Adam Sędziwy[*]

DISTORTED PATTERN RECOGNITION AND ANALYSIS WITH THE HELP OF IE$_f$ GRAPH REPRESENTATION

An algorithm for distorted pattern recognition is presented. It's generalization of M. Flasiński results (Pattern Recognition, 27, 1–16, 1992). A new formalism allows to make both qualitative and quantitive distortion analysis. It also enlarges parser flexibility by extending the set of patterns which may be recognized.
Keywords: *distorted pattern recognition, syntatic pattern recognition, ETPL$_f$ graph grammar, graph parsing*

ROZPOZNAWANIE I ANALIZA OBRAZÓW ROZMYTYCH REPREZENTOWANYCH PRZEZ IE$_f$ GRAFY
Praca zawiera algorytm syntaktycznego rozpoznawania obrazów rozmytych (zniekształconych), reprezentowanych przez IE$_f$ grafy. Jest on uogólnieniem algorytmu parsera dla gramatyk ETPL(k), podanego przez M. Flasińskiego dla obrazów zniekształconych. Zaproponowany formalizm pozwala na ilościową i jakościową analizę rozmycia badanego obiektu.
Słowa kluczowe: *rozpoznawanie obrazów rozmytych, syntaktyczne metody rozpoznawania obrazów, gramatyka ETPLS, parsing gramatyk grafowych*

1. Introduction

Graph grammars became an object of interest as a generalization of string grammars well known in the formal languages theory. They appeared to be a powerful formalism in various branches of applications: parsiηg theory, syntactic pattern recognition, parallel and concurrent systems, artificial intelligence, complers design, programming languages, CAD/CAM tools [5, 6, 7]. In that article we focus ourselves on syntactic pattern recognition methods.

The first step we have to make before we apply a syntactic graph model is the description of an analyzed phenomena in terms of graph formalism. Finding a proper graph representation is our basic task. The difference between theoretical graph model and real world situation is a common problem arising in the application of a graph methods to pattern recognition. This difference may be caused either by the distortions and fuzziness generated by registering devices (for example a camera) or by the nature of certÎ physical pheno-

* University of Mining and Metallurgy, Institute of Automatics, Cracow, e-mail: sedziwy@agh.edu.pl

mena (for example the shadows around an object). In all those cases we have to use some formalism taking into account possible pattern distortions. Such approach was already presented in [1]. However, a descriptive power of introduced grammar class was restricted by the finite number of elements in the set of possible distortions.

To resolve the above problem and make a recognizing algorithm more flexible we assume that the fuzziness may be parameterized, and that it's possible to define distortion function, measuring the fuzziness of analyzed object, in the parameter space. Such functions, associated with each of graph edge and node, play a key role during graph parsing, allowing to choose an appropriate production.

The pattern recognition process divides into two phases. The first (*syntactic*) phase gives the answer if analyzed graph (namely a graph representation of pattern) belongs to the language generated by our grammar. In this step we use a modified parsing algorithm for ETPL(k) grammars (see [2]). If the answer is yes, we can go to the second phase.

In the second (*semantic*) phase we check whether distortions do not exceed the limitations imposed on a problem. Those limitations may concern node and/or edges fuzziness. For example a total value of nodes fuzziness may not exceed a certain threshold. Such an analysis is possible since we assume that distortions can be parameterized.

2. Preliminaries

2.1. An indexed edge-unambiguous graph (IE graph)

We begin our considerations with the definition of IE graphs. This family of graphs was introduced in [4] for an ambiguous scene representation.

An IE graph is a quintuple $G = (V, E, \Sigma, \Gamma, \phi)$, where:

V – the finite, nonempty set of nodes to which indices have been ascribed in an unambiguous way;

$\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_n\}$ – the finite, nonempty set of edge labels ordered by the relation of simple ordering $\leq$; the elements of Γ can be interpreted as the spatial relations between the pattern objects, represented by the nodes of G; Γ is assumed to be a family of non-symmetric binary relations: for each label $\lambda \in \Gamma$ there exists label λ^{-1} such that the edges connecting nodes u and v, (u, λ, v) and (v, λ^{-1}, u), describe the same spatial relation (so called *semantically equivalent edges*);

Σ – the finite, nonempty set of node labels,

E – the set of graph edges of the form $(v, \lambda, w) \in V \times \Gamma \times V$, fulfilling conditions:

- each the edge is directed from the node having a smaller index to the node having a greater one,
- for each $v \in V$: if $(v, \lambda, \omega) \in E$ then there doesn't exist $(v, \lambda, z) \in$ E or $(z, \lambda^{-1}, v) \in E$,

$\phi: V \rightarrow \Sigma$ – the node labelling function.

2.2. Distortion functions

Let $G = (V, E, \Sigma, \Gamma, \phi)$ be an IE graph. We make the generalization concerning the sets of node and edge labels Σ, Γ:

- $\Sigma \subset \Sigma_u$, $\Gamma \subset \Gamma_u$, where Σ_u, Γ_u may be finite or infinite (continuous) sets;
- with each element $x \in \Sigma$ we associate the function $\mu_x^\Sigma: \Sigma_u \to R_+ \cup \{0\}$.

Similarly with each $\gamma \in \Gamma$ we associate $\mu_\gamma^\Gamma: \Gamma_u \to R_+ \cup \{0\}$.

We impose following conditions on μ_x^Σ, μ_γ^Γ:

$$\mu_x^\Sigma(x) = \sup_{\tilde{x} \in \Sigma_u} \mu_x^\Sigma(\tilde{x}),$$

$$\mu_\gamma^\Gamma(\gamma) = \sup_{\tilde{\gamma} \in \Gamma_u} \mu_\gamma^\Gamma(\tilde{\gamma}).$$

Values $\mu_\gamma^\Gamma(\tilde{\gamma})$, $\mu_x^\Sigma(\tilde{x})$ measures the degree of similarity between $\tilde{\gamma}$, γ and $\tilde{x}$, x respectively. Functions μ_x^Σ, μ_γ^Γ are called distortion functions.

Example 1

The set of labelled edges is shown in Figure 1. The labels in $\Gamma = \{p, r, s, t, u, v, x, y\}$, can be viewed as the values of an angular coordinate (analogously to the compass quarters: N, NE, E, SE and so on): $p = 0$, $r = \frac{\pi}{4}$, ..., $y = \frac{7}{4}\pi$.

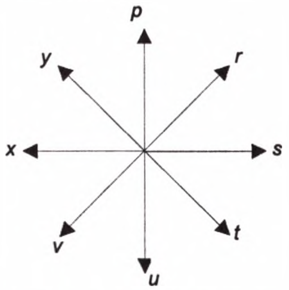

Fig. 1. The set of edge labels Γ

Assume the fuzziness of directions in the range $\pm\Delta\alpha$, where $\Delta\alpha = \frac{\pi}{12}$ and associate the functions μ_γ^Γ with them:

$$\mu_p^\Gamma(\alpha) = \begin{cases} \exp(-k(2\pi - \alpha)^2), & \alpha \in [2\pi - \Delta\alpha, 2\pi), \\ \exp(-k\alpha^2), & \alpha \in [0, \Delta\alpha], \\ 0, & \text{otherwise}, \end{cases}$$

$$\mu_{\gamma_j}{}^\Gamma(\alpha) = \begin{cases} \exp\left(-k\left(\frac{\pi j}{4} - \alpha\right)^2\right), & \alpha \in \left[\frac{\pi j}{4} - \Delta\alpha, \frac{\pi j}{4} + \Delta\alpha\right], \\ 0, & \text{otherwise}, \end{cases}$$

where $k > 0$ is constant, $\gamma_j = r, s, ..., y$, for $j = 1, ..., 7$ (see Fig. 2). At the horizontal axis we mark the angle values as described above.

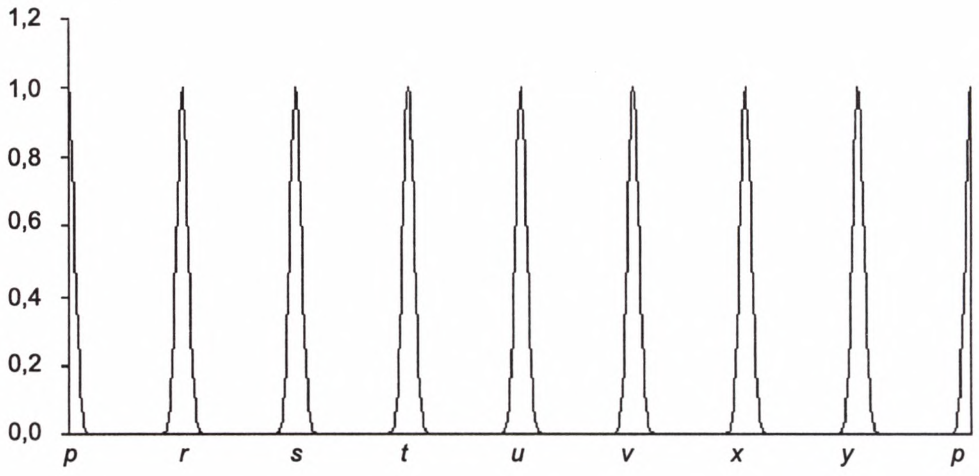

Fig. 2. Functions μ_γ^Γ for the edge labels

2.3. Acceptable distortions of the nodes and edges

Define supports of μ_x^Σ and μ_γ^Γ as the sets:

$$\operatorname{supp}(\mu_x^\Sigma) = \{\,\tilde{x} \in \Sigma_u\colon \mu_x^\Sigma(\tilde{x}) > 0\},$$

$$\operatorname{supp}(\mu_\gamma^\Gamma) = \{\tilde{\gamma} \in \Gamma_u\colon \mu_\gamma^\Gamma(\tilde{\gamma}) > 0\}.$$

One should interpret supports as the sets of labels with it *non-zero* similarity to the not distorted ones.

Define the set of acceptable object distortions (*SAOD*) and the set of acceptable spatial relation distortions (*SASRD*) by:

$$SAOD(x) \overset{\text{def}}{=} \operatorname{supp}(\mu_x^\Sigma),\ x \in \Sigma,$$

$$SASRD(\gamma) \overset{\text{def}}{=} \operatorname{supp}(\mu_\gamma^\Gamma),\ \gamma \in \Gamma.$$

Remarks

(i) *SAOD* and *SASRD* are in general the continuous sets (see conditions imposed on Γ_u, Σ_u).

(ii) We will assume in the sequel that for a distinct arguments the corresponding sets are disjoint i.e:

$$\forall x_1, x_2 \in \Sigma,\ x_1 \neq x_2\colon SAOD(x_1) \cap SAOD(x_2) = \varnothing,$$

$$\forall \lambda_1, \lambda_2 \in \Gamma,\ \lambda_1 \neq \lambda_2\colon SASRD(\lambda_1) \cap SASRD(\lambda_2) = \varnothing.$$

(iii) For nonterminal nodes we define: $SAOD(x) = \{x\}$.

2.4. Fuzzy labels sets

Using the notion of *SAOD* and *SASRD*, associate with Σ, Γ the sets (the fuzzy extensions of Σ and Γ):

$$\Sigma_f \overset{\text{def}}{=} \bigcup_{x \in \Sigma} SAOD(x),$$

$$\Gamma_f \overset{\text{def}}{=} \bigcup_{\gamma \in \Gamma} SAOD(\gamma).$$

Remarks

By remark (ii) in the previous point 2.3 we assume that $\forall \tilde{x} \in \Sigma_f$ ($\tilde{\gamma} \in \Gamma_f$) there exists and can be effectively pointed out only one $x \in \Sigma(\gamma \in \Gamma)$ such that $\tilde{x} \in SAOD(x)$, ($\tilde{\gamma} \in SASRD(\gamma)$).

2.5. The distance between node/edge labels

In Σ_f, Γ_f define distance functions:

$$\delta^{\Sigma}(x_1, x_2) \overset{\text{def}}{=} \begin{cases} |\mu_x^{\Sigma}(x_1) - \mu_x^{\Sigma}(x_2)|, & x_1, x_2 \in SAOD(x), \text{ for some } x, \\ \infty, & \text{otherwise,} \end{cases}$$

$$\delta^{\Gamma}(\lambda_1, \lambda_2) \overset{\text{def}}{=} \begin{cases} |\mu_{\lambda}^{\Sigma}(x_1) - \mu_{\lambda}^{\Sigma}(x_2)|, & \lambda_1, \lambda_2 \in SAOD(x), \text{ for some } \lambda, \\ \infty, & \text{otherwise,} \end{cases}$$

Note that we will say about *distance between nodes/edges* remembering that it concerns their labels.

Remark

We make two assumption about ∞:

(i) $\infty + \infty = \infty$,
(ii) $\infty + c = \infty$.

2.6. A node of *n*-th level

A node v_0 of IE graph $G = (V, E, S, \Gamma, \phi)$ having an index 1 is called a node of the first level. We introduce recursively a notion of n-th level node. A node v is a node of the n level if:

(i) there exists such an edge $(w, \gamma, v) \in E$ that w is a node of the $n - 1$ level,
(ii) for each $[(u, \gamma, v) \in E$ or $(v, \gamma, u) \in E]$: u is a node at least $n - 1$ level.

We define in the same way, a node of n-th level for an ief graph, introduced in the following section.

3. Graph structures

3.1. IE$_f$ Graphs

Since we introduced the node and edge labels fuzziness and constructed the sets of distorted labels, it is possible to define an extension of IE graphs family. These new, distorted graphs, called IE$_f$ graphs, will be helpful for the representation of a scene containing deformed objects.

Let $G = (V, E, \Sigma, \Gamma, \psi)$ be given an IE graph. We construct new "fuzzy" graph $G_f = (V, E_f, \Sigma_f, \Gamma_f, \phi)$ called an IE_f[1] graph (an indexed edge-unambiguous fuzzy graph), where:

V – a finite, nonempty set of nodes to which indices have been ascribed in an unambiguous way;

E_f – a set of graph edges of the form $(w, \lambda, v) \in V \times \Gamma_f \times V$, satisfying conditions:
  (i) each edge is directed from the node having a smaller index to the node having a greater one,
  (ii) for each $v \in V$: if $(v, \lambda, w) \in E_f$ then there doesn't exist $(v, \lambda_1, z) \in E_f$ such that $\delta^{\Gamma}(\lambda_1, \lambda_2) \neq \infty$ or $(z, \gamma, v) \in E_f$ such $\delta^{\Gamma}(\lambda, \gamma^{-1}) \neq \infty$;

Σ_f and Γ_f – the fuzzy extensions of Σ, Γ, defined previously; we assume that Γ_f preserves all the properties of Γ: it's ordered by the certain relation of simple ordering $\leq$, Γ_f is a set of non-symmetric binary relations and for each label $\lambda \in \Gamma_f$ there exists label $\lambda^{-1} \in \Gamma_f$ such that the edges (u, λ, v) and (v, λ^{-1}, u) describe the same spatial relation between the nodes u and v;

$\phi: V \to \Sigma_f$ – a node labelling function such that $\phi|_G = \psi$.

Remark

We assume that there exist well defined distortion functions for the edges and nodes in all the definitions where the sets Σ_f, Γ_f are present.

3.2. Distance between two IE_f graphs

Now we define a measure of distortion, which tells us how much distant are the deformed graphs G and H. Let:

$$G = (V_G, E_{f,G}, \Sigma_f, \Gamma_f, \phi_G),$$

$$H = (V_H, E_f, H, \Sigma_f, \Gamma_f, \phi_H)$$

be isomorphic[2] IE_f graphs. (We can assume that $V_G = V_H = V$). A distance between G and H is defined as below

$$\delta(G, H) = \overbrace{\sum_{(x,\tilde{x})} \delta^{\Sigma}(x, \tilde{x})}^{C_n} + \overbrace{\sum_{(\gamma,\tilde{\gamma})} \delta^{\Gamma}(\gamma, \tilde{\gamma})}^{C_e},$$

where x is the label of a node belonging to G and indexed with i, analogously $\tilde{x}$ is the label of a node belonging to H and indexed with the same index i, γ and $\tilde{\gamma}$ are the labels of corresponding edges belonging to G and H respectively, coming from the nodes indexed with j to the nodes indexed with k. The terms denoted as C_n and C_e are called respectively node cost and edge cost.

---

[1] In the sequel a subscript f refers to the word *fuzzy*.

[2] We appy the definition of IE graph isomorphis (see [2], p. 5) to IE_f graphs: two IE_f graphs $A = (V_A, E_{f,A}, \Sigma_f, \Gamma_f, \phi_A)$ and $B = (V_B, E_{f,B}, \Sigma_f, \Gamma_f, \phi_B)$ are isomorphic if there exists bijection $h: V_A \to V_B$ such that (i) $\phi_B \circ h = \phi_A$, (ii) $E_{f,B} = \{(h(x), \lambda, h(y)) : (x, \lambda, y) \in E_{f,A}\}$.

3.3. Characteristic description

Let $G = (V, E_f, \Sigma_f, \Gamma_f, \phi)$ be an IE_f graph. A characteristic description of the node $n_k \in V$, having an index k and labelled with n, is a sequence of the form $I_k = [n_k, r, e_{i_1}, ..., e_{i_r}]$, where $i_1 < i_2 < ... < i_r$, r is the number of edges going out from this node e_{i_j}, $(j = 1, ..., r)$[3] footnotemark denotes a label of the edge directed from the node n_k to the node having an index i_j, i_j is attached as a subscript to that label.

A characteristic description of graph G is a sequence $I = [I_1; I_2; ...; I_m]$ of characteristic descriptions of its nodes.

Example 2
The characteristic description of the node, say v, indexed with 1 (Fig. 3) is

$$I_1 = [a_1, 2, t_2u_3].$$

The subscript "1", of the node label a indicates just an index of v; the subscripts "2" and "3" are the indices of the nodes to which the edges labelled with t and u are directed.

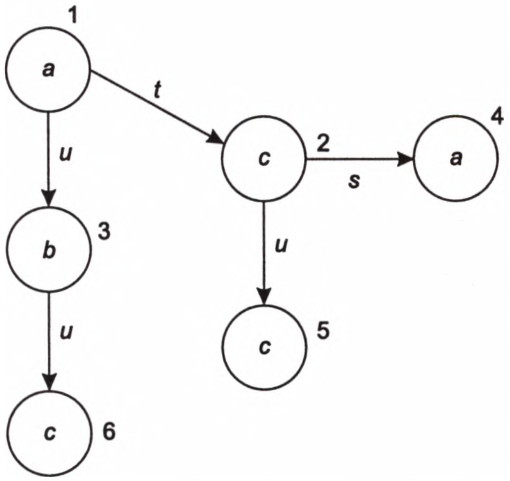

Fig. 3. Characteristic description of a node

The description for entire graph is

$$I = [a_1, 2, t_2u_3; c_2, 2, s_4u_5; b_3, 1, u_6; a_4, 0, -; c_5, 0, -; c_6, 0, -].$$

The comparison of the characteristic descriptions of two corresponding nodes of derived and analyzed graph, plays a key role in the parsing algorithm. In definitions of the parser procedures following notions will be helpful.

[3] In several cases we will denote the string $e_{i_1} ... e_{i_r}$ as E_k, where k is an index of the node n_k.

3.4. Context-identical nodes

Let n_k, $\bar{n}_k$ be the nodes having the same index and labelled with n and $\bar{n}$ respectively.

(i) Let $G = (V, E_f, \Sigma_f, \Gamma_f, \phi)$ and $\bar{G} = (\bar{V}, \bar{E}_f, \Sigma_f, \Gamma_f, \phi)$ be the IE$_f$ graphs. Nodes $n_k \in V$ and $\bar{n}_k \in \bar{V}$ are *context-identical* if they have the same characteristic descriptions.

(ii) Let's assume that $\# V > \# \bar{V}$. Nodes n_k, $\bar{n}_k$ having the characteristic descriptions $I_k = [n_k, r, e_{i_1} \dots e_{i_r}]$, $\bar{I}_k = [\bar{n}_k, \bar{r}, d_{\bar{i}_1} \dots d_{\bar{i}_{\bar{r}}}]$ are potentially *context-identical* if following conditions are satisfied.

1. $\delta^{\Sigma}(n, \bar{n}) \neq \infty$.
2. $r = \bar{r}$.
3. If for each $j \in \{1, \dots, r\}$ there exists $k \in \{1, \dots, r\}$, such that $i_j = i_k$, then $\delta^{\Gamma}(e_{i_j}, d_{\bar{i}_k}) \neq \infty$.
4. Let $A_s = (i_1' \dots i_p')$ and $B_s = (\bar{i}_1' \dots \bar{i}_p')$ be the subsequences of $A = (i_1 \dots i_r)$ and $B = (\bar{i}_1 \dots i_r)$ correspondly. Moreover A_s and B_s are assumed to have following two properties:
   (i) sequence B_s does not contain any member of A,
   (ii) sequence A_s does not contain any member of B.

Nodes having indices from A_s are terminal and those having indices belonging to B_s are nonterminal ones.

3.5. Potentially contextual nodes

The nodes having the indices i_1', ..., i_p' (i.e. belonging to A_s) are called textitpotentially contextual for the node indexed with k.

The edges $(v_k, e_{i'_1}, v_{i'_1})$, ..., $(v_k, e_{i'_p}, v_{i'_p})$, where v_q denotes a node indexed with q, are called *potentially contextual* edges for the node indexed with k.

Let (v_k, e, v_q) be a potentially contextual edge. The pair of the form (k, e) is called a description of a potentially contextual identity. We ascribe to index q a list of such descriptions: L_q (see the notions in a section 5.1).

Above definitions are introduced to overcome following difficulty. During the parsing process we compare characteristic descriptions of two corresponding nodes belonging to generated and analyzed graph. It may happen that in a moment of comparison not all nodes adjacent (i.e. giving a contribution to the characteristic description) to the considered node are generated yet. In such a case we have to store this fact in a list L_q and check the description later (for a detailed discussion concerning this problem see [2], p. 10 and [1] p. 770.)

Example 3.

In Figure 4 the parts of graphs bar $\bar{G}$ (generated graph) and G (analyzed graph) are shown.

Keeping the notation introduced above we have:

$$\bar{I}_3 = [d_3, 3, u_4 r_5 s_6],\ I_3 = [d_3, 3, r_5 s_6 t_7];$$

$$n = \bar{n} = d;$$

$$k = 3,\ r = \bar{r} = 3;$$

$$A = (i_1 i_2 i_3) = (5, 6, 7),\ \mathrm{B} = (i_1 i_2 i_3) = (4, 5, 6);$$

$$A_s = (i_1') = (7),\ B_s = (i_1') = (4).$$

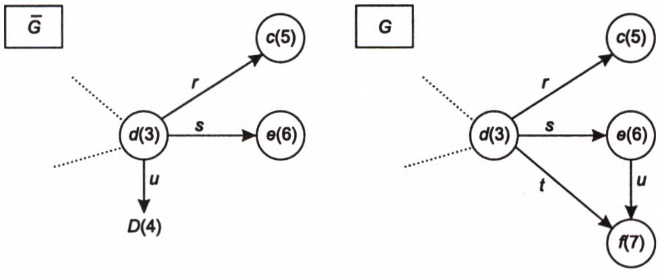

Fig. 4. Potentially contextual mode

The node f_7 is potentially contextual for the node d_3. Since the edge (d_3, t, f_7) is potentially contextual for d_3 we have: $L_7 = \{(3, t)\}$. After a node f_7 is generated we have to verify if the edge $(d_3, \tilde{t}, f_7)$ (where $\delta^{\Gamma}(\tilde{t}, t) \neq \infty$) in graph bar $\overline{G}$ is present (see procedure CHECK).

3.6. Subgraph *m*-TL(*G*, *l*, *j*)

Let G be an IE_f graph and let l be an index of its node n_l having characteristic description $I_l = [n_l, r, e_{i1} \dots e_{ir}]$. A subgraph H of G consisting of node n_l, nodes indexed with $i_{a+1}, \dots, i_{a+m}$, $(a \geq 0, a + m \leq r)$ and edges connecting those nodes is denoted: $H = m - \text{TL}(G, l, i_{a+1})$. In a special case when $m = r - a$ we denote: $H = \text{CTL}(G, l, i_{a+1})$.

4. Graph grammars

4.1. TLP$_f$ grammar

A quintuple $g = (\Sigma_f, \Delta, \Gamma_f, P, Z)$ is called a TLP$_f$ graph grammar (a two-level productions fuzzy graph grammar), if following conditions are fulfilled:

1. $\Delta \subset \Sigma_f$ – set of terminal labels.
2. P – finite set of production of the form (l, D, C_f), where: l – label of the nonterminal node – left hand side of production, $D \in \text{IE}_f$ – right hand side of production having characteristic description which satisfies condition:

$$I = [n_1, r_1, E_1; \dots; n_m, r_m, E_m] \text{ or}$$
$$I = [n_1, 0, -], \qquad \textbf{(W)}$$

   where: $n_1 \in \Delta$, $n_2, \dots, n_m$ – the nodes of 2nd level;
   $C_f: \Gamma_f \times \{in, out\} \rightarrow \Sigma_f \times \Sigma_f \times \Gamma_f \times \{in, out\}$ – embedding transformation.
3. $Z \in \text{IE}_f$ – initial graph (the axiom), having characteristic description satisfying condition **(W)**.

To explain the way how embedding transformation should be applied let's consider expression $C_f(\gamma, in) = (x, y, \lambda, out)$. Each edge labelled with $\tilde{\gamma} \in SASRD(\gamma)$ coming into the node of lhs of production should be replaced by the edge labelled with $\tilde{\lambda} \in SASRD(\lambda)$ directed from the the node having the label $\tilde{x} \in SAOD(x)$ of production rhs to the node labelled with $\tilde{y} \in SAOD(y)$ in the rest-graph (i.e. graph with the production lhs node removed).

$\mathcal{G}$ is called a closed TLP$_f$ graph grammar if for any derivation

$$Z = G_0 \to G_1 \to G_2 \to \dots G_n$$

of this grammar G_i, $(i = 0, \dots, n)$ is an IE$_f$ graph.

Example 5

Let's consider embedding transformation for the production $P : B \to H = [b_1, 1, s_2; \tilde{c}_2, 0, -]$ (Fig. 5):

$$C_f(r, in) = \{(b, a, r, in)\},$$

$$C_f(u, out) = \{(b, A, u, out), (c, A, r, in)\}.$$

In this case the edges r, u remain unchanged, and new edge r going out from nonterminal node A to $\tilde{c}$ is created.

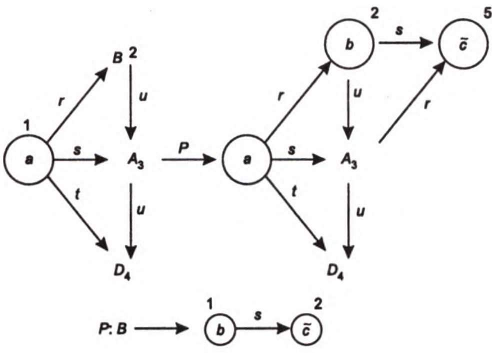

Fig. 5. The production of the TLP$_f$ grammar

Example 6

We may define following example of TLP$_f$ graph grammar (Fig. 6) and make derivation using its productions (Fig. 7).

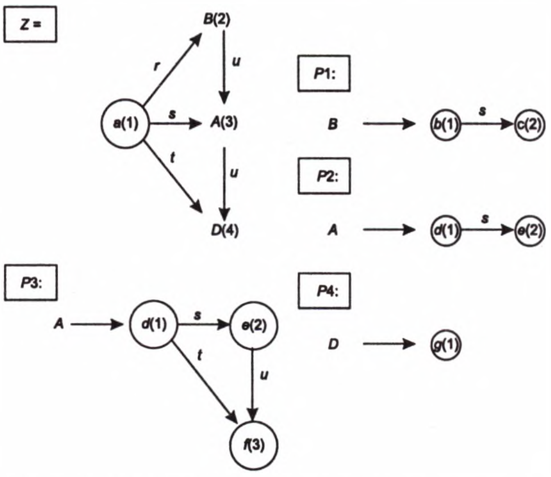

Fig. 6. Initial graph Z and productions of grammar $\mathcal{G}$

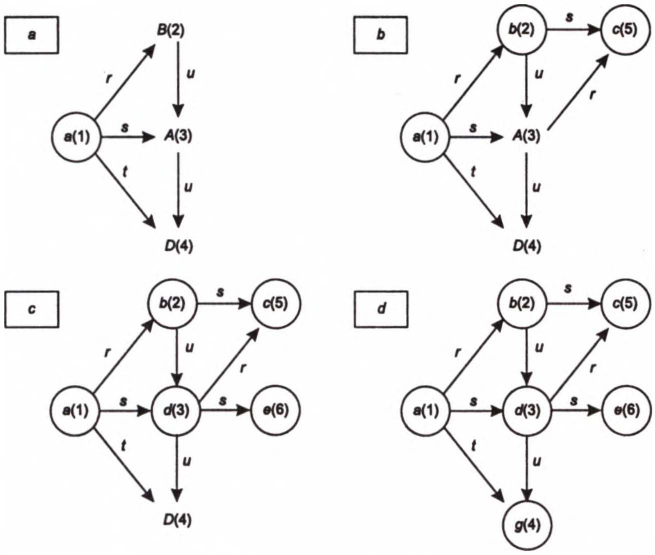

Fig. 7. Derivation in the grammar $\mathcal{G}$

$$\mathcal{G} = (\Sigma_f, \Delta, \Gamma_f, P, Z),$$

where:

$$\Sigma = \{a, b, c, d, e, f, g, A, B, D\},\ \Sigma_f = \bigcup_{x\in\Sigma} SAOD(x),$$

$$\Delta = \{a, b, c, d, e, f, g\},$$

$$\Gamma = \{p, r, s, t, u, v, x, y\},\ \ \Gamma_f = \bigcup_{\gamma\in\Gamma} SASRD(\gamma).$$

Remark

We assume the existence of μ^{Σ} and μ^{Γ} functions for all the elements of Σ and Γ sets.

Embedding transformations for the grammar productions (a lower index includes the number of production to which the transformation is ascribed).

1) $C_{f,1}$:

$$C_f(r, in) = \{b, a, r, in)\},$$

$$C_f(u, out) = \{(b, A, u, out), (c, A, r, in)\};$$

2) $C_{f,2}$:

$$C_f(s, in) = \{(d, a, s, in)\},$$
$$C_f(u, in) = \{(d, b, u, in)\},$$
$$C_f(r, out) = \{(d, c, r, out)\},$$
$$C_f(u, out) = \{(d, D, u, out)\};$$

3) $C_{f,3} = C_{f,2}$;

4) $C_{f,4}$:

$$C_f(t, in) = \{(g, a, t, in)\},$$

$$C_f(u, in) = \{(g, d, u, in)\}.$$

4.2. PL$_f(k)$ graph grammar

Let $G = (\Sigma_f, \Delta, \Gamma_f, P, Z)$ be a closed TLP$_f$ graph grammar. Let

$$Z \underset{r(\mathcal{G})}{\overset{*}{\to}} X_1AX_2 \underset{r(\mathcal{G})}{\to} G_1 \underset{r(\mathcal{G})}{\overset{*}{\to}} H_1$$

and

$$Z \underset{r(\mathcal{G})}{\overset{*}{\to}} X_1AX_2 \underset{r(\mathcal{G})}{\to} G_2 \underset{r(\mathcal{G})}{\overset{*}{\to}} H_2,$$

be regular left-hand side derivations of grammar $\mathcal{G}$, where $\to^*_{r(\mathcal{G})}$ denotes transitive and reflexive closure of $\to_{r(\mathcal{G})}$. A is a characteristic description of the node indexed with l. X_1, X_2 are substrings, max – a number of nodes of the graph X_1AX_2.

$\mathcal{G}$ is called a PL$_f$ (k) graph grammar (a production-ordered k-left nodes unambiguous fuzzy graph grammar) if the following condition is fulfilled.

If $k - \mathrm{TL}(H_1, l, max + 1) \cong k - \mathrm{TL}(H_2, l, max+1)$,

then $\mathrm{CTL}(G_1, l, max + 1) \cong \mathrm{CTL}\ (G_2, l, max + 1)$ **(PL)**

The symbol $\cong$ denotes the isomorfsm of graphs.

4.3. Potential previous context

Let $\mathcal{G} = (\Sigma_f, \Delta, \Gamma_f, P, Z)$ be PL$_f(k)$ grammar. A pair $(y, \lambda) \in \Delta \times \Gamma_f$ is called a potential previous context for the node label $x \in \Sigma$, if there exists such IE$_f$ graph $G = (V, E_f, \Sigma_f, \Gamma_f, \phi)$ belonging to some regular left-hand side derivation in $\mathcal{G}$, that $(u, \lambda, v) \in E$, $\phi(u) = \widetilde{y}$, $\phi(v) = = \widetilde{x}$, $u, v \in V$ and $\delta^{\Gamma}(\widetilde{\lambda}, \lambda) \neq \infty$, $\delta^{\Sigma}(\widetilde{x}, x) \neq \infty$, $\delta^{\Sigma}(\widetilde{y}, y) \neq \infty$.

4.4. ETPL$_f(k)$ graph grammar

$\mathcal{G} \in$ PL$_f(k)$ is called an ETPL$_f(k)$ graph grammar (embedding transformation-preserved production-ordered k-left nodes unambiguous fuzzy graph grammar), if each production

$$A \to [X_1, r_1, E_1; \ldots; X_m, r_m, E_m]$$

belonging to that grammar satisfies condition: if (b, y) is a potential previous context for A, then there exists only one quadruple $(X_i, b, z, in) \in C_f(y, in)$, where C_f is an embedding transformation for this production; in case of $i = 1$, we have $z = y$ and $(X_1, b, y, in) \in C_f(y, in)$.

4.5. TTLP and TTLN sets

We define two families of sets associated with the grammar productions.

(i) Let $A \to G$ be the l-th production of the grammar $\mathcal{G} \in \text{ETPL}_f(k)$. Let's define a set of terminal graphs generated by this production

$$\text{TTLP}(l) = \{\hat{G} \in \text{IE}_f\text{: 1. } \hat{G} \subset \text{H} \quad \text{where} \quad A \underset{r(G)}{\overset{*}{\to}} H, \text{2. } \hat{G} = k - \text{TL}(H, 1, 2)\}.$$

(ii) Let (l_1) A $\to G_1, ..., (l_n)\ A \to G_n$ be all productions of a grammar $\mathcal{G} \in \text{ETPL}_f(k)$ having nonterminal symbol A in their lhs. A set of all terminal graphs generated from A has a form: $\text{TTLN}(A) = \{(l_i, \text{TTLP}(l_i)), i = 1, ..., n\}$.

5. Parser

5.1. Parsing procedures and functions

Notations:

G	– an analyzed graph
H	– a derived graph
Z	– an axiom graph (an initial graph)
L_i	– a list of descriptions of potentially contextual identities, ascribed to the node indexed with i
$\phi_H(i)$	– a label of the node of graph H, having an index i
n	– a number of nodes of graph G
c_n	– a node cost
c_e	– a edge cost
S	– a sequence of triples: (number of production, c_n, c_e)
v_i	– a node indexed with i

Procedures and functions

Remark

Contrary to [1] we will compute separately the edge cost (c_e) and node cost (c_n). It will allow us to make also a qualitative description of pattern distortion.

MAXIND(H) returns a number of nodes of a graph H.

DEFk-TL(G, i, m, E) creates a graph E =k-TL(G, i, m).

GIVETTLN(A) returns a set TTLN(A).

CHOOSEPROD(E, R, k)

Step 1. Chooses such graphs $H_1, ..., H_p$ from R that $\exists\ E_i$ – subgraph of E: $E_i \cong H_i$, and $\delta(E_i, H_i) \neq \infty$ for $i = 1, ..., p$.

Step 2. Chooses from the set $H_1, ..., H_p$ a subset $H_{i1}, ..., H_{iq}$ of best fitting subgraphs

$$\delta(H_{ij}, E_{ij}) = \min_{1 \leq k \leq p} \delta(H_k, E_k), j = 1, ..., q.$$

Step 3. Chooses the maximal graph H_m from the set $H_{i1}, ..., H_{iq}$. If $H_m \neq \varnothing$ then k := number of production giving H_m, otherwise k := 0.

PRODUCTION(H, i, k)

Step 1. The application of the k-th production for the node v_i of graph H.

Step 2. The calculation of the costs:

$$c_n = \sum_{(\tilde{x}, x)} \delta^{\Sigma}((\tilde{x}, x), \quad c_e = \sum_{(\lambda, \tilde{\lambda})} \delta^{\Gamma}(\lambda, \tilde{\lambda}),$$

where the sums extend over all terminal nodes generated in result of the current production application (for c_n) and over all new terminal edges[4] (for c_e); it's assumed that $x \in \phi_H(V_H)$ is a corresponding node label for $x \in \phi_G(V_G)$ and analogously: $\tilde{\lambda}$ and λ are the labels of two corresponding edges.

Step 3. $S := S \cup (k, c_n, c_e)$.

CONID(G, H, i)

The Boolean function testing if nodes of graphs G and H having index i are context-identical.

PCONID(G, H, i, err)

If the nodes of graphs G and H indexed with i are potentially context-identical then for each potentially contextual edge (v_i, e, v_q): $L_q := L_q \cup (i, e)$. Otherwise err := 2.

CHECK(L_i, H, err)

```
for each $(k, e) \in L_i$:
  if $\exists\tilde{e} \in SASRD(e)$ such that $(v_k, \tilde{e}, v_i) \in H$ then
    $L_i := L_i - (k, e)$
  else err := 3
```

5.2. Parsing algorithm

```
$H := Z$; err := 0; $S := (0, c_{v0}, c_{e0})$[5];
for $i = 0$ to n do
  if err=0 then
  begin
    if $\phi_H(i)$ is a nonterminal node then
    begin
      m := MAXIND($H$) + 1
      DEFk-TL($G$, $i$, $m$, $E$)
      R := GIVETTLN($\phi_H(i)$)
      CHOOSEPROD($E$, $R$, $k$)
      if $k$=0 then err := 1
      else PRODUCTION($H$, $i$, $k$)
    end
    if not CONID($G$, $H$, $i$) then PCONID($G$, $H$, $i$, $err$)
    CHECK($L_i$, $H$, err)
  end
```

[4] A terminal edge is an edge between two terminal nodes.

[5] The values c_{n0},c_{e0} are the sums of distances between corresponding terminal nodes and terminal edges of an axiom Z and analyzed graph G.

Example 7

Let us modify grammar $\mathcal{G}$ from the last example by replacing production P_3 with two new productions: P_{3a} and P_{3b} (Fig. 8a). The embedding transformation remains unchanged: $C_{f,3a} = C_{f,3b} = C_{f,3}$. Analyzed graph G (Fig. 8b) is an input for the parser.

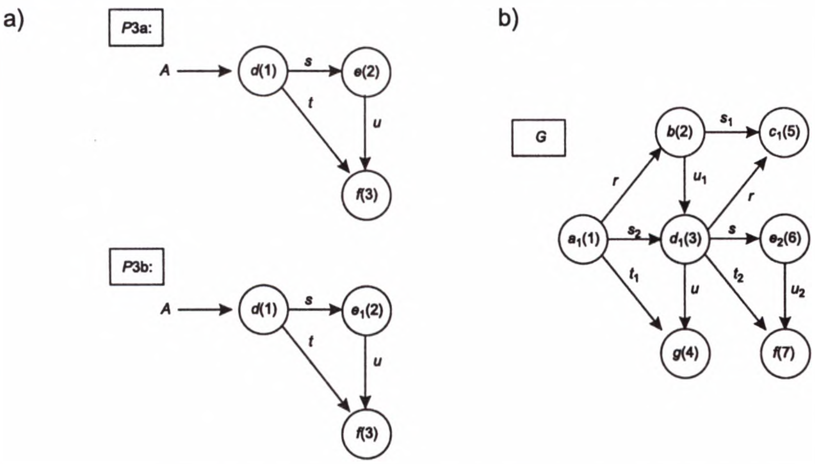

Fig. 8. New productions in $\mathcal{G}$ (a); the analyzed graph – G (b)

In a result of parsing we get graph H which is generated in subsequent steps (Fig. 9). The node and edge costs are calculated as it follows:

Step 0. $c_{n0} = \delta^{\Sigma}(a, a_1)$, $c_e0 = 0$;

Step 1. $c_{n1} = \delta^{\Sigma}(c, c_1)$, $c_{e1} = \delta^{\Gamma}(s, s_1) + \delta^{\Gamma}(u, u_1)$;

Step 2. $c_{n2} = \delta^{\Sigma}(d, d_1) + \delta^{\Gamma}(e_1, e_2)$, $c_{e2} = \delta^{\Gamma}(s, s_2) + \delta^{\Gamma}(t, t_2) + \delta^{\Gamma}(u, u_2)$;

Step 3. $c_{n3} = 0$, $c_{e3} = \delta^{\Gamma}(t, t_1)$,

where:

$$\begin{aligned} a_1 &\in SAOD(a), \\ c_1 &\in SAOD(c), \\ d_1 &\in SAOD(d), \\ e_1, e_2 &\in SAOD(e), \\ s_1, s_2 &\in SASRD(s), \\ t_1, t_2 &\in SASRD(t), \\ u_1, u_2 &\in SASRD(u). \end{aligned}$$

We also assume that $\delta^{\Sigma}(e_1, e_2) < \delta^{\Gamma}(e, e_2)$.

Remark

It should be noted that production P_{3a} has been rejected in the parsing process, because the distance between rhs graph and a subgraph CTL(G, 3, 6) for this production is greater than in case of P_3b.

The total distance (cost) between G and H, $\delta(G, H)$, is given by

$$\delta(G, H) = C_n + C_e = \sum_{i=0}^{3} c_{ni} + \sum_{i=0}^{3} c_{ei}.$$

Now after the parser accepted G as belonging to the language we can begin to analyze the type of distortion. C_e and C_n will be a helpful quantities for that purpose.

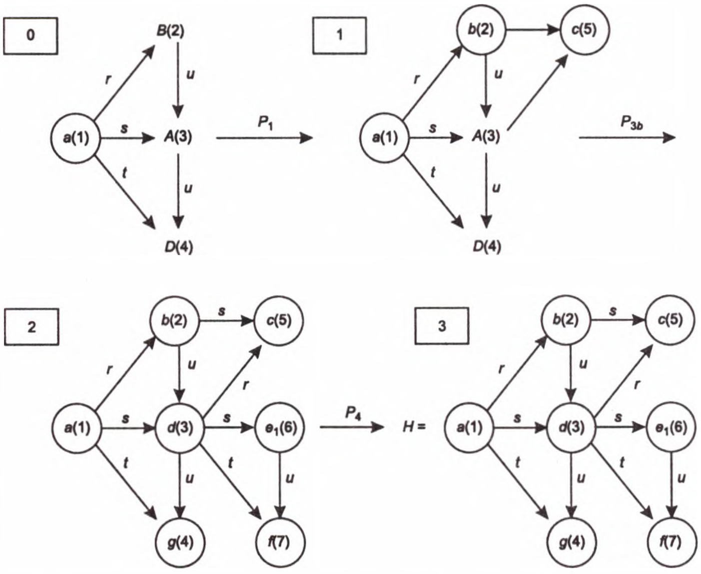

Fig. 9. The derivation steps during graph G analysis (see Fig. 8b)

6. Semantic phase – graph distortion function

In the Preliminaries 2. we have ascribed the functions μ_x^Σ and μ_γ^Γ to nondistorted nodes and edges. These functions give a similarity measure between two nodes/edges. We can define a similar function for the entire graph μ: $\mathrm{IE}_f \times \mathrm{IE}_f \rightarrow \mathrm{R}_+ \cup \{0\}$. A value of μ tells how much differs the analyzed graph from the generated one. It may be another criteria of acceptance of the graph G (besides the belonging to the language). Such criteria may be regarded as semantic ones: we reject the graph belonging to the language (satisfying syntactic criteria) on a basis of some additional (semantic) information, for instance $C_e > 0.5$. A semantic information given by the function μ depends on its shape and may vary depending on the problem. Consider for example the problem in which distortions of nodes may be neglected, while distortions of the spatial relations among them are restricted by μ:

$$\mu(G,H) = \begin{cases} \exp(-C_e^2), & \text{for } C_e^2 < \text{const}, \\ 0, & \text{otherwise.} \end{cases}$$

The following examples illustrates the basic cases of semantic pattern analysis (note that we may neglect either spatial relations (c) or nodes distortions (d)):

a) $\mu > 0 \Leftrightarrow f_1(C_n) < \mathrm{const}_1$ and $f_2(C_e) < \mathrm{const}_2$,

b) $\mu > 0 \Leftrightarrow f\,(C_n, C_e) < \mathrm{const}$,

c) $\mu > 0 \Leftrightarrow f\,(C_n) < \mathrm{const}$ (edge fuzziness neglected),

d) $\mu > 0 \Leftrightarrow f\,(C_e) < \mathrm{const}$ (node fuzziness neglected),

where: $C_n = \sum_S c_n$,

$C_e = \sum_S c_e$.

For the case (b) the μ function may has a form:

$$\mu(G,H) = \begin{cases} \exp\,(-C_n^2 + C_e^2), & \text{for } C_n^2 + C_e^2 < \mathrm{const}, \\ 0, & \text{otherwise.} \end{cases}$$

References

[1] Flasiński M.: *Distorted pattern analysis with the help of node label controlled graph languages.* Pattern Recognition, 23, 1990, 765–774

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

[3] Flasiński M.: *Power properties of NLC graph grammars with a polynomial membership problem.* Theoretical Computer Science, 201, 1998, 189–231

[4] Flasiński M.: *Parsing of edNLC-graph grammars for scene anaysis.* Pattern Recognition, 21, No. 6, 1988

[5] Ehrig H., Kreowski H.J., Rozenberg G. (Eds): *Graph Grammars and their Application to Computer Science.* Lecture Notes in Computer Science, 153, Berlin, Springer 1983

[6] Ehrig H., Kreowski H.J., Rozenberg G. (Eds): *Graph Grammars and their Application to Computer Science.* Lecture Notes in Computer Science, 291, Berlin, Springer 1987

[7] Ehrig H., Kreowski H.J., Rozenberg G. (Eds): *Graph Grammars and their Application to Computer Science.* Lecture Notes in Computer Science, 532, Berlin, Springer 1991