A NOTE ON SE-SYSTEMS AND REGULAR CANONICAL SYSTEMS

F.L. ŢIPLEA AND ERKKI MÄKINEN

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF TAMPERE

REPORT A-2000-14

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER AND
INFORMATION SCIENCES
SERIES OF PUBLICATIONS A
A-2000-14, OCTOBER 2000

A NOTE ON SE-SYSTEMS AND REGULAR CANONICAL SYSTEMS

F.L. ŢIPLEA AND ERKKI MÄKINEN

University of Tampere
Department of Computer and Information Sciences
P.O.Box 607
FIN-33014 University of Tampere, Finland

ISBN 951-44-4985-1
ISSN 1457-2060

A Note on SE-Systems and Regular Canonical Systems

Ferucio Laurenţiu Ţiplea [1]

Faculty of Computer Science, "Al.I.Cuza" University of Iasi, 6600 Iasi, Romania

Erkki Mäkinen [2]

Department of Computer and Information Sciences, P.O. Box 607, FIN-33014 University of Tampere, Finland

Abstract

A *synchronized extension system* is a 4-tuple $G = (V, L_1, L_2, S)$, where V is an alphabet and L_1, L_2 and S are languages over V. Such systems generate languages extending L_1 by L_2 to the left or to the right, and synchronizing on words in S.

In this note we consider the relationship between synchronized extension systems and regular canonical systems. We are able to give a simplified and generalized proof for the classical result concerning the regularity of the languages defined by regular canonical systems.

Keywords: formal languages, synchronized extension systems, regular canonical systems.

1 Introduction and Preliminaries

Synchronized extension systems (SE-systems, for short) have been introduced in [2] as 4-tuples $G = (V, L_1, L_2, S)$, where V is an alphabet and L_1, L_2 and S are languages over V. L_1 is called the *initial language*, L_2 the *extending language*, and S the *synchronization set* of G. For an SE-system G, define the binary relations $\Rightarrow_{G,r}$, $\Rightarrow_{G,r^-}$, $\Rightarrow_{G,l}$ and $\Rightarrow_{G,l^-}$ over V^* as follows:

(i) $u \Rightarrow_{G,r} v$ iff $(\exists w \in L_2)(\exists s \in S)(\exists x, y \in V^*)(u = xs \wedge w = sy \wedge v = xsy)$;
(ii) $u \Rightarrow_{G,r^-} v$ iff $(\exists w \in L_2)(\exists s \in S)(\exists x, y \in V^*)(u = xs \wedge w = sy \wedge v = xy)$;
(iii) $u \Rightarrow_{G,l} v$ iff $(\exists w \in L_2)(\exists s \in S)(\exists x, y \in V^*)(u = sx \wedge w = ys \wedge v = ysx)$;
(iv) $u \Rightarrow_{G,l^-} v$ iff $(\exists w \in L_2)(\exists s \in S)(\exists x, y \in V^*)(u = sx \wedge w = ys \wedge v = yx)$.

[1] E-mail: `fltiplea@infoiasi.ro`

[2] E-mail: `em@cs.uta.fi`. Work supported by the Academy of Finland (Project 35025).

In an SE-system $G = (V, L_1, L_2, S)$, the words in S act as synchronization words. They can be kept or neglected in the final result, and r, r^-, l, and l^- are called the *modes of synchronizations*. In what follows, we restrict ourselves to the mode l^-.

We say that an SE-system $G = (V, L_1, L_2, S)$ is *of type* (p_1, p_2, p_3) if the L_1, L_2, and S are languages having the properties p_1, p_2, and p_3, respectively. We use the abbreviations f and reg for the properties of finiteness and regularity, respectively.

A derivation $u \stackrel{*}{\Rightarrow}_{l^-} v$ is called an *l^--derivation*. The *language of type l^-* generated by an *SE-system* $G = (V, L_1,\ L_2, S)$ is defined as

$$L^{l^-}(G) = \{v \in V^* | \exists u \in L_1 :\ u \stackrel{*}{\Rightarrow}_{G,l^-} v\}.$$

(Naturally, the other modes of synchronization as well define their own classes of languages, but we do not need them here.)

The following result is essential for this note.

Theorem 1 ([2]) *For any SE-system G of type (reg, reg, f), the language $L^{l^-}(G)$ is regular.*

Next we recall from [1] some concepts related to regular canonical systems.

A *regular canonical system* is a 3-tuple $C = (V, V_T, P)$, where V is a (finite) alphabet, $V_T \subseteq V$ and $P \subseteq V^* \times V^*$ is a finite set of *productions*.

A regular canonical system C induces a binary relation $\Rightarrow_C \subseteq V^* \times V^*$ defined by

$$u \Rightarrow_C v \quad \Leftrightarrow \quad u = \alpha u',\ v = \beta u',\ (\alpha, \beta) \in P,$$

for all $u, v \in V^*$.

For a regular canonical system and two finite languages L_1 and L_2 over V, we define

$$L_g(C, L_1, L_2) = \{w \in V_T^* | (\exists u \in L_1)(\exists v \in L_2)(u \stackrel{*}{\Rightarrow}_C vw)\}$$

and

$$L_a(C, L_1, L_2) = \{w \in V_T^* | (\exists u \in L_1)(\exists v \in L_2)(uw \stackrel{*}{\Rightarrow}_C v)\},$$

and call them the *language generated* and, respectively, *accepted* by (C, L_1, L_2).

We use the notation $\partial^l_K(L)$ for the left quotient of L by K. If a is a symbol and L is a language, aL stands for the language $\{aw | w \in L\}$.

2 Regular Canonical Systems and SE-systems

The languages $L_g(C, L_1, L_2)$ and $L_a(C, L_1, L_2)$ defined by a regular canonical system C are known to be regular (see e.g. Chapter IV.10 of [1]). Using the results in [2] we can prove the regularity of these languages in a very simple way.

Theorem 2 *Let $C = (V, V_T, P)$ be a regular canonical system and let L_1 and L_2 be finite languages over V. Then, $L_g(C, L_1, L_2)$ and $L_a(C, L_1, L_2)$ are regular languages.*

Proof. Let $G_1 = (V, L'_1, L'_2, S)$ be the SE-system of type (f, f, f) defined by

- $L'_1 = \#L_1 \cup L_1$;
- $L'_2 = \{\#\beta\#\alpha | (\alpha, \beta) \in P\} \cup \{\beta\#\alpha | (\alpha, \beta) \in P\}$;
- $S = \{\#\alpha | \exists\beta : \ (\alpha, \beta) \in P\}$.

It is easy to see that $L_g(C, L_1, L_2) = \partial^l_{L_2}(L^{l^-}(G_1)) \cap V_T^*$, which shows that $L_g(C, L_1, L_2)$ is regular, because $L^{l^-}(G_1)$ is regular by Theorem 1.

Consider now $G_2 = (V, L''_1, L''_2, S')$ of type (f, f, f) defined by

- $L''_1 = \#L_2 \cup L_2$;
- $L'_2 = \{\#\alpha\#\beta | (\alpha, \beta) \in P\} \cup \{\alpha\#\beta | (\alpha, \beta) \in P\}$;
- $S' = \{\#\beta | \exists\alpha : \ (\alpha, \beta) \in P\}$.

As above, it is easy to see that $L_a(C, L_1, L_2) = \partial^l_{L_1}(L^{l^-}(G_2)) \cap V_T^*$, implying the regularity of $L_a(C, L_1, L_2)$.

The proof of Theorem 2 shows that regular canonical systems are not more powerful than SE-systems of type (f, f, f) (in fact, we can easily see that each computation step in C starting from a word in L_1 or L_2 is "simulated" by a derivation step in G_1 or G_2, respectively).

If we change the definition of C by removing the set V_T and, correspondingly, change the definition of $L_g(C, L_1, L_2)$ by replacing V_T with V, we obtain the concept of *pure regular canonical systems.* Theorem 10.6 of [1] can now be written as follows:

- For every regular language L, there is a pure regular canonical system $C = (V, P)$, a subset $V_T \subseteq V$, and a finite language L_1 over V such that

$$L = L_g(C, L_1, \{\lambda\}) \cap V_T^*.$$

Now, it is interesting to compare this result with that from Example 2.1.2 in [2] which says that each regular language L can be written in the form $L = L^{l^-}(G) \cap V_T^*$, where G is an SE-system of type (f, f, f) and V_T is a subset of the alphabet of G.

Therefore, Theorem 10.7 in [1] can be reformulated in terms of SE-systems as follows.

Theorem 3 *A language L is regular iff it is of the form $L = L^{l^-}(G) \cap V_T^*$, for some SE-system G of type (f, f, f) and subset V_T of the alphabet of G.*

Finally, we want to point out again that our SE-systems are more general than regular canonical systems, and the results we got (for example, Theorem 1) are more general than the corresponding results with regular canonical systems.

References

[1] A. Salomaa. *Theory of Automata*, Pergamon Press, 1969.

[2] F.L. Ţiplea, E. Mäkinen, C. Apachiţe. *Synchronized Extension Systems*, to appear in *Acta Inform.*