On the Supremal L_m-closed and the Supremal L_m-closed and L-controllable Sublanguages of a Given Language

Roberto M. Ziller and José E. R. Cury
LCMI -- EEL - UFSC - Cx. postal 476 -- 88040-900 - Florianópolis - SC - Brazil
e-mail: ziller@lcmi.ufsc.br, cury@lcmi.ufsc.br

1 Introduction

In the last ten years, the Ramadge-Wonham framework for Discrete Event Systems modeling, analysis and controller (supervisor) synthesis has advanced to an important position among the various models suggested in this research field.

In the basic RW-model, both the system to be controlled and the desired behavior are specified through the use of languages. Solving a synthesis problem amounts to find a controller - called *supervisor* - that restricts the physically possible behavior of the system to be controlled - called *plant* - to the desired one. Solutions are stated in terms of the *supremal L_m-closed and L-controllable sublanguage* of the language representing the target behavior. While several authors have contributed to the computation of the supremal L-controllable sublanguage (see especially [WR87] and also [Rudi88], [LVW88], [BGK$^+$90] and [KGM91]), no algorithms or formulas for the supremal L_m-closed and the supremal L_m-closed and L-controllable sublanguages of a given language could be found in the surveyed literature.

In this paper we present formulas for these sublanguages. Section 2 recalls some results from RW-theory, section 3 presents the formulas; computational complexity is analyzed in section 4, and an example is given in section 5.

2 Preliminaries

A detailed description of the Ramadge-Wonham framework is given in [RW87] and related articles, the reader being referred to these sources for background knowledge. We only recall some facts needed to present our contribution.

System behavior is represented by a 5-tuple $G = \langle \Sigma, Q, \delta, q_0, Q_m \rangle$ called *generator*, where Σ is a set of event labels, also called the *event alphabet*, Q is a set of states, and $\delta: \Sigma \times Q \to Q$ is a (generally partial) transition function defined at each $q \in Q$ for a subset of the $\sigma \in \Sigma$ so that $q' = \delta(\sigma, q)$ represents the state transition $q \xrightarrow{\sigma} q'$, meaning that the occurrence of event σ takes the system from state q to state q'. $q_0 \in Q$ is the *initial state* and $Q_m \subseteq Q$ is a set of *marker states*. These are used to mark the termination of certain event sequences, representing the completion of a task by the system.

Each generator G has two associated *languages:* $L(G)$, the language *generated* by G, and $L_m(G)$, the language *marked* by G. These are sets of *words* formed with symbols of Σ. $L(G)$ represents the physically possible behavior of the system, while $L_m(G)$ stands for the tasks it is able to complete.

The alphabet Σ is partitioned into *controllable* and *uncontrollable* events according to $\Sigma = \Sigma_c \bigcup \Sigma_u$ and $\Sigma_c \bigcap \Sigma_u = \varnothing$. Control action is performed by an external agent called *supervisor*, which observes the events generated by the plant and applies a *control input* $\gamma \subseteq \Sigma$ to the system in response to them. The events in γ are those specified to be enabled by the supervisor.

This control action restricts the system generated and marked languages. The languages representing the physically possible behavior and the tasks the system may complete under supervision are denoted by $L(S/G)$ and $L_c(S/G)$, respectively.

The prefix-closure $\overline{K}$ of a language K is the set of all prefixes (initial segments) of strings in K. K is said to be *prefix-closed* iff $K = \overline{K}$. Given two arbitrary languages $K, L \subseteq \Sigma^*$ and a partition $\Sigma = \Sigma_c \bigcup \Sigma_u$, K is said to be *L-closed* iff $K = \overline{K} \bigcap L$; K said to be *L-controllable* iff $\overline{K}\Sigma_u \bigcap L \subseteq \overline{K}$.

The main synthesis problem in the RW-model can be stated as follows:

Supervisory Control Problem (SCP): Given a plant *G*, a target language $E \subseteq L_m(G)$ and a minimal acceptable language $A \subseteq E$, construct a proper supervisor *S* for *G* such that

$$A \subseteq L_c(S/G) \subseteq E.$$

The language *E* is interpreted as the desired behavior under supervision, while *A* stands for the minimal closed-loop behavior that is still acceptable.

It is shown in [RW87] that the class $C(E)$ of all $L(G)$-controllable sublanguages of *E* and the class $F(E)$ of all $L_m(G)$-closed sublanguages of *E* are both non empty and closed under arbitrary unions. Consequently, these classes contain the supremal elements $\sup C(E)$ and $\sup F(E)$, called *supremal* $L(G)$*-controllable sublanguage of E* and *supremal* $L_m(G)$*-closed sublanguage of E*, respectively. The same applies to the class $CF(E) = C(E) \cap F(E)$, so there also exists $\sup CF(E)$, the *supremal* $L(G)$*-controllable and* $L_m(G)$*-closed sublanguage of E*.

The solution to SCP is then given by the following theorem ([RW87], theorem 7.1, part (ii)):

Theorem 1: SCP is solvable iff $\sup CF(E) \supseteq A$.

3 Computing Solutions

In this section we present formulas for both the supremal $L_m(G)$-closed sublanguage and the supremal $L_m(G)$-closed and $L(G)$-controllable sublanguage of a given language. From this point on, we abbreviate $L(G)$ by *L* and $L_m(G)$ by L_m whenever no confusion is possible. We need the following lemmas:

Lemma 1: Given a language $K \subseteq L$, if $K = \overline{K}$, then $K \cap L_m$ is L_m-closed.

Proof: ($\subseteq$): $K \cap L_m \subseteq \overline{K \cap L_m}$ and $K \cap L_m \subseteq L_m$, so $K \cap L_m \subseteq \overline{K \cap L_m} \cap L_m$.

($\supseteq$): $K = \overline{K} \Rightarrow K \supseteq \overline{K \cap L_m} \Rightarrow K \cap L_m \supseteq \overline{K \cap L_m} \cap L_m$. ♦

Lemma 2: Given a language $E \subseteq L_m$, if *E* is L_m-closed, so is $\sup C(E)$.

Proof: Let *K* be an arbitrary controllable (non necessarily L_m-closed) sublanguage of *E*, so $K \in C(E)$. Then $K \subseteq E \Rightarrow \overline{K} \subseteq \overline{E} \Rightarrow \overline{K} \cap L_m \subseteq \overline{E} \cap L_m = E$. Since $K \subseteq \overline{K}$ and $K \subseteq E \subseteq L_m$, it is also true that $\overline{K} \cap L_m \supseteq K$. By lemma 1, $\overline{K} \cap L_m$ is L_m-closed. As shown below, this language is also controllable:

$$\left[\overline{\overline{K} \cap L_m}\right]\Sigma_u \cap L(G) \subseteq \left[\overline{K} \cap \overline{L_m}\right]\Sigma_u \cap L(G) \subseteq \overline{K}\Sigma_u \cap L(G) \subseteq \overline{K} \subseteq \overline{\overline{K} \cap L_m}.$$

This means that, for every language $K \in C(E)$ that is not L_m-closed, there is an L_m-closed and controllable language $\overline{K} \cap L_m \supseteq K$ that is also contained in $C(E)$, and so $\sup C(E)$ is L_m-closed. ♦

Lemma 3: Given a language $E \subseteq L_m$, $\sup CF(E) = \sup C\left[\sup F(E)\right]$.

Proof: By the definitions of $\sup F(E)$ and $\sup CF(E)$ it follows that $\sup CF(E) \subseteq \sup F(E)$, so $\sup CF(E) = \sup CF\left[\sup F(E)\right]$. By lemma 2, $\sup C\left[\sup F(E)\right] = \sup CF\left[\sup F(E)\right]$, and the result is immediate. ♦

We emphasize that this result guarantees that computing the supremal *L*-controllable sublanguage of the target language is sufficient to solve SCP when the target language is known to be L_m-closed.

In the following development let $\sup P(K)$ denote the *supremal prefix-closed sublanguage of K*, defined as $\sup P(K) = \{s : s \in K \wedge \overline{s} \subseteq K\}$, where $\overline{s}$ stands for $\overline{\{s\}}$, the set of all prefixes of the word *s*. The class of all prefix-closed sublanguages of *K* is clearly non empty (since the empty language $\varnothing$ is prefix-closed) and closed under arbitrary unions, so the supremal element defined above is guaranteed to exist. For the arbitrary language *K* let $\overline{K} - K$ denote the set-theoretic difference $\overline{K} \cap K^c$, where K^c is the complement of *K* with respect to Σ^*.

The following proposition presents our main result:

Proposition 1: Given a language $E \subseteq L_m$, let $M = \sup P\left[\overline{E} - \left(\overline{E} - E\right) \cap L_m\right]$. Then $\sup F(E) = M \cap L_m$ and $\sup CF(E) = \sup C(M \cap L_m)$.

Proof: First part: $\sup F(E) = M \cap L_m$.

(i) $\sup F(E) \supseteq M \bigcap L_m$: demonstrating this relation amounts to show that $M \bigcap L_m$ is L_m-closed and that $M \bigcap L_m \subseteq E$. $M \bigcap L_m$ is L-closed by lemma 1 and by the fact that $M = \overline{M}$; $M \bigcap L_m \subseteq E$ because

$$s \in M \bigcap L_m \Rightarrow s \in \sup P\left[\overline{E} - \left(\overline{E} - E\right) \bigcap L_m\right] \bigcap L_m \Rightarrow s \in \left[\overline{E} - \left(\overline{E} - E\right) \bigcap L_m\right] \bigcap L_m \Rightarrow$$

$$\Rightarrow s \in \overline{E} \bigcap L_m - \left(\overline{E} - E\right) \bigcap L_m \Rightarrow s \in E \bigcap L_m \Rightarrow s \in E.$$

(ii) $\sup F(E) \subseteq M \bigcap L_m$: demonstrating this relation amounts to show that

$\forall s: s \in E \wedge \bar{s} \bigcap L_m = \bar{s} \bigcap E \Rightarrow s \in M \bigcap L_m$

It is immediate that $s \in E \Rightarrow s \in L_m$, since $E \subseteq L_m$. To show $s \in M$ we start from the hypothesis $s \in E \wedge \bar{s} \bigcap L_m = \bar{s} \bigcap E$. Then

$$w \in \bar{s} \bigcap L_m \Rightarrow w \in \bar{s} \bigcap E \Rightarrow w \notin \overline{E} - \bar{s} \bigcap E \Rightarrow (\bar{s} \bigcap L_m) \bigcap \left(\overline{E} - \bar{s} \bigcap E\right) = \varnothing \Rightarrow$$

$$\Rightarrow \bar{s} \bigcap \left(\overline{E} - \bar{s} \bigcap E\right) \bigcap L_m = \varnothing.$$

Now $s \in E \Rightarrow \bar{s} \subseteq \overline{E}$, so $\bar{s} \subseteq \overline{E} - \left(\overline{E} - \bar{s} \bigcap E\right) \bigcap L_m \subseteq \overline{E} - \left(\overline{E} - E\right) \bigcap L_m$ and $s \in M$,

and thus $\sup F(E) = M \bigcap L_m$. The second part of the proposition, namely $\sup CF(E) = \sup C(M \bigcap L_m)$, is immediate by lemma 3, so the proof is complete. ♦

4 Computational Complexity

This section presents an analysis of the computational complexity of the algorithms needed to determine $\sup F(E)$ and $\sup CF(E)$. It is based on the following results from [Rudi88]:

- given two generators G_1 and G_2 of common alphabet Σ, let $|\Sigma| = s$, $|\Sigma_u| = s_u$, let n_i and e_i $(i = 1,2)$ denote the number of states and transitions of generator G_i, respectively. Constructing a generator G such that $L_m(G)$ is the supremal $L(G_1)$-controllable sublanguage of $L_m(G_2)$ is a task of complexity $O(sn_1n_2 + s_ue_1e_2)$;
- constructing a generator G such that $L_m(G) = L_m(G_1) \bigcap L_m(G_2)$ is a task of complexity $O(n_1n_2 + e_1e_2)$;
- given G with n states, m marker states, e transitions and event set of cardinality s, constructing a generator for $[L_m(G)]^c$ is a task of complexity $O[s(n+e) + nm]$.

The above expressions can be simplified considering (i) that for any deterministic finite automaton, $e \leq sn$ and (ii) that s can be viewed as a parameter, rather than a variable which necessarily increases as does the size of the state set. Also, $m = \alpha n$ for some constant α. This gives the expressions $O(n_1n_2)$, $O(n_1n_2)$ and $O(n^2)$ for the complexities of the three algorithms above, respectively.

It is easy to see that, given a generator $D = \langle \Sigma, R, \rho, p_0, R_m \rangle$ such that $L_m(D) = N$, a generator for $\sup P(N)$ is

$D' = \mathrm{Ac}\left(\langle \Sigma, R_m, \rho | \Sigma \times R_m, r_0, R_m \rangle\right)$ if $r_0 \in R_m$

and $\Sigma_\varnothing = \langle \Sigma, \varnothing, \varnothing, \varnothing, \varnothing \rangle$ otherwise,

where Ac(.) denotes the accessible component of the operand and $L(\Sigma_\varnothing) = \varnothing$.

An algorithm that produces a generator for $\sup C(K)$ is given in [WR87].

It is now easy to verify that the highest degree of complexity arising in the computation of $\sup F(E)$ and $\sup CF(E)$ is $O[\max(n_H^2, n_G n_H)]$, where n_G and n_H are the cardinalities of the state sets of the generators G and H representing the plant and the target behavior, respectively. The computations are hence quadratic in time.

5 Example

Figure 1 shows an imaginary generator G of alphabet $\{\alpha, \beta, \lambda, \mu\}$, where we assume that $\Sigma_c = \{\alpha, \lambda\}$ and $\Sigma_u = \{\beta, \mu\}$. The target language corresponding to the desired closed-loop behavior is the language E marked by generator H shown in figure 2.

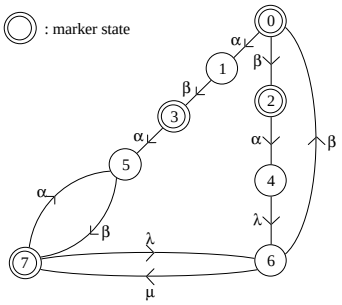

Fig. 1 - Generator *G:* the plant

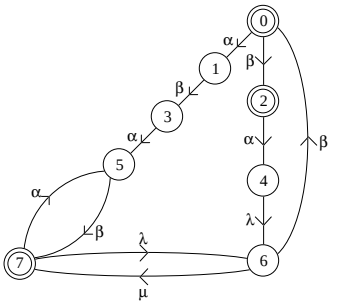

Fig. 2 - Generator *H:* desired behavior

A generator for $\sup F(E)$ is shown in figure 3, where the "bad state" (state 3) that caused E not to be $L_m(G)$-closed has been eliminated. The solution to SCP is given by the supremal $L_m(G)$-closed and $L(G)$-controllable sublanguage of E. A generator for this language is shown in figure 4.

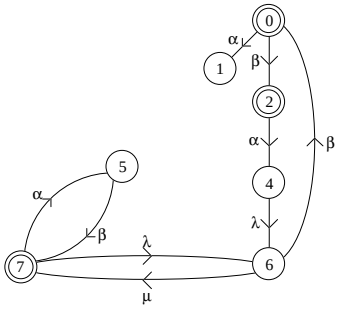

Fig. 3 - Generator for sup*F(E)*

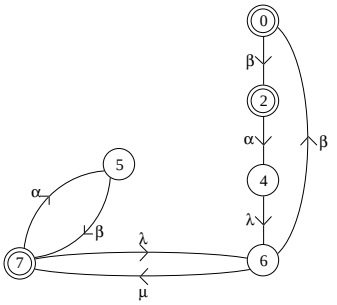

Fig. 4 - Generator for sup*CF(E)*

6 Conclusion

Computing a solution for the *supervisory control problem* (SCP) requires an algorithm for the supremal $L_m(G)$-closed and $L(G)$-controllable sublanguage of the target language E. Only when E is already $L_m(G)$-closed the solution reduces to the computation of $\sup C(E)$. The proposed formula for $\sup CF(E)$ (based on the computation of $\sup F(E)$) allows solving SCP in the general case.

References

[BGK+90] Brandt, R. D.; Garg, V.; Kumar, R.; Lin, F.; Marcus, S. I.; Wonham, W. M.: "Formulas for Calculating Supremal Controllable and Normal Sublanguages". Systems & Control Letters, 15(2):111-117 (1990)

[KGM91] Kumar, R.; Garg, V.; Marcus, S. I.: "On Controllability and Normality of Discrete Event Dynamical Systems". *Systems & Control Letters*, 17:157-168 (1991)

[LVW88] Lin, F.; Vaz, A. F.; Wonham, W. M.: "Supervisor Specification and Synthesis for Discrete Event Systems". *Int. J. Control*, 48(1):321-332 (1988)

[Rudi88] Rudie, Karen G.: "Software for the Control of Discrete Event Systems: A Complexity Study". *M. A. Sc. Thesis*, Department of Electrical Engineering, University of Toronto, Toronto, Canada (1988)

[RW87] Ramadge, P. J.; Wonham, W. M.: "Supervisory Control of a Class of Discrete Event Processes". *SIAM J. Control and Optimization*, 25(1):206-230 (1987)

[WR87] Wonham, W. M.; Ramadge, P. J.: "On the Supremal Controllable Sublanguage of a Given Language". *SIAM J. Control and Optimization*, 25(3):637-659 (1987)

P.S.: This paper was presented at the 11th International Conference on Analysis and Optimization of Systems - Discrete Event Systems (Sophia-Antipolis, June 15-17, 1994) and published in *Lecture Notes in Control and Information Sciences 199*, Springer-Verlag, London, UK, pp. 80-85 (1994).