ГОДИШНИК НА СОФИЙСКИЯ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ"

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
Книга 1 — Математика и механика
Том 91, 1997

ANNUAIRE DE L'UNIVERSITE DE SOFIA „ST. KLIMENT OHRIDSKI"

FACULTE DE MATHEMATIQUES ET INFORMATIQUE
Livre 1 — Mathématiques et Mecanique
Tome 91, 1997

DIRECT BUILDING OF MINIMAL AUTOMATON FOR A GIVEN LIST

STOYAN MIHOV

This paper presents a method for direct building of minimal acyclic finite states automaton which recognizes a given finite list of words in lexicographical order. The size of the temporary automata which are necessary for the construction is less than the size of the resulting minimal automata plus the length of one of the longest words in the list. This property is the main advantage of our method.

Keywords: acyclic automata, automata building
1991/95 Math. Subject Classification: 68Q68

1. INTODUCTION

The standard methods for building minimal finite states (FS) automaton are building temporary automata which generally are huge compared to the resulting minimal automaton. This grounds the interest in the development of more direct methods. Building an acyclic minimal FS automaton that recognizes a given list sorted in lexicografical order is of special interest for practical applications. A linear algorithm for that case is presented in [1, 2]. The Revuz' method in the first stage builds a tree-like deterministic FS automaton. Then at the second stage this automaton is minimized efficiently. The drawback of this method is that the tree-like automaton is huge in respect of the resulting minimal automata. To make this method more efficient, Roche [3] proposes to divide the list into parts for which to build the corresponding minimal automata. After that those automata are united. At the end it is necessary to minimize the result.

It is claimed [1] that a method for direct building of minimal FS automaton for a given list does not exist. We will show that in general this statement is not valid. We shall present our new method for direct building of minimal automaton.

2. FORMAL BACKGROUND AND NOTATIONS

Definition 1. A deterministic FS automaton is a tuple $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$, where:

- Σ is a finite alphabet;
- S is a finite set of states;
- $s \in S$ is the starting state;
- $F \subseteq S$ is the set of final states;
- $\mu : S \times \Sigma \to S$ is a partial function called the transition function.

The function μ is extended naturally over $S \times \Sigma^*$ by induction:

$$\begin{cases} \mu^*(r, \varepsilon) = r, \\ \mu^*(r, \sigma a) = \begin{cases} \mu(\mu^*(r,\sigma), a), & \text{in case } \mu^*(r,\sigma) \text{ and } \mu(\mu^*(r,\sigma),a) \text{ are defined,} \\ \text{not defined,} & \text{otherwise,} \end{cases} \end{cases}$$

where $r \in S$, $\sigma \in \Sigma^*$, $a \in \Sigma$.

We shall work with a definition of FS automata with a partial transition function. The only difference from the definition with a total transition function is the absence of the necessity to introduce a dead state (a state r, for which $\forall a \in \Sigma\ (\mu(r,a) = r)$). Later we use $!\mu(r,\sigma)$ to denote that $\mu(r,\sigma)$ is defined and when writing $\mu^*(r,\sigma) \cong x$, we mean $!\mu(r,\sigma)\ \&\ \mu(r,\sigma) = x$.

Definition 2. Let $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ be a deterministic FS automaton. Then the set $L(\mathcal{A}) \subseteq \Sigma^*$, defined as

$$L(\mathcal{A}) = \{\sigma \in \Sigma^* \mid !\mu^*(s,\sigma)\ \&\ \mu^*(s,\sigma) \in F\},$$

is called the language of the automaton $\mathcal{A}$ or the language recognized by $\mathcal{A}$.

Two automata $\mathcal{A}$ and $\mathcal{A}'$ are called equivalent when $L(\mathcal{A}) = L(\mathcal{A}')$. An automaton is called acyclic when $\forall r \in S\ \forall \sigma \in \Sigma^+\ (\mu^*(r,\sigma) \ncong r)$. The language of an acyclic FS automaton is finite.

Definition 3. Let $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ be a deterministic FS automaton:

1. The state $r \in S$ is called reachable from $t \in S$ when $\exists \sigma \in \Sigma^*\ (\mu^*(t,\sigma) \cong r)$.

2. We define the subautomaton starting in $s' \in S$ as $\mathcal{A}|_{s'} = \langle \Sigma, S', s', F \cap S', \mu|_{S' \times \Sigma} \rangle$, where $S' = \{r \in S \mid r \text{ is reachable from } s'\}$.

3. Two states $s_1, s_2 \in S$ are called equivalent when $L(\mathcal{A}|_{s_1}) = L(\mathcal{A}|_{s_2})$.

Definition 4. The deterministic FS automaton $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ with language $L(\mathcal{A})$ is called minimal (with language $L(\mathcal{A})$) when for every other deterministic FS automaton $\mathcal{A}' = \langle \Sigma, S', s', F', \mu' \rangle$ with language $L(\mathcal{A}') = L(\mathcal{A})$, it holds that $|S| \leq |S'|$.

From the classical FS theory the next theorem is well-known.

Theorem 5. *A deterministic FS automaton with non-empty language is minimal if and only if every state is reachable from the starting state, from every state a final state is reachable and there are no different equivalent states. There exists an unique (up to isomorphism) minimal automaton for a given language.*

3. METHOD DESCRIPTION

Further we assume that a finite alphabet Σ is given and there is a linear order in Σ. This order induces a lexicographical order in Σ^*.

Definition 6. Let $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ be an acyclic deterministic FS automaton with language $L(\mathcal{A})$. Then the automaton $\mathcal{A}$ is called *minimal except for the word* $\omega \in \Sigma^*$ when the following conditions hold:

1. Every state is reachable from the starting state and from every state a final state is reachable.

2. ω is a prefix of the last word in the lexicographical order of $L(\mathcal{A})$.

In that case we can introduce the notations

$$\omega = w_1^{\mathcal{A}} w_2^{\mathcal{A}} \dots w_k^{\mathcal{A}}, \quad \text{where } w_i^{\mathcal{A}} \in \Sigma \text{ for } i = 1, 2, \dots, k, \tag{1}$$

$$t_0^{\mathcal{A}} = s;\;\; t_1^{\mathcal{A}} = \mu(t_0^{\mathcal{A}}, w_1^{\mathcal{A}});\;\; t_2^{\mathcal{A}} = \mu(t_1^{\mathcal{A}}, w_2^{\mathcal{A}});\;\; \dots;\;\; t_k^{\mathcal{A}} = \mu(t_{k-1}^{\mathcal{A}}, w_k^{\mathcal{A}}), \tag{2}$$

$$T = \{t_0^{\mathcal{A}}, t_1^{\mathcal{A}}, \dots, t_k^{\mathcal{A}}\}. \tag{3}$$

3. In the set $S \setminus T$ there are no different equivalent states.

4. $\forall r \in S\, \forall i \in \{0, 1, \dots, k\} \forall a \in \Sigma \left(\mu(r, a) \cong t_i \leftrightarrow (i > 0 \& r = t_{i-1} \& a = \omega_i^{\mathcal{A}})\right)$.

Further, when working with minimal except for a given word automaton, we use the notations (1)–(3). In case the notation is not ambiguous, we write t_i, w_i instead of $t_i^{\mathcal{A}}$, $w_i^{\mathcal{A}}$. Clearly, if an automaton is minimal except for two different words, one is a prefix of the other.

Proposition 7. *Let the automaton* $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ *be minimal except for* ω. *Then*:

1. $\forall r \in S \setminus T\; \forall a \in \Sigma\; (!\mu(r, a) \to \mu(r, a) \in S \setminus T)$;

2. $\mu^*(s, \sigma) \cong t_i \to \sigma = w_1 w_2 \dots w_i$.

The proof of this proposition is derived directly from Definition 6.

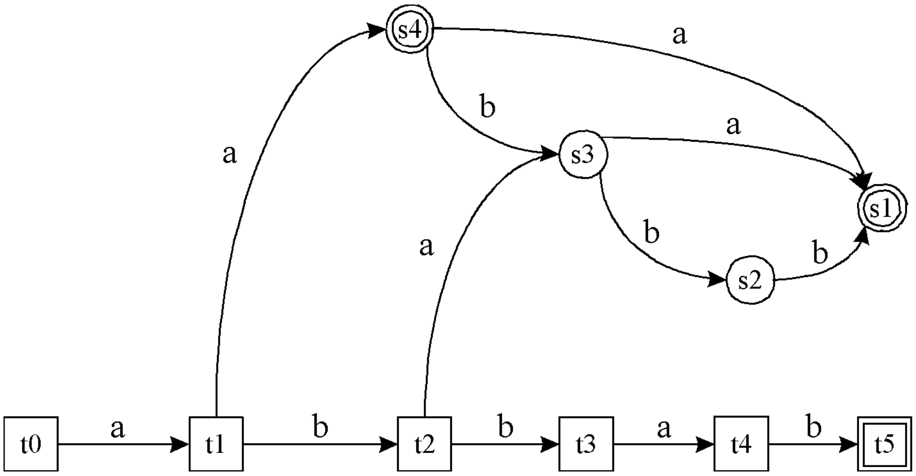

Fig. 1. The FS automaton minimal except for $abbab$

Example 1. On Fig. 1 an acyclic FS automaton over the alphabet $\{a, b\}$ is given. The language of the automaton is $\{aa, aaa, aaba, aabbb, abaa, ababb, abbab\}$. This automaton is minimal except for $abbab$.

Proposition 8. *An automaton which is minimal except for the empty word ε is minimal.*

Proof. Every state is reachable from the starting state and from every state a final state is reachable. Hence, to prove that the automaton is minimal, we have to show that there are no equivalent states. From the definition we know that there are no equivalent states in $S \setminus \{s\}$.

Let assume that $r \in S$ and s are equivalent and $r \neq s$. Let ω be one of the longest words recognized by the automaton. There exists longest word(s), because the language is finite. The states r and s are equivalent, hence $\omega \in L(\mathcal{A}|_r)$. The state r is reachable from s. Hence there exists $\sigma \in \Sigma^*$ and $\mu^*(s, \sigma) \cong r$. Then the word $\sigma\omega \in L(\mathcal{A})$ and from $\sigma \neq \varepsilon$ we have that $|\sigma\omega| > |\omega|$. This contradicts with the fact that ω is the longest word in the language. $\square$

Lemma 9. *Let the automaton $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ be minimal except for $\omega = w_1 w_2 \ldots w_k$, $\omega \neq \varepsilon$. Let there be no state equivalent to t_k in the set $S \setminus T$. Then $\mathcal{A}$ is also minimal except for the word $\omega' = w_1 w_2 \ldots w_{k-1}$.*

Proof. We have to check the conditions of Definition 6. The conditions 1 and 2 are obviously satisfied. Condition 3 follows from the fact that in $S \setminus \{t_0, t_1, \ldots, t_k\}$ there are no states equivalent to t_k. Condition 4 follows also directly from condition 4 of the definition for minimality except for ω. $\square$

Lemma 10. *Let the automaton $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ be minimal except for*

$\omega = w_1 w_2 \dots w_k$, $\omega \neq \varepsilon$. *Let the state* $p \in S \setminus T$ *be equivalent to the state* t_k. *Then the automaton* $\mathcal{A}' = \langle \Sigma, S', s, F', \mu' \rangle$ *defined as follows:*

$$S' = S \setminus \{t_k\},$$

$$F' = F \setminus \{t_k\},$$

$$\mu'(r,a) = \begin{cases} \mu(r,a), & \textit{in case } r \neq t_{k-1} \vee a \neq w_k \textit{ and } \mu(r,a) \textit{ is definied,} \\ p, & \textit{in case } r = t_{k-1},\ a = w_k, \\ \textit{not defined,} & \textit{otherwise,} \end{cases}$$

is equivalent to the automaton $\mathcal{A}$ *and is minimal except for the word* $\omega' = w_1 w_2 \dots w_{k-1}$.

Proof. The automaton $\mathcal{A}$ is equivalent to $\mathcal{A}'$, because the new automaton is derived from the old one by removing the state t_k, and the only transition to t_k (refer to Proposition 7) is exchanged with a transition to an equivalent state. Conditions 1–3 from the definition for minimality except for ω' are trivially satisfied. Condition 4 is obviously satisfied for $t_0, t_1, \dots, t_{k-2}$, and holds also for t_{k-1}, because $\mu'(t_{k-1}, w_k) \cong p \in S \setminus T$. $\square$

Theorem 11. *Let the automaton* $\mathcal{A} = \langle \Sigma, S, s, F, \mu \rangle$ *be minimal except for* $\omega' = w_1 w_2 \dots w_m$. *Let* $\psi \in L(\mathcal{A})$ *be the last word in the lexicographical order of the language of the automaton. Let* ω *be a word which is greater in lexicographical order than* ψ. *Let* ω' *be the longest common prefix of* ψ *and* ω. *In that case we can denote* $\omega = w_1 w_2 \dots w_m w_{m+1} \dots w_k$, $k > m$. *Then the automaton* $\mathcal{A}' = \langle \Sigma, S', s, F', \mu' \rangle$ *defined as follows:*

$$t_{m+1}, t_{m+2}, \dots, t_k \textit{ are new states such that } S \cap \{t_{m+1}, t_{m+2}, \dots, t_k\} = \emptyset,$$

$$S' = S \cup \{t_{m+1}, t_{m+2}, \dots, t_k\},$$

$$F' = F \cup \{t_k\},$$

$$\mu'(r,a) = \begin{cases} t_{m+1}, & \textit{in case } r = t_m,\ a = w_{m+1}, \\ \mu(r,a), & \textit{in case } r \in S,\ !\mu(r,a) \textit{ and } r \neq t_m \vee a \neq w_{m+1}, \\ t_{i+1}, & \textit{in case } r = t_i,\ m+1 \leq i \leq k-1,\ a = w_{i+1}, \\ \textit{is not defined,} & \textit{otherwise,} \end{cases}$$

is minimal except for ω *and recognizes the language* $L(\mathcal{A}) \cup \{\omega\}$.

Proof. First we shall show that $L(\mathcal{A}') = L(\mathcal{A}) \cup \{\omega\}$. We have to show that the automaton $\mathcal{A}'$ includes $\mathcal{A}$. Clearly, $S' \supset S$ and $F' \supset F$. We have to check that $\mu' \supset \mu$. Considering the definition of μ', it is clear that the only problem could be the case $\mu'(t_m, w_{m+1})$. But $\mu(t_m, w_{m+1})$ is not defined, because otherwise either ω' could not be the longest common prefix of ψ and ω or ω could not be greater in lexicographical order than ψ — the last word in the lexicographical order of the language of the automaton.

Hence we have that $L(\mathcal{A}') \supset L(\mathcal{A})$. New words could be recognized only by passing the additional new states. From the definition of μ' we have that they are reachable only from t_m. The only new word in the language $L(\mathcal{A}'|_{t_m})$ is $w_{m+1}w_{m+2}\dots w_k$. From Proposition 7 we have that t_m is reachable from s only by the word ω'. Hence the only new word in $L(\mathcal{A}')$ is the word $\omega' w_{m+1}w_{m+2}\dots w_k = \omega$.

We have to check that $\mathcal{A}'$ is minimal except for ω. Let us consider the conditions of Definition 6. Conditions 1–3 are obviously satisfied. Condition 4 is satisfied for $t_0, t_1, \dots, t_{m-1}$ from the definition of minimality except for ω' of the automaton $\mathcal{A}$, for the states $t_m, t_{m+1}, \dots, t_k$ the condition clearly holds because of the definition of μ'. $\square$

Method for direct building of minimal FS automaton for a given list. Let a non-empty finite list of words L in lexicographical order be given. Let $\omega^{(i)}$ denote the i-th word of the list. We start with the minimal automaton which recognizes only the first word of the list. This automaton can be built trivially and it is also minimal except for $\omega^{(1)}$. Using it as basis, we carry out an induction on the words of the list. Let us assume that the automaton $\mathcal{A}^{(n)} = \langle \Sigma, S, s, F, \mu \rangle$ with language $L^{(n)} = \{\omega^{(i)} \mid i = 1, 2, \dots, n\}$ has been built and that $\mathcal{A}^{(n)}$ is minimal except for $\omega^{(n)}$. We have to build the automaton $\mathcal{A}^{(n+1)}$ with language $L^{(n+1)} = \{\omega^{(i)} \mid i = 1, 2, \dots, n+1\}$ which is minimal except for $\omega^{(n+1)}$.

Let ω' be the longest common prefix of the words $\omega^{(n)}$ and $\omega^{(n+1)}$. Using several times Lemma 9 and Lemma 10 (corresponding to the actual case), we build the automaton $\mathcal{A}' = \langle \Sigma, S', s, F', \mu' \rangle$ which is equivalent to $\mathcal{A}^{(n)}$ and is minimal except for ω'. Now we can use Theorem 11 and build the automaton $\mathcal{A}^{(n+1)}$ with language $L^{(n+1)} = L^{(n)} \cup \{\omega^{(n+1)}\} = \{\omega^{(i)} \mid i = 1, 2, \dots, n+1\}$ which is minimal except for $\omega^{(n+1)}$.

In this way by induction we build the minimal except for the last word of the list automaton with language the list L. At the end, using again Lemma 9 and Lemma 10, we build the automaton equivalent to the former one which is minimal except for the empty word. From Proposition 8 we have that it is the minimal automaton for the list L. The check of state equivalence needed to distinguish between Lemma 9 and Lemma 10 is performed efficiently using the following property:

$$
\begin{aligned}
& t_k \text{ is equivalent to } r \in S \setminus T \\
\longleftrightarrow\ & \big((t_k \in F \leftrightarrow r \in F) \,\&\, \forall a \in \Sigma((\neg!\mu(t_k, a) \,\&\, \neg!\mu(r, a)) \\
& \qquad \vee (!\mu(t_k, a) \,\&\, !\mu(r, a) \,\&\, \mu(t_k, a) = \mu(r, a))\big).
\end{aligned}
$$

$\square$

Clearly, all the temporary automata built during the construction of the resulting minimal automaton have less states than the resulting automaton plus the size of the longest word of the list. This is the main advantage of our method.

Example 2. To illustrate our method, let us consider the following example. On Fig. 1 the automaton recognizing the list $\{aa, aaa, aaba, aabbb, abaa, ababb,$

abbab} is given. This automaton is minimal except for the last word of the list — *abbab*. Let the next word be *baa*. The longest common prefix of those two words is ε. We have first to construct the automaton equivalent to the one on Fig. 1, which is minimal except for ε, by using Lemma 9 and Lemma 10. First we have to apply Lemma 9 twice and then to apply Lemma 10 three times. At the end, using Theorem 11, we construct the automaton which is minimal except for *baa*, given on Fig. 2. In this way we added the next word of the list to the language of the temporary automaton.

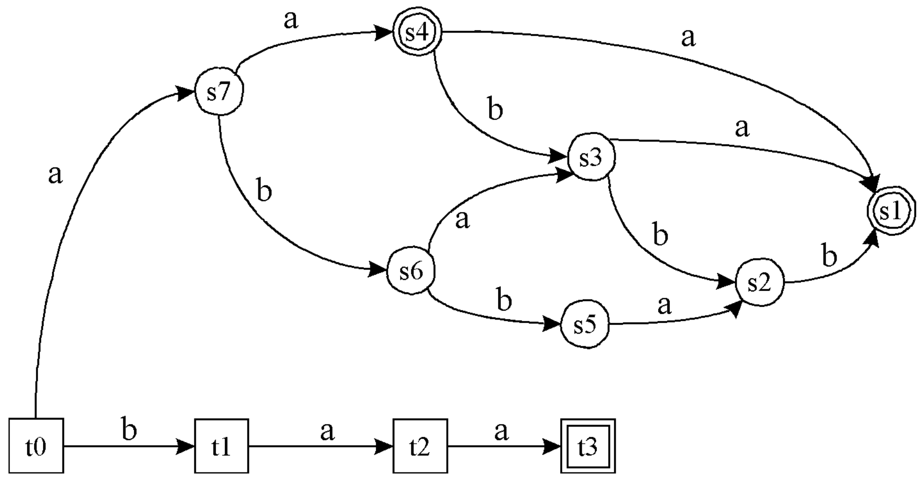

Fig. 2. The FS automaton minimal except for *baa*

4. CONCLUSION

The presented method for direct building of minimal automata can be extended for building minimal automata with labels on the final states and for automata which are returning the index of the recognized word in the list (for a presentation of those kinds of automata see, for example, [3]. In this way the method could be widely applied for building of large grammatical dictionaries, for indexing of huge lists, etc. The algorithms based on the method are distinguished with an excellent memory efficiency.

ACKNOWLEDGEMENTS. I would like to thank to professor Dimiter Skordev and Anton Zinoviev for the very valuable remarks on the presentation of our method, and to Bogdan Alexandrov who is a co-author of the idea for the presented method.

REFERENCES

1. Dominique Revuz. Dictionaires et lexiques, méthodes et algorithmes. Ph.D. dissertation, Université Paris 7, Paris, 1991.

2. Dominique Revuz. Minimization of acyclic deterministic automata in linear time. *Theoretical Computer Science*, **92**, 1, 1992.
3. Emmanuel Roche. Dictionary Compression Experiments. Technical report, LADL, Université Paris 7, Paris, 1992.

Received February 26, 1998

Linguistic Modelling Laboratory
LPDP — Bulgarian Academy of Sciences
E-mail: `stoyan@lml.acad.bg`