KYBERNETIKA — VOLUME *10* (1974), NUMBER 4

# Finite Branching Automata

IVAN M. HAVEL

A new abstract device, the finite branching automaton, is introduced and explored. The main source of motivation for this notion can be found in the area of state-space problem solving. The finite branching automaton differs from the ordinary finite automaton in its accepting behavior: instead of strings it accepts languages and thus it recognizes a family of languages rather than a single language. The structure of recognizable families is investigated and a necessary and sufficient condition for a family of languages to be recognizable is obtained. Some operations on families of languages are also examined in order to determine whether they preserve recognizability or not.

## 1. INTRODUCTION AND PRELIMINARIES

Finite automata were originally developed as abstract models of discrete finite-state devices that change their internal states in response to inputs. There is also another motivation, which comes from the theory of state-space problem solving in artificial and human intelligence. The latter motivation explains the idea behind the new model introduced in the present paper.

The state-space search methods are well-known techniques in problem solving (cf. [4], Chapter 3). In principal, a finite sequence of elementary operators is constructed, leading from a given initial state to one of the goal states of a particular problem domain (think about configurations on a chessboard or about situations in a robot's environment; the operators are chessplayers's moves and actions of the robot, respectively). Let us call any such sequence a plan (because it serves as a prescription how to achieve the goal). We shall restrict ourselves to the case when the set of all states is finite: it is the most usual case for applications. Now, the notion of a finite automaton is a natural and obvious mathematical abstraction of such a state space. One just interprets the input alphabet of the automaton as the set (of names) of elementary operators (the operators are usually partial functions, but this makes no

principal difference). Thus the set of all plans in a particular problem domain is nothing else than the regular language recognized by the corresponding finite automaton – and this seems to be all in which one field can contribute to the other.

However, there exists a radical generalization of the concept of a plan which is important in problem solving and quite appealing from the automata-theoretic point of view. It is the case of so called "branching" or "conditional" plans. The above concept of a plan as a sequence does not formally distinguish the plan itself from the *execution sequence*, i.e., the sequence of real actions (or moves) physically performed when the goal is approached. Assume now that a lack of prior information about some particular future state makes it necessary to consider two or more outgoing operators in parallel (hence the term "branching"). This occurs, in particular, when the result of some operator is not uniquely predictable (cf. [1]). In such a case we have to consider plans that are actually *sets of finite sequences* rather than a single sequence. Furthermore, there are no goal states but only "goal opportunities": it may not be clear whether a given state is a goal or not. Consequently, prefixes of sequences of the plan may also belong to the plan. It is obvious that now one has to distinguish strictly the plan (as a set) and the execution sequence (one of the elements of the set).

This generalized case occurs, for instance, in the image-space approach to problem solving [6] related to the idea of STRIPS [2]. Roughly speaking, instead of states we have *images*, which comprise only a partial knowledge about the real state of affairs. Formally, an image can be represented by a theory in first-order predicate calculus; the knowledge about the world consists of the assertions that can be proved as theorems in the image. Instead of goal states we have a *goal formula*: the task is to change the world to make this formula true. Thus when planning one searches for a proper sequence of operators leading to an image in which the goal formula is a theorem. The applicability of the operators is specified by their conditions, furmulas which may or may not be provable. Sometimes, of course, only a disjunction of such conditions is provable in a particular image, and consequently no single operator can be found applicable. This disjunction indicates a set of operators to be considered in parallel at the time of planning. (During the execution, of course, only one of the operators will be used according to real circumstances.) The situation is even more intricated when one can prove the goal formula only in disjunction with one or more operator conditions. This reflects the mentioned "goal opportunity". Thus, at the time of planning, the success has to be considered in parallel with some continuing operators. (We suggest the reader to return to this discussion after reading first two definitions in the next section.)

Mathematically we can treat plans as formal languages, that is, sets of finite sequences. The set of all plans for a given problem is a family of languages; families of this type will be our main concern. (Perhaps a more intuitive mathematical model for a branching plan would be a tree in the graph-theoretical sense, but there is a formal reason for the language approach. In any case, a tree can be associated in a natural way with any given language, or at least with its prefix closure.)

The generalized state space (e.g., the image space) will find its formal conterpart in the so called *finite branching automaton*. The difference from conventional finite automata lays mainly in the form of acceptance: instead of strings the new device accepts languages and instead of languages it recognizes families of languages. From a different viewpoint these peculiar automata are a generalization of AND-OR graphs, another well-known concept from heuristic problem solving.

This paper consists of five sections. After this introduction, in Sec. 2 we begin with an exposition of finite branching automata and recognizable families of languages. Some interesting structural properties of recognizable families are then investigated in Sec. 3. Their characterization, which is our main result, is established in Sec. 4. In Sec. 5 we conclude studying some closure properties of the class of all recognizable families (with mostly negative results).

Apart from the motivation, the paper should be considered more as a treatise in "unusual automata theory" than as a contribution to problem-solving techniques. The paper is self-contained; however, a reader with a prior knowledge of automata theory will find the reading easier.

An *alphabet* is a finite nonempty set of objects called *letters* (usually denoted $a$, $b$, $c$, ...). For formal reasons we assume that all alphabets are subsets of a fixed infinite set of letters. In the following, $\Sigma$ will always denote an arbitrary, but in a local context fixed alphabet. We denote $\Sigma^*$ the free monoid generated by $\Sigma$ under concatenation. The identity element in $\Sigma^*$ is the *empty string* $\Lambda \in \Sigma^*$. Elements of $\Sigma^*$ are finite sequences of letters. They are called *strings* and usually denoted $u, v, w, \ldots$ For $u \in \Sigma^*$, $\lg(u)$ denotes the length of $u$ (the number of occurrences of letters in $u$). In particular, $\lg(\Lambda) = 0$.

A *language* (*over* $\Sigma$) is any subset $L \subseteq \Sigma^*$. We use the customary notation for *concatenation* $(L_1L_2 := \{uv \mid u \in L_1, v \in L_2\})$, *power* $(L^0 := \{\Lambda\},\ L^n := LL^{n-1})$, ant the *star closure* $(L := \bigcup_{n=0}^{\infty} L^n)$ of languages, including the usual simplifying conventions, e.g., for $a, b \in \Sigma$, $\Lambda \cup a^*b$ is the same as $\{\Lambda\} \cup \{a\}^* \{b\}$.

We denote

$$\mathscr{L}(\Sigma) := 2^{\Sigma^*} - \{\emptyset\}$$

the set of all nonempty languages over $\Sigma$. A *family of languages over* $\Sigma$ (in short, a *family*) is any subset $X \subseteq \mathscr{L}(\Sigma)$. Note that we admit empty families but not families with empty elements (this restriction yields some formal simplifications and conforms with our motivation: an empty plan is the same as no plan at all).

Let $u, v \in \Sigma^*$; $u$ is a *prefix* (*proper prefix*) of $v$, denoted $u \leqq v$ ($u < v$) iff $v = uw$ for some $w \in \Sigma^*$ ($w \in \Sigma^* - \{\Lambda\}$). For $L \subseteq \Sigma^*$ we define the *prefix closure* of $L$ as

$$\mathrm{Pref}(L) := \{u \in \Sigma^* \mid (\exists v \in L)\, u \leqq v\},$$

ant the set of *first letters* of $L$ as

$$\mathrm{Fst}(L) := \{a \in \Sigma \mid (\exists v \in L)\ av \in L\} = \mathrm{Pref}(L) \cap \Sigma .$$

For $L \subseteq \Sigma^*$ and $u \in \Sigma^*$ we define the *derivative of L with resp. to u* [5] as the language

$$\partial_u L := \{w \in \Sigma^* \mid uw \in L\} .$$

Note that $u \in \mathrm{Pref}(L)$ iff $\partial_u L \neq \emptyset$, and $u \in L$ iff $\Lambda \in \partial_u L$.

A *finite automaton* is a quintuple

$$\mathscr{A} := \langle Q, \Sigma, \delta, q_0, F \rangle ,$$

where $Q$ is a finite nonempty set (of *states*), $\Sigma$ is an alphabet, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. We extend $\delta$ in the usual way to $\delta : Q \times \Sigma^* \to Q$ so that $\delta(q, \Lambda) = q$ and $\delta(q, va) = \delta(\delta(q, v), a)$. Sometimes we omit $F$ to obtain a *finite automaton without final states*, $\mathscr{A}^0 := \langle Q, \Sigma, \delta, q_0 \rangle$.

A string $u \in \Sigma^*$ is *accepted* by a finite automaton $\mathscr{A}$ iff $\delta(q_0, u) \in F$. We denote $T(\mathscr{A})$ the language of all strings accepted by $\mathscr{A}$ and say that $\mathscr{A}$ *recognizes* $L$ iff $L = T(\mathscr{A})$. We call $L \subseteq \Sigma^*$ *regular* iff $L = T(\mathscr{A})$ for some $\mathscr{A}$.

The reader will find further details about finite automata and regular languages in literature, e.g. [5] (a survey on regular languages is in [3]).

## 2. FINITE BRANCHING AUTOMATA: THE DEFINITION AND BASIC PROPERTIES

**Definition 2.1.** A *finite branching automaton is a quintuple*

$$\mathscr{B} := \langle Q, \Sigma, \delta, q_0, B \rangle , \tag{1}$$

where $\langle Q, \Sigma, \delta, q_0 \rangle$ is an ordinary finite automaton without final states and $B := \langle B_0, B_1 \rangle$ is a pair of subsets of $Q \times 2^\Sigma$ (the *transient* and the *terminal branching relations*).

While finite automata accept strings and recognize languages, finite branching automata accept languages and recognize families of languages (we make a formal distinction between acceptance and recognizability).

**Definition 2.2.** A language $L \in \mathscr{L}(\Sigma)$ is *accepted* by a finite branching automaton of the form (1) iff for each $w \in \mathrm{Pref}(L) - L$

$$(\delta(q_0, w), \mathrm{Fst}(\partial_w L)) \in B_0 \tag{2}$$

and for each $w \in L$

$$(\delta(q_0, w), \mathrm{Fst}\,(\partial_w L)) \in B_1 \,. \tag{3}$$

We denote $T(\mathscr{B})$ the family of all languages accepted by $\mathscr{B}$. Since we exclude the empty language, $T(\mathscr{B}) \subseteq \mathscr{L}(\Sigma)$.

**Definition 2.3.** A family $X \subseteq \mathscr{L}(\Sigma)$ is *recognizable* iff $X = T(\mathscr{B})$ for some finite branching automaton $\mathscr{B}$.

In order to facilitate pictorial presentation of finite branching automata we adopt the following conventions. Given $\mathscr{B} = \langle Q, \Sigma, \delta, q_0, B\rangle = \langle \mathscr{A}^0, B\rangle$ we first draw the usual state graph for $\mathscr{A}^0$ (circles for states and edges for transitions). Then for each state $q$ we indicate the corresponding transient and terminal branching relations by chaining the appropriate edges from $q$ with white and black dots, respectively (if $(q, \emptyset) \in B_i$, a dot is placed inside the circle for $q$).

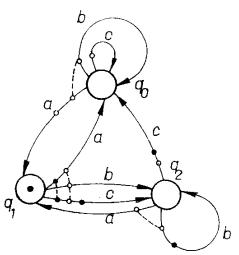


**Fig. 1.**

**Example 1.** Fig. 1 illustrates a finite branching automaton $\mathscr{B}$ with

$$B_0 = \{(q_0, \{a\}), (q_0, \{a, b\}), (q_0, \{c\}), (q_1, \{a, b, c\}),$$
$$(q_2, \{a, b\}), (q_2, \{c\})\} \,,$$
$$B_1 = \{(q_1, \emptyset), (q_1, \{a, c\}), (q_1, \{c\}), (q_2, \{b\}), (q_2, \{c\})\} \,.$$

As an exercise the reader may verify that the following regular sets belong to $T(\mathscr{B})$:

$$L_1 = \{a, acb^n ca\} \cap \{acb^i \mid 0 \leqq i \leqq n\} \,, \quad (n \geqq 0) \,,$$
$$L_2 = b^* a \,,$$
$$L_3 = ca \cap cac(a \cap bb^*) \,.$$

(Note that $a \in L_1$ because $(\delta(q_0, a), \{c\}) \in B_1$ and *not* because $(\delta(q_0, a), \emptyset) \in B_1$.)

There are two natural ways of regarding ordinary finite automata as a special case of finite branching automata. Let us associate with a given finite automaton $\mathscr{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ two finite branching automata $\mathscr{B}^{(1)}_{\mathscr{A}}$ and $\mathscr{B}^{(2)}_{\mathscr{A}}$ in the following way.

$$\mathscr{B}^{(1)} := \langle Q, \Sigma, \delta, q_0, B^{(1)}\rangle ,$$

where $B^{(1)}_0 := \{(q, \{a\}) \mid q \in Q, a \in \Sigma\}$ and $B^{(1)}_1 := \{(q, \emptyset) \mid q \in F\}$.

Define $A_q := \{a \in \Sigma \mid (\exists u \in \Sigma^*)\, \delta(q, au) \in F\}$. Then

$$\mathscr{B}^{(2)} := \langle Q, \Sigma, \delta, q_0, B^{(2)}\rangle ,$$

where $B^{(2)}_0 := \{(q, A_q) \mid q \in Q - F\}$ and $B^{(2)}_1 := \{(q, A_q) \mid q \in F\}$.

**Theorem 2.1.** Let $\mathscr{A}$ be a finite automaton and let $\mathscr{B}^{(1)}_{\mathscr{A}}$ and $\mathscr{B}^{(2)}_{\mathscr{A}}$ be two finite branching automata associated with $\mathscr{A}$ in the above way. Then

$$T(\mathscr{B}^{(1)}_{\mathscr{A}}) = \{\{u\} \mid u \in T(\mathscr{A})\} .$$

Let $T(\mathscr{A}) \neq \emptyset$. Then

$$T(\mathscr{B}^{(2)}_{\mathscr{A}}) = \{T(\mathscr{A})\} .$$

Proof. I. For the first part of the theorem let us first note that each $L \in T(\mathscr{B}^{(1)}_{\mathscr{A}})$ is a singleton. Indeed, combining the definition of $B^{(1)}$ with Definition 2.2 we deduce that $\mathrm{Fst}\,(\partial_w L)$ is a singleton if $w \in \mathrm{Pref}\,(L) - L$, and empty otherwise.

Now we shall show that $u \in T(\mathscr{A})$ iff $\{u\} \in T(\mathscr{B}^{(1)}_{\mathscr{A}})$. We have

$$u \in T(\mathscr{A}) \text{ iff } \delta(q_0, u) \in F \text{ iff } (\delta(q_0, u), \emptyset) \in B^{(1)}_1 .$$

But this holds iff $\{u\} \in T(\mathscr{B}^{(1)}_{\mathscr{A}})$ since for each $w < u$

$$(\delta(q_0, w), \mathrm{Fst}\,(\partial_w\{u\})) \in B^{(1)}_0 .$$

II. For the second part of the theorem we first show that $T(\mathscr{A}) \in T(\mathscr{B}^{(2)}_{\mathscr{A}})$. Let $w \in \Sigma^*$ and denote $q_w := \delta(q_0, w)$. Then

$$\begin{aligned}\mathrm{Fst}\,(\partial_w T(\mathscr{A})) &= \{a \in \Sigma \mid (\exists u \in \Sigma^*)\, wau \in T(\mathscr{A})\} = \\ &= \{a \in \Sigma \mid (\exists u \in \Sigma^*)\, \delta(q_w, au) \in F\} = \\ &= A_{q_w} .\end{aligned}$$

Therefore

$$(\delta(q_0, w), \mathrm{Fst}\,(\partial_w T(\mathscr{A}))) = (q_w, A_{q_w}) ,$$

and by definition of $B^{(2)}$, this pair belongs to $B^{(2)}_0$ if $q_w \in Q - F$, i.e., if $w \notin T(\mathscr{A})$, hence (2); it belongs to $B^{(2)}_1$ if $q_w \in F$, i.e., if $w \in T(\mathscr{A})$, hence (3).

It remains to show that $T(\mathscr{B}^{(2)}_{\mathscr{A}})$ cannot contain more than one language. To see this let $L_1, L_2 \in T(\mathscr{B}^{(2)}_{\mathscr{A}})$, $L_1 \neq L_2$. Assume, without loss of generality, that $L_1 - L_2 \neq$

$\neq \emptyset$ and choose $w \in L_1 - L_2$. Let $u$ be the maximal prefix of $w$ such that $u \in \mathrm{Pref}(L_2)$ (this prefix exists: it is at least $\Lambda$ since $L_2 \neq \emptyset$ implies $\Lambda \in \mathrm{Pref}(L_2)$). Denote $q_u := \delta(q_0, u)$. Since $u \in \mathrm{Pref}(L_1) \cap \mathrm{Pref}(L_2)$ we have, by Definition 2.2, 

$$(q_u, \mathrm{Fst}(\partial_u L_i)) \in B_0 \cup B_1 \quad (i = 1, 2).$$

Hence $\mathrm{Fst}(\partial_u L_1) = A_{q_u} = \mathrm{Fst}(\partial_u L_2)$.

Now if $u \neq w$, we could take a longer prefix of $w$ with the above properties, contrary to the claim of maximality of $u$. Thus $u = w$. But now, by (3),

$$(q_u, \mathrm{Fst}(\partial_u L_1)) \in B_1$$

since $u \in L_1$, while

$$(q_u, \mathrm{Fst}(\partial_u L_2)) \notin B_1$$

since $u \notin L_2$. Hence $\mathrm{Fst}(\partial_u L_1) \neq \mathrm{Fst}(\partial_u L_2)$, a contradiction. Q. E. D.

**Example 2.** Using the above construction we obtain for the regular set $a^*$ two finite branching automata $\mathscr{B}^{(1)}$ and $\mathscr{B}^{(2)}$ as illustrated in Fig. 2. We have $T(\mathscr{B}^{(1)}) = \{\{a^n\} \mid n \geq 0\}$ and $T(\mathscr{B}^{(2)}) = \{a^*\}$ in agreement with Theorem 2.1.

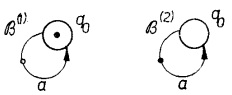


**Fig. 2.**

From the second part of Theorem 2.1 it follows that every nonempty regular language can be accepted by a finite branching automaton. A closer inspection immediately shows that there are also nonregular languages acceptable by these devices. For instance the automaton $\mathscr{B}$ from Fig. 1 accepts, among others, the language

$$L_K := a(cca)^* \cap \{a(cca)^k c \mid k \in K\}$$

for an arbitrary set $K$ of natural numbers. What languages, in general, can be accepted by the finite branching automata?

The following theorem gives a surprisingly simple answer to this question.

**Theorem 2.2.** $\mathscr{L}(\Sigma)$ is recognizable for any $\Sigma$.

Proof. Define $\mathscr{B} := \langle \{q_0\}, \Sigma, \delta, q_0, B \rangle$, where $\delta(q_0, a) = q_0$ for each $a \in \Sigma$ and $B_0 = B_1 = \{(q, A) \mid A \subseteq \Sigma\}$ (for the case of $\Sigma = \{a, b\}$ cf. Fig. 3). A direct check of Definition 2.2 shows that any $L \in \mathscr{L}(\Sigma)$ belongs to $T(\mathscr{B})$. Q. E. D.


**Fig. 3.**

According to this theorem any family $X \subseteq \mathscr{L}(\Sigma)$ is a subset of some recognizable family (namely of $\mathscr{L}(\Sigma)$). Let us call $X$ *strong* iff $\mathscr{L}(\Sigma)$ is the only recognizable family containing $X$ as a subfamily.

**Open problem.** Are there any finite strong families?

We conclude this section with the complete census of all families over one-letter alphabet $\Sigma = \{a\}$ recognizable by a one-state finite branching automaton:

$$\emptyset$$
$$\{\{\Lambda\}\}$$
$$\{\{a^n\} \mid n \geqq 0\}$$
$$\{a^*\}$$
$$\{\{a^i \mid 0 \leqq i \leqq n\} \mid n \geqq 0\}$$
$$\{L \mid L \subseteq a^*, L \text{ infinite}\}$$
$$\{L \mid L \subseteq a^*\}\,.$$

## 3. THE REPLACEMENT PROPERTY

From the fact that finite branching automata can accept very "complex" (e.g., nonrecursive) languages one should not wrongly conclude that there are also very "complex" recognizable families (after reading the proof of Theorem 2.2 one should not call $\mathscr{L}(\Sigma)$ complex). Some of the results in this and the next sections give a hint in this respect: they show that recognizable families, in general, contain lots of "uninteresting" languages (exactly as in conventional automata theory: the regular language $a^*ba^*$ contains besides palindromes also many less impressive strings).

Let us introduce a natural operation on languages consisting in replacing a certain part of one language by another language.

**Definition 3.1.** For every $u \in \Sigma^*$ we define a binary *replacement operator* $R_u$ as follows. For each $L_1, L_2 \subseteq \Sigma^*$,

$$R_u(L_1, L_2) := (L_1 - u\Sigma^*) \cup uL_2\,. \tag{4}$$

Note that $R_u(\emptyset, \emptyset) = \emptyset$, $R_\Lambda(L_1, L_2) = L_2$, and $R_u(L_1, L_2) = R_u(L_1, \emptyset) \cup R_u(\emptyset, L_2)$. Furthermore, for any $u, v \in \Sigma^*$ and $L_1, L_2, L_3 \in \mathscr{L}(\Sigma)$,

$$R_u(R_{uv}(L_1, L_2), L_3) = R_u(L_1, L_3)\,,$$
$$R_{uv}(R_u(L_1, L_2), L_3) = R_u(L_1, R_v(L_2, L_3))\,.$$

We shall be particularly interested in a special application of the replacement operator, namely that of the form $R_u(L_1, \partial_u L_2)$. In problem solving it may be interpreted as "jumping" from one plan to another. The result of such a composition of plans should yield again a genuine plan.

**Definition 3.2.** A family $X \subseteq \mathscr{L}(\Sigma)$ has the *replacement property* iff for each $L_1, L_2 \in X$ and each $u \in \mathrm{Pref}(L_1) \cap \mathrm{Pref}(L_2)$,

$$R_u(L_1, \partial_u L_2) \in X .$$

*Remark.* Because $R_u(L_1, \partial_u L_2) = (L_1 \cap (\Sigma^* - u\Sigma^*)) \cup L_2 \cap (u\Sigma^*)$, any family of languages closed under union and intersection with regular sets has the replacement property. Furthermore, since $R_u(L, \partial_u L) = L$, any family-singleton has trivially the replacement property.

**Lemma 3.1.** Let $\mathscr{B}$ be a finite branching automaton of the form (1). Let $L_1, L_2 \in T(\mathscr{B})$ and let $u \in \mathrm{Pref}(L_1)$ and $v \in \mathrm{Pref}(L_2)$ be such that

$$\delta(q_0, u) = \delta(q_0, v) .$$

Then

$$R_u(L_1, \partial_v L_2) \in T(\mathscr{B}) .$$

Proof. Consider

$$L := R_u(L_1, \partial_v L_2) = (L_1 - u\Sigma^*) \cap u\, \partial_v L_2 .$$

Here $L \neq \emptyset$ since $\partial_v L_2 \neq \emptyset$. Let $w \in \mathrm{Pref}(L)$. Two cases arise.

Case 1: $u \not\leqq w$. Then $w \in \mathrm{Pref}(L_1)$, $\mathrm{Fst}(\partial_w L) = \mathrm{Fst}(\partial_w L_1)$ and $w \in L$ iff $w \in L_1$ (we have tacitly used the assumption that $u \in \mathrm{Pref}(L_1)$). That is,

$$(\delta(q_0, w), \mathrm{Fst}(\partial_w L)) = (\delta(q_0, w), \mathrm{Fst}(\partial_w L_1))$$

and since $L_1 \in T(\mathscr{B})$, this pair belongs to $B_0$ if $w \notin L_1$, i.e., if $w \notin L$, hence (2); it belongs to $B_1$ if $w \in L_1$, i.e., if $w \in L$, hence (3).

Case 2: $u \leqq w$, that is, $w = uw'$ for some $w' \in \Sigma^*$. Then

$$\partial_w L = \partial_{uw'} \partial_v L_2 = \partial_{w'} \partial_v L_2 = \partial_{vw'} L_2 .$$

This implies that $vw' \in \mathrm{Pref}(L_2)$ and $w \in L$ iff $vw' \in L_2$. Thus

$$\begin{aligned}(\delta(q_0, w), \mathrm{Fst}(\partial_w L)) &= (\delta(q_0, uw'), \mathrm{Fst}(\partial_{vw'} L_2)) = \\ &= (\delta(q_0, vw'), \mathrm{Fst}(\partial_{vw'} L_2)) ,\end{aligned}$$

where we used the assumption about $u$ and $v$. Since $L_2 \in T(\mathscr{B})$ the above pair belongs to $B_0$ if $vw' \notin L_2$, i.e. if $w \notin L$, hence (2); it belongs to $B_1$ if $vw' \in L_2$, i.e. if $w \in L$, hence (1).

Thus $L$ satisfies Definition 2.2 and $L \in T(\mathscr{B})$. Q. E. D.

By setting $u = v$ in Lemma 3.1 we obtain immediately

**Theorem 3.1.** Every recognizable family has the replacement property.

Furthermore, using the finiteness of $Q$ we obtain the following two results.

**Theorem 3.2.** For every recognizable family $X$ there exists a constant $n \geqq 1$ such that for any $L \in X$ and any $K \subseteq \mathrm{Pref}(L)$, if $|K| > n$ then there exist two distinct strings $u, v \in K$ such that

$$R_u(L, \partial_v L) \in X \tag{5}$$

and

$$R_v(L, \partial_u L) \in X\,. \tag{6}$$

(Note that the requirement of $u \neq v$ makes the theorem nontrivial: as we already noted, $R_u(L, \partial_u L) = L \in X$.)

Proof. $X = T(\mathscr{B})$ for some $\mathscr{B}$ of the form (1). Set $n := |Q|$. Let $L \in X$ and $K \subseteq \mathrm{Pref}(L)$ such that $|K| > n$ (there is nothing to prove if such $L$ and $K$ do not exist). Since $|K| > |Q|$ there are two distinct strings $u, v \in K$ such that

$$\delta(q_0, u) = \delta(q_0, v)\,.$$

Setting $L_1 = L_2 = L$ in Lemma 3.1 we obtain immediately (5) and (6). Q. E. D.

The last theorem of this section is an analogy to the "pumping lemma" known from conventional automata theory.

**Theorem 3.3.** For every recognizable family $X$ there exists a constant $n \geqq 1$ such that for any $L \in X$ if $L$ contains a string $w$ of length $|w| > n$, then there are three languages $L_1, L_2, L_3$ and two strings $u, v_0$ such that

$$uv_0 \leqq w\,, \tag{7}$$

$$L = L_1 \cup uL_2 \cup uv_0L_3\,, \tag{8}$$

and for each $m \geqq 0$

$$L_1 \cup \bigcup_{i=0}^{m-1} uv_i^0L_2 \cup uv_0^mL_3 \in X\,. \tag{9}$$

Proof. Similarly as in the proof of Theorem 3.2, consider $X = T(\mathscr{B})$ and set $n := |Q|$. Since $|w| > |Q|$ there are two strings $u, v \in \Sigma^*$ such that

$$v = uv_0 \leqq w$$

for some $v_0 \neq \Lambda$, and

$$\delta(q_0, u) = \delta(q_0, v)\,. \tag{10}$$

Define

$$L_1 := L - u\Sigma^* = L - u\,\partial_u L,$$
$$L_2 := \partial_u L - v_0\Sigma^* = \partial_u L - v_0\,\partial_{uv_0} L,$$
$$L_3 := \partial_{uv_0} L.$$

We have

$$L_1 \cup uL_2 \cup uv_0L_3 = (L - u\,\partial_u L) \cup (u\,\partial_u L - uv_0\,\partial_{uv_0} L) \cup uv_0\,\partial_{uv_0} L = L.$$

Thus we can satisfy (7) and (8). It remains to prove (9). For $m \geqq 0$ define

$$L^{(m)} := L_1 \cup \bigcup_{i=0}^{m-1} uv_0^i L_2 \cup uv_0^m L_3 . \tag{11}$$

We have

$$L^{(0)} = L_1 \cup uL_3 = (L - u\Sigma^*) \cup u\,\partial_{uv_0} L = R_u(L, \partial_v L)$$

and thus using (10) and Lemma 3.1, $L^{(0)} \in X$. For $m \geqq 1$ we prove (9) by induction on $m$.

Basis: $m = 1$. We have

$$L^{(1)} = L_1 \cup uL_2 \cup uv_0L_3 = L \in X .$$

Inductive step: Let $m \geqq 1$ and assume $L^{(m)} \in X$. First we show that

$$L^{(m+1)} = R_v(L^{(m)}, \partial_u L^{(m)}) , \tag{12}$$

Indeed, by substituting (11) into (4) we compute

$$\begin{aligned} R_v(L^{(m)}, \partial_u L^{(m)}) &= (L^{(m)} - uv_0\Sigma^*) \cup uv_0\,\partial_u L^{(m)} = \\ &= (L_1 \cup uL_2) \cup \bigcup_{i=0}^{m-1} uv_0^{i+1} L_2 \cup uv_0^{m+1} L_3 = \\ &= L_1 \cup \bigcup_{i=0}^{m} uv_0^i L_2 \cup uv_0^{m+1} L_3 = \\ &= L^{(m+1)} . \end{aligned}$$

Note that $L_3 \neq \emptyset$, hence $u, v \in \mathrm{Pref}(L^{(m+1)})$. Now using (10), the inductive assumption ($L^{(m)} \in X$), and Lemma 3.1, together with (12), we conclude that $L^{(m+1)} \in X$. Q. E. D.

## 4. CHARACTERIZATION OF RECOGNIZABLE FAMILIES

In this section we shall give the necessary and sufficient conditions for a family of languages to be recognizable.

We start with a well-known result from conventional automata theory.

**Fact.** A language $L \subseteq \Sigma^*$ is regular iff the set $\{\partial_w L \mid w \in \Sigma^*\}$ of all its derivatives is finite.

Our first objective is to show that a similar condition is necessary (but not sufficient) for recognizability by finite branching automata.

**Definition 4.1.** Let $X \subseteq \mathscr{L}(\Sigma)$ and $w \in \Sigma^*$. We define

$$\partial_w X := \{\partial_w L \mid L \in X\} - \{\emptyset\}, \tag{13}$$

$$\mathscr{D}(X) := \{\partial_w X \mid w \in \Sigma^*\}. \tag{14}$$

We shall call the family $\partial_w X$ the *derivative of $X$ with resp. to $w$.*

**Theorem 4.1.** If $X$ is a recognizable family then $\mathscr{D}(X)$ is finite.

Proof. Let $X = T(\mathscr{B})$ for $\mathscr{B} = \langle Q, \Sigma, \delta, q_0, B\rangle$. For each $u \in \Sigma^*$ define $q_u := \delta(q_0, u)$ and $\mathscr{B}_u := \langle Q, \Sigma, \delta, q_u, B\rangle$.

**Assertion 1.** For each $u \in \Sigma^*$, $\partial_u X \subseteq T(\mathscr{B}_u)$.

This is trivial if $\partial_u X = \emptyset$. Let $L \in \partial_u X$. By (13) $\emptyset \neq L = \partial_u L'$ for some $L' \in X$. Let $w \in \mathrm{Pref}(L)$. Then $uw \in \mathrm{Pref}(L')$ and

$$(\delta(q_0, w), \mathrm{Fst}(\partial_w L)) = (\delta(q_0, uw), \mathrm{Fst}(\partial_{uw} L')).$$

By (2) from Definition 2.2 used for $uw$, this pair belongs to $B_0$ if $uw \notin L'$, that is, if $w \notin L$, hence (2) holds for $w$. By (3) it belongs to $B_1$ if $uw \in L'$, i.e., if $w \in L$, hence (3) for $w$. Thus $L \in T(\mathscr{B}_u)$ which proves the assertion.

**Assertion 2.** For each $u \in \Sigma^*$, if $\partial_u X \neq \emptyset$ then $\partial_u X = T(\mathscr{B}_u)$.

Assume $\partial_u X \neq \emptyset$. In view of Assertion 1 it is enough to show that $T(\mathscr{B}_u) \subseteq \partial_u(X)$. This is trivial if $T(\mathscr{B}_u) = \emptyset$. Let $L_2 \in T(\mathscr{B}_u)$. According to (13) all we need is to find $L \in X$ such that $L_2 = \partial_u L$.

As $\partial_u X \neq \emptyset$ we can choose $L_1 \in X$ such that $u \in \mathrm{Pref}(L_1)$. Define

$$L := R_u(L_1, L_2). \tag{15}$$

Clearly $\partial_u L = L_2$ and $L \neq \emptyset$ since $L_2 \neq \emptyset$.

It remains to show that $L \in X$. For this let $w \in \mathrm{Pref}(L)$. Two cases arise.

Case 1: $u \not\leq w$. Then $w \in \mathrm{Pref}(L_1)$ and the proof that (2) and (3) of Definition 2.2 hold is identical to Case 1 in the proof of Lemma 3.1.

Case 2: $u \leq w$, that is, $w = uv$ for some $v \in \mathrm{Pref}(L_2)$.
By (15) $\partial_w L = \partial_v L_2$ and $w \in L$ iff $v \in L_2$. Thus

$$(\delta(q_0, w), \mathrm{Fst}(\partial_w L)) = (\delta(q_0, uv), \mathrm{Fst}(\partial_v L_2)) =$$
$$= (\delta(q_u, v), \mathrm{Fst}(\partial_v L_2)).$$

Now, since $L_2 \in T(\mathscr{B}_u)$, this pair belongs to $B_0$ if $v \notin L_2$, i.e., if $w \notin L$ and it belongs to $B_1$ if $v \in L_2$, i.e., if $w \in L$. Therefore $L \in X$ which ends the proof of Assertion 2.

Now, since $Q$ is finite there is a finite number of distinct $\mathscr{B}_u$ and hence a finite number of distinct families $T(\mathscr{B}_u)$. Therefore, by Assertion 2, $\{\partial_u(X) \mid u \in \Sigma^*\} = \mathscr{D}(X)$ is finite. Q. E. D.

*Remark.* The assumption of $\partial_u X \neq \emptyset$ in Assertion 2 is indeed necessary. For $\mathscr{B}$ in Fig. 4 we have $T(\mathscr{B}) = \{\{b\}\}$, $T(\mathscr{B}_a) = \{\{a\}\}$ and $\partial_a T(\mathscr{B}) = \emptyset$.

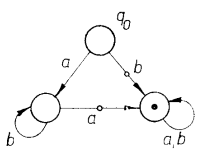


**Fig. 4.**

As a corollary to Theorem 4.1 we obtain the regularity of the union of all languages in a recognizable family $X$. We use the notation

$$\cup X := \bigcup_{L \in X} L.$$

**Theorem 4.2.** If $X$ is recognizable then $\cup X$ is regular.

Proof. By Theorem 4.1, $\mathscr{D}(X)$ is finite. Thus $Y := \{\cup \partial_w X \mid w \in \Sigma^*\}$ is also finite. Since

$$\begin{aligned} \cup \partial_w X &= \bigcup_{L \in X} \partial_w L = \\ &= \bigcup_{L \in X} \{u \mid wu \in L\} = \\ &= \{u \mid wu \in \cup X\} = \\ &= \partial_w (\cup X), \end{aligned}$$

we have $\{\partial_w (\cup X) \mid w \in \Sigma^*\} = Y$ finite. Hence $\cup X$ is regular. Q. E. D.

**Corollary.** A singleton $X = \{L\}$ is recognizable iff $L$ is a nonempty regular language.

In general, the converse to Theorem 4.2 does not hold: let $X := \{L, \Sigma^* - L\}$ where $L$ is not regular. Then $\cup X = \Sigma^*$ is regular while $\mathscr{D}(X)$ is infinite.

Given a finite automaton $\mathscr{A}$ it is not hard to construct a finite branching automaton $\mathscr{B}$ such that $\cup T(\mathscr{B}) = T(\mathscr{A})$. In fact, Theorem 2.1 provides two such constructions.

We leave as an exercise to find a direct construction of a finite automaton $\mathscr{A}$ from a given finite branching automaton $\mathscr{B}$, with $T(\mathscr{A}) = \cup T(\mathscr{B})$.

Let us return to the characterization problem. Because the finiteness of $\mathscr{D}(X)$ does not yield the replacement property (cf., e.g., $X = \{\{aa, b\}, \{ab\}\}$) it is not a sufficient condition for recognizability. In our search for such a condition we explore a certain generalization of the replacement property.

**Definition 4.2.** Let $X \subseteq \mathscr{L}(\Sigma)$ and $L \in \mathscr{L}(\Sigma)$. We say that $L$ is *compatible with* $X$ iff for each $w \in \Sigma^*$ there is a language $L_w \in X$ such that

$$w \in L \quad \text{iff} \quad w \in L_w \tag{16}$$

and

$$\text{Fst}\,(\partial_w L) = \text{Fst}\,(\partial_w L_w)\,. \tag{17}$$

We denote $C(X)$ the family of all languages compatible with $X$.

**Example 3.** Let $X := \{\{aa, ba\}, \{ab, bb\}\}$. Then $\{aa, bb\} \in C(X)$ but $\{aa, ab\} \notin C(X)$.

It can be easily observed, that $X \subseteq C(X)$ and $C(X) = CC(X)$; however, it is not true, in general, that $X = C(X)$.

**Definition 4.3.** A family $X \subseteq \mathscr{L}(\Sigma)$ is *self-compatible* iff $X = C(X)$.

**Theorem 4.3.** Every recognizable family is self-compatible.

Proof. Let $X = T(\mathscr{B})$ for $\mathscr{B} = \langle Q, \Sigma, \delta, q_0, B\rangle$ and let $L \in C(X)$. For $w \in \Sigma^*$ let $L_w \in X$ be the language satisfying (16) and (17). Thus $L = \{w \mid w \in L_w\}$. To show that $L \in X$ let $w \in \text{Pref}\,(L)$. By (17)

$$(\delta(q_0, w), \text{Fst}\,(\partial_w L)) = (\delta(q_0, w), \text{Fst}\,(\partial_w L_w))\,.$$

By (16) and since $L_w \in X$ this pair belongs to $B_0$ if $w \notin L_w$, i.e., if $w \notin L$; it belongs to $B_1$ if $w \in L_w$, i.e., if $w \in L$. We conclude that $L \in X$. Q. E. D.

The following lemma makes sometimes easier to prove compatibility by restricting Definition 4.2 only to $w \in \text{Pref}\,(L)$.

**Lemma 4.1.** Let $X \subseteq \mathscr{L}(\Sigma)$ and $L \in \mathscr{L}(\Sigma)$. Assume that for each $w \in \text{Pref}\,(L)$ there exists $L_w \in X$ satisfying (16) and (17). Then $L \in C(X)$.

Proof. Under the assumptions of the lemma let $w \notin \text{Pref}\,(L)$. Then there exist $u \in \Sigma^*$ and $a \in \Sigma$ such that $ua \leqq w$, $u \in \text{Pref}\,(L)$ but $ua \notin \text{Pref}\,(L)$ (this existence follows from the fact that $\Lambda \in \text{Pref}\,(L)$ since $L \neq \emptyset$). By the assumption, there is $L_u \in X$ satisfying (16) and (17) (for $u$). Set $L_w := L_u$. Note that $w \notin \text{Pref}\,(L_w)$; other-

wise we would have $ua \in \mathrm{Pref}(L_u)$ and hence $a \in \mathrm{Fst}(\partial_u L_u) = \mathrm{Fst}(\partial_u L)$ and thus $ua \in \mathrm{Pref}(L)$, a contradiction. Consequently, $w \notin L$ and $w \notin L_w$ which proves (16). 

Furthermore, $\mathrm{Fst}(\partial_w L) = \emptyset$ since $w \notin \mathrm{Pref}(L)$ and $\mathrm{Fst}(\partial_w L_w) = \emptyset$ since $w \notin \mathrm{Pref}(L_w)$. Therefore $\mathrm{Fst}(\partial_w L) = \mathrm{Fst}(\partial_w L_w)$ which proves (17). Q. E. D.

Now we can prove our main theorem.

**Theorem 4.4.** (Characterization Theorem). A family $X \subseteq \mathscr{L}(\Sigma)$ is recognizable iff $X$ is self-compatible and $\mathscr{D}(X)$ is finite.

Proof. The $\Rightarrow$ direction was already established by theorems 4.1 and 4.3. For the $\Leftarrow$ direction assume that $X = C(X)$ and $\mathscr{D}(X)$ is finite. Define the following finite branching automaton:

$$\mathscr{B}_X := \langle \mathscr{D}(X), \Sigma, \delta, X, B \rangle , \tag{18}$$

where $\delta(Y, a) := \partial_a Y$ for each $Y \in \mathscr{D}(X)$, and $a \in \Sigma$ (and hence $\delta(Y, u) = \partial_u Y$ for each $u \in \Sigma^*$);

$$B_0 := \{(Y, \mathrm{Fst}(L)) \mid Y \in \mathscr{D}(X), \Lambda \notin L \in Y\} ; \tag{19}$$

$$B_1 := \{(Y, \mathrm{Fst}(L)) \mid Y \in \mathscr{D}(X), \Lambda \in L \in Y\} . \tag{20}$$

We shall prove that $X = T(\mathscr{B}_X)$.

I. $X \subseteq T(\mathscr{B}_X)$. This is immediate for $X = \emptyset$. Let $L \in X$; we shall show that $L \in T(\mathscr{B}_X)$. Let $u \in \mathrm{Pref}(L)$. Then $\partial_u L \neq \emptyset$ and $\partial_u L \in \partial_u X$. Now,

$$(\delta(q_0, u), \mathrm{Fst}(\partial_u L)) = (\partial_u X, \mathrm{Fst}(\partial_u L))$$

which belongs to $B_0$ if $\Lambda \notin \partial_u L$, i.e., if $u \notin L$; it belongs to $B_1$ if $\Lambda \in \partial_u L$, if $u \in L$. Therefore $L \in T(\mathscr{B}_X)$.

II. $T(\mathscr{B}_X) \subseteq X$. This is immediate for $T(\mathscr{B}) = \emptyset$. Let $L \in T(\mathscr{B}_X)$. To show that $L \in X$ we first prove that $L$ is compatible with $X$ and then use the self-compatibility of $X$.

To apply Lemma 4.1, let $w \in \mathrm{Pref}(L)$. Then $(\partial_w X, \mathrm{Fst}(\partial_w L))$ belongs to $B_0$ if $w \notin L$, and to $B_1$ if $w \in L$. Using (19) and (20),

$$(\partial_w X, \mathrm{Fst}(\partial_w L)) = (\partial_w X, \mathrm{Fst}(L'))$$

for some $L' \in \partial_w X$ such that $\Lambda \in L'$ iff $w \in L$. Find $L_w \in X$ such that $L' = \partial_w L_w$. Consequently,

$$\mathrm{Fst}(\partial_w L) = \mathrm{Fst}(L') = \mathrm{Fst}(\partial_w L_w)$$

which proves (17).

Furthermore, $w \in L$ iff $\Lambda \in L'$ iff $\Lambda \in \partial_w L_w$ iff $w \in L_w$. This proves (16).

Thus by Lemma 4.1, $L \in C(X) = X$. We conclude that $T(\mathscr{B}_X) \subseteq X$ and hence $T(\mathscr{B}_X) = X$. Therefore, $X$ is recognizable. Q. E. D.

*Remark.* The two conditions from Theorem 4.4 which are together necessary and sufficient for recognizability, are mutually independent. Example 3 shows a family $X$ which is not self-compatible but with finite $\mathscr{D}(X)$. An example of a self-compatible family with infinite $\mathscr{D}(X)$ is any singleton $X = \{L\}$ where $L$ is not regular. (It can be easily shown that any singleton is self-compatible.)

**Theorem 4.5.** Any self-compatible family has the replacement property.

Proof. Let $X = C(X) \subseteq \mathscr{L}(\Sigma)$. Let $L_1, L_2 \in X$ and for $u \in \mathrm{Pref}(L_1) \cap \mathrm{Pref}(L_2)$ consider $L := R_u(L_1, \partial_u L_2)$. For any $w \in \mathrm{Pref}(L)$ we can easily satisfy (16) and (17) by

$$L_w := \begin{cases} L_1 & \text{if } u \not\leqq w, \\ L_2 & \text{if } u \leqq w. \end{cases}$$

Therefore by Lemma 4.1, $L \in C(X) = X$. Q. E. D.

**Open problem.** Is the converse also true? More specifically, can the condition of self-compatibility in the characterization theorem be replaced by the replacement property?

The following result is rather easy.

**Theorem 4.6.** Any finite family is self-compatible iff it has the replacement property.

Proof. For $\Rightarrow$ cf. Theorem 4.5. For $\Leftarrow$ let $X \subseteq \mathscr{L}(\Sigma)$ be a finite family with the replacement property. Let $L \in C(X)$ and suppose for contradiction that $L \notin X$. Then there exists $w$ such that either $w \in L$ and $w \notin \bigcup_{L_i \in X} L_i$, or $w \notin L$ and $w \in \bigcap_{L_i \in X} L_i$. But this contradicts to the existence of $L_w \in X$ satisfying (16). Hence $L \in X$. We conclude $C(X) = X$. Q. E. D.

**Corollary.** A finite family $X \subseteq \mathscr{L}(\Sigma)$ is recognizable iff it has the replacement property and $\mathscr{D}(X)$ is finite.

## 5. OPERATIONS ON FAMILIES

In the preceding sections we have been concerned with the internal structure of recognizable families. We now consider some operations on families of languages which do or do not preserve recognizability. We start with one of few positive cases.

**Theorem 5.1.** If $X_1$ and $X_2$ are recognizable families then $X_1 \cap X_2$ is also recognizable.

Proof. Assume $X_i = T(\mathscr{B}_i)$ where

$$\mathscr{B}_i := \langle Q_i, \Sigma_i, \delta_i, {}^iq_0, {}^iB \rangle \quad (i = 1, 2).$$

Let $\Sigma := \Sigma_1 \cup \Sigma_2$. Define a new finite branching automaton

$$\mathscr{B}_1 \times \mathscr{B}_2 := \langle Q_1 \times Q_2, \Sigma, \delta, ({}^1q_0, {}^2q_0), B \rangle ,$$

where for any $q_i \in Q_i$ $(i = 1, 2)$, $a \in \Sigma$, and $A \subseteq \Sigma$

$$\delta((q_1, q_2), a) := (\delta_1(q_1, a), \delta_2(q_2, a)) \tag{21}$$

and for $j = 0, 1$,

$$((q_1, q_2), A) \in B_j \quad \text{iff} \quad (q_1, A) \in {}^1B_j \quad \text{and} \quad (q_2, A) \in {}^2B_j . \tag{22}$$

By (21), for any $L \in \mathscr{L}(\Sigma)$ and $u \in \Sigma^*$,

$$\begin{aligned} &(\delta(({}^1q_0, {}^2q_0), u), \operatorname{Fst}(\partial_u L)) = \\ &= ((\delta_1({}^1q_0, u), \delta_2({}^2q_0, u)), \operatorname{Fst}(\partial_u L)) \end{aligned} \tag{23}$$

and, by (22), this belongs to $B_j$ $(j = 0, 1)$ iff

$$(\delta_i({}^iq_0, u), \operatorname{Fst}(\partial_u L)) \in {}^iB_j \quad (i = 1, 2) . \tag{24}$$

In particular, set $j = 0$ if $u \notin L$ and $j = 1$ if $u \in L$. Then (23) holds for all $u \in \operatorname{Pref}(L)$ iff (24) holds for all $u \in \operatorname{Pref}(L)$. Therefore,

$$L \in T(\mathscr{B}_1 \times \mathscr{B}_2) \quad \text{iff} \quad L \in T(\mathscr{B}_1) \cap T(\mathscr{B}_2) . \qquad \text{Q. E. D.}$$

In problem solving the intersection of families has a natural interpretation: Let $X_1$ be the set of plans for a goal $G_1$ and $X_2$ the set of plans for $G_2$. Then $X_1 \cap X_2$ is precisely the set of plans realizing both $G_1$ and $G_2$ jointly.

For union and complement we have negative results.

**Example 4.** Let $X_1 = \{L_1\}$ and $X_2 = \{L_2\}$ where $L_1 = \{a\}$ and $L_2 = \{\Lambda, aa\}$. Both $X_1$ and $X_2$ are obviously recognizable. However, $X_1 \cup X_2 = \{\{a\}, \{\Lambda, aa\}\}$ fails to satisfy the replacement property since

$$R_a(L_1, \partial_a L_2) = a\, \partial_a L_2 = \{aa\} \notin X_1 \cup X_2 .$$

Hence $X_1 \cup X_2$ is not recognizable.

Thus, while the class of all recognizable families over $\Sigma$ is closed under intersection, it is not closed under union. Hence it cannot be closed under complement $\mathscr{L}(\Sigma) - X$. We have

**Theorem 5.2.** The class of all recognizable families is not closed under union and complement.

Next we consider the concatenation. It can be defined in two different ways, each a natural generalization of language concatenation.

**Definition 5.1.** Let $X_1 \in \mathscr{L}(\Sigma_1)$ and $X_2 \in \mathscr{L}(\Sigma_2)$ be two families. A *strong concatenation* of $X_1$ and $X_2$ is

$$X_1 \,.\, X_2 := \{L_1 L_2 \mid L_1 \in X_1, L_2 \in X_2\} \,. \tag{25}$$

A *weak concatenation* of $X_1$ and $X_2$ is

$$X_1 \circ X_2 := \{ \bigcup_{v \in L_1} vF(v) \mid L_1 \in X_1 \text{ and } F \text{ is a function } L_1 \to X_2 \} \,. \tag{26}$$

**Theorem 5.3.** The class of all recognizable families is not closed under strong concatenation.

This follows from the following example:

**Example 5.** Let $X_1 = \{\{a, b\}\}$ and $X_2 = \{\{b\}, \{c\}\}$. Then $X_1 \,.\, X_2 = \{\{ab, bb\}, \{ac, bc\}\}$ is not recognizable since the replacement property would require also $\{ab, bc\}$ and $\{ac, bb\}$ to belong to $X_1 \,.\, X_2$.

The weak concatenation is somewhat more appealing from the intuitive point of view. In problem solving it corresponds to the chaining of plans in the following manner: Let $X_1$ and $X_2$ be the sets of plans for goals $G_1$ and $G_2$, resp. Then, to obtain a plan realizing first $G_1$ and then $G_2$, one can just take any $L_1 \in X_1$ and concatenate each $v \in L_1$ with some $L_2 \in X_2$. Doing that in all possible ways yields precisely $X_1 \circ X_2$.

**Open problem.** For two recognizable families $X_1$ and $X_2$, is $X_1 \circ X_2$ also recognizable?

There are certain particular cases where the union and weak concatenation preserve recognizability. Let us define

$$\mathrm{Fst}(X) := \bigcup_{L \in X} \mathrm{Fst}(L) \,.$$

**Theorem 5.4.** Let $X_i \subseteq \mathscr{L}(\Sigma_i)$, $i = 1, 2$, be two recognizable families. If $\mathrm{Fst}(X_1) \cap \cap \mathrm{Fst}(X_2) = \emptyset$ then $X_1 \cup X_2$ is recognizable and if $\Sigma_1 \cap \mathrm{Fst}(X_2) = \emptyset$ then $X_1 \circ X_2$ is recognizable.

Proof. For each case we give only a construction of the corresponding finite branching automaton and we omit the tedious but straightforward verification of its correctness.

For $i = 1, 2$ let $X_i = T(\mathscr{B}_i)$, where

$$\mathscr{B}_i := \langle Q_i, \Sigma_i, \delta_i, {}^i q_0, {}^i B \rangle \,.$$

Assume, without loss of generality, that $Q_1 \cap Q_2 = \emptyset$.

I (union). Define

$$\mathscr{B}_1 \cup \mathscr{B}_2 := \langle Q_1 \cup Q_2 \cup \{q_0\}, \Sigma_1 \cup \Sigma_2, \delta, q_0, B\rangle ,$$

where $q_0 \notin Q_1 \cup Q_2$,

$$\delta(q, a) := \begin{cases} \delta_i({}^i q_0, a) & \text{if} \quad q = q_0 \quad \text{and} \quad a \in \mathrm{Fst}(X_i), \\ \delta_i(q, a) & \text{if} \quad q \in Q_1 \quad \text{and} \quad a \in \Sigma_i \end{cases}$$

(in all other cases $\delta$ may be arbitrary), and for $j = 0, 1$,

$$B_j := {}^1B_j \cup {}^2B_j \cup C_j ,$$

where

$$C_j := \{(q_0, A) \mid ({}^i q_0, A) \in {}^iB_j, A \subseteq \mathrm{Fst}(X_i), i \in \{1, 2\}\} .$$

Then

$$T(\mathscr{B}_1 \cup \mathscr{B}_2) = X_1 \cup X_2 .$$

II (weak concatenation). Define

$$\mathscr{B}_1 \circ \mathscr{B}_2 := \langle Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, {}^1q_0, B\rangle ,$$

where

$$\delta(q, a) := \begin{cases} \delta_i(q, a) & \text{if} \quad q \in Q_i \quad \text{and} \quad a \in \Sigma_i , \\ \delta_2({}^2q_0, a) & \text{if} \quad q \in Q_1 \quad \text{and} \quad a \in \mathrm{Fst}(X_2) \end{cases}$$

(in other cases $\delta$ may be arbitrary),

$$B_0 := {}^1B_0 \cup {}^2B_0 \cup D_0 ,$$

$$B_1 := {}^2B_1 \cup D_1 ,$$

where

$$D_0 := \{(q, A_1 \cup A_2) \mid (q, A_1) \in {}^1B_1, ({}^2q_0, A_2) \in {}^2B_0, A_2 \neq \emptyset\} ,$$

$$D_1 := \{(q, A_1 \cup A_2) \mid (a, A_1) \in {}^1B_1, ({}^2q_0, A_2) \in {}^2B_1\} .$$

Then

$$T(\mathscr{B}_1 \circ \mathscr{B}_2) = X_1 \circ X_2 . \qquad \text{Q. E. D.}$$

As we have seen in Examples 4 and 5, the main obstacle for general closure properties was the failure of the resulting family to be self-compatible. Next we modify the operations on families to guarantee the self-compatibility. First we shall prove some auxiliary results.

**Lemma 5.1.** For any $u \in \Sigma^*$ and $X \subseteq \mathscr{L}(\Sigma)$,

$$\partial_u C(X) = C(\partial_u X) . \tag{27}$$

Proof. I. ($\subseteq$). Let $L \in \partial_u C(X)$. Then $L \neq \emptyset$ and $L = \partial_u L'$ for some $L'$ compatible with $X$. For each $w \in \mathrm{Pref}(L')$ let $L'_w$ be the language in $X$ satisfying

$$w \in L' \quad \text{iff} \quad w \in L'_w \tag{28}$$

and

$$\mathrm{Fst}(\partial_u L') = \mathrm{Fst}(\partial_w L'_w) \tag{29}$$

according to Definition 4.2.

Now let $v \in \mathrm{Pref}(L)$ and define $L_v := \partial_u L'_{uv}$. Clearly $L_v \in \partial_u X$. By (28),

$$v \in L \quad \text{iff} \quad uv \in L' \quad \text{iff} \quad uv \in L'_{uv} \quad \text{iff} \quad v \in L_v .$$

By (29),

$$\mathrm{Fst}(\partial_v L) = \mathrm{Fst}(\partial_{uv} L') = \mathrm{Fst}(\partial_{uv} L'_{uv}) = \mathrm{Fst}(\partial_v L_v) .$$

Thus by Definition 4.2, $L \in C(\partial_u X)$.

II. ($\supseteq$). Let $L$ be compatible with $\partial_u X$. Then, in particular, there exists $L_\Lambda \in \partial_u X$ such that $\Lambda \in L$ iff $\Lambda \in L_\Lambda$ and $\mathrm{Fst}(L) = \mathrm{Fst}(L_\Lambda)$. Let $L_0 \in X$ be such that $L_\Lambda = \partial_u L_0$ and consider the language $L' := R_u(L_0, L)$. We shall show that $L'$ is compatible with $X$. Let $w \in \mathrm{Pref}(L')$.

Case 1: $u \not\leqq w$. Then $w \in \mathrm{Pref}(L_0)$ and thus

$$w \in L' \quad \text{iff} \quad w \in L_0$$

and

$$\mathrm{Fst}(\partial_w L') = \mathrm{Fst}(\partial_w L_0) .$$

Case 2: $u \leqq w$, that is, $w = uv$ for some $v \in \mathrm{Pref}(L)$. Since $L$ is compatible with $\partial_u X$, for any such $v$ there are $L_v \in \partial_u X$ and $L'_w \in X$ such that

$$w \in L' \quad \text{iff} \quad v \in L \quad \text{iff} \quad v \in L_v \quad \text{iff} \quad w \in L'_w .$$

Also,

$$\mathrm{Fst}(\partial_w L') = \mathrm{Fst}(\partial_v L) = \mathrm{Fst}(\partial_v L_v) = \mathrm{Fst}(\partial_w L'_w) .$$

Thus $L' \in C(X)$ and since $L = \partial_u L'$ we can conclude that $L \in \partial_u C(X)$. Q. E. D.

We can use the above lemma to show that the derivative preserves recognizability.

**Theorem 5.5.** Let $X \subseteq \mathscr{L}(\Sigma)$ be recognizable. Then for any $u \in \Sigma^*$, $\partial_u X$ is also recognizable.

Proof. Suppose $X$ recognizable, i.e., $X = C(X)$ and $\mathscr{D}(X)$ is finite. Then by Lemma 5.1

$$C(\partial_u X) = \partial_u C(X) = \partial_u X$$

and since $\mathscr{D}(\partial_u X) \subseteq \mathscr{D}(X)$, $\mathscr{D}(\partial_u X)$ is finite. Thus, by the characterization theorem, $\partial_u X$ is recognizable. Q. E. D.

**Lemma 5.2.** For any $X \subseteq \mathscr{L}(\Sigma)$, if $\mathscr{D}(X)$ is finite then $\mathscr{D}(C(X))$ is also finite. 

Proof.

$$\begin{aligned}(C(X)) &= \{\partial_u C(X) \mid u \in \Sigma^*\} \\ &= \{C(\partial_u X) \mid u \in \Sigma^*\} \quad \text{by Lemma 5.1}, \\ &= \{C(Y) \mid Y \in \mathscr{D}(X)\}.\end{aligned}$$

Thus $|\mathscr{D}(C(X))| \leqq |\mathscr{D}(X)|$. Q. E. D.

**Theorem 5.6.** Let $X_1$ and $X_2$ be two recognizable families. Then $C(X_1 \cup X_2)$ and $C(X_1 \,.\, X_2)$ are also recognizable.

Proof. Since, in general, $CC(X) = X$, both families are self-compatible. Using the characterization theorem and Lemma 5.2 it is enough to show that if $\mathscr{D}(X_1)$ and $\mathscr{D}(X_2)$ are finite then also $\mathscr{D}(X_1 \cup X_2)$ and $\mathscr{D}(X_1 \,.\, X_2)$ are finite.

I. $$\begin{aligned}\mathscr{D}(X_1 \cup X_2) &= \{\partial_u(X_1 \cup X_2) \mid u \in \Sigma^*\} = \\ &= \{\partial_u X_1 \cup \partial_u X_2 \mid u \in \Sigma^*\}\end{aligned}$$

and thus

$$|\mathscr{D}(X_1 \cup X_2)| \leqq |\mathscr{D}(X_1)| \,.\, |\mathscr{D}(X_2)|.$$

II. $$\begin{aligned}\mathscr{D}(X_1 \,.\, X_2) &= \{\partial_u(X_1 \,.\, X_2) \mid u \in \Sigma^*\} = \\ &= \{\{\partial_u(L_1 L_2) \mid L_1 \in X_1, L_2 \in X_2\} \mid u \in \Sigma^*\}\end{aligned}$$

and since

$$\partial_u(L_1 L_2) = \begin{cases}(\partial_u L_1)\, L_2 & \text{if} \quad \Lambda \notin L_1, \\ (\partial_u L_1)\, L_2 \cup \partial_u L_2 & \text{if} \quad \Lambda \in L_1,\end{cases}$$

we have

$$|\mathscr{D}(X_1 \,.\, X_2)| \leqq |\mathscr{D}(X_1)| \,.\, |\mathscr{D}(X_2)|. \qquad \text{Q. E. D.}$$

There is a well-known result in conventional automata theory which says that the class of regular languages is the smallest class containing all finite languages and closed under union, concatenation, and star closure. As a consequence, one can specify any regular language by means of a finite formal term (called a regular expression) – cf. [3]. It would be very important, indeed, to have a similar result for the class of families recognizable by finite branching automata:

**Open problem.** Give an algebraic characterization of recognizable families.

Possible candidates for the basic operations might be $C(X \cup Y)$, $C(XY)$ (or $C(X \circ Y)$), and $C(X^*)$ with a suitable definition of $X^*$.

*Acknowledgement.* The idea of this work was greatly influenced by the creative and stimulating atmosphere of the Computer Science Department of the Aarhus University, Denmark, where, during his visit in fall 1973, the author realized the existence of the large, fertile, and inherently inexhaustible field of "unusual automata theory".

(Received March 1, 1974.)

REFERENCES

[1] R. E. Fikes, P. E. Hart, N. J. Nilsson: Some new directions in robot problem solving. In: Machine Intelligence 7 (B. Meltzer, D. Michie, eds.). University Press, Edinburgh 1972.

[2] R. E. Fikes, N. J. Nilsson: STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence *2* (1971), 189—208.

[3] I. M. Havel: The theory of regular events I, II. Kybernetika *5* (1969), 400—419; 520—544.

[4] N. J. Nilsson: Problem-solving methods in artificial intelligence. McGraw-Hill, New York 1971.

[5] A. Salomaa: Theory of automata. Pergamon Press, Oxford 1969.

[6] O. Štěpánková, I. M. Havel: Image space and its relationship to situation calculus. Tech. Rpt. No 9/1973, Institute of Computation Technique, Prague. (Cf. also Proc. Symp. MFCS, High Tatras 1973.)

*Ing. Ivan M. Havel, CSc. Ph.D.: Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 180 76 Praha 8. Czechoslovakia.*