PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série, tome 25 (39), 1979, pp. 210—218

ALGORITHMIC CONSTRUCTIONS INSPIRED BY CAPORASO

Kajetan Šeper

(Received October 5, 1978)

Abstract. In algorithm theory, one usually recognizes a *decidable* property by the *type of result* of the process of applying an algorithm to admissible words, the property being true, if the *empty word* appears as the result, and false, otherwise. So does Markov. According to Caporaso, such a property is recognized by the *type of termination* of the process, the property being true, if the process terminates by a *stop command* carried out, and false, otherwise. The paper slightly generalizes and modifies Caporaso's notions (Definitions 3.1, 4.2) and statements (Theorems 1,4), and carries the constructions out directly.[1]

0. Terminological and notational conventions

We use the same terminology and notation as it is used in [3] with a few exceptions. 1) Alphabets are designated by script capitals. Lower case italic letters x and y are used for unspecified letters of a given alphabet. The union of alphabets, say $\mathcal{A}$, $\mathcal{B}$ and $\{\alpha, \beta\}$, is designated by concatenation, $\mathcal{A}\,\mathcal{B}\,\alpha\beta$. Given any alphabet $\mathcal{A}$, the term "$\mathcal{A}$-*word*" is used as an abbreviation of the term "*word in alphabet* $\mathcal{A}$", and the term "$\mathcal{A}$-*property*" as an abbreviation of the term "*property of* $\mathcal{A}$-*words*", the property considered extensionally. 2) "*Normal algorithms*" are called "*Markov algorithms*" instead, abbreviated MA's, and designated by bold face Roman capitals. Accordingly, "*normalization*" is called "*Markovization*" instead. For any MA over alphabet $\mathcal{A}$, the term "$\mathcal{A}$-*total*" is used as an abbreviation of the term "*applicable to all* $\mathcal{A}$-*words*". 3) Given a MA $\mathbf{M}$ and letters α and β, the following stipulations are applied. $\mathbf{M}^{\alpha}$ will mean the MA obtained from $\mathbf{M}$ as a result of replacing in its scheme every command of the form $A\rightarrow\circ B$ by the command $A\rightarrow\alpha B$; similarly, $\mathbf{M}_{\beta}$ will mean the MA obtained from $\mathbf{M}$ by replacing every command of the form $\rightarrow B$ or $\rightarrow\circ B$ by the command $\beta\rightarrow\beta B$ or $\beta\rightarrow\circ\beta B$, respectively. When more indices are involved, the following hierarchy is to be applied: superscripts rank over subscripts, and in each group separately, any index ranks over the next one at the right-hand side.

Key words: Markov (normal) algorithms, decidable properties, graph-scheme composition method

AMS (MOS) subject classification (1970): Primary 02 E 10

1. Markov-and Caporaso-properties

Let $\mathcal{A}$ be an alphabet. We shall define several types of $\mathcal{A}$-properties.

Definition 1. Each $\mathcal{A}$-total MA **C** over $\mathcal{A}$ defines an $\mathcal{A}$-property Π_C of $\mathcal{A}$-words P:

$$\Pi_C(P): \Leftrightarrow \mathbf{C}(P) = \Lambda.$$

The class of all these $\mathcal{A}$-properties, called Π-or synonimously *Markov-properties*, will be designated by Π.

Definition 2.1. Each $\mathcal{A}$-total MA **S** over $\mathcal{A}$ defines an $\mathcal{A}$-property Ω_S:

$$\Omega_S(P): \Leftrightarrow \mathbf{S}: P \models\circ Q,$$

for some word Q in alphabet of **S**.

The class of all these Ω-*properties* will be designated by Ω.

Definition 2.2. In case **S** is an $\mathcal{A}$-*identity algorithm* (i.e. such an algorithm that, for every $\mathcal{A}$-word P, $\mathbf{S}(P) = P$ holds), the property that **S** defines will be designated by Ω^i_S, and the corresponding class of Ω^i-properties by Ω^i.

One easily shows that any Ω-property belongs to Π. Also, that the Π-property Π_0 defined by MA $\{\rightarrow\circ$ does not belong to Ω. Indeed, *exactly the empty word* possesses the property Π_0. Suppose Π_0 belongs to Ω. Let **S** be any MA that defines Π_0. Then **S** should be applicable to the empty word and the process should terminate by a stop command carried out. So, its scheme should contain at least one command with the empty word standing at its left-hand side. Henceforth, *every* $\mathcal{A}$-*word* would possess the property Π_0. This contradicts our supposition.

Thus we have proved

Lemma 1. The class Π properly contains the class Ω.

Obviously, Ω contains Ω^i; *prove* by an example

Lemma 2. The class Ω properly contains the class Ω^i.

Lemmata 1 and 2 suggest the following generalizations.

Definition 3.1. Let **T** be an $\mathcal{A}$-total MA over $\mathcal{A}$. Let **S** be a MA in a certain alphabet applicable to any result $\mathbf{T}(P)$ of the process of applying algorithm **T** to $\mathcal{A}$-words P. Each such ordered pair of MA's **S** and **T** defines an $\mathcal{A}$-property $\Sigma_{S,T}$:

$$\Sigma_{S,T}(P): \Leftrightarrow \mathbf{S}: \mathbf{T}(P) \models\circ Q,$$

for some word Q in alphabet of **S**.

The corresponding class of Σ-or synonimously *Caporaso-properties* will be designated by Σ.

Definition 3.2. In case the *composition* $\mathbf{S}\circ\mathbf{T}$ of $\mathbf{T}$ and $\mathbf{S}$ is an $\mathcal{A}$*-identity algorithm* (i.e. $(\mathbf{S}\circ\mathbf{T})(P)=\mathbf{S}(\mathbf{T}(P))=P$), the property that $\mathbf{S}$ and $\mathbf{T}$ define will be designated by $\Sigma^i_{\mathbf{S},\mathbf{T}}$, and the corresponding class of Σ^i*-properties* by Σ^i.

Remark 1. Just this property has been introduced and handled with in [1].

Theorem 1. *The classes* Π, Σ *and* Σ^i *coincide.*

Proof. To prove the theorem it is sufficient to prove three inclusions indicated by arrows in the scheme

$$\Pi\to\Sigma^i\to\Sigma\to\Pi,$$

where the nontrivial inclusions are the first and the third one, only.

Proof of $\Pi\to\Sigma^i$. Suppose MA $\mathbf{C}$ defines a Π-property. Let $\mathbf{C}$ be in alphabet $\mathcal{B}$. Associate to each letter x of $\mathcal{B}$ a new letter $\bar{x}$, distinct x's having distinct associates $\bar{x}$'s. Denote the alphabet of associates by $\bar{\mathcal{B}}$. Introduce new letters α, β, γ, δ, ε_1 and ε_2, mutually distinct and distinct from the $\bar{x}$'s. Construct MA's $\mathbf{T}$ in $\mathcal{B}\bar{\mathcal{B}}\alpha\beta\gamma\delta\varepsilon_1\varepsilon_2$ and $\mathbf{S}$ in $\mathcal{B}\varepsilon_1$ in the following way:

$$\mathbf{T}\left\{\begin{array}{lll}
(1) & \alpha\to\beta\gamma & \\
(2) & \gamma x\to x\bar{x}\gamma & (\mathrm{I})\\
(3) & \bar{x}y\to y\bar{x} & \\
(4) & \gamma\to & \\
(5) & x\delta\to\delta x & (\mathrm{II.0})\\
(6) & \beta\delta x\to\varepsilon_2 & (\mathrm{II.2})\\
(7) & \beta\delta\to\varepsilon_1 & (\mathrm{II.1})\\
(8) & \varepsilon_2 x\to\varepsilon_2 & \\
(9) & \varepsilon_2\bar{x}\to x\varepsilon_2 & (\mathrm{III.2})\\
(10) & \varepsilon_2\to\circ & \\
(11) & \varepsilon_1\bar{x}\to x\varepsilon_1 & (\mathrm{III.1})\\
(12) & \varepsilon_1\to\circ\varepsilon_1 & \\
(13) & \mathbf{C}_\beta^{\bullet\delta} & \\
(14) & \to\alpha &
\end{array}\right.$$

$$\mathbf{S}\{\varepsilon_1\to\circ$$

The symbols x and y occurring in some commands of the scheme of **T** stand, independently of each command, for unspecified letters of $\mathscr{B}$.
Suppose

$$\mathbf{C}(P)=Q.$$

Then we have

$$\mathbf{T}:P\overset{14}{\vdash}\alpha P\overset{\mathrm{I}}{\models}\beta P\bar{P}\overset{13}{\models}\beta\, Q_1\,\delta\, Q_2\,\bar{P}\overset{\mathrm{II.0}}{\models}\beta\,\delta\, Q\bar{P},$$

followed by

$$\overset{\mathrm{II.1,\,III.1}}{\models}\ \circ P\,\varepsilon_1,\quad \text{if}\quad Q=\Lambda;$$

or

$$\overset{\mathrm{II.2,\,III.2}}{\models}\ \circ P,\quad \text{otherwise i.e. if}\quad Q\neq\Lambda;$$

Eventually, we have

$$\mathbf{S}:P\,\varepsilon_1\models_1\circ P$$

or

$$P\models_0 P\,\urcorner,$$

respectively.

Thus we have proved that **S** and **T** define a Σ^i-property that coincides with the Π-property.

P r o o f o f $\Sigma\rightarrow\Pi$. Suppose MA's **S** and **T** define a Σ-property. Let **S** and **T** be in alphabets $\mathscr{B}_1$ and $\mathscr{B}_2$, respectively. Let α be a new letter, distinct from the letters of $\mathscr{B}_1$. Let c be a fixed letter of $\mathscr{B}_1$. Construct MA **S′** in $\mathscr{B}_1\,\alpha$ thus

$$\mathbf{S}'\begin{cases}\left.\begin{array}{ll}(1) & x\,\alpha\rightarrow\alpha\\ (2) & \alpha\, x\rightarrow\alpha\\ (3) & \alpha\rightarrow\circ\end{array}\right\}(\mathrm{I})\\ \begin{array}{ll}(4) & \qquad\mathbf{S}^{\alpha}\\ (5) & \quad\rightarrow\circ c\end{array}\end{cases}$$

where x stands, independently of each command, for an unspecified letter of $\mathscr{B}_1$. Construct MA **C** as the composition

$$\mathbf{S}'\circ\mathbf{T}.$$

Suppose

$$(\mathbf{S}\circ\mathbf{T})(P)=Q.$$

Then we have

$$\mathbf{S}':\mathbf{T}(P)\overset{4}{\models} Q_1\,\alpha\,Q_2 \overset{I}{\models}\circ\Lambda, \quad \text{if} \quad \mathbf{S}:\mathbf{T}(P)\models\circ Q;$$

or

$$\mathbf{T}(P)\overset{4}{\models} Q \overset{5}{\models}\circ c Q, \quad \text{otherwise i.e. if} \quad \mathbf{S}:\mathbf{T}(P)\models Q\,\urcorner\,.$$

Q. E. D.

Remark 2. In [1] the inclusion $\Pi \to \Sigma^i$ has been proved, **T** consisting of a *composition* of *five* MA's. Obviously, by the composition theorem, this composition can be replaced by *one* MA. Our construction above shows **T** *explicitly* and does not make use of the composition theorem. Moreover, **S** *decides* the property by working *not more than one step*. Precisely, if P possesses the property, the process of applying **S** to $\mathbf{T}(P)$ terminates by the sole stop command carried out; otherwise, the command can not be effected at all.

2. Markov algorithms containing no stop command

It is well-known that MA's the schemes of which contain no stop command realize a rather *small* class of effective processes i.e. word functions realizable by MA's. This is true also if MA's containing stop commands only are considered. However, the special MA's containing no stop command are in a sense equivalent to the general MA's.

Theorem 2. *Given any MA* **C** *in alphabet* $\mathcal{B}$*, another MA* **D** *over* $\mathcal{B}$ *containing no stop command can be constructed such that for every* $\mathcal{B}$*-word* P

$$\mathbf{C}(P)\simeq\mathbf{D}(\alpha P),$$

where α *is a new letter not contained in* $\mathcal{B}$.

Construction. **D** can be obtained from **C** similarly as **T** has been obtained from **C** in sec. **1** by choosing or modifying some commands of **T**:

$$\mathbf{D}\begin{cases}(1) & \\ (2') & \gamma x \to \bar{x}\gamma \\ (4) & \\ (5') & \bar{x}\delta \to \delta\bar{x} \\ (7) & \\ (11) & \\ (12') & \varepsilon_1 \to \\ (13') & \qquad \bar{\mathbf{C}}_\beta^{\circ\delta}\end{cases}$$

3. Ter-Zaharjan's and Caporaso's algorithms

Now we shall introduce two kinds of composition methods for MA's.

D e f i n i t i o n 4.1—4.2. Let GS be a Kalužnin's graph-scheme (cf. [2]), linearly ordered, and with MA's $\mathbf{A}_h$, $1 \leq h \leq n$, associated to its active vertices. Let $\mathcal{A}_h$ be alphabets of $\mathbf{A}_h$, respectively, and $\mathcal{A}$ the union of all $\mathcal{A}_h$. Let the input vertex of GS be omitted and its output vertex designated by $\mathbf{A}_{n+1}$.

Let

Ter-Zaharjan's (cf. [4]) *Caporaso's* (cf. [1])

quasi-normal algorithm,

TQNA, CQNA,

be defined by the following transformation and branching rules that prescribe the process of running of $\mathcal{A}$-words through GS:

1° (*Initiation of the process.*) Put an $\mathcal{A}$-word P under $\mathbf{A}_1$.

2° (*Transition from one to another subprocess.*) If $\mathcal{A}$-word Q stands under $\mathbf{A}_h$, and $\mathbf{A}_h$ is

2.1° a "*transformator*" vertex with the unique arrow leading to $\mathbf{A}_i$, then put $\mathcal{A}$-word $R = \mathbf{A}_h(Q)$ under $\mathbf{A}_i$;

else, if $\mathbf{A}_h$ is

2.2° a "*discernator*" vertex with the "+" and "−" arrow leading to $\mathbf{A}_i$ and $\mathbf{A}_j$, respectively, then put

Q $R = \mathbf{A}_h(Q)$

under $\mathbf{A}_i$ or $\mathbf{A}_j$ in accordance with

$\mathbf{A}_h(Q) = \Lambda$ or $\neq \Lambda$, $\mathbf{A}_h : Q \models\circ R$ or $\models R \urcorner$,

respectively.

It is supposed here that Q is an $\mathcal{A}_h$-word; otherwise, the process is considered to terminate giving no result.

3° (*Termination to the process.*) If $\mathcal{A}$-word Q stands under $\mathbf{A}_{n+1}$, the process terminates and Q is considered to be its result.

R e m a r k 3. The interesting and distinguishing feature of CQNA's, in respect to TQNA's, is the role of its discernator vertices. When some information reaches a discernator vertex, the associated MA tests the corresponding Σ-property, not Π-property, and transforms the information sending the temporary result, i.e. the transformed information, not the information itself, to another vertex accordingly.

T h e o r e m 3. *The classes of TQNA's and of CQNA's are $\mathcal{A}$-equivalent, i.e. both classes of algorithms realize the same $\mathcal{A}$-word functions.*

Construction. All we need do to construct, for each

TQNA,	CQNA,

an equivalent

CQNA,	TQNA,

is to replace in GS every discernator vertex $\mathbf{A}_h$ by

the segment $\overset{\circ \rightarrow \circ}{\mathbf{T}\ \mathbf{S}}$,	another discernator vertex $\mathbf{A}_h'$,

where

T and **S** are	$\mathbf{A}_h'$ is

constructed from $\mathbf{A}_h$ exactly in the same way as

they were	**S**′ was

constructed from

C	**S**

in sec. **1.** (The arrows that get in $\mathbf{A}_h$, get in **T**; those that come from $\mathbf{A}_h$, come from **S**.)

4. Markovization of Caporaso's algorithms

Theorem 4. *For any given CQNA* **K** *an equivalent MA* **M** *can be constructed.*

Construction. We suppose that all MA's $\mathbf{A}_h$ are in alphabet $\mathcal{A}$. (The general case can easily be reduced to this one merely by using formal extensions of MA's under consideration to the union of their alphabets.) Introduce new letters α, β, ε and η, mutually distinct and distinct from the letters of $\mathcal{A}$. Given a letter and any integer, the exponential notation is used for the word consisting of the indicated number of successive occurrences of that letter. For every MA $\mathbf{A}_h$, $1 \leq h \leq n$, add in its scheme at the bottom, as the m_hth command, the stop command $\rightarrow \circ$. Define $s(h) := \sum_{k=1}^{h-1} m_k$, $1 \leq h \leq n+1$, where $s(1) := 0$ is stipulated. Then replace every command F_{hk_h} of $\mathbf{A}_h$, $1 \leq h \leq n$, $1 \leq k_h \leq m_h$, of the form

$$A_{hk_h} \rightarrow (\circ)\, B_{hk_h}$$

by the command F'_{hk_h}

$$\varepsilon \alpha^{s(h)+k_h} \eta\, A_{hk_h} \rightarrow (\circ)\, \beta^{t(h,\, k_h)}\, B_{hk_h},$$

where $t(h, k_h)$ is defined by cases:

$t(h, k_h) = h$, if 1) F_{hk} is a nonstop command;

$= i$, if 2) F_{hk} is a stop command, and: $\mathbf{A}_h$ is a transformator vertex with the unique arrow getting in $\mathbf{A}_i$, or, $\mathbf{A}_h$ is a discernator vertex with the "+" arrow getting in $\mathbf{A}_i$ and $k_h < m_h$;

$= j$, if 3) F_{hk_h} is a stop command, $\mathbf{A}_h$ is a discernator vertex with the "−" arrow getting in $\mathbf{A}_j$, and $k_h = m_h$;

it is supposed in cases 2) and 3) that i and $j \leq n$;

$= 0$, if else 4) i or $j = n+1$;

and F'_{hk_h} is a stop command exactly in case 4). Now, the required MA **M** can be determined by the following scheme:

$$\mathbf{M}\begin{cases} \varepsilon\alpha^{s(h)+k_h}\eta A_{hk_h} \to (\circ)\, \beta^{t(h,k_h)} B_{hk_h} & \text{(I)} \\ \varepsilon\alpha^p\eta x \to x\varepsilon\alpha^p\eta & \text{(II.1)} \\ \left.\begin{array}{l} \varepsilon \to \alpha \\ x\alpha \to \alpha x \\ x\eta \to \eta x \\ \alpha \to \varepsilon\alpha \end{array}\right\} & \text{(II.2)} \\ x\beta \to \beta x & \text{(III.1)} \\ \beta^q \to \varepsilon\alpha^{s(q)+1}\eta & \text{(III.2)} \\ \to \varepsilon\alpha\eta & \end{cases}$$

where

$h = 1, 2, \ldots, n$; $k_h = 1, 2, \ldots, m_h$,

p stands for any number of

$$\{1, 2, \ldots, s(n+1)\} - \{s(2), s(3), \ldots, s(n+1)\};$$

$q = n, n-1, \ldots, 1$; N. B. Take notice of the *descending* order, the order being essential in the construction.

x stands, independently of each command, for any letter of $\mathcal{A}$.

Remark 4. The idea of Markovization of CQNA proposed in [1] is very interesting and attractive. However, we could not reduce the mistakes in the construction (and the proof) to misprints, so we have given another one, making use of that idea.[2]

Obviously, the construction can be carried out making no use of η. In the following construction no use is made of auxiliary letters others than ε.

Modified construction. Let c be a fixed letter of $\mathcal{A}$. Define $m := \sum_{k=1}^{n} m_h = s(n+1)$. Define $\tilde{t}(h, k_h) := t(h, k_h) + 2m$. In case 4), $\varepsilon c^{2m}\varepsilon$ has to be omitted, and the command considered as stop command. Then define **M** in $\mathcal{A}\varepsilon$, i.e. in the extension of $\mathcal{A}$ with ε as the auxiliary letter only, by the following scheme:

$$
\mathbf{M}\begin{cases}
\varepsilon c^{s(h)+kh}\varepsilon A_{hk_h}\rightarrow(\circ)\,\varepsilon c^{\tilde{t}(h,\,kh)}\varepsilon B_{hk_h} & \text{(I)}\\
\varepsilon c^{p}\varepsilon x\rightarrow x\varepsilon c^{p}\varepsilon & \text{(II.1)}\\
\varepsilon c^{p}\varepsilon\rightarrow\varepsilon c^{p+m}\varepsilon & \text{(II.2.1)}\\
x\varepsilon c^{p+m}\varepsilon\rightarrow\varepsilon c^{p+m}\varepsilon x & \text{(II.2.2)}\\
\varepsilon c^{p+m}\varepsilon\rightarrow\varepsilon c^{p+1}\varepsilon & \text{(II.2.3)}\\
x\varepsilon c^{h+2m}\varepsilon\rightarrow\varepsilon c^{h+2m}\varepsilon x & \text{(III.1)}\\
\varepsilon c^{h+2m}\varepsilon\rightarrow\varepsilon c^{s(h)+1}\varepsilon & \text{(III.2)}\\
\rightarrow\varepsilon c\varepsilon &
\end{cases}
$$

Acknowledgment. I would like to express my gratitude to Mr. Andrej Ščedrov for his 1977 review of the paper for my personal usage.

NOTES

1 I wish to give here some comments. The first version of this paper, entitled *A correction and addenda to Caporaso's "A composition method for Markov normal algorithms"* was written ten years ago. When it was completed for publication, it disappeared under very strange circumstances. After some time it was given back to me. Recently I accidently met the dislaid paper together with the related drafts. I found it hard readable, so I rewrote it and made some corrections and improvements.

2 Among the drafts I found one additional sheet that indicates how Caporaso's construction can be simply corrected: the command $\eta\beta\rightarrow\beta\eta$ is to be replaced by $\eta\beta\rightarrow\alpha\eta$ and shifted to stand ahead of the preceding one in the scheme.

REFERENCES

[1] S. Caporaso, *A Composition Method for Normal Markov Algorithms*, **ICC Bul. 2** (1963), 195—204,

[2] L. A. Kalužnin, *Über die Algorithmierung mathematischer Aufgaben*, **Probleme der Kybernetik 2** (1963), 54—74, Akademie-Verlag, Berlin,

[3] A. A. Markov, *Theory of Algorithms*, The Israel Program for Scientific Translations, Jerusalem, 1961,

[4] N. P. Ter-Zaharjan, *O normalizuemosti graf-shemnyh algorifmov*, **Matematičeskie voprosy kibernetiki i vyčislitel'noĭ tehniki 1** (1963), 30—39.

Institute of Mechanical Constructions,
Faculty of Mechanical Engeenering and Naval Architecture,
Zagreb University