ON INFERENCING DYNAMICS OF AUTOMATA FROM FINITE SAMPLES

By

Seiiti HUZINO

(Received Feb. 17, 1978)

1°. It is well known that techniques of merging and differencing are quite useful for solving automatic program synthesis problems and for determining grammatical inference of formal languages ([1], [2]). These techniques seem to be also useful for inferencing dynamics of automata from finite data, since differences among automata, programs and grammars only lie in considering these as either fixed mechanism or variable operations to hardwares. In this paper we shall seek how inference of dynamics of automata can be done by using merging technique and differencing one to a given data.

2°. In studying the problem, we restrict our considerations to the class of regular or context-free languages. Assume in the following that languages on consideration contain infinite words in it. First we have the following result:

THEOREM 2.1. *Let X be an alphabet and L a language over X . Assume that there exists a family of monotone increasing regular languages $\{L_m\}_{m=1,2,\dots}$ such that $L_m \subset L_{m+1}$ ($m=1, 2, \dots$) and $L = \bigcup_{m=1}^{\infty} L_m$. And assume also that there exist two words u and v in X^* with the property that $\partial_u L_{m+1} = \partial_v L_m$ ($m=1, 2, \dots$). Then the relation $\partial_u L = \partial_v L$ holds.*

PROOF. From the assumptions of the family $\{L_m\}_{m=1,2,\dots}$, we have

$$\begin{aligned}\partial_u L &= \partial_u \left(\bigcup_{m=1}^{\infty} L_m \right) = \bigcup_{m=1}^{\infty} \partial_u L_m = \bigcup_{m=2}^{\infty} \partial_u L_m \\ &= \bigcup_{m=2}^{\infty} \partial_v L_{m-1} = \bigcup_{m=1}^{\infty} \partial_v L_m = \partial_v \left(\bigcup_{m=1}^{\infty} L_m \right) \\ &= \partial_v L.\end{aligned}$$

3°. Although the above result is obtained trivially, it is quite useful for

solving dynamics-inference problems from finite data as the following examples show:

EXAMPLE 3.1. Let X be an alphabet $\{a, b\}$ and L a language $\{(ab)^n \mid n=1, 2, \dots\}$, i.e., $\{ab, abab, \dots, \overbrace{abab \cdots ab}^n, \dots\}$. Let L_m be a language $\{(ab)^n \mid n=1, 2, \dots, m\}$ ($m=1, 2, \dots$). Figure 3.1 shows a relationship holding among derivative languages of the family $\{L_m\}_{m=1,2,\dots}$, that is,

$$\partial_{aba}L_{m+1} = \partial_a L_m \quad (m=1, 2, \dots).$$

Fig. 3.1.

Thus, from theorem 2.1, we have

$$\partial_{aba}L = \partial_a L.$$

Of course the above result can be ascertained by directly calculating the derivative languages $\partial_{aba}L$ and $\partial_a L$, and by comparing these results.

EXAMPLE 3.2. Let X be an alphabet $\{a, b\}$, and L a language $\{a^n b \mid n=1, 2, \dots\} + \{b^n a \mid n=1, 2, \dots\}$. Let L_m be a language $\{a^n b \mid n=1, 2, \dots, m\} + \{b^n a \mid$

Fig. 3.2.

$$L \xrightarrow{\partial_a} \partial_a L \xrightleftharpoons[\partial_a]{\partial_b} \partial_{ab} L$$

Fig. 3.3.

$n = 1, 2, \dots, m\}$ ($m = 1, 2, \dots$). Then a relationship is shown among each derivative languages of the family $\{L_m\}_{m=1,2,\dots}$ as in figure 3.2.

That is,

$$\begin{cases} \partial_{aa}L_{m+1} = \partial_a L_m, \\ \partial_{ab}L_{m+1} = \partial_{ab}L_m, \\ \partial_{ba}L_{m+1} = \partial_{ba}L_m, \\ \partial_{bb}L_{m+1} = \partial_b L_m, \end{cases} \quad (m = 1, 2, \dots)$$

Therefore we get from the first and the fourth equations

$$\partial_{aa}L = \partial_a L,$$

and

$$\partial_{bb}L = \partial_b L.$$

The fact also can be directly ascertained by computing derivative-languages of L itself. Tending m to ∞ , we obtain from the diagrams of figures 3.1 and 3.2 the following structure-diagrams of the original languages as shown in figures 3.3 and 3.4, respectively.

Fig. 3.4.

These structure-diagrams determine the following dynamics of automata recognizing the original languages, respectively (Fig. 3.5, and 3.6), by using so called 'H-transformation' to the diagrams ([3]).

Fig. 3.5.

Fig. 3.6.

Now look at the neighbouring part of L_m in figure 3.1, m being fixed. Since $\partial_a L_m \supset \partial_a L_{m-1}$, it follows that the set of words accepted by the automaton with the state corresponding to $\partial_a L_m$ as an initial state, which is derived by the H -transformation, contains the set of words accepted by another automaton with the initial state corresponding to $\partial_a L_{m-1}$ as a derivative language of the former. If an operation to a state can substitute for an operation to another state, we merge the former state with the latter on the way of dynamics-inference. We call this identification by 'merging'. In the present case, $\partial_a L_{m-1}$ obtained from $\partial_{ab} L_m$ by ∂_a -derivation is merged into $\partial_a L_m$, so we get the diagram which formally coincides with the figure 3.3. Similarly, in the neighbouring part of L_m in figure 3.2, we formally get the diagram (figure 3.4) by merging $\partial_a L_{m-1}$ with $\partial_a L_m$, and $\partial_b L_{m-1}$ with $\partial_b L_m$, respectively.

4°. From this we can define a technique of merging on inference problems of automata-dynamics as follows: If a finite set D of words in a language L is given, and if, on the way of construction of structure-diagram of D , there exists two words u and v in X^* such that $\partial_u D \supset \partial_v D$, then we put the positions of $\partial_u D$ and $\partial_v D$ together. We call an automaton, obtained by H -transformation from the diagram constructed by applying the merging techniques (if possible) to the sample set D , the first inference of an automaton recognizing the given language L . In the case when the structure of the language L is simple, the first inference, thus obtained, may often coincides with an automaton recognizing L .

Let us apply this merging technique in the case when the language L is regular.

EXAMPLE 4.1. Let X be an alphabet $\{a, b\}$, and L a language $\{a^{2n+1}b \mid n=0, 1, 2, \dots\}$. Let us take a set of finite samples D a language $\{ab, aaab\}$. Then the following diagram of D is obtained (Figure 4.1), where the position of $\{ab\}$ is a merging place with the position of D , because of the relation $\{ab\} \subset D$.

Fig. 4.1.

Thus, by applying a merging technique, we get the following diagram as in figure 4.2.

$$L \begin{matrix} \xrightarrow{\partial_a} \\ \xleftarrow{\partial_a} \end{matrix} \partial_a L \xrightarrow{\partial_b} \partial_{ab} L$$

Fig. 4.2.

Here we rewrite in place of the sample set name D by the original language name L . The diagram coincides with the structure-diagram of L . Therefore, by H -transformation, we get an automaton, which recognizes L and is also minimal (Figure 4.3).

Fig. 4.3.

EXAMPLE 4.2. (A. W. Biermann) Let X be an alphabet $\{a, b\}$, and L a language $\{ab, bb, aba, bba, \dots\}$. The language L is the one recognized by the automaton as shown in figure 4.4. Let us take a set of finite samples D a language $\{ab, bb, aba, aaab, abbab\}$. Then, on the way of inferencing structure diagram of L , we can apply a merging technique to the derivative languages of D . (Figure 4.5).

Fig. 4.4.

Fig. 4.5.

Thus, by rewriting D by L , we get the following inferred structure diagram of L (Figure 4.6).

Fig. 4.6.

Clearly, the first inference of the automaton obtained from this by H -transformation coincides with the automaton in figure 4.4.

Notice that appearances of applications of merging techniques mainly depend on the choice of D .

5°. Now, we shall show that a “differencing” technique may be useful for

inferencing dynamics of pushdown automata when a given language L is context-free. Let us show it by example:

EXAMPLE 5.1. Let X be an alphabet $\{a, b\}$, and L a language $\{a^n b^n \mid n=1, 2, \dots\}$. Let us take L_m a finite language $\{a^n b^n \mid n=1, 2, \dots, m\}$ ($m=1, 2, \dots$). The relationship among the derivative languages of the family $\{L_m\}_{m=1,2,\dots}$ are shown as in figure 5.1, that is,

$$\begin{cases} \partial_{ab} L_m = \partial_{ab} L_{m-1}, \\ \partial_{aab} L_m = \partial_{aab} L_{m-1}, \\ \partial_{a^{m-1}b} L_m = \partial_{a^{m-1}b} L_{m-1}, \\ \partial_{a^m b} L_m = \partial_{a^{m-1}b} L_{m-1}, \end{cases} \quad (m=2, 3, 4, \dots).$$

Fig. 5.1.

If we only extract the diagram of L_m in the figure, then we get a diagram as shown in figure 5.2.

Fig. 5.2.

As we can easily see, by formally adding to the structure diagram of L_{m-1} the last part ringed round by a dotted curve, we formally get the external form of the structure diagram of L_m . We call this part a differencing part, the occurrence of such a part lying on the effect of so called 'u v w x y' theorem. By finding out which part is a differencing one, and by continuing infinitely the operations of adding the part, we finally get the inferred structure diagram of L . In this case the inferred one just meets with the original one (Figure 5.3), which is easily ascertained.

Fig. 5.3.

We have known that we can construct a pushdown automaton recognizing the language L by using such diagram ([3]). This is only an example. However we think that such differencing techniques could be effectively used on considering inference problems of automata dynamics.

References

- [1] A. W. BIERMANN: *Approaches to automatic programming*, Advances in Computers, **15** (1976), 1-63.
- [2] K. S. FU and T. L. BOOTH: *Grammatical inference: Introduction and survey-Part I*, IEEE Trans. System, Man and Cybernetics, SMC-5, No. 1 (1975), 95-111.
- [3] S. IMORI and S. HUZINO: *On relationship between structure diagrams of context-free languages and mechanisms of pushdown automata*, Abstracts in autumn meeting of mathematical society of Japan (1976), 166-171.