Theoretical Computer Science 32 (1984) 331–337
North-Holland

NOTE

PROJECTION LEMMAS FOR ω-LANGUAGES

Ludwig STAIGER

Zentralinstitut für Kybernetik und Informationsprozesse der Akademie der Wissenschaften, DDR-1086 Berlin, German Democractic Republic

Communicated by M. Nivat
Received March 1984

Abstract. The paper presents projection lemmas relating the infinite behaviour of deterministic and unboundedly branching nondeterministic infinite automata.

Introduction

Projection lemmas have been used to relate in an elegant way the behaviours of deterministic and nondeterministic devices accepting ω-languages, as finite automata [9], finite partial automata [7] or Turing machines [8]. In all these cases the branching of the nondeterministic devices has been bounded in all situations. A different and more difficult problem arises if the branching of the nondeterministic devices is unbounded (though finite in every situation) or even countably infinite (cf. [1]).

We shall prove that the same projection lemma as in the bounded case holds true for finitely branching automata, and is valid with a slight modification in the case of countably branching automata.

1. Preliminaries

For a finite alphabet X, as usual X^* and X^ω are the sets of finite words and infinite sequences on X. Let e denote the empty word in X^*, and by wb we denote the concatenation of $w \in X^*$ and $b \in X^* \cup X^\omega$, $|w|$ is the length of the word $w \in X^*$. If $i \leq |b|$, then $b(i)$ is the ith letter of the string $b \in X^* \cup X^\omega$. Finally, $A(b)$ denotes the set of all finite initial words of $b \in X^* \cup X^\omega$. We regard the product topology in X^ω. This topology makes X^ω (card $X \geq 2$) homeomorphic to the Cantor discontinuum and, hence, a compact space [9]. By $\mathcal{F}$, $\mathcal{F}_\sigma$ and $\mathcal{G}_\delta$ we denote the classes of closed, of countable unions of closed, and of countable intersections of open subsets of X^ω, respectively.

If X, Y are finite alphabets, we consider the product alphabet $X \times Y$ together with the projection $\mathrm{pr}: X \times Y \mapsto X$ defined by $\mathrm{pr}((x, y)) = x$. The obvious extension of pr to $(X \times Y)^* \cup (X \times Y)^\omega$ is also denoted by pr. Clearly, $\mathrm{pr}: (X \times Y)^\omega \mapsto X^\omega$ is a continuous and (since $(X \times Y)^\omega$ is compact) closed mapping [5]. Hence, $\mathscr{F} = \{\mathrm{pr}\, F: F \in \mathscr{F}\}$ and $\mathscr{F}_\sigma = \{\mathrm{pr}\, F: F \in \mathscr{F}_\sigma\}$, whereas $\mathscr{S} = \{\mathrm{pr}\, F: F \in \mathscr{G}_\delta\}$ equals the set of Souslin subsets of X^ω (or sets of the first projective class) [5].

We remark that for $Y = \{0, 1\}$ the set

$$E \overset{\mathrm{df}}{=} ((X \times \{0, 1\})^* \cup (X \times \{1\}))^\omega \tag{1}$$

is in $\mathscr{G}_\delta \setminus \mathscr{F}_\sigma$ (cf. [6]).

2. Automata

An (infinite) automaton (or transition system) $\mathscr{A}$ over X is a quadruple $(Z, Z_0, Z_{\mathrm{fin}}, R)$, where Z is a set of states, $Z_0 \subseteq Z$ and $Z_{\mathrm{fin}} \subseteq Z$ are the sets of initial and final states, resp. $R \subseteq Z \times X \times Z$ is the transition relation s.t. $R(z, x) \overset{\mathrm{df}}{=} \{(z, x, z'): z' \in Z \text{ and } (z, x, z') \in R\}$ is nonempty for all $z \in Z$ and $x \in X$.

An automaton $\mathscr{A}$ is called deterministic if card $Z_0 = 1$ and card $R(z, x) = 1$ for all $z \in Z$ and $x \in X$. In this case we will also write $\mathscr{A} = (Z, z_0, Z_{\mathrm{fin}}, R)$, and $R: Z \times X \mapsto Z$ as a function. We call $\mathscr{A}$ finitely (countably) branching provided Z_0 and $R(z, x)$ are finite (at most countable) sets for $z \in Z$ and $x \in X$. Since for a countably branching automaton $\mathscr{A}$ at most a countable number of states is accessible from the initial states, we can assume its set of states to be at most countable.

For a word $w \in X^*$ we call the finite sequence $z_0, z_1, \ldots, z_{|w|}$ of states a w-computation on $\mathscr{A}$ iff $z_0 \in Z_0$ and $(z_{i-1}, w(i), z_i) \in R$ for $i = 1, \ldots, |w|$. The set of all words $w \in X^*$ having a w-computation with $z_{|w|} \in Z_{\mathrm{fin}}$ is called the language accepted by the automaton $\mathscr{A}$ (abbreviated $T(\mathscr{A})$).

We note in passing that every subset $W \subseteq X^*$ is acceptable by the deterministic automaton (X^*, e, W, R) where $R(w, x) = wx$.

For a sequence $\beta \in X^\omega$ we call an infinite sequence of states $z_0, z_1, \ldots,$ a β-computation on $\mathscr{A}$ iff $z_0 \in Z$ and $(z_{i-1}, \beta(i), z_i) \in R$ for $i = 1, 2, \ldots$.

A β-computation $z_0, z_1, \ldots$ on $\mathscr{A}$ is called everywhere (e-) successful, almost everywhere (ae-) successful, or infinitely often (io-) successful provided
- $z_i \in Z_{\mathrm{fin}}$ for all $i \in N$,
- $z_i \in Z_{\mathrm{fin}}$ except for a finite number of $i \in N$, and
- $z_i \in Z_{\mathrm{fin}}$ for infinitely many $i \in N$, resp.

A sequence $\beta \in X^\omega$ is called e-accepted (ae-accepted, io-accepted) if there is a e-successful (ae-successful, io-successful) β-computation on $\mathscr{A}$. By $T_e(\mathscr{A})(T_{ae}(\mathscr{A}), T_{io}(\mathscr{A}))$ we denote the ω-language (subset of X^ω)) of all $\beta \in X^\omega$ which are e-accepted (ae-accepted, io-accepted) on $\mathscr{A}$.

(In contrast to [1] we use the terms e-, ae- and io-successfulness instead of 1-, 2- and 3-successfulness because, in our opinion, they are more suggesting and less confusing with Landweber's [6] types of acceptance.)

As a shorthand notation we introduce $\mathscr{D}$, $\mathscr{N}$ and $\mathscr{C}\mathscr{N}$ to denote the classes of deterministic, finitely branching nondeterministic and countably branching nondeterministic automata, respectively. Moreover, for $\alpha \in \{\mathrm{e}, \mathrm{ae}, \mathrm{io}\}$ and $\mathscr{Q} \in \{\mathscr{D}, \mathscr{N}, \mathscr{C}\mathscr{N}\}$ let $T_\alpha(\mathscr{Q}) \overset{\mathrm{df}}{=} \{T_\alpha(\mathscr{A}) : \mathscr{A} \in \mathscr{Q}\}$. Clearly, if $\mathscr{Q} = \mathscr{N}$ or $\mathscr{Q} = \mathscr{C}\mathscr{N}$, then $T_\alpha(\mathscr{Q})$ is closed under projection.

3. Deterministic automata

In this section we briefly derive the relation of the finite and infinite behaviours of deterministic automata. To this end we introduce the following operators mapping languages to ω-languages. According to Elgot [4] we define

$$\lim W \overset{\mathrm{df}}{=} \{\beta : \beta \in X^\omega \text{ and } A(\beta) \subseteq W\}, \tag{2}$$

the limit of the language $W \subseteq X^*$, and according to [2] we define

$$W^\delta \overset{\mathrm{df}}{=} \{\beta : \beta \in X^\omega \text{ and } A(\beta) \cap W \text{ infinite}\}, \tag{3}$$

the δ-limit of the language $W \subseteq X^*$.

(The reader is warned not to confuse this with Eilenberg's [3] notation where $\lim W$ denotes the δ-limit.) We prefer the notation W^δ for the following reason.

Lemma 1 (Davis [2]). *A subset $F \subseteq X^\omega$ is a $\mathscr{G}_\delta$-set if and only if there is a $W \subseteq X^*$ such that*

$$F = W^\delta.$$

Similarly, one has the following lemma.

Lemma 2. *A subset $F \subseteq X^\omega$ is closed iff*

$$F = \lim W \quad \textit{for some } W \subseteq X^*.$$

It is easily verified, that, for a deterministic automaton $\mathscr{A}$,

$$T_\mathrm{e}(\mathscr{A}) = \lim T(\mathscr{A}), \tag{4}$$

$$T_\mathrm{ae}(\mathscr{A}) = X^\omega \setminus (X^* \setminus T(\mathscr{A}))^\delta, \tag{5}$$

$$T_\mathrm{io}(\mathscr{A}) = T(\mathscr{A})^\delta. \tag{6}$$

Hence, one has the following lemma.

Lemma 3

$$T_{\mathrm{e}}(\mathscr{D}) = \mathscr{F}, \tag{7}$$

$$T_{\mathrm{ae}}(\mathscr{D}) = \mathscr{F}_{\delta}, \tag{8}$$

$$T_{\mathrm{io}}(\mathscr{D}) = \mathscr{G}_{\delta}. \tag{9}$$

4. The projection lemmas

Lemma 4. *Let $\mathscr{A}$ be a finitely branching automaton over X. Then there is a deterministic automaton $\mathscr{A}'$ over $X\times\{0,1\}$ such that*

$$T_\alpha(\mathscr{A}) = \mathrm{Pr}\ T_\alpha(\mathscr{A}')$$

for $\alpha \in \{\mathrm{e}, \mathrm{ae}, \mathrm{io}\}$.

Lemma 5. *Let $\mathscr{A}$ be a countably branching automaton over X. Then there is a deterministic automaton $\mathscr{A}'$ over $X\times\{0,1\}$ such that*

$$T_\alpha(\mathscr{A}) = \mathrm{Pr}\,(T_\alpha(\mathscr{A}')\cap E),$$

where $\alpha \in \{\mathrm{e}, \mathrm{ae}, \mathrm{io}\}$ and E is defined by (1).

Proof of Lemmas 4 and 5. For both lemmas we use the same proof which is carried out via the following queue-buffer-construction.

Let $\mathscr{A} = (Z, Z_0, Z_{\mathrm{fin}}, R)$, where Z is w.l.o.g. countable. To every pair (z, x) we associate a fixed string $\mu(z,x)\in Z^*$ (or $Z^*\cup Z^\omega$ when $\mathscr{A}$ is countably branching) consisting of all and only those $z'\in R(z,x)$. Moreover, let $\mu_0\in Z^*$ (or $Z^*\cup Z^\omega$, resp.) be the string consisting of all initial states of $\mathscr{A}$. Now let Z^0 consist of all $\mu(z,x)$ $(z\in Z, x\in X)$ and μ_0, and all of its nonempty tails.

Define $\mathscr{A}' = (Z', s_0, Z'_{\mathrm{fin}}, R')\in\mathscr{D}$ as follows:

$$Z' \overset{\mathrm{df}}{=} Z^0\times(X^*\setminus\{e\})\times\{0,1\}\cup\{s_0, s\},$$

where s_0 and s are a separate initial state and dead sink, resp;

$$Z'_{\mathrm{fin}} \overset{\mathrm{df}}{=} Z^0\times(X^*\setminus\{e\})\times\{1\}\cup\{s_0\},$$

and for $x, x'\in X$, $y, y'\in\{0,1\}$, $w\in Z^0$ and $v\in X^*$,

$$R(s_0, (x, y)) \overset{\mathrm{df}}{=} (\mu_0, x, 1),$$

$$R(s, (x, y)) \overset{\mathrm{df}}{=} s,$$

and

$$R((zw, xv, y), (x', y')) \overset{\text{df}}{=} \begin{cases} s & \text{if } y'=0,\ w=e, \\ (w, vx', y) & \text{if } y'=0,\ w\neq e, \\ (\mu(z,x), vx, 0) & \text{if } y'=1,\ z\notin Z_{\text{fin}}, \\ (\mu(z,x), vx, 1) & \text{if } y'=1,\ z\in Z_{\text{fin}}. \end{cases}$$

This queue-buffer-construction functions in the following manner.

On reading the first letter of the input sequence the automaton $\mathscr{A}'$ buffers the X-label and puts all initial states of $\mathscr{A}$ to the state-queue. As long as the $\{0, 1\}$-label of the following letters is 0, the automaton $\mathscr{A}'$ buffers the X-labels of the corresponding letters and deletes the first state from the state-queue. (If this queue is empty, it switches to the dead sink s.) Whenever the $\{0, 1\}$-label of the input letter is 1, the first state z of the state-queue becomes active and creates with the first letter of the buffered word xv (which, by construction, cannot be the empty word) the new state-queue $\mu(z, x)$, whereas the X-label of the input letter is buffered as in the other case.

The third label of the states in Z' stores the information whether the latest active state was a final one or not. In the passive phase of the work of $\mathscr{A}'$ this information is not changed unless $\mathscr{A}'$ switches to the dead sink s.

It is now evident that an infinite computation on $\mathscr{A}'$ either switches to the dead sink s or otherwise has infinitely many active moves if and only if the input sequence contains infinitely many times a 1 as $\{0, 1\}$-label. If $\mathscr{A}$ is finitely branching, then it is impossible that an active move of $\mathscr{A}'$ is followed by only passive moves unless $\mathscr{A}'$ switches to s.

Thus, for a finitely branching automaton $\mathscr{A}$ an infinite computation on $\mathscr{A}'$ can be (e-, ae- or io-) successful only if the $\{0, 1\}$-labels of the input sequence $\eta \in (X \times \{0, 1\})^{\omega}$ are infinitely often '1'. Contrarily, if $\mathscr{A}$ is countably branching then $\mathscr{A}'$ can have a successful infinite computation whose input sequence η has only a finite number of '1' as $\{0, 1\}$-labels.

On the other hand, from the construction of $\mathscr{A}'$ it is evident, that every β-computation on $\mathscr{A}$ can be simulated in the active moves of a computation on $\mathscr{A}'$ by an appropriate choice of the $\{0, 1\}$-labels of the input sequence η where $\beta = \operatorname{pr} \eta$.

This yields that for $\beta \in X^{\omega}$ there is a (e-, ae- or io-) successful β-computation on $\mathscr{A}$ iff there is an $\eta \in (X \times \{0, 1\})^{\omega}$ with $\operatorname{pr} \eta = \beta$ such that the η-computation is (e-, ae-, or io-) successful on $\mathscr{A}'$ and $\eta \in E$ (which latter condition is implied by successfulness in the case of finitely branching automata). □

As an immediate consequence of Lemmas 4 and 3 we obtain the following.

Corollary 6

$$T_{\text{e}}(\mathscr{N}) = \mathscr{F}, \qquad T_{\text{ae}}(\mathscr{N}) = \mathscr{F}_{\delta}, \qquad T_{\text{io}}(\mathscr{N}) = \mathscr{G}.$$

5. Countably branching automata

For countably branching automata, Lemmas 5 and 3 yield

$$T_e(\mathscr{CN}) \subseteq \{\mathrm{Pr}(F \cap E) : F \in \mathscr{F}\} \subseteq \mathscr{S},$$

$$T_{ae}(\mathscr{CN}) \subseteq \{\mathrm{Pr}(F \cap E) : F \in \mathscr{F}_\delta\} \subseteq \mathscr{S},$$

$$T_{io}(\mathscr{CN}) \subseteq \{\mathrm{Pr}(F) : F \in \mathscr{G}_\delta\} = \mathscr{S}.$$

Since all of the considered classes are closed under projection, and since $\mathscr{F} \subseteq \mathscr{F}_\delta$, it remains to show that $\mathscr{G}_\delta \subseteq T_\alpha(\mathscr{CN})$ for $\alpha \in \{\mathrm{e}, \mathrm{ae}, \mathrm{io}\}$ in order to prove that equality holds everywhere.

To this end we use a guess-and-check-method.

Let $F \subseteq X^\omega$ be a set in $\mathscr{G}_\delta$. According to Lemma 1 there is a $V \subseteq X^*$ such that $V^\delta = F$.

Let, without loss of generality, $e \in V$. Define

$$M_V \overset{\mathrm{df}}{=} \{(v, w) : vw \in V \text{ and } w \neq e\}.$$

We construct $\mathscr{A}_V = (Z, \{z_0\}, Z_{\mathrm{fin}}, R)$ as follows:

$$Z \overset{\mathrm{df}}{=} V \cup M_V \cup \{s\}, \quad \text{where } s \notin V \cup M_V \text{ is a dead sink},$$

$$z_0 \overset{\mathrm{df}}{=} e \in V, \quad Z_{\mathrm{fin}} \overset{\mathrm{df}}{=} Z \setminus \{s\} \quad \text{and} \quad R \subseteq Z \times X \times Z,$$

consisting of all triples of the form $(x \in X)$

(0) $(z, x, s), \quad z \in Z;$

(1) $(v, x, (vx, w)), \quad v \in V, vxw \in V, w \neq e;$

(2) $((v, xw), x, (vx, w)), \quad vxw \in V, w \neq e;$

(3) $((v, x), x, vx)), \quad vx \in V.$

The automaton $\mathscr{A}_V$ successively guesses (by rules of type (1)) and checks (by rules of types (2) and (3)) whether initial parts of the input sequence belong to V. If $\mathscr{A}_V$ fails, it switches to the dead sink s. Clearly $T_e(\mathscr{A}_V) = T_{ae}(\mathscr{A}_V) = T_{io}(\mathscr{A}_V) = V^\delta$.

This finishes the proof of

$$T_e(\mathscr{CN}) = T_{ae}(\mathscr{CN}) = T_{io}(\mathscr{CN}) = \mathscr{S}.$$

6. Conclusion

We have proved projection lemmas for ω-languages in the very general case of arbitrary infinite automata. The problem becomes more complicated when we restrict the class of automata.

It is an open question for which classes of infinite-state devices accepting ω-languages a deterministic device in the class satisfying the projection lemma can be always constructed. Up to now this problem has been solved (positively) for the class of Turing-acceptors (cf. [8]).

Acknowledgment

I am grateful to Prof. Dr. A. Arnold for encouraging me to the present investigation.

References

[1] A. Arnold, Topological characterizations of infinite behaviours of transition systems, Res. Rept. 13, Univ. of Poitiers, 1982.

[2] M. Davis, Infinitary games of perfect information, in: *Advances in Game Theory* (Princeton Univ. Press, Princeton, NJ, 1964) pp. 89–101.

[3] S. Eilenberg, *Automata, Languages and Machines, Vol. A* (Academic Press, New York, 1974).

[4] C.C. Elgot, Decision problems of finite automata design and related arithmetics, *Trans. Amer. Math. Soc.* **98** (196[illegible]) 21–51.

[5] K. Kuratowski, *Topology I* (Academic Press, New York, 1966).

[6] L.H. Landweber, Decision problems for ω-automata, *Math. Systems Theory* **3** (1969) 376–384.

[7] L. Staiger, Zur Topologie der regulären Mengen, Diss. A. Friedrich-Schiller-Univ., Jena, 1976.

[8] L. Staiger and K. Wagner, Rekursive Folgenmengen I, *Zeitschr. Math. Logik Grundl. Math.* **24** (1978) 523–538.

[9] B.A. Trachtenbrot and J.M. Barzdin, *Finite Automata, Behaviour and Synthesis* (North-Holland, Amsterdam, 1973).