches usually brings satisfactory results. Those methods, though promising in some situations also have a significant disadvantage – the knowledge gathered during the recognition process and subsequently used for analysis as well as classification unknown objects remains unrevealed. This implies considerable problems when the system is supposed to analyze characteristic features of the teaching sample which was the base for recognition process. We can avoid such inconvenience if the syntactic methods are used as the model for knowledge representation. For those methods the knowledge takes a form of formal language describing the teaching sample or set of input objects and is stored as collection of rules corresponding formal grammar. This may be significantly facilitated if the variability of objects would be analyzed with regard to their spatial structure and some other features. As far as the quality of the analysis is concerned, it is important to ensure the appropriate descriptive power of the applied syntactic methodology and effective of the parsing mechanism.

The aim of this work is to present the author's original concept based on the use of IE graphs [2], random IE graphs [12] and probabilistic grammars [4] belonging to the ETPL(k) class with the purpose of gathering and analyzing the knowledge about the structure and features of ambiguous objects (variant or distorted). An example of recognizing the components of a handwritten signature (*graphemes*) should serve as a means to show the work of such model, however it may equally well be used for the representation and analysis of some other patters with variable structure. In the next chapter (chapter II) the way of describing the objects (*scenes, patterns*) if the attributed IE graphs are applied (*section II*) is presented. The following parts (*section III*) refer to the construction of knowledge (*language*) by the use of IE graphs and probabilistic grammars of ETPL(k) class as well as to the model of pattern analysis i.e. parsing process (*section IV*).

II. REPRESENTATION OF THE AMBIGUOUS/DISTORTED PATTERNS

In general every scene (*pattern*) should be described in the meaning of its structural representation, namely the graph structure. The scope of initial processing and type of parametric information which should be collected in order to ensure the precise and firm identification of the recognized objects depends on how specific are the analyzed patterns. For the purpose of this work the model of representation of handwritten signatures described in [9,10] will be applied. More technical details about the discussed model can also be found there. According to that model a single object (*grapheme*) is presented as the attributed IE graph [8,10], in which semantic information could be associated both with the nodes and the edges. In case of handwritten signs representation additional information referring to the shape parameters are associated only with the graph nodes. Graph nodes represent the primary components (*curves*), while edges - the relation of direct contiguity (*the touching of curves*).

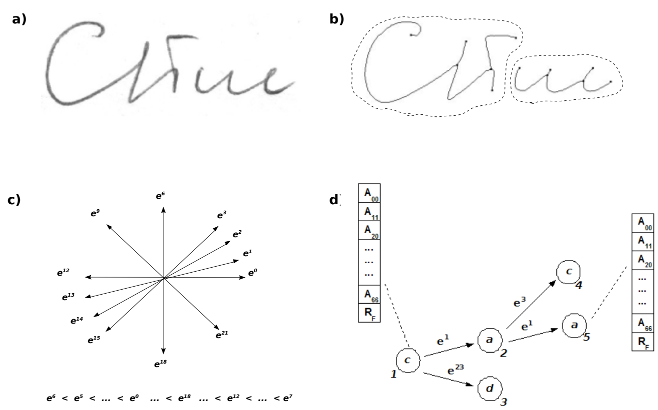


Figure 1. Representation of a single pattern: a) original, grayscale image of the signature, b) a skeleton of the signature with indicated points that separate the primary components and graphemes (*dashed line*), c) ordered set of the edge labels, d) representation of the first grapheme in a form of attributed IE graph with indicated sets of parameters.

In Fig. 1 a sample representation of a grapheme in a form of an attributed IE graph is presented. Set of the directional labels determining spatial relations among the components of the object described with a resolution implied by the specifics of the object. Similarly the sets of attributes associated with the graph nodes may consist of any composition of the parameters properly reflecting the features of primary components. In the described model in [9,10] the appropriate set of attributes constitutes a collection of Zernike moments while the directional labels are defined with resolution of 15 angle degrees. In practical use of pattern recognition methods the necessary knowledge is derived from a finite set of examples, that create so called teaching sample.

For the purpose of this work it has been assumed that the input teaching sample consists of the handwritten signatures set. For every signature we can create a structural representation which has a form of deterministic attributed IE graph (*compare theoretical considerations in [10]*). In Fig. 2 are presented separate graphs corresponding to the leading graphemes of each signature.

In the knowledge construction process apart from the singular objects representation also the definition of mechanism that allows to store aggregated information concerning acceptable level of scenes variation (*acceptable variants*) is required. In order to fulfill such requirements the formalisms based on random attributed IE graphs has been proposed (def. 1), as it is necessary to associate the parametric semantic information creating the shape vector with every node as it was in case of graphs representing individual objects (*graphemes*).

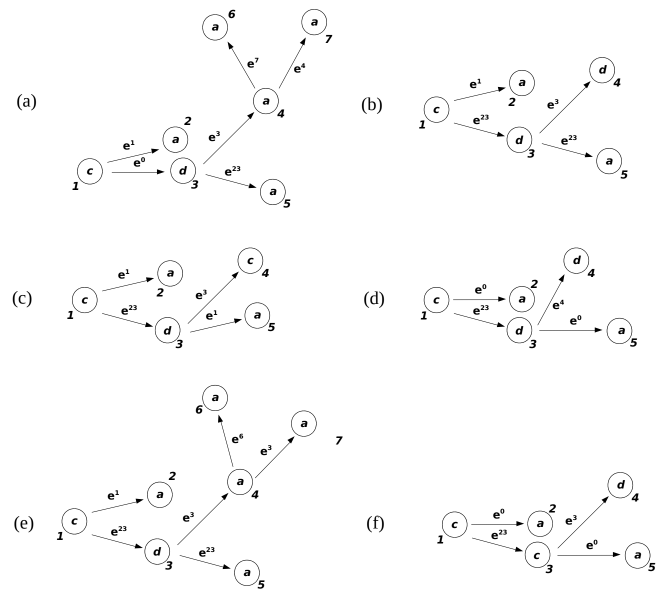


Figure 2. Attributed IE graphs presenting the leading graphemes of every signature belonging to training set (*sets of attributes associated to the nodes have been omitted for better clarity of the picture*).

Definition 1. The random attributed IE graph is known as a seven $G = (V, E, \Sigma, \Gamma, \phi, \eta_V, \eta_E)$, where:

a) $G^* = (V, E, \Sigma, \Gamma, \phi)$ is a random IE graph concordant with the definition in [12],

b) η_V, η_E are representations attributing graph nodes (1) and edges (2), respectively:

$$\eta_V : V \rightarrow A = \{A_i\} \; A_i = \bigcup_{\delta \in \Sigma} \Omega_\delta \;\; \forall i \neq j \;\; A_i \cap A_j = \varnothing \quad (1)$$

$$\eta_E : E \rightarrow B = \{B_i\} \; B_i = \bigcup_{\psi \in \Gamma} \Omega_\psi \;\; \forall i \neq j \;\; B_i \cap B_j = \varnothing \quad (2)$$

and fulfilling conditions (3) and (4):

$$\forall v \in V \qquad \eta_V(v) \in \Omega_{\phi(v)} \qquad (3)$$

$$\forall e = (u, \sigma, w) \in E \qquad \eta_E(e) \in \Omega_\sigma \qquad (4)$$

From figure 2 it can be quite easily noted that the graph representation of individual graphemes described in a form of attributed IE graphs shows some structural similarity. Between the groups of graphs indicated as {3a, 3e} and {3b, 3c, 3d, 3f} there is a structural isomorphism, that can be represented by the random IE graphs. This isomorphism should be understood here as a bijection [12] preventing the change of the graph structure but allows some variance on the level of node and directional labels.

Transformation from the deterministic description (*representation of individual graphemes*) to the random one (*aggregated information about all accepted variants of the object*) can be performed by the creation of the appropriate subsets representing those elements of structurally isomorphic graphs and then constructing the random graphs on their basis. Fig. 3 shows random graphs corresponding to isomorphic categories of graphs presented in Fig. 2.

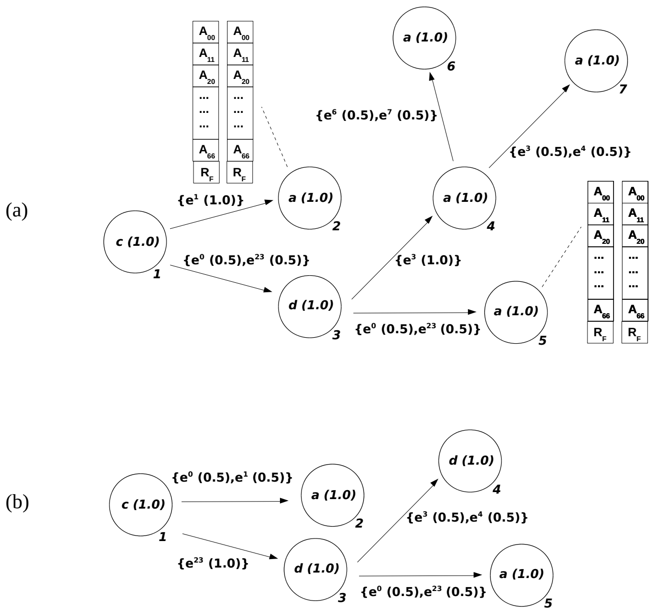


Figure 3. Attributed random of the IE graphs representing variability of graphemes (*structural isomorphism*) for graphs presented in Fig. 2: a) random graph that aggregates the features of graphs presented in Fig. 2a and 2e, b) random graph which aggregates the features of graphs presented in Fig. 2b, 2c, 2d and 2f.

Therefore the random graphs resulting from that process, describe full range of variability of teaching sample both in its structural and parametric layer. Subsequently an appropriate random language which can be a base for a recognition system may be constructed. With regard to the practical applicability of such language creation its automation must be ensured.

III. RANDOM LANGUAGE – GRAMMATICAL INFERENCE

In order to apply syntactic methods in pattern recognition processes the appropriate base of knowledge, consisting of the descriptions sample images (graphemes) must take a form of certain language L(G), generated by the adequately defined formal grammar. The example used for the purpose of this work was based on a stochastic (probabilistic) graph grammar G which belongs to the class ETPL(k) [4]. Grammars of that type generate random IE graphs, which are suitable for describing the structure of ambiguous scenes, where the disorders and unexpected changes of shapes are highly likely to appear.

The process of automatic inference for grammars, and for graph grammars in particular is always very sophisticated and expensive. In order to reduce such requirements certain limitations concerning the structure of words belonging to the language inferred and in consequence on the way how the graphs describe single images must be imposed. Namely it is assumed that there are no edges joining the nodes on a particular level. It must be stressed however that such limits imposed on the way the graph is stretched out on the elements of scene do not significantly decline the possibility of representing effectively the recognized scene in the classical pattern recognition tasks, which means the descriptive power of such scenes is not decreased. On the other hand, introduction of those limits has allowed to suggest a fast grammatical inference method aimed to distinguish category of graphs, which is less demanding for calculation power than the other general inference algorithms for IE graphs already described [1]. A brief concept of such algorithm can be described as follows:

Let us assume that the algorithm has a task to indicate complete set of production rules for a grammar G, which is capable to generate the random graph R, given as the input for that algorithm. In addition, every node will be generated directly i.e. by the use of a non-terminal label. While processing an input graph R will be reduced by eliminating every normalized, two-level complete NCTL graph (def. 2) and corresponding production rules will be generated instead. The direction of reduction process reflects the decreased order of the terminal graph nodes indexes.

Definition 2. If R is a random IE graph and CTL(i) is a complete two-level CTL graph, originated in a terminal node of the graph R indexed i as defined in [2]. A CTL*(i)* graph from which all the nodes that have a level equal to a terminal node i have been removed is called a normalized graph. Such graph is described by the symbol NCTL*(i).*

The explanation how the grammatical inference mechanism is conducted will be presented using the random IE graph R (Fig. 4) as a sample. It is necessary to note that every node in graph R has individually calculated node level [2]. Precise information about node levels is useful during concluding process when the right sides of productions will be created. The modifications like removing consistent subgraph from analyzed graph R make no effect on node levels if are performed in the order of decreasing indexes beginning from the maximum one. Therefore the disposable calculation of the node levels is enough.

Each concluding stage requires obtaining the following set of data needful to complete production:

- *NCTL(i)* – normalized complete CTL graph originated in node indexed with *i* and denoted by terminal label,
- *CD(i)* – characteristic description of terminal node indexed with *i*,
- *PC(i)* – preceding context for terminal node indexed with *i* [Fla93],
- *PCC(i,j)* (ang. *Preceding Context for Consequent*) – preceding context for consequent node belonged to NCTL(i) graph and indexed with *j* – only edges outer against NCTL(i) graph are included,
- *RE(i)* (ang. *Rejected Edges*) – set of nodes belonged to CTL(i) graph and rejected because level conflict with originated node is found.

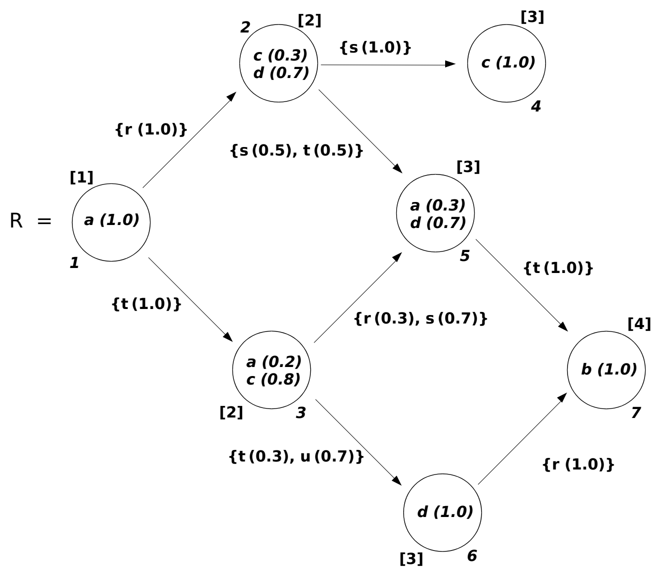


Fig. 4. Input random IE graph for grammatical inference algorithm.

All productions are constructed sequentially and denoted by temporary non-terminal labels A_i where $i \in N^+$. In graph R (Fig. 4) the procedure starting from the node indexed with 7 since it is the greatest index in graph R where terminal label exists. In this case the CTL(7) graph has a form as in Fig. 5 and it is identical with NCTL(7) (*one terminal node without consequents*).

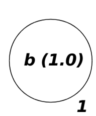


Fig 5. Random IE graph denoted as NCTL(7) .

The rest of datasets state as following:

$\underline{CD(7)}$	$\underline{PC(7)}$	$\underline{PCC(7,-)} = \varnothing$
$b(1.0)_7$	$(\{a(0.3), d(0.7)\}, \{t(1.0)\})$	
–	$(\{d(1.0)\}, \{r(1.0)\})$	$RE(7) = \varnothing$
–		
–		

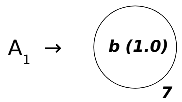
Basing on data presented above we can create a production of stochastic ETPL(k) graph grammar in the form $p = (l, D, C)$ where the terminal node $l = A_1$. The right side of the production denoted by *D* makes a form of *NCTL(7)* graph. Embedding transformations from dataset denoted by *C* are composed on the basis of PC set (*input edges associated with currently removing node*), PCC set (*input edges associated with all the rest nodes of D graph*) and characteristic description of originated node (*output edges*). To formulate the general rules determining how to construct embedding transformations it is necessary to put some denotations. Let us assume that index *i* is associated with originated node for *NCTL(i)*, *n(i)* gives the random label for node indexed with *i*, $PC(i) = \{(A_k, X_k)\}$ where A_k i B_k are random labels for graph nodes and $PCC(i,j) = \{(B_k, Y_k)\}$ where X_k i Y_k are random labels for graph edges. In such case the embedding transformations connected with input edges are constructed as following:

- for input edges basing on *PC* set: $C(X_k, \{in\}) = \{(n(i), A_k, X_k, \{in\})\}$
- for input edges basing on *PCC* set: $C(Y_k, \{in\}) = \{(n(j), B_k, Y_k, \{in\})\}$

Moreover, assuming that $RE(i) = \{w_l\}$ is the set of indices representing the nodes removed from *NCTL(i)* graph due to constraints connected with node levels (*nodes from the set* $V\{CTL(i)\}/V\{NCTL(i)\}$) and characteristic description *CD(i)* includes a random edge in the form (i, λ_l, w_l) then the embedding transformations connected with output edges are constructed according to the general rule presented below:

- for output edges leading into nodes belonging to *RE* set: $C(n(i), \{out\}) = \{(n(i), \lambda_l, n(w_l), \{out\})\}$

According to introduced scheme the embedding transformations for terminal node indexed with 7 can be determined basing only on *PC* set because *PCC* and *RE* sets are both empty. In this case *PC(7)*includes two elements, thus two embedding transformations will be created. These transformations are associated with the input edges leading to node indexed with 7. The appropriate production with embedding transformations is presented in Fig. 6.



$C_1(\{t\,(1.0)\}, in) = \{(\{b\,(1.0)\},\{a\,(0.3), d\,(0.7)\},\{t\,(1.0)\}, in)\}$

$C_1(\{r\,(1.0)\}, in) = \{(\{b\,(1.0)\},\{d\,(1.0)\},\{r\,(1.0)\}, in)\}$

Fig 6. Production of stochastic grammar graph defined for subgraph of R originated from node indexed with 7.

The last step, the subgraph NCTL(7) of graph *R* is replacing by non-terminal label A_1. Subsequently the next terminal node with greatest index is determined and concluding process continues. The operations are repeated until terminal node indexed with 1 will be encountered. Then such NCTL(1) graph is recognized as grammar axiom and inference algorithm stops.

Finally the inference scheme can be presented as in Fig. 7:

```
S:=R;
P=ø;
SetNodeLevels(S);
repeat
 PCC=ø; NCTL=ø; G=ø;
 i:=GetMaxTermIndex(S);
 if i=1 then
  Z:=S; {grammar axiom}
 else
  begin
  RunInferenceRandomIE(S,I,G);
  ConstructNCTL(S,I,G,NCTL,PCC,RE);
  CD:=GiveCD(S,i);
  PC:=MakePC(S,i);
  ComposeProduction(P,NN,NCTL,CD,PC,PCC,RE,CS);
  RemoveNCTL(S,I,NCTL,NN,CS);
  end
until i=1;
```

Fig 7. Algorithm of application rules of the unknown grammar sETPL(k) for input sentence in the form of random IE graph.

Formally computational complexity of the proposed inference algorithm as a function of the size (*number of nodes*) random input graph can be estimated as $O(n^2)$. Presented inference algorithm could also be directly applied for the attributed random IE graphs. No extra modifications of the algorithm are required provided that the information concerning the sets of attributes, linked to the terminal nodes of input graph is transferred into the corresponding NCTL graphs that represent right side of the production rules of an expected stochastic grammar. As the inference process is always carried indirectly i.e. by the use of non-terminal node such collocation in case of a particular NCTL sub-graph must be fixed only for the attachment graph, as only this graph represents a terminal node in the entire graph. In such a case appropriate procedures – called semantic actions – must be included directly in *ConstructNCTL* procedure. The number of parameters sets assigned to the particular terminal node is represented by constant, independent from the size of input graph; therefore it does not influence the computational complexity of the entire inference algorithm.

IV. PARSING OF THE DETERMINISTIC STRUCTURES FOR THE GRAPH LANGUAGES

Though the base of knowledge describing the distorted or variant objects could easily be presented in the form of language that generates random IE graphs (*attributed*) the single object (*scene*) being recognized must be described by the deterministic IE graphs i.e. graphs describing the images without any distortions. In such a case the necessity of syntactic analysis heterogeneous structures arises as the adherence of the inspected deterministic IE graph (describing the object) to the language generated by the probabilistic grammar of sETPL(k) class (*i.e. the language consisting of the random IE graphs*) must be examined. In the subject literature we can find the description of the syntactic analyzers of homogenous structures which can be applied both for deterministic graph grammars of ETPL(k) class [2] as well sd for stochastic (*probabilistic*) grammars of ETPL(k) class [4]. However none of them are adequate for the model considered here. Searching for an appropriate concept of parsing graph grammars of ETPL(k) class, that would include also heterogenic structures, as it is consideration here, the works of [1,4] have to be referred. The author of those works presents the solution to the problem of syntactic analysis where the adherence of IE graph derived from a random rIE graph to the language generated by the deterministic graph grammar of ETPL(k) class has been investigated. It means the following reasoning scheme: **deterministic** graph grammar of ETPL(k) class → **random** IE graph. For the purpose of recognition model described in this work another mechanism is needed, namely: **stochastic** graph grammar of ETPL(k) class → **deterministic** IE graph. That is why necessary was to define a new parsing algorithm devoting to the described task. Proposed model of an syntactic analyzer was based on the classical reasoning scheme usually applied for ETPL(k) class grammars [2]. In this scheme generational type of a single-run parsing scheme without return (*top-down strategy*) is applied. Because of the semantic information (*parameters*) built into the graphs the attribute controlled reasoning must be applied (*def. 3*).

Definition 3. A stochastic attribute controlled graph grammar of class ETPL(k) above sets of attributes A, B is known as a six G = (Σ, Δ, Γ, P, Z, f_Z), where:

a) Σ, Δ, Γ, Z are determined as for the grammar sETPL(k) defined in accordance with [4],

b) $f_Z : \Theta_{A,B}(\overline{D}) \rightarrow \{TRUE, FALSE\}$ is a starting predicate, where the graph $\overline{Z}$ is created from the start symbol Z by removing non-terminal nodes,

c) P is a production set of the form p=(l, D, C, f), where:

– (l, D, C) is a probabilistic production in accordance with the definition of the sTLP grammar (definition in [6]),

$f : \Theta_{A,B}(\overline{D}) \rightarrow \{TRUE, FALSE\}$ is the predicate of applicability of production p, where the graph $\overline{D}$ is created from graph G by the removal of non-terminal nodes.

In the described model the recognition process of an unknown image, represented by the deterministic IE graph H means finding the answer to the question whether it is possible to derive a random IE graph R which belongs to the language L(G) generated by the stochastic graph grammar of class ETPL(k), for which exists a deterministic output IE graph that is isomorphic with the investigated graph H. The applicability predicates added to attribute-controlled grammar allow us to determine, this on the basis of the semantic context (*attribute values*), whether a specific production may be applied at the given stage of argumentation.

Formally the parsing algorithm can be presented as in the Fig. 8.

```
R:=Z;
PCEL=ø; error:=0;
TestAxiomAP(H,R,error);
for i:=1 to n do
 if error=0 then
  begin
   if ϕR(i)is a nonterminal node then
   begin
    m:=GetMaxInd(R)+1;
    Construct_k-TL(H,i,m,W);
    T:=GenerateTTLN(ϕR(i));
    O:=GenerateOTLN(W,T);
    ChooseProduction(W,O,p);
    if p=0 then error:=1 else ApplyProduction(R,i,p);
   end;
  if not CheckCI(H,R,i) then CheckPCI(H,R,i,error);
  TestPCI(PCEL,R,error);
 end.
```

Fig 8. Parsing algorithm of deterministic attributed IE graphs for probabilistic grammars class ac-sETPL(k)

Single step of reasoning process in ac-sETPL(k) grammar has a pessimistic complexity of O(n). Parsing requires exactly *n* steps so the final estimation of time complexity of the full parser algorithm for attribute controlled grammar sETPL(k) is $O(n^2)$ and it is similar to classical algorithms available for class deterministic ETPL(k) languages.

V. Conclusion

The mechanism of description and inference of variable structures presented in this work has been applied in practice as an element of system for graph biometric verification based on the handwritten signatures [9,10]. It seems plausible that the proposed model of description the distorted scenes based on the use random languages of ETPL(k) class could equally be applied in other domains. Particularly promising could be its application in cognitive analysis [5-7] and knowledge collecting systems e.g. image understanding [11] or learning systems [14].

References

[1] M. Flasinski, "Structural pattern recognition using ETPL(k) graph grammar," Jagiellonian University, D. Sc. Thesis, Cracow 1992.

[2] M. Flasinski, "On the Parsing of Deterministic Graph Languages for Syntactic Pattern Recognition," Pattern Recognition, Vol. 26, 1993. No. 1, pp. 1-16.

[3] M. Flasinski, "Power Properties of NLC Graph Grammars with a Polynomial Membership Problem," Theoretical Computer Science, Vol. 201, 1998, No. 1, pp. 189-231.

[4] M. Flasinski, M. Skomorowski, "Parsing of Random Graph Languages for Automated Inspection in Statistical-based Quality Assurance Systems," Machine GRAPHICS & VISION International Journal, Vol. 7, 1998, No. 3, pp. 565-623.

[5] L. Ogiela, M. R. Ogiela, "Cognitive Techniques in Visual Data Interpretation," Springer Verlag, Berlin-Heidelberg, 2009

[6] M. R. Ogiela, R. Tadeusiewicz, L. Ogiela, "New Classes of UBIAS and E-UBIAS Cognitive Vision Systems", ICISA 2010 - The International Conference on Information Science and Applications (ICISA 2010), April 21st - 23rd, 2010, Seoul, Korea, Vol.2, pp. 721-726.

[7] M. R. Ogiela, R. Tadeusiewicz, "Towards New Classes of Cognitive Vision Systems", CISIS 2010 – the 4th International Conference on Complex, Intelligent and Software Intensive Systems, February, 15th - 18th 2010, Krakow, Poland, pp. 851-855.

[8] P. Oleksik., "Syntactic pattern recognition in visual inspection system using stochastic ETPL(k) graph grammars," AGH University of Science and Technology, Ph.D. thesis, Cracow 2000.

[9] Piekarczyk, M., "Hierarchical Random Graph Model for Off-line Handwritten Signatures Recognition", Proc. of International Conference on Complex, Intelligent and Software Intensive Systems 2010 (CISIS2010/IMIS2010), IEEE CS Press.

[10] Piekarczyk M., Ogiela M.R., "Hierarchical Graph-Grammar Model for Secure and Efficient Handwritten Signatures Classification," Vol. 17, 2011, Issue 6, pp. 926 – 94.

[11] Ryszard Tadeusiewicz, Marek R. Ogiela, "Medical Image Understanding Technology," Springer Verlag, Berlin-Heidelberg, 2004.

[12] M. Skomorowski, "On the parsing of random graphs for syntactic pattern recognition," Machine GRAPHICS & VISION International Journal, Vol. 5, 1996, pp. 433-464.

[13] Skomorowski, M., " Use of random graph parsing for scene labelling by probabilistic relaxation," Pattern Recognition Letters, Vol. 20, 1999, Issue 9, pp. 949-956.

[14] K. Wojcik, "Hierachical Knowledge Structure Applied to Image Analysis System – Possibilities of Practical Usage," LNCS, 2011, Vol. 6908/2011, pp. 149-163.