Languages Representable by Vertex-labeled Graphs

Igor Grunsky[1], Oleksiy Kurganskyy[1] and Igor Potapov[2]

[1] Institute of Applied Mathematics and Mechanics,
Ukrainian National Academy of Sciences
74 R. Luxemburg St, Donetsk, Ukraine,
grunsky@iamm.ac.donetsk.ua, kurgansk@gmx.de
[2] Department of Computer Science,
University of Liverpool, Chadwick Building,
Peach St, Liverpool L69 7ZF, U.K.,
igor@csc.liv.ac.uk

Abstract. In this paper we study the properties of undirected vertex-labeled graphs and the limitations on the languages that they represent. As a main result of this paper we define the necessary and sufficient conditions for the languages to be representable by a class of undirected vertex-labeled graphs and its subclasses. We assume that all obtained results and techniques are transferable to the case of undirected edge-labeled graphs and might give us similar results. The simplicity of necessary conditions emphasizes the naturalness of the result. The proof of their sufficiency is quite non-trivial and it is based on a new notion of quasi-equivalence, that is significantly different from Myhill-Nerode equivalence and might not be reduced to it.

1 Introduction

A formal language is a set of finite-length words that can be specified in a great variety of ways and be associated with different combinatorial objects [2, 4, 6, 7]. For example we can consider a set of paths in a labeled graph that define a language in an alphabet of graph labels. Actually, different classes of labeled graphs can place different limitations on the languages that they represent. It is a well known fact that the necessary and sufficient condition for the language to be representable by finite edge-labeled graphs, is its regularity. The vertex-labeled graphs place another structural limitation:

Theorem 1. *[6]. A language L can be represented by a vertex-labeled graph iff the following two base properties hold:*

1) *all words in L start from the same symbol and*
2) *language L is closed under all nonempty prefixes.*

It is also known from [6] that a language L can be represented by *finite* vertex-labeled graph iff L satisfies to the *base properties* and it is regular.

In this paper we study the properties of undirected vertex-labeled graphs with one initial vertex and all final vertices and the limitations on the languages that they represent. The language that we consider is just a set of labels for all paths (in a graph) starting from the initial vertex. As a main result of this paper we define the necessary and sufficient conditions for the languages to be representable by a class of undirected vertex-labeled graphs and its subclasses. In principal the presented results and techniques are transferable to the case of undirected edge-labeled graphs as well and it is most likely that they can give us similar results.

The simplicity of necessary conditions emphasizes the naturalness of the result. The proof of their sufficiency is quite non-trivial and it is based on a new notion of *quasi-equivalence* of vertices, that is significantly different from Myhill-Nerode equivalence and might not be reduced to it. An example of a graph with three quasi-equivalent vertices (labeled by "b") is shown on the Figure 1.

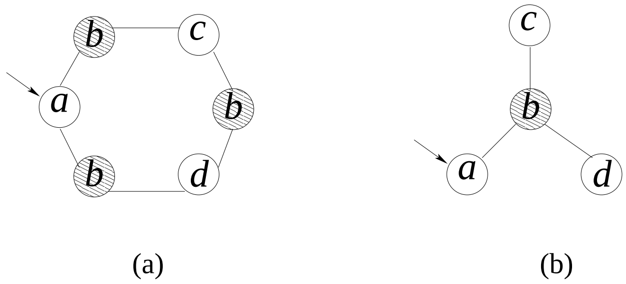

Fig. 1. An example of undirected graph with the initial vertex "a" (before and after the gluing of three quasi-equivalent vertices).

The only common property of equivalent and quasi-equivalent vertices in an undirected vertex-labeled graph is that the gluing of equivalent as well as quasi-equivalent vertices does not change the language (representable by this graph).

This paper is organised as follows. Next section contains preliminaries. In the third section we give a technical proofs of intermediate results about graph modifications, that we use in Section 4 and 5. Next we introduce a new notion of quasiequivalent vertices and prove some of their properties, which are the core elements of proofs from Section 5. The Section 5 contains the main results of this paper. We present here the characterization theorems for classes of languages representable by finite undirected graphs, infinite undirected graphs and finite undirected deterministic graphs. The paper ends with some conclusions.

2 Preliminaries

Let X be a finite alphabet and X^* be a free monoid on the set X. We denote an empty word by e and $X^* - \{e\}$ by X^+. The length of a word $w = x_1x_2 \dots x_k$, where $x_i \in X$, $1 \leq i \leq k$, is denoted by $|w|$ and w^{rev} is the reverse word $x_kx_{k-1} \dots x_1$. We also introduce a partial binary associative operation of *splicing* on a set X^+ denoted by $\circ$.

Definition 1. *The splicing of two words is defined as follows:*

$$wx \circ yu = \begin{cases} wxu, & x = y \\ undefined, & x \neq y \end{cases}$$

where $x, y \in X$ and $w, u \in X^*$.

Now we define other notations in terms of splicing. Let w be a word that can be represented as $u_1 \circ u_2 \circ u_3$, where $u_1, u_2, u_3 \in X^+$. We call u_1 as a prefix of the word w and u_3 as a suffix of w. A language $L \subseteq X^+$ is closed under all non-empty prefixes if $u \in L$ follows from $u \circ w \in L$, where $u, w \in X^+$.

Definition 2. *Let $L, L' \subseteq X^+$. The basic language operations of splicing and prefix contraction are defined as follows:*

- *a splicing of two languages: $L \circ L' = \{w \circ u | w \in L, u \in L'\}$*
- *a prefix contraction of a word w from a language L: $w^{-1} \circ L = \{u | w \circ u \in L\}$*
- *a prefix contraction of a language L' from a language L: $(L')^{-1} \circ L = \bigcap_{w \in L'} w^{-1} \circ L$*

Definition 3. *Let $G = (S, E, X, \mu)$ be a simple (i.e without multiple edges), directed vertex-labeled graph, where S is a (possibly infinite, but counted) set of vertices, E is a set of directed edges, X is a finite set of labels, $\mu : S \to X$ is a mapping from set of vertices to the set of labels and $\mu(s)$ is a label of a vertex $s \in S$.*

The vertices s and t are *adjacent* if they are connected by an edge in G that we denote by st or (s, t). A graph G is *undirected* if for any edge $st \in E$ there is the edge $ts \in E$. A pair of edges st and ts is an undirected edge that we denote by $\{s, t\}$. A graph G is called initial if in the set of vertices there is the initial vertex s_0. In order to indicate that a vertex s_0 is the initial vertex in a graph G we denote it by $G(s_0)$. From now on we only consider initial graphs and assume that intersection of set of vertices for different graphs is empty.

A finite sequence of vertices $p = s_1s_2 \dots s_k$ such that $s_is_{i+1} \in E, 1 \leq i < k$ is a path of length $k - 1$ in a graph G. Let us say that the path p begins in the vertex s_1 and ends in s_k. Then the distance between two vertices in a graph is the length of the shortest path between them.

The label $\mu(p)$ of a path $p = s_1s_2 \dots s_k$ in a graph is a word

$$\mu(s_1)\mu(s_2) \dots \mu(s_k),$$

that is concatenation of vertex labels on p.

Definition 4. *Let P and Q are two subsets of the set of vertices S, i.e. $P, Q \subseteq S$. By $P^{-1}Q$ we denote the set of labels for each paths in a graph G that begins in a vertex from the set P and ends in a vertex from the set Q.*

Now we introduce several special cases for the notation $P^{-1}Q$. Let us consider the case where $P = \{s\}$ and $Q = \{t\}$ then we have that $P^{-1}Q = s^{-1}t$. If s is the initial vertex in a graph G then we denote $s^{-1}t$ by $\mathbf{t}$. Thus, the notation $\mathbf{t}$ in an initial graph corresponds to the set of labels of all paths from the initial vertex to the vertex t. The notation $\mathbf{s^{-1}}$ stands for the set $P^{-1}Q$, where $P = \{s\}$ and $Q = S$. The notation $\mathbf{s^{-1}}$ denotes a set of labels of all paths from a vertex s that we call a language generated by the vertex $s \in S$ in a graph G.

Definition 5. *By $L(G)$ we denote a language generated from the initial vertex s_0 in an initial graph G, i.e. $L(G) = \mathbf{s_0}^{-1}$.*

By sw, where $s \in S$, $w \in X^+$, we denote the set of all reachable vertices from a vertex s by any path with a label w, i.e. $sw = \{t \in S | \exists p = s_1 s_2 \dots s_k \in S^* : s_1 = s, s_k = t, \mu(p) = w\}$.

Definition 6. *Two vertices $s, t \in S$ are equivalent, if $\mathbf{s}^{-1} = \mathbf{t}^{-1}$. This equivalence relation is denoted as ϵ.*

Definition 7. *A graph G is reduced if it does not contain any equivalent vertices.*

Definition 8. *A graph G is deterministic if any vertex $s \in S$ does not have any two adjacent vertices with the same label.*

Let us given some equivalence relation $\rho \subseteq S \times S$ (reflexive, transitive and symmetric relation), such that two vertices s_1 and s_2 have the same labels if they are in the same equivalence class generated by ρ. By $\rho(s)$, where $s \in S$, we denote a set $\{t | (s, t) \in \rho\}$.

Definition 9. *The graph $G/\rho = (G', E', X', \mu')$ is a factor graph of a graph G if $G' = \{\rho(s) | s \in S\}$, $\rho(s)\rho(s') \in E'$ for all edges $ss' \in E$, $\mu'(\rho(s)) = \mu(s)$, $\rho(s_0)$ and s_0 are initial vertices in G/ρ and G, respectively.*

A graph G/ρ from G can be constructed by gluing all vertices in equivalence class $\rho(s)$ for each $s \in S$. In particular we say that G' is constructed from G by gluing two vertices s and t if G' and G/ρ are isomorphic and the equivalence relation ρ partitions the set of vertices of a graph G into one-element equivalence classes excepting a single two-element equivalence class $\rho(s) = \{s, t\}$.

Now we define two kind of simple graphs that we call $Line$ and $Yarn$. Let $w = x_1 x_2 \dots x_k$ is a word of a length k. By $Line(w)$ we denote an undirected graph with k vertices $s_1, \dots, s_k$, where s_1 is the initial vertex, $\{\{s_i, s_{i+1}\} | 1 \leq i < k\}$ is a set of edges and each vertex s_i has a label x_i, $1 \leq i \leq k$. An example for a graph $Line(w)$ where $w = abcbb$ is shown on the Figure 2.

Definition 10. *By $Yarn(L)$ for a language L, in which all words start from the same symbol, we denote an undirected graph that is constructed from graphs $\{Line(w) | w \in L\}$ by their direct sum and by gluing the initial vertices of all graphs $Line(w)$ into one initial vertex of $Yarn(L)$.*

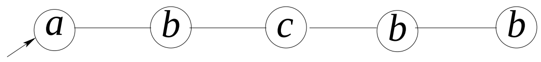

Fig. 2. An example for a $Line(w)$ where $w = abcbb$.

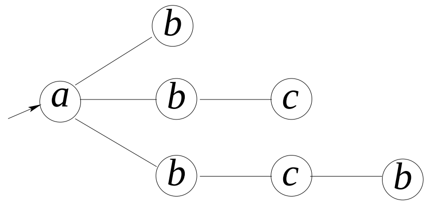

Fig. 3. An example for a $Yarn(L)$, where $L = \{ab, abc, abcb\}$.

An example for $Yarn(L)$, where $L = \{ab, abc, abcb\}$, is shown on the Figure 3.

A class of regular languages we denote by *Reg*. We say that a language $L \subseteq X^+$ is representable by a graph (that is denoted by $L \in Graph$) if there exist an initial graph G, such that $L(G) = L$.

We also use the following notations for languages generated by different classes of graphs: *FGraph* is a class of languages generated by finite graphs, *DGraph* is a class of languages generated by deterministic graphs, *UGraph* is a class of languages generated by undirected graphs, *FUGraph* is a class of languages generated by finite undirected graphs, *FUDGraph* is a class of languages generated by finite undirected deterministic graphs. For example, $L \in FUGraph$ means that there is a finite undirected graph G such that $L(G) = L$.

3 Intermediate Results

In this section we show a number of intermediate results that we use for characterization of classes *UGraph*, *FUGraph* and *FUDGraph*. Note, that the languages and graph labels are defined in the same finite alphabet X.

Lemma 1. *Let G be a directed vertex-labeled graph and s, t are two vertices in G with labels x and y, respectively and the following two properties:*

- *an edge ts does not belong to the graph G and*
- $(yx)^{-1} \circ \mathbf{t}^{-1} \supseteq \mathbf{s}^{-1}$.

If G' is a graph constructed from G by adding an edge ts then the language generated by G' does not differ from a language generated by G, i.e. $L(G) = L(G')$.

Proof. Let us add one edge ts to a directed vertex-labeled graph G, that is obviously can only extend language $L(G)$. Thus, it is enough to show that for any path p which goes via an edge ts from the initial vertex we can find a path p' with the same label that also starts from the initial vertex without passing an edge ts, i.e. $\mu(p) = \mu(p')$.

Let us consider a path $p = s_1 s_2 \ldots s_k$ from the initial vertex such that passes an edge ts in a graph G'. Let $s_1s_2, s_2s_3, \ldots, s_{k-1}s_k$ be a sequence of edges that corresponds to the path p and contains an edge ts. Let i be a maximum number, such that $ts = s_i s_{i+1}$. According to the initial condition $(yx)^{-1} \circ \mathbf{t}^{-1} \supseteq \mathbf{s}^{-1}$ there is a path $s_i s'_{i+1} s'_{i+2} \ldots s'_k$ with a label $x_i x_{i+1} x_{i+2} \ldots x_k$, where $x_i x_{i+1} = yx$, but which does not pass an edge ts. Thus there is a path $r = s_1 \ldots s_i s'_{i+1} s'_{i+2} \ldots s'_k$ in G' with a label $x_1 x_2 \ldots x_k$ such that a number of ts-edges in r is less than in p by one. From it follows that for any path p in G' we can construct a path p' in G', that does not contain any ts edges, by repeating the above reduction a finite number of times. In such case we have that for any path p in G there is a path p' in G such that $\mu(p) = \mu(p')$ and therefore $L(G) = L(G')$.

Lemma 2. *For any (finite or infinite) graph G the identity $L(G) = L(G/\epsilon)$ holds.*

Proof. Let G be an initial vertex-labeled graph and let s and t be an equivalent vertices in G. Let us assume that for some vertex s' there is an edge ss' and there is no edge ts' in a graph G. In this case we meet initial condition of Lemma 1, so the language represented by a graph G would not be changed by extending G with an edge ts'. Moreover, a language generated by any vertex in the extended graph would not be changed as well.

Let us extend the graph G in the following way: for any equivalent vertices s and t (i.e. $(s,t) \in \epsilon$) and some vertex s' we add an edges ts' iff an edge ss' belongs to the graph.

From it follows that the sequence of vertices $\epsilon(s_1), \epsilon(s_2), \ldots, \epsilon(s_k)$ is path starting from the initial vertex in a factor graph G/ϵ iff there exist a sequence of vertices $r_1, r_2, \ldots, r_k$, where $r_i \in \epsilon(s_i)$, $1 \leq i \leq k$, that is a path in a graph G and r_1 is its initial vertex. Above facts show that for any graph G the equality $L(G) = L(G/\epsilon)$ holds.

4 Quasiequivalent Vertices

Let us define a quasi-equivalence relation that is significantly different from ϵ.

Definition 11. *Given a graph G that generates a language L. Two vertices s and t from a graph G are quasiequivalent if $(\mathbf{s})^{-1} \circ L = (\mathbf{t})^{-1} \circ L$.*

Let us denote the quasiequivalence relation of vertices by α.

Definition 12. *A graph G is irreducible if it does not contain any quasiequivalent vertices.*

It is clear that equivalence and quasiequivalence relations coincide in deterministic graphs. Lemma 4 shows us another property of quasiequivalence relation that $L(G) = L(G/\alpha)$.

Lemma 3. *Given a graph G and a language L with the following base properties*

- *all words in L start from the same symbol and*
- *language L is closed under all nonempty prefixes.*

The language $L(G) \subseteq L$ iff we have that $\mathbf{s}^{-1} \subseteq (\mathbf{s})^{-1} \circ L$ for any vertex s in a graph G.

Proof. Let us remind, that in expression "$(\mathbf{s})^{-1}$" from "$\mathbf{s}^{-1} \subseteq (\mathbf{s})^{-1} \circ L$" by $\mathbf{s}$ we denote the set of labels of all paths that start from the initial vertex of a graph G and end in the vertex s, which are words from X^+. The above lemma follows from the fact that $\mathbf{s}^{-1} \subseteq (\mathbf{s})^{-1} \circ L$ is equivalent to $\mathbf{s} \circ \mathbf{s}^{-1} \subseteq L$.

Lemma 4. *For any (finite or infinite) graph G the identity $L(G) = L(G/\alpha)$ holds.*

Proof. Let G be an initial vertex-labeled graph. Note that the operation of gluing any two vertices in a vertex-labeled graph can only extend the language that it represents. Let us assume that:

– $L = L(G)$,
– s and t are quasiequivalent vertices and
– $W = (\mathbf{s})^{-1} \circ L = (\mathbf{t})^{-1} \circ L$.

From Lemma 3 follows that $\mathbf{s}^{-1} \subseteq W$ and $\mathbf{t}^{-1} \subseteq W$.

Let us construct an extension of graph G by adding new vertices and edges in such a way that $\mathbf{s}^{-1} = \mathbf{t}^{-1} = W$ in an extended graph holds. In particular we can do it as follows. Let us add by direct sum a graph $Line(w)$ to a graph G for the vertex s and each word $w \in W$ and glue a vertex s with the initial vertex of $Line(w)$. Next we can repeat the same construction with the vertex t. According to the Lemma 3 the language represented by a graph G after its extension is not changed.

Since a pair of vertices s and t are now equivalent in the extended graph G we can glue it without changing the language represented by this graph. Now we can delete all introduced edges and nodes to get a graph that can be constructed by just gluing quasiequivalent vertices s and t.

Lemma 5. *If G be infinite graph and $L(G)$ be a language such that $L(G) \in FGraph$ then a graph G/α is finite.*

Proof. Since the power of a set $\{(W)^{-1} \circ L(G) | W \subseteq L(G)\}$ is bounded by a number 2^N, where $N = |\{w^{-1} \circ L(G) | w \in L(G)\}|$ we have a finite number of classes with quasiequivalent vertices.

Note, that the proposition in Lemma 5 does not hold for the relation ϵ.

The example of an infinite directed graph G with quasiequivalent vertices is shown on the Figure 4. It is clear that $L(G) = a(aa)^*b$. The vertices with the same pattern in the Figure 4 are quasiequivalent. This graph has no equivalent vertices except those that labeled with b. The graph G/ϵ on the Figure 5, is infinite and the graph G/α on the Figure 6 is finite. However it follows from Lemma 2 and Lemma 5 that $L(G) = L(G/\epsilon) = L(G/\alpha)$.

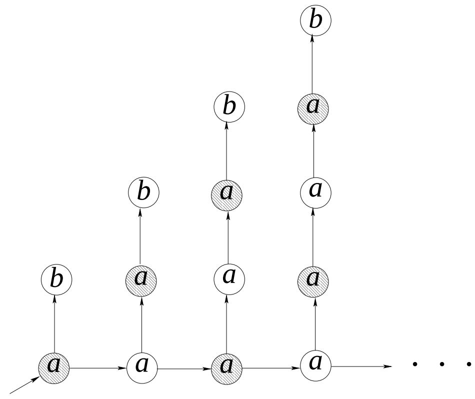

Fig. 4. An example of quasiequivalent vertices. The vertices with the same pattern are quasiequivalent, but not equivalent.

5 Characterization of Graph Representable Languages

In this section we present our main results about characterization of languages that can be represented by undirected graphs.

Definition 13. *Symmetric closure of a language $L \in Graph$ is a minimal language $[L] \in UGraph$ such that $L \subseteq [L]$ (note that $L \in UGraph$ iff $L = [L]$).*

Theorem 2. $L(Yarn(L)) = [L]$.

Proof. By Definition 10, $Yarn(L)$ is an undirected graph that represents a language that include all words from a language L. From it follows that $[L] \subseteq L(Yarn(L))$. Let us prove the inverse proposition $L(Yarn(L)) \subseteq [L]$.

Let G be a graph such that $L(G) = [L]$ and $p = s_1 s_2 \dots s_k$ is a path from G, that starts in the initial vertex and $\mu(p) = w$. Let a graph $Line(w)$ has a set of vertices $\{t_1, t_2, \dots, t_k\}$, and set of edges $\{t_i t_{i+1}\}$, $1 \leq i < k$ and t_1 is the initial vertex. Now we define a mapping ϕ such that $\phi(t_i) = s_i$, $1 \leq i \leq k$, which is a homomorphism of graph $Line(L)$ to G. From the above mapping and the definition of $Yarn(L)$ follow that there is a homomorphism from $Yarn(L)$ to G. Thus, we have that $L(Yarn(L)) \subseteq [L]$.

Now we have a straightforward but useful corollary.

Corollary 1. *$L \in UGraph$ iff $L = L(Yarn(L))$.*

Theorem 3. *A language L can be represented by undirected graph, i.e. $L \in UGraph$, iff it satisfies to the following properties:*

1) all words of the language L are starting from the same symbol
2) the language L is closed under prefixes

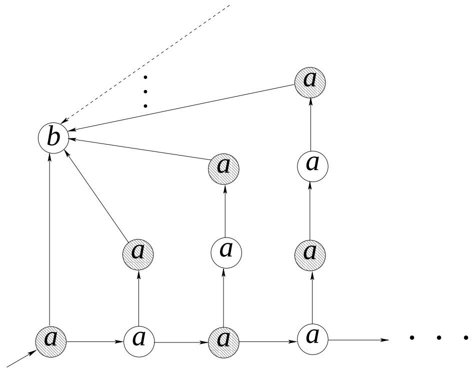

Fig. 5. The graph G/ϵ, where G is shown on Figure 4.

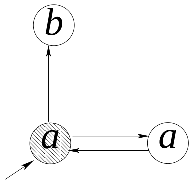

Fig. 6. The graph G/α, where G is shown on Figure 4.

3) if $w, u \in L$ *then* $w \circ w^{rev} \circ u \in L$
4) if $w \circ u \circ v \in L$ *then* $w \circ u \circ u^{rev} \circ u \circ v \in L$.

Proof. The straightforward check of the above four properties give us the necessity of conditions. Let us prove that these four properties are also sufficient. Namely we prove that $L = L(Yarn(L))$ follows from the properties 1-4.

In the first part of the proof we show that if $w = x_1x_2 \dots x_k \in L$ then $L(Line(w)) \subseteq L$. A graph $Line(w)$ has k vertices (be definition) that we denote by s_i, $1 \leq i \leq k$, with initial vertex s_1 and a set of edges $\{\{s_i, s_i+1\} | 1 \leq i < k\}$, where s_i has a label x_i. We will also say that i is the coordinate of a vertex s_i.

Let p be a path from a vertex s_1 in a graph $Line(w)$ which differs from $s_1s_2 \dots s_k$. Let us prove that the label of the path p belongs to the language L, i.e. $\mu(p) \in L$. From property 2 of this theorem follows that if extension of any path belongs to a language L the original path also belongs to L, so without loss of generality we can assume that p ends in a vertex s_k. Let us represent p in the form

$$r_1 \circ l_1 \circ r_2 \circ l_2 \circ \dots \circ r_{n-1} \circ l_{n-1} \circ r_n,$$

where l_i is a path in $Line(w)$ with decreasing coordinates (movement to the left) and r_i is a path in $Line(w)$ with increasing coordinates (movement to the right). Now we can show that there exist a path p' in $Line(w)$, where p' starts and ends at the same vertex as p and $|p'| < |p|$, such that from $\mu(p') \in L$ follows $\mu(p) \in L$. Note that we can get p' from p by deleting some of its subpaths. If we continue the process of deletion we show that from $\mu(s_1 s_2 \ldots s_k) \in L$ follows $\mu(p) \in L$.

Let a path r_i, $1 \leq i \leq n$ starts and ends in vertices with coordinates u_i and o_i, respectively. Actually there are only two cases: either $u_1 < u_2 < \ldots < u_n$ or $u_{i+1} \geq u_{i+2}$ for some i, $0 < i < n-1$.

Case 1. Let us assume that $u_1 < u_2 < \ldots < u_n$ for a set of subpaths $r_1, \ldots, r_n$ in path p. It is obvious that the property $o_1 > o_2 > \ldots > o_n$ does not hold for p since r_n ends with a maximal coordinate k, i.e. $o_n = k$. Thus, there is i, $1 \leq i < n$, such that $o_i \leq o_{i+1}$. From it follows that a path r_i can be represented as $r_i = r \circ l_i^{rev}$ and r_{i+1} as $r_{i+1} = l_i^{rev} \circ r'$. Thus we have that

$$p = r_1 \circ l_1 \circ \ldots \circ r \circ l_i^{rev} \circ l_i \circ l_i^{rev} \circ r' \circ \ldots \circ r_n,$$

and therefore

$$p' = r_1 \circ l_1 \circ \ldots \circ r \circ l_i^{rev} \circ r' \circ \ldots \circ r_n.$$

is a path in $Line(w)$. It is easy to check that $\mu(p) \in L$ if $\mu(p') \in L$, since we can construct $\mu(p)$ from $\mu(p')$ by applying the property 4 of the theorem.

Case 2. Let us assume that $u_{i+1} \geq u_{i+2}$ for some i, $0 \leq i < n-1$. There are also two subcases: $o_i \leq o_{i+1}$ and $o_i > o_{i+1}$. In the proof of the first subcase we mimic the case 1 argumentation. So we consider here only the second subcase.

If $o_i > o_{i+1}$ then a subpath l_i can be represented as $l_i = l \circ r_{i+1}^{rev}$ and a subpath l_{i+1} can be represented as $l_{i+1} = r_{i+1}^{rev} \circ l'$. From it follows that

$$p = r_1 \circ \ldots \circ l \circ r_{i+1}^{rev} \circ r_{i+1} \circ r_{i+1}^{rev} \circ l' \circ \ldots \circ r_n$$

and therefore there is a path p' in a graph $Line(w)$ such that

$$p' = r_1 \circ \ldots \circ l \circ r_{i+1}^{rev} \circ l' \circ \ldots \circ r_n.$$

So we have that $\mu(p) \in L$ if $\mu(p') \in L$, since we can construct $\mu(p)$ from $\mu(p')$ by applying the property 4 of the theorem. Now we can apply the same method a number of times to transfer path p into a path $s_1 s_2 \ldots s_k$, that has a label from a language L. Thus $\mu(p) \in L$ and $L(Line(w)) \subseteq L$.

It is clear that the deletion of all incoming edges for the initial vertex of a graph $Yarn(L)$ give us a graph that generate L. In the second part of the proof we show that the property 3 allows us to restore the deleted edges without any changes in a language generated by that graph. It implies the fact that $L = L(Yarn(L))$.

Let us consider a graph G that generates a language L and let a vertex t be an adjacent to the initial vertex s. It follows from the property 3 of the Theorem that any vertex t satisfies to Lemma 1. It means that any edge, outgoing from the initial vertex in a graph G can be transformed into undirected edges without any changes of the language that G generates. Now we can come to the conclusion that $L = L(Yarn(L))$, since we can restore all edges (incoming to the initial vertex) that we deleted from $Yarn(L)$.

The Theorem 3 shows the characterization of languages generated by undirected vertex-labeled graphs (i.e $UGraph$). Now we consider the class $FUGraph$, which is a subclass of $UGraph$.

Theorem 4. $FUGraph$=$FGraph \cap UGraph$

Proof. Let L be a language, such that $FGraph \cap UGraph$. From Lemma 5 follows that any infinite undirected graph has a finite number of quasiequivalent classes. By gluing all quasiequivalent vertices in each class we have a finite graph that generates L. Since the graph obtained from an undirected graph by gluing quasiequivalent vertices is undirected, we have that $FUGraph$=$FGraph \cap UGraph$.

Since $FUGraph = FGraph \cap UGraph$ is equivalent to $FUGraph = Reg \cap UGraph$ we have the following corollary:

Corollary 2. *The langauge L is representable by a finite undirected graph, i.e $L \in FUGraph$ iff L satisfies to the properties of Theorem 3 and $L \in Reg$.*

Since Lemma 5 gives us an upper bound on a number vertices in any irreducible graph that generate a language $L \in FUGraph$, the following corollary holds:

Corollary 3. *The membership problem for FUGraph, i.e. checking whether a language L belongs to the class FUGraph, is decidable.*

Proof. From Lemma 5 follows that the number of quasieqivalent vertices is bounded by a constant k. Thus we can run over the whole set of undirected detrministic and nondetrministic graphs with a number of verices less then k and check if the language L can be represented by some of these graphs, which is just the comparison of regular languages. If we can find at least one such graph then the language L is in $FUGraph$ and $L \notin FUGraph$ otherwise.

As a last result of this section we present a characterization of languages generated by finite undirected deterministic graphs.

Theorem 5. *Given a reduced deterministic graph G which generates a language L. The language $L \in FUDGraph$ iff G is undirected.*

Proof. If a reduced deterministic graph G, which generates L, is undirected then $L \in FUDGraph$ by definition. Let us prove the necessary part of this proposition.

Let G be a finite undirected deterministic graph, which generates a language L. Since $L \in FUDGraph$ we can state that such graph exists. Since equivalence and quasiequivalent relations coincide in deterministic graphs we have that G/α is a deterministic graph, which generates L. On the other hand G/α is undirected graph, which required to be proved.

Theorem 6. *The langauge L is representable by a finite undirected deterministic graph, i.e $L \in FUDGraph$ iff L satisfies to the following properties:*

1) $L \in Reg$*;*
2) all words in L *starting from the same symbol;*
3) language L *is closed under all nonempty prefixes;*
4) if $wxy \in L$*, where* $x, y \in X$ *and* $w \in X^+$ *then* $(wxyx)^{-1} \circ L = (wx)^{-1} \circ L$.

Proof. The necessity of the above properties is obvious, so we only prove that they are sufficient. Let G be a irreducible graph and $L = L(G)$. Then we have that the set $\{\mathbf{s}^{-1} | s \in G\}$ is equal to the set $\{w^{-1} \circ L | w \in L\}$. From property 4 of this Theorem follows that if from a vertex s to a vertex t in G there exist an edge st then there exist an edge from t to s. Thus, the graph G is undirected, q.e.d.

6 Conclusion

In this work we gave a characterization of languages that can be represented by different types of undirected graphs. Note that all results are ultimate in terms that both necessary and sufficient propositions were proved. The topic is also related to exploration of graphs by software entity (i.e. finite automata, mobile robots, software agents etc.) and map validation problems that have received a widespread attention in recent years $[1, 5, 7, 8]$.

References

1. Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. Map Validation and Robot Self-Location in a Graph-Like World. Robotics and Autonomous Systems, Vol. 22(2), 159-178 (1997)
2. Samuel Eilenberg. Automata, Languages, and Machines. Academic Press, Inc. Orlando, FL, USA (1974)
3. I.S. Grunsky, Oleksiy Kurganskyy. Properties of languages that generated by undirected vertex-labeled graps. In proceeding of 8-th International Seminar on Discrete mathematics and its applications, Moscow State University, 267-269 (2004) [in Russian].
4. Matthias Jantzen, Alexy Kurganskyy: Refining the hierarchy of blind multi-counter languages and twist-closed trios. Inf. Comput. 185(2): 159-181 (2003)
5. Tod S. Levitt, Daryl T. Lawton. Qualitative Navigation for Mobile Robots. Artif. Intell. 44(3): 305-360 (1990)
6. Yu. V. Kapitonova and A.A. Letichevsky. Mathematical theory of computer systems design, Nauka, 286 pages (1988) [in Russian]
7. V. B. Kudryavtsev, Sh. M. Ushchumlich and G. Kilibarda. On behaviour of automata in labyrinths. Discrete Mathematics and Applications Volume 3, No. 1, 1-28 (1993)
8. S.V. Sapunov. An equivalence of labeled graphs. Tr. Inst. Prikl. Mat. Mekh. 7, 162-167 (2002) [in Russian]