World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:7, 2007

Syntactic Recognition of Distorted Patterns

Marek Skomorowski

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/14780/pdf

***Abstract*—**In syntactic pattern recognition a pattern can be represented by a graph. Given an unknown pattern represented by a graph g, the problem of recognition is to determine if the graph g belongs to a language $L(G)$ generated by a graph grammar G. The so-called *IE* graphs have been defined in [1] for a description of patterns. The *IE* graphs are generated by so-called $ETPL(k)$ graph grammars defined in [1]. An efficient, parsing algorithm for $ETPL(k)$ graph grammars for syntactic recognition of patterns represented by *IE* graphs has been presented in [1]. In practice, structural descriptions may contain pattern distortions, so that the assignment of a graph g, representing an unknown pattern, to a graph language $L(G)$ generated by an $ETPL(k)$ graph grammar G is rejected by the $ETPL(k)$ type parsing. Therefore, there is a need for constructing effective parsing algorithms for recognition of distorted patterns. The purpose of this paper is to present a new approach to syntactic recognition of distorted patterns. To take into account all variations of a distorted pattern under study, a probabilistic description of the pattern is needed. A *random IE* graph approach is proposed here for such a description ([2]).

***Keywords*—**Syntactic pattern recognition, Distorted patterns, Random graphs, Graph grammars.

I. Introduction

THE so-called *IE* graphs have been defined in [1] for a description of patterns in syntactic pattern recognition. Nodes in an *IE* graph denote pattern primitives. Edges between two nodes in an *IE* graph represent spatial relations between pattern primitives. We do not present a definition of an *IE* graph here. Instead of it, we show an example pattern and its representation by an *IE* graph. In order to do it, let us assume that a set of edge labels describing spatial relations between primitives of an analyzed pattern is shown in Fig. 1. Now, let us assume that we analyze an example pattern shown in Fig. 2. The analyzed pattern shown in Fig. 2 can be represented by an *IE* graph shown in Fig. 3.

In practice structural descriptions may contain pattern distortions. For example, because of errors in the primitive extraction process, an *IE* graph g representing a pattern under study may be distorted, either in primitive properties or in their spatial relations, so that the assignment of the *IE* graph g to a graph language $L(G)$ generated by an $ETPL(k)$ graph grammar G is rejected by the $ETPL(k)$ type parsing. Therefore, there is a need for constructing effective parsing algorithms for recognition of distorted patterns, which is the motivation to do research.

Marek Skomorowski is with the Institute of Computer Science, Jagiellonian University, Kraków, Poland.

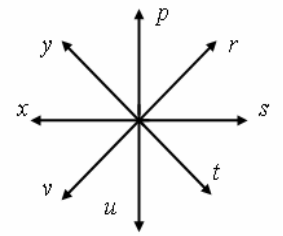


Fig. 1 A set of edge labels describing spatial relations between pattern primitives of an analyzed pattern

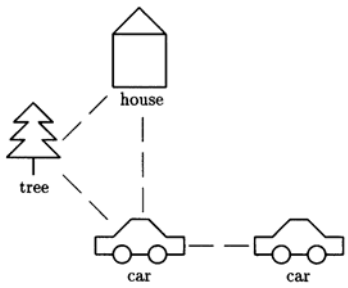


Fig. 2 An example pattern

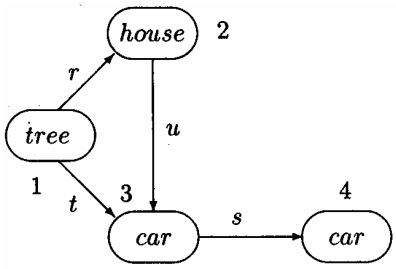


Fig. 3 An *IE* graph representing the example pattern shown in Fig. 2

The purpose of this paper is to present an idea of a new approach to syntactic recognition of distorted patterns. To take into account all variations of a distorted pattern under study, a probabilistic description of the pattern is needed. A *random IE* graph approach is proposed here for such a description ([2]). In this paper the idea of an efficient (that is the time complexity is $O(n^2)$) parsing algorithm presented in [1] is extended so that distorted patterns, represented by *random IE* graphs, can be recognized.

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/14780/pdf

II. Representation of Distorted Patterns

Let us assume that due to distortions possible (distorted) *IE* graphs associated with the example pattern shown in Fig. 2 might also be like *IE* graphs shown in Fig. 4.

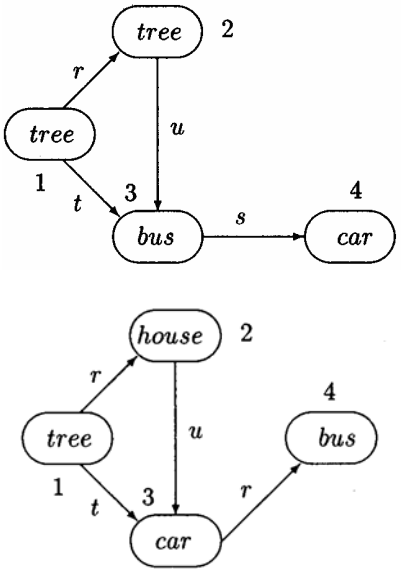


Fig. 4 Possible (distorted) *IE* graphs associated with the example pattern shown in Fig. 2

The *IE* graphs shown in Fig. 4 can be considered as outcome *IE* graphs obtained from a *random IE* graph ([2]) shown in Fig. 5.

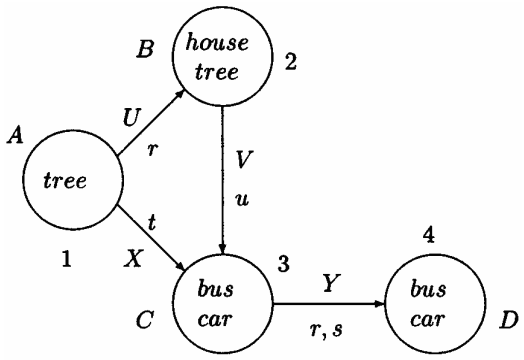


Fig. 5 A random *IE* graph

The *random IE* graph shown in Fig. 5 consists of random nodes {*A*, *B*, *C*, *D*} and random edges {*U*, *V*, *X*, *Y*}. The ranges of *A*, *B*, *C* and *D* are {*tree*}, {*house*, *tree*}, {*bus*, *car*} and {*bus*, *car*} respectively. The ranges of *U*, *V*, *X* and *Y* are {*r*}, {*u*}, {*t*}. and {*r*, *s*} respectively. Thus, in the proposed approach, a distorted pattern can be represented by a *random IE* graph. An example *random IE* graph with given nodes and edges distributions is shown in Fig. 6.

In the proposed approach it is assumed that random nodes and edges are mutually independent. Thus, the probability of an outcome *IE* graph, obtained form a *random IE* graph, is equal to the product of the probabilities of the node labels and the edge labels. For example, Fig. 7 shows an outcome IE graph r_1 obtained from the *random IE* graph shown in Fig. 6. Similarly, Fig. 8 shows an outcome *IE* graph r_2 obtained from the *random IE* graph shown in Fig. 6.

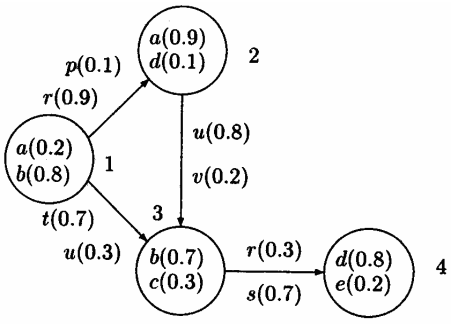


Fig. 6 An example *random IE* graph with given nodes and edges distributions

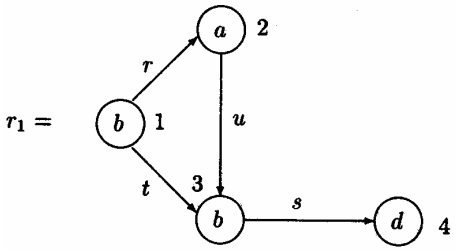


Fig. 7 An outcome *IE* graph r_1 obtained from the *random IE* graph shown in Fig. 6

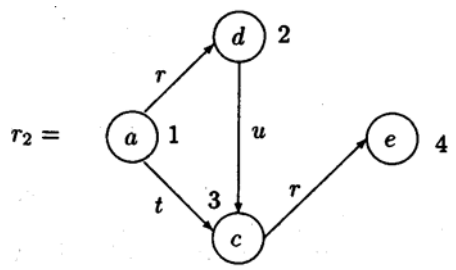


Fig. 8 An outcome *IE* graph r_2 obtained from the *random IE* graph shown in Fig. 6

The probabilities of the outcome *IE* graphs r_1 (Fig. 7) and r_2 (Fig. 8) are the following:

$p(r_1) = p(b_1)\ p(a_2)\ p(b_3)\ p(d_4)\ p(r_{1,2})\ p(t_{1,3})\ p(u_{2,3})\ p(s_{3,4}) =$
$= (0.8)\ (0.9)\ (0.7)\ (0.8)\ (0.9)\ (0.7)\ (0.8)\ (0.7) = 0.1422$

and

$p(r_2) = p(a_1)\ p(d_2)\ p(c_3)\ p(e_4)\ p(r_{1,2})\ p(t_{1,3})\ p(u_{2,3})\ p(r_{3,4}) =$
$= (0.2)\ (0.1)\ (0.3)\ (0.2)\ (0.9)\ (0.7)\ (0.8)\ (0.3) = 0.0002$

where indexes denote node numbers.

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/14780/pdf

III. Parsing of Random Graphs for Recognition of Distorted Patterns

Given an unknown distorted pattern represented by a *random IE* graph *R*, the problem of recognition of a pattern under study is to determine if an outcome *IE* graph *r*, obtained from the *random IE* graph *R*, belongs to a graph language $L(G)$ generated by an *ETPL*(k) graph grammar *G*.

It is generally impossible to explore all possible derivation paths during parsing because of combinatorial explosion. Therefore, we propose the following strategy of *random IE* graph parsing for an efficient (that is the time complexity is $O(n^2)$) analysis of distorted patterns: during a derivation step a number of simultaneously derived graphs spread through the search tree, but only the best graph, that is with maximum probability, is expanded.

Now, we show a derivation step of an *ETPL*(k) graph grammar *G* by means of the following example. Let us consider a graph shown in Fig. 9 (a) and a production shown in Fig. 9 (b). Suppose that the embedding transformation for the production shown in Fig. 9 (b) is the following:

$$C(r, input) = \{(d, b, r, input)\}$$

$$C(u, output) = \{(e, B, r, input)\}$$

The detailed discussion on the embedding transformation is presented in [1]. After the application of the production shown in Fig. 9 (b) to the node indexed by 2 of the graph shown in Fig. 9 (a) we obtain a graph shown in Fig. 9 (c).

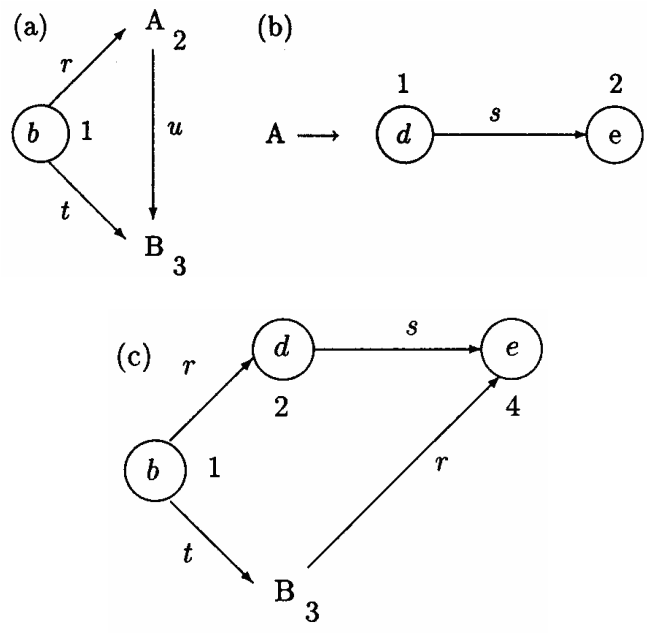


Fig. 9 An example derivation step of an *ETPL*(k) graph grammar

Parsing of *ETPL*(k) graph grammars simulates a left-hand derivations in these grammars ([1]). As it has been discussed in [1] and its time complexity $O(n^2)$ has been proved also in [1] we do not introduce it here formally. Instead of it we show, by means of an example, how a distorted pattern can be analyzed using left-hand side derivation performed with the *ETPL*(k) type parsing. In order to do it, suppose that we analyze an unknown distorted pattern represented by a *random IE* graph shown in Fig. 10.

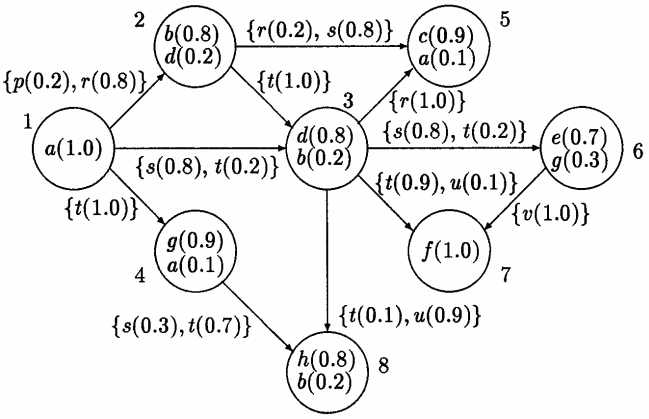


Fig. 10 An example *random IE* graph representing an unknown distorted pattern

Further, let us assume that we are given an *ETPL*(2) graph grammar *G* with a starting graph *Z* shown in Fig. 11 and a set of productions shown in Fig. 12.

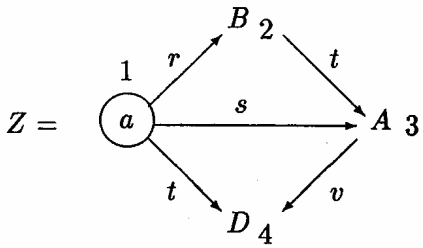


Fig. 11 A starting graph of an *ETPL*(2) graph grammar *G*

After the application of the production 1 (shown in Fig. 12) to the node indexed by 2 of the starting graph *Z* (shown in Fig. 11) we obtain a graph q_1 shown in Fig. 13. Similarly, after the application of the production 2 to the node indexed by 2 of the starting graph *Z* we obtain a graph q_2 shown in Fig. 14. Now we calculate the following probability of a subgraph of the graph q_1 :

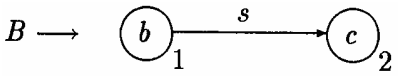
$$p(q_1) = p(a_1)\, p(b_2)\, p(c_5)\, p(r_{1,2})\, p(s_{2,5}) =$$
$$= (1.0)\ (0.8)\ (0.9)\ (0.8) = 0.4608$$

Similarly, we calculate the following probability of a subgraph of the graph q_2 :

$$p(q_2) = p(a_1)\, p(b_2)\, p(a_5)\, p(r_{1,2})\, p(s_{2,5}) =$$
$$= (1.0)\ (0.8)\ (0.1)\ (0.8)\ (0.8) = 0.0512$$

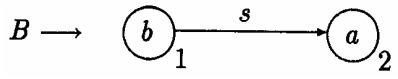
A production 1:

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/14780/pdf



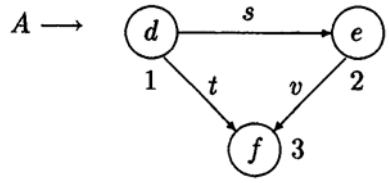
$C(r, input) = \{(b, a, r, input)\}$
$C(t, output) = \{(b, A, t, output), (c, A, r, input)\}$

A production 2:



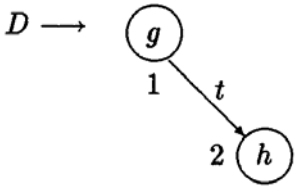
$C(r, input) = \{(b, a, r, input)\}$
$C(t, output) = \{(b, A, t, output), (a, A, r, input)\}$

A production 3:



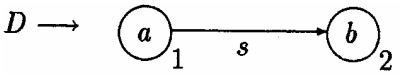
$C(s, input) = \{(d, a, s, input)\}$
$C(t, input) = \{(d, b, t, input)\}$
$C(r, output) = \{(d, c, r, output), (d, a, r, output)\}$
$C(v, output) = \{(d, D, v, output)\}$

A production 4:



$C(t, input) = \{(g, a, t, input)\}$
$C(v, input) = \{(h, d, u, input), (h, b, u, input)\}$

A production 5:



$C(t, input) = \{(a, a, t, input)\}$
$C(v, input) = \{(b, d, u, input), (b, b, u, input)\}$

Fig. 12 A set of productions of an *ETPL*(2) graph grammar G

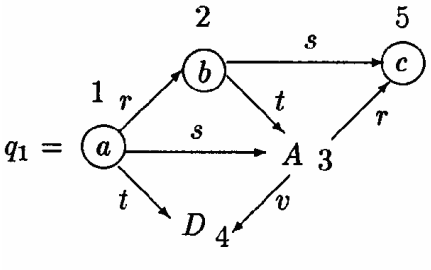


Fig. 13 A derived graph q_1

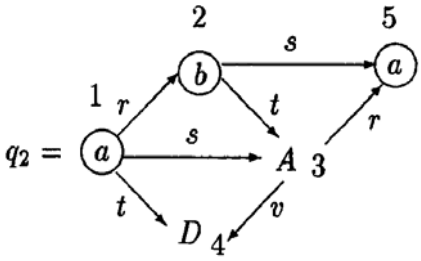


Fig. 14 A derived graph q_2

Because $p(q_1) > p(q_2)$ we choose the graph q_1 (Fig. 13) for further derivation. Now, after the application of the production 3 (Fig. 12) to the node indexed by 3 of the derived graph q_1 (Fig. 13) we obtain a graph shown in Fig. 15.

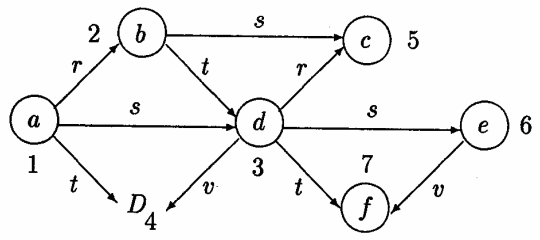


Fig. 15 A derived graph

Then, after the application of the production 4 (Fig. 12) to the node indexed by 4 of the derived graph shown in Fig. 15 we obtain a graph shown in Fig. 16.

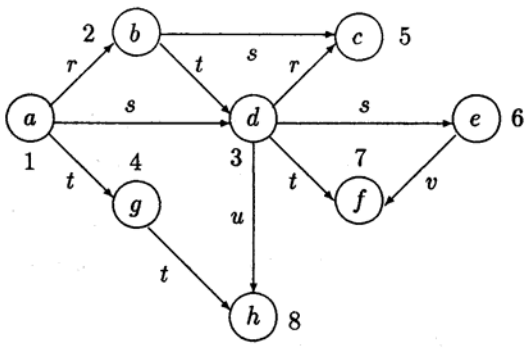


Fig. 16 A derived graph

Open Science Index, Computer and Information Engineering Vol:1, No:7, 2007 publications.waset.org/14780/pdf

The derived graph shown in Fig. 16 belongs to a language $L(G)$ generated by the example $ETPL(2)$ graph grammar G (Fig. 11 and Fig. 12). On the other hand the derived graph shown in Fig. 16 is simultaneously an outcome *IE* graph obtained from the parsed *random IE* graph shown in Fig. 10. Thus the parsed *random IE* graph shown in Fig. 10 is accepted. The path of search to the solution is shown in Fig. 17.

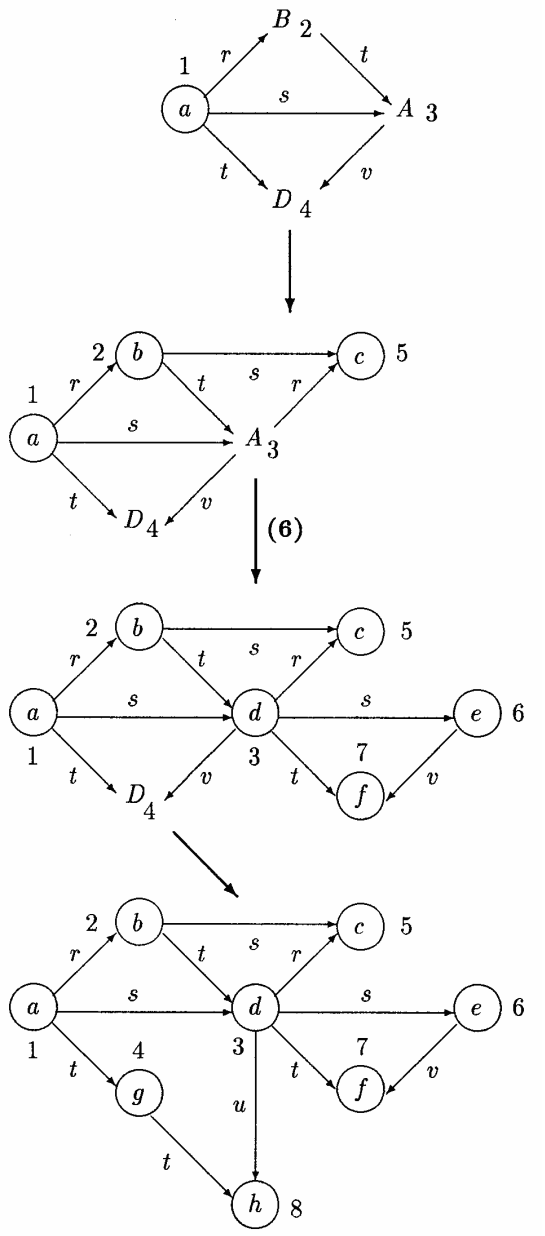


Fig. 17 The path of search to the solution

Similarly, after the application of the production 5 (Fig. 12) to the node indexed by 4 of the derived graph shown in Fig. 15 we obtain a graph shown in Fig. 18. The derived graph shown in Fig. 18 is simultaneously an outcome *IE* graph obtained from the parsed *random IE* graph shown in Fig. 10.

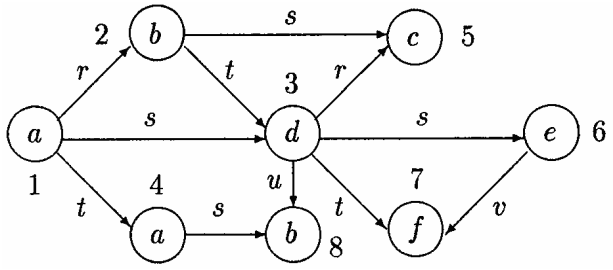


Fig. 18 A derived graph

IV. Conclusion

In this paper we have proposed a new approach to recognition of distorted patterns. To take into account all variations of a distorted pattern under study, a probabilistic description of the analyzed pattern is needed. The *random IE* graph ([2]) approach has been proposed here for such a description. In this paper the idea of an efficient (that is the time complexity is $O(n^2)$) parsing algorithm presented in [1] is extended, so that distorted patterns, represented by *random IE* graphs, can be recognized. The proposed approach involves $ETPL(k)$ type parsing ([1]) of *random IE* graphs.

A system for analysis of distorted patterns, represented by *random IE* graphs, based on the proposed here approach, has been implemented at the Institute of Computer Science, Jagiellonian University, Kraków, Poland ([3]). The time complexity $O(n^2)$ of the proposed $ETPL(k)$ type parsing of *random IE* graphs have been checked by several experiments. The results of a one of such experiments connected with measuring parsing time with a number of nodes of an analyzed *random IE* graph are shown in Fig. 19 ([3]).

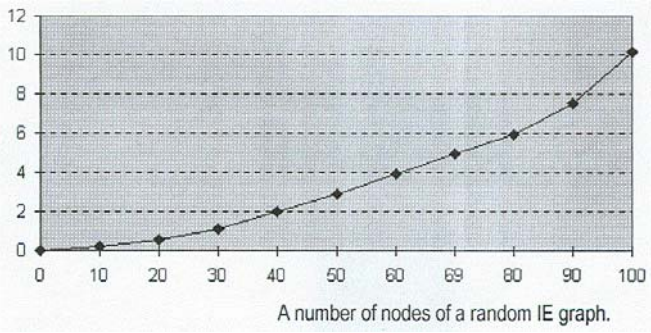


Fig. 19 Time of parsing (in seconds) with a number of nodes of an analyzed *random IE* graph

References

[1] M. Flasiński, On the parsing of deterministic graph languages for syntactic pattern recognition, *Pattern Recognition*, 26, 1993, pp. 1-16.

[2] M. Flasiński, M. Skomorowski, Parsing of random graph languages for automated inspection in statistical-based quality assurance systems, *Machine GRAPHICS & VISION International Journal*, 7, 1998, pp. 565-623.

[3] M. Uliasz, Application of graph grammars for syntactic pattern recognition of distorted patterns (in Polish), M. S. thesis, Institute of Computer Science, Jagiellonian University, Kraków, Poland, 1999.