Computing Small Nondeterministic Finite Automata

Oliver Matz, Andreas Potthoff
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität Kiel
D-24098 Kiel
e-mail: oma@informatik.uni-kiel.de

Abstract

We study the minimization problem for nondeterministic finite automata. Two approaches are discussed, based on the construction of two versions of "canonical" automata (for a given regular language), in which minimal automata occur as subautomata. We introduce a heuristic for this search, which has been implemented in the program AMoRE.

Introduction

Minimization of nondeterministic finite automata ("NFA") is more difficult than that of deterministic finite automata. The problem is PSPACE-complete (except for the case of one-letter-alphabets) [6], whereas deterministic finite automata can be minimized in time $O(n \cdot \log n)$ [4]. Moreover, there is in general not a unique minimal NFA recognizing a given regular language. Procedures for minimization of nondeterministic finite automata are investigated in several papers, e.g. Kameda and Weiner [10] (see also Indermark [7]) and Kim [11]. Further references are given in Brauer's monograph [2].

The purpose of this note is twofold. First, we describe a general technique of constructing minimal NFA which is implicit in older papers like [10] and has been suggested in a different version in a recent contribution of Arnold, Dicky and Nivat [1]. The idea is to construct a nondeterministic "canonical automaton" in which any NFA recognizing the considered language occurs (via a homorphism) as a subautomaton. This means that the construction of the minimal NFA can be done in two steps: construction of the canonical automaton and search for a minimal subautomaton therein which accepts the same language. We compare two versions of "canonical automata" and develop a method of computing one of these versions (called "fundamental automaton").

The second issue is the search of a minimal NFA within a given (canonical) automaton. We outline a heuristic which improves the one of [11]. It yields an algorithm with worst-case-behaviour time $\mathcal{O}(2^{(n^2)})$, where n is the number of states of the minimal deterministic automaton of the considered language. The algorithm is practical in the sense that in many non-trivial examples the actual implementation works in acceptable time. However, we show that the heuristic fails in some cases to produce a minimal NFA. (This is true for both considered versions of canonical NFA.) It is open how to sharpen the heuristic in order to get an algorithm which always outputs a proper minimal NFA and is as "practical" as our heuristic.

In the first section basics and terminology on finite automata are recalled (powerset construction, minimization of deterministic finite automata). In the second section we construct the "fundamental automaton" $\mathcal{F}$ of a regular language L and show that a minimal NFA accepting L can be found as a subautomaton of $\mathcal{F}$. Then we give the announced sufficient (but not necessary) criterion for subautomata of $\mathcal{F}$ that accept L; the heuristic is then simply to find a subautomaton of $\mathcal{F}$ that satisfies this criterion and has a minimal number of states. Section 3 offers one of the rare examples where this heuristic fails to yield a minimal automaton.

Section 4 outlines an implementation for the described strategy.

In the last section, we give a different definition for the fundamental automaton in order to compare it to the "canonical automaton" as introduced in [1] (which contains as well any minimal NFA as subautomaton). We show that the fundamental and the canonical automaton may be different, and transfer the criterion mentioned above to the canonical automaton.

An implementation (written in C) of the heuristic, based on the fundamental automaton, is part of the AMoRE program (computing Automata, MOnoids, and Regular Expressions), which is available via FTP (ftp.informatik.uni-kiel.de:pub/kiel/amore).

We thank Wolfgang Thomas for helpful comments on previous versions of the paper.

1 Preliminaries

We use the standard notation of set theory and formal language theory. For a set P of sets we write $\bigcap P$ for $\bigcap_{p \in P} p$. We denote by $|Q|$ the number of elements of Q. $\emptyset$ is the empty set and 2^Q is the set of all subsets of Q. $\biguplus$ denotes a union of disjoint sets. By Σ we indicate a finite alphabet, by $a, b, c, \ldots$ letters and by $u, v, \ldots$ words. The empty word is written ϵ.

A *nondeterministic finite automaton (NFA)* is of the form $\mathcal{A} = (Q, \Sigma, I, \delta, F)$ with a finite non-empty set of states Q, a finite non-empty alphabet Σ, a non-empty set of initial states $I \subseteq Q$, a transition function $\delta : Q \times \Sigma \longrightarrow 2^Q$ and a set of final states $F \subseteq Q$. If $|I| = 1$ and $\forall q \in Q \quad \forall a \in \Sigma \quad |\delta(q, a)| = 1$, then $\mathcal{A}$ is a *deterministic finite automaton (DFA)*. A triple $(q, a, q') \in Q \times \Sigma \times Q$ with $q' \in \delta(q, a)$ is called a *transition*.

The transition function $\delta : Q \times \Sigma \longrightarrow 2^Q$ can be extended in a natural way to a function from $2^Q \times \Sigma^*$ to 2^Q, usually also denoted by δ. We write $p \underset{\mathcal{A}}{\overset{w}{\rightarrow}} q$ to indicate that $q \in \delta(p, w)$ and say that there exists a *path* from p to q labelled w.

If the components of an NFA $\mathcal{A}$ are not listed explicitly, we will denote by $Q_\mathcal{A}$, $I_\mathcal{A}$, $F_\mathcal{A}$ and $\delta_\mathcal{A}$ its set of states (initial states, final states resp.) and its transition function. We will drop the subscripts only if the NFA is clear from the context.

A word $u \in \Sigma^*$ is *accepted by* $\mathcal{A}$, if $\delta(I, u) \cap F \neq \emptyset$. The *language accepted by* $\mathcal{A}$ is $L(\mathcal{A}) = \{ u \in \Sigma^* \mid \delta(I, u) \cap F \neq \emptyset \}$. Two NFA $\mathcal{A}$ and $\mathcal{B}$ are called *equivalent* if $L(\mathcal{A}) = L(\mathcal{B})$. An NFA $\mathcal{A}$ is called *minimal* if all NFA that are equivalent to $\mathcal{A}$ have at least as many states as $\mathcal{A}$.

We assign two languages to every state of an NFA.

Definition 1.1 *([10],[5])* Let $\mathcal{A}$ be an NFA and $q \in Q$. Then the *pre-language of* q and the *post-language of* q are given by $pre(\mathcal{A}, q) = L((Q_{\mathcal{A}}, \Sigma, I_{\mathcal{A}}, \delta_{\mathcal{A}}, \{q\}))$ and by $post(\mathcal{A}, q) = L((Q_{\mathcal{A}}, \Sigma, \{q\}, \delta_{\mathcal{A}}, F_{\mathcal{A}}))$, respectively. q is *reachable* if $pre(\mathcal{A}, q) \neq \emptyset$ and q is *productive* if q is reachable and $post(\mathcal{A}, q) \neq \emptyset$. Two states $p, q \in Q$ are *equivalent*, written $p \sim q$, if $post(\mathcal{A}, p) = post(\mathcal{A}, q)$. $[q] = \{\, p \mid p \sim q \,\}$ denotes the equivalence class to which q belongs.

In [1], pre- and post-language are called *history* and *prophecy*.
In the sequel all NFA have productive states only.
Now we introduce three classical operations on automata and analyze pre- and post-languages of the resulting automata.

Definition 1.2 *([10])* Let $\mathcal{A}$ be an NFA. Its *reversed NFA* is $\overline{\mathcal{A}} = (Q, \Sigma, F, \overline{\delta}, I)$ with $\forall a \in \Sigma \ \forall p, q \in Q : \ p \in \overline{\delta}(q, a) \Longleftrightarrow q \in \delta(p, a)$.
Given a word $u = a_1 \ldots a_n$ the word $\overline{u} = a_n \ldots a_1$ is its *reversed word.* Given a language L the language $\overline{L} = \{\, \overline{u} \mid u \in L \,\}$ is its *reversed language.*

Obviously $L(\overline{\mathcal{A}}) = \bigcup_{q \in F} post(\overline{\mathcal{A}}, q) = \overline{\bigcup_{q \in F} pre(\mathcal{A}, q)} = \overline{L(\mathcal{A})}$ for all NFA $\mathcal{A}$. Furthermore $\mathcal{A}$ is a minimal NFA accepting L iff $\overline{\mathcal{A}}$ is a minimal NFA accepting $\overline{L}$.
The next operation is the wellknown powerset construction.

Definition 1.3 *([12])* Let $\mathcal{A}$ be an NFA. The *subset automaton for* $\mathcal{A}$, denoted $D(\mathcal{A})$, is the DFA $\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, I_{\mathcal{B}}, \delta_{\mathcal{B}}, F_{\mathcal{B}})$, where $Q_{\mathcal{B}} = \{\, \delta_{\mathcal{A}}(I_{\mathcal{A}}, u) \mid u \in \Sigma^* \,\} \subseteq 2^Q$, $I_{\mathcal{B}} = \{\, I_{\mathcal{A}} \,\}$, $F_{\mathcal{B}} = \{\, P \in Q_{\mathcal{B}} \mid P \cap F_{\mathcal{A}} \neq \emptyset \,\}$ and with $\delta_{\mathcal{B}}(P, a) = \delta_{\mathcal{A}}(P, a)$ for all $P \in Q_{\mathcal{B}}$ and for all $a \in \Sigma$.

Obviously $L(\mathcal{A}) = \bigcup_{q \in I} post(\mathcal{A}, q) = post(\mathcal{B}, I_{\mathcal{B}}) = L(D(\mathcal{A}))$.
Next we give the constructive definition of the minimal DFA for a given NFA.

Definition and Theorem 1.4 *([5],[10])*
Let $\mathcal{A}$ be an NFA and $\mathcal{B} = D(\mathcal{A})$. The *minimal DFA* $\mathcal{D} = (Q_{\mathcal{D}}, \Sigma, I_{\mathcal{D}}, \delta_{\mathcal{D}}, F_{\mathcal{D}})$ *for* $\mathcal{A}$, denoted $M(\mathcal{B})$ or $MD(\mathcal{A})$, can be constructed the following way: $Q_{\mathcal{D}} = \{[\,P\,] | P \in Q_{\mathcal{B}}\}$, $I_{\mathcal{D}} = [I_{\mathcal{B}}]$, $F_{\mathcal{D}} = \{[\,P\,] | P \in F_{\mathcal{B}}\}$ and for all states P of $Q_{\mathcal{B}}$ and for all $a \in \Sigma$, $\delta_{\mathcal{D}}([P], a) = [\delta_{\mathcal{B}}(P, a)]$.
Let q be a state of $\mathcal{A}$ and $[P]$ be a state of $MD(\mathcal{A})$. We write $q < [P]$ as an abbreviation for: $\exists P' \in [P] \ q \in P'$. We give some relations that hold between post- and pre-languages in $\mathcal{A}$ and $\mathcal{D}$.

(1) $$q \in F \ \wedge \ q < [P] \quad \Rightarrow \quad [P] \in F_{\mathcal{D}}$$

(2) $$q \in I \quad \Rightarrow \quad q < [P] \text{ where [P] is the initial state of } \mathcal{D}$$

(3) $$p \xrightarrow[\mathcal{A}]{a} q \ \wedge \ p < [P] \quad \Rightarrow \quad q < \delta_{\mathcal{D}}([P], a)$$

Remark 1.5
For every regular language L, there is a unique (up to isomorphsm) minimal DFA $\mathcal{D}$ with $L(\mathcal{D}) = L$, which we call "the" minimal DFA for a language L.
The analogue is not true for minimal NFA since there are regular languages for which several non-isomorphic minimal NFA accepting these languages exist.
All states of a minimal DFA are reachable and at most one state can be non-productive. If it exists, this state is called the sink state or — because its post-language is the empty set — the empty state. Pre-languages of different states of a DFA are disjoint and post-languages of different states in a minimal DFA are different.
The minimal DFA $MD(\mathcal{A})$ for an NFA $\mathcal{A}$ with n states can have up to 2^n states.

The next theorem is interesting because it offers a way to compute the minimal DFA for a given language without computing the equivalence classes of states (as it is suggested by 1.4).

Theorem 1.6 (Brzozowski [3])
Let L be a regular language and $\mathcal{D}$ an arbitrary DFA accepting L. Then $\mathcal{E} := D(\overline{\mathcal{D}})$ is the minimal DFA accepting $\overline{L}$.

Definition 1.7 Let $\mathcal{A} = (Q, \Sigma, I, \delta, F)$ and $\mathcal{B}$ be nondeterministic automata. A mapping $h : Q \to Q_\mathcal{B}$ is called a *morphism* if $h(I) \subseteq I_\mathcal{B}$, $h(F) \subseteq F_\mathcal{B}$ and $h(\delta(q, a)) \subseteq \delta_\mathcal{B}(h(q), a)$ for all $q \in Q$ and $a \in \Sigma$.
$\mathcal{A}$ is called a *subautomaton* of $\mathcal{B}$ if $Q \subseteq Q_\mathcal{B}$, $F = F_\mathcal{B} \cap Q$, $I = I_\mathcal{B} \cap Q$ and $\delta(q, a) \subseteq \delta_\mathcal{B}(q, a) \cap Q$ for all $q \in Q$ and $a \in \Sigma$. If $\delta(q, a) \subseteq \delta_\mathcal{B}(q, a) \cap Q$, we call $\mathcal{A}$ the *subautomaton of $\mathcal{B}$ induced by Q*.
If h is a morphism from $\mathcal{A}$ to $\mathcal{B}$ then we denote by $h(\mathcal{A})$ the subautomaton of $\mathcal{B}$ induced by the set $h(Q)$ of states.

Remark 1.8 Let $\mathcal{A}$ and $\mathcal{B}$ nondeterministic automata and h a morphism from $\mathcal{A}$ to $\mathcal{B}$. Then $L(\mathcal{A}) \subseteq L(h(\mathcal{A})) \subseteq L(\mathcal{B})$.

Proof We have $p \xrightarrow[\mathcal{A}]{w} q \Rightarrow h(p) \xrightarrow[h(\mathcal{A})]{w} h(q) \Rightarrow h(p) \xrightarrow[\mathcal{B}]{w} h(q)$ and $p \in F\,(I) \Rightarrow h(p) \in F_\mathcal{B}\,(I_\mathcal{B})$. Thus an accepting path in $\mathcal{A}$ is mapped to an accepting path in $h(\mathcal{A})$ which is an accepting path of $\mathcal{B}$ by definition. □

2 The Fundamental Automaton

In this section we construct for every regular language L an NFA $\mathcal{F}$, called the fundamental automaton of L, such that for every NFA $\mathcal{A}$ accepting L there exists a morphism from $\mathcal{A}$ in $\mathcal{F}$. We will use this fact to show that a minimal automaton accepting L can be found as a subautomaton of $\mathcal{F}$.
In order to simplify the notation we fix the language L. Let $\mathcal{A} = (Q, \Sigma, I, \delta, F)$ be an arbitrary NFA accepting L, $\mathcal{D} = (Q_\mathcal{D}, \Sigma, I_\mathcal{D}, \delta_\mathcal{D}, F_\mathcal{D})$ the minimal DFA accepting L and $\mathcal{E} = (Q_\mathcal{E}, \Sigma, I_\mathcal{E}, \delta_\mathcal{E}, F_\mathcal{E}) = D(\overline{\mathcal{D}})$. (Because of 1.6, $\mathcal{E}$ is the minimal DFA accepting $\overline{L}$.) If $\mathcal{E}$ has a sink state then this state is not productive in the reversed automaton

$\overline{\mathcal{E}} = (Q_{\overline{\mathcal{E}}}, \Sigma, I_{\overline{\mathcal{E}}}, \delta_{\overline{\mathcal{E}}}, F_{\overline{\mathcal{E}}})$ and we assume that this state and all transitions from this state are deleted in $\overline{\mathcal{E}}$.
$\overline{\mathcal{E}}$ has several important properties described in the following remark.

Remark 2.1 $\overline{\mathcal{E}}$ is a reversed automaton to a deterministic finite automaton. Therefore there exists for all words in L a unique accepting path in $\overline{\mathcal{E}}$, and deletion of any transition in $\delta_{\overline{\mathcal{E}}}$ yields an automaton which accepts a proper subset of L. Moreover, a post-language of an arbitrary NFA accepting L is a subset of a disjoint union of post-languages of $\overline{\mathcal{E}}$. $\mathcal{E}$ is isomorphic to $MD(\overline{\mathcal{A}})$ and thus we can associate to every state r of $\mathcal{E}$ an equivalence class of sets of states of $\mathcal{A}$ (cf. Definition 1.4). For every state q of $\mathcal{A}$ we denote by $\dot{q}$ the set of states r of $\mathcal{E}$ with $q < r$.

Now we arrive at the definition of the fundamental automaton, given $\mathcal{E} = D(\overline{\mathcal{D}})$.

Definition 2.2 (The Fundamental Automaton)
The *fundamental automaton* $\mathcal{F} = (Q_{\mathcal{F}}, \Sigma, I_{\mathcal{F}}, \delta_{\mathcal{F}}, F_{\mathcal{F}})$ is defined as follows:

$$Q_{\mathcal{F}} = \{\, P \subseteq Q_{\overline{\mathcal{E}}} \mid \bigcap P \neq \emptyset \}$$
$$I_{\mathcal{F}} = \{\, P \in Q_{\mathcal{F}} \mid P \subseteq I_{\overline{\mathcal{E}}} \,\}$$
$$F_{\mathcal{F}} = \{\, P \in Q_{\mathcal{F}} \mid P \cap F_{\overline{\mathcal{E}}} \neq \emptyset \,\}$$
$$\delta_{\mathcal{F}}(P, a) = \{\, P' \in Q_{\mathcal{F}} \mid P' \subseteq \delta_{\overline{\mathcal{E}}}(P, a) \,\} \text{ for all } P \text{ in } Q_{\mathcal{F}} \text{ and } a \text{ in } \Sigma.$$

We proceed in three steps. First, we show that the fundamental automaton accepts L. Then we verify that for each automaton $\mathcal{A}$ accepting L there exists a morphism from $\mathcal{A}$ into the fundamental automaton of L. This proves that a minimal automaton accepting L can be found as a subautomaton of $\mathcal{F}$. We conclude with a condition that describes how to find certain subautomata of $\mathcal{F}$ that accept L.

Lemma 2.3 The fundamental automaton $\mathcal{F}$ is equivalent to $\overline{\mathcal{E}}$.

Proof $L(\mathcal{F}) \supseteq L(\overline{\mathcal{E}})$ because $\overline{\mathcal{E}}$ is a subautomaton of $\mathcal{F}$ (take all subsets of $Q_{\mathcal{F}}$ of size 1). The following fact on the transiton function of $\mathcal{F}$ can be shown easily by an induction on the length of words w:

$$\forall P, P' \in Q_{\mathcal{F}}\ \forall w \in \Sigma^*\ P \underset{\mathcal{F}}{\overset{w}{\rightarrow}} P' \ \Rightarrow\ \forall p' \in P'\ \exists p \in P\ p \underset{\overline{\mathcal{E}}}{\overset{w}{\rightarrow}} p' \tag{4}$$

Then the definition of initial and final states of $\mathcal{F}$ yields: if $P \underset{\mathcal{F}}{\overset{w}{\rightarrow}} P'$ is an accepting path in $\mathcal{F}$ then there exists a state $p' \in P' \cap F_{\overline{\mathcal{E}}}$ and a state $p \in P \subseteq I_{\overline{\mathcal{E}}}$ with $p \underset{\overline{\mathcal{E}}}{\overset{w}{\rightarrow}} p'$.
Thus $L(\mathcal{F}) \subseteq L(\overline{\mathcal{E}})$. □

Lemma 2.4
Let $\mathcal{A}$ be an NFA accepting L and $\mathcal{F}$ the fundamental automaton of L. The mapping $q \mapsto \dot{q}$ is a morphism from $\mathcal{A}$ to $\mathcal{F}$.

Proof

Because of Theorem 1.6, we may regard $\mathcal{E}$ as $D(\overline{D(\mathcal{A})})$. The states of $\mathcal{E}$ are thus sets of states of $\overline{D(\mathcal{A})}$. For a state q' of $\overline{D(\mathcal{A})}$) and a state r of $\mathcal{E}$, we have then $q' < r$ iff $q' \in r$.

We will begin by showing that $q \mapsto \overset{\bullet}{q}$ is a mapping from $Q_{\mathcal{A}}$ in $Q_{\mathcal{F}}$.

Let q be a state of $\mathcal{A}$ and $r \in \overset{\bullet}{q}$. Since all states of $\mathcal{A}$ are supposed to be productive, there is a state q' of $D(\mathcal{A})$ such that $q \in q'$. Since $q < r$ and $\mathcal{E}$ is isomorphic to $MD(\overline{\mathcal{A}})$, there is a state R of $D(\overline{\mathcal{A}})$ such that $q \in R$ and $post(D(\overline{\mathcal{A}}), R) = post(\mathcal{E}, r)$. (Each state of $\mathcal{E}$ corresponds to a non-empty set of states of $D(\overline{\mathcal{A}})$ whose elements have the same post-language.)

Now we have

$$post(\overline{D(\mathcal{A})}, q') = pre(D(\mathcal{A}), q') \subseteq pre(\mathcal{A}, q) = post(\overline{\mathcal{A}}, q) \subseteq \bigcup_{p \in R} post(\overline{\mathcal{A}}, p)$$
$$= post(D(\overline{\mathcal{A}}), R) = post(\mathcal{E}, r) = post(D(\overline{D(\mathcal{A})}), r) = \bigcup_{p' \in r} post(\overline{D(\mathcal{A})}, p')$$

Since different pre-languages of a DFA are disjoint, so are different post-language of $\overline{D(\mathcal{A})}$. Thus we may conclude that $q' \in r$, i.e. $r \in \overset{\bullet}{q}'$. Since r was chosen arbitrarily from $\overset{\bullet}{q}$, we have shown $\overset{\bullet}{q} \subseteq \overset{\bullet}{q}'$.

We have $\overset{\bullet}{q}' = \{r \in Q_{\mathcal{E}} \mid q' \in r\}$, thus $q' \in \bigcap \overset{\bullet}{q}' \subseteq \bigcap \overset{\bullet}{q}$, showing that $\overset{\bullet}{q}$ is a state of $\mathcal{F}$. So we have shown that $q \mapsto \overset{\bullet}{q}$ is a mapping into $\mathcal{F}$.

To show that it is a morphism, we have to verify $q \in F\,(I) \Rightarrow \overset{\bullet}{q} \in F_{\mathcal{F}}\,(I_{\mathcal{F}})$ and $p \underset{\mathcal{A}}{\overset{a}{\rightarrow}} q \Rightarrow \overset{\bullet}{p} \underset{\mathcal{F}}{\overset{a}{\rightarrow}} \overset{\bullet}{q}$.

$$\begin{array}{llll}
q \in I_{\mathcal{A}} & q \in F_{\mathcal{A}} & p \underset{\mathcal{A}}{\overset{a}{\rightarrow}} q & \\
 & \Downarrow & & \text{Definition of the reversed automaton} \\
q \in F_{\overline{\mathcal{A}}} & q \in I_{\overline{\mathcal{A}}} & q \underset{\overline{\mathcal{A}}}{\overset{a}{\rightarrow}} p & \\
 & \Downarrow & & \text{Equations (1,2,3)} \\
\overset{\bullet}{q} \subseteq F_{\mathcal{E}} & \overset{\bullet}{q} \cap I_{\mathcal{E}} \neq \emptyset & \overset{\bullet}{p} \supseteq \delta_{\mathcal{E}}(\overset{\bullet}{q}, a) & \\
 & \Downarrow & & \text{Definition of the reversed automaton} \\
\overset{\bullet}{q} \subseteq I_{\overline{\mathcal{E}}} & \overset{\bullet}{q} \cap F_{\overline{\mathcal{E}}} \neq \emptyset & \overset{\bullet}{q} \subseteq \delta_{\overline{\mathcal{E}}}(\overset{\bullet}{p}, a) & \\
 & \Downarrow & & \text{Definition of the fundamental automaton} \\
\overset{\bullet}{q} \in I_{\mathcal{F}} & \overset{\bullet}{q} \in F_{\mathcal{F}} & \overset{\bullet}{p} \underset{\mathcal{F}}{\overset{a}{\rightarrow}} \overset{\bullet}{q} &
\end{array}$$

□

Theorem 2.5 A minimal automaton accepting L can be found as a subautomaton of $\mathcal{F}$.

Proof Let $\mathcal{A}$ be a minimal NFA accepting L and let $h : q \mapsto \overset{\bullet}{q}$ be the morphism of Lemma 2.4. The number of states of $h(\mathcal{A})$ is less or equal to the number of states of

$\mathcal{A}$. Furthermore we have: $L = L(\mathcal{A}) \subseteq L(h(\mathcal{A})) \subseteq L(\mathcal{F}) = L$. Thus $L(h(\mathcal{A})) = L$ and $h(\mathcal{A})$ is a minimal NFA accepting L and a subautomaton of $\mathcal{F}$. $\square$

With the next lemma we give a criterion for equivalent subautomata of $\mathcal{F}$ that improves the one given by Kim [11] and was suggested by Kahlert [9]. Kim claimed that his criterion was necessary, which is not true in general.
We start with an auxiliary definition. Let Q be a set, $\mathcal{P} \subseteq 2^Q$ and $R \subset Q$. We say $\mathcal{P}$ *covers R with subsets* iff $R = \bigcup\{P \in \mathcal{P} \mid P \subseteq R\}$.

Lemma 2.6
Let $\mathcal{F}_\mathcal{P}$ be a subautomaton of $\mathcal{F}$ induced by the set $\mathcal{P} \subseteq Q_\mathcal{F}$ of states. Assume

(i) $\mathcal{P}$ covers $I_{\overline{\mathcal{E}}}$ with subsets and

(ii) $\mathcal{P}$ covers $\delta_{\overline{\mathcal{E}}}(P, a)$ with subsets for all $P \in \mathcal{P}$, $a \in \Sigma$.

Then $\mathcal{F}_\mathcal{P}$ is equivalent to $\mathcal{F}$.

Proof Let $\mathcal{P}$ fulfill conditions (i) and (ii) and let $p_0 \underset{\overline{\mathcal{E}}}{\overset{a_1}{\rightarrow}} \dots \underset{\overline{\mathcal{E}}}{\overset{a_n}{\rightarrow}} p_n$ be an accepting path in $\overline{\mathcal{E}}$. We show that there exists an accepting path $P_0 \underset{\mathcal{F}_\mathcal{P}}{\overset{a_1}{\rightarrow}} \dots \underset{\mathcal{F}_\mathcal{P}}{\overset{a_n}{\rightarrow}} P_n$ in $\mathcal{F}_\mathcal{P}$ with $p_i \in P_i$. By $p_0 \in I_{\overline{\mathcal{E}}}$ and condition (i) there is a set $P_0 \in \mathcal{P}$ with $P_0 \subseteq I_{\overline{\mathcal{E}}}$ and $p_0 \in P_0$. Furthermore P_0 is initial in $\mathcal{F}_\mathcal{P}$. We find $P_1 \dots P_n$ by induction in the following way: if $p_i \in P_i$ then $p_{i+1} \in \delta_{\overline{\mathcal{E}}}(P_i, a_{i+1})$; by condition (ii) there exists $P_{i+1} \subseteq \delta_{\overline{\mathcal{E}}}(P_i, a_{i+1})$, $P_{i+1} \in \mathcal{P}$ with $p_{i+1} \in P_{i+1}$. By definition of $\mathcal{F}$ we get $P_i \underset{\mathcal{F}}{\overset{a_{i+1}}{\rightarrow}} P_{i+1}$. P_n is final in $\mathcal{F}$ because $p_n \in P_n$ and p_n is final in $\overline{\mathcal{E}}$. Thus $L(\overline{\mathcal{E}}) \subseteq L(\mathcal{F}_\mathcal{P})$. $\square$

3 An Example

In this section we present a simple example that shows that the criterion given in 2.6 is not necessary for subautomata of the fundamental automaton which are equivalent to the given NFA. Thus it does not yield an algorithm for determining a minimal NFA in general.
Consider the NFA $\mathcal{A}$ and its reversed NFA $\overline{\mathcal{A}}$ given by the following transition tables. Initial and final states are marked with "i" or "f", respectively.

$\mathcal{A}$

	a	b
f0	$0,2$	
1	3	$2,3$
i2	1	$0,1$
3	3	1

$\overline{\mathcal{A}}$

	a	b
i0	0	2
1	2	$2,3$
f2	0	1
3	$1,3$	1

The following tables show the minimal DFA $\mathcal{E}$ equivalent to $\overline{\mathcal{A}}$, the reverse of this DFA and the mapping $q \mapsto \overset{\bullet}{q}$ from $Q_\mathcal{A}$ into $2^{Q_\mathcal{E}}$. The leftmost column of the first table contains for each state of $\mathcal{E}$ the corresponding states of $\overline{\mathcal{A}}$.

		a	b
$\{0\}$	ie_0	e_0	e_1
$\{2\}$	fe_1	e_0	e_2
$\{1\}$	e_2	e_1	e_3
$\{2,3\}$	fe_3	e_4	e_2
$\{0,1,3\}$	e_4	e_5	e_5
$\{0,1,2,3\},\{1,2,3\}$	fe_5	e_5	e_5

	a	b
fe_0	e_0, e_1	
ie_1	e_2	e_0
e_2		e_1, e_3
ie_3		e_2
e_4	e_3	
ie_5	e_4, e_5	e_4, e_5

$h(0) = \{e_0, e_4, e_5\}$
$h(1) = \{e_2, e_4, e_5\}$
$h(2) = \{e_1, e_3, e_5\}$
$h(3) = \{e_3, e_4, e_5\}$

The states of the fundamental automaton are certain subsets of $Q_{\mathcal{E}}$. In fact, a look at the correspondence between $\mathcal{E}$ and $D(\overline{D(\mathcal{A})})$, which is not presented here, would show that the fundamental automaton contains actually *all* subsets of $Q_{\mathcal{E}}$. That is why we do not present it here. But we show the transitition table of the image of $\mathcal{A}$ under the morphism h.

	a	b
f$\{e_0, e_4, e_5\}$	$\{e_0, e_1, e_3, e_4, e_5\}$	$\{e_4, e_5\}$
$\{e_2, e_4, e_5\}$	$\{e_3, e_4, e_5\}$	$\{e_1, e_3, e_4, e_5\}$
i$\{e_1, e_3, e_5\}$	$\{e_2, e_4, e_5\}$	$\{e_0, e_2, e_4, e_5\}$
$\{e_3, e_4, e_5\}$	$\{e_3, e_4, e_5\}$	$\{e_2, e_4, e_5\}$

This transition table has to be interpreted like this: if $Q \subseteq Q_{\mathcal{E}}$ is in the line corresponding to $P \subseteq Q_{\mathcal{E}}$ and letter a, then in the fundamental automaton there is a transition from P with a to each subset of Q. So the subautomaton of the fundamental automaton induced by the image of h is:

	a	b
f$h(0)$	$h(0), h(2), h(3)$	
$h(1)$	$h(3)$	$h(2), h(3)$
i$h(2)$	$h(1)$	$h(0), h(1)$
$h(3)$	$h(3)$	$h(1)$

The automaton is (up to isomorphism) the same as $\mathcal{A}$, except for one superflous transition.
The chosen set of states $\mathcal{P} = \{h(0), h(1), h(2), h(3)\}$ of the fundamental automaton covers indeed the set of initial states of $\overline{\mathcal{E}}$ with subsets, so the first criterion of 2.6 holds. But the transition table shows that the set $\delta_{\overline{\mathcal{E}}}(\{e_0, e_4, e_5\}, b) = \{e_4, e_5\}$ is not covered with subsets; so the second property does not hold. Thus $h(\mathcal{A})$ is a subautomaton of the fundamental automaton that accepts L, but does not fulfill the conditions (i) and (ii) of Lemma 2.6.
In fact, there is no set of states with the property of Lemma 2.6 that has less than 5 states, so for this example language the suggested heuristic fails to find a minimal NFA.

4 Some Comments on the Practical Computation

In this section we outline an algorithm to compute a small NFA. This algorithm is based on the results of Section 2.
An implementation of this strategy (with some improvements not described here) is part of the AMoRE-System, available via Anonymous FTP.

Main Idea

The main idea how to compute a small NFA for a given regular language L is to proceed in four steps:

1. Compute a minimal DFA $\mathcal{E}$ for $\overline{L}$.
2. Determine the *allowed* sets, i.e. those subsets of the state set of $\mathcal{E}$ that are states of the fundamental automaton.
3. Search for a selection $\mathcal{P}$ of allowed sets that satisfies the cover/closure property (i), (ii) of Lemma 2.6 and is minimal with that property.
4. Compute the subautomaton of the fundamental automaton induced by the selection $\mathcal{P}$.

The most important point is, of course, the step 3. For this step we will give (in Figure 1) a recursive procedure `cover` that explores in a backtracking strategy all reasonable extensions of a given selection to a selection satisfying the cover/closure property. `cover` receives three arguments called `allowed`, `selected`, `remain` for subsets of $Q_{\mathcal{E}}$. The intuitive meaning is the following: `allowed` contains all allowed sets that must be considered for possible extensions of the set `selected`. `selected` contains the subsets of $Q_{\mathcal{E}}$-states that have been selected so far. `remain` contains all subsets of $Q_{\mathcal{E}}$-states for which we will have to make sure (possibly by further selections) that `selection` covers them by subsets.
More precisely, the pre- and postcondition of `cover` are given by:

Precondition: $I_{\overline{\mathcal{E}}} \notin$ `remain` $\Rightarrow$ (`selected` covers $I_{\overline{\mathcal{E}}}$ by subsets) and
$\forall R \in$ `selected` $\forall a \in \Sigma : \delta_{\overline{\mathcal{E}}}(R, a) \notin$ `remain` $\Rightarrow$ (`selected` covers $\delta_{\overline{\mathcal{E}}}(R, a)$ by subsets)
Postcondition: If there is a $\mathcal{P} \supseteq$ `selected` with $\mathcal{P} \setminus$`selected` $\subseteq$ `allowed` such that $\mathcal{P}$ fulfills the cover/closure property of Lemma 2.6, then the global variable `best_solution` contains such a $\mathcal{P}$ of minimal size, otherwise `best_solution` remains unchanged. In any case, `best_known = |best_solution|`.

With this procedure `cover`, the algorithm for the computation of a small NFA looks like this:

1. Compute the minimal DFA $\mathcal{D}$ for L.
2. Apply the powerset constuction to $\overline{\mathcal{D}}$ and obtain $\mathcal{E}$.
3. Let `best_known` $= |Q_{\mathcal{E}}| + 1$.
4. `cover`$(\{P \subseteq Q_{\mathcal{E}} \mid \bigcap P \neq \emptyset\}, \emptyset, \{I_{\overline{\mathcal{E}}}\})$
5. Define the output automaton $\mathcal{A} = (Q, \Sigma, I, \delta, F)$ by

$$\begin{aligned}
Q &= \texttt{best_solution}, \\
I &= \{P \in Q \mid P \subseteq I_{\overline{\mathcal{E}}}\}, \\
F &= \{P \in Q \mid P \cap F_{\overline{\mathcal{E}}} \neq \emptyset\}, \\
\delta(R, a) &= \{P \in Q \mid P \subseteq \delta_{\overline{\mathcal{E}}}(R, a)\} \text{ for all } R \in Q,\ a \in \Sigma.
\end{aligned}$$

```
cover(allowed,selected,remain)
   /* allowed, selected, remain ⊆ Q_E fulfill the precondition */
   /* explores all reasonable extensions of selected
   to a set satisfying the cover/closure property of Lemma 2.6 */
   if remain = ∅ then
      /* no sets remain to be covered, we have a solution! */
      best_solution := selected
   else
      /* there are still sets that remain to be covered */
      for all current ∈ remain do
         if selected covers current with subsets then
            /* current is already covered, treat the rest of remain */
            cover(allowed, selected, remain\{current} )
         else
            /* current has to be covered by further selections */
            if |selected| + 1 < best_known
               /* stop backtracking if best known solution cannot be improved */
               for all P ⊆ current with P ∈ allowed do
                  allowed := allowed\{P}
                  cover(allowed, selected ∪ {P}, remain ∪ {δ_Ē(P, a) | a ∈ Σ})
               rof
            fi
         fi
      rof
   fi
```

Figure 1: The recursive procedure `cover`

One important and simple way to improve this strategy is to compare the size of the state set of $\mathcal{E}$ and $\mathcal{D}$ after line 2 and exchange them if $\mathcal{D}$ is smaller. In this case, the algorithm computes a small NFA for the reversed language $\overline{L}$, thus the output NFA has to be reversed as well.

Note that the modification of `allowed` in the last for-loop is to avoid that one and the same selection of subsets is explored more than once: When inside one incarnation of `cover` the addition of a certain set P to the current selection has been fully explored, it is not necesarry to consider P any more unless this incarnation is exited and the current selection becomes smaller.

In the programm system AMoRE, a similar (but non-recursive) algorithm has been implemented. One great problem one has to deal with is how to represent the subsets of $Q_{\mathcal{E}}$ and $2^{Q_{\mathcal{E}}}$ without wasting too much space. We chose to represent the subsets of $Q_{\mathcal{E}}$ by 32-bit integers, so the available implementation does not work if both the minimal DFA for L and for $\overline{L}$ have more than 32 states.

Another problem is how to realize the last for-loop running over all subsets of a given set. Here, we have decided to trace the sets in the canonical ordering they have when represented by integers. Experience shows that this order influences the performance of the algotithm very much, so it might be advantageous to choose a more sophisticated ordering: For example, one could try to treat the large subsets before the small ones or vice versa.

5 Fundamental vs. Canonical Automaton

In this section, we will give a more abstract definition of the fundamental automaton, which is, however, less suitable for actual calculation. Using this characterization, we compare the fundamental automaton to a different automaton, the so-called *canonical automaton*, defined by Arnold, Dicky and Nivat in [1]. This one has similar properties as the fundamental automaton, but it may be smaller. Moreover, we transfer the mentioned sufficient cover/closure property (conditions (i),(ii) of Lemma 2.6) for equivalent subautomata from the fundamental to this canonical automaton.

5.1 Fundamental Automaton Revisited

Define the MDFA $\mathcal{D}'$ (minimal deterministic finite automaton) for a language L by

$$Q_{\mathcal{D}'} = \{w^{-1}L \mid w \in \Sigma^*\}$$
$$I_{\mathcal{D}'} = \{L\}$$
$$\delta_{\mathcal{D}'}(w^{-1}L, a) = (wa)^{-1}L$$
$$F_{\mathcal{D}'} = \{w^{-1}L \mid \epsilon \in w^{-1}L\}$$

Observing the fact $\overline{w}^{-1}\overline{L} = \overline{Lw^{-1}}$, we get that the MDFA for $\overline{L}$ is isomorphic to the following DFA $\mathcal{E}'$ via the mapping $M \mapsto \overline{M}$.

$$Q_{\mathcal{E}'} = \{Lw^{-1} \mid w \in \Sigma^*\}$$
$$I_{\mathcal{E}'} = \{L\}$$
$$\delta_{\mathcal{E}'}(Lw^{-1}, a) = L(aw)^{-1}$$
$$F_{\mathcal{E}'} = \{Lw^{-1} \mid \epsilon \in Lw^{-1}\}$$

With this definition we can define the fundamental automaton $\mathcal{F}'$ differently than in Section 2 by:

$$Q_{\mathcal{F}'} = \{P \subseteq Q_{\mathcal{E}'} \mid \bigcap P \neq \emptyset\}$$
$$I_{\mathcal{F}'} = \{P \in Q_{\mathcal{F}'} \mid \epsilon \in \bigcap P\}$$
$$\delta_{\mathcal{F}'}(P, a) = \{R \in Q_{\mathcal{F}'} \mid R \subseteq \delta_{\overline{\mathcal{E}'}}(P, a)\}$$
$$F_{\mathcal{F}'} = \{P \in Q_{\mathcal{F}'} \mid L \in P\}$$

To see that this definition is indeed equivalent to the one given before, let $\mathcal{E} = D(\overline{\mathcal{D}'})$ with D' being the minimal DFA for the language L as described above, i.e. the states of D' are left quotients. (Note that D' is a possible choice for the D of the last section, so $\mathcal{E}$ is a possible choice for the $\mathcal{E}$ in SectionSecFunAut). Let $\mathcal{F}'$ be the fundamental automaton as defined in Definition 2.2 of Section 2, i.e. with states in $2^{Q_{\mathcal{E}}}$.

Lemma 5.1 $\mathcal{F}'$ is isomorphic to $\mathcal{F}$.

Proof It is a well known fact that there exists an isomorphism α from $\mathcal{E}$ onto $\mathcal{E}'$, namely

$$\alpha : Q_{\mathcal{E}} \longrightarrow Q_{\mathcal{E}'} \quad , \quad (\{u^{-1}L \mid uw \in L\}) \mapsto Lw^{-1}.$$

We claim that the extension from α to $Q_{\mathcal{F}} \subseteq 2^{Q_{\mathcal{E}}}$ into $2^{Q_{\mathcal{E}'}}$ is an isomorphism from $Q_{\mathcal{F}}$ onto $Q_{\mathcal{F}'}$. We will denote this extension again by α. Since the original α is injective, so is this extension.

Let $\emptyset \neq M \subseteq Q_{\mathcal{E}} = \{\{u^{-1}L \mid uw \in L\} \mid w \in \Sigma^*\}$. Then we have the following equivalence showing that α is indeed a mapping from $Q_{\mathcal{F}}$ into $Q_{\mathcal{F}'}$.

$$\begin{aligned}
M \in Q_{\mathcal{F}} &\iff \bigcap M \neq \emptyset \\
&\iff \exists u^{-1}L \in Q_{\mathcal{D}'} : \forall P \in M : u^{-1}L \in P \\
&\iff \exists u \in \Sigma^* : u^{-1}L \neq \emptyset \wedge \forall Lw^{-1} \in \alpha(M) : uw \in L \\
&\iff \bigcap \alpha(M) \neq \emptyset \\
&\iff \alpha(M) \in Q_{\mathcal{F}'}
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
M \in I_{\mathcal{F}} &\iff M \subseteq I_{\overline{\mathcal{E}}} = F_{\mathcal{E}} \\
&\iff \forall P \in M : P \cap I_{\mathcal{D}'} \neq \emptyset \\
&\iff \forall P \in M : L \in P \\
&\iff \forall Lw^{-1} \in \alpha(M) : w \in L \\
&\iff \epsilon \in \bigcap \alpha(M) \\
&\iff \alpha(M) \in I_{\mathcal{F}'}
\end{aligned}$$

The correspondence of the set of final states is shown by the following equivalences:

$$\begin{aligned}
M \in F_{\mathcal{F}} &\iff \emptyset \neq M \cap F_{\overline{\mathcal{E}}} = M \cap I_{\mathcal{E}} \\
&\iff I_{\overline{\mathcal{D}'}} \in M \\
&\iff \{u^{-1}L \in Q_{\mathcal{D}} \mid \epsilon \in u^{-1}L\} \in M \\
&\iff L = \alpha(\{u^{-1}L \mid \epsilon \in u^{-1}L\}) \in \alpha(M) \\
&\iff \alpha(M) \in F_{\mathcal{F}'}
\end{aligned}$$

Finally, we check that α commutes with the transition relations. This is verified as follows:

$$\begin{aligned}
N \in \delta_{\mathcal{F}}(M, a) &\iff N \subseteq \delta_{\overline{\mathcal{E}}}(M, a) \\
&\iff \alpha(N) \subseteq \alpha(\delta_{\overline{\mathcal{E}}}(M, a)) \\
&\iff \alpha(N) \subseteq \delta_{\overline{\mathcal{E}'}}(\alpha(M), a) \\
&\iff \alpha(N) \in \delta_{\mathcal{F}'}(\alpha(N), a)
\end{aligned}$$

This completes the proof and establishes the claimed correspondence of the two definitions for the fundamental NFA. In the remainder of the paper we will write $\mathcal{F}$ instead of $\mathcal{F}'$. $\square$

5.2 The Canonical Automaton

We compare the fundamental automaton to a different NFA introduced by Arnold, Dicky and Nivat in [1]. They defined the *canonical automaton* $\mathcal{C}$ as described now. We start with an auxiliary definition. For $K \subseteq \Sigma^*$, let $\phi(K) = \{u \in \Sigma \mid uK \subseteq L\} = \bigcap_{v \in K} Lv^{-1}$.

Definition 5.2 The *canonical automaton* $\mathcal{C}$ is given by

$$\begin{aligned} Q_{\mathcal{C}} &= \{\phi(K) \mid K \subseteq \Sigma^* \text{ and } K, \phi(K) \neq \emptyset\} \\ &= \{\textstyle\bigcap P \neq \emptyset \mid P \text{ is a nonempty set of right quotients of } L\} \end{aligned}$$

$$\begin{aligned} I_{\mathcal{C}} &= \{\textstyle\bigcap P \in Q_{\mathcal{C}} \mid \epsilon \in \bigcap P\} \\ \delta_{\mathcal{C}}(\textstyle\bigcap P, a) &= \{\textstyle\bigcap R \in Q_{\mathcal{C}} \mid (\bigcap P)a \subseteq \bigcap R\} \text{ for all } \textstyle\bigcap P \in Q_{\mathcal{C}} \text{ and all } a \in \Sigma^* : \\ F_{\mathcal{C}} &= \{\textstyle\bigcap P \in Q_{\mathcal{C}} \mid \bigcap P \subseteq L\} \end{aligned}$$

The canonical and the fundamental automaton have similar properties. For example, it is shown in [1] that for any NFA $\mathcal{A}$ accepting L, there exists a morphism from $\mathcal{A}$ into $\mathcal{C}$, namely $q \mapsto \phi(post(\mathcal{A}, q))$. Thus any minimal NFA accepting L is isomorphic to some subautomaton of $\mathcal{C}$.
An interesting fact about the canonical automaton is that it is in some sense the smallest NFA with that property, because any epimorphism from $\mathcal{C}$ onto an equivalent NFA is injective.
Let us denote by $\mathcal{F}$ the fundamental automaton as defined in this section. The relation between the fundamental automaton and the canonical automaton is illustrated by the observation of [1] that a morphism f from $\mathcal{F}$ into $\mathcal{C}$ is given by

$$f : P \mapsto \phi(post(\mathcal{F}, P)) = \bigcap P$$

To see the last equality we observe that for any $w \in \Sigma^*$ and any $P \in Q_{\mathcal{F}}$ we have

$$\begin{aligned} w \in post(\mathcal{F}, P) &\iff \exists R \in Q_{\mathcal{F}} : R \in F_{\mathcal{F}} \wedge R \in \delta_{\mathcal{F}}(P, w) \\ &\iff \exists R \in Q_{\mathcal{F}} : L \in R \wedge \delta_{\mathcal{E}}(R, \overline{w}) \subseteq P \\ &\iff \exists R \subseteq Q_{\mathcal{E}} : \bigcap R \neq \emptyset \wedge L \in R \wedge \forall Lv^{-1} \in R(L(wv)^{-1} \in P) \\ &\iff Lw^{-1} \in P. \end{aligned}$$

(To see the last implication from right to left choose $R := \{L\}$.) This shows for any $u \in \Sigma^*$ and any $P \in Q_{\mathcal{F}}$

$$\begin{aligned} u \in \phi(post(\mathcal{F}, P)) &\iff \forall w \in post(\mathcal{F}, P) : uw \in L \\ &\iff \forall w \in \Sigma^* : Lw^{-1} \in P \Rightarrow uw \in L \\ &\iff \forall w \in \Sigma^* : Lw^{-1} \in P \Rightarrow u \in Lw^{-1} \\ &\iff u \in \bigcap P \end{aligned}$$

The morphism f is always surjective, and it is injective if there is no right quotient that is a superset of a nonempty intersection of other right quotients. There are

examples where there is such a right quotient. For example for $L = \{aa, ab, bb\}$, the automaton $\mathcal{F}$ has 5 states (namely $\{La^{-1}\}$, $\{Lb^{-1}\}$, $\{La^{-1}, Lb^{-1}\}$, $\{L\}$ and $\{\{\epsilon\}\}$), whereas $\mathcal{C}$ has 4 states (namely $\{a\}$, $\{a, b\}$, L and $\{\epsilon\}$); so the two automata $\mathcal{F}$ and $\mathcal{C}$ are in general not isomorphic.
The conditions of Lemma 2.6 leading to the minimization heuristic can be transfered to $\mathcal{C}$, yielding another heuristic for finding a small NFA for a given language.

5.3 A Criterion for Equivalent Subautomata of the Canonical NFA

Lemma 5.3 Let $\mathcal{P} \subseteq Q_{\mathcal{F}}$ and $\mathcal{P}_{\cap} \subseteq Q_{\mathcal{C}}$ with $\mathcal{P} = \{P \subseteq Q_{\mathcal{E}} \mid \bigcap P \in \mathcal{P}_{\cap}\}$. (That is, $\mathcal{P} = f^{-1}(\mathcal{P}_{\cap})$, with f being the morphism $P \mapsto \bigcap P$ from above.) Let the following two properties be fulfilled:

(i) $\mathcal{P}$ covers $I_{\overline{\mathcal{E}}}$ with subsets and

(ii) For all $R \in \mathcal{P}$ and for all $a \in \Sigma$, $\mathcal{P}$ covers $\delta_{\overline{\mathcal{E}}}(R, a)$ with subsets.

We claim that the subautomaton $\mathcal{C}_{\mathcal{P}_{\cap}}$ of $\mathcal{C}$ induced by $\mathcal{P}_{\cap}$ recognizes L.

Proof The direct proof of this result is possible, but we can argue simply like this: Because of Lemma 2.6, the subautomaton $\mathcal{F}_{\mathcal{P}}$ of $\mathcal{F}$ induced by $\mathcal{P}$ is equivalent to $\mathcal{A}$. The subautomaton $\mathcal{C}_{\mathcal{P}_{\cap}}$ is simply the image of $\mathcal{F}_{\mathcal{P}}$ under the morphism $P \mapsto \bigcap P$, so we have $L \subseteq L(\mathcal{C}_{\mathcal{P}_{\cap}}) \subseteq L(\mathcal{C}) = L$, showing our claim. □

Unfortunately, this cover/closure property of Lemma 5.3 is not necessary for equivalent subautomata of $\mathcal{C}$, so also for $\mathcal{C}$ it does not give a heuristic that always produces a minimal NFA.
One way of making use of this lemma would be to proceed as described in Secion 4, but put together states that have the same image under $P \mapsto \phi(post(\mathcal{F}, P)$ before the backtracking is started.

6 Conclusion

The minimization of nondeterministic finite automata is a difficult problem. One possible approach is to search among the subautomata of either the fundamental or the canonical automaton of a language. Both of these do contain all minimal NFA as subautomata, but we have no efficiently testable criterion that characterizes those subautomata that accept the same language. All we have is a sufficient criterion, so one can at least search for subautomata fulfilling this. It remains an open question if and how this criterion can be modified so that it characterizes all equivalent subautomata of the fundamental or the canonical automaton. If this modification is not possible, it would be interesting at least to know for which languages the mentioned sufficient criterion will be necessary.
Apart from the minimization problem the definition and the properties of the fundamental and canonical automaton are interesting. Both of these are unique for a given regular language. Both contain homomorphic images of all automata equivalent to themselves, and the canonical automaton is in some sense the smallest with that property.

References

[1] A. Arnold, A. Dicky, M. Nivat, A note about minimal non-deterministic automata; in: *Bulletin of the EATCS* 47, June 1992, pp. 166-169.

[2] W. Brauer, *Automatentheorie*, Teubner, Stuttgart 1984.

[3] J.A. Brzozowski: Canonical regular expressions and minimal state graphs for definite events, in: *Proc. Symp. on Math. Theory of Automata*, Vol. 12, Brooklyn, N. Y.: Brooklyn Polytechnic Institute, 1963, pp. 529 -561.

[4] J.E. Hopcroft: An n log(n) algorithm for minimizing states in a finite automaton; in: Z. Kohavi, A. Paz (Eds.): *Theory of Machines and Computation*; Academic Press, New York etc., 1971, pp. 189-196.

[5] J.E. Hopcroft, J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation*; Addison-Wesley, 1979.

[6] M.R. Garey, D.S. Johnson: *Computers and Intractability – A Guide to the Theory of NP-Completeness*; Freeman, San Francisco, 1979.

[7] K. Indermark: Zur Zustandsminimisierung nichtdeterministischer erkennender Automaten; GMD Seminarberichte Bd. 33, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin – Bonn, 1970.

[8] T. Jiang, B. Ravikumar: Minimal NFA Problems Are hard (Extended Abstract); in: J.Leach Albert, B. Monien, M.Rodriguez Artalejo (Eds.): *Automata, Languages and Programming, 18th International Colloquium*, Madrid, July 1991, Proceedings, LNCS 510, Springer, Berlin etc., 1991, pp. 629-640.

[9] T. Kahlert: *Ein Verfahren für die Minimierung nichtdeterministischer endlicher Automaten und seine Implementierung*; Diplomarbeit, Christian-Albrechts-Universität, Kiel, 1991.

[10] T. Kameda, P. Weiner: On the State Minimization of Nondeterministic Finite Automata; *IEEE Trans. Comp. C–19*(1970), pp. 617-627.

[11] J. Kim: State minimization of nondeterministic machines; IBM Thomas J. Watson Res. Center Rep. RC 4896, 1974.

[12] M.O. Rabin, D. Scott: Finite Automata and their decision problems; *IBM J. Res. and Develop.* 3, April 1959, pp. 114-125.

[13] (This paper) O Matz, A. Potthoff, Computing Small Nondeterministic Automata, Proceedings of the Workshop on Tools and Algorithms for the Construction and Analysis of Systems; U.H. Engberg, K.G. Larsen, A. Skou (eds.); BRICS Notes Series, May 1995, ISSN 0909-3206, pp. 74-88.