Modal Specifications for the Control Theory of Discrete Event Systems

Guillaume Feuillade and Sophie Pinchinat
IRISA, Campus de Beaulieu, Rennes, France

Contents

1 Introduction

The present paper advocates the use of logical specifications for the control theory of reactive systems, modeled by discrete event systems.

The terminology *reactive systems* is due to David Harel and Amir Pnueli [HP84] to qualify the class of systems, or programs, where the emphasis is put on their interaction with the environment[1] rather than on their ultimate computation, if relevant. Plethora of examples exists, such as operational systems, information systems, embedded systems, and systems for e-commerce.

While formal methods for the validation and the verification of reactive systems are now well understood and their use well established, synthesis procedures, that is automated methods to construct programs, still need investigations to render systems development safe and cost-efficient. Such methods strongly rely on a rigorous formalism for the specification of the synthesis objectives, such as formal languages, and logic-based formalisms. Although in general *control theory* designates the study of methods to regulate e.g. physical devices, manufacturing, or chemical plants, the whole expression *control theory of discrete-event systems* refines the field by explicitly make use of fairly simple mathematical objects to model the systems to be controlled. Typically, one considers models like finite state automata, Kripke structures, labeled transition systems, or linear systems. As programs can possibly be abstracted and modeled akin, control theory and synthesis issues are tightly coupled: indeed the control device is first expected to be a pro-

[1] in this regard, they should rather be called *interactive*

gram, and second it is intended to be automatically generated regarding the large scaled and complex programs under consideration.

The use of temporal and modal logics for the control theory of reactive systems has expanded recently [RP03, AVW03, TP03, JK05, PR05, RP05, Bri06], regarding the well established approaches based on formal languages [CL99, Won89, KG95]. Although the supremacy of logics' expressive power over regular languages is taken for granted, a clear connection between the two worlds has to our knowledge never been clearly established in the setting of control theory. In this article, we embed the results of the control theory of discrete event systems in a logical framework. We show a strict fragment of the branching-time modal mu-calculus logic [Koz83] whose formulas naturally extend statements based on regular prefix-closed languages, and which moreover maintains the maximally permissiveness existence of controllers.

In the considered fragment of the mu-calculus, only diamond and box modalities are allowed, with conjunctions and greatest fix-points. We call this fragment the *Conjunctive Nu-Calculus* and write it $\mathbf{L}_\nu$. A formula is interpreted as a set of languages (over a fixed alphabet), each language denoting the set of runs of some discrete event system.
From the expressiveness point of view, $\mathbf{L}_\nu$ is strictly less expressive than the full mu-calculus as neither eventualities nor disjunction can be stated. However, the formalism allows for sentences that go beyond stating the membership in an interval of languages, or a finite union of those.
From the theoretical point of view, we study the logic $\mathbf{L}_\nu$ on its own; in particular, we introduce *modal specifications* as an alternative presentation of the logic. We prove that modal specifications and the logic $\mathbf{L}_\nu$ are equivalent in terms of their set of models, by giving a two-way effective translation from the former to the latter. Additionally, modal specifications support graphical representations, the *modal automata*, which generalize the standard finite state automata: modal automata allow for "must" (standard) and "may" transitions (Similar objects were considered by [Lar89] but for the purpose of studying the refinement of actions; no fix-points were taken into account). We establish several properties of modal specifications: the decomposition theorem, an algorithm making a modal specification coherent, and the maximal model theorem.
From the application point of view, we show that the modal specifications are strongly adequate for the description and the resolution of discrete event systems' centralized control problems, as originally studied by [RW89]. Indeed,

by the maximal model theorem, modal specifications are as good as regular languages: the existence of a maximally permissive solution does hold. Moreover, and contrary to regular languages, modal specifications can express liveness properties like "any stimulus a (if any) is followed by a reaction b". As explained in the paper, the procedure to compute a coherent specification achieves the controller synthesis, and it generalizes the algorithm of [RW89] for the computation of the supremal controllable sub-language.

The paper is organized as follows: Section 2 is dedicated to the introduction of the logic $\mathbf{L}_\nu$ and the modal specifications; in particular, we establish the decomposition theorem, an algorithm making a modal specification coherent, and the maximal model theorem, Theorem 22, resulting from the complete distributive lattice of the models. The decomposition theorem is technical but provides the constructive correspondence between $\mathbf{L}_\nu$ and the modal specifications (Theorem 11). Section 3 focuses on applications to control theory: we show that the setting captures intervals of (regular prefix-closed) languages and that it is strictly more expressive. However, instances of the centralized control problem, with $\mathbf{L}_\nu$-definable objectives, still have a maximal solution, as stated by Theorem 30. Finally, Section 4 summarizes the impact of the contribution.

2 Conjunctive Nu-calculus and Modal Specifications

In this article, we assume that the reader is familiar with formal language theory. From this point, we assume given $\Sigma = \{a_1, \dots a_n\}$ a finite alphabet. We consider languages over Σ, with $L, R \dots$ as typical elements and with the standard operations L^*, $L.a$ (with $a \in \Sigma$,), $L \cup R$, etc. The empty word is denoted 1. For all $u, v \in \Sigma^*$, $u.v$ is the concatenation of u and v and $u^* = \{u^k \mid k \in \mathbb{N}\}$ where u^k is the concatenation of k times the word u.

Let L be a language, we say that L is *prefix-closed* if and only if $1 \in L$ and for any non-empty word $a_1 \dots . a_m \in L$, we also have $a_1 \dots . a_{m-1} \in L$. The prefix-closure of L is the least prefix-closed language containing L. Given $w \in \Sigma^*$, we note $L_{/w} = \{v \in \Sigma^* \mid w.v \in L\}$ for the set of suffixes of w in L, and for $a \in \Sigma$, we note $L.a^{-1}$ the set $\{w \in \Sigma^* \mid w.a \in L\}$.

Let us remark that the empty language is not prefix-closed by definition, hence we will have to treat it separately when needed; in particular, for a prefix-closed language L, the language $L_{/w}$ is either prefix-closed – if $w \in L$ – or empty. In the following, L implicitly denotes a prefix-closed language.

2.1 The Conjunctive Nu-calculus

In this section we present a syntactic fragment of the modal mu-calculus [Koz83, AN01], called *the Conjunctive Nu-calculus*. We propose an interpretation of the formulas on prefix-closed languages rather than the more classic interpretation on (deterministic) labeled transitions systems. These interpretations coincide anyway, by arranging a language as a tree, namely the computation tree of the transition system. However, the language-based semantics makes the technical parts easier to reading and the comparisons with other works more immediate.

2.1.1 Syntax

Assume we are given a set of variables $Var = \{X, X_1, X_2, \ldots\}$. The set of nu-calculus formulas is denoted $\mathbf{L}_\nu$ and is defined by the following grammar:

$$(\mathbf{L}_\nu \ni) \quad \beta_1, \beta_2 ::= \texttt{true} \mid X \mid \ \rightarrow^a \ \mid [a]\beta_1 \mid \ \not\rightarrow^a \ \mid \beta_1 \wedge \beta_2 \mid \nu X.\beta_1(X)$$

where $a \in \Sigma$ and with the requirement that each occurrence of the variable X is under the scope of an even number of negation symbols $\neg$ in $\beta_1(X)$ for all formula $\nu X.\beta_1(X)$ – fixed-point-based formulas of the form $\nu X.\beta_1(X)$ will have a meaning.

We say that the variable X is *free in* β if it is not under the scope of any νX. The set of free variables in β is denoted $var(\beta)$. A formula β without any free variable is called a *sentence*.

2.1.2 Semantics

We interpret formulas on prefix-closed languages. The formula on L is the set of words of L satisfying the formula according to a given valuation *val*

of its free variables (if any); note that the set denoted by a formula is not necessarily prefix-closed.

The interpretation of $\beta \in \mathbf{L}_\nu$ on $L \subseteq \Sigma^*$ according to a valuation $val : Var \to L$ is written $[\![\beta]\!]_L^{[val]}$ and is given by induction on the structure of β as follows:

$$
\begin{array}{l}
[\![\texttt{true}]\!]_L^{[val]} = L \\
[\![X]\!]_L^{[val]} = val(X) \\
[\![\rightarrow^a]\!]_L^{[val]} = \{w \in L \mid w.a \in L\} \\
[\![\not\rightarrow^a]\!]_L^{[val]} = \{w \in L \mid w.a \notin L\} \\
[\![[a]\beta]\!]_L^{[val]} = \{w \in L \mid w.a \in [\![\beta]\!]_L^{[val]}\} \cup \{w \in L \mid w.a \notin L\} \\
[\![\beta_1 \wedge \beta_2]\!]_L^{[val]} = [\![\beta_1]\!]_L^{[val]} \cap [\![\beta_2]\!]_L^{[val]} \\
[\![\nu X.\beta(X)]\!]_L^{[val]} = \bigcup \{V \subseteq L \mid [\![\beta]\!]_L^{[val(V/X)]} \supseteq V\}
\end{array}
$$

As the semantics of sentences does not depend on the valuation val, we simply write $[\![\beta]\!]_L$. We say that "L *satisfies* the sentence β" – $L \models \beta$ for short – if and only if $1 \in [\![\beta]\!]_L$. We take the convention to note $\mathbf{Inv}(\beta)$,the formula $\nu X.\beta \wedge \bigwedge_{a \in \Sigma} [a]X$. $\mathbf{Inv}(\beta)$ is true if and only if $L_{/w} \models \beta$ for all $w \in L$. Also, we write $\langle a \rangle \beta$ for the formula $[a]\beta \wedge \rightarrow^a$. By definition, $[\![\langle a \rangle \beta]\!]_L^{[val]}$ is $\{w \in L \mid w.a \in [\![\beta]\!]_L^{[val]}\}$. The logic $\mathbf{L}_\nu$ is strictly less expressive than the full mu-calculus, since for example the operators $\vee$ (disjunction), $\mu X.\beta(X)$ (the least fix-point) and `false` cannot be expressed in $\mathbf{L}_\nu$. Notice that the operator $[a]\beta$ contains a flavor of disjunction as its meaning is $\langle a \rangle \beta \vee \not\rightarrow^a$.

2.2 Modal Specifications and Their Models

We propose a new formalism, the *modal specifications*, to specify sets of prefix-closed languages. Interestingly in the next section, we show that the modal specifications and the sentences of $\mathbf{L}_\nu$ coincide. Although at first sight modal specifications may be complicated objects, their graphical representations are merely finite state automata with two types of transitions. We investigate theoretical features before applying the framework to the control theory of discrete event systems.

2.2.1 Modal Specifications

Definition 1 *A modal specification is a tuple* $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ *where the* C_a*'s and* I *are regular languages.* C_a *is the set of words in which extension by an* a *is valid (in a sense to be formalized below) and* I *is a set of forbidden words.*

The *completion operator* associated to S is the application $C_S : 2^{\Sigma^*} \to 2^{\Sigma^*}$ defined by : $C_S(L) = \bigcup_{a\in\Sigma}(L \cap C_a).a$.

Models of a modal specification are prefix-closed languages; formally:

$$mod(S) = \{L \subseteq \Sigma^* \mid C_S(L) \subseteq L \wedge L \cap I = \emptyset\}$$

S is *satisfiable* if $mod(S) \neq \emptyset$ and L is a *model* of S if $L \in mod(S)$. By interpreting the definition of $mod(S)$, we have L is a model of S if and only if the two following conditions hold: (1) for each w of L in C_a, $w.a$ is also in L, and (2) no word of L may be in I.

In general, models are not regular, but because modal specifications are $\mathbf{L}_\nu$ sentences (Theorem 11), hence mu-calculus sentences, they inherit from the *finite model property*: if S is satisfiable, then $mod(S)$ has a regular element.

2.2.2 Graphical Representations

Modal specifications can be represented graphically, as *modal automata*. Given a modal specification S, its modal automaton possesses an initial state, no final state, and possibly two types of edges according to the components of S. Actually, it is easier to interpret modal automata as modal specifications as made clear by the example of Figure 1: the alphabet Σ is $\{a, b, c\}$. The automaton has solid and dashed edges, and some edges are missing for particular actions. The picture should be interpreted as follows:

- An a-labeled solid edge exiting from a state q means that any model L of S containing a word w, which leads to the state q in the automaton, contains also the word $w.a$. This constraint is read "in q, the transition a *must* exist".

- An a-labeled dashed edge exiting from a state q dually means "the transition a *may* exist".
- Finally, the absence of an a-labeled edge (of any kind) exiting from a state q means "the transition a is *forbidden*".

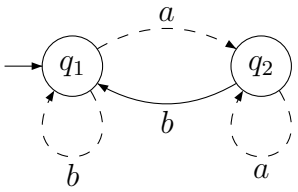


Figure 1: An example of a modal automaton

Assuming a modal automaton is given. We note $\mathcal{L}q$ the language recognized by the automaton when interpreted in a classical manner (the dashed edges are made solid) and q is assumed final. Now, the three informal rules above lead to the modal specification $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ where: given any state q of the automaton:

- An a-labeled solid edge exiting q stands for $\mathcal{L}q \subseteq C_a$.
- Absence of an a-labeled edge exiting q stands for $\mathcal{L}q.a \subseteq I$.

Notice that dashed edges stand only for the structure. Henceforth, from the automaton of Figure 1, as $\mathcal{L}q_1 = (a^*b^+)^*$ and $\mathcal{L}q_2 = (a+b)^*.a$, the corresponding specification is $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ with $C_a = \emptyset$, $C_b = (a+b)^*.a$, $C_c = \emptyset$ and $I = (a+b)^*.c$.

Let us consider another example, useful in Section 3. Consider the sentence $\phi = \mathbf{Inv}(\bigwedge_{a\in\Sigma}\langle a\rangle \mathtt{true})$. We claim that the models of ϕ are those of the specification $S_{\Sigma'}$ depicted in Figure 2. $mod(S_{\Sigma'})$ is the set $\{L \subseteq \Sigma^* \mid L.\Sigma' \subseteq L\}$. The sentence ϕ states any event of $S_{\Sigma'}$ can always occur, so does the picture of Figure 2.

2.2.3 Coherency and S-closure

A modal specification $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ is *coherent* if, whenever S is satisfiable, $I \cap C_S(\Sigma^*) = \emptyset$.

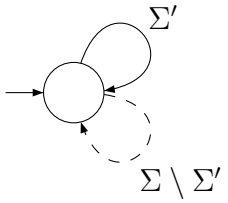


Figure 2: The modal specification $S_{\Sigma'}$

In a coherent specification, for every word w (if any), no action a is both required ($w \in C_a$) and forbidden ($w.a \in I$) concurrently.

Lemma 2 *Every modal specification is equivalent model-wise to a coherent modal specification.*

Proof Let $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ be the original specification, maybe not coherent. If $I \cap C_S(\Sigma^*) \neq \emptyset$, we compute a new specification $S' = \langle \{C'_a\}_{a\in\Sigma}, I'\rangle$ with $C'_a = C_a \setminus I.a^{-1}$ for each $a \in \Sigma$, and $I' = I \cup \bigcup_{a\in\Sigma}(I.a^{-1} \cap C_a)$. Since the obtained specification S' need not be coherent, this computation must be iterated.

We define a component-wise mapping F between modal specifications. $S' = \langle \{C'_a\}_{a\in\Sigma^*}, I'\rangle$ is mapped onto $S'' = \langle \{C''_a\}_{a\in\Sigma^*}, I''\rangle$ according to:

$$\left\{ \begin{array}{lcll} D''_a & = & D'_a \cup I'.a^{-1} \cup D_a & \forall a \in \Sigma^* \\ I'' & = & I' \cup \bigcup_{a\in\Sigma^*}(D'_a \cap I'.a^{-1}) \cup I & \end{array} \right.$$

where D_a, D'_a and D''_a are the respective complementary sets (in Σ^*) of C_a, C'_a and C''_a. Since modal specifications are tuples of languages, the mapping can also be seen as a monotonic function over languages (in the complete product lattice). By the Tarski Theorem, fix-points exist. In particular, the least fix-point of F, say T, is a coherent modal specification s.t. $mod(T) = mod(S)$:

Write $T = \langle \{C^T_a\}_{a\in\Sigma^*}, I^T\rangle$, and define $D^T_a = \Sigma^* \setminus C^T_a$. Since T is a fix-point of F, we have for all $a \in \Sigma$, $D^T_a = D^T_a \cup I^T.a^{-1} \cup D_a$, or equivalently $I^T.a^{-1} \subseteq D^T_a$; by complementation $C^T_a \cap I^T.a^{-1} = \emptyset$. This is sufficient to prove that $I^T \cap C_T(\Sigma^*) = \emptyset$, which make T a coherent specification.

Now, we prove that $mod(T) = mod(S)$. Consider the bottom element $\perp$ of the lattice: $\perp$ is the "empty" modal specification (all components are empty sets). Now, .

We prove that each iteration as above (except the first one, where $F(\perp) = S$) preserves the set of models: namely, $mod(S') = mod(S'')$.

$mod(S') \subseteq mod(S'')$: let $L \in mod(S')$. Since $C''_a \subseteq C'_a$ for each $a \in \Sigma^*$, we have $C_{S''}(L) \subseteq C_{S'}(L)$ and since $C_{S'}(L) \subseteq L$, we get $C_{S''}(L) \subseteq L$. For all $w \in L$ and for all $a \in \Sigma$, we have $w \notin D'_a \cap I'.a^{-1}$, otherwise $w.a$ would belong to $I' \cap L$, since $w \in C'_a$; we deduce that, since $L \cap I' = \emptyset$, we have $w \notin I''$. We have proved that $L \cap I'' = \emptyset$ and $C_{S''}(L) \subseteq L$ for all $L \in mod(S')$, this shows that $mod(S') \subseteq mod(S'')$.

$mod(S'') \subseteq mod(S')$: let $L \in mod(S'')$, since $I' \subseteq I''$, $L \cap I'' = \emptyset$ implies $L \cap I' = \emptyset$. For each $w \in L$, $w \in C'_a$ implies $w \in C''_a$, otherwise w would belong to $I'.a^{-1} \cap C'_a$ which implies $w \in I''$; we deduce that $C_{S'}(L) \subseteq C_{S''}(L) \subseteq L$. We have proved that $L \cap I' = \emptyset$ and $C_{S'}(L) \subseteq L$, hence $L \in mod(S'')$. □

From now on, we assume only satisfiable and coherent modal specifications.

Let $S = \langle \{C_a\}_{a \in \Sigma}, I \rangle$ be a modal specification, and let L be a language s.t. $L \cap I = \emptyset$ but not necessarily s.t. $C_S(L) \subseteq L$. We can "complete" L in order to obtain a model of S.

Definition 3 (S-closure) *The S-closure of a prefix-closed language L, denoted $L{\uparrow}^S$, is the least language L' s.t. $L \subseteq L'$ and $L' \in mod(S)$.*

Lemma 4 *The S-closure of a regular language is a regular language.*

Proof We build a finite state automaton $\mathcal{A}'$ which recognized language $\mathcal{L}(\mathcal{A}')$ is $L{\uparrow}^S$: Let $\mathcal{A}$ be an automaton for L. Build the automaton for $L \cup \bigcup_{a \in \Sigma} C_a.a$, and remove from it all the non-terminal states. Call the resulting automaton $\mathcal{A}'$.

Clearly $\mathcal{A}'$ recognizes the greatest prefix-closed language contained in $L \cup \bigcup_{a \in \Sigma} C_a.a$. Since L is prefix-closed, then $L \subseteq \mathcal{L}(\mathcal{A}')$ and obviously $L{\uparrow}^S \subseteq \mathcal{L}(\mathcal{A}')$; now if $L{\uparrow}^S \neq \mathcal{L}(\mathcal{A}')$ then, since $L{\uparrow}^S$ (and $\mathcal{L}(\mathcal{A}')$) is prefix-closed, there exist $w \in L{\uparrow}^S$ and $a \in \Sigma$ s.t. $w.a \in \mathcal{L}(\mathcal{A}')$ but $w.a \notin L{\uparrow}^S$, thus $w \in C_a$, which contradicts the fact that $L{\uparrow}^S$ is a model of S . □

The S-closure can be characterized also according to:

Lemma 5 *The S-closure of a prefix-closed language L is the least solution of the equation $R = L \cup C_S(R)$.*

Proof By definition $L\uparrow^S \in mod(S)$, then $C_S(L\uparrow^S) \subseteq L\uparrow^S$. Since $L \subseteq L\uparrow^S$, we get $L \cup C_S(L\uparrow^S) \subseteq L\uparrow^S$. From $L\uparrow^S$ being the least language we get the equality $L\uparrow^S = L \cup C_S(L\uparrow^S)$. □

Referring to Figure 1, the S-closure of a^* is $(a^* \cup a^*.b)$.

2.2.4 The Theorem of Decomposition

We show that each modal specification decomposes into *atomic* modal specifications. The Decomposition Theorem is at the base of the Theorem of Equivalence (next section) stating that modal specifications and $\mathbf{L}_\nu$ sentences are equivalent.

Definition 6 (Atomic specifications and operators)
Let $S_1 = \langle \{C_a^1\}_{a\in\Sigma}, I^1 \rangle$ and $S_2 = \langle \{C_a^2\}_{a\in\Sigma}, I^2 \rangle$ be two modal specifications.

- *The* intersection *of S_1 and S_2 is*

$$S_1 \cap S_2 = \langle \{C_a^1 \cup C_a^2\}_{a\in\Sigma}, I^1 \cup I^2 \} \rangle$$

- *Given a regular language $R \subseteq \Sigma^*$, the* prefixing *of S_1 by R is*

$$R.S_1 = \langle \{R.C_a^1\}_{a\in\Sigma}, R.I^1 \rangle$$

Lemma 7 $mod(S_1 \cap S_2) = mod(S_1) \cap mod(S_2)$.

Proof Let $L \in mod(S_1 \cap S_2)$. We have $C_{S_1\cap S_2}(L) \subseteq L$, and because $C_{S_1\cap S_2}(L) = C_{S_1}(L) \cup C_{S_2}(L)$, we get $C_{S_1}(L) \subseteq L$ and $C_{S_2}(L) \subseteq L$. Moreover $I_1 \cap L = I_2 \cap L = \emptyset$ follow from $(I_1 \cup I_2) \cap L = \emptyset$. Hence $L \in mod(S_1) \cap mod(S_2)$.

Reciprocally, $L \in mod(S_1) \cap mod(S_2)$ implies $C_{S_1}(L) \cup C_{S_2}(L) \subseteq L$ and $I_1 \cap L = I_2 \cap L = \emptyset$, which concludes the proof. □

L is a model of $R.S$ whenever for any word w of R, the language $L_{/w}$ (the set of suffixes of w in L) is a model of S:

Lemma 8 *$L \in mod(R.S)$ if and only if $\forall w \in R, L_{/w} = \emptyset$ or $L_{/w} \in mod(S)$.*

Proof For this proof, the two implications below are used several times without being mentioned explicitly:

$$\begin{array}{lcl} L \cap v.\Sigma^* = v.L_{/v} & \Rightarrow & L \cap v.L' = v.(L_{/v} \cap L') \\ w.L_1 \subseteq L_2 & \Rightarrow & L_1 \subseteq L_{2/w} \end{array}$$

$\Rightarrow$) Let $L \in mod(R.S)$, by definition:

$$C_{R.S}(L) = \bigcup_{a\in\Sigma}(L \cap R.C_a).a = \bigcup_{a\in\Sigma}\bigcup_{w\in R}(L \cap w.C_a).a = \bigcup_{a\in\Sigma}\bigcup_{w\in R} w.(L_{/w} \cap C_a).a$$

Since $C_{R.S}(L) \subseteq L$, for all $w \in R$ and for all $a \in \Sigma^*$, $w.(L_{/w} \cap C_a).a \subseteq L$. Then $C_s(L_{/w}) \subseteq L_{/w}$. Similarly, since $L \cap R.I = \emptyset$, we get $L \cap w.I = \emptyset$ and then $L_{/w} \cap I = \emptyset$, for all $w \in R$. Finally $L_{/w} = \emptyset$ or $L_{/w} \in mod(S)$.

$\Leftarrow$) We first show that $(L \cap R.C_a).a \subseteq L$, for all $a \in \Sigma$. if $v \in L \cap R.C_a$, there exist $w \in R$ and $u \in C_a$ s.t. $v = wu$. Then $u \in L_{/w} \cap C_a$ with $L_{/w} \neq \emptyset$. By hypothesis, $L_{/w} \in mod(S)$, then $(L_{/w} \cap C_a).a \subseteq L_{/w}$ and in particular $u.a \in L_{/w}$. We deduce $v.a = w.u.a \in L$. We show now that $L \cap R.I = \emptyset$: for all $w \in R$, if $L_{/w} = \emptyset$ then $L \cap w.I = \emptyset$; otherwise $L_{/w} \in mod(S)$ then $L \cap w.I = \emptyset$; finally $L \cap R.I = \emptyset$. □

Definition 9 *We define the following* atomic *specifications:*
$S_{\mathtt{true}} = \langle \{\emptyset\}_{a\in\Sigma}, \emptyset \rangle$,
$S_{\not\rightarrow^b} = \langle \{\emptyset\}_{a\in\Sigma}, \{b\} \rangle$, *for each $b \in \Sigma$, and*
$S_{\rightarrow^b} = \langle \{C_a\}_{a\in\Sigma}, \emptyset \rangle$, *where C_a is $\emptyset$ if $a \neq b$, and $\{1\}$ otherwise.*

By definition, we have:

$$\begin{array}{lcl} mod(S_{\mathtt{true}}) & = & \{L \in \Sigma^*\} \\ mod(S_{\not\rightarrow_b}) & = & \{L \subseteq \Sigma^* \mid b \notin L\} \\ mod(S_{\rightarrow_b}) & = & \{L \subseteq \Sigma^* \mid b \in L\} \end{array}$$

Theorem 10 (of Decomposition) *Each modal specification can be decomposed as a combination of atomic ones, according to the intersection and the language-prefixing operators*

Proof Let $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ be a modal specification. For each a in Σ, define the set $I^a = \{u \in \Sigma^* \mid u.a \in I\}$, and let S' be the specification defined by

$$S' = \bigcap_{a\in\Sigma} C_a.S_{\to a} \cap \bigcap_{a\in\Sigma} I^a.S_{\not\to a}$$

$S = S'$, by applying the Definitions 6 and 9. □

2.3 The Theorem of Equivalence

This section is dedicated to the proof of the following theorem:

Theorem 11 *For each set E of prefix-closed languages, E is formed by all the languages which satisfy some $\mathbf{L}_\nu$ sentence if and only if E coincide with a set $mod(S)$ of modal specification S.*

In order to prove this theorem, we introduce the notion of variable paths:

Definition 12 (variable paths) *To each $\beta \in \mathbf{L}_\nu$ (not necessarily closed), we associate a mapping $P_\beta : var(\beta) \to \mathcal{P}(\Sigma^*)$; for $X \in var(\beta)$, the language $P_\beta(X)$ is the set of* variable paths of X in β. *It is defined by induction on the structure of β:*

- $P_{\mathtt{true}}(X) = P_{\to^a}(X) = P_{\not\to^a} = \emptyset$,
- $P_Y(X) = \emptyset$ *if* $Y \neq X$, *and* $P_X(X) = \{1\}$,
- $P_{[a]\beta}(X) = a.P_\beta(X)$,
- $P_{\beta_1\wedge\beta_2}(X) = P_{\beta_1}(X) \cup P_{\beta_2}(X)$,
- $P_{\nu Y.\beta(Y)}(X) = P_\beta(Y)^*.P_\beta(X)$.

The variable paths of X in β are the words that "lead" to an occurrence of X in the formula: when $w \in [\![\beta]\!]_L^{[val]}$, $P_\beta(X)$ is the set of words v s.t. $w.v \in [\![X]\!]_L^{[val]}$ or equivalently $w.v \in val(X) \subseteq L$. For example, $P_{[a]X}(X) = \{a\}$, $P_{[a][b]X \wedge [c]X}(X) = (a.b + c)$, and $P_{\nu Y.([a][b]Y \wedge [c]X)}(X) = (a.b)^*.c$.

2.3.1 From a Sentence to a Specification

We show here how to construct a modal specification S_β from a sentence β of $\mathbf{L}_\nu$ s.t. $mod(S_\beta)$ is the set of languages satisfying β. The proof is constructive. As it is led by induction over the sentence β, we need to introduce some technical material: since modal specifications are not designed to deal with valuations, consider the following property involving a valuation val, a formula β, a language L and a word w of L:

$$\forall X \in var(\beta), w.P_\beta(X) \cap L \subseteq val(X) \tag{1}$$

Property (1) states that the words of L which belong to the the variable paths of X also belong to $val(X)$.

Definition 13 (Modal specification associated to a formula of $\mathbf{L}_\nu$) *The modal specification* S_β associated to a formula $\beta \in \mathbf{L}_\nu$ *is defined inductively over the structure of* β *as follows:*

- *for* $\beta \in \{\mathtt{true}, \rightarrow^a, \not\rightarrow^a\}$, *use Definition 9.*
- $S_X = S_{\mathtt{true}}$
- $S_{[a]\beta} = a.S_\beta$
- $S_{\beta_1 \wedge \beta_2} = S_{\beta_1} \cap S_{\beta_2}$
- $S_{\nu Y.\beta(Y)} = P_\beta(Y)^*.S_\beta$

The modal specification associated to $[a]\nu X.([b]X \wedge \rightarrow^a \wedge \not\rightarrow^c)$, depicted in Figure 3, is $(a.b^*).(S_{\rightarrow^a} \cap S_{\not\rightarrow^c})$. It is s.t. $C_a = (a.b^*)$, $C_b = \emptyset$, $C_c = \emptyset$, and $I = (a.b^*).c$.

Consider now the following general result and its corollary:

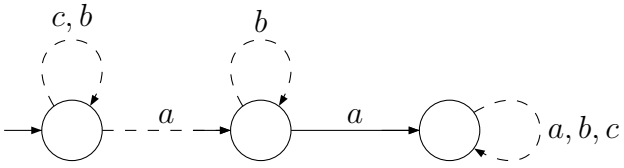


Figure 3: The Specification S_β

Proposition 14 *Let $\beta \in \mathbf{L}_\nu$, let val be a valuation, let L be a prefix-closed language, and let $w \in L$. We have:*

$w \in [\![\beta]\!]_L^{[val]}$ *if and only if* $L_{/w} \in mod(S_\beta)$ *and Hypothesis* (1) *is verified.*

The first corollary of Proposition 14 is:

Corollary 15 *$L \in mod(S_\beta)$ if and only if $L \models \beta$, for every $\mathbf{L}_\nu$ sentence β.*

Let $\beta \in \mathbf{L}_\nu$, *val* be a valuation, L be a prefix-closed language and w be a word of L. To prove proposition 14, we prove these 3 following lemmas:

Lemma 16 $w \in [\![\beta]\!]_L^{[val]}$ *implies Hypothesis* (1)*.*

Lemma 17 $w \in [\![\beta]\!]_L^{[val]}$ *implies* $L_{/w} \in mod(S_\beta)$*.*

Lemma 18 $L_{/w} \in mod(S_\beta)$ *and Hypothesis* (1) *imply* $w \in [\![\beta]\!]_L^{[val]}$*.*

Proof (of Lemma 16) by induction over the structure of β (where α is some $\mathbf{L}_\nu$ formula). Let $w \in [\![\beta]\!]_L^{[val]}$:

- if $\beta \in \{\mathtt{true}, \rightarrow^a, \not\rightarrow^a\}$, then $var(\beta) = \emptyset$.
- if $\beta = X$, then $var(\beta) = \{X\}$, and $P_X(X) = \{1\}$ and $w \in [\![X]\!]_L^{[val]}$, thus $w \in val(X)$ which implies $w.\{1\} \subseteq val(X)$.

- if $\beta = [a]\alpha$, then $var(\beta) = var(\alpha)$ and $P_\beta(X) = a.P_\alpha(X)$, then $w.P_\beta(X) \cap L = w.a.P_\alpha(X) \cap L$. Since $w.a \in [\![\alpha]\!]_L^{[val]}$, by ind. hyp., $w.P_\beta(X) \cap L \subseteq val(X)$.

- if $\beta = \beta_1 \wedge \beta_2$, then $w.P_\beta(X) \cap L = (w.P_{\beta_1}(X) \cap L) \cup (w.P_{\beta_2}(X) \cap L)$ and by ind. hyp., $(w.P_{\beta_1}(X) \cap L) \cup (w.P_{\beta_2}(X) \cap L) \in val(X)$.

- assume $\beta = \nu Y.\alpha(Y)$, and let us note $V = [\![\beta]\!]_L^{[val]}$; by definition, $V = [\![\alpha(X)]\!]_L^{val(V/X)}$, that is $w \in [\![\beta]\!]_L^{[val]}$ is equivalent to $w \in [\![\alpha(Y)]\!]_L^{[val(V/Y)]}$.
  We show by induction on n that $w.P_\alpha(Y)^n.P_\alpha(X) \cap L \subseteq val(X)$:

  - For $n = 0$, we use the ind. hyp. over β to obtain $w.P_\alpha(X) \cap L \subseteq val(X)$.
  - For $n + 1$, $w.P_\alpha(Y)^{n+1}.P_\alpha(X) \cap L$ rewrites as $w.P_\alpha(Y).P_\alpha(Y)^n.P_\alpha(X) \cap L$.
    By ind. hyp. over β, $w \in [\![\alpha(Y)]\!]_L^{[val(V/Y)]}$ then $w.P_\alpha(Y) \cap L \subseteq V$ and then for all $v \in w.P_\alpha(Y) \cap L$, $v \in V$; now, y ind. hyp. over n, we get $v.P_\alpha(Y)^n.P_\alpha(X) \cap L \subseteq val(X)$ and finally $w.P_\alpha(Y).P_\alpha(Y)^n.P_\alpha(X) \cap L \subseteq val(X)$

□

Proof (of Lemma 17) by induction over the structure of β:

- if $\beta \in \{\texttt{true}, \rightarrow^a, \not\rightarrow^a, X\}$, by Definition 9, $L_{/w} \in mod(S_\beta)$.

- if $\beta = [a]\alpha$, then $S_\beta = a.S_\alpha$, and if $a \in L_{/w}$ then $w.a \in [\![\alpha]\!]_L^{[val]}$. By ind. hyp., $L_{/w.a} \in mod(S_\alpha)$, which implies $L_{/w} \in mod(S_\beta)$ (Lemma 8).

- if $\beta = \beta_1 \wedge \beta_2$, using Lemma 7 entails $mod(S_\beta) = mod(S_{\beta_1}) \cap mod(S_{\beta_2})$. By ind. hyp., $L_{/w} \in mod(S_\beta)$.

- assume $\beta = \nu X.\alpha(X)$. As above, by letting $V = [\![\beta]\!]_L^{[val]}$, we have $V = [\![\alpha(X)]\!]_L^{val(V/X)}$. We show by ind. hyp. over n that for all $v \in (P_{\alpha(X)}(X))^n$, $w.v \in L$ implies $L_{/w.v} \in mod(S_\alpha)$.

- For $n = 0$, $w \in [\![\alpha(X)]\!]_L^{val(V/X)}$ and ind. hyp. over β, $L_{/w} \in mod(S_\alpha)$.
- For $n+1$, $w \in V$ and since $v = u.u'$ with $u \in P_{\alpha(X)}(X)$, by Lemma 16 we have $(u' \in L_{/w.u})$ implies $L_{/w.u.u'} \in val(X) = V$. It follows by ind. hyp. over n that $L_{/w.u.u'} \in mod(S_\alpha)$, since $u' \in (P_{\alpha(X)}(X))^n$.

Finally, for all $v \in (P_{\alpha(X)}(X))^*$, $w.v \in L$ implies $L_{/w.v} \in mod(S_\alpha)$. Applying Lemma 8 entails $L_{/w} \in mod(S_\beta)$.

□

Proof (of Lemma 18) by induction over β:

- if $\beta \in \{\texttt{true}, \rightarrow^a, \not\rightarrow^a X\}$, by Definition 9, $w \in [\![\beta]\!]_L^{[val]}$.
- if $\beta = [a]\alpha$, $S_\beta = a.S_\alpha$, if $a \in L_{/w}$ then Lemma 8 guarantees $L_{/w.a} \in mod(S_\alpha)$; by ind. hyp., $w.a \in [\![\alpha]\!]_L^{[val]}$. In both cases $w \in [\![\beta]\!]_L^{[val]}$,
- if $\beta = \beta_1 \wedge \beta_2$, then $S_\beta = S_{\beta_1} \cap S_{\beta_2}$. By Lemma 7, $L_{/w} \in mod(S_{\beta_1}) \cap mod(S_{\beta_2})$, and by Definition 12 and Hypothesis (1), for all $v \in P_\beta(X)$, $v \in P_{\beta_1}(X) \cup P_{\beta_2}(X)$. We can now apply the ind. hyp. over β_1 and β_2 to conclude that $w \in [\![\beta]\!]_L^{[val]}$.
- assume $\beta = \nu X.\alpha(X)$. We show that $L \cap w.P_\alpha(X)^*$ is a post fix-point:

$$L \cap w.P_\alpha(X)^* \subseteq [\![\alpha]\!]_L^{[val((L \cap w.P_\alpha(X)^*)/X)]}$$

For all $v \in L_{/w} \cap w.P_\alpha(X)^*$:

1. $w \in mod(S_\beta)$ implies $w.v \in mod(S_\alpha)$, by Lemma 8),
2. For all $Y \in var(\beta)$, with $Y \neq X$, $w.P_\beta(Y) \cap L \subseteq val(Y)$ and $P_\beta(Y) = (P_\alpha(X))^*.P_\alpha(Y)$ imply

$$w.v.P_\alpha(Y) \cap L \subseteq val(Y)$$

3. (For X) $v \in L_{/w} \cap P_\alpha(X)^*$ implies

$$w.v.P_\alpha(X) \cap L \subseteq L \cap w.P_\alpha(X)^*) = val((L \cap w.P_\alpha(X)^*)/X)(X)$$

Points 2. and 3. above entail Hypothesis (1), which combined with Point 1. allow to apply the ind. hyp. to obtain
$w.v \in [\![\alpha]\!]_L^{[val((L \cap w.P_\alpha(X)^*)/X]}$.

Since $(L \cap w.P_\alpha(X)^*)$ is a post-fix-point, if $w \in (L \cap w.P_\alpha(X)^*)$, then $w \in [\![\beta]\!]_L^{[val]}$. □

Now the proof of Proposition 14 is immediate:

Proof (of Proposition 14) Lemma 18 gives the $\Rightarrow$) direction, and Lemmas 16 and 17 give the $\Leftarrow$) direction. □

2.3.2 From a Specification to a Sentence

Given a modal specification S, we construct a sentence $\beta_S \in \mathbf{L}_\nu$ s.t. the languages satisfying β_S coincide with the elements of $mod(S)$. To do so, we decompose S into atomic modal specifications (according to Theorem 10) and we construct β_S step by step s.t. S_{β_S} and S are equal component-wise, henceforth equal model-wise.

Lemma 19 *For any regular language $R \subseteq \Sigma^*$, we can construct a formula $\alpha_R(X) \in \mathbf{L}_\nu$ s.t. $S_{\alpha_R(\beta/X)} = R.S_\beta$, for any sentence $\beta \in \mathbf{L}_\nu$.*

Proof by induction over a regular expression of the language R. We consider the following grammar for the regular expressions generating regular languages (where $a \in \Sigma$): $\{1\} \mid a.R_1 \mid R_1 \cup R_2 \mid R_1^*$.

We define $\alpha_R(X)$ inductively over (the regular expression of) R, and simultaneously prove that $S_{\alpha_R(\beta/X)} = R.S_\beta$ and $P_{\alpha_R(X)}(X) = R$, for all $\beta \in \mathbf{L}_\nu$.

- $R = \{1\}$: define $\alpha_{\{1\}}(X) = X$. We trivially have $S_{\alpha_{\{1\}}(\beta/X)} = \{1\}.S_\beta$ and $P_{\alpha_{\{1\}}(X)}(X) = \{1\}$,
- $R = a.R_1$: define $\alpha_{a.R_1}(X) = [a]\alpha_{R_1}(X)$. From Definition 13, $S_{\alpha_{a.R_1}(\beta/X)} = a.R_1.S_\beta$, and from Definition 12, $P_{\alpha_{a.R_1}(X)}(X) = a.R_1$.

- $R = R_1 \cup R_2$: define $\alpha_{R_1 \cup R_2}(X) = \alpha_{R_1}(X) \wedge \alpha_{R_2}(X)$.
  From Definition 13, $S_{\alpha_{R_1 \cup R_2}(\beta/X)} = (R_1 \cup R_2).S_\beta$, and from Definition 12, $P_{\alpha_{R_1 \cup R_2}(X)}(X) = R_1 \cup R_2$.

- $R = R_1^*$: define $\alpha_{R_1^*}(X) = \nu Y.\alpha_{R_1}(X/Y) \wedge X$. Since by ind. hyp. $P_{S_{\alpha_{R_1}(Y/X)}} = R_1.S_\beta$, by Definition 13, $S_{\alpha_{R_1^*}(\beta/X)} = R_1^*.S_\beta = R_1^*.S_\beta$. It follows immediately from Definition 12 that $P_{\alpha_{R_1^*}(X)}(X) = R_1^*$.

□

Lemma 20 *For any modal specification S, we can construct a sentence $\beta_S \in \mathbf{L}_\nu$ s.t. $mod(S)$ coincides with the set of languages satisfying β_S.*

Proof By Theorem 10, we can decompose S, and build a sentence β_S s.t. $S_{\beta_S} = S$. The only nontrivial operator is the language-prefixing which is given by Lemma 19. □

Consider the modal specification S of Figure 4. Below, the first line is the decomposition of S and the second line is the formula β_S:

$$\begin{array}{cccccccccccc} S_{\not\rightarrow^b} & \cap & a. & (& a.(b.a)^*.S_{\rightarrow^b} & \cap & b. & (a.b)^*.S_{\rightarrow^a} &) \\ \not\rightarrow^b & \wedge & \rightarrow^a \wedge [a] & (& [a]\nu X.(\,[b][a]X \wedge \rightarrow^b) & \wedge & [b] & \nu Y.(\,[a][b]X \wedge \rightarrow^a) &) \end{array}$$

2.3.3 Main Proof

The previous results combine to prove Theorem 11: the logic $\mathbf{L}_\nu$ and the class of modal specifications are equal regarding the models. Indeed, if E is the set of languages satisfying some sentence $\beta \in \mathbf{L}_\nu$, then by Corollary 15, $E \subseteq mod(S_\beta)$. Reciprocally, if E is the set of models of some specification S, then by Lemma 20, we construct β_S s.t. $E = \{L \mid L \models \beta_S\}$.

2.4 The Lattice of Models

For any satisfiable specification S, we show that the set $mod(S)$ forms a lattice which extrema can easily be built.

In the following, we fix $S = \langle \{C_a\}_{a\in\Sigma}, I\rangle$ a coherent modal specification, and we distinguish the two following elements of $mod(S)$: $\{1\}\uparrow^S$ and $\Sigma^*\backslash I.\Sigma^*$, respectively written $L^S_\bot$ and $L^S_\top$.

Lemma 21 *The following four statements are equivalent:*

1. *S is satisfiable*
2. *$L^S_\bot \in mod(S)$*
3. *$L^S_\bot \cap I = \emptyset$*
4. *$L^S_\top \in mod(S)$*

Proof Since $2 \Rightarrow 1$, $4 \Rightarrow 1$ et $2 \Rightarrow 3$ are trivial, we only prove $3 \Rightarrow 2$, $1 \Rightarrow 4$ and $4 \Rightarrow 3$.

- $3 \Rightarrow 2$: by Lemma 5, $L^S_\bot = \{1\} \cup C_S(L^S_\bot)$. Hence $C_S(L^S_\bot) \subseteq L^S_\bot$, and by assumption $L^S_\bot \cap I = \emptyset$. We conclude $L^S_\bot \in mod(S)$;
- $1 \Rightarrow 4$: by monotonicity, $C_S(L^S_\top) \subseteq C_S(\Sigma^*)$. By definition of $L^S_\top$, this is equivalent to $C_S(L^S_\top) \subseteq L^S_\top$. Moreover, as S is coherent, $C_S(\Sigma^*) \cap I = \emptyset$, which implies $L^S_\top \cap I = \emptyset$; finally $L^S_\top \in mod(S)$.
- $4 \Rightarrow 3$: from 4, we get $L^S_\top \cup C_S(L^S_\top) = L^S_\top$ and $L^S_\top \cap I = \emptyset$. Now from the former and the fact that C_S is monotonic, we have $L^S_\bot \subseteq \{1\} \cup C_S(L^S_\top) \subseteq L^S_\top$. By the latter and the last inclusion, $L^S_\bot \cap I = \emptyset$.

□

Lemma 21 latches the *finite model property* of the mu-calculus: if S is satisfiable, then it has a regular model; $L^S_\bot$ is regular by lemma 4, and so is $L^S_\top$ by definition). By definition, $L^S_\bot$ and $L^S_\top$ are, the extrema in the set $mod(S)$ ordered by inclusion : $L^S_\top$ is the greatest element and $L^S_\bot$ is the least element. Actually, we have the following:

Theorem 22 *Let S be a satisfiable modal specification. Then $(mod(S), \subseteq)$ is a distributive complete lattice.*

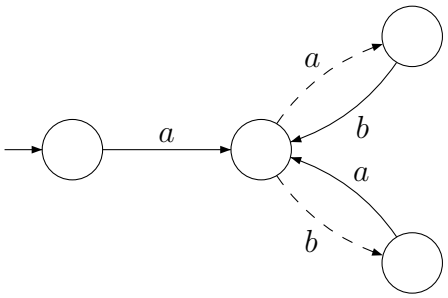


Figure 4: The modal specification S

Let S be the modal specification of Figure 4. Some models of S are depicted in Figure 5, and arranged s in the lattice. The boxes represent the models and the arrows between the boxes represent the language inclusion relation. We have $L_\perp = L1$ and $L_\top = L7$. It can be shown that there exist infinitely many models in $mod(S)$, as one can imagine between the elements $L2$ and $L4$, or between the elements $L3$ and $L5$, or between the elements $L4$ and $L7$. The model $L6$ shows that $L4 \cup L5 \neq L7$.

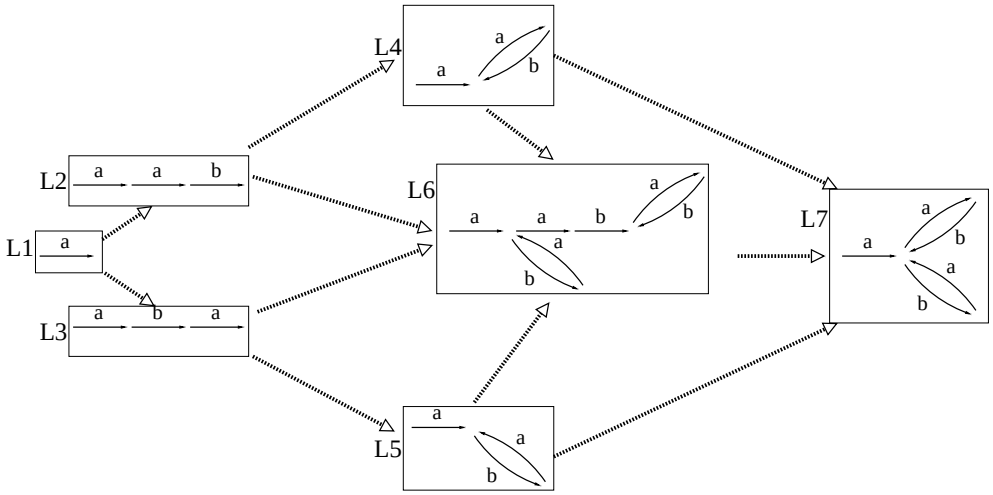


Figure 5: Some elements of the distributive lattice $mod(S)$.

3 Application to Control Theory

We consider the class of centralized control problems with total observation as in [RW89], and enlarge it to attain $\mathbf{L}_\nu$ definable control objectives.

By the previous section, we use modal specifications rather than logical formulas, mostly because they generalize finite state automata, more commonly used to describe interval of prefix-closed languages.

In the following, we first recall what the control theory as proposed by [RW89] is; next, we explain how centralized control problems for control objectives described by an interval of languages, translate into modal specifications. In the proposed setting, the various aspects of the control problem are easy to distinguish: for example, the plant decomposes into two separated modal specifications, the plant skeleton and its controllability features, whereas the additional modal specifications define the (possibly many) control objectives. A simple illustration is given by the Example 1.
We take the convention that the world "language" means "regular prefix-closed languages".

3.1 The Centralized Control Problem

Classically, one considers a set $\Sigma = \{e_1, e_2, \ldots, e_n\}$ of *events*, a finite state automaton G representing the *plant*, which is fully determined by the prefix-closed language $L \subseteq \Sigma^*$ of its trajectories. Then, two subsets Σ_c and Σ_{uc} partition the set of Σ into the uncontrollable and controllable events sets.

Given an ongoing trajectory of the plant, say $w \in L$, a controller indicates which event is allowed to prolong w; in particular, uncontrollable events should always be allowed. More formally, a controller is a function $\mathcal{C} : L \to 2^\Sigma$, with the constraint that $\mathcal{C}(w) \supseteq \Sigma_{uc}$; called the Σ_{uc}-*admissibility* property of the controller $\mathcal{C}$.

Given a controller $\mathcal{C}$, the controlled plant $\mathcal{C}/G$ has its trajectories in $L(\mathcal{C}/G) \subseteq L$, defined as the greatest sub-language of L s.t.: if L is empty then so is $L(\mathcal{C}/G)$, otherwise $1 \in L(\mathcal{C}/G)$, and for any $w \in L(\mathcal{C}/G)$, $w.e \in L(\mathcal{C}/G)$, provided $e \in \mathcal{C}(w)$.

We now turn to the control objectives. Classically, they are precisely given by a pair of prefix-closed languages, to form an interval: let $\mathtt{A}$ and $\mathtt{E}$ be two languages over Σ. The *interval of languages* $[\mathtt{A}, \mathtt{E}]$ is the set of all prefix-closed languages R s.t. $\mathtt{A} \subseteq R \subseteq \mathtt{E}$.

Definition 23 (Centralized Control Problem (CCP)) *An instance of the* **Centralized Control Problem (CCP)** *is a triple $\langle G, \Sigma_{uc}, [\mathtt{A}, \mathtt{E}]\rangle$, where G is a plant, $\{\Sigma_c, \Sigma_{uc}\}$ is a partition of Σ, and $[\mathtt{A}, \mathtt{E}]$. is an interval of languages. The problem consists in computing, when it exists, a Σ_{uc}-admissible controller $\mathcal{C}$ s.t. $L(\mathcal{C}/G) \in [\mathtt{A}, \mathtt{E}]$.*

In general, one seeks for a maximally permissive controller, when it exists: a maximally permissive controller is s.t. $\mathcal{C}(w)$ is a maximal set for all $w \in L$. When control objectives are intervals of languages, a maximally permissive controller always exists and it is unique. Notice that allowing for any mu-calculus definable control objective cuts off this nice property, either for the existence or for the uniqueness – this is precisely why non-blockingness issues lead to difficulties. For example, the control objective stating that "eventually no more b can happen" cannot be achieved by a maximally permissive controller if the plant's language is for example $\{a, b\}^*$. Indeed, postponing for ever the moment where b is eventually disallowed would violate the original objective (see [RP04]).

There is an important amount of contributions addressing the development of controller synthesis methods of $\mathcal{C}$ [Won89, CL99, GW00]. In the next section, we focus on the use of modal specifications for solving centralized control problems, with total observation.

3.2 Particular Modal Specifications

We first introduce modal specifications that are meaningful in the control theory of reactive systems.

Definition 24 *The modal specification $S_{\Sigma_{uc}} = \langle \{\Sigma^*\}_{e \in \Sigma_{uc}^G} \cup \{\emptyset\}_{e \in \Sigma_c^G}, \emptyset\rangle$ is called the* Σ_{uc}-controllability property.

$S_{\Sigma_{uc}}$ specifies Σ_{uc}-admissible controllers: indeed, $mod(S_{\Sigma_{uc}})$ coincides with the set of languages R s.t. $R.\Sigma_{uc} \cap \Sigma^* \subseteq R$ (that is the Σ_{uc}-controllability notion of [RW89]).

Definition 25 *Now, let K be a prefix-closed language, we define the two particular modal specifications $S_K = \langle \{K.e^{-1}\}_{e\in\Sigma}, \emptyset \rangle$, and $S^K = \langle \{\emptyset\}_{e\in\Sigma}, \Sigma^* \setminus K \rangle$. Recall that $K.e^{-1} = \{w \in \Sigma^* \mid w.e \in K\}$.*

We can establish the following:

Lemma 26 *Given language K, $mod(S_K)$ is the set $\{R \subseteq \Sigma^* \mid K \subseteq R\}$ and $mod(S^K)$ is the set $\{R \subseteq \Sigma^* \mid R \subseteq K\}$.*

Proof It is sufficient to prove that $K = L_\perp^{S_K}$. Since $K = \bigcup_{e\in\Sigma} K.e^{-1}.e$ which is equal to $C_{S_K}(\Sigma^*)$, we necessarily have $L_\perp^{S_K} \subseteq K$; reciprocally, we prove by induction over the word $u \in \Sigma^*$ that $u \in K$ implies $u \in L_\perp^{S_K}$. If $u = 1$, then it is trivial as both languages are prefix-closed. Let $u.e \in K$, then $u \in K.e^{-1}$, and because K is prefix-closed, $u \in K$. Now by induction hypothesis $u \in L_\perp^{S_K}$. Since $L_\perp^{S_K}$ is S_K-closed, it also contains $u.e$.
For S^K, it is sufficient to remark that $R \cap \Sigma^* \setminus K = \emptyset$, hence $R \subseteq K$. □

Definition 27 *Given K_1 and K_2 be two prefix-closed languages over Σ, the modal specification $S_{K_1}^{K_2} = S_{K_1} \cap S^{K_2}$ is called the modal specification* associated *to the interval of languages $[K_1, K_2]$.*

From Lemma 26, it is immediate that $mod(S_{K_1}^{K_2}) = [K_1, K_2]$.

3.3 The Control Theory as Modal Specifications

We combine several modal specifications in a conjunctive manner to characterize the set of solutions of the centralized control problems, with total observations. Because by Theorem 22 a modal specification denotes a lattice of languages, a maximally permissive solution always exists and is unique (the top element of the lattice); incidentally, so does a minimally permissive solution (the bottom element).

Proposition 28 *Consider an instance* $\langle G, \Sigma_{uc}, [\mathtt{A}, \mathtt{E}]\rangle$ *of the* **(CCP)**. *We can build a modal specification* S *s.t. the set of controllers* $\mathcal{C} : L \to 2^{\Sigma}$ *with* $L(\mathcal{C}/G) \in [\mathtt{A}, \mathtt{E}]$ *coincides with* $mod(S)$. *Moreover, the maximally and the minimally permissive controllers can be effectively computed.*

Proof We successively define three modal specifications:

The modal specification associated to the plant skeleton: Consider the modal specification $S^L = \langle \{\emptyset\}_{e\in\Sigma}, I_G\rangle$. Its modal automaton is like G but where all edges are in dashed lines, as in Figure 6.

The modal specification to obtain Σ_{uc}-admissible controllers: Consider $S_{\Sigma_{uc}} = \{\Sigma^*\}_{e\in\Sigma_{uc}} \cup \{\emptyset\}_{e\in\Sigma_c}, \emptyset\rangle$. Its picture is a single state automaton with looping transitions, those labeled by events in Σ_{uc} are solid, and those labeled by events in Σ_c are dashed, as the top left modal automaton of Figure 7.

The modal specification for the interval of languages $[\mathtt{A}, \mathtt{E}]$: Consider $S^{\mathtt{E}}_{\mathtt{A}}$ from Definition 27, and write it $S^{\mathtt{E}}_{\mathtt{A}} = \langle \{C_e\}_{e\in\Sigma}, I\rangle$.

The overall modal specification is S is $S^L \cap S_{\Sigma_{uc}} \cap S^{\mathtt{E}}_{\mathtt{A}}$

Theorem 29 *There is a one-to-one correspondence between the set* $mod(S)$ *and the controllers that are solutions of any instance of the* **(CCP)**.

Proof
Recall that by definition of $\cap$ we have $S = \langle \{C_e\}_{e\in\Sigma_c} \cup \{\Sigma^*\}_{e\in\Sigma_{uc}}, I \cup I_G\rangle$ Proving its correctness is done in two stages:

Let $M \in mod(S)$. In particular $M \in [\mathtt{A}, \mathtt{E}]$, because $M \in mod(S^{\mathtt{E}}_{\mathtt{A}})$. Also $M \subseteq L$ as $M \in mod(S^L)$.

We first prove that M is $L(\mathcal{C}_M/G)$, where $\mathcal{C}_M$ is some admissible controller. Define $\mathcal{C}_M : L \to 2^{\Sigma}$ by $\mathcal{C}_M(v) = \{e | v.e \in M\}$. $\mathcal{C}_M$ is admissible. let us compute $C_{S_{\Sigma_{uc}}}(M)$. By the definition of $S_{\Sigma_{uc}}$, $C_{S_{\Sigma_{uc}}}(M) = \bigcup_{e\in\Sigma_c}(\emptyset \cap M).e \cup \bigcup_{e\in\Sigma_{uc}}(\Sigma^* \cap M).e$, which reduces to $C_{S_{\Sigma_{uc}}}(M) = \bigcup_{e\in\Sigma_{uc}} M.e$. Henceforth, since $M \in mod(S_{\Sigma_{uc}})$, we have $C_{S_{\Sigma_{uc}}}(M) \subseteq M$, that is $\bigcup_{e\in\Sigma_{uc}} M.e \subseteq M$. This entails that $\mathcal{C}_M(v) \supseteq \Sigma_{uc}$ for all $v \in L$, namely $\mathcal{C}_M$ is admissible.

Reciprocally, let $\mathcal{C}$ be a Σ_{uc}-admissible controller with $L(\mathcal{C}/G) \in [\mathtt{A}, \mathtt{E}]$. We prove that $L(\mathcal{C}/G) \in mod(S)$. From $L(\mathcal{C}/G) \in [\mathtt{A}, \mathtt{E}]$, we get $L(\mathcal{C}/G) \in$

$mod(S^{\mathtt{E}}_{\mathtt{A}})$ which entails $C_{S^{\mathtt{E}}_{\mathtt{A}}}(L(\mathcal{C}/G)) \subseteq L(\mathcal{C}/G)$ and $L(\mathcal{C}/G) \cap I = \emptyset$. Also because $L(\mathcal{C}/G) \subseteq L$, and $L \cap I_G = \emptyset$, we get $L(\mathcal{C}/G) \cap I_G = \emptyset$. It remains to prove that $C_{S_{\Sigma_{uc}}}(L(\mathcal{C}/G)) \subseteq L(\mathcal{C}/G)$ to conclude that $L(\mathcal{C}/G) \in mod(S)$. Now by computing $C_{S_{\Sigma_{uc}}}(L(\mathcal{C}/G))$, similarly to $C_{S_{\Sigma_{uc}}}(M)$ above, we obtain $\bigcup_{e\in\Sigma_{uc}} L(\mathcal{C}/G).e$, which is contained in $L(\mathcal{C}/G)$, because $\mathcal{C}$ is assumed admissible. □

Now let us focus on the effective computation of the maximally and minimally permissive solutions. By Lemma 4, these solutions can effectively be computed according to:

1. the computation of the coherent modal specification equivalent to S, say the modal specification $S' = \langle \{C'_e\}_{e\in\Sigma}, I' \rangle$, and
2. if S' is non-empty, the computation $L^{S'}_{\top} = \Sigma^* \setminus I'.\Sigma^*$ and $L^{S'}_{\bot} = (\{1\})\uparrow^{S'}$ to obtain the maximally and minimally permissive solutions; those are regular, as expected.

□

We postpone to the next section the discussion on the complexity. Let us remark that, according to Lemma 20, the $\mathbf{L}_\nu$-formula $\beta_{S_{\Sigma_{uc}}}$ associated to $S_{\Sigma_{uc}}$ is $\mathbf{Inv}(\bigwedge_{e\in\Sigma_{uc}} \langle e \rangle \mathtt{true})$, as originally introduced by [RP03] and [AVW03].

3.4 Extended Control Objectives

In the light of the previous section, we continue to use the elegant setting of modal specifications. The progress is made at the level of the control objectives: from now on we allow for any $\mathbf{L}_\nu$-expressible property, say by the formula β. Basically, to have a logical description of an instance $\langle G, \Sigma_{uc}, \beta \rangle$ of the **(CCP)**; three modal specifications must be given: the specification S^L for the plant, the specification $S_{\Sigma_{uc}}$ which enforces admissible controllers, and the modal specification S_{obj} of the control objectives (actually equivalent to S_β).

Regarding $S_{\Sigma_{uc}}$, one can consider a more general specification S_c to define, e.g. several modes of control where uncontrollability of events depends on the

mode: the modal automaton then has as many states as modes, and dashed vs. solid edges issuing these modes determine the controllability status of events. This is done independently of the plant's description, and clearly simplifies the design process.

Referring to the specification S_{obj}, we already have shown that it can define the interval of languages the controlled plant's language must belong to. Allowing for any $\mathbf{L}_\nu$ property attains a strictly larger class of control objectives. To prove this claim, we exhibit a modal specification S with an increasing chain of distinct models $\{R_n\}_{n\in\mathbb{N}}$ s.t. none of the intervals $[R_n, R_{n+1}]$ is contained in $mod(S)$; as a result, S cannot be equivalent to any (finite union of) intervals.of languages. Let S be the modal specification of depicted in Figure 1 (page 8). The models of S are the languages over $\{a, b\}$ (c is forbidden) s.t. "any occurrence of a can be extended by a b". Clearly, each language $R_n = \{a^m.b \mid m \leq n\}$ belongs to $mod(S)$. For any pair of natural numbers $k < l$, the prefix-closed language $U_l = R_l \setminus \{a^l.b\}$ belongs to $[R_k, R_l]$, but $U_l \notin mod(S)$. However, there still exists a maximal model of S, namely the one which contains all the R_n's. Hence, we can state the following general theorem:

Theorem 30 *The* **(CCP)** *has a maximally permissive solution for any objective expressed in the logic* $\mathbf{L}_\nu$.

To our knowledge, a proof that for any control objective expressible in $\mathbf{L}_\nu$ the maximal permissive control problem has a solution have never been established so far; this reinforces the interest of using modal specifications and emphasizes the scope of this contribution.

Let us now conclude the section by discussing the computation of the maximal language model of $S^L \cap S_c \cap S_{obj}$: as explained in the previous section, we first need to compute the coherent modal specification S equivalent to $S^L \cap S_c \cap S_{obj}$. If $mod(S)$ is empty, then the control problem has no solution. Otherwise, the maximal solution is $L^S_\top$ (the minimal one is $L^S_\bot$).

Notice that, provided S is given, the computation of $L^S_\top$ is $O(1)$ since it is sufficient to interpret all edges of S as solid ones. Henceforth, computing the maximal solution is carried forward while computing S. This computation given Lemma 2 consist in running the algorithm of [RW89] to remove incoherent states of the modal automaton. To some extent, computing the

coherent modal specification amounts to a generalization of the supremal controllable language's computation.

Example 1 *Consider a coffee machine. The machine can perform 3 actions: insert money (m), give money back (mb), and order a drink (o). The plant is depicted Figure 6: the user must insert some money before ordering a drink or getting her money back. Additionally, it is not possible to control the mb action right after an m action has occurred (the user can always get her money back when she has not ordered yet); this is shown on Figure 7 with the specification $S_{\Sigma_{uc}}$. Now, we want to control the plant so that:*

- *initially, the user can put as much money as she wants;*
- *it is not possible to order twice in a row;*
- *after ordering a drink or getting the money back, it is only possible to insert money.*

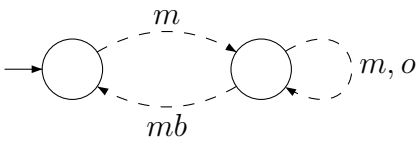
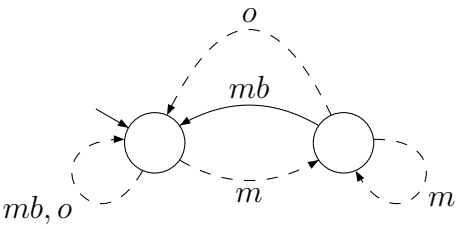


Figure 6: Plant of the coffee machine

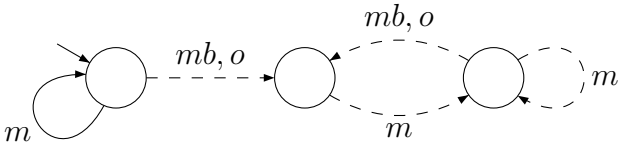
The control objectives translates into the following interval of languages:

$$[\mathtt{A}, \mathtt{E}] = [m^*.\Sigma^*, (\Sigma^* \setminus (\Sigma^*.o.o \cup \Sigma^*.o.mb))]$$

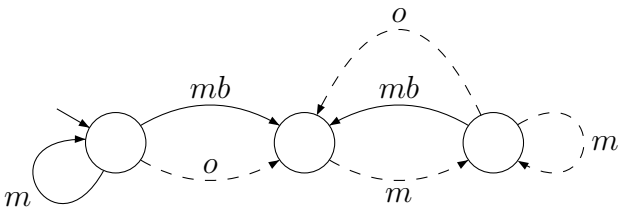
The whole picture is given by Figure 7. Note that the specification $S_{\Sigma_{uc}}$ is similar to the one of Figure 1 (with additional o-transitions, and where m stands for a, and mb stands for b). The argumentation of Subsection 3.4 regarding the expressiveness of this specification remains valid: the controllability constraints we consider here are not expressible by an interval of languages. Finally, we synthesize the maximally permissive controller (by solidifying all the edges of $S_{\Sigma_{uc}} \cap S^{\mathtt{E}}_{\mathtt{A}}$, as it is coherent). The resulting controlled plant is depicted Figure 7.



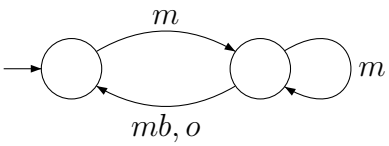
The specification $S_{\Sigma_{uc}}$



The specification $S_{\mathtt{A}}^{\mathtt{E}}$



The specification $S_{\Sigma_{uc}} \cap S_{\mathtt{A}}^{\mathtt{E}}$



The resulting controlled plant

Figure 7: The Coffee Machine Example

4 Conclusion

The contribution proposes a logical framework to specify and solve control problems for reactive systems, when represented by discrete event systems.

The formal framework relies on a syntactic fragment of the powerful mu-calculus logic [Koz83, AN01], namely the Conjunctive Nu-calculus. With this logic, we strictly extend the classical control theory of [RW89] based on regular prefix-closed languages, while preserving the maximal model theorem. Regular prefix-closed languages are acknowledged by the very maximal solution property. However, they lack expressiveness as basically, only (finite unions of) intervals of languages can be considered. Therefore, only a very few liveness features can be addressed. On the other hand, it is well known that enlarging the setting to handle e.g. non-blockingness, leads to much more complicated situations where the maximal model theorem does not hold anymore. As shown here, the Conjunctive Nu-calculus lying in between the regular prefix-closed languages and the full mu-calculus lifts up the maximal model theorem. Moreover, as to avoid the use of formulas, one can equivalently consider modal specifications (finite state automata with "may" kind of transitions). By the results of Section 3, there is a clear method to translates instances of the Centralized Control Problem into modal specifications which models deliver solutions, in an effective manner.

To our knowledge, no investigation has ever been pursued on how far one can go while preserving the maximal model theorem; this is decisive in the control theory of discrete event systems. The present work demonstrates the relevance of the Conjunctive Nu-calculus for the control theory of reactive systems.

References

[AN01] A. Arnold and D. Niwinski. *Rudiments of mu-calculus.* North-Holland, 2001.

[AVW03] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 303(1):7–34, June 2003.

[Bri06] X. Briand. *Sur la décidabilité de certains problèmes de synthèse de contrôleurs*. PhD thesis, Université de Bordeaux I, June 2006.

[CL99] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[GW00] P. Gohari and W. Murray Wonham. On the complexity of supervisory control design in the RW framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 30(5):643–652, 2000.

[HP84] D. Harel and A. Pnueli. On the development of reactive systems. In Krzysztof R. Apt, editor, *Logic and Model of Concurrent Systems*, volume 13 of *NATO ASI*, pages 477–498. Springer-Verlag, October 1984.

[JK05] S. Jiang and R. Kumar. Supervisory control of discrete event systems with ctl* temporal logic specifications. *SIAM Journal on Control and Optimization*, 44(6), January 2005.

[KG95] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, 1995.

[Koz83] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[Lar89] K. G. Larsen. Modal specifications. In *Proc. Workshop Automatic Verification Methods for Finite State Systems, Grenoble, LNCS 407*, pages 232–246. Springer-Verlag, June 1989.

[PR05] S. Pinchinat and S. Riedweg. A decidable class of problems for control under partial observation. *Inf. Process. Lett.*, 95(4):454–460, 2005.

[RP03] S. Riedweg and S. Pinchinat. Quantified mu-calculus for control synthesis. In *Mathematical Foundations of Computer Science*, Bratislava, Slovak Republic, August 2003.

[RP04] S. Riedweg and S. Pinchinat. Maximally permissive controllers in all contexts. In *Workshop on Discrete Event Systems*, Reims, France, sep 2004.

[RP05] J.B. Raclet and S. Pinchinat. The control of non-deterministic systems: a logical approach. In Pavel Piztek, editor, *Proceedings of the 16th IFAC Word Congress*, Prague, Czech Republic, jul 2005. ELSEVIER.

[RW89] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.

[TP03] P. Tabuada and G.J. Pappas. Model checking ltl over controllable linear systems is decidable. In Oded Maler and Amir Pnueli, editors, *HSCC*, volume 2623 of *Lecture Notes in Computer Science*, pages 498–513. Springer, 2003.

[Won89] W. M. Wonham. *On the control of discrete-event systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.