IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 24 May 2022, accepted 3 June 2022, date of publication 15 June 2022, date of current version 27 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183198

Supremal Marker-Controllable Subformula of a Given Canonical Temporal-Safety Formula

KIAM TIAN SEOW, (Senior Member, IEEE)
Robot Intelligence Technology Laboratory, School of Electrical Engineering, KAIST, Daejeon 305-701, South Korea
e-mail: kiamtian@singnet.com.sg

ABSTRACT The existence of marker-progressive supervisory control – about ensuring constant marker progress under specified temporal safety for a class of fair discrete-event systems (DES's) – is a new control problem formulation that has been studied in terms of DES marker-controllability of a linear-time temporal logic (LTL) safety formula given in canonical form. In this paper, provided it exists, the supremal marker-controllable subformula of a given canonical temporal-safety formula for the fair DES model considered is characterized as the weakest fixpoint of some monotone operator Ω. In the case where the DES model is finite state and the complete specification for constant marker progress under temporal safety is a formula of a decidable LTL fragment, it is shown that this fixpoint can be computed as the limit of the (finite) sequence of iterations of computing operator Ω in the syntax of LTL. Marker-progressive control synthesis by fixpoint computation can therefore be made in the same natural-language motivated algebra of LTL as writing the specification, providing the unique opportunity to exploit not only the role of fair events in DES's, but also the human readability of LTL formulas and the associated, syntax-based calculational approach that is transparent; such fixpoint computation is illustrated with four examples. A discussion examines and illuminates the significance of this paper and its potential impact on the logic foundation of supervisory control; it includes making comparisons with related work, and explaining a straightforward generalization of DES marker-controllability that directly extends the proposed fixpoint computation to cover the full specification hierarchy of canonical LTL.

INDEX TERMS Fair discrete-event systems, linear-time temporal logic, supervisory control.

I. INTRODUCTION

The rapid advancement in the Internet, robotics, and artificial intelligence has accelerated the pace of reimagining our living space as one supported by a cyber-physical world of innovative applications. These applications are deployed in ubiquitous electronic devices and robots, offering capabilities of smart service systems that enhance not only the productivity but also the welfare and well-being of humans in everyday life and work. Arguably limited only by human imagination, these applications – in domains such as home and office automation, transportation, and manufacturing – can be modeled and controlled as discrete-event systems (DES's) at some level of design focus using a systems and control design approach. To support this approach, the DES field of supervisory control, founded in the 1980's [1], [2], has been enriched to-date in various ways in the control literature. That these applications are amenable to DES modeling is because a DES is a model of state evolution induced by the abrupt transitional occurrence of various discrete qualitative changes called events [3]. Events are characteristic of an application's core design focus, such as 'lights turned on' and 'window blinds lowered' for a home service robot [4]. The objective of supervisory control theory is to understand and control systems of the discrete-event type; being behaviorally non-continuous in time, such systems cannot generally be modeled and controlled in continuous or discrete time differential-equations.

In the field of supervisory control, this paper continues the study, initiated in [5], of a new supervisory control problem formulation in linear-time temporal logic (LTL) [6]. Introduced in [5], the concept of marker-controllable safety formula in LTL is shown to play a fundamental role in the existence of marker-progressive supervisory control for fair DES's. Marker-progressive control is about ensuring constant marker progress in a DES under specified temporal safety. The fair DES model considered is one whose infinite evolution is directed by event-occurrence conditions governing the subset of system events designated as 'fair'. Over an event space in a rudimentary language,

The associate editor coordinating the review of this manuscript and approving it for publication was Laura Celentano.

the founding DES theory of nonblocking control [1] and its generalization to multitasking [7] are shown in [5] to be conceptually unified, extended, and refined by the LTL theory of marker-progressive control over a state space under DES event fairness [5]. Besides, the founding theory [1], its extensions [3], [8] including [7], and generally the DES control literature to date, are not augmented with greater transparency and structure endowed by the richer setting of fair DES's and canonical LTL, in a uniform framework [6], [9] as adopted in this paper and its predecessor [5]. Under the canonical formula classification [9], two key classes crystallize the notion of marker-progressive control, namely response and safety. The former, in specifying constant progress of markers, is about 'regular completion of tasks' and the latter is about 'no bad occurrences', as respectively expressed in canonical form by the infinite oftenity and invariance of past formulas.

Based on the theoretical foundation laid in [5] (and summarized in Section II), in this paper, provided it exists, the supremal marker-controllable 'subformula' of a given canonical LTL safety formula studied in [5] for the fair DES model considered is shown to be characterized as the weakest fixpoint of a certain monotone operator Ω (see Corollary 1, as developed in Section III). Considering the case where the DES is finite state and the fragment (or sublanguage) of LTL used for safety and marker-response specification is decidable, it is shown that this weakest fixpoint can be computed, in the syntax of LTL, as the limit of some (finite) iteration sequence of Ω (see Theorems 1 and 2, as developed in Section IV). With regard to operator fixpoint and successive iteration for supremal control synthesis, the approach of this paper is developed principally in the same vein as the approach for multitasking control [7], which extends that [2] for nonblocking control [1]. Importantly, it provides the unique opportunity for control synthesis by fixpoint computation to be made in the same natural-language motivated algebra of LTL as writing the specification, exploiting not only the role of fair events in DES's, but also the human readability of LTL formulas and the associated, syntax-based calculational approach that is transparent. In the case considered, together with DES logic modeling, four examples are worked out to some detail, to illustrate the iterative weakest Ω-fixpoint computation, synthesizing the supremal marker-controllable safety formula with syntax-based calculations in LTL over Ω (Section V). A discussion (Section VI) with technically related work and beyond examines and illuminates the significance of this paper and its potential impact on the logic foundation of supervisory control. For a general review of related but different DES control research using temporal logic, refer to [5] for one recent perspective. Finally, a conclusion is presented (Section VII).

II. MARKER-CONTROLLABLE FORMULAS

The theoretical LTL control foundation [5] needed for this paper is summarized in this section.

A. DES MODEL STRUCTURE

Consider the model G of a DES in the form of a basic transition system $(\Pi, Q, \Sigma, \delta, \theta)$. Π denotes the finite state variable set which is typed; the type of each state variable $v \in \Pi$ indicates the domain $Range(v)$ over which the variable ranges. Q denotes the state set, defined by the cross product of the ranges of the variables in Π, i.e., $Q \overset{\text{def}}{=} \bigotimes_{v \in \Pi} Range(v)$, such that every state $q \in Q$ is unique in terms of its assignment of domain values to all state variables in Π. Σ denotes the finite event set with the subset of uncontrollable events Σ_u – events that cannot be disabled by a supervisor; $\Sigma \backslash \Sigma_u$ is the subset of controllable events that can be. $\delta : \Sigma \times Q \to Q$ is a (deterministic) state transition function that is partial. θ is the initial condition – a Boolean valued formula that characterizes the set of initial states $Q_0 \subseteq Q$ of G, such that $q \in Q_0$ provided (the value assignment by) $q \in Q$ satisfies θ. It is assumed that $Q_0 \neq \emptyset$, $\Sigma \neq \emptyset$ due to nontrivial system modeling.

B. LTL AND DES – SYNTAX & SEMANTICS

LTL [6] is a language of predicate logic that is augmented with a temporal operator set to facilitate reasoning over sequences of states. These sequences are producible by DES G along its state trajectories or interpretations. Each interpretation I is a 'labeling' of a string $e(1)e(2)\cdots e(k)\cdots$ generated by G with $e(k) \in \Sigma$, in that $I \overset{\text{def}}{=} q_0 - q_1 - \cdots - q_k \cdots$, where $q_0 \in Q_0$ (an initial state) and for $k \geq 1$, $q_k = \delta(e(k), q_{k-1})$. With $k \geq 0$, the k-prefix of I is $q_0 - q_1 - \cdots - q_k$, and denoted by $I_{(k)}$. A state $q \in Q$ is said to be terminal (in G) if $(\forall \sigma \in \Sigma)(\delta(\sigma, q)$ is not defined). An interpretation I is finite (in length) and said to be terminating if it ends in a state q_k that is terminal, i.e., $I = I_{(k)}$; otherwise, it is infinite and said to be non-terminating, i.e., $I = I_{(\infty)}$. Note that $I_{(0)} = q_0$. Two interpretations or, respectively, their k-prefixes, are defined to be equal (or the same) if the two have the same sequence of states and label the same string.

This paper assumes reader familiarity with LTL [6] with regard to the construction of LTL formulas and the sound LTL proof system (of axioms and theorems) for syntax-based or symbolic reasoning. As reviewed in [5], the formula construction is over a finite set of atomic propositions expressed in terms of state variables in Π of DES G (over their domains) and system transition logics, using temporal operators and Boolean connectives. The system transition logics and temporal operators will be defined later. The symbols used for basic connectives *and*, *and*-ing (or logical product), *not*, and quantifier '*there exists*' are, respectively, $\cdot$ (a dot), $\prod$, $^{-}$ (an overhead bar), and $\exists$. The symbols for derived connectives *or*, *or*-ing (or logical sum), *implies*, *equals*, and quantifier '*for all*' are, respectively, $+$, $\sum$, $\rightarrow$, $=$, and $\forall$. Also included are the propositional constants, namely *validity true* and *inconsistency false*. The symbol for abbreviation or syntactic equality is $\equiv$, to relate formulas that are 'always equal'.

The satisfaction relation $\left(\models^{I^{(k)}} \omega\right) \in \{true, false\}$ (read: 'I at its state q_k satisfies ω', or simply 'I satisfies ω' if $k = 0$, since $I^{(0)} \overset{\text{def}}{=} I$) defines the semantics of an arbitrary LTL formula ω at state q_k ($k \geq 0$) along an arbitrary interpretation I of DES model G. In addition to the standard rules for Boolean connectives, LTL uses satisfaction relation rules for temporal operators to inductively evaluate the satisfaction of an arbitrary $I^{(k)}$ ($k \geq 0$) over an LTL formula. Below, the rules are defined for the basis sets $\{$*always* $\Box$, *next* $\bigcirc$, *until* $\mathcal{U}\}$, $\{$*has-always-been* $\boxminus$, *previously* $\ominus$, *since* $\mathcal{S}\}$ of future and past operators, by which a formula constructed with no future (past) operators is called a past (future) formula, and more specifically called a state formula if it contains no future or past operators. If ω is a state formula, then over $I^{(k)}$ and in state q_k, $\models^{I^{(k)}} \omega$ iff $\models^{q_k} \omega$, with $(\models^{q_k} \omega) \in \{\mathit{true}, \mathit{false}\}$ (read: 'q_k satisfies ω') defining the semantics of the state formula ω in state q_k. The rule for operator $\bigcirc$ requires the following event-transition logic to account for a trajectory I that is finite.

Definition 1 (The σ-Transition Logic): Given $\sigma \in \Sigma$, for an arbitrary state trajectory I of DES G, $I = q_0 - q_1 - \cdots - q_k \cdots$, the function $\tau : \sigma \to (I \to \{\mathit{true}, \mathit{false}\})$ is a system σ-transition logic, defined at $q_k \in Q$ such that

$$\models^{I^{(k)}} \tau_\sigma \text{ iff } (\exists I_{(k+1)})\, q_{k+1} = \delta(\sigma, q_k).$$

Now, given LTL formulas $\omega, \omega_1, \omega_2$:

1) $\models^{I^{(k)}} \Box\omega$ iff for all $j \geq k$, $\models^{I^{(j)}} \omega$.
2) $\models^{I^{(k)}} \bigcirc\omega$ iff $\models^{I^{(k)}} \tau \;\to\; \models^{I^{(k+1)}} \omega$, where $\tau \equiv \sum_{\sigma \in \Sigma} \tau_\sigma$.
3) $\models^{I^{(k)}} \omega_1 \mathcal{U} \omega_2$ iff there is a j ($j \geq k$) such that $\models^{I^{(j)}} \omega_2$ and for all i ($k \leq i < j$), $\models^{I^{(i)}} \omega_1$.
4) $\models^{I^{(k)}} \boxminus\omega$ iff for all j ($0 \leq j \leq k$), $\models^{I^{(j)}} \omega$.
5) $\models^{I^{(k)}} \ominus\omega$ iff $k > 0$ and $\models^{I^{(k-1)}} \omega$.
6) $\models^{I^{(k)}} \omega_1 \mathcal{S} \omega_2$ iff there is a j ($0 \leq j \leq k$) such that $\models^{I^{(j)}} \omega_2$ and for all i ($j < i \leq k$), $\models^{I^{(i)}} \omega_1$.

The derived temporal operators *eventually* $\Diamond$, *unless* $\mathcal{W}$, *once* $\Diamond\!\!\!\!-$, *weak previously* $\tilde{\bigcirc}$, and *back-to* $\mathcal{B}$ are defined by the following abbreviations ($\equiv$): 1) $\Diamond\omega \equiv \overline{\Box(\overline{\omega})} \equiv \mathit{true}\,\mathcal{U}\omega$, 2) $\omega_1 \mathcal{W} \omega_2 \equiv \Box\omega_1 + \omega_1 \mathcal{U} \omega_2$, 3) $\Diamond\!\!\!\!-\,\omega \equiv \overline{\boxminus(\overline{\omega})} \equiv \mathit{true}\,\mathcal{S}\omega$, 4) $\tilde{\bigcirc}\omega \equiv \overline{\ominus(\overline{\omega})}$, and 5) $\omega_1 \mathcal{B} \omega_2 \equiv \boxminus\omega_1 + \omega_1 \mathcal{S} \omega_2$. These abbreviations define and relate useful temporal operators, providing alternative formulas insightful for control. For a clearer exposition, Abbreviation (4) may be presented by the following satisfaction relation:

$$\models^{I^{(k)}} \tilde{\bigcirc}\omega \text{ iff } k = 0 \text{ or } \models^{I^{(k)}} \ominus\omega.$$

Lastly, LTL formulas can be expanded; the following abbreviations are formula expansion rules applicable to past formulas [6, p. 219]: 1) $\boxminus\omega \equiv \omega \cdot \tilde{\bigcirc}(\boxminus\omega)$, and 2) $\omega_1 \mathcal{S} \omega_2 \equiv \omega_2 + \omega_1 \cdot \ominus(\omega_1 \mathcal{S} \omega_2)$.

To model DES transitional behavior in LTL formulas more compactly, system dynamic event-operators and event-transition operators are used. Below, the former operators are defined in terms of either the σ-transition logic in Definition 1 or the following logic; in turn, each latter operator is defined in terms of a former.

Definition 2 (The Conditioned σ-Transition Logic): Given an arbitrary LTL formula ψ over DES G and $\sigma \in \Sigma$, for an arbitrary state trajectory I of G, $I = q_0 - q_1 - \cdots - q_k \cdots$, and an arbitrary $I' \in \mathcal{I}(G)$, $I' = I_{(k)} - q'_{k+1} \cdots$ (if it exists), the function $\tau_x : (\sigma, \psi) \to (I \to \{\mathit{true}, \mathit{false}\})$ is a system ψ-conditioned transition logic, defined at $q_k \in Q$ such that

$$\models^{I^{(k)}} \left(\tau_{x|\sigma}(\psi) = \tau_\sigma \cdot (\forall I', I'_{(k+1)} \neq I_{(k+1)}) \models^{I'^{(k)}} \bigcirc\overline{\psi} \right).$$

The logic $\tau_{x|\sigma}(\psi)$ may be called the transition of event σ in the next ψ-barred neighborhood.

Then given arbitrary LTL formulas ψ, φ over DES G and $\sigma \in \Sigma$, the system dynamic event-operators $\ominus_\sigma$, $\bigcirc_\sigma$, $\ominus_{x|\sigma}(.,.)$ over an arbitrary state trajectory I of G, $I = q_0 - q_1 - \cdots - q_k \cdots$, are defined as follows:

1) $\models^{I^{(k)}} \ominus_\sigma(\varphi) = \ominus(\tau_\sigma \cdot \varphi)$.
2) $\models^{I^{(k)}} \bigcirc_\sigma(\varphi) = (\tau_\sigma \to \bigcirc\varphi)$.
3) $\models^{I^{(k)}} \ominus_{x|\sigma}(\psi, \varphi) = \ominus\left(\tau_{x|\sigma}(\psi) \cdot \varphi\right)$.

The respective system uncontrollable and conditioned event-transitions τ_u, $\tau_x(.)$ are characterized as follows:

1) $\tau_u \equiv \sum_{\sigma \in \Sigma_u} \tau_\sigma$. 2) $\tau_x(.) \equiv \sum_{\sigma \in \Sigma} \tau_{x|\sigma}(.)$.

The system dynamic event-transition operators $\ominus_u$, $\bigcirc_u$, $\ominus_x(.,.)$ are characterized as follows:

1) $\ominus_u \equiv \sum_{\sigma \in \Sigma_u} \ominus_\sigma$. 2) $\bigcirc_u \equiv \prod_{\sigma \in \Sigma_u} \bigcirc_\sigma$.
3) $\ominus_x(.,.) \equiv \sum_{\sigma \in \Sigma} \ominus_{x|\sigma}(.,.)$.

Let $\mathcal{T}$ be a unary temporal operator. Then $\mathcal{T}^n$, for $n \geq 0$, is defined over an arbitrary formula ω as follows:

$$\mathcal{T}^n(\omega) \equiv \overbrace{\mathcal{T}(\mathcal{T}(\mathcal{T}(\cdots\ \mathcal{T}}^{n\ \mathit{times}}(\omega)\ \cdots))).$$

Note that $\mathcal{T}^0$ is an identity operator, i.e., $\mathcal{T}^0(\omega) \equiv \omega$.

The model operational premise is this: From every non-terminal state that DES G is in, one event will occur and transition the DES into another state.

Only interpretations or state trajectories that refer to the actual behavior of DES G are of interest; these are legal and constitute the legal set $\mathcal{I}(G)$, on which the notion of G-validity of an LTL formula ω, denoted by $G \models \omega$, is defined:

$$G \models \omega \text{ iff } (\forall I \in \mathcal{I}(G)) \models^{I} \omega.$$

In LTL semantics, for an arbitrary set $\mathcal{I}(G)$, $\omega_1 \equiv \omega_2$ denotes $G \models \Box(\omega_1 = \omega_2)$. Define *always-implies* $\Rightarrow$ such that $\omega_1 \Rightarrow \omega_2$ denotes $G \models \Box(\omega_1 \to \omega_2)$; therefore $(\omega_1 \equiv \omega_2) \equiv (\omega_1 \Rightarrow \omega_2) \cdot (\omega_2 \Rightarrow \omega_1)$. In addition, let $\omega_1 \approx \omega_2$, $\omega_1 \rightsquigarrow \omega_2$ denote $G \models (\omega_1 = \omega_2)$, $G \models (\omega_1 \to \omega_2)$, respectively, where the connectives $\approx$, $\rightsquigarrow$ are said to be the anchored versions of $\equiv$, $\Rightarrow$, respectively.

C. FAIR DES MODEL

Let $\Sigma_{\mathcal{F}} = \Sigma_{\mathcal{C}} \cup \Sigma_{\mathcal{J}}$ denote the set of fair events, where $\Sigma_{\mathcal{C}}$ denotes the strongly fair set of compassionate events, and $\Sigma_{\mathcal{J}}$ denotes the weakly fair set of just events.

Definition 3 (The σ-Definition Logic): Given $\sigma \in \Sigma$, for an arbitrary state $q \in Q$ of DES G, the function $\xi : \sigma \to (q \to \{\mathit{true}, \mathit{false}\})$ is a system σ-definition logic, defined such that

$$\models^{q} \xi_\sigma \text{ iff } (\exists q' \in Q) q' = \delta(\sigma, q).$$

Then the DES model G considered is said to be fair [6, p. 256] (with respect to $\Sigma_{\mathcal{F}} \subseteq \Sigma_u$), where $\Sigma_{\mathcal{F}} = \Sigma_{\mathcal{C}} \cup \Sigma_{\mathcal{J}}$ such that, for every state trajectory I of G, $I \in \mathcal{I}(G)$ iff I satisfies the event-fairness formulas:

1) $(\forall \sigma \in \Sigma_{\mathcal{C}}) \models^{I} \square\Diamond\xi_\sigma \rightarrow \square\Diamond\tau_\sigma$. (Strong fairness)
2) $(\forall \sigma \in \Sigma_{\mathcal{J}}) \models^{I} \Diamond\square\xi_\sigma \rightarrow \square\Diamond\tau_\sigma$. (Weak fairness)

The characterization above may assume that $\Sigma_{\mathcal{C}} \cap \Sigma_{\mathcal{J}} = \emptyset$ without loss of generality. The event-fairness formulas constitute the legal conditions that model the set $\mathcal{I}(G)$.

D. CONTROL OF FAIR DES's

In supervisory control, the trajectory set of interest for fair DES G is $\mathcal{I}^{\circledast}(G)$, given by $\mathcal{I}^{\circledast}(G) \ = \mathcal{I}(G) \cup \mathcal{I}^{\circledast}(G)$, where

$$\mathcal{I}^{\circledast}(G) = \{I_{(k)} \mid I \in \mathcal{I}(G),\ \text{finite } k \geq 0,\ \text{and } I_{(k)} \notin \mathcal{I}(G)\}$$

is the legally prefix-admissible set; $\mathcal{I}(G) \cap \mathcal{I}^{\circledast}(G) = \emptyset$.

An LTL formula φ is an invariant if $\varphi \equiv \boxminus\psi$, where ψ is some past formula; and this ψ is called the kernel of φ if it has no operator $\boxminus$ in its outermost scope. The DES theory of supervisory control centers around the invariant and its kernel.

Bring in the specification pair $(P, \mathcal{M})$ over DES G to denote

$$\square\left(P \cdot \prod_{i=1}^{m} \Diamond M_i\right),$$

where P is the kernel of some arbitrary invariant, and $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$ is the system marker set, where each $M_i \in \mathcal{M}$ $(1 \leq i \leq m)$ is an arbitrary past formula specifying a system marker condition. In their respective forms [6], [9], $\square P$ is a canonical LTL safety formula and $\square\Diamond M_i$ is a canonical LTL response formula. An arbitrary invariant φ over DES G is said to be (with respect to G): 1) P-history bounded if $G \models \square(\varphi \rightarrow \boxminus P)$, 2) initially satisfied if $G \models \varphi$ (or, equivalently, $G \models \square(\text{ini} \rightarrow \varphi)$, where $\text{ini} \equiv \widetilde{\ominus}\mathit{false}$), 3) Σ_u-invariant if $G \models \square\big(\ominus_u(\varphi) \rightarrow \varphi\big)$, 4) $(\mathcal{M}, \varphi)$-condition invariant if

$$G \models \square\left(\ominus_x\left(\varphi, \varphi \cdot \sum_{i=1}^{m} \overline{M_i}\right) \rightarrow \varphi\right),$$

and 5) (marker- or) $\mathcal{M}$-alive under conditional invariance if

$$G \models \square\varphi \rightarrow \square\left(\prod_{i=1}^{m} \Diamond M_i\right).$$

It then follows that $\square P$ is said to be (with respect to G): 1) controllable if $\boxminus P$ is initially satisfied and Σ_u-invariant, 2) $\mathcal{M}$-directing if $\boxminus P$ is initially satisfied, $(\mathcal{M}, \boxminus P)$-condition invariant, and $\mathcal{M}$-alive under conditional invariance, and 3) $\mathcal{M}$-controllable if $\square P$ is controllable and $\mathcal{M}$-directing.

For the specification pair $(P, \mathcal{M})$, the set of all $\mathcal{M}$-controllable temporal-safety formulas whose invariants are not weaker than $\boxminus P$ is introduced:

$$\mathcal{C}(P, \mathcal{M}) = \left\{ \square\psi \;\middle|\; \begin{array}{l} \square\psi \text{ is } \mathcal{M}\text{-controllable, where} \\ \psi \text{ is the kernel of an invariant} \\ \text{that is } P\text{-history bounded} \end{array} \right\}.$$

If $\mathcal{M} = \emptyset$, then

$$\mathcal{C}(P, \emptyset) = \left\{ \square\psi \;\middle|\; \begin{array}{l} \square\psi \text{ is controllable, where} \\ \psi \text{ is the kernel of an invariant} \\ \text{that is } P\text{-history bounded} \end{array} \right\}.$$

In this case, let $\mathcal{C}(P) \overset{\text{def}}{=} \mathcal{C}(P, \emptyset)$.

Proposition 1: Consider the kernel P of an arbitrary invariant over fair DES G with system marker set $\mathcal{M}$, and assume $\mathcal{C}(P, \mathcal{M}) \neq \emptyset$. Then $\mathcal{C}(P, \mathcal{M})$ is closed under arbitrary *or*-ings. Specifically, $\mathcal{C}(P, \mathcal{M})$ contains a (unique) supremal element (which is hereby denoted by $\sup \mathcal{C}\,(P, \mathcal{M})$).

Proof: See [5]. ■

Note that, in logic terms, $\sup \mathcal{C}(P, \mathcal{M}) \approx \mathit{false}$ provided $\mathcal{C}(P, \mathcal{M}) = \emptyset$. Thus, in general, $\sup \mathcal{C}(P, \mathcal{M}) \in \mathcal{C}(P, \mathcal{M}) \cup \{\mathit{false}\}$. Provided $\mathcal{C}(P, \mathcal{M}) \neq \emptyset$, $\sup \mathcal{C}(P, \mathcal{M})$ is the supremal or weakest $\mathcal{M}$-controllable subformula of $\square P$.

Based on the foregoing technical summary, the synthesis of $\sup \mathcal{C}(P, \mathcal{M})$ as the weakest fixpoint of a monotone operator may now be formulated and investigated.

III. PRELIMINARIES

A. A RELATION $\overset{G}{\simeq}$ DELIMITING PROGRESS IN SAFETY

A relation $\overset{G}{\simeq}$ is first defined: Given two arbitrary LTL formulas ψ, χ over fair DES G,

$\square\psi \overset{G}{\simeq} \chi$ (read: '$\square\psi$ exists that delimits χ in G') iff:
ψ is the kernel of some invariant, and:

$$\begin{cases} (\forall I \in \mathcal{I}(G))(\forall k \geq 0) \\ \models^{I} (\square\psi = \chi), \models^{I_{(k)}} (\chi \rightarrow \square\psi), \text{ and} \\ \models^{I_{(k)}} \square\psi \rightarrow (\exists I' \in \mathcal{I}(G)) \models^{I'} \square\psi \text{ or} \models^{I'_{(j)}} \chi, \\ \qquad\qquad \text{for } I'_{(k)} = I_{(k)} \text{ and some } j \geq k. \end{cases}$$

Intuitively, where χ is an LTL progress formula, the relation $\overset{G}{\simeq}$ means an LTL safety formula $\square\psi$ exists that bounds exactly the progress specified by χ in DES G, in that, every state trajectory of $\mathcal{I}^{\circledast}(G)$ satisfying χ, satisfies $\square\psi$, and each prefix of an arbitrary state trajectory of $\mathcal{I}(G)$ satisfying $\square\psi$ either can be extended to or is a state trajectory of $\mathcal{I}^{\circledast}(G)$ satisfying χ. The LTL formula $\square\psi$ is said to be the exact 'delimiting safety-closure' or 'prefixing' formula for χ.

The relation $\overset{G}{\simeq}$ above is closely related to the notion of topological closure of an LTL formula χ studied in [10], [11]. While the latter notion captures the strongest safety formula that is not stronger than χ, the former relation captures a safety formula that exists as the exact delimitation of χ, which is of control interest in this paper. Besides, this relation admits finite state trajectories possibly present in DES G, and prefix state trajectories that may result due to control. The ensuing results studied are believed to be quite new and of theoretical interest in the LTL context of supervisory control.

Proposition 2: If $\Box\psi \overset{G}{\simeq} \chi$, then $\Box\psi \approx \chi$.

Proof: By definitions of $\overset{G}{\simeq}$, G-validity, and $\approx$. ■

Proposition 3: Consider the kernels ψ, P of two arbitrary invariants and an arbitrary LTL formula Y over fair DES G. If $\Box\psi \overset{G}{\simeq} Y \cdot \Box P$, then:

P3.1) $\Box\psi \overset{G}{\simeq} Y \cdot \Box\psi$, and P3.2) $\Box\psi \approx Y \cdot \Box P \approx Y \cdot \Box\psi$.

Proof: Consequent P3.1 follows from the definition of $\overset{G}{\simeq}$ and the following reasoning: Consider an arbitrary $I \in \mathcal{I}(G)$ and an arbitrary index $k \geq 0$. Since $Y \cdot \Box P \equiv Y \cdot Y \cdot \Box P$, it follows that:

$$\begin{cases} \models^{I} (\Box\psi = Y \cdot \Box P), \models^{I_{(k)}} (Y \cdot \Box P \rightarrow \Box\psi), \text{ and} \\ \models^{I_{(k)}} \Box\psi \rightarrow (\exists I' \in \mathcal{I}(G)) \models^{I'} \Box\psi \text{ or } \models^{I'_{(j)}} Y \cdot \Box P, \\ \qquad \text{for } I'_{(k)} = I_{(k)} \text{ and some } j \geq k. \end{cases}$$

implies

$$\begin{cases} \models^{I} (\Box\psi = Y \cdot \Box\psi), \models^{I_{(k)}} (Y \cdot \Box P \rightarrow \Box\psi), \text{ and} \\ \models^{I_{(k)}} \Box\psi \rightarrow (\exists I' \in \mathcal{I}(G)) \models^{I'} \Box\psi \text{ or } \models^{I'_{(j)}} Y \cdot \Box\psi, \\ \qquad \text{for } I'_{(k)} = I_{(k)} \text{ and some } j \geq k. \end{cases}$$

implies

$$\begin{cases} \models^{I} (\Box\psi = Y \cdot \Box\psi), \models^{I_{(k)}} (Y \cdot \Box\psi \rightarrow \Box\psi), \text{ and} \\ \models^{I_{(k)}} \Box\psi \rightarrow (\exists I' \in \mathcal{I}(G)) \models^{I'} \Box\psi \text{ or } \models^{I'_{(j)}} Y \cdot \Box\psi, \\ \qquad \text{for } I'_{(k)} = I_{(k)} \text{ and some } j \geq k. \end{cases}$$

Consequent P3.2 follows from Proposition 2 and Consequent P3.1. ■

Proposition 4: Consider the kernels ψ, P of two arbitrary invariants over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. If

$$\Box\psi \overset{G}{\simeq} \Box\left(P \cdot \prod_{i=1}^{m} \Diamond M_i\right),$$

then $\boxminus\psi$ is: 1) $(\mathcal{M}, \boxminus\psi)$-condition invariant, 2) $\mathcal{M}$-alive under conditional invariance, and 3) P-history bounded.

Proof: Suppose

$$\Box\psi \overset{G}{\simeq} \Box\left(P \cdot \prod_{i=1}^{m} \Diamond M_i\right),$$

where ψ, P are the kernels of two arbitrary invariants over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. The proof then proceeds as follows.

1) To prove that $\boxminus\psi$ *is* $(\mathcal{M}, \boxminus\psi)$*-condition invariant*: By Proposition 3: P3.1,

$$\Box\psi \overset{G}{\simeq} \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right).$$

By contradiction, assume that $\boxminus\psi$ is not $(\mathcal{M}, \boxminus\psi)$-condition invariant with respect to DES G, i.e., there exists a state trajectory $I \in \mathcal{I}(G)$ such that

$$\models^{I} \Diamond\left(\ominus_x\left(\psi, \boxminus\psi \cdot \sum_{i=1}^{m} \overline{M_i}\right) \cdot \overline{\psi}\right).$$

This implies that there is a $k \geq 0$ such that $\models^{I_{(k)}} \Box\psi$ and $\models^{I^{(k)}} \sum_{i=1}^{m} \overline{M_i}$; and so where I'' is I' or $I'_{(j)}$, for $I' \in \mathcal{I}(G)$, $I'_{(k)} = I_{(k)}$, and some $j \geq k$, it follows that for all such I', $\models^{I'^{(k+1)}} \overline{\psi}$, and for all I'',

$$\begin{cases} \models^{I''} \overline{\Box\psi}, & \text{if } I'' = I' \\ \models^{I''} \overline{\Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right)}, & \text{if } I'' = I'_{(j)}. \end{cases}$$

It follows that the relation $\Box\psi \overset{G}{\simeq} \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right)$ is contradicted.

2) To prove that $\boxminus\psi$ *is* $\mathcal{M}$*-alive under conditional invariance*: Note that the result is implied by the relation $\overset{G}{\simeq}$.

3) To prove that $\boxminus\psi$ *is* P*-history bounded*: Note that, as implied by the relation $\overset{G}{\simeq}$, $(\forall I \in \mathcal{I}(G))(\forall k \geq 0) \models^{I_{(k)}} \Box\psi \rightarrow (\exists I'') \models^{I''} \Box\left(P \cdot \prod_{i=1}^{m} \Diamond M_i\right)$, where I'' is I' or $I'_{(j)}$, for $I' \in \mathcal{I}(G)$, $I'_{(k)} = I_{(k)}$, and some $j \geq k$. Thus $(\forall I \in \mathcal{I}(G))$ $(\forall k \geq 0) \models^{I^{(k)}} (\boxminus\psi \rightarrow \boxminus P)$, and the result follows. ■

Proposition 5: Consider the kernel ψ of an arbitrary invariant over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. Then $\Box\psi \overset{G}{\simeq} \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right)$ iff $\boxminus\psi$ is: 1) $(\mathcal{M}, \boxminus\psi)$-condition invariant, and 2) $\mathcal{M}$-alive under conditional invariance.

Proof: Consider the kernel ψ of an arbitrary invariant over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. The proof then proceeds as follows.

(If) Because $\boxminus\psi$ is $(\mathcal{M}, \boxminus\psi)$-condition invariant, it follows that, for each $I \in \mathcal{I}(G)$ and each $k \geq 0$, such that $\models^{I^{(k)}} \boxminus\psi$ or, equivalently, $\models^{I_{(k)}} \Box\psi$, there exists an $I' \in \mathcal{I}(G)$, $I'_{(k)} = I_{(k)}$, such that

either: $\models^{I'} \Box\psi$,

or: for some $j \geq k$, $\models^{I'_{(j)}} \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right)$.

Since $\boxminus\psi$ is also $\mathcal{M}$-alive under conditional invariance, it follows that

$$(\forall I \in \mathcal{I}(G)) \models^{I} \left(\Box\psi \rightarrow \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right)\right).$$

Since $\models^{I''} \left(\Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right) \rightarrow \Box\psi\right)$ for an arbitrary I'' that is $I \in \mathcal{I}(G)$ or its prefix $I_{(k)}$ $(k \geq 0)$, the result follows.

(Only if) The result follows by Proposition 4 (with $P \equiv \psi$). ■

Proposition 6: Consider the kernel ψ of an arbitrary invariant over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. Then $\Box\psi$ is $\mathcal{M}$-directing iff ψ is initially satisfied and

$$\Box\psi \overset{G}{\simeq} \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right).$$

Proof: The result follows by logical reasoning when applying Proposition 5 and the fact $\boxminus\psi \approx \psi$ to the definition of $\mathcal{M}$-directingness. ■

Proposition 7: Consider the kernel ψ of an arbitrary invariant over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. Then $\Box\psi$ is $\mathcal{M}$-controllable iff $\Box\psi$ is controllable and

$$\Box\psi \overset{G}{\simeq} \Box\left(\psi \cdot \prod_{i=1}^{m} \Diamond M_i\right).$$

Proof: It is a definitional fact that the controllability of $\Box\psi$ implies ψ is initially satisfied. The result then follows by logical reasoning when applying Proposition 6 and this fact to the definition of $\mathcal{M}$-controllability. ■

Proposition 8: Consider the kernels ψ_1, ψ_2 of two arbitrary invariants over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. If, for all i $(1 \leq i \leq 2)$, $\Box\psi_i \overset{G}{\simeq} \Box\left(\psi_i \cdot \prod_{i=1}^{m} \Diamond M_i\right)$, then

$$\Box(\boxminus\psi_1 + \boxminus\psi_2) \overset{G}{\simeq} \Box\left((\boxminus\psi_1 + \boxminus\psi_2) \cdot \prod_{i=1}^{m} \Diamond M_i\right).$$

Proof: The result follows by logical reasoning over the definition of $\overset{G}{\simeq}$. ■

B. AN OPERATOR Ω CHARACTERIZING sup $\mathcal{C}(P, \mathcal{M})$

Two modularity results of interest are first presented.

Proposition 9: Consider the kernels ψ_1, ψ_2 of two arbitrary invariants over fair DES G. If, for all i $(1 \leq i \leq 2)$, $\Box\psi_i$ is controllable, then $(\Box\psi_1 + \Box\psi_2)$ is controllable.

Proof: By logical reasoning over the constituents in the controllability definition of $\Box\psi_i$ $(1 \leq i \leq 2)$, it can be shown that $\Box(\boxminus\psi_1 + \boxminus\psi_2)$ is controllable. The result then follows by the fact that $\Box(\boxminus\psi_1 + \boxminus\psi_2) \approx (\Box\psi_1 + \Box\psi_2)$. ■

Proposition 10: Consider the kernels ψ_1, ψ_2 of two arbitrary invariants over fair DES G with system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. If, for all i $(1 \leq i \leq 2)$, $\Box\psi_i$ is $\mathcal{M}$-directing, then $(\Box\psi_1 + \Box\psi_2)$ is $\mathcal{M}$-directing.

Proof: Given that $\Box\psi_i$ $(1 \leq i \leq 2)$ is $\mathcal{M}$-directing. By logical reasoning when applying Propositions 6 and 8, it can be shown that $\Box(\boxminus\psi_1 + \boxminus\psi_2)$ is $\mathcal{M}$-directing. The result then follows by the fact that $\Box(\boxminus\psi_1 + \boxminus\psi_2) \approx (\Box\psi_1 + \Box\psi_2)$. ■

By mathematical induction, the results in Propositions 9 and 10 can be extended to more than two invariants.

Now, consider the specification pair $(P, \mathcal{M})$, where system marker set $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. Let $Y_m \equiv \Box\left(\prod_{i=1}^{m} \Diamond M_i\right)$ and $\mathcal{R}$ be the set of LTL formulas, each of the form $\left(Y_m \cdot \Box P'\right)$, where P' is the kernel of some invariant that is P-history bounded. LTL formulas in this form are temporal-response formulas (under $\approx$) [6].

Define the operator $\Omega : \mathcal{R} \to \mathcal{R}$ according to

$$\Omega\left(Y_m \cdot \Box P'\right) \approx Y_m \cdot \sup \mathcal{C}(\psi), \text{ where}$$
$$\Box\psi \overset{G}{\simeq} \left(Y_m \cdot \Box P'\right). \tag{1}$$

Operator Ω is well defined in that, inferring from Proposition 1 that is proved in [5], the set $\mathcal{C}(.) \cup \{false\}$ is closed under arbitrary *or*-ings. Note that, based on the definition of $\mathcal{M}$-controllability, the proof of Proposition 1 may also, in essence, follow from the result extensions of Propositions 9 and 10.

Proposition 11: Consider the kernel ψ of an arbitrary invariant over fair DES G with system marker set $\mathcal{M}$. Then $\Box\psi \in \mathcal{C}(P, \mathcal{M}) \cup \{false\}$ iff $\Omega(\Box\psi) \approx \Box\psi$, where $\Box\psi \overset{G}{\simeq} Y_m \cdot \Box\psi$.

Proof: If $\mathcal{C}(P, \mathcal{M}) = \emptyset$, the result is trivially true. If $\mathcal{C}(P, \mathcal{M}) \neq \emptyset$, then to prove that the result is also true, first show that given the kernel ψ of an arbitrary invariant such that $\Box\psi$ is not *false* over DES G with system marker set $\mathcal{M}$, $\Box\psi$ is $\mathcal{M}$-controllable and $\boxminus\psi$ is P-history bounded iff

$$\Omega\left(Y_m \cdot \Box\psi\right) \approx Y_m \cdot \Box\psi,$$

where $\Box\psi \overset{G}{\simeq} Y_m \cdot \Box\psi$. This follows by Proposition 7, Ω (1), and the fact that since $\Box\psi$ is not *false*, $\Box\psi$ is controllable provided $\sup \mathcal{C}(\psi) \approx \Box\psi$. It then follows by Proposition 2 that $\Box\psi \approx Y_m \cdot \Box\psi$, thereby completing the proof. ■

Proposition 11 characterizes $\mathcal{C}(P, \mathcal{M}) \cup \{false\}$ as the set of fixpoints of Ω. Since, for every $\Box\psi \in \mathcal{C}(P, \mathcal{M}) \cup \{false\}$, $\Box\psi \rightsquigarrow \sup \mathcal{C}(P, \mathcal{M})$, the following corollary is immediate.

Corollary 1: $\sup \mathcal{C}(P, \mathcal{M})$ is the weakest fixpoint of Ω.

IV. FIXPOINT COMPUTATION OF sup $\mathcal{C}(P, \mathcal{M})$

A. ITERATION OF Ω & COMPUTING sup $\mathcal{C}(P, \mathcal{M})$

In view of Corollary 1, consider computing $\sup \mathcal{C}(P, \mathcal{M})$ by iteration of Ω along the following sequence of formulas:

$$\begin{cases} K_0 \approx Y_m \cdot \Box P, \\ K_{j+1} \approx \Omega(K_j), \quad j = 0, 1, \cdots \\ \qquad \approx Y_m \cdot \sup \mathcal{C}(\psi_j), \text{ where } \Box\psi_j \overset{G}{\simeq} K_j. \end{cases} \tag{2}$$

Proposition 12: Given the sequence $\{K_j\}$ (2) for the specification pair $(P, \mathcal{M})$, the (logic-theoretic) limit $K \approx \lim_{j \to \infty} K_j$ exists such that $\sup \mathcal{C}(P, \mathcal{M}) \rightsquigarrow K$.

Proof: Let $S \approx \sup \mathcal{C}(P, \mathcal{M})$, $L \approx Y_m \cdot \Box P$.

It is clear that Ω is monotone, i.e., for $X, Y \in \mathcal{R}$:
If $X \rightsquigarrow Y$, then $\Omega(X) \rightsquigarrow \Omega(Y)$.
Now, the following is true:
Base case: $K_1 \approx \Omega(K_0) \rightsquigarrow L \approx K_0$.
Induction case (for $j \geq 1$, established by monotone Ω):
If $K_j \rightsquigarrow K_{j-1}$, then $K_{j+1} \approx \Omega(K_j) \rightsquigarrow \Omega(K_{j-1}) \approx K_j$.
It follows that $K_{j+1} \rightsquigarrow K_j$ for $j = 0, 1, \cdots$.

Thus $K \approx \lim_{j \to \infty} K_j \approx \prod_{j=0}^{\infty} K_j$ exists.

Now, the following is also true:

Base case: $S \approx \Omega(S) \rightsquigarrow L = K_0$.

Induction case (for $j \geq 1$, established by monotone Ω):

If $S \rightsquigarrow K_j$, then $S \approx \Omega(S) \rightsquigarrow \Omega(K_j) \approx K_{j+1}$.

It follows that $S \rightsquigarrow K_j$ for $j = 0, 1, \cdots$.

Hence, $S \rightsquigarrow K$. ■

The sequence $\{K_j\}$ (2) is said to converge if there exists an index $i \geq 0$ such that $(\forall j \geq i) K_j \approx K_i$. Proposition 13 states the necessary and sufficient condition for its convergence.

Proposition 13: Under the sequence $\{K_j\}$ (2) for the specification pair $(P, \mathcal{M})$,

$$(\forall j \geq i) K_j \approx K_i \text{ iff } \Omega(K_i) \approx K_i, \quad \text{where } i \geq 0.$$

Proof: Under $\{K_j\}$ (2) for $(P, \mathcal{M})$, the proof proceeds as follows:

(Only If) $(\forall j \geq i) K_j \approx K_i$ implies $K_{i+1} \approx K_i$, where $i \geq 0$. Since $K_{i+1} \approx \Omega(K_i)$, it follows that $\Omega(K_i) \approx K_i$.

(If) Trivially, $K_j \approx K_i$ for $j = i$. Given that $\Omega(K_i) \approx K_i$, it remains to show by mathematical induction that, $(\forall j \geq i) K_{j+1} \approx K_i$, as follows:

Base case ($j = i$): $K_{i+1} \approx \Omega(K_i) \approx K_i$.

Induction case: Assume for $j = k$, $K_{(k+1)} \approx K_i$. Then for $j = k+1$, $K_{(k+1)+1} \approx \Omega(K_{(k+1)}) \approx \Omega(K_i) \approx K_i$. ■

Proposition 14: Consider the sequence $\{K_j\}$ (2) for the specification pair $(P, \mathcal{M})$. If at iteration $i+1$ $(i \geq 0)$, $\Omega(K_i) \approx K_i$ where $\Box\phi \overset{G}{\simeq} K_i$, then $\sup\mathcal{C}(P, \mathcal{M}) \approx \Box\phi$.

Proof: Suppose at iteration $i+1$ $(i \geq 0)$, $\Omega(K_i) \approx K_i$, where $\Box\phi \overset{G}{\simeq} K_i$. Then $K_i \approx Y_m \cdot \Box\phi$ and by Proposition 2, $\Box\phi \approx Y_m \cdot \Box\phi$. By Proposition 13 and from the proof of Proposition 12 that $K_{j+1} \rightsquigarrow K_j$ for all $j \geq 0$, the limit $K \approx K_i$. Thus by Proposition 12, $\sup\mathcal{C}(P, \mathcal{M}) \rightsquigarrow \Box\phi$. Together with the fact that $K_i \approx \Box\phi$, it follows by Proposition 11 that $\Box\phi \in \mathcal{C}(P, \mathcal{M}) \cup \{\mathit{false}\}$. Therefore, $\Box\phi \rightsquigarrow \sup\mathcal{C}(P, \mathcal{M})$, and combining, $\sup\mathcal{C}(P, \mathcal{M}) \approx \Box\phi$. Hence the result. ■

Inferring from Proposition 13 on Proposition 14, the desired result is obtained if Ω (1) can be iteratively computed to convergence along the sequence $\{K_j\}$ (2).

To iterate Ω along $\{K_j\}$ for computing the desired LTL formula $\sup\mathcal{C}(P, \mathcal{M})$, the key question of this paper may now be posed:

Under what condition(s) is $\{K_j\}$ (2) (syntactically) convergent to $\sup\mathcal{C}(P, \mathcal{M})$?

This $\{K_j\}$-convergence question is studied in the case of finite state DES G, where all variables in Π range over finite domains [12].

B. SYNTACTIC COMPUTABILITY OF Ω

To answer the $\{K_j\}$-convergence question requires a syntactic computability result for Ω (1) in $\{K_j\}$ (2).

Some facts about the decidability of LTL are first stated. An LTL fragment of interest and its formulas are said to be decidable if, over DES G, the satisfaction relation and hence G-validity of these formulas are provable to be *true* or *false*. Propositional and monodic LTL refer to LTL's respective propositional and monodic (predicate) fragments. The propositional fragment of LTL restricts formulas to syntactic constructions from Boolean variables and propositions (with a unique truth value – *true* or *false*), with no quantifier added. The monodic fragment [13] of LTL restricts its temporal part to formulas of the following construction: Every constituent of a monodic formula that is a formula of the form $\mathcal{T}_1\psi$ or $\psi_1\mathcal{T}_2\psi_2$, where $\mathcal{T}_1, \mathcal{T}_2$ are temporal operators, has at most one free variable; a free variable x in a constituent being one that occurs at least once in the constituent without being introduced by a quantification $(\exists x)$ or $(\forall x)$. Roughly speaking, a monodic subfragment is decidable if its non-temporal part is restricted to some decidable fragment of predicate logic. Propositional LTL is decidable [12], [14], and so are various subfragments of the monodic fragment [13], [15] of LTL, each subsuming and is therefore more expressive than propositional LTL.

In the case of finite state (or finite domain) DES G, despite the claim of adequacy inferred from [14] that modeling the DES and specification can be expressly made in propositional LTL [14], such modeling may, in principle, also be made in one decidable subfragment of monodic LTL.

At this juncture, it may well be noted that, because the states of DES G are defined as unique, every state $q \in Q$ can be simply characterized by the proposition

$$p_q \equiv \prod_{v_i \in \Pi} (v_i = a_i) \text{ for some } a_i \in \mathit{Range}(v_i)$$

– a state formula which is a decidable predicate such that $\models^{q} p_q$ and $\left(\forall q' \in Q \backslash \{q\}\right) \models^{q'} \overline{p_q}$.

In view of the various known decidable LTL fragments that the specification pair $(P, \mathcal{M})$ can be expressed in, the syntactic computability result of Theorem 1 that follows immediately after Lemma 1 below is quite general for the case of finite state DES's.

Lemma 1: Consider the case that DES G is finite state. Then for an arbitrary decidable LTL formula χ over G, there exists a decidable ψ which is the kernel of some invariant, such that $\Box\psi \overset{G}{\simeq} \chi$.

Proof: A transition system model G' exists for an arbitrary decidable LTL formula χ over DES G, such that

$$\mathcal{I}(G') = \left\{I \in \mathcal{I}(G) \mid \models^{I} \chi\right\} \cup \left\{I_{(j)} \mid I \in \mathcal{I}(G) \;\&\; \models^{I_{(j)}} \chi\right\}.$$

Clearly, $\mathcal{I}(G') \subseteq \mathcal{I}^{\circledast}(G)$. Now, the closure of $\mathcal{I}(G')$, denoted by $\mathit{clo}[\mathcal{I}(G')]$, is the smallest temporal-safety set such that $\mathcal{I}(G') \subseteq \mathit{clo}[\mathcal{I}(G')]$, and is defined by

$$\mathit{clo}[\mathcal{I}(G')] = A_* \cup A_\infty, \text{ where, with index } k \geq 0,$$

$$A_* = \left\{I'_{(k)} \mid I'_{(k)} \in \mathcal{I}(G'), \text{ where } k \text{ is finite}\right\},$$

$$A_\infty = \left\{I_{(\infty)} \mid I'_{(\infty)} \in \mathcal{I}(G') \;\&\; (\forall k) \left(I'_{(\infty)} = I_{(k)} - \cdots\right)\right\}.$$

By a modest generalization of the case [10], [11] where $A_* = \emptyset$, the result remains that $\mathit{clo}[\mathcal{I}(G')]$ is expressible by an LTL safety formula. And note that it is always the case that

$$\mathcal{I}(G') \subseteq \mathit{clo}[\mathcal{I}(G')] \cap \mathcal{I}^{\circledast}(G).$$

Given that DES G is finite state, it thus follows that $\mathcal{I}(G')$ is a temporal safety also expressible by some LTL safety formula $\Box\,\psi$, such that ψ is the kernel of some invariant and:

$$\begin{cases} (\forall I \in \mathcal{I}(G))(\forall k \geq 0) \\ \models^{I} (\Box\,\psi = \chi), \models^{I_{(k)}} (\chi \rightarrow \Box\,\psi), \text{ and} \\ \models^{I_{(k)}} \Box\,\psi \rightarrow (\exists I' \in \mathcal{I}(G)) \models^{I'} \Box\,\psi \text{ or } \models^{I'_{(j)}} \chi, \\ \text{for } I'_{(k)} = I_{(k)} \text{ and some } j \geq k. \end{cases}$$

Accordingly, a decidable ψ exists which is the kernel of some invariant, such that $\Box\,\psi \overset{G}{\simeq} \chi$. ■

The required syntactic computability result may now be presented.

Theorem 1: Consider the case that DES G is finite state and the specification pair $(P, \mathcal{M})$ is decidable. Then operator Ω (1) is syntactically computable in the sequence $\{K_j\}$ (2) (for the specification pair).

Proof: Define $\bigcirc_{u*} \equiv \prod_{n\geq 0} \bigcirc_u^n$, called the conjunctive or product closure of $\bigcirc_u$. Given an arbitrary decidable ψ_j as the kernel of some invariant over DES G, the weakest solution (in R) of two 'simultaneous equations' on, respectively, the conditions of Σ_u-invariance and ψ_j-history boundedness:

$$G \models \Box(\ominus_u(R) \rightarrow R) \text{ and } G \models \Box(R \rightarrow \boxminus\,\psi_j),$$

is obtained as $R \equiv \bigcirc_{u*}(\boxminus\,\psi_j)$ [16, Theorem 20] or, equivalently, $R \equiv \boxminus\bigcirc_{u*}(\psi_j)$. Over DES G, a model with finite state set Q and a deterministic state transition function, it can be shown for the decidable $\bigcirc_{u*}(\psi_j)$, with the aid of the syntax-based, iterative computing method in [16], that

$$\bigcirc_{u*}(\psi_j) \equiv \psi_j \cdot \prod_{i=1}^{n} B_i \text{ for some (finite) } n \geq 0, \tag{3}$$

where, for some P'_i that is the kernel of some invariant,

$$B_i \equiv P'_i, \text{ or } B_i \equiv \left(T_i \rightarrow P'_i\right) \text{ with } T_i \equiv \prod_{p=0}^{r} \bigcirc^p(\tau_{\sigma_i^p})$$

for some $r \geq 0$, such that $\sigma_i^p \in \Sigma_u$ $(0 \leq p \leq r)$ and, for an arbitrary $I \in \mathcal{I}(G)$ and an arbitrary $k \geq 0$,

$$\left((\forall I' \in \mathcal{I}(G), I'_{(k)} = I_{(k)}) \models^{I'^{(k)}} (T_i \rightarrow P'_i)\right) \text{ iff } \models^{I^{(k)}} P'_i.$$

This is because each such P'_i exists such that

$$\left((\forall I' \in \mathcal{I}(G), I'_{(k)} = I_{(k)}) \models^{I'^{(k)}} (T_i \rightarrow P'_i)\right) \rightarrow \models^{I^{(k)}} P'_i. \tag{4}$$

This P'_i is constructed such that $(T_i \rightarrow \bigcirc P''_i) \equiv (T_i \rightarrow P'_i)$ or, equivalently, $T_i \cdot \bigcirc\overline{P''_i} \equiv T_i \cdot \overline{P'_i}$ for the given or a previously computed kernel P''_i of some invariant, and such that it is logically in the form:

$$P'_i \equiv \overline{H'_i} + \overline{\sum_{q\in Q_i} p_q}, \tag{5}$$

where H'_i is some past formula and state set $Q_i = \left\{q_k \in Q \mid (\exists I' \in \mathcal{I}(G)) \models^{I'^{(k)}} T_i \cdot \bigcirc\overline{P''_i}\right\}$, thus implying that

$$\models^{I^{(k)}} \overline{P'_i} \rightarrow \left((\exists I' \in \mathcal{I}(G), I'_{(k)} = I_{(k)}) \models^{I'^{(k)}} T_i \cdot \overline{P'_i}\right)$$

– the contrapositive of (4).

Now, let

$$P_{j+1} \equiv \left(\psi_j \cdot \prod_{i=1}^{n} P'_i\right) \tag{6}$$

– and refer to it as the embedded kernel of $\bigcirc_{u*}(\psi_j)$ (3). Then $\boxminus P_{j+1}$ is the weakest solution that is an invariant (a past formula, whose kernel is P_{j+1}). Along with Proposition 1 that $\mathcal{C}(\psi_j)$, i.e., $\mathcal{C}(\psi_j, \emptyset)$, contains a unique supremal or weakest controllable (canonical safety) subformula of $\Box\,\psi_j$ if $\mathcal{C}(\psi_j) \neq \emptyset$, it then follows that $\mathcal{C}(\psi_j) \neq \emptyset$, such that $\sup\mathcal{C}(\psi_j) \approx \Box P_{j+1}$ iff $G \models P_{j+1}$.

In the sequence $\{K_j\}$ (2) for the specification pair $(P, \mathcal{M})$, K_0 is the pair $(P, \mathcal{M})$. Given that $(P, \mathcal{M})$ is decidable, K_0 is decidable. Now, assume K_j $(j \geq 0)$ is decidable. Then, since DES G is finite state, by Lemma 1, a decidable ψ_j, for which $\bigcirc_{u*}(\psi_j)$ is also decidable over DES G, can be found such that $\Box\,\psi_j \overset{G}{\simeq} K_j$, with $\sup\mathcal{C}(\psi_j)$ in K_{j+1} given by

$$\sup\mathcal{C}(\psi_j) \approx \begin{cases} \Box P_{j+1}, & \mathit{if} G \models P_{j+1} \\ \mathit{false}, & \mathit{otherwise}, \end{cases} \tag{7}$$

where the kernel P_{j+1} (6) in (7) can be computed by the following syntax-based method:

Define an operator H as follows: $H(R) \equiv \psi_j \cdot \bigcirc_u(R^0)$, where R^0 is 'kernelized' R, i.e., R^0 is always equal to R but with every product component $\left(\tau_\sigma \rightarrow P'\right)$, $\sigma \in \Sigma_u$, replaced by P' that is the kernel of some invariant. Then, procedurally identical to the syntax-based method in [16] for computing $\bigcirc_{u*}(\psi_j)$ over DES G, compute via finite iteration of operator H along the monotone decreasing sequence:

$$R_0 \equiv \psi_j \text{ and } R_{k+1} \equiv H(R_k), \quad k = 0, 1, \cdots$$

until a fixpoint W of H is reached, i.e., $H(W) \equiv W$, in which case $P_{j+1} \equiv W$ if it is in the form (6) [where its component past formulas P'_i $(1 \leq i \leq n)$ are each of the form (5)]. To converge to the form (6), it suffices to construct, at iteration $k+1$, some state formulas $p'_{1,s}, p'_{2,s}$ such that

$$\tau_\sigma \cdot \left(\overline{H'} + \bigcirc\overline{p''_s}\right) \equiv \tau_\sigma \cdot \left(\overline{H'} + p'_{1,s}\right) \cdot (p'_{1,s} + p'_{2,s})$$

for every $\left(\tau_\sigma \rightarrow H' \cdot \bigcirc p''_s\right)$ present following past formula expansion [6, p. 219] and LTL reasoning, where $\sigma \in \Sigma_u$, H' is some past formula and p''_s is some state formula, and such that they are logically in the form:

$$p'_{1,s} \equiv \sum_{q\in Q'_1} p_q, \;\; p'_{2,s} \equiv \sum_{q\in Q'_2} p_q,$$

where

$$Q'_1 = \left\{q_k \in Q \mid (\exists I' \in \mathcal{I}(G)) \models^{I'^{(k)}} \tau_\sigma \cdot \bigcirc\overline{p''_s}\right\},$$
$$Q'_2 = \left\{q_k \in Q \mid (\exists I' \in \mathcal{I}(G)) \models^{I'^{(k)}} \tau_\sigma \cdot \overline{H'}\right\}.$$

Since $\bigcirc_{u*}(\psi_j)$ is decidable, so is kernel P_{j+1}. It follows that $K_{j+1} \approx Y_m \cdot \sup\mathcal{C}(\psi_j)$ is decidable. Therefore, by mathematical induction, K_j is decidable for all $j \geq 0$. This implies Ω (1) is syntactically computable in $\{K_j\}$ (2). ■

C. CONVERGENCE OF Ω-ITERATION TO $\sup\mathcal{C}(P, \mathcal{M})$

The key result of this section answers the $\{K_j\}$-convergence question.

Theorem 2: Consider the case that DES G is finite state and the specification pair $(P, \mathcal{M})$ is decidable. Then the sequence $\{K_j\}$ (2) over DES G for $(P, \mathcal{M})$ converges after a finite number of iterations i to the limit $K_i, \Box\phi \overset{G}{\simeq} K_i$, where $\sup\mathcal{C}(P, \mathcal{M}) \approx \Box\phi$.

Proof: In the sequence $\{K_j\}$ (2) over DES G, $K_0 \approx Y_m \cdot \Box P, \Box\psi_0 \overset{G}{\simeq} Y_m \cdot \Box P$, and so by Proposition 2,

$$\Box\psi_0 \approx Y_m \cdot \Box P.$$

Since Ω (1) is syntactically computable in $\{K_j\}$ (2) by Theorem 1, $K_j \approx Y_m \cdot \sup\mathcal{C}(\psi_{j-1})$ exists for all $j \geq 1$. And since $\Box\psi_j \overset{G}{\simeq} Y_m \cdot \sup\mathcal{C}(\psi_{j-1})$, likewise,

$$\Box\psi_j \approx Y_m \cdot \sup\mathcal{C}(\psi_{j-1}). \tag{8}$$

Thus, for all $j \geq 0$, since $\sup\mathcal{C}(\psi_j) \rightsquigarrow \Box\psi_j$, it follows that

$$\sup\mathcal{C}(\psi_0) \rightsquigarrow \Box P,$$

and for all $j \geq 1$,

$$\sup\mathcal{C}(\psi_j) \rightsquigarrow \sup\mathcal{C}(\psi_{j-1}).$$

Because the state set of DES G is finite, the resulting state trajectory set $\mathcal{I}^{\circledast}(G)$ is finite, as immediate from the (standard) description of finite state G and elementary combinatorics. In syntactically computing the component formula $\sup\mathcal{C}(\psi_j)$ under $\{K_j\}$ (2), each iteration $j+1$ therefore removes some $n_j \geq 0$ state trajectories from the subset $\mathcal{L}_j \subseteq \mathcal{I}^{\circledast}(G)$ of DES state trajectories satisfying $\Box\psi_j$, successively reducing $\mathcal{L}_j$ by n_j trajectories, and to an empty set provided $\sup\mathcal{C}(\psi_j) \approx \mathit{false}$. Therefore, there exists a $j = i \geq 0$,

$$K_i \approx \begin{cases} Y_m \cdot \Box P, & \text{if } i = 0 \\ Y_m \cdot \sup\mathcal{C}(\psi_{i-1}), & \text{otherwise,} \end{cases}$$

such that

$$\begin{aligned} K_{i+1} &\approx \Omega(K_i) \\ &\approx Y_m \cdot \sup\mathcal{C}(\psi_i), \text{ where } \Box\psi_i \overset{G}{\simeq} K_i \\ &\approx Y_m \cdot \Box\psi_i \ (\because \Box\psi_i \text{ is controllable or } \mathit{false}) \\ &\approx K_i \text{ (By Proposition 3: P3.2).} \end{aligned}$$

This means $\Omega(K_i) \approx K_i$ at iteration $i + 1$ with $i \geq 0$, where, in letting $\psi_i \equiv \phi$ so that $\Box\phi \overset{G}{\simeq} K_i$, the result follows by Propositions 13 and 14. ■

D. COMPUTATION OF DELIMITING SAFETY CLOSURE

Lemma 1 is an important supporting result on existence of the exact delimiting safety-closure formula for an LTL formula over the DES model. Its proof, however, does not furnish a general procedure for determining or constructing the formula. In pointing a general direction for finding such a delimiting safety-closure formula in the case of finite state DES G and an arbitrary decidable specification pair $(P', \mathcal{M})$, it is noted that, under DES state-uniqueness, the ψ that exists by Lemma 1, for which $\Box\psi \overset{G}{\simeq} Y_m \cdot \Box P'$, i.e., $\Box\psi$ is the exact delimiting safety closure formula for $Y_m \cdot \Box P'$, can be logically expressed in the general form:

$$\psi \equiv \left(D \cdot P'\right). \tag{9}$$

In general, formula D is called the conditional state-forbiddance refinement on P' by ψ under invariance to assure exact delimitation of $Y_m \cdot \Box P'$. It assumes one of the three possible cases, DSF-1 to DSF-1, as explained below. Under the first two cases, at least one state trajectory $I \in \mathcal{I}^{\circledast}(G)$ satisfies $Y_m \cdot \Box P'$. Below, the cases are described based on the definition of $\overset{G}{\simeq}$.

DSF-1) $D \equiv \prod_{i=1}^{n} (A_i \to C_i)$.

In Case DSF-1, starting from an initial DES state, D on P' under invariance lets P' stay true without the DES entering any state identified as forbidden in the consequent state formula C_i, whenever the corresponding antecedent past formula A_i is true. This refinement of D on P' is such that no state trajectory $I \in \mathcal{I}^{\circledast}(G)$ satisfying $Y_m \cdot \Box P'$ does not satisfy $\Box(D \cdot P')$, every $\Box(D \cdot P')$-satisfied $I \in \mathcal{I}(G)$ satisfies $Y_m \cdot \Box P'$, and every $\Box(D \cdot P')$-satisfied prefix $I_{(k)}$ of an arbitrary $I \in \mathcal{I}(G)$ can be extended to (i.e., is a prefix of) some $I' \in \mathcal{I}(G)$ satisfying $\Box(D \cdot P')$ or its prefix $I'_{(j)}$ $(j \geq k)$ satisfying $Y_m \cdot \Box P'$.

DSF-1) $D \equiv \mathit{true}$.

In Case DSF-1, every $\Box P'$-satisfied $I \in \mathcal{I}(G)$ satisfies $Y_m \cdot \Box P'$, and every $\Box P'$-satisfied prefix $I_{(k)}$ of an arbitrary $I \in \mathcal{I}(G)$ can be extended to some $I' \in \mathcal{I}(G)$ satisfying $\Box P'$ or its prefix $I'_{(j)}$ $(j \geq k)$ satisfying $Y_m \cdot \Box P'$. Note that, by Proposition 5, the former condition defines the $\mathcal{M}$-liveness under conditional invariance while the latter defines the $\left(\mathcal{M}, \boxminus P'\right)$-condition invariance, both of invariant $\boxminus P'$.

DSF-1) $D \equiv \mathit{false}$.

In Case DSF-1, no state trajectory $I \in \mathcal{I}^{\circledast}(G)$ satisfies $Y_m \cdot \Box P'$, i.e., no $I \in \mathcal{I}(G)$ or its prefix $I_{(k)}$ that is $\Box P'$-satisfied satisfies Y_m.

V. LOGIC MODELING & WORKED EXAMPLES

In logic modeling of DES G, transition relations [6] axiomatize the DES's possible transitions by abbreviating event-transition logics in terms of state variables in Π. For the purpose of mathematical computation by logic reasoning, DES G is axiomatized by an LTL formula κ_G that is a product of DES G's initial condition, transition relations of events in Σ, and event-fairness formulas (the legal conditions) of those in $\Sigma_{\mathcal{F}}$, such that for every state trajectory I of G,

$$I \in \mathcal{I}(G) \text{ iff } \models^{I} \kappa_G.$$

The DES model G is usually a modular (synchronous) composition of a finite number of component process models $G_1, G_2, \cdots, G_n$ ($n \geq 1$) of the same type (in terms of model structure). The overall model runs by events interleaving and synchronization of shared events among its component processes. An event σ is said to be shared between processes G_j and G_k ($j \neq k$) if $\sigma \in \Sigma_j \cap \Sigma_k$. Let $G = G_1 \parallel G_2 \parallel \cdots \parallel G_n$ represent a modular DES, where $G_i = (\Pi_i, Q_i, \Sigma_i, \delta_i, \theta_i)$ and their (finite) state variable sets are mutually disjoint, i.e., for $j \neq k$, $\Pi_j \cap \Pi_k = \emptyset$. Then the synchronous operator $\parallel$ for DES G with (finite) event set $\Sigma = \bigcup_{i=1}^{n} \Sigma_i$ is logically defined with:

1) the system's initial condition θ given by $\theta \equiv \prod_{i=1}^{n} \theta_i$, and
2) the transition relation of an arbitrary $\sigma \in \Sigma$ expressed in the form:
$$\tau_\sigma \equiv \prod_{i} \sum_{j,k} \left(p_{q_{i,j}} \cdot \bigcirc p_{q_{i,k}}\right),$$
where G_i ($1 \leq i \leq n$) is every process with $\sigma \in \Sigma_i$ that is defined at some state in Q_i, such that for every state pair $(q_{i,j}, q_{i,k}) \in Q_i \times Q_i$, $q_{i,k} = \delta_i(\sigma, q_{i,j})$. By the asynchrony of occurrences of $\sigma \notin \Sigma_{i'}$ ($i' \neq i$) in every such G_i with $G_{i'}$,
$$\tau_\sigma \cdot \bigcirc p_q \equiv \tau_\sigma \cdot p_q, \quad \forall q \in Q_{i'}.$$

Note that the transition relation model above is a modular generalization and a slight logical variant of that in [6] which was first used in [17] for LTL control synthesis. The transition relation form may be expanded into a logical sum of product terms, each term of the form $h \cdot \bigcirc t$, where h, t are state formulas. Possible natural system dynamics that help simplify a relation include the following:

1) Inaccessibility of h-satisfied state (in state set $Q = Q_1 \times Q_2 \times \cdots \times Q_n$).

This may arise due to event synchronization under DES G's state transition function with respect to θ. Such an inaccessibility constraint is of the form $h \equiv \mathit{false}$ (if it exists).

After applying every inaccessibility constraint and re-expressing into the original product form, the new transition relation is obtained:

$$\tau_\sigma \equiv \prod_{i} \sum_{j',k'} \left(p_{q_{i,j'}} \cdot \bigcirc p_{q_{i,k'}}\right),$$

where i ($1 \leq i \leq n$) is the index of every process involved in the original transition relation, such that $(q_{i,j'}, q_{i,k'}) \in Q_i \times Q_i$, where $q_{i,k'} = \delta_i(\sigma, q_{i,j'})$, is every remaining state pair. Equivalently,

$$\tau_\sigma \equiv \sum_{j',k'} \left(p_{q_{i,j'}} \cdot \bigcirc p_{q_{i,k'}}\right),$$

where i is the index of any process involved in the new transition relation above.

1) Guaranteed accessibility of t-satisfied state (next).

This is due to event singularity at h-satisfied state, in that $\sigma \in \Sigma$ is the only event defined at every such state. Such an accessibility constraint is of the form $h \cdot \bigcirc t \equiv h$ (if it exists).

Next, the natural structure of DES G, by design, may be such that, under the invariance of some kernel ψ starting from an initial state, the DES, in reaching some state $q_d \in Q$, must have evolved from some state $q_s \in Q$ upstream. This structural attribute, if it exists, induces an accessibility constraint of the form $\Box\psi \cdot \Diamond p_{q_d} \rightsquigarrow \Diamond p_{q_s}$ or, equivalently, $\Box\psi \cdot \Box\overline{p_{q_s}} \rightsquigarrow \Box\overline{p_{q_d}}$. Because of their logical truth over every k-prefix of an arbitrary DES state trajectory, such accessibility constraints, along with the LTL proof system [6], may be used in logic calculations to help simplify an LTL formula for DES G.

With each component process G_i ($1 \leq i \leq n$) itself a DES model, the model operational premise (regarding event occurrences) applies locally to G_i, albeit subject to the same premise being applied to modular DES G which is necessarily constrained by synchronization of shared events between the component processes. The events that are fair in each G_i are as fair in modular DES G.

In what follows, four worked examples are provided to mainly illustrate the synthesis results of syntactic computability (Theorem 1) and convergence (Theorem 2), with each example DES model axiomatized as described above. In these examples, the first three of which are adapted from [2] while the last is adapted from [5], propositional LTL is used; each example DES G is finite state and has one initial state. An edge-labeled directed graph is used to represent a finite state DES model G_i. In this graph, a node denotes a DES state; a σ-labeled edge, directing a node denoting a state $q \in Q_i$ to a node denoting a state $q' \in Q_i$, denotes the transition of event $\sigma \in \Sigma_i$ from q to q', as defined by $\delta_i(\sigma, q) = q'$. The node with an entering arrow denotes the initial state. As appropriate to each example, a characterizing proposition or a denoting symbol for a DES state is written beside its node. Besides, in these examples, a given DES G, whether monolithic ($n = 1$) or modular ($n \geq 2$), has every component process G_i associated with one marker condition in the overall system marker set $\mathcal{M}$. As each system marker condition is specified by a state formula for G_i, a darkened node can be and is used to identify each state in G_i satisfying the associated marker condition.

For these examples, accordingly, Theorem 2 ensures a finite number of iterations computing $\Omega(K_j)$, starting from $j = 0$ along the sequence $\{K_j\}$ (2) over DES G to obtain $\sup \mathcal{C}(P, \mathcal{M})$. Let $P_0 \equiv P$. At iteration $j + 1$, with $K_j \approx Y_m \cdot \Box P_j$, first find the kernel ψ_j that exists by Lemma 1, such that $\Box\psi_j \overset{G}{\simeq} K_j$, where $\psi_j \equiv \left(D_j \cdot P_j\right)$ (9) and D_j assumes one of the three possible cases, DSF-1 to DSF-1, as discussed earlier. Then apply the syntax-based method contained in the proof of Theorem 1, to obtain $\sup \mathcal{C}(\psi_j)$ in $\Omega(K_j)$ as follows: Using the transition relation modeling of DES G as the axiomatic basis, perform syntax-based calculations accordingly to obtain the successive P_{j+1} – the embedded

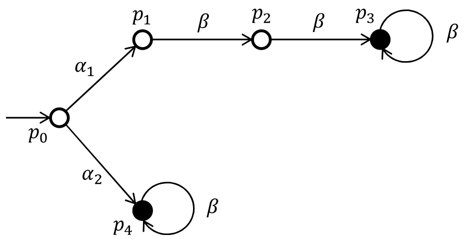

FIGURE 1. The DES *G* for Example 1.

kernel of $\bigcirc_{u*}(\psi_j)$. In completing iteration $j+1$, determine $\sup\mathcal{C}(\psi_j)$ in $\Omega(K_j)$ as either $\Box P_{j+1}$ or *false*, according to (7).

In general solution form, $\sup\mathcal{C}(P,\mathcal{M}) \approx \Box(D\cdot P)$, where D is an iteration outcome of $\{K_j\}$-convergence. In the interesting case where D is neither always equal to *true* nor *false*, it means that for each component formula of the product D added in one iteration of Ω to ensure supremal controllability of temporal safety, there is in general another component of D added in the subsequent iteration to establish the required $\overset{G}{\simeq}$-relation, so as to, as explained by Proposition 4, the fact that $(D\cdot P)$ is initially satisfied, and the formal definition of $\mathcal{M}$-directingness, thwart the violation of $\mathcal{M}$-directingness that adding the former component otherwise causes.

A. EXAMPLE 1

For the DES G depicted in Fig. 1, the (unique state) propositions over state set Q are related as follows:

$$p_i \equiv \overline{\sum_{j\in Q\setminus\{i\}} p_j}, \quad \forall i \in Q.$$

Initial condition $\theta \equiv p_0$. With event set $\Sigma = \{\alpha_1, \alpha_2, \beta\}$, the transition relations are as follows:

$$\begin{aligned}
\tau_{\alpha_1} &\equiv p_0 \cdot \bigcirc p_1,\\
\tau_{\alpha_2} &\equiv p_0 \cdot \bigcirc p_4,\\
\tau_{\beta} &\equiv p_1 \cdot \bigcirc p_2 + p_2 \cdot \bigcirc p_3 + p_3 \cdot \bigcirc p_3 + p_4 \cdot \bigcirc p_4\\
&\equiv p_1 + p_2 + p_3 + p_4 \text{ (By event singularity).}
\end{aligned}$$

Finally, it is given that $\Sigma_u = \{\beta\}$, $\Sigma_{\mathcal{F}} = \emptyset$.

Consider the specification pair $(P, \mathcal{M})$:

$$\begin{aligned}
P &\equiv \ominus\overline{p_3},\\
\mathcal{M} &= \{p_3 + p_4\};\ \text{ and therefore } Y_m \equiv \Box\Diamond(p_3 + p_4).
\end{aligned}$$

$G \models Y_m$ by the transition structure of and operational premise for the DES G. Hence any invariant over G is $\mathcal{M}$-alive under conditional invariance. In the following computation, each exact delimiting safety-closure formula $\Box\psi$ (under $\overset{G}{\simeq}$) for the given and successive pair $(P', \mathcal{M})$ is both determined with $\psi \equiv P'$ (a case DSF-1), by observation that invariant $\boxminus P'$ is $\big(\mathcal{M}, \boxminus P'\big)$-condition invariant, and by applying Proposition 5.

$$\begin{aligned}
K_0 &\approx Y_m \cdot \Box P.\\
K_1 &\approx \Omega(K_0),\ \ \Box P \overset{G}{\simeq} Y_m \cdot \Box P \ \text{ (A case DSF-1)}\\
&\approx Y_m \cdot \sup\mathcal{C}(P).
\end{aligned}$$

To compute the embedded kernel P_1 of $\bigcirc_{u*}(P)$, let $R_0 \equiv P$. Then:

$$\begin{aligned}
R_1 &\equiv H(R_0)\\
&\equiv P \cdot \bigcirc_u(R_0^0)\\
&\equiv P \cdot \bigcirc_u(P)\\
&\equiv P \cdot \big(\tau_u \rightarrow \bigcirc(\ominus\overline{p_3})\big)\\
&\equiv P \cdot (\tau_\beta \rightarrow \overline{p_3})\\
&\equiv P \cdot \overline{p_3}\ \ [\because \tau_\beta \cdot p_3 \equiv p_3].\\
R_2 &\equiv H(R_1)\\
&\equiv P \cdot \bigcirc_u(R_1^0)\\
&\equiv P \cdot \bigcirc_u\,(P \cdot \overline{p_3})\\
&\equiv P \cdot \bigcirc_u(P) \cdot \bigcirc_u(\overline{p_3})\\
&\equiv R_1 \cdot \big(\tau_\beta \rightarrow \bigcirc\overline{p_3}\big)\\
&\equiv P \cdot \overline{p_3} \cdot \overline{p_2}\ \ [\because \tau_\beta \cdot \bigcirc p_3 \equiv (p_2 + p_3)].\\
R_3 &\equiv H(R_2)\\
&\equiv P \cdot \bigcirc_u(R_2^0)\\
&\equiv P \cdot \bigcirc_u\,(P \cdot \overline{p_3} \cdot \overline{p_2})\\
&\equiv P \cdot \bigcirc_u\,(P \cdot \overline{p_3}) \cdot \bigcirc_u(\overline{p_2})\\
&\equiv R_2 \cdot \big(\tau_\beta \rightarrow \bigcirc\overline{p_2}\big)\\
&\equiv P \cdot \overline{p_3} \cdot \overline{p_2} \cdot \overline{p_1}\ \ [\because \tau_\beta \cdot \bigcirc p_2 \equiv p_1].\\
R_4 &\equiv H(R_3)\\
&\equiv P \cdot \bigcirc_u(R_3^0)\\
&\equiv P \cdot \bigcirc_u\,(P \cdot \overline{p_3} \cdot \overline{p_2} \cdot \overline{p_1})\\
&\equiv P \cdot \bigcirc_u\,(P \cdot \overline{p_3} \cdot \overline{p_2}) \cdot \bigcirc_u(\overline{p_1})\\
&\equiv R_3 \cdot \big(\tau_\beta \rightarrow \bigcirc\overline{p_1}\big)\\
&\equiv R_3\ \ [\because \tau_\beta \cdot \bigcirc p_1 \equiv \mathit{false}].
\end{aligned}$$

$\therefore$ Embedded kernel $P_1 \equiv R_3^0 \equiv P \cdot \overline{p_3} \cdot \overline{p_2} \cdot \overline{p_1}$.
Since $G \models P_1$ $[\because (\theta \rightarrow P_1) \approx \mathit{true}]$,

$$\begin{aligned}
\sup\mathcal{C}(P) &\approx \Box P_1\\
&\approx \Box\,(P \cdot \overline{p_3} \cdot \overline{p_2} \cdot \overline{p_1})\\
&\approx \Box\,\big((\ominus\overline{p_3}) \cdot \overline{p_3} \cdot \overline{p_2} \cdot \overline{p_1}\big)\\
&\approx \Box\,(\overline{p_3} \cdot \overline{p_2} \cdot \overline{p_1})\ \ \ [\because \Box\overline{p_3} \leadsto \Box\ominus\overline{p_3}]\\
&\approx \Box\overline{p_1}\ \ [\because \Box\overline{p_1} \leadsto \Box\overline{p_3}, \Box\overline{p_1} \leadsto \Box\overline{p_2}\\
&\qquad\qquad \text{(two accessibility constraints)}].
\end{aligned}$$

$$\therefore K_1 \approx Y_m \cdot \Box P_1 \approx Y_m \cdot \Box\overline{p_1}.$$

$\because \Box\overline{p_1} \overset{G}{\simeq} Y_m \cdot \Box\overline{p_1}$ (A case DSF-1), $\Box\overline{p_1} \approx \Box P_1$, and $\Box P_1$ is controllable,

$$\begin{aligned}
K_2 &\approx \Omega(K_1),\ \ \Box P_1 \overset{G}{\simeq} Y_m \cdot \Box P_1\\
&\approx Y_m \cdot \sup\mathcal{C}(P_1)\\
&\approx Y_m \cdot \Box P_1\\
&\approx K_1.\\
\therefore \sup\mathcal{C}(P,\mathcal{M}) &\approx \Box P_1\\
&\approx \Box\overline{p_1}.
\end{aligned}$$

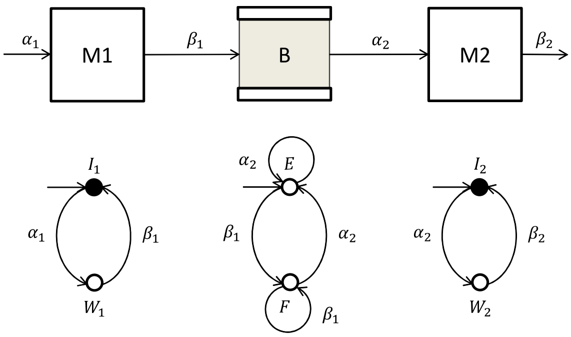

FIGURE 2. **The DES G = M1 ∥ B ∥ M2 for Example 2. In this example of a simple manufacturing system, M1, M2 are DES models of two machines connected in tandem. Each machine Mi ($1 \leq i \leq 2$) is either idling (I_i) or working (W_i). In its idling state, the machine takes (α_i) a workpiece from one buffer for processing and transitions to its working state. Once finished with processing in its working state, it deposits (β_i) the finished piece into another buffer that transitions it back to its idling state. Only the buffer in between the two machines is shown. Of unit size, this buffer is modeled by B defining the propositions E (buffer empty), F (buffer full), such that M1 ∥ B ∥ M2 is M1 ∥ M2 with state characterization enriched by these propositions.**

B. EXAMPLE 2

For the modular DES G = M1 ∥ B ∥ M2 depicted and described in Fig. 2, let Z_i be the set of (unique process-state) propositions over state set Q_i of process G_i ($1 \leq i \leq 3$), with $G_1 = \text{M1}$, $G_2 = \text{M2}$, $G_3 = \text{B}$. Then:

$$p \equiv \overline{\sum_{p' \in Z_i \setminus \{p\}} p'}, \quad \forall p \in Z_i.$$

Initial condition $\theta \equiv (I_1 \cdot E \cdot I_2)$. With event set $\Sigma = \Sigma_1 \cup \Sigma_2$, where $\Sigma_1 = \{\alpha_1, \beta_1\}$, $\Sigma_2 = \{\alpha_2, \beta_2\}$, $\Sigma_3 = \{\beta_1, \alpha_2\}$, the transition relations are as follows:

$$\begin{aligned}
\tau_{\alpha_1} &\equiv I_1 \cdot \bigcirc W_1, \\
\tau_{\beta_1} &\equiv (W_1 \cdot \bigcirc I_1) \cdot (E \cdot \bigcirc F + F \cdot \bigcirc F) \\
&\equiv (W_1 \cdot \bigcirc I_1) \cdot \bigcirc F \text{ (By logic simplification)} \\
&\equiv (W_1 \cdot \bigcirc I_1) \\
&\equiv \bigcirc F, \\
\tau_{\alpha_2} &\equiv (I_2 \cdot \bigcirc W_2) \cdot (F \cdot \bigcirc E + E \cdot \bigcirc E) \\
&\equiv (I_2 \cdot \bigcirc W_2) \cdot \bigcirc E \text{ (By logic simplification)} \\
&\equiv (I_2 \cdot \bigcirc W_2) \\
&\equiv \bigcirc E, \\
\tau_{\beta_2} &\equiv W_2 \cdot \bigcirc I_2.
\end{aligned}$$

For $\sigma \in \Sigma_i$, but $\sigma \notin \Sigma_j$ ($j \neq i$, $1 \leq i, j \leq 3$):

$$\tau_\sigma \cdot \bigcirc p_q \equiv \tau_\sigma \cdot p_q, \quad \forall q \in Q_j.$$

Finally, it is given that $\Sigma_u = \{\beta_1, \beta_2\}$, $\Sigma_{\mathcal{F}} = \emptyset$.

Consider the specification pair $(P, \mathcal{M})$ with $P \equiv (P^1 \cdot P^2)$:

$$\begin{aligned}
P^1 &\equiv \left(\ominus(F \cdot W_1) \rightarrow \overline{I_1}\right), \\
P^2 &\equiv \left(\ominus(E \cdot I_2) \rightarrow \overline{W_2}\right),
\end{aligned}$$

$$\mathcal{M} = \{I_1, I_2\}; \text{ and therefore } Y_m \equiv \prod_{i=1}^{2} \Box \Diamond I_i.$$

The first product component of the temporal-safety part $\Box(P^1 \cdot P^2)$ may be paraphrased as follows: 'Machine M1 is not to have deposited another workpiece into Buffer B whenever, previously, the buffer is full while it is working.' The second component may be paraphrased as follows: 'Machine M2 is not to have acted to take another workpiece from Buffer B whenever, previously, the buffer is empty while it is idling.' In short, the former specifies no overflow and the latter specifies no underflow for Buffer B. Paraphrasing Y_m, each machine must regularly process workpieces to completion, one at a time, taking from and depositing into their respective buffers.

$G \models Y_m$ by the transition structure of and operational premise for each component process G_i ($1 \leq i \leq 2$), and the fact that under ∥, it remains that $\tau_\sigma \not\equiv false$ for each $\sigma \in \Sigma$. Hence any invariant over G is $\mathcal{M}$-alive under conditional invariance. In the following computation, each exact delimiting safety-closure formula $\Box\psi$ (under $\overset{G}{\simeq}$) for the given and successive pair $(P', \mathcal{M})$ is both determined with $\psi \equiv P'$ (a case DSF-1), by reasoning that invariant $\boxminus P'$ is $\left(\mathcal{M}, \boxminus P'\right)$-condition invariant, and by applying Proposition 5. The reasoning for condition invariance in each instance is that although the respective temporal-safety part $\Box P'$ specifies a maintenance of P', the imposed sequencing or ordering among events in G_1, G_2 that results permits all the events in the DES G to occur infinitely often.

$$\begin{aligned}
K_0 &\approx Y_m \cdot \Box P. \\
K_1 &\approx \Omega(K_0), \ \ \Box P \overset{G}{\simeq} Y_m \cdot \Box P \ \ \text{(A case DSF-1)} \\
&\approx Y_m \cdot \sup \mathcal{C}(P) \\
&\approx Y_m \cdot \sup \mathcal{C}(P^1 \cdot P^2).
\end{aligned}$$

To compute the embedded kernel P_1 of $\bigcirc_{u*}(P^1 \cdot P^2)$, let $R_0 \equiv (P^1 \cdot P^2)$. Then:

$$\begin{aligned}
R_1 &\equiv H(R_0) \\
&\equiv (P^1 \cdot P^2) \cdot \bigcirc_u(R_0^0) \\
&\equiv (P^1 \cdot P^2) \cdot \bigcirc_u(P^1 \cdot P^2) \\
&\equiv (P^1 \cdot P^2) \cdot \bigcirc_u(P^1) \cdot \bigcirc_u(P^2) \\
&\equiv (P^1 \cdot P^2) \cdot \overline{F \cdot W_1}, \quad \text{since:} \\
\bigcirc_u(P^1) &\equiv \left(\tau_u \rightarrow \left(F \cdot W_1 \rightarrow \bigcirc\overline{I_1}\right)\right) \\
&\equiv \left(F \cdot W_1 \rightarrow \left(\tau_{\beta_1} \rightarrow \bigcirc\overline{I_1}\right) \cdot (\tau_{\beta_2} \rightarrow \bigcirc\overline{I_1})\right) \\
&\equiv \overline{F \cdot W_1} \ \ [\because \tau_{\beta_1} \cdot \bigcirc I_1 \equiv true]. \\
\bigcirc_u(P^2) &\equiv \left(\tau_u \rightarrow \left(E \cdot I_2 \rightarrow \bigcirc\overline{W_2}\right)\right) \\
&\equiv \left(E \cdot I_2 \rightarrow \left(\tau_{\beta_1} \rightarrow \bigcirc\overline{W_2}\right)\right) \\
&\qquad \cdot \left(E \cdot I_2 \rightarrow \left(\tau_{\beta_2} \rightarrow \bigcirc\overline{W_2}\right)\right) \\
&\equiv \left(E \cdot I_2 \rightarrow \left(\tau_{\beta_1} \rightarrow \overline{W_2}\right)\right) \\
&\qquad \cdot \left(E \cdot I_2 \rightarrow \left(\tau_{\beta_2} \rightarrow \bigcirc\overline{W_2}\right)\right) \\
&\qquad [\because \tau_{\beta_1} \cdot \bigcirc W_2 \equiv \tau_{\beta_1} \cdot W_2] \\
&\equiv true \cdot \left(E \cdot I_2 \rightarrow \left(\tau_{\beta_2} \rightarrow \bigcirc\overline{W_2}\right)\right) \\
&\qquad \left[\because W_2 \equiv \overline{I_2}\right] \\
&\equiv true \ \ [\because \tau_{\beta_2} \cdot \bigcirc W_2 \equiv false]. \\
R_2 &\equiv H(R_1)
\end{aligned}$$

$$
\begin{aligned}
&\equiv (P^1 \cdot P^2) \cdot \bigcirc_u(R_1^0)\\
&\equiv (P^1 \cdot P^2) \cdot \bigcirc_u \left(P^1 \cdot P^2 \cdot \overline{F \cdot W_1}\right)\\
&\equiv (P^1 \cdot P^2) \cdot \bigcirc_u(P^1 \cdot P^2) \cdot \bigcirc_u \left(\overline{F \cdot W_1}\right)\\
&\equiv R_1 \cdot \bigcirc_u \left(\overline{F \cdot W_1}\right)\\
&\equiv R_1 \cdot \left(\tau_{\beta_1} \to \bigcirc\overline{F \cdot W_1}\right) \cdot \left(\tau_{\beta_2} \to \bigcirc\overline{F \cdot W_1}\right)\\
&\equiv R_1 \cdot \mathit{true} \cdot \left(\tau_{\beta_2} \to \bigcirc\overline{F \cdot W_1}\right)\\
&\qquad [\because \tau_{\beta_1} \cdot \bigcirc(F \cdot W_1) \equiv \mathit{false}]\\
&\equiv R_1 \cdot \left(\tau_{\beta_2} \to \overline{F \cdot W_1}\right)\\
&\qquad [\because \tau_{\beta_2} \cdot \bigcirc(F \cdot W_1) \equiv \tau_{\beta_2} \cdot (F \cdot W_1)]\\
&\equiv (P^1 \cdot P^2) \cdot \overline{F \cdot W_1} \cdot \left(\tau_{\beta_2} \to \overline{F \cdot W_1}\right)\\
&\equiv (P^1 \cdot P^2) \cdot \overline{F \cdot W_1}\\
&\equiv R_1.
\end{aligned}
$$

$\therefore$ Embedded kernel $P_1 \equiv R_1^0 \equiv (P^1 \cdot P^2) \cdot \overline{F \cdot W_1}$.
Since $G \models P_1$ [$\because (\theta \to P_1) \approx \mathit{true}$],

$$
\begin{aligned}
\sup \mathcal{C}(P^1 \cdot P^2) &\approx \Box P_1\\
&\approx \Box \left((P^1 \cdot P^2) \cdot \overline{F \cdot W_1}\right)\\
&\approx \Box \left(\overline{F \cdot W_1} \cdot \left(\ominus(F \cdot W_1) \to \overline{I_1}\right) \cdot P^2\right)\\
&\approx \Box \left(\overline{F \cdot W_1} \cdot \left(\tilde{\ominus}\overline{F \cdot W_1} + \overline{I_1}\right) \cdot P^2\right)\\
&\approx \Box \left(\overline{F \cdot W_1} \cdot P^2\right)\\
&\qquad \left[\because \Box\overline{F \cdot W_1} \rightsquigarrow \Box\left(\tilde{\ominus}\overline{F \cdot W_1} + \overline{I_1}\right)\right].\\
\therefore K_1 \approx Y_m \cdot \Box P_1 &\approx Y_m \cdot \Box\left(\overline{F \cdot W_1} \cdot P^2\right).
\end{aligned}
$$

$\because \Box\left(\overline{F \cdot W_1} \cdot P^2\right) \overset{G}{\simeq} Y_m \cdot \Box\left(\overline{F \cdot W_1} \cdot P^2\right)$ (A case DSF-1), $\Box\left(\overline{F \cdot W_1} \cdot P^2\right) \approx \Box P_1$, and $\Box P_1$ is controllable,

$$
\begin{aligned}
K_2 \approx \Omega(K_1),\ \ \Box P_1 \overset{G}{\simeq} Y_m \cdot \Box P_1 &\\
&\approx Y_m \cdot \sup \mathcal{C}\,(P_1)\\
&\approx Y_m \cdot \Box P_1\\
&\approx K_1.\\
\therefore \sup \mathcal{C}(P, \mathcal{M}) &\approx \Box P_1\\
&\approx \Box\left(\overline{F \cdot W_1} \cdot P^2\right)\\
&\approx \Box\left(\overline{F \cdot W_1} \cdot \left(\ominus(E \cdot I_2) \to \overline{W_2}\right)\right).
\end{aligned}
$$

C. EXAMPLE 3

For the modular DES G = CAT $\|$ MOUSE depicted and described in Fig. 3, $\Pi = \{i, j\}$, where $(i = a)$, $(j = b)$, with $a, b \in N = \{0, 1, 2, 3, 4\}$, are the (unique process-state) propositions over the process state sets of CAT, MOUSE, respectively. Then for $a, b \in N$,

$$
(i = a) \equiv \overline{\sum_{a' \in N \setminus \{a\}} (i = a')}, \quad (j = b) \equiv \overline{\sum_{b' \in N \setminus \{b\}} (j = b')}.
$$

Initial condition $\theta \equiv (i = 2) \cdot (j = 4) \equiv (i, j) = (2, 4)$. With G_1 = CAT, G_2 = MOUSE, event set $\Sigma = \Sigma_1 \cup \Sigma_2$, where $\Sigma_1 = \{c_k \mid (1 \leq k \leq 7)\}$, $\Sigma_2 = \{m_l \mid (1 \leq l \leq 6)\}$, and the transition relations are as follows:

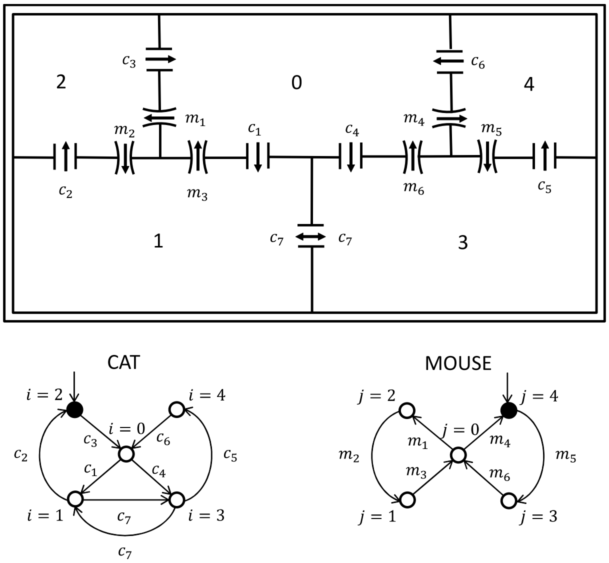

FIGURE 3. The DES G = CAT ‖ MOUSE for Example 3. In this example, a cat and a mouse are placed in a 5-room maze, as shown. CAT, MOUSE are DES models of the free room-to-room movements of the cat, mouse, respectively, in the maze. The models have the animal location variables *i, j* taking a room number (0 to 4) that the cat, mouse occupy, respectively, as states; the room-to-room movements of the animals are modeled by transitions via gateways set up for their exclusive use, with gateway depicted by ⊣ ⊢ for the cat, and gateway depicted by -) (- for the mouse. As shown in the maze, indicated by the direction of movement through a gateway, c_k ($1 \leq k \leq 7$) denotes a gateway transition by the cat, and m_l ($1 \leq l \leq 6$) denotes a gateway transition by the mouse.

$$
\begin{aligned}
\tau_{c_1} &\equiv (i = 0) \cdot \bigcirc(i = 1), & \tau_{c_5} &\equiv (i = 3) \cdot \bigcirc(i = 4),\\
\tau_{c_2} &\equiv (i = 1) \cdot \bigcirc(i = 2), & \tau_{c_6} &\equiv (i = 4) \cdot \bigcirc(i = 0),\\
\tau_{c_3} &\equiv (i = 2) \cdot \bigcirc(i = 0), & \tau_{c_7} &\equiv (i = 1) \cdot \bigcirc(i = 3)\\
\tau_{c_4} &\equiv (i = 0) \cdot \bigcirc(i = 3), & &\quad + (i = 3) \cdot \bigcirc(i = 1),\\
\tau_{m_1} &\equiv (j = 0) \cdot \bigcirc(j = 2), & \tau_{m_4} &\equiv (j = 0) \cdot \bigcirc(j = 4),\\
\tau_{m_2} &\equiv (j = 2) \cdot \bigcirc(j = 1), & \tau_{m_5} &\equiv (j = 4) \cdot \bigcirc(j = 3),\\
\tau_{m_3} &\equiv (j = 1) \cdot \bigcirc(j = 0), & \tau_{m_6} &\equiv (j = 3) \cdot \bigcirc(j = 0).
\end{aligned}
$$

For c_k ($1 \leq k \leq 7$), m_l ($1 \leq l \leq 6$), $a \in N$:

$$
\begin{aligned}
\tau_{c_k} \cdot \bigcirc(j = a) &\equiv \tau_{c_k} \cdot (j = a),\\
\tau_{m_l} \cdot \bigcirc(i = a) &\equiv \tau_{m_l} \cdot (i = a).
\end{aligned}
$$

Finally, it is given that $\Sigma_u = \{c_2, c_7, m_4\}$, $\Sigma_{\mathcal{F}} = \Sigma_{\mathcal{C}} = \Sigma_u$.
Consider the specification pair $(P, \mathcal{M})$:

$$
\begin{aligned}
P &\equiv i \neq j,\\
\mathcal{M} &= \{i = 2, j = 4\};\\
&\qquad \text{and therefore } Y_m \equiv \Box\left(\Diamond(i = 2) \cdot \Diamond(j = 4)\right).
\end{aligned}
$$

The specification states that the cat and the mouse must never be in the same room simultaneously, and each must regularly return to the room it initially occupied.

$G \models Y_m$ by the transition structure and compassionate events of, and the operational premise for, the individual component processes CAT, MOUSE, coupled with the fact that under ∥, it remains that $\tau_\sigma \not\equiv \mathit{false}$ for each $\sigma \in \Sigma$. In the following computation, each exact delimiting safety-closure formula (under $\overset{G}{\simeq}$) for the given and successive specification pair is determined to be a case DSF-1 and a case DSF-1, respectively.

Four accessibility constraints are found to be useful in the following calculations for formula simplification, as listed below:

Ex3-C1) $\Box(i \neq j) \cdot \Box((i,j) \neq (1,3)) \rightsquigarrow \Box((i,j) \neq (3,1))$,
Ex3-C2) $\Box(i \neq j) \cdot \Box((i,j) \neq (1,3)) \rightsquigarrow \Box((i,j) \neq (3,2))$,
Ex3-C3) $\Box(i \neq j) \cdot \Box((i,j) \neq (1,3)) \rightsquigarrow \Box((i,j) \neq (1,2))$,
Ex3-C4) $\Box(i \neq j) \cdot \Box((i,j) \neq (1,3)) \rightsquigarrow \Box((i,j) \neq (4,0))$.

$$K_0 \approx Y_m \cdot \Box P.$$

$$\begin{aligned} K_1 &\approx \Omega(K_0), \quad \Box P \overset{G}{\simeq} Y_m \cdot \Box P \ \ \text{(A case DSF-1)} \\ &\approx Y_m \cdot \sup \mathcal{C}(P). \end{aligned}$$

To compute the embedded kernel P_1 of $\bigcirc_{u*}(P)$, let $R_0 \equiv P$. Then:

$$\begin{aligned} R_1 &\equiv H(R_0) \\ &\equiv P \cdot \bigcirc_u(R_0^0) \\ &\equiv P \cdot \bigcirc_u(P) \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \\ &\equiv (i \neq j) \cdot \big((\tau_{c_2} + \tau_{c_7} + \tau_{m_4}) \rightarrow \bigcirc(i \neq j)\big) \\ &\equiv (i \neq j) \cdot \big(\tau_{c_2} \rightarrow \bigcirc(i \neq j)\big) \\ &\qquad \cdot \big(\tau_{c_7} \rightarrow \bigcirc(i \neq j)\big) \cdot \big(\tau_{m_4} \rightarrow \bigcirc(i \neq j)\big) \\ &\equiv (i \neq j) \cdot \big(\tau_{c_2} \rightarrow (i,j) \neq (1,2)\big) \\ &\qquad \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (1,3) \cdot (i,j) \neq (3,1)\big) \\ &\qquad \cdot \big(\tau_{m_4} \rightarrow (i,j) \neq (4,0)\big) \\ &\qquad [\because \tau_{c_2} \cdot \bigcirc(i = j) \equiv \tau_{c_2} \cdot (i,j) = (1,2),\ \tau_{c_7} \cdot \bigcirc \\ &\qquad (i = j) \equiv \tau_{c_7} \cdot ((i,j) = (1,3) + (i,j) = (3,1)), \\ &\qquad \tau_{m_4} \cdot \bigcirc(i = j) \equiv \tau_{m_4} \cdot (i,j) = (4,0)]. \end{aligned}$$

$$\begin{aligned} R_2 &\equiv H(R_1) \\ &\equiv P \cdot \bigcirc_u(R_1^0) \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (1,2)) \cdot \bigcirc_u((i,j) \neq (1,3)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (3,1)) \cdot \bigcirc_u((i,j) \neq (4,0)) \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot \bigcirc_u((i,j) \neq (1,2)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (1,3)) \cdot \bigcirc_u((i,j) \neq (3,1)) \\ &\qquad [\because \tau_{c_2} \cdot \bigcirc((i,j) = (4,0)) \\ &\qquad \equiv \tau_{c_7} \cdot \bigcirc((i,j) = (4,0)) \\ &\qquad \equiv \tau_{m_4} \cdot \bigcirc((i,j) = (4,0)) \equiv \mathit{false}] \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot \bigcirc_u((i,j) \neq (1,2)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (1,3)) \cdot (\tau_{c_7} \rightarrow (i,j) \neq (1,1)) \\ &\qquad [\because \tau_{c_2} \cdot \bigcirc((i,j) = (3,1)) \\ &\qquad \equiv \tau_{m_4} \cdot \bigcirc((i,j) = (3,1)) \equiv \mathit{false}, \\ &\qquad \tau_{c_7} \cdot \bigcirc((i,j) = (3,1)) \equiv \tau_{c_7} \cdot (i,j) = (1,1)] \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot \bigcirc_u((i,j) \neq (1,2)) \\ &\qquad \cdot (\tau_{c_7} \rightarrow (i,j) \neq (3,3)) \cdot (\tau_{c_7} \rightarrow (i,j) \neq (1,1)) \\ &\qquad [\because \tau_{c_2} \cdot \bigcirc((i,j) = (1,3)) \\ &\qquad \equiv \tau_{m_4} \cdot \bigcirc((i,j) = (1,3)) \equiv \mathit{false}, \\ &\qquad \tau_{c_7} \cdot \bigcirc((i,j) = (1,3)) \equiv \tau_{c_7} \cdot (i,j) = (3,3)] \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot (\tau_{c_7} \rightarrow (i,j) \neq (3,2)) \\ &\qquad \cdot (\tau_{c_7} \rightarrow (i,j) \neq (3,3)) \cdot (\tau_{c_7} \rightarrow (i,j) \neq (1,1)) \\ &\qquad [\because \tau_{c_2} \cdot \bigcirc((i,j) = (1,2)) \\ &\qquad \equiv \tau_{m_4} \cdot \bigcirc((i,j) = (1,2)) \equiv \mathit{false}, \\ &\qquad \tau_{c_7} \cdot \bigcirc((i,j) = (1,2)) \equiv \tau_{c_7} \cdot (i,j) = (3,2)] \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot (\tau_{c_7} \rightarrow (i,j) \neq (3,2)) \\ &\qquad [\because (i \neq j) \equiv (i \neq j) \cdot (i,j) \neq (3,3 \\ &\qquad \equiv (i \neq j) \cdot (i,j) \neq (1,1)] \\ &\equiv R_1 \cdot (\tau_{c_7} \rightarrow (i,j) \neq (3,2)) \\ &\equiv (i \neq j) \cdot \big(\tau_{c_2} \rightarrow (i,j) \neq (1,2)\big) \\ &\qquad \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (1,3)\big) \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (3,1)\big) \\ &\qquad \cdot \big(\tau_{m_4} \rightarrow (i,j) \neq (4,0)\big) \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (3,2)\big). \end{aligned}$$

$$\begin{aligned} R_3 &\equiv H(R_2) \\ &\equiv P \cdot \bigcirc_u(R_2^0) \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot \bigcirc_u((i,j) \neq (1,2)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (1,3)) \cdot \bigcirc_u((i,j) \neq (3,1)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (4,0)) \cdot \bigcirc_u((i,j) \neq (3,2)) \\ &\equiv R_2 \cdot \bigcirc_u((i,j) \neq (3,2)) \\ &\equiv R_2 \cdot (\tau_{c_7} \rightarrow (i,j) \neq (1,2)) \\ &\qquad [\because \tau_{c_2} \cdot \bigcirc((i,j) = (3,2)) \\ &\qquad \equiv \tau_{m_4} \cdot \bigcirc((i,j) = (3,2)) \equiv \mathit{false}, \\ &\qquad \tau_{c_7} \cdot \bigcirc((i,j) = (3,2)) \equiv \tau_{c_7} \cdot (i,j) = (1,2)] \\ &\equiv (i \neq j) \cdot \big(\tau_{c_2} \rightarrow (i,j) \neq (1,2)\big) \\ &\qquad \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (1,3)\big) \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (3,1)\big) \\ &\qquad \cdot \big(\tau_{m_4} \rightarrow (i,j) \neq (4,0)\big) \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (3,2)\big) \\ &\qquad \cdot \big(\tau_{c_7} \rightarrow (i,j) \neq (1,2)\big). \end{aligned}$$

$$\begin{aligned} R_4 &\equiv H(R_3) \\ &\equiv P \cdot \bigcirc_u(R_3^0) \\ &\equiv (i \neq j) \cdot \bigcirc_u(i \neq j) \cdot \bigcirc_u((i,j) \neq (1,2)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (1,3)) \cdot \bigcirc_u((i,j) \neq (3,1)) \\ &\qquad \cdot \bigcirc_u((i,j) \neq (4,0)) \cdot \bigcirc_u((i,j) \neq (3,2)) \\ &\equiv R_3. \end{aligned}$$

∴ Embedded kernel $P_1 \equiv R_3^0 \equiv (i \neq j) \cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3) \cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0) \cdot (i,j) \neq (3,2)$.
Since $G \models P_1$ [$\because (\theta \rightarrow P_1) \approx \mathit{true}$],

$$\begin{aligned} \sup \mathcal{C}(P) &\approx \Box P_1 \\ &\approx \Box((i \neq j) \cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3) \\ &\qquad \cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0) \cdot (i,j) \neq \\ &\qquad (3,2)) \\ &\approx \Box((i \neq j) \cdot (i,j) \neq (1,3)) \\ &\qquad [\because \text{of Ex3-C1} - \text{C4}] \\ &\approx \Box P_1', \text{ where } P_1' \equiv (i \neq j) \cdot (i,j) \neq (1,3). \end{aligned}$$

$$\therefore K_1 \approx Y_m \cdot \Box P_1 \approx Y_m \cdot \Box P_1'.$$

$\because\ \Box\left(P_1' \cdot (i,j) \neq (0,3)\right) \overset{G}{\simeq} Y_m \cdot \Box P_1'$ (A case DSF-1) and $\Box P_1' \approx \Box P_1$,

$$K_2 \approx \Omega(K_1),\ \ \Box(P_1 \cdot (i,j) \neq (0,3)) \overset{G}{\simeq} Y_m \cdot \Box P_1$$
$$\approx Y_m \cdot \sup \mathcal{C}\left(P_1 \cdot (i,j) \neq (0,3)\right).$$

To compute the embedded kernel P_2 of $\bigcirc_{u*}(P_1 \cdot (i,j) \neq (0,3))$, let $R_0 \equiv P_1 \cdot (i,j) \neq (0,3)$. Then:

$$\begin{aligned}
R_1 &\equiv H(R_0)\\
&\equiv P_1 \cdot (i,j) \neq (0,3) \cdot \bigcirc_u(R_0^0)\\
&\equiv P_1 \cdot (i,j) \neq (0,3) \cdot \bigcirc_u(P_1) \cdot \bigcirc_u((i,j) \neq (0,3))\\
&\equiv P_1 \cdot (i,j) \neq (0,3) \cdot \bigcirc_u(P_1)\\
&\qquad [\because \tau_{c_2} \cdot \bigcirc((i,j) = (0,3))\\
&\qquad \equiv \tau_{c_7} \cdot \bigcirc((i,j) = (0,3))\\
&\qquad \equiv \tau_{m_4} \cdot \bigcirc((i,j) = (0,3)) \equiv \mathit{false}]\\
&\equiv (i \neq j) \cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3)\\
&\qquad \cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0) \cdot (i,j) \neq (3,2)\\
&\qquad \cdot (i,j) \neq (0,3) \cdot \bigcirc_u(i \neq j) \cdot \bigcirc_u((i,j) \neq (1,2))\\
&\qquad \cdot \bigcirc_u((i,j) \neq (1,3)) \cdot \bigcirc_u((i,j) \neq (3,1))\\
&\qquad \cdot \bigcirc_u((i,j) \neq (4,0)) \cdot \bigcirc_u((i,j) \neq (3,2))\\
&\equiv (i \neq j) \cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3)\\
&\qquad \cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0) \cdot (i,j) \neq (3,2)\\
&\qquad \cdot (i,j) \neq (0,3) \cdot \left(\tau_{c_2} \rightarrow (i,j) \neq (1,2)\right)\\
&\qquad \cdot \left(\tau_{c_7} \rightarrow (i,j) \neq (1,3)\right) \cdot \left(\tau_{c_7} \rightarrow (i,j) \neq (3,1)\right)\\
&\qquad \cdot \left(\tau_{m_4} \rightarrow (i,j) \neq (4,0)\right) \cdot \left(\tau_{c_7} \rightarrow (i,j) \neq (3,2)\right)\\
&\qquad \cdot \left(\tau_{c_7} \rightarrow (i,j) \neq (1,2)\right)\\
&\qquad [\text{By } R_4 \equiv R_3 \text{ under iteration } K_1]\\
&\equiv (i \neq j) \cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3)\\
&\qquad \cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0) \cdot (i,j) \neq (3,2)\\
&\qquad \cdot (i,j) \neq (0,3)\\
&\equiv P_1 \cdot (i,j) \neq (0,3)\\
&\equiv R_0.
\end{aligned}$$

$\therefore$ Embedded kernel $P_2 \equiv R_0^0 \equiv P_1 \cdot (i,j) \neq (0,3) \equiv (i \neq j) \cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3) \cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0) \cdot (i,j) \neq (3,2) \cdot (i,j) \neq (0,3)$.
Since $G \models P_2$ $[\because (\theta \rightarrow P_2) \approx \mathit{true}]$,

$$\begin{aligned}
\sup \mathcal{C}(P_1 \cdot (i,j) \neq (0,3)) &\approx \Box P_2\\
&\approx \Box(P_1 \cdot (i,j) \neq (0,3)).\\
\therefore K_2 &\approx Y_m \cdot \Box P_2\\
&\approx Y_m \cdot \Box(P_1 \cdot (i,j) \neq (0,3))\\
&\approx Y_m \cdot \Box P_1\\
&\qquad [\text{By Proposition 3: P3.2}]\\
&\approx K_1.
\end{aligned}$$

Now,

$$\begin{aligned}
\Box P_2 \approx \Box((i \neq j) &\cdot (i,j) \neq (1,2) \cdot (i,j) \neq (1,3)\\
&\cdot (i,j) \neq (3,1) \cdot (i,j) \neq (4,0)\\
&\cdot (i,j) \neq (3,2) \cdot (i,j) \neq (0,3))\\
\approx\ &\Box((i \neq j) \cdot (i,j) \neq (1,3) \cdot (i,j) \neq (0,3))\\
&\qquad [\because \text{ of Ex3-C1} - \text{C4}]\\
\approx\ &\Box P_2', \text{ where}\\
&\qquad P_2' \equiv (i \neq j) \cdot (i,j) \neq (1,3) \cdot (i,j) \neq (0,3).
\end{aligned}$$

$$\begin{aligned}
\therefore \sup \mathcal{C}(P, \mathcal{M}) &\approx \Box P_2\\
&\approx \Box P_2'\\
&\approx \Box((i \neq j) \cdot (i,j) \neq (1,3) \cdot (i,j) \neq (0,3)).
\end{aligned}$$

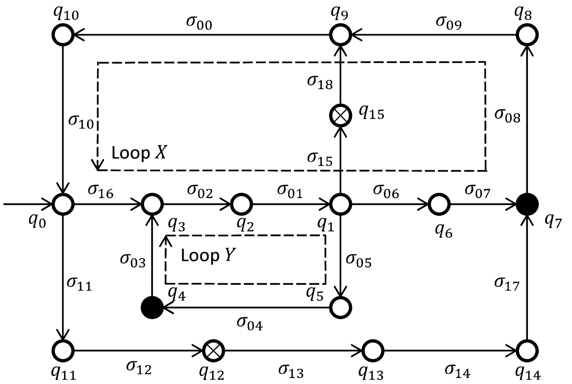

FIGURE 4. The DES G for Example 4. The forbidden states specified by P are denoted by nodes marked with a cross.

D. EXAMPLE 4

For the DES G depicted in Fig. 4, the states are unique, initial condition $\theta \equiv p_{q_0}$, and it is given that $\Sigma_u = \emptyset$; therefore $\Sigma_{\mathcal{F}} = \emptyset$.

Consider the specification pair $(P, \mathcal{M})$:

$$P \equiv \overline{p_{q_{12}} + p_{q_{15}}},$$
$$\mathcal{M} = \{p_{q_4}, p_{q_7}\};\ \text{ and therefore } Y_m \equiv \prod_{i \in \{4,7\}} \Box\Diamond p_{q_i}.$$

With $\Sigma_u = \emptyset$, clearly $\Box P$ is controllable, since $G \models P$ because $(\theta \rightarrow P) \approx \mathit{true}$. However, $\boxminus P$ is not $(\mathcal{M}, \boxminus P)$-condition invariant; this condition fails at state q_{12}. In the absence of strong fairness in events σ_{05}, σ_{06}, it is also not $\mathcal{M}$-alive under conditional invariance; this condition fails for the legal state trajectory that stays forever traversing in Loop X formed by the state-transition sequence $q_6 - q_7 - q_8 - q_9 - q_{10} - q_0 - q_3 - q_2 - q_1 - q_6$ (see Fig. 4), satisfying $\Box P$ but violating $\Box\Diamond p_{q_4}$. It also fails for any legal trajectory satisfying $\Box P$ that enters and stays forever traversing in Loop Y formed by the state-transition sequence $q_5 - q_4 - q_3 - q_2 - q_1 - q_5$ (see Fig. 4), violating $\Box\Diamond p_{q_7}$. For $\boxminus P$ to be violating either condition, $\Box P$ is not $\mathcal{M}$-directing, and hence is not $\mathcal{M}$-controllable.

$$K_0 \approx Y_m \cdot \Box P.$$
$$K_1 \approx \Omega(K_0),\ \ \Box(D \cdot P) \overset{G}{\simeq} Y_m \cdot \Box P\ \ \text{(A case DSF-1)},$$

TABLE 1. Example 4: Product components of *D*, the state-forbiddance refinement on *P*.

1.	$\overline{p_{q_{11}}}$
2.	$\ominus\Big(\lim\limits_{x\to\infty}\prod\limits_{n=1}^{x}\ominus^{9(n-1)}\Big(p_{q_1}\cdot\ominus(p_{q_2})\cdot\ominus^{2}(p_{q_3})\cdot\ominus^{3}(p_{q_0})\cdot \ominus^{4}(p_{q_{10}})\cdot\ominus^{5}(p_{q_9})\cdot\ominus^{6}(p_{q_8})\cdot\ominus^{7}(p_{q_7})\cdot\ominus^{8}(p_{q_6})\Big)\Big)\rightarrow\overline{p_{q_6}}$
3.	$\ominus\Big(\lim\limits_{y\to\infty}\prod\limits_{n=1}^{y}\ominus^{5(n-1)}\Big(p_{q_1}\cdot\ominus(p_{q_2})\cdot\ominus^{2}(p_{q_3})\cdot\ominus^{3}(p_{q_4})\cdot\ominus^{4}(p_{q_5})\Big)\Big)\rightarrow\overline{p_{q_5}}$

Note: Evaluating whenever the DES G (in Fig. 4) is previously at state q_1, x, y refer to the maximum **positive** number of consecutive, complete cycles of Loops X, Y, respectively, that the DES is permitted to traverse each time before the respective states q_6, q_5 are forbidden.

where past formula D, constructed by some human ingenuity, is a logical product of three formulas as shown in Table 1.

Now, $G \models (D \cdot P)$ [$\because (\theta \rightarrow (D \cdot P)) \approx$ *true*].

$$\begin{aligned}
\therefore K_1 &\approx Y_m \cdot \sup\mathcal{C}(D \cdot P)\\
&\approx Y_m \cdot \Box(D \cdot P)\\
&\qquad [\because \Box(D \cdot P) \text{ is controllable}]\\
&\approx Y_m \cdot \Box P\\
&\qquad [\text{By Proposition 3: P3.2}]\\
&\approx K_0.\\
\therefore \sup\mathcal{C}(P, \mathcal{M}) &\approx \Box\,(D \cdot P)\,.
\end{aligned}$$

Depicted in Fig. 4, Loops X, Y in the DES G are called $\mathcal{M}$-incomplete loops; in general, an $\mathcal{M}$-incomplete loop is with respect to a state trajectory of a DES that enters and stays traversing therein forever, without meeting at least one marker condition in system marker set $\mathcal{M}$ infinitely often. By their execution to exit $\mathcal{M}$-incomplete loops in a DES, strategically defined fair events play a crucial role in $\mathcal{M}$-controllability that is of pragmatic importance in simplifying supervisor design. Take for instance: If events σ_5, σ_6 were or could be made compassionately fair, the refinement D on P would have been reduced to $\overline{p_{q_{11}}}$, leading to a simpler $\sup\mathcal{C}(P, \mathcal{M})$ as the supremal $\mathcal{M}$-controllable subformula of $\Box P$. But as it is, the absence of fairness in events σ_5, σ_6 (that are controllable) leads to constructing a formula $\sup\mathcal{C}(P, \mathcal{M})$ that is more complex. This is because the exact delimiting safety-closure formula $\Box\,\psi$ for $Y_m \cdot \Box P$ required in the LTL control synthesis also needs to 'emulate' the necessary but missing event fairness. This is done by $\psi \equiv D \cdot P$, with the invariance of the synthesized formula D (see Table 1) constraining the DES further around every existent $\mathcal{M}$-incomplete loop under the invariance of P, essentially specifying a breakout to exit the loop in the temporal loop limit.

Interestingly, with the supremal $\mathcal{M}$-controllable subformula of $\Box P$ existing in the complex form constructed, the realization of a supervisor [5] in practice requires an *a priori* arbitrary setting of x, y in D (explained in the note under Table 1) to possibly different, finite positive numbers. Optimality of supervision, as defined by $\sup\mathcal{C}(P, \mathcal{M})$, becomes purely a theoretical condition because any such supervisor realization is suboptimal; however, the loss of optimality or permissiveness is due, reciprocally, only to each of x, y being finitely set with regard to permitting the maximum number of consecutive cycles the DES can traverse in the respective $\mathcal{M}$-incomplete loops under supervision. The setting of x, y may be made to a finite numerical extent that this loss is deemed immaterial.

VI. DISCUSSION WITH RELATED WORK & BEYOND

A. DES MODELING & A SYNTHESIS ALTERNATIVE

Fair DES model G, by the sets of labeled strings of finite and infinite length arising from the pair $(\mathcal{I}^{\circledast}(G), \mathcal{I}(G))$, is a state augmented version of the 'live' DES model due to [18]. Besides, unlike the latter model, the legal evolution of DES G is explicitly described, over its model structure, by fairness formulas of events in the fair event set $\Sigma_{\mathcal{F}}$. This makes the fair model G conceptually cleaner and more explainable from the design and synthesis perspective – a unique feature that will be elaborated and made clear by the end of Section VI.

The 'live' DES model is adopted in [18], [19] for progressive control that is more arbitrary and realizes an ω-language (i.e., a set of strings of infinite length, as opposed to a language which refers to a set of strings of finite length), for which state-of-the-art algorithms for synthesis of controllers in ω-automata (accepting ω-languages termed regular) are available. One might then suggest that an alternative to this paper is to use propositional LTL – the widely used fragment of LTL that is translatable [20], [21] to ω-automata – as a DES modeling and specification language over a finite state space, and then proceed in principle as follows, to perform controller synthesis that is not syntax-based or in state space:

1) Construct some ω-automaton model for a given fair DES G. This ω-automaton is defined using a deterministic state transition function over the event set Σ (and hence is termed deterministic). Its construction (see [22, Ch. 5: Sec. 5.1.6 & Rem. 5.44]) entails translating from the LTL formula κ_G that axiomatizes G's transition structure (in terms of initial condition and transition relations of events in Σ) and its fair evolution (in terms of fairness formulas of events in the fair event set $\Sigma_{\mathcal{F}}$).
2) Translate a specified LTL formula of finite length (in terms of the number of symbols) into some deterministic ω-automaton.
3) Apply an appropriate ω-automata-based controller synthesis algorithm.

However, current such controller synthesis algorithms, the earliest of which is given in [19], are about deadlock-free or infinite progressive control, not the more structured marker-progressive control that admits and in fact unifies both types of controlled behavior, finite [1] and infinite [18], [19], in a common framework. The opinion held herein is that, from the design perspective, what matters most is having a unified framework for finding an understandable control solution that conforms to correctly stated specifications, not whether the resultant controlled behavior is infinite (or deadlock-free, meaning it has no deadlocks) or otherwise – for, that is a solution output, not a problem input. The term 'deadlock' used in [19] means entering a state that is terminal or made terminal (under control), and carries a negative connotation; this paper is of the opinion that, provided the specification is correctly stated, a terminal state that exists and can be entered in a controlled behavior is part of the solution, such as a graceful system cessation. Insisting on infinite control[1] at the outset as a problem objective might therefore distort the intended control solution. Given a DES that has no terminal state (that it can transit into from an initial state), if one wishes to check if the controlled behavior due to a specification pair $(P, \mathcal{M})$ is infinite, one only needs to check that, for the supremal control solution obtained with ϕ as its kernel, the solution invariant $\boxminus\phi$ is $(\{\mathit{false}\}, \boxminus\phi)$-condition invariant. In the case that the DES under a control solution can enter an unexpected terminal state, the principled approach then is to reflect on and iteratively refine, as needed, the system design and specification.

B. INFINITE CONTROL

Used in the respective frameworks – specifically [5, Def. 20 & Thm. 3] versus [18, Prop. 3.1],[2] [19, Prop. 4.5] – $\mathcal{M}$-directingness [5] is analogous to relative ω-closure [18], [19], in the sense that each concept characterizes the solvability or existence of controls in conjunction with their associated basis concept of controllability. The associated basis concept used in [19] is called ω-controllability; it refines the standard concept of controllability [1], used in [18, Prop. 3.1] in the originating language form [1] and in [5, Def. 20 & Thm. 3] in an LTL form [5], with a characterization in terms of some controllability prefix, and is shown to reduce to the standard concept under relative ω-closure [19, Prop. 4.4] (for a specification given by an ω-sublanguage of the DES). However, in general, the union of relative ω-closed sublanguages (of the DES) is not relative ω-closed, although that of ω-controllable sublanguages is ω-controllable, thus implying that the supremal ω-controllable and relative ω-closed sublanguage does not exist in general [19], [23]. This contrasts with the optimality result for the specification pair $(P, \mathcal{M})$ that the LTL control synthesis method (2) is based on: Inferring from Proposition 1, $\sup\mathcal{C}(P, \mathcal{M})$ exists which the method (2) computes, such that it is the supremal $\mathcal{M}$-controllable subformula of $\Box P$ provided $\sup\mathcal{C}(P, \mathcal{M}) \not\approx \mathit{false}$.

In fact, for the specification pair $(P, \mathcal{M})$ considered, relative ω-closure is technically stronger or more restricting than $\mathcal{M}$-directingness in general; to intuitively explain this technicality in the context of an arbitrary specification pair $(\psi, \mathcal{M})$, where $\psi \Rightarrow P$, relative ω-closure of the DES ω-sublanguage equivalent of the pair $(\psi, \mathcal{M})$ only permits entering any $\mathcal{M}$-incomplete loop that the DES will exit under the invariance of ψ, either by itself or after a finite number of cycles (of the loop) specified by ψ under invariance. This is not the only case permitted with $\mathcal{M}$-directingness of $\Box\psi$; as evident by Proposition 6 and a reading of the relation $\overset{G}{\simeq}$ therein according to definition, the concept also allows the DES to enter and stay in each $\mathcal{M}$-incomplete loop that it may not exit by itself, if not for the arbitrary number of cycles in the temporal loop limit posed by ψ under invariance. This is illustrated in Example 4 by the supremal $\mathcal{M}$-controllable solution $\Box\phi$ obtained for some $\psi \equiv \phi$, where some practical design implications are also discussed, and will be elaborated further below in connection to related work.

Despite the differences, for common problem settings, one may, in practice, apply either the LTL control synthesis method (2) or an appropriate ω-automata-based synthesis algorithm available. Then, where a non-trivial solution exists, the former method yields a (satisfiable) supremal marker-controllable formula. For the ω-automaton equivalent of specification pair $(P, \mathcal{M})$, after limiting it suitably to represent a regular ω-sublanguage that is relative ω-closed, the latter algorithms in [19], [24], [25] each uses the resultant ω-automaton to compute a controllability state subset over a finite state automaton (to be controlled), which it then uses, if the subset is nonempty, to compute the supremal ω-controllable sublanguage [19] as an ω-automaton; as the computed sublanguage is also relative ω-closed [18, Prop. 3.2], it is hence a (specification-conforming) control solution. An algorithm in [23] uses a reactive synthesis approach that also yields an ω-automaton as solution. However, due to the stronger concept of relative ω-closure as discussed above, a stronger solution may result for the latter algorithms, depending on the relative ω-closed sublanguage of the given specification that is selected *a priori*; such a selection may not be straightforward, be it from a regular ω-sublanguage specification not known *a priori* to be ω-closed, or from the nonempty supremal ω-controllable sublanguage first computed that only ascertains the existence of infinite control [19, Thm. 5.3]. In contrast, a supremal LTL control solution obtained may at times turn out to be purely theoretical, in that it is not practically realizable or implementable as a supervisor. Consider such a supremal LTL control solution obtained and discussed in Example 4. Its solution form, however, can serve as a transparent basis for selecting *a posteriori* a suitably permissive but necessarily suboptimal solution for practical supervisor realization, which in this example corresponds to an ω-automaton control solution (whose regular ω-controllable language is relative ω-closed).

[1] Infinite control and finite control are understood to refer to control that effectively realizes an ω-sublanguage and a sublanguage of a DES, respectively.

[2] Restated in [19, Prop. 4.2], where the term 'deadlock-free' is used instead of the term 'nonblocking'.

Clearly, though related, the concepts of $\mathcal{M}$-directingness and relative ω-closure have resulted in different treatments to handling $\mathcal{M}$-incomplete loops under the invariance of P, for control synthesis of the specification pair $(P, \mathcal{M})$ on, respectively, its LTL formula and equivalently translated ω-automaton. Related is the DES dynamics or evolution induced by the fair event set, which provides a more concrete means to explaining DES behavior, especially with regard to whether the DES can guarantee exiting, by itself, any $\mathcal{M}$-incomplete loop it may enter. The basic role of fair events in driving DES behavior, with implications for control synthesis, is, however, abstracted out in the original infinite or ω-language control theory [18], [19].

To aid further comparison, define a (weakly) control-forcible event as one that must eventually occur if it is infinitely often control-enabled and defined in an automaton. Then in [25], a finite state automaton is assumed to satisfy an additional condition called state fairness, asserting that every event is control-forcible. Over an automaton endowed with such control dynamics, the synthesis algorithm in [25] computes a controllability state subset in polynomial instead of exponential time by that in [24]. However, such an approach might inadvertently accommodate what, at the design outset, is infeasible regarding the control dynamics of some events.

C. FINITE CONTROL (IN THE LIMIT)

Last but not least, consider the, perhaps, simplest infinite control problem, where the fair event set is not accounted for, and the DES model and specification are expressible by finite *non-terminal* state automata[3] – accepting the language, say L, hence termed regular, of the DES, and its sublanguage, say E, that is L-closed,[4] respectively – in the limits, or these regular languages in the limits [27, Sec. 6]. The problem instances in Examples 1 to 3 consider DES models and specifications that are, language-wise, correspondingly expressible as such under the condition that the fair event set, which is not empty in the case of Example 3, is not accounted for. This is because, in these examples, the DES G has no terminal state (that it can transit into, starting from an initial state), and the specification given by the pair $(P, \mathcal{M})$, where $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$, is under the following modeling restriction: The j-prefix of every state trajectory $I' \in \mathcal{I}(G)$ satisfying $\Box P$ can be extended to some $I \in \mathcal{I}(G)$, $I_{(j)} = I'_{(j)}$, satisfying $\Box P \cdot \Box\Diamond \left(\prod_{i=1}^{m} M_i\right)$ that is logically stronger than (the LTL formula denoted by) the specification pair $(P, \mathcal{M})$.

The correspondence, however, is only up to those DES modeling and specification in the limits. As explained earlier, the LTL control synthesis method (2) does not specially seek an infinite control solution as that is not an objective of the problem it addresses, unlike the finite state automata-based algorithm in [27]. Besides, like nonblocking (finite) control synthesis [1], [2], the synthesis algorithm in [27] seeks the most 'optimistic' control solution without accounting for the fair event set. This means that, keeping specified safety uncompromised, the infinite control solution sought permits the DES to enter any $\mathcal{M}$-incomplete loop, so long as the DES *can logically* transition out of it – although it *need not* happen at runtime – and proceed onto a state trajectory $I \in \mathcal{I}(G)$ satisfying $\Box P \cdot \Box\Diamond \left(\prod_{i=1}^{m} M_i\right)$. This approach of optimistically admitting $\mathcal{M}$-incomplete loops by the algorithm in [27] produces a generally more permissive solution than that by the method (2), whenever both yield an infinite control solution for the same problem instance. As a result, whether more or as permissive, the former's solution may not always be truly specification-conforming, due to the possibility of the control solution permitting the DES to entering an $\mathcal{M}$-incomplete loop and staying in there forever.

As it turns out, if the nonblocking synthesis algorithm [2] yields a control solution that does not render terminal any state which it permits the DES to reach, then the problem instance addressed is equivalent to that which the algorithm in [27] can address with the same solution. Such is indeed the case for Examples 1 to 3 that originated in [2]. The reader may therefore treat the control solutions presented in [2] for these examples as those yielded by the algorithm in [27].

With that said, for each of the Examples 1 to 3, it is coincidental that the LTL control synthesis method (2) and the finite state automata-based algorithm in [27] yield, automaton-wise, the same supremal control solution, with the latter yielding the so-called complete or live supremal controllable ω-sublanguage as a finite (non-terminal) state automaton in the limit. On why the solutions coincide, firstly, applying either the method (2) or the algorithm in [27], supremal *infinite control* solutions exist. Secondly, with the aid of, automaton-wise, the same control solutions presented as automata in [2], it can be observed that, for Example 1 (see [2, Sec. 7.1: On p. 650]) and Example 2 (see [2, Sec. 7.2: On p. 653]), there is no $\mathcal{M}$-incomplete loop to be admitted by the control solution; and for Example 3 (see [2, Sec. 7.3: Fig. 7.5]), although there is one $\mathcal{M}$-incomplete loop, formed by transitions of only event c_7 due to the model CAT (see Fig. 3) which the solution by the algorithm in [27] admits, the LTL control solution synthesized permits the DES to also enter this loop, since the fair event $c_2 \in \Sigma_C$ is there to help guarantee exiting this loop.

D. MARKER-PROGRESSIVE CONTROL & BEYOND

In an overall remark, the preceding discussions under common problem settings should be regarded as reconciliatory. After all, though technically related, LTL and ω-languages are different formalisms with their own independent analysis frameworks, and so are the problem settings and objectives in general, as formulated in these frameworks. In the former is about (state-based) logic control of fair DES's while in

[3] In this paper, an automaton [1], [26] is conveniently referred to as non-terminal state if every state of the automaton that can be reached from its initial state is not terminal; the term 'terminal' is in the sense defined for the DES model G of this paper.

[4] To recall from [27], the prefix closure of a language E, denoted by $\overline{E}$, is $\overline{E} = \{s' \mid (\exists s, s't \in E)s't = s\}$; and E is L-closed if $E = \overline{E} \cap L$.

the latter is about (event-based) ω-language control of 'live' DES's.

1) ON SPECIFICATION CORRECTNESS GUARANTEE

That said, the discussion on the concept of $\mathcal{M}$-directingness has exposed the necessity to exit or 'break out' of $\mathcal{M}$-incomplete loops to guarantee specification correctness, meaning, to ensure that a controlled DES can never trace out a state trajectory in runtime that violates the specification it is designed for. This necessity was previously hidden under the concept of relative ω-closure for infinite controlled behavior [18], [19]. Formulated in a unified behavioral framework, this necessity carries over to finite controlled behavior [1], unearthing a fundamental insight about specification-correctness guarantee. Now, it is hitherto almost conventional thinking that a controlled DES meeting a specification (in terms of language containment) provides specification-correctness guarantee. But a DES in runtime is 'generative' of strings (of events) [1], not all of which are 'accepting' in the standard automata-theoretic sense [26]. The unintuitive insight unearthed is this: In synthesizing finite controlled behavior, it is necessary but not sufficient in general to guarantee specification correctness, if a controlled DES meeting a language specification is achieved by optimal control synthesis [2] in finite state automata [26] based on the founding nonblocking control theory [1], or by that slightly modified based on related work [27], with nonblocking control understood as always permitting a DES to transition and enter or reenter a state of the marker state set that is a special case of the system marker set (see [5, $\mathcal{M}$ (2)]). This guarantee insufficiency is due to the optimal control synthesis [2], [27] being carried out optimistically without modeling and accounting for DES fairness dynamics, thus possibly allowing $\mathcal{M}$-incomplete loops that exist in the DES with no loop exit assurance to be admitted. Note then, that, for Example 3, the LTL control solution yielded by the synthesis method (2) guarantees specification correctness with the cooperation of fair event $c_2 \in \Sigma_{\mathcal{C}}$. But the same cannot be said of the automaton solution [2, Sec. 7.3: Fig. 7.5] yielded by the synthesis algorithm in [2] or [27]; as discussed earlier, this solution is, automaton-wise, no different from the corresponding LTL control solution, but it is the same solution the synthesis algorithms [2], [27] yield regardless of whether event c_2 is a fair event of the compassionate type or not.

Clearly, the cause of not guaranteeing specification correctness by nonblocking control synthesis [2] is its synthesized control solutions in finite state automata [26] possibly admitting $\mathcal{M}$-incomplete loops, with no exit assurance of such loops. This cause may have ramifications for the nonblocking control framework [1], [2] and its subsequent developments (e.g., [3], [8], [27]) to-date.

2) SUPERVISORY CONTROL UNIFIED IN TRANSPARENT LTL SYNTHESIS

Besides bringing into focus the issue of specification-correctness guarantee in the cited literature on finite controlled behavior, a more obvious and undeniable fact is that all the aforementioned research efforts are based on formal languages and automata which are rather elementary. Inherent therefore is formal language control giving little regard to human designer considerations, fundamental of which are clearer system dynamics modeling in terms of evolution characterizable by fair events, and transparency of synthesis along with solution readability in LTL, all of which are desirable in engendering a higher level of explainability that might, for instance, help a human designer decide if a solution is right, and not just getting it right. By applying the mathematical method (2), these fundamental considerations are fulfillable for marker-progressive control synthesis of fair DES's. As illustrated by examples, the synthesis transparency of this method is enhanced by an equational style of logic reasoning it supports, referred to as calculational logic [28], in the readable syntax of LTL that applies LTL syntactic logic rules [6] in an algebraic style, manipulating and presenting LTL formulas in 'a sequence of substitutions of equals for equals'. This reasoning style is akin to calculations in many fields of mathematics, including linear algebra, modern algebra, and calculus; its use motivates formal workings that human analysts can freely make in their own line of logic reasoning as long as it is correct, rendering the mathematical problem solving of control synthesis more transparent in general.

As carefully illustrated by examples, applying the LTL control synthesis method (2) currently involves working out by hand and, in some cases, it requires human ingenuity. But the process is demonstrably transparent and the solution obtained is a readable LTL formula. These, along with explicitly engaging the role of fair events for guaranteed marker-liveness, are needed for a more holistic mathematical treatment of understandable control design in general, setting this paper uniquely apart.

Admittedly, the method (2) is presently not computer-algorithmic, as its current development is not primarily motivated by problem solving using algorithms and software tools. But in some respects, it has conceptually unified and extended the aforementioned research efforts under canonical LTL for the specification pair $(P, \mathcal{M})$ considered. This specification pair may seem limited to some, but in the generic realm of supervisory control – largely of accomplishing tasks regularly without violating safety – it is about the most general that a control designer can practically think of.

3) EXTENSION TO GENERAL-PROGRESSIVE SUPERVISORY CONTROL

That said, to handle specification beyond the specification pair $(P, \mathcal{M})$, let $\mathcal{N} = \{N_1, N_2, \cdots, N_m\}$ be some system asymptote set, where each $N_i \in \mathcal{N}$ $(1 \leq i \leq m)$ is an arbitrary past formula specifying a system asymptote condition. That each asymptote condition $N_i \in \mathcal{N}$ is to be stable, in the sense of being eventually met and maintained henceforth in DES G, is specified by $\Diamond\Box N_i$, an LTL formula in canonical temporal-persistence form [6], [9]. Then the LTL control foundation

[5] and associated synthesis results of this paper are directly extendible from the specification pair $(P, \mathcal{M})$, to the pair $(P, \mathcal{M} \times \mathcal{N})$ denoting the LTL formula $\Box P \cdot Z_m$, where

$$Z_m \equiv \prod_{i=1}^{m} (\Box\Diamond M_i + \Diamond\Box N_i)$$

is an LTL formula of the canonical reactivity class – the most general class situated topmost in the complete hierarchy of canonical classes of LTL formulas [6], [9]; the formula Z_m may be called the most general reactivity formula of $\mathcal{M} \times \mathcal{N}$-rank m. In essence, this extension, to general-progressive supervisory control that covers the full specification hierarchy of canonical LTL, is anchored on the following two main concept generalizations:

The first is generalizing $\mathcal{M}$-directingness of $\Box P$ to $\mathcal{M} \times \mathcal{N}$-directingness, such that $\Box P$ is said to be $\mathcal{M} \times \mathcal{N}$-directing if $\boxminus P$ is initially satisfied, $(\mathcal{M} \times \mathcal{N}, \boxminus P)$-condition invariant, i.e.,

$$G \models \Box \left(\ominus_x \left(\boxminus P, \boxminus P \cdot \sum_{i=1}^{m} \overline{M_i + N_i} \right) \rightarrow \boxminus P \right),$$

and $\mathcal{M} \times \mathcal{N}$-alive under conditional invariance, i.e.,

$$G \models \Box P \rightarrow Z_m.$$

Being $(\mathcal{M} \times \mathcal{N}, \boxminus P)$-condition invariant and $\mathcal{M} \times \mathcal{N}$-alive under conditional invariance are direct extensions of being $(\mathcal{M}, \boxminus P)$-condition invariant and $\mathcal{M}$-alive under conditional invariance, respectively.

The second is generalizing $\mathcal{M}$-controllability of $\Box P$ to $\mathcal{M} \times \mathcal{N}$-controllability, such that $\Box P$ is said to be $\mathcal{M} \times \mathcal{N}$-controllable if $\Box P$ is controllable and $\mathcal{M} \times \mathcal{N}$-directing.

Note that the extension to general-progressive control is prescriptively a simple scale-up, in that the results for $(P, \mathcal{M} \times \mathcal{N})$ are obtainable from those for $(P, \mathcal{M})$ as presented in this paper and its predecessor [5], by replacing Y_m with Z_m, and every $\mathcal{M}$-related concept with the corresponding $(\mathcal{M} \times \mathcal{N})$-related concept defined above.

Note also that

$$(\Box\Diamond M_i + \Diamond\Box N_i) \equiv (\Box\Diamond\overline{N_i} \rightarrow \Box\Diamond M_i).$$

One may therefore think of general-progressive supervisory control as temporal-safety or invariance control of P to bring about marker-progressive (or $\mathcal{M}$-progressive) responses to asymptotic instability (or $\mathcal{N}$-instability) triggers. Interestingly, by interpreting the specified triggers as arising conditionally from a system operating environment, it becomes conceptually clearer that DES G is a behavioral model of a *system in an environment*.

Finally, in the special case of finite state DES G and specification pair $(P, \mathcal{M} \times \mathcal{N})$, where $P \equiv true$, $\mathcal{M} \times \mathcal{N}$-rank $m = 1$, and $\mathcal{M}$, $\mathcal{N}$ are sets of state formulas, the general-progressive control problem reduces to a version of the reactive synthesis problem [29], [30] that is extended to admit uncontrollable events [23], and unified to subsume finite reactive behavior [31]. This resultant reactive synthesis problem may be efficiently solved by adapting a two-player game approach [29], [32]. However, it entails a formal investigation and is, in any case, beyond the scope of this paper.

4) TOWARDS COMPUTER-ALGORITHMIC SYNTHESIS IN FUTURE WORK

Finally, the research direction pursued in this paper and its predecessor [5] is relatively new; not unexpectedly, this paper lags the related literature in algorithmic complexity studies, among others. Certainly, making the LTL control synthesis method (2) computer-algorithmic (or automated) – fully or partially, to help construct supremal marker-controllable formulas which are *satisfiable*, is a challenging subject for future research; so is making the method's generalization to specification pair $(P, \mathcal{M} \times \mathcal{N})$ automated. Already, the complexity of satisfiability of propositional LTL in finite state DES's is PSPACE-complete, as inferred from [14]. Synthesis complexity is thus expected to be analyzed only for interesting problem subclasses of practical interest in a future comparative study with existing related research. The synthesis efforts include deriving axiomatic modeling constraints due to the natural dynamics and structure of a given DES for logic simplification purposes, and constructing the exact delimiting safety-closure formula in compact form for a given specification pair.

VII. CONCLUSION

This paper has presented an existence characterization of supremal marker-controllable safety formula for fair DES's. This new LTL characterization result of Corollary 1 should be of theoretical interest, and lends itself to an algebraic syntax-based framework for transparent control synthesis in the 'regular' case of finite state DES's and specification pairs given by decidable LTL formulas. Future research of interest includes finding subclasses of infinite state DES's for which $\{K_j\}$ (2) converges, extending the main synthesis results of Theorems 1 and 2.

In conclusion, DES's and their controls span a wide range of modern engineering systems that are human-designed. It is thus advantageous to develop a mathematical control-theoretic framework supporting human readable specification and transparent control synthesis in the same natural-language motivated algebra. Together with the predecessor paper [5], this paper is a step in this direction, for fair DES's and marker-progressive control readily extendible to general-progressive control in the algebra of canonical LTL [6]. Importantly, this line of research has given new insights into supervisory control, and provided the basis for further progress in the field.

Looking ahead, on the horizons are no doubt new unique opportunities in DES control theory research. Anchoring on the well-organized hierarchy of canonical classes of LTL formulas [6], [9], the goal (and hope) is to continue the high-level and more structured control-theoretic development in future research endeavors, bringing new control-theoretic findings on board in the process.

REFERENCES

[1] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jul. 1987.

[2] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 637–659, 1987.

[3] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 3rd ed. Cham, Switzerland: Springer, 2021.

[4] U.-H. Kim and J.-H. Kim, "A stabilized feedback episodic memory (SF-EM) and home service provision framework for robot and IoT collaboration," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2110–2123, May 2020.

[5] K. T. Seow, "Supervisory control of fair discrete-event systems: A canonical temporal logic foundation," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5269–5282, Nov. 2021.

[6] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems: Specification*. New York, NY, USA: Springer-Verlag, 1992.

[7] M. H. de Queiroz, J. E. R. Cury, and W. M. Wonham, "Multitasking supervisory control of discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 15, pp. 375–395, Oct. 2005.

[8] W. M. Wonham and K. Cai, *Supervisory Control of Discrete-Event Systems*, A. Isidori, J. H. van Schuppen, E. D. Sontag, and M. Krstic, Eds. Cham, Switzerland: Springer, 2019.

[9] Z. Manna and A. Pnueli, "Completing the temporal picture," *Theor. Comput. Sci.*, vol. 83, no. 1, pp. 97–130, Jun. 1991.

[10] B. Jonsson and T. Yih-Kuen, "Assumption/guarantee specifications in linear-time temporal logic," *Theor. Comput. Sci.*, vol. 167, nos. 1–2, pp. 47–72, 1996.

[11] G. Petric Maretić, M. Torabi Dashti, and D. Basin, "LTL is closed under topological closure," *Inf. Process. Lett.*, vol. 114, no. 8, pp. 408–413, Aug. 2014.

[12] N. Piterman and A. Pnueli, "Temporal logic and fair discrete systems," in *Handbook of Model Checking*, E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, Eds. Cham, Switzerland: Springer, 2018, pp. 27–73.

[13] I. Hodkinson, F. Wolter, and M. Zakharyaschev, "Decidable fragments of first-order temporal logics," *Ann. Pure Appl. Log.*, vol. 106, nos. 1–3, pp. 85–134, Dec. 2000. [Online]. Available: https://www.sciencedirect.com/journal/annals-of-pure-and-applied-logic/vol/106/issue/1

[14] A. P. Sistla and E. M. Clarke, "The complexity of propositional linear temporal logics," *J. ACM*, vol. 32, no. 3, pp. 733–749, Jul. 1985.

[15] I. Hodkinson, "Monodic packed fragment with equality is decidable," *Studia Logica*, vol. 72, no. 2, pp. 185–197, 2002.

[16] K. T. Seow, "Syntax-based synthesis for temporal-safety supervision," *Automatica*, vol. 41, no. 11, pp. 1965–1972, Nov. 2005.

[17] K. T. Seow and R. Devanathan, "A temporal logic approach to discrete event control for the safety canonical class," *Syst. Control Lett.*, vol. 28, no. 4, pp. 205–217, Aug. 1996.

[18] P. J. Ramadge, "Some tractable supervisory control problems for discrete-event systems modeled by Büchi automata," *IEEE Trans. Autom. Control*, vol. 34, no. 1, pp. 10–19, Jan. 1989.

[19] J. G. Thistle and W. M. Wonham, "Supervision of infinite behavior of discrete-event systems," *SIAM J. Control Optim.*, vol. 32, no. 4, pp. 1098–1113, Jul. 1994.

[20] D. Gabbay, "The declarative past and imperative future: Executable temporal logic for interactive systems," in *Temporal Logic in Specification* (Lecture Notes in Computer Science), vol. 398, B. Banieqbal, H. Barringer, and A. Pnueli, Eds. Berlin, Germany: Springer-Verlag, 1989, pp. 409–448.

[21] J. Esparza, J. Křetínský, and S. Sickert, "A unified translation of linear temporal logic to ω-automata," *J. ACM*, vol. 67, no. 6, p. 33, 2020.

[22] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.

[23] A.-K. Schmuck, T. Moor, and R. Majumdar, "On the relation between reactive synthesis and supervisory control of non-terminating processes," *Discrete Event Dyn. Syst.*, vol. 30, no. 1, pp. 81–124, Mar. 2020.

[24] J. G. Thistle, "On control of systems modelled as deterministic rabin automata," *Discrete Event Dyn. Systems: Theory Appl.*, vol. 5, no. 4, pp. 357–381, Sep. 1995.

[25] J. G. Thistle and R. P. Malhamé, "Control of ω-automata under state fairness assumptions," *Syst. Control Lett.*, vol. 33, no. 4, pp. 265–274, Apr. 1998.

[26] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Reading, MA, USA: Addison-Wesley, 1979.

[27] T. Moor, C. Baier, T.-S. Yoo, F. Lin, and S. Lafortune, "On the computation of supremal sublanguages relevant to supervisory control," in *Proc. 11th Int. Workshop Discrete Event Syst.*, Guadalajara, Mexico, Oct. 2012, pp. 175–180.

[28] D. Gries and F. B. Schneider, *A Logical Approach to Discrete Math*. Berlin, Germany: Springer-Verlag, 1993. [Online]. Available: https://www.cs.cornell.edu/gries/Logic/intro.html

[29] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Saar, "Synthesis of reactive(1) designs," *J. Comput. Syst. Sci.*, vol. 78, no. 3, pp. 911–938, May 2012.

[30] R. Bloem, R. Ehlers, S. Jacobs, and R. Könighofer, "How to handle assumptions in synthesis," *Proc. Electron. Theor. Comput. Sci., 3rd Workshop Synth. (SYNT)*, Vienna, Austria, vol. 157, Jul. 2014, pp. 34–50.

[31] R. Ehlers, S. Lafortune, S. Tripakis, and M. Y. Vardi, "Supervisory control and reactive synthesis: A comparative introduction," *Discrete Event Dyn. Syst.*, vol. 27, no. 2, pp. 209–260, Jun. 2017.

[32] R. Majumdar, N. Piterman, and A.-K. Schmuck, "Environmentally-friendly GR(1) synthesis," in *Tools and Algorithms for the Construction and Analysis of Systems* (Lecture Notes in Computer Science), vol. 11428, T. Vojnar and L. Zhang, Eds. Cham, Switzerland: Springer, 2019, pp. 229–246.

KIAM TIAN SEOW (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from the National University of Singapore, Singapore, in 1990, and the M.Eng. and Ph.D. degrees in electrical and computer engineering from Nanyang Technological University (NTU), Singapore, in 1993 and 1998, respectively.

He was a Full-Time Faculty Member with the School of Computer Science and Engineering, NTU, from 2003 to 2014, where he was an Adjunct Associate Professor, from 2014 to 2016. He has held visiting research appointments with the Systems Control Group, University of Toronto, Toronto, ON, Canada, in 1997; the School of Electrical Engineering, KAIST, Daejeon, South Korea, in 2002; the Nippon Telegraph and Telephone Corporation (NTT) Communication Science Laboratories, Kyoto, Japan, in 2003; and the Institute of Information Science, Academia Sinica, Taipei, Taiwan, in 2005. Since 2014, he has been a Visiting Professor with the Robot Intelligence Technology Laboratory, KAIST. He has authored or coauthored over 60 articles in refereed journals and conference proceedings, and coauthored the monograph "*Soccer Robotics*" (Heidelberg: STAR Series, Springer Verlag, 2004). His current research interests include modeling, control design, and applications of discrete-event, and agent systems.

Dr. Seow was an Associate Editor of the IEEE Transactions on Automation Science And Engineering, from 2009 to 2013, and the IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans, from 2010 to 2012. He has been an Associate Editor of the IEEE Transactions on Systems, Man and Cybernetics: Systems, since 2013.

• • •