

THE AUTOMATA COMPLEXITY OF THE LANGUAGE TRANSFORMATION SCHEMA THAT CONTAINS OPERATIONS WITH RESTRICTED DEGREE

NGUYEN VAN DINH

Abstract. In accordance with the concept of *automaton with the output* we have built the *language transformation schema* Σ (see [6]). In this paper we study the relation between the *automata complexity* $g(\Sigma)$, the *number of essential vertices* $|\Sigma|$ and the *depth of operations* s on a *language transformation schema*. The estimation of the automata complexity of a language transformation schema that holds operations with restricted degree is also given.

Tóm tắt. Dựa trên khái niệm ô tô máy có lối ra, ta xây dựng được lược đồ biến đổi ngôn ngữ (BDNN) [6]. Khi các lược đồ BDNN chứa các phép lặp không hạn chế thì độ phức tạp ô tô máy của nó không phụ thuộc vào số phép lặp và đã được đánh giá bởi [6], nhưng khi số phép lặp, cũng như một số phép toán khác như phép lấy các cặp từ chẵn, lấy các cặp từ lẻ, phép bỏ sung..., có bậc bị hạn chế thì độ phức tạp ô tô máy của các lược đồ BDNN sẽ phụ thuộc vào độ sâu đặt dấu (bậc) của các phép toán.

Bài này trình bày mối liên hệ giữa độ phức tạp ô tô máy, số đỉnh cốt yếu và độ sâu đặt dấu các phép toán của một lược đồ BDNN có chứa các phép toán có bậc được hạn chế.

1. INTRODUCTION

Using notation $\mathcal{L} = \mathcal{X} \times \mathcal{Y}$, wherein $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ is input alphabet of source (original) language, and $\mathcal{Y} = \{y_1, y_2, \dots, y_m\}$ is output alphabet of target (final) language. An *automaton with the output*, $M = (S, \mathcal{X}, \mathcal{Y}, s_0, \delta, \lambda, F)$, recognizes a language on the input alphabet, symbolized T_X , and transforms it into another language on the output alphabet, that is symbolized T_Y . This language transformation is due to automata mappings, on which the lengths of words are completely preserved. (See [6]). In brief, the automaton M recognizes a language pair $R = (T_X, T_Y)$. Corresponding to automaton with the output M , we can build the language transformation schema equivalent to M , which also recognizes the same language pair with the initial automaton. (See [6]). When the language transformation schema holds unrestricted repetitions, its automata complexity does not depend on the number of the repetitions and has been appraised by [6]. However, if the number of the repetitions, as well as another operations: creating even word pairs, odd word pairs or complement... have limited degrees, the *automata complexity* of the *language transformation schema* will be reliant on the *depth of operations*. This paper is to analyze evaluate the automata complexity of a language transformation schema that contains operations with restricted degree.

2. PRELIMINARY CONCEPTS

Some preliminary concepts about the language transformation schema presented in [6] are summarized in this part. In addition, new relating concepts will be separately described. The language pair (T_X, T_Y) on the dual alphabet $\mathcal{L} = \mathcal{X} \times \mathcal{Y}$, denoted R , and R^m is pair of repeated language with *degree* m of T_X and T_Y . The subset of R , contains word pairs with either even or odd lengths, denoted $\mathcal{E}(R)$ or $\mathcal{O}(R)$, respectively. The set of all initial parts of word pairs, of which lengths are at most s , denoted $\mathcal{H}(R, s)$.

The *language transformation graph* on the dual alphabet $\mathcal{L} = \mathcal{X} \times \mathcal{Y}$ is a *directed graph* G , in which, a particular vertex called *entry vertex*, denoted $\sigma(G)$, and a nonempty set of vertices called the *set of final vertices*, denoted $F(G)$, and we denote the set of all vertices as $A(G)$. Initial vertex and

final vertex of an edge a are denoted $\alpha(a)$ and $\beta(a)$, respectively. On each edge a of the graph G is labelled by a group of word pairs on the dual alphabet \mathcal{L} , this group is denoted $M_G(a)$. Suppose that every edge a_1, a_2, \dots, a_t ($t \geq 1$) is a certain path in G and $T_i \in M_G(a_i)$ is word pairs corresponding to edge a_i ($1 \leq i \leq t$), then word pairs $T_1 T_2 \dots T_t$ are considered is created by this path. We denote the set of all word pairs generated by path initiating from vertex α and ending at vertex β as $N_G(\alpha, \beta)$. The set $N_G(\alpha, \beta)$ is inductively defined as the following principles:

1. $\Lambda = (\varepsilon, \varepsilon)$, wherein ε is an empty word of \mathcal{X}^* and \mathcal{Y}^* .
2. If X, Y are word pairs, wherein $X \in N_G(\alpha, \gamma)$, and $\gamma = \alpha(a)$, $\beta = \beta(a)$, $Y \in M_G(a)$ then $XY \in N_G(\alpha, \beta)$.

Definition 2.1. The language transformation schema Σ on the dual alphabet $\mathcal{L} = \mathcal{X} \times \mathcal{Y}$ is a range of the language transformation graphs on \mathcal{L} :

$$\Sigma = (G_1, G_2, \dots, G_n),$$

and on this range, we build a function $m_\Sigma(a)$, defined on the set of all edges of all graphs G_i ($1 \leq i \leq n$) and satisfied the following conditions:

1. For a certain edge a of G_i ($1 \leq i \leq n$), the function $m_\Sigma(a)$ is satisfied one of following standards:
 - 1.1. $m_\Sigma(a) = \Lambda$ and $M_{G_i}(a) = \Lambda$, then a is called an empty edge.
 - 1.2. $m_\Sigma(a) = (x, y) \in \mathcal{L}$ and $M_{G_i}(a) = \{x, y\}$, then edge a notes a pair of symbols (x, y) and edge a is called essential edge.
 - 1.3. $m_\Sigma(a) = G_j$ ($1 \leq j \leq n - 1$) and $M_{G_i}(a) = \mathcal{E}(N(G_j))$, then edge a is stated to depending on graph G_j and is called an even edge.
 - 1.4. $m_\Sigma(a) = G_k$ ($1 \leq k \leq n - 1$) and $M_{G_i}(a) = \mathcal{O}(N(G_k))$, then edge a is stated to depending on graph G_k and is called an odd edge.
 - 1.5. $m_\Sigma(a) = G_r$ ($1 \leq r \leq n - 1$) and $M_{G_i}(a) = N^S(G_r)$, then edge a is stated to depending on graph G_r and is called repeated edge with degree s .
 - 1.6. $m_\Sigma(a) = G_t$ ($1 \leq t \leq n - 1$) and $M_{G_i}(a) = \mathcal{H}(N(G_t), s)$, then edge a is stated to depending on graph G_t and it is called complement edge with degree s .
2. Each graph G_1, G_2, \dots, G_{n-1} has one and only one edge of graph G_n depending on. Graph G_n called a base graph.
3. Graphs G_1, G_2, \dots, G_n have no common vertex.

G_n is called the base graph of the language transformation schema. If Σ contains only one graph, this unique graph is also signed Σ , and called a simple language transformation schema. The set of word pairs $N(G_n)$ is considered to be created by Σ , and denoted $N(\Sigma)$. We at times use $N_X(\Sigma)$ and $N_Y(\Sigma)$ to denote separately the set of origin and the final words defined by Σ . Obviously, $N(\Sigma) \subseteq N_X(\Sigma) \times N_Y(\Sigma)$.

The vertex α of graph G is called an essential vertex if it is entered by at least one essential edge. The number of essential vertices of graph G is signed $|G|$. The number of essential vertices of all graph belong to Σ is signed $|\Sigma|$. We state that graph G_i depends on graph G_j if it contains some edge depending on graph G_j .

Definition 2.2. For the language transformation schema, the depth of operations, denoted $l(\Sigma)$, and determined as follow:

Let a is an unintentional edge of graph G_r ($1 \leq r \leq n$), then the depth of operations of a is signed $l(a)$, and we define the depth of operation in G by the formula:

$$l(G) = \max_{a \in G} l(a),$$

wherein $l(a)$ is inductively determined as follows:

1. If a is an even edge, odd edge, repeated edge with degree n or complement edge with degree s and a depends on graph G_j ($1 \leq j \leq i$), and $l(G_j)$ has been defined, then:

$$l(a) = l(G_j) + 1.$$

2. For the remaining cases, $l(a) = 0$.

The equation $l(\Sigma) = l(G_n)$ is admitted.

Definition 2.3. Possible minimum number of states of an weak deterministic finite automaton with output (see [6]) which recognizes $N(\Sigma)$, is called the automata complexity of the language transformation schema Σ , and denoted $g(\Sigma)$.

3. THE RESULTS

In this part, we will prove a theorem so that to appraise the automata complexity $g(\Sigma)$ of the language transformation schema Σ which depends on the number of essential vertices $|\Sigma|$ and the depth of operation s .

Firstly, we prove some lemmas.

Lemma 3.1. *For every simple language transformation schema Σ , there exists an weak deterministic finite automaton with the output A , such that:*

$$T(A) = N(\Sigma) \text{ and } |A| \leq 2^{|\Sigma|} + 1.$$

Proof. As Σ is a simple language transform schema, thus, according to [6], it is possible to build an automaton with output, $M = (S, \mathcal{X}, \mathcal{Y}, s_0, \delta, \lambda, F)$ as follows:

- S : the set of states of automaton M , includes all vertex signs of Σ .
- Entry vertex sign of Σ is regarded as initiating state s_0 of automaton M .
- The set of final vertices of Σ is admitted to be a set of final states of M .
- State transitional function $\delta : S \times \Sigma \rightarrow S$ of automaton M is determined: $\forall s \in S, \forall x \in \mathcal{X}$ then $\delta(s, x) = \{s_{i1}, s_{i2}, \dots, s_{it}\} \Leftrightarrow \forall j$ with $1 \leq j \leq t$ then: $x \in N_X(s, s_{ij}), s_{ij} \in S$.
- The output function $\lambda : S \times \mathcal{X} \rightarrow \mathcal{Y}$ of automaton M is determined: $\forall s \in S, \forall x \in \mathcal{X}$ then $\lambda(s, x) = y \in \mathcal{Y}$, wherein y is an element corresponding to x in the pair (x, y) which is written in the edge $(s, \delta(s, x))$ of the simple language transformation schema Σ .
- The input and output alphabet \mathcal{X}, \mathcal{Y} of the simple language transformation schema Σ are considered as the input and output alphabet, respectively, of automaton M .

In this way, it is obviously that $T_X(M) = N_X(\Sigma)$ and $T_Y(M) = N_Y(\Sigma)$. Indeed, we have build an automaton M that recognizes the same language pair with the simple language transformation schema Σ .

With automaton M , using algorithm and determinazing automata program [7], we can build an *weak deterministic finite automaton with output A* that equates to M (i.e. automata A recognizes the same set of word pairs with automaton M). In addition, as the results in [2], then:

$$T(A) = T(M) = N(\Sigma) \text{ and } |A| \leq 2^{|M|} + 1 = 2^{|\Sigma|} + 1.$$

Thus, the lemma is proved. □

Lemma 3.2. *For every simple language transformation schema Σ , there exists another simple schema Σ' , such that:*

$$N(\Sigma') = \mathcal{E}(N(\Sigma)) \text{ and } |\Sigma'| \leq 2|\Sigma|.$$

Proof. 1. Builds schéma Σ_0 : This schema includes two vertices α_0 and α_1 . Vertex α_0 is entry as well as final vertex of Σ_0 . From α_0 to α_1 and conversely from α_1 to α_0 , there are just right n edges on each of which, one of symbols from the dual alphabet $\mathcal{L} = \mathcal{X} \times \mathcal{Y}$ is written, and two different edges are written with two different pairs of symbols. Thus, the schema Σ_0 just right two essential vertices, i.e. $|\Sigma_0| = 2$, and $N(\Sigma_0)$ contains word pairs whose lengths are even.

2. Build schema Σ' : We regard Σ' as an intersection of schemas Σ_0 and Σ , as Σ' is the intersection of simple schemas, according to the results of [6], the conclusion is:

$$N(\Sigma') = N(\Sigma_0) \cap N(\Sigma) \text{ and } |\Sigma'| \leq |\Sigma_0| \cdot |\Sigma| = 2 \cdot |\Sigma|.$$

Obviously, $N(\Sigma')$ is a set of word pairs possessing even lengths.

The lemma has been proved. \square

Lemma 3.3. *For every simple language transformation schema Σ , there exists another simple schema Σ' , such that:*

$$N(\Sigma') = \mathcal{O}(N(\Sigma)) \text{ and } |\Sigma'| \leq 2|\Sigma|.$$

Proof. 1. Builds a schema Σ_1 : This schema is structured the same as Σ_0 , except α_0 is input vertex, whereas α_1 is unique final vertex. From α_0 to α_1 and conversely from α_1 to α_0 , there are just right n edges, on each of which one pair of symbols from the dual alphabet $\mathcal{L} = \mathcal{X} \times \mathcal{Y}$ is written, and two different edges are written with two different pairs of symbols. Hence, the schema Σ_1 has just right two essential vertices, obviously, $|\Sigma_1| = 2$, and $N(\Sigma_1)$ contains word pair whose lengths are odd only.

2. Build up a schema Σ' : We regard Σ' as an intersection of schemas Σ_1 and Σ , as Σ' is the intersection of simple schemas, according to the results of [6], the conclusion is:

$$N(\Sigma') = N(\Sigma_1) \cap N(\Sigma) \text{ and } |\Sigma'| \leq |\Sigma_1| \cdot |\Sigma| = 2 \cdot |\Sigma|.$$

On the other hand, obviously $N(\Sigma')$ is a set of word pairs possessing odd lengths.

The lemma has been proved. \square

Lemma 3.4. *For every simple language transformation schema Σ and with any integer $s > 0$, there exists another simple schema Σ' , such that:*

$$N(\Sigma') = N_s^*(\Sigma) \text{ and } |\Sigma'| \leq s|\Sigma|.$$

Proof. Build up a schema Σ' : Put from left to right a range of language transformation schemas that are structured as same as schema Σ . Nevertheless, their vertices are symbolized variously. Each final vertex of i^{th} -schema ($0 \leq i \leq s-1$) has an empty edge linking with the entry vertex of $i+1^{\text{th}}$ -schema.

The entry vertex of the first schema is considered as the entry of Σ' , and the set of final vertex of s^{th} -schema become the final vertex of Σ' . Obviously:

$$N(\Sigma') = N_s^*(\Sigma) \text{ and } |\Sigma'| \leq s \cdot |\Sigma|.$$

The lemma is proved. \square

Lemma 3.5. *For every simple language transformation schema Σ and with any integer $m > 0$, there exists another simple schema Σ' , such that:*

$$N(\Sigma') = \mathcal{H}(N(\Sigma), m) \text{ v\`a } |\Sigma'| \leq m \cdot |\Sigma|.$$

Proof. 1. Build up a schema Σ_m , including $m+1$ vertices: $\alpha_0, \alpha_1, \dots, \alpha_{m-1}, \alpha_m$. From α_i to α_{i+1} ($0 \leq i \leq m-1$) there are n edges on which different pairs of signs extracted from the dual alphabet \mathcal{L} are written. From α_{m-1} to α_m , there are n edges on which different pairs of signs from \mathcal{L} are written. The vertex α_0 is regarded as the entry, and $\alpha_0, \alpha_1, \dots, \alpha_{m-2}, \alpha_{m-1}$ are acknowledged as final vertices of Σ_m . Thus, schema Σ_m recognizes all possible word pairs on the dual alphabet \mathcal{L} , whose lengths are bounded at m and they have just right m essential vertices.

2. Build up a schema Σ' as an intersection of Σ_m and Σ , of which the pair of vertices containing final vertex of Σ_m is recognized as final vertex of Σ' . It is clearly that Σ' is a simple schema, and according to the results of [6], the conclusion is: $N(\Sigma') = \mathcal{H}(N(\Sigma), m)$, this is a set of all first parts of word pairs whose lengths are at most m , and $|\Sigma'| \leq m \cdot |\Sigma|$.

The lemma is proved. \square

Lemma 3.6. *For any the language transformation schema Σ , of which the depth of operations restricted at s , there exists a simple schema Σ' , such that:*

$$N(\Sigma') = N(\Sigma) \text{ and } |\Sigma'| \leq s^{l(\Sigma)} \cdot |\Sigma|.$$

Proof. The lemma is proved by mathematical induction on to the depth of operations on the schema Σ .

1. If Σ is a simple schema, obviously, we can regard Σ' as Σ , therefore:

$$N(\Sigma') = N(\Sigma) \text{ and } |\Sigma'| = |\Sigma| = s^0 \cdot |\Sigma| = s^{l(\Sigma)} \cdot |\Sigma|.$$

2. Suppose that, $\Sigma = (G_1, G_2, \dots, G_{m-1})$ and a_1, a_2, \dots, a_p are even edges, odd edges or complement edges with degree t , repeated edges with degree u on the graph G_n . As the depth of operations not exceed s , thus $t \leq s$ and $u \leq s$.

Suppose that, a_i ($1 \leq i \leq p$) depends on the graph G_{k_i} . For every i ($1 \leq i \leq p$) it is possible to use all the graphs on which G_{k_i} depends, including G_{k_i} , to build schema Σ_{k_i} . Then we have:

$$N(\Sigma_{k_i}) = N(G_{k_i}).$$

According to the definition of the depth of operations, there is a conclusion:

$$l(\Sigma) = l(G_n) \geq l(a_i) + 1 = l(G_{k_i}) + 1 = l(\Sigma_{k_i}) + 1.$$

Hence:

$$l(\Sigma_{k_i}) \leq l(\Sigma) - 1.$$

To match the definition of induction, for each i ($1 \leq i \leq p$), there exists a simple schema Σ'_{k_i} , such that:

$$N(\Sigma'_{k_i}) = N(\Sigma_{k_i}) \text{ and } |\Sigma'_{k_i}| \leq |\Sigma_{k_i}| \cdot s^{l(\Sigma_{k_i})} \leq |\Sigma_{k_i}| \cdot s^{l(\Sigma)-1}.$$

If a_i is an even edge (or an odd edge), then, in accordance with Lemma 3.2 (or Lemma 3.3) we can build up a simple schema Δ_i , such that:

$$\begin{aligned} N(\Delta_i) &= \mathcal{E}(N(\Sigma'_{k_i})) = \mathcal{E}(N(\Sigma_{k_i})) = \mathcal{E}(N(G_{k_i})) \\ \text{(or } N(\Delta_i) &= \mathcal{O}(N(\Sigma'_{k_i})) = \mathcal{O}(N(\Sigma_{k_i})) = \mathcal{O}(N(G_{k_i})), \text{ respectively)} \end{aligned}$$

with:

$$|\Delta_i| \leq 2|\Sigma'_{k_i}| \leq 2|\Sigma_{k_i}| \cdot s^{l(\Sigma)-1} \leq |\Sigma_{k_i}| \cdot s^{l(\Sigma)}.$$

If a_i ($1 \leq i \leq p$) is the complement edge with degree r (or repeated edge with degree r), then according to Lemma 3.5 (or Lemma 3.4) we can build a simple schema Δ_i , such that:

$$\begin{aligned} N(\Delta_i) &= \mathcal{H}(N(\Sigma'_{k_i}), r) = \mathcal{H}(N(\Sigma_{k_i}), r) = \mathcal{H}(N(G_{k_i}), r) \\ \text{(or } N(\Delta_i) &= N_r^*(\Sigma'_{k_i}) = N_r^*(\Sigma_{k_i}) = N_r^*(G_{k_i}), \text{ respectively)} \end{aligned}$$

with:

$$|\Delta_i| \leq r \cdot |\Sigma'_{k_i}| \leq r \cdot |\Sigma_{k_i}| \cdot s^{l(\Sigma)-1} \leq |\Sigma_{k_i}| \cdot s^{l(\Sigma)}.$$

Replace a_i ($1 \leq i \leq p$) on G_n with schema Δ_i , in accordance with the definitions of substitution (see[6]), we have a simple schema Σ' , such that:

$$N(\Sigma') = N(\Sigma)$$

with:

$$\begin{aligned} |\Sigma'| &= |G_n| + \sum_{1 \leq i \leq p} |\Delta_i| \\ &\leq |G_n| + \sum_{1 \leq i \leq p} |\Sigma_{k_i}| \cdot s^{l(\Sigma)} \\ &\leq |G_n| \cdot s^{l(\Sigma)} + \sum_{1 \leq i \leq p} |\Sigma_{k_i}| \cdot s^{l(\Sigma)} \\ &= \left(|G_n| + \sum_{1 \leq i \leq p} |\Sigma_{k_i}| \right) s^{l(\Sigma)} \\ &= |\Sigma| \cdot s^{l(\Sigma)}. \end{aligned}$$

The lemma is proved. □

Theorem. For any language transformation schema Σ , on which the depth of operations restricted at s , then:

$$g(\Sigma) \leq 2^{|\Sigma| \cdot s^{l(\Sigma)}} + 1.$$

Proof. 1. For the language transformation schema Σ as above, using Lemma 3.6, we can build up a simple schema Σ' , such that:

$$N(\Sigma') = N(\Sigma) \text{ and } |\Sigma'| \leq |\Sigma| \cdot s^{l(\Sigma)}.$$

2. For this simple schema Σ' , using Lemma 3.1, we can build up an weak deterministic finite automaton with output A , such that:

$$T(A) = N(\Sigma') = N(\Sigma), \text{ and:}$$

$$g(\Sigma) = |A| \leq 2^{|\Sigma'|} + 1 \leq 2^{|\Sigma|s^{i(\Sigma)}} + 1.$$

The theorem has been proved. □

REFERENCES

- [1] Arto Saloma., *Computation and Automata* (in Vietnamese), Science Publishers, 1992.
- [2] Dang Huy Ruan, О сложности конечного автомата, соответствующего щего обобщенному регуляриарному выражению, ДАН СССР, Том 213, No. 1 (1973).
- [3] Dominique Perrin, Finite Automata, J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Elsevier Science Publishers B.V., 1990, p. 3–53.
- [4] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, MA, 1979.
- [5] Lung H., “An Algebraic method for Solving decision problems in Finite Automata theory”, Ph.D. Thesis, Penn. State Univ. University Park, PA, 1987.
- [6] Nguyen Van Dinh, “Builds the language transformation schema and analyses its automata complexity”, Master Thesis, Vietnam National University (VNU), Hanoi, 1997.
- [7] Nguyen van Dinh, Solving determinization problems of automata on the computer, *VNU Journal of Science, Nat. Sci.*, **XIV** (1) (1998).
- [8] Peterson J.L., Petri nets, *Computing Surveys* **9** (3) (1997).

Received August 20, 2000

Revised January 10, 2001

United Nations International School-Hanoi