Nested Supervisory Control of State-Tree Structures (Technical Report)

Xi Wang and Zhiwu Li

Abstract

Generally, complex dynamic systems can be hierarchically abstracted to be superstates with layered internal structures. With the state explosion problem managed, state-tree structures (STS) are a powerful framework to model such systems in a compact and natural way. This study presents an approach to decompose an STS into a set of STS nests (the largest flat fragments) automatically. Each STS nest tracks the system dynamics partially on a level of hierarchy. The communication among STS nests is investigated, which guarantees that the system dynamics in a lower-level STS nest will not block the adjacent higher-level STS nest's dynamics. A top-down iteration approach is presented to synthesize the optimal behavior of STS nests. Finally, given an STS, without tracking its global dynamics to synthesize its global optimal behavior, a nested optimal nonblocking supervisor is obtained. The computational complexity of the synthesis process is reduced from exponential to additive costs with respect to the numbers in the binary decision diagram used for encoding STS nests symbolically.

Index Terms

Hierarchical discrete-event system, state-tree structure, self-similarity structure, symbolic computation, nested supervisory control.

I. Introduction

Hierarchical finite state machines (HFSM) [1]–[3] are defined as finite state machines (FSM) [4], [5] with multi-levels. As stated in [2] and [6], the main feature of an HFSM is that its state space contains *superstates*. A superstate in an HFSM represents the *abstraction* of lower-level HFSM (or FSM). The linguistic studies on HFSM can be found in [6]–[14], in which programming languages *Argos* [13] and *Heptagon* [14] (for reactive systems) feature hierarchical state structures, similar to state-tree structures (STS), that can be compiled into symbolic equations.

By following a bottom-up approach, hierarchical supervisory control is first proposed in [15] as a direct derviation of hierarchical consistency. According to [16], this research topic is mainly covered by four approaches: standard bottom-up design [13]–[15], [17]–[21], top-down design [2], [3], [12], [17], [22]–[24], state aggregation (similar to bottom-up) design [25]–[27], and interface-based design with different levels strictly decoupled (the higher-level and lower-level hierarchies share interface events only) [28]–[30].

A. State-Tree Structures (STS)

Influenced by [31] and [32], the top-down design of an HFSM is first developed in [33]. Thereafter, STS are proposed in [23] and [24] to model complex hierarchical dynamic systems in a compact and natural structure. By encoding the global dynamics of an STS into a predicate, its optimal behavior is synthesized by state feedback control (SFBC). Based on the powerful computational representation of *binary decision diagrams* (BDD) [34], the notorious state explosion problem faced by supervisory control theory is tactfully managed.

Several theoretical extensions of the supervisory control of STS and related applications have been made. The modular supervisory control of an STS is studied in [35]. By viewing a plant to be controlled as comprised of independent asynchronous agents, supervisor localization based on STS is proposed in [36] to calculate the controller of a controllable event by considering an agent's neighborhood information only. The research in [37] studies the symmetry of STS with parallel components. The supervisory control of STS with partial observation is investigated in [38] and [39]. In [40], the supervisory control of STS with conditional-preemption matrices is proposed. A matrix is considered as a specification describing the preemption relations among events. In [41], the supervisory control of STS is used for finding out the safe execution sequences of real-time systems with both conditional-preemption priority and dynamic priority specifications.

This study presents a method to decompose an STS into a set of STS nests (the largest flat fragments) automatically. Suppose that two STS nests are on two adjacent levels in a hierarchical structure of a system, respectively. From the perspective of the lower-level STS nest, the higher-level STS nest is viewed as its *exosystem* (outside world) [43]–[45]; on the contrary, the lower-level STS nest is viewed (abstracted) as a *simple* (*regular*) state in its exosystem's state-space.

B. Nested Supervisory Control of STS

Given an HFSM modelled by an STS with specifications fully cover the *event occurrence prevention problem* and *mutual exclusion problem* [23], [24], this study presents an approach to decompose the STS into multiple hierarchical subordinates (if any) with their specifications assigned properly. Thereafter, a top-down iteration approach is presented to implement the developed nested supervisory control such that the closed-loop behavior of the STS is minimally restrictive. The main contributions of this study are stated as follows:

1) Communication verification of STS nests: Without building the *monolithic* (*global*) *transition structure* of an STS, its *nested transitions* with *multiple-input multiple-output* (MIMO) are developed. For any STS nest, a property namely *communication* is given, which requires that all the *exits* of an STS nest should be accessed by the *paths* starting from any *entrance*. This guarantees that a lower-level nest will never block the system behavior of the subordinating higher-level STS nest.

2) Performance extension: A main feature of STS that makes it neat, compact, and nature, is *boundary consistency* [23], [24], i.e., "*plugging*" a lower-level structure into a leaf state of a higher-level structure does not change their input/output transitions. In this study, the nested supervisory control of STS is addressed in the boundary consistency of STS, i.e., the lower-level closed-loop (under control) STS nests can be "plugged" into the leaf states of a higher-level STS nest without changing their control logics.

3) Nested supervisory control: Given any STS, its nested optimal nonblocking supervisor is synthesized by a top-down iteration approach. Eventually, the optimal behavior of the monolithic STS is obtained. Without tracking the monolithic (global) system behavior of the STS, the computational complexity of the synthesis process is reduced from exponential (for supervisory control) to additive (for nested supervisory control) costs in the numbers of BDD nodes used to synthesize all STS nests' optimal closed-loop behavior. Finally, the control functions (control logic) for controllable events can be implemented to the global STS directly without any change.

4) Finally, we find that the dependence relation among STS nests falls into the application sphere of internal model principle (IMP) of control theory [43]–[45]. We prove that the nested supervisory control of STS satisfies IMP in two-fold significance:

- the STS nests in a closed-loop communicating STS (under control) satisfy the IMP-like property; and

- a closed-loop communicating STS satisfies IMP, if we consider all of its STS nests as the exosystems of an STS.

If an STS is communicated and its specifications can be partitioned properly, its nested optimal nonblocking supervisor is synthesized by following the top-down approach developed in Section VI. Otherwise, users need to remodel the STS or reassign its specifications properly. In the worst case, an STS without hierarchical subordinates cannot be decoupled, which is viewed as a singleton STS nest.

C. *Outline of This Study*

The rest of this paper is organized as follows. Section II presents the STS terminology used throughout the paper. The nested structure of STS is studied in Section III. The nested transitions and communication of STS nests are discussed in Section IV. In Section V, the nested transitions are encoded into predicates; thereafter, the communication of STS nests is verified. The nested supervisory control of STS is presented in Section VI. Specification management and controller implementations for STS are studied in Section VII. Two case studies are presented in Section VIII to demonstrate the nested supervisory control of STS. Nested supervisory control of STS satisfying IMP is discussed in Section IX. Finally, conclusions and future work are presented in Section X.

II. STS Preliminaries

Similar to the hierarchical organizations in the real world, state-tree structures (STS) are proposed in [22] for the purpose of incorporating the *hierarchy* and *concurrency structures* of complex DES into a compact and natural model. Thereafter, it is completed in [23] and [24]. An STS is viewed as a hierarchical finite state machines (HFSM) [1]–[3], i.e., a set of DES with multiple-levels. In this report, we introduce STS by starting from *superstates* defined in statecharts [32]. A superstate, similar to a hierarchical organization or hierarchy, is generally made of several subordinates that may also be hierarchical organizations.

A. *Superstates*

A *superstate* of a system is an *aggregation* (or *abstraction*) of its components [23], [32]. Let X be a finite collection of sets that are called *states* of a system. Given a state $x \in X$ and a non-empty set

$$Y = \{x_1,\, x_2,\, \ldots,\, x_n\} \subsetneqq X$$

with $x \notin Y$, i.e., Y is a proper subset of X that does not contain x, as stated below, x is said to be a *superstate* in X *expanded* by Y if x can be obtained by one of the two expansions.

- OR expansion: x is the *disjoint union* of states in Y, i.e.,

  $$x = \dot{\bigcup}_{x_i \in Y} x_i.$$

  In this case, x is called an OR superstate of X and x_i is called an OR-*component* of $x \in X$. Disjointness means that the semantics of x is the *exclusive-or* of x_i, i.e., a system at state x implies that it is at exactly one state of Y.
- AND expansion: x is the *Cartesian product* of states in Y, i.e.,

  $$x = (x_1, x_2, \ldots, x_n).$$

  For simplification, write

  $$x = \prod_{x_i \in Y} x_i$$

  or

  $$x = x_1 \times x_2 \times \ldots \times x_n.$$

  In this case, x is called an AND superstate and x_i $(i \in [1, n])$ is called an AND-*component* of $x \in X$. The semantics of an AND superstate x means that a system at state x is at all the states of Y simultaneously.

Otherwise, $x \in X$ is a said to be a *simple* state, denoted by SIM, if there does not exist a non-empty set $Y = \{x_1,\, x_2,\, \ldots,\, x_n\} \subsetneqq X$ that expands x.

Formally, given a state set X, the *type function*

$$\mathcal{T} : X \to \{\mathsf{AND},\ \mathsf{OR},\ \mathsf{SIM}\}$$

and *expansion function*

$$\mathcal{E} : X \to 2^X$$

are defined by

$$\mathcal{T}(x) := \begin{cases} \mathsf{AND}, & \text{if } x \text{ is an } \mathsf{AND} \text{ superstate} \\ \mathsf{OR}, & \text{if } x \text{ is an } \mathsf{OR} \text{ superstate} \\ \mathsf{SIM}, & \text{otherwise} \end{cases},$$

and with $x \in X$, $\emptyset \subset Y \subsetneqq X$, and $x \notin Y$,

$$\mathcal{E}(x) := \begin{cases} Y, & \text{if } \mathcal{T}(x) \in \{\mathsf{AND},\ \mathsf{OR}\} \\ \emptyset, & \text{if } \mathcal{T}(x) = \mathsf{SIM} \end{cases},$$

that is, for $x \in X$ with $\mathcal{T}(x) \neq \mathsf{SIM}$, there exists a set $Y \subsetneqq X$ such that $\mathcal{E}(x) = Y$; for $x \in X$ with $\mathcal{T}(x) = \mathsf{SIM}$, $\mathcal{E}(x) = \emptyset$.

Intuitively, a simple state has no children. An OR superstate has several children, and the system is only allowed to stay at exactly one child at a time. An AND superstate also has several children, but the system must stay at all of its children simultaneously.

Example.

Consider the diagram depicted in Fig. 1. We have a state collection $X = \{A, a, b, c, a_1, a_2\}$, in which

- State A is an OR superstate expanded by states a, b, and c;
- State a is an AND superstate expanded by states a_1, and a_2; and
- States b and c are two simple states without children.

In Fig. 1, a superstate is represented by a box and a simple state is depicted by a circle. Generally, the components of a superstate are on the adjacent lower-level. As shown in Fig. 1, superstate A is expanded by three states a, b, and c and x_2, i.e., $\mathcal{E}(A) = \{a, b, c\}$, in which the AND superstate a is further expanded by two OR superstates a_1 and a_2, i.e., $\mathcal{E}(a) = \{a_1, a_2\}$. As structured in Fig. 1, the dashed line between the two boxes labelled with a_1 and a_2 represents that they are the expansions of superstate a. Based on a top-down modelling approach, the expansions of superstates are built inside the boxes iteratively. The state set X is continually growing during the modelling of an STS. We require that any state in X only appears once.

Clearly, from the perspective of superstate A, the system must be at exactly one state of a, b, or c; and from the perspective of superstate a, the system must be at states a_1 and a_2 simultaneously. The latter is consistent with synchronous product defined in DES. Holons defined below describe the internal structures of a_1 and a_2. $\square$

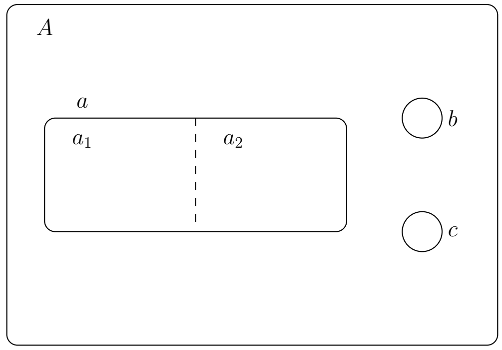


Fig. 1: States in statecharts.

After building the local transitions among the OR components, *holons* [23], [24] are created. Automatically, a set of superstates (or holons) structured in this way is nested.

B. Holons

Both the hierarchy and horizontal transition relations of an STS are described in a family of holons. A holon consists of an internal structure and a (possibly empty) external structure. The internal structure of a holon matches an OR superstate x, and the internal state set X_I^x of H^x is equal to the expansion of superstate x. Formally, $\mathcal{E}(x) = X_I^x$ is true.

Holons are with internal and external structures. The external structure is defined in the adjacent higher level to build transitions with other states. Hierarchically, a holon H is defined as a five-tuple

$$H := (X, \Sigma, \delta, X_0, X_m)$$

where

- X is the nonempty state set, structured as the disjoint union of the (possibly empty) external state set X_E and the nonempty internal state set X_I, i.e.,

$$X = X_E \dot{\cup} X_I;$$

- Σ is the event set, structured as the disjoint union of the boundary event set Σ_B and the internal event set Σ_I, i.e.,

$$\Sigma = \Sigma_B \dot{\cup} \Sigma_I;$$

- The transition structure

$$\delta : X \times \Sigma \to X$$

is a partial function. Write $\delta(x, \sigma)!$ if $\delta(x, \sigma)$ is defined. δ is the disjoint union of two transition structures, the internal transition structure $\delta_I : X_I \times \Sigma_I \to X_I$ and the boundary transition structure δ_B which is again the disjoint union of two transition structures:

$$\delta_{BI} : X_E \times \Sigma_B \to X_I$$

(*incoming boundary transitions*) and

$$\delta_{BO} : X_I \times \Sigma_B \to X_E$$

(*outgoing boundary transitions*).

- $X_0 \subseteq X_I$ is the initial state set, where X_0 has exactly the target states of incoming boundary transitions if δ_{BI} is defined. Otherwise X_0 is a nonempty subset of X_I selected according to convenience.
- $X_m \subseteq X_I$ is the terminal state set, where X_m has exactly the source states of the outgoing boundary transitions if δ_{BO} is defined. Otherwise X_m is a selected nonempty subset of X_I.

A set of holons is denoted by $\mathcal{H}$. For a holon H, its event set Σ can also be partitioned to be the disjoint union of *controllable events* Σ_c and *uncontrollable events* Σ_u by users, i.e.,

$$\Sigma = \Sigma_c \dot{\cup} \Sigma_u.$$

A holon with an empty external structure is identical with a DES proposed in [4].

Example.

Given an HFSM $\mathbf{G}^T$ as the synchronous product of an HFSM x and an FSM y, which can be viewed as three superstates structured in Fig. 2. Superstate T is an AND superstate and it is expanded by two superstates x and y. Suppose that the inner behavior of superstates x and y are identical with two DES generators $\mathbf{G}_x$ and $\mathbf{G}_y$, as depicted in Fig. 3. In Fig. 3(a) the superstate x_1 marked in blue. As a consequence, x and y are OR superstates, and HFSM $\mathbf{G}^T$ is reformed as the two holons H^x and H^y illustrated in Fig. 4. In Particular, we consider that $\mathbf{G}_x$ shown in Fig. 3(a) (identical with holon H^x in Fig. 4) is hierarchical.

On the one hand, suppose that superstate x_1 is an OR superstate, and its internal behavior is depicted by a holon H^{x_1} shown in Fig. 5. Finally, by plugging H^{x_1} into superstate x_1 in Fig. 4, we obtain the monolithic dynamic structure of $\mathbf{G}^T$ illustrated in Fig. 6. The set of holon describe the dynamic of $\mathbf{G}^T$ is denoted by $\mathcal{H}^T = \{H^x, H^y, H^{x_1}\}$. Holon H^{x_1} shown in Fig. 5 is with

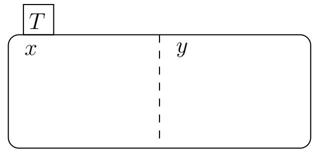
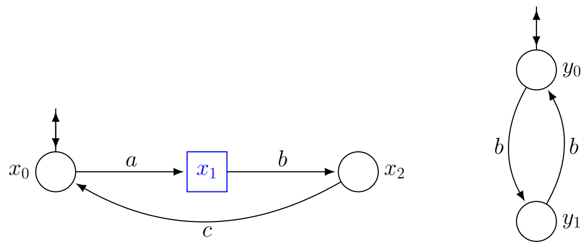


Fig. 2: Three superstates.



(a) $\mathbf{G}_x$.

(b) $\mathbf{G}_y$.

Fig. 3: Two DES generators.

internal structure and external structures, i.e.,

- X^{x_1} is a nonempty state set structured as the disjoint union of the external state set $X_E^{x_1} = \{x_0, x_2\}$, and $X_I^{x_1} = \{0, 1, 2, 3, 4\}$. Formally, $X^{x_1} = X_E^{x_1} \cup X_I^{x_1} = \{x_0, x_2, 0, 1, 2, 3, 4\}$ and $X_E^{x_1} \cap X_I^{x_1} = \emptyset$;
- Σ^{x_1} is the event set, structured as the disjoint union of the boundary event set $\Sigma_B^{x_1}$ and the internal event set $\Sigma_I^{x_1}$ with $\Sigma_B^{x_1} = \{a, b\}$ and $\Sigma_I^{x_1} = \{\alpha, \beta, \lambda\}$;
- There are an incoming boundary transition $\delta_{BI}^{x_1}(x_0, a) = 0$ and an outgoing boundary transition $\delta_{BO}^{x_1}(4, b) = x_2$;
- $X_0 = \{0\}$ is the initial state set; and
- $X_m = \{f\}$ is the terminal state set.

On the other hand, suppose that superstate x_1 in Fig. 4 is an AND superstate, and its internal behavior is depicted by holons $H^{x_{11}}$ and $H^{x_{12}}$ shown in Fig. 7. Finally, by plugging the holons into superstate x_1 in Fig. 4, we obtain the monolithic dynamic structure of $\mathbf{G}^T$ illustrated in Fig. 8. The set of holon describe the dynamic of $\mathbf{G}^T$ is denoted by $\mathcal{H}^{\mathrm{T}} = \{H^x, H^y, H^{x_{11}}, H^{x_{12}}\}$. $\square$

Generally, considering a holon H^x, its external state set X_E^x belongs to X_I^y of holon H^y on the adjacent higher level. The occurrence of $\sigma \in \Sigma_B^x$ leads the system from H^x to H^y or vice versa. We say that superstate x satisfies $x \in X_I^y$, i.e., a lower level holon H^x is considered as

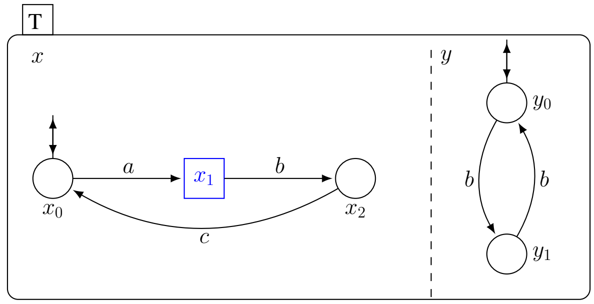


Fig. 4: A set of two holons.

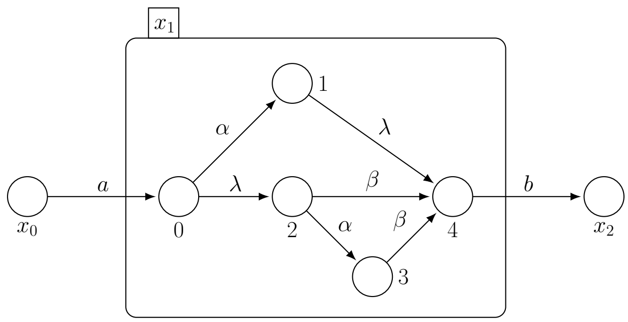


Fig. 5: Holon H^{x_1}.

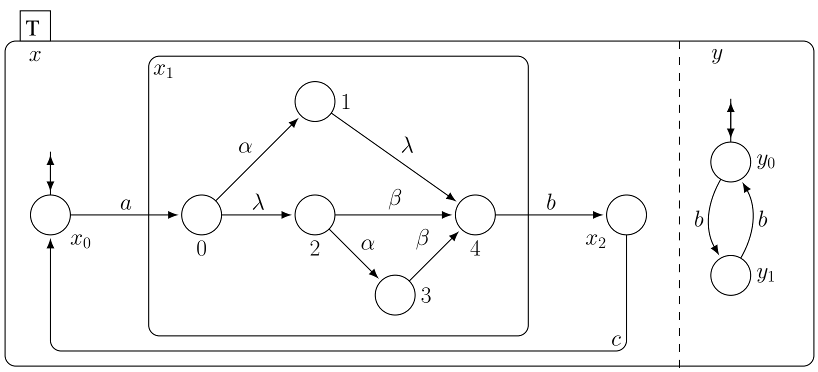


Fig. 6: Monolithic dynamic structure of $\mathbf{G}^T$ (1).

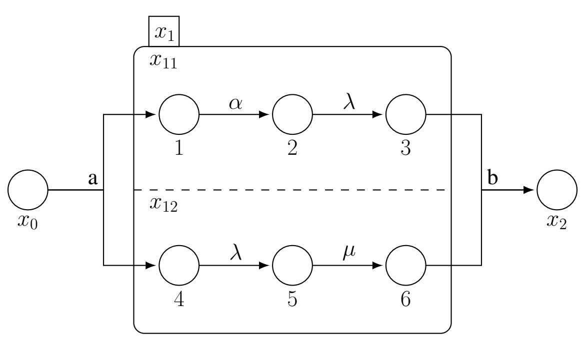


Fig. 7: Holons describe internal behavior of superstate x_1.

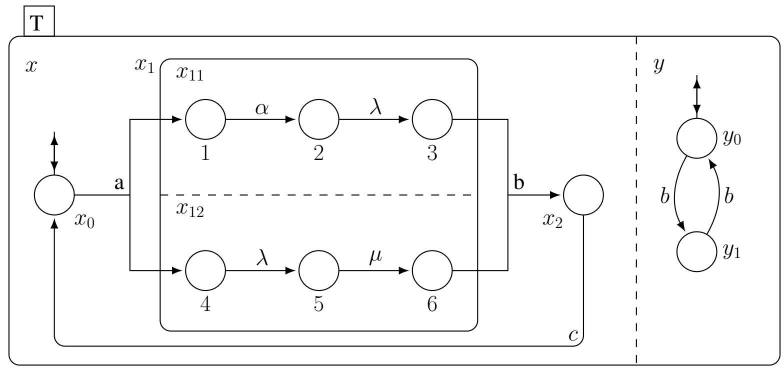


Fig. 8: Monolithic dynamic structure of $\mathbf{G}^T$ (2).

an internal state of H^y. We require $\Sigma_I^x \cap \Sigma_I^y = \emptyset$ holds.

Example.

Holon H^{x_1} illustraged in Fig. 5 is with $X_E^{x_1} = \{x_0, x_2\}$. The state set of holon H^x shown in Fig. 4 is $H^x = \{x_0, x_1, x_2\}$. Clearly, H^{x_1} is viewed as an internal state of H^x. Moreover, with $\Sigma_B^{x_1} = \{a, b\}$ and $\Sigma_I^x = \{a, b, c\}$, we have $\Sigma_E^{x_1} \subset \Sigma_I^x$ and $\Sigma_I^{x_1} \cap \Sigma_I^x = \emptyset$ hold. □

C. State-Trees

Both the hierarchy and horizontal transition relations in an STS are described by a family of holons. The internal structure of a holon matches an OR superstate x, and the external structure of a holon connects its internal behavior with the exosystem (outside world) [43]–[45] that is on the adjacent higher level. The global state space of a set of holon is represented by a state-tree that is hierarchical. Note that the holons with the same state space (and possibly different transition relations) match the same state-tree.

Given a structured state set X. The *reflexive and transitive closure* of $\mathcal{E}$ is written as

$$\mathcal{E}^* : X \to 2^X.$$

Consequently, given a superstate x, the *unfolding* of $\mathcal{E}(x)$ is denoted by

$$\mathcal{E}^+(x) = \mathcal{E}^*(x) - \{x\}.$$

Recursively, a state-tree is a four-tuple

$$\mathbf{S}T = (X, x_0, \mathcal{T}, \mathcal{E}),$$

where X is a finite state set with $X = \mathcal{E}^*(x_0)$ and $x_0 \in X$ is the *root state*. $\mathbf{S}T = (X, x_0, \mathcal{T}, \mathcal{E})$ is a state-tree satisfying:

1) (terminal case) $X = \{x_0\}$ represents that X contains only one simple state; or
2) (recursive case) $(\forall y \in \mathcal{E}(x_0))\mathbf{S}T^y = (\mathcal{E}^*(y), y, \mathcal{T}_{\mathcal{E}^*(y)}, \mathcal{E}_{\mathcal{E}^*(y)})$ is also a state-tree where

$$(\forall y, y' \in \mathcal{E}(x_0))(y \neq y' \Rightarrow \mathcal{E}^*(y) \cap \mathcal{E}^*(y') = \emptyset)$$

and

$$\dot{\bigcup}_{y \in \mathcal{E}(x_0)} \mathcal{E}^*(y) = \mathcal{E}^+(x_0).$$

Example.

The holons shown in Fig. 6 match the state-tree $\mathbf{S}T^T$ depicted in Fig. 9. In a state-tree, the symbol $\times$ (resp., $\dot{\cup}$) is placed between any two adjacent AND (resp., OR) components. In state-tree $\mathbf{S}T^T$, we have

- $X^T = \{T, x, y, x_0, x_1, x_2, y_0, y_1, 0, 1, 2, 3, 4\}$;
- $\mathcal{T}(T) = \mathsf{AND}$;
- $\mathcal{T}(x) = \mathcal{T}(y) = \mathcal{T}(x_1) = \mathsf{OR}$;
- $\mathcal{T}(x_0) = \mathcal{T}(x_2) = \mathcal{T}(y_0) = \mathcal{T}(y_1) = \mathcal{T}(0) = \mathcal{T}(1) = \mathcal{T}(2) = \mathcal{T}(3) = \mathcal{T}(4) = \mathsf{SIM}$; and
- $\mathcal{E}(T) = \{x, y\}$, $\mathcal{E}(x) = \{x_0, x_1, x_2\}$, $\mathcal{E}(y) = \{y_0, y_1\}$, $\mathcal{E}(x_1) = \{0, 1, 2, 3, 4\}$, $\mathcal{E}(x_0) = \emptyset$, $\mathcal{E}(x_2) = \emptyset$, $\mathcal{E}(y_0) = \emptyset$, $\mathcal{E}(y_1) = \emptyset$, $\mathcal{E}(0) = \emptyset$, $\mathcal{E}(1) = \emptyset$, $\mathcal{E}(2) = \emptyset$, $\mathcal{E}(3) = \emptyset$, and $\mathcal{E}(4) = \emptyset$.

For the state-tree $\mathbf{S}T^T$ depicted in Fig. 9, we have

- $\mathcal{E}^*(T) = \{T, x, y, x_0, x_1, x_2, y_0, y_1, 0, 1, 2, 3, 4\}$ and
- $\mathcal{E}^+(T) = \{x, y, x_0, x_1, x_2, y_0, y_1, 0, 1, 2, 3, 4\}$. $\square$

Say that $\mathbf{S}T^y$ is a child-state-tree of x_0 in $\mathbf{S}T$, rooted by y. For convenience, if $y \in \mathcal{E}^+(x)$, we call y a *descendant* of x and x an *ancestor* of y, which is denoted by $x < y$. States x and y are incomparable if x is neither the ancestor nor the descendant of y. An OR superstate y is *AND-adjacent* to an AND superstate x, denoted by $x <_\times y$, if

$$x < y \;\&\; \mathcal{T}(x) = \mathsf{AND} \;\&\; (\forall z) x < z < y \Rightarrow \mathcal{T}(z) = \mathsf{AND}.$$

State z is the *nearest common ancestor* (NCA) of x and y if

$$z < x \;\&\; z < y \;\&\; \neg(\exists a \in \mathcal{E}^+(z)) a < x \;\&\; a < y.$$

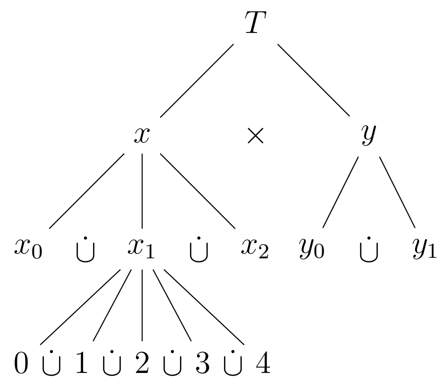
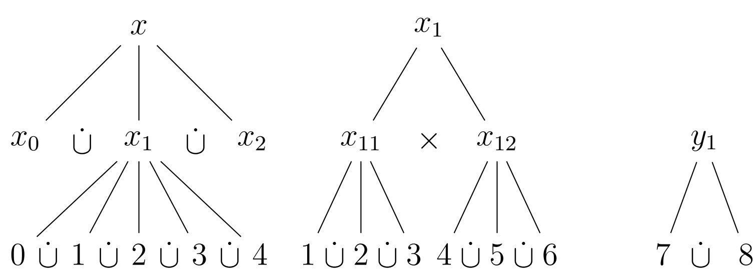


Fig. 9: State-tree matching holons in Fig. 6.

Example.

For the state-tree $\mathbf{S}T^T$ depicted in Fig. 9, we have $T <_\times x_0$, and the NCA of states 0 and y_1 is state T. Moreover, as depicted in Fig. 10, we can obtain three child-state-trees $\mathbf{S}T^x$, $\mathbf{S}T^{x_1}$, and $\mathbf{S}T^y$ rooted by states x, x_1, and y, respectively. $\square$



(a) $\mathbf{S}T^x$. (b) $\mathbf{S}T^{x_1}$. (c) $\mathbf{S}T^{y_1}$.

Fig. 10: Child-state-trees.

A sub-state-tree is denoted by

$$sub\mathbf{S}T = (Y, x_0, \mathcal{T}', \mathcal{E}')$$

with $\mathcal{E}' : Y \to 2^Y$ defined for $y \in Y$ by

$$\begin{cases} \mathcal{E}'(y) = \mathcal{E}(y), & \text{if } \mathcal{T}'(y) \neq \mathsf{OR} \\ \emptyset \subset \mathcal{E}'(y) \subseteq \mathcal{E}(y), & \text{if } \mathcal{T}'(y) = \mathsf{OR} \end{cases}.$$

A well-formed state-tree is a basic-state-tree if any OR superstate has exactly one expansion (or child).

Example.

The state-tree illustrated in Fig. 11 is a sub-state-tree of $\mathrm{S}T^T$ depicted in Fig. 9 and it is also a basic-state-tree. □

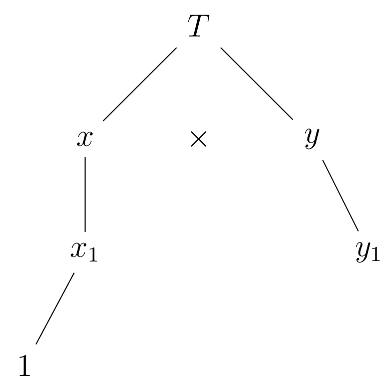


Fig. 11: A basic-state-tree of the state-tree in Fig. 9.

A state-tree is *well-formed* if:

- for any two states x and y, one of the following statements is satisfied:
  - $x \leq y$ or $y \leq x$;
  - $x|y$, namely the NCA of incomparable states x and y is an AND superstate;
  - $x \oplus y$, namely the NCA of incomparable states x and y is an OR superstate;
- $(\forall x, y \in X)\mathcal{T}(x) = \mathsf{AND}\ \&\ y \in \mathcal{E}(x) \Rightarrow \mathcal{T}(x) \neq \mathsf{SIM}$, i.e., AND components cannot be simple states; and
- all the leaf states are simple states.

Example.

The state-tree $\mathrm{S}T^T$ depicted in Fig. 9 is a well-formed state-tree. Moreover, suppose that an AND superstate A is expanded by two OR superstates x and y, i.e., $\mathcal{E}(A) = \{x, y\}$, and the superstate x is further expanded by two simple states x_1 and x_2, i.e., $\mathcal{E}(x) = \{x_1, x_2\}$. The global expansion relation structured in Fig. 12 can be represented by state-tree $\mathrm{S}T^A$ illustrated in Fig. 13(a). Fig. 13(b) depicts a sub-state-tree of state-tree $\mathrm{S}T^A$ illustrated in Fig. 13(a). Both are not well-formed since the leaf state y is an OR superstate. □

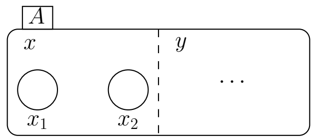
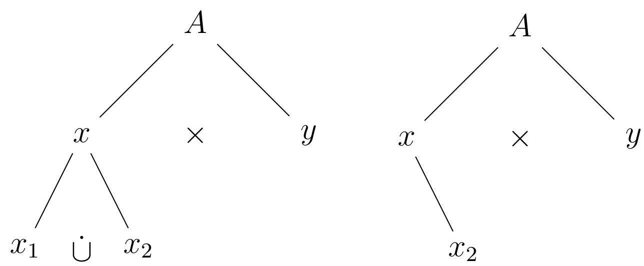


Fig. 12: Superstate expansions.



(a) The matching tree. (b) A sub-state-tree.

Fig. 13: Matching state-tree and a sub-state-tree.

Given a proper sub-state-tree $T = (Y, x_0, \mathcal{T}', \mathcal{E}')$, it can be equivalently represented by its *leaf state set*

$$V(T) = \{x \in Y | \mathcal{E}'(x) = \emptyset\}.$$

For simplification, the corresponding *key leaf state set* is defined below.

$$\mathcal{V}(T) := \begin{cases} V(T), & \text{if } (\nexists x) X_{\mathcal{A}}(x) \subseteq V(T) \\ V(T) - \bigcup_{(\forall x \in X) X_{\mathcal{A}}(x) \subseteq V(T)} X_{\mathcal{A}}(x), & \text{otherwise} \end{cases}$$

It shows that the key leaf states $\mathcal{V}(T)$ only record the proper subsets of OR expansions. Given a state-tree and $\mathcal{V}(T)$, T can be restored.

Example.

The key leaf state set of the basic-state-tree shown in Fig. 11 is denoted by $\mathcal{V}(T) = \{y_1, 1\}$. No matter what is the expansion of the OR superstate y in the sub-state-tree depicted in Fig. 13(b), $\mathcal{V}(\mathbf{S}T^A) = \{x_2\}$ is always true. □

In accordance with [42], the *state aggregation* bonded with a superstate x is denoted by $X_{\mathcal{A}}(x)$. Formally,

$$X_{\mathcal{A}}(x) := \begin{cases} \mathcal{E}(x), & \text{if } \mathcal{T}(x) = \mathsf{OR} \\ \bigcup_{x <_\times y} \mathcal{E}(y), & \text{if } \mathcal{T}(x) = \mathsf{AND} \end{cases}.$$

Example.

For the state-tree $\mathbf{S}T^T$ depicted in Fig. 9, we have four state aggregations listed below:

- $X_{\mathcal{A}}(T) = \{x_0, x_1, x_2, y_0, y_1\}$,
- $X_{\mathcal{A}}(x) = \{x_0, x_1, x_2\}$,
- $X_{\mathcal{A}}(y) = \{y_0, y_1\}$, and
- $X_{\mathcal{A}}(x_1) = \{0, 1, 2, 3, 4\}$. □

D. State-Tree Structures

With holons and state-trees defined, now we are ready to recall the defination of state-tree structures (STS) formally. An STS is a six-tuple

$$\mathbf{G} = (\mathbf{S}T, \mathcal{H}, \Sigma, \Delta, \mathbf{S}T_0, \mathbf{S}T_m),$$

where

- $\mathbf{S}T$ is a *state-tree*;
- $\mathcal{H}$ is the set of *holons*;
- Σ is the union of events appearing in $\mathcal{H}$;
- Δ is the *global transition function* $\mathcal{S}T(\mathbf{S}T) \times \Sigma \to \mathcal{S}T(\mathbf{S}T)$, where $\mathcal{S}T(\mathbf{S}T)$ is the set of all *sub-state-trees*;
- $\mathbf{S}T_0$ is the *initial state-tree*; and
- $\mathbf{S}T_m$ is the *marker state-tree set*.

Example.

The holons shown in Fig. 6 and the matching state-tree $\mathbf{S}T^T$ depicted in Fig. 9 together form an STS. □

An STS $\mathbf{G}$ is well-formed if it satisfies:

- $\mathbf{S}T$ is a well-formed state-tree;
- the states in any holon H^x are *boundary consistency*, i.e., state $y \in X_I^x$ satisfies

$$y \in \mathcal{E}(x)$$

and $y \in X_E^x$ satisfies

$$(\exists z, w \in X) z <_\times w \ \& \ x, y \in \mathcal{E}(w);$$

and

- the states in any holon are *local coupling*, i.e., for holons $H^x, H^y \in \mathcal{H}$,

$$\Sigma_I^x \cap \Sigma_I^y \neq \emptyset \Rightarrow (\exists z) z <_\times x \ \& \ z <_\times y$$

holds.

The boundary consistency requires that the boundary transitions in a holon should not skip holon levels. The local coupling requires that only the holons that have an AND superstate as the NCA of their matching superstates should share events. Hence, this NCA superstate is viewed as the synchronous product of these holons. Unless otherwise stated, in this study, the STS under analysis are well-formed.

The synchronous product principle (an event σ occurring in local coupling holons simultaneously) [4] is integrated in the *largest eligible state-tree* and *largest next state-tree*, denoted by

$$Elig_{\mathbf{G}} : \Sigma \to \mathcal{S}T(\mathbf{S}T)$$

and

$$Next_{\mathbf{G}} : \Sigma \to \mathcal{S}T(\mathbf{S}T),$$

respectively. The key leaf states of $Elig_{\mathbf{G}}(\sigma)$ and $Next_{\mathbf{G}}(\sigma)$ are the exits and entrances of event σ in all the holons where it appears, respectively. The *forward transitions* are defined as

$$\Delta : \mathcal{S}T(\mathbf{S}T) \times \Sigma \to \mathcal{S}T(\mathbf{S}T).$$

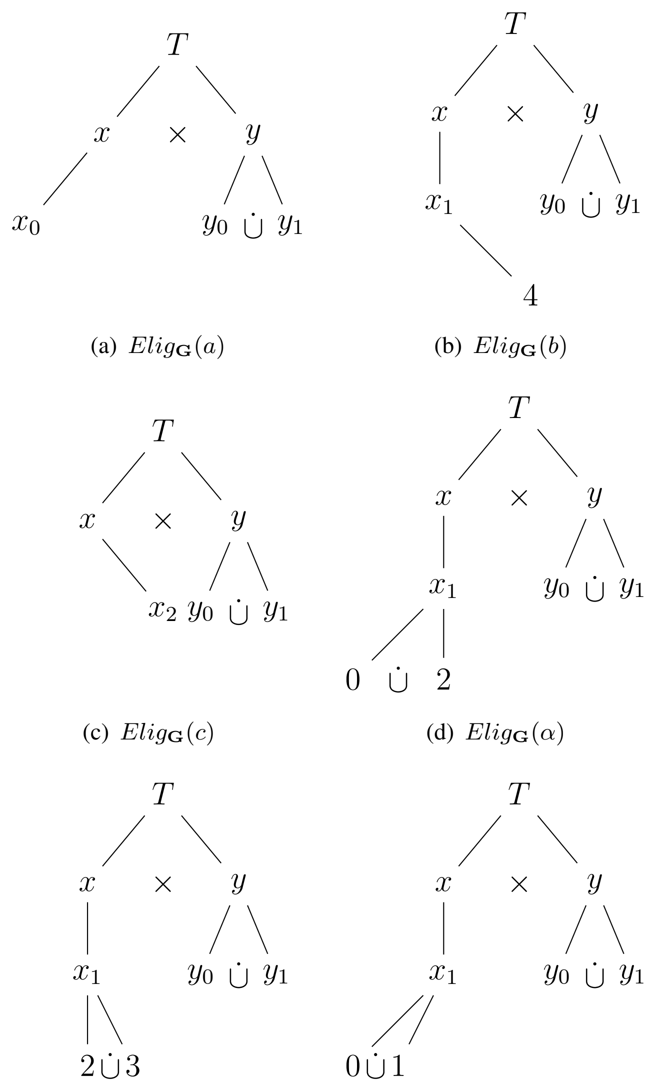
Given any sub-state-tree $T \in \mathcal{S}T(\mathbf{S}T)$, $T' = \Delta(T, \sigma)$ is obtained via replacing the source states of σ in $T \wedge Elig_{\mathbf{G}}$ by the corresponding target states simultaneously. The *backward transitions* are defined as

$$\Gamma : \mathcal{S}T(\mathbf{S}T) \times \Sigma \to \mathcal{S}T(\mathbf{S}T)$$

in a dual route.

Example.

For all the events σ appearing in the holons shown in Fig. 6, $Elig_{\mathbf{G}}(\sigma)$ and $Next_{\mathbf{G}}(\sigma)$ are depicted in Figs. 14 and 15, respectively. Moreover, all the corresponding key leaf state sets are listed in Table I.



(a) $Elig_{\mathbf{G}}(a)$ (b) $Elig_{\mathbf{G}}(b)$

(c) $Elig_{\mathbf{G}}(c)$ (d) $Elig_{\mathbf{G}}(\alpha)$

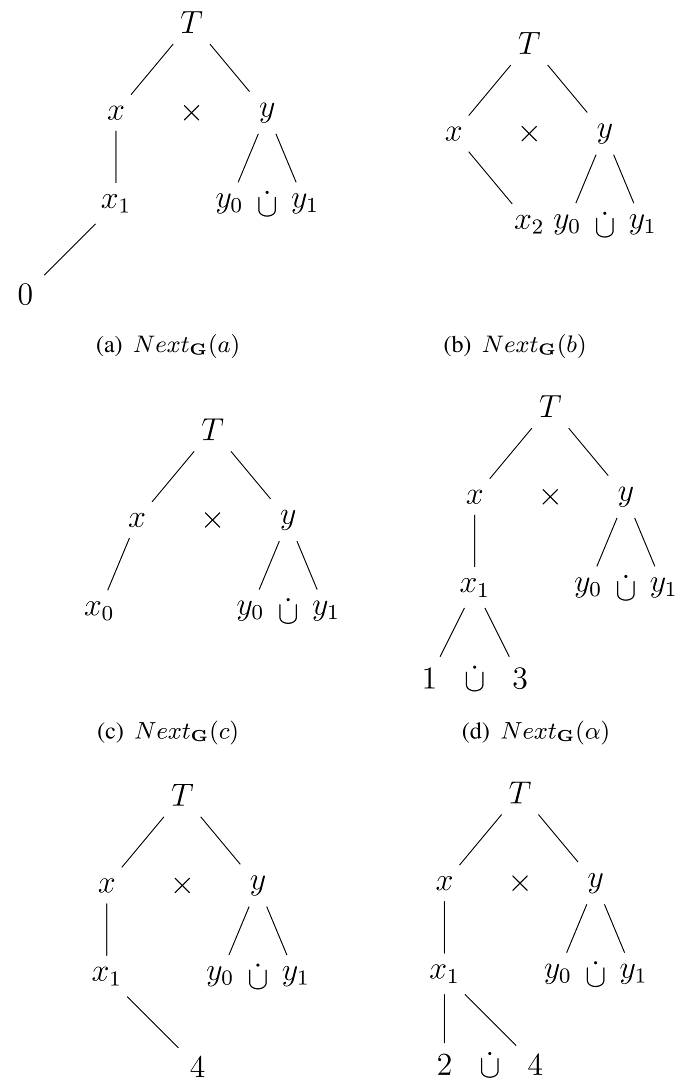
(e) $Elig_{\mathbf{G}}(\beta)$ (f) $Elig_{\mathbf{G}}(\lambda)$

Fig. 14: $Elig_{\mathbf{G}}(\sigma)$ for $\sigma \in \Sigma$.

We have $\mathbf{S}T_0 \in \mathcal{S}T(\mathbf{S}T)$ and $a \in \Sigma$, then we obtain

$$\mathbf{S}T_0 \wedge Elig_{\mathbf{G}}(a) \neq \emptyset$$

and



(a) $Next_{\mathbf{G}}(a)$ (b) $Next_{\mathbf{G}}(b)$

(c) $Next_{\mathbf{G}}(c)$ (d) $Next_{\mathbf{G}}(\alpha)$

(e) $Next_{\mathbf{G}}(\beta)$ (f) $Next_{\mathbf{G}}(\lambda)$

Fig. 15: $Next_{\mathbf{G}}(\sigma)$ for $\sigma \in \Sigma$.

$$\Delta(\mathbf{S}T_0, a) = \mathbf{S}T_1$$

that is $Next_{\mathbf{G}}(a)$ shown in Fig. 15(a). For all the other events $\sigma \in \Sigma - \{a\}$, we have

$$T \wedge Elig_{\mathbf{G}}(\sigma) = \emptyset$$

and

$$\Delta(\mathbf{S}T_0, \sigma) = \emptyset.$$

TABLE I: $Elig_{\mathbf{G}}(\sigma)$ and $Next_{\mathbf{G}}(\sigma)$ for $\sigma \in \Sigma$

Event σ	$Elig_{\mathbf{G}}(\sigma)$	$Next_{\mathbf{G}}(\sigma)$
a	$\{x_0\}$	$\{0\}$
b	$\{4\}$	$\{x_2\}$
c	$\{x_2\}$	$\{x_0\}$
α	$\{0, 2\}$	$\{1, 3\}$
β	$\{2, 3\}$	$\{4\}$
λ	$\{0, 1\}$	$\{2, 4\}$

We say that at state-tree $\mathbf{S}T_0$, event a is enabled. By repeating this process iteratively, we can calculate all the individual sub-state-trees in $\mathbf{S}T$ and the corresponding enabled event sets, which are listed in Table II. □

TABLE II: Enabled events at each sub-state-tree in $\mathbf{S}T$

Sub-state-tree	Key Leaf States	Enabled Event Set
$\mathbf{S}T_0$	$\{x_0\}$	$\{a\}$
$\mathbf{S}T_1$	$\{0\}$	$\{\alpha, \lambda\}$
$\mathbf{S}T_2$	$\{1\}$	$\{\lambda\}$
$\mathbf{S}T_3$	$\{2\}$	$\{\alpha, \beta\}$
$\mathbf{S}T_4$	$\{3\}$	$\{\beta\}$
$\mathbf{S}T_5$	$\{4\}$	$\{b\}$
$\mathbf{S}T_6$	$\{x_2\}$	$\{c\}$

The computation of the total function Γ is started from $\mathbf{S}T_m$ in an opposite way. The details are omitted.

Given an HFSM, there always exists an equivalent single level DES representing its global behavior [6], [23], [24]. Similarly, given an STS, the set of its *basic-state-trees* is denoted by $\mathcal{B}(\mathbf{S}T)$, in which an element T corresponds to a state in a single level DES representing its global behavior. The presented transition relations Δ or Γ maps an element $T \in \mathcal{B}(\mathbf{S}T)$ to another. In this study, these basic-state-trees are symbolically encoded into predicates that are represented by binary decision diagrams (BDD).

Example.

For the STS shown in Figs. 6 and 9, its initial state-tree being a basic-state-tree is depicted in Fig. 16. Suppose that there exists an equivalent single level DES with the initial state representing the initial state-tree depicted in Figs. 6. Clearly, such a DES can be built by tracking the transition relations in the STS. $\square$

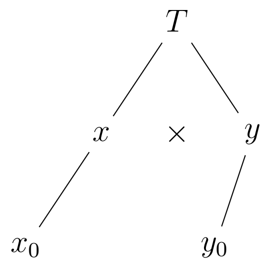


Fig. 16: Initial state-tree.

E. Predicates Representing STS

Given an STS $\mathbf{G}$, the components of $\mathcal{B}(\mathbf{S}T)$ are symbolically encoded into predicates that are represented by BDD. Intuitively, a predicate P (or a *characteristic function*) is defined over $\mathcal{B}(\mathbf{S}T)$, i.e.,

$$P\colon \mathcal{B}(\mathbf{S}T) \to \{0, 1\}.$$

The truth-value 1 (resp., 0) represents logical *true* (resp., *false*). The truth-value 1 (resp., 0) represents logical *true* (resp., *false*). The predicate containing all the basic-state-trees is denoted by a predicate

$$P_{\mathbf{S}T} := \{b \in \mathcal{B}(\mathbf{S}T) | P(b) = 1\}.$$

The truth-value 1 (resp., 0) represents logical *true* (resp., *false*). The predicate containing all the basic-state-trees is denoted by a predicate

$$P_{\mathbf{S}T} := \{b \in \mathcal{B}(\mathbf{S}T) | P(b) = 1\}.$$

Formally,

$$P(b) = 1$$

is represented by

$$b \models P.$$

Propositional logic operators are defined by:

- $(\neg P)(b) = 1$ iff $P(b) = 0$;
- $(P_1 \wedge P_2)(b) = 1$ iff $P_1(b) = 1$ and $P_2(b) = 1$; and
- $(P_1 \vee P_2)(b) = 1$ iff $P_1(b) = 1$ or $P_2(b) = 1$.

Example.

The initial state-tree $\mathbf{S}T_0$ and the marker state-tree set $\mathbf{S}T_m$ are represented by two predicates

$$P_0 := \{b \in \mathcal{B}(\mathbf{S}T_0) | P(b) = 1\}$$

and

$$P_m := \{b \in \mathcal{B}(\mathbf{S}T_m) | P(b) = 1\},$$

respectively. The predicate containning all the basic-state-trees denoted by a predicate

$$P_{\mathbf{S}T} := \{b | b \in \mathcal{B}(\mathbf{S}T) | P(b) = 1\}.$$

$\square$

The set of all predicates on $\mathcal{B}(\mathbf{S}T)$ is defined by $Pred(\mathbf{S}T)$. The partial order for subset containment is defined by $P_1 \preceq P_2$ iff $P_1 \wedge P_2 = P_1$. It is clear that P_1 is stronger than P_2 and $(Pred(\mathbf{S}T), \preceq)$ is a complete lattice. The top and bottom elements of a predicate are denoted as *true* ($\top$) and *false* ($\bot$), respectively.

Example.

Clearly, we have $P_0 \preceq P_{\mathbf{S}T}$ and $P_m \preceq P_{\mathbf{S}T}$. As shown in Fig. 17, for a given STS, $P_{\mathbf{S}T}$ is the weakest predicate which is identified by all the basic-state-trees in $\mathcal{B}(\mathbf{S}T)$. $\square$

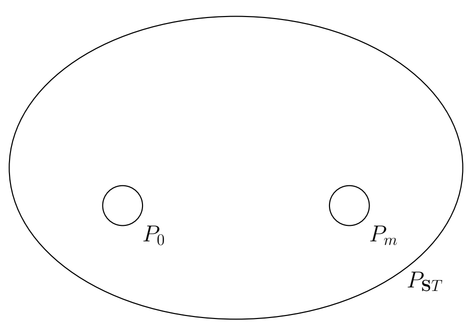


Fig. 17: Predicate containment.

Let $P \in Pred(\mathbf{S}T)$. According to [23] and [24], the reachability predicate $R(\mathbf{G}, P)$ holds the basic-state-trees that can be reached in $\mathbf{G}$, from some $b_0 \models P \wedge P_0$, via a sequence of basic-state-trees all satisfying P. Formally,

- $P \wedge P_0 = \bot \Rightarrow R(\mathbf{G}, P) = \bot$;
- $(b_0 \models P \wedge P_0) \Rightarrow (b_0 \models R(\mathbf{G}, P))$;
- $b \models R(\mathbf{G}, P)$ & $\sigma \in \Sigma$ & $\Delta(b, \sigma) \neq \emptyset$ & $\Delta(b, \sigma) \models P \Rightarrow \Delta(b, \sigma) \models R(\mathbf{G}, P)$; and
- no other basic-state-trees satisfy $R(\mathbf{G}, P)$.

Dually, the coreachability predicate $CR(\mathbf{G}, P)$ is defined holds all the basic-state-trees that can reach some $b_m \models P \wedge P_m$ in $\mathbf{G}$ by a sequence of basic-state-trees all satisfying P. Formally,

- $P \wedge P_m = \bot \Rightarrow CR(\mathbf{G}, P) = \bot$;
- $(b_m \models P \wedge P_m) \Rightarrow (b_m \models CR(\mathbf{G}, P))$;
- $b \models CR(\mathbf{G}, P)$ & $\sigma \in \Sigma$ & $\Gamma(b, \sigma) \neq \emptyset$ & $\Gamma(b, \sigma) \models P \Rightarrow \Gamma(b, \sigma) \models CR(\mathbf{G}, P)$; and
- no other basic-state-trees satisfy $CR(\mathbf{G}, P)$.

Given a predicate P, a predicate transformer $[P]$ in $\mathbf{G}$ is defined by

1) $b \models P \Rightarrow b \models [P]$;
2) $b \models P$ & $\sigma \in \Sigma_u \Rightarrow \Gamma(b, \sigma) \models [P]$; and
3) no other basic-state-trees satisfy $[P]$.

Given a predicate P, by SFBC, the supremal element of *weakly controllable and coreachable behavior*, i.e., *optimal behavior*, of $\mathbf{G}$, is denoted by a nonblocking subpredicate sup$\mathcal{C}^2\mathcal{P}(P)$. It is synthesized iteratively by the following steps:

1) Let $K_0 := P$;
2) compute $K_{i+1} := P \wedge CR(\mathbf{G}, \neg[\neg K_i])$; and
3) If $K_{i+1} = K_i$, then sup$\mathcal{C}^2\mathcal{P}(P) = K_i$. Otherwise, go back to step 2).

F. Supervisory Control

Nonblocking supervisory control of STS utilizes predicates to record the system's behavior. The *weakest liberal precondition* $M_\sigma(P)$ is defined in [23] and [24] as

$$b \models M_\sigma(P)$$

iff

$$\Delta(b, \sigma) \models P.$$

Let $\mathbf{G}$ be an STS, $T \in \mathcal{B}(\mathbf{S}T)$, and $\sigma \in \Sigma$. In STS [23], [24], according to SFBC, preventing the occurrence of an uncontrollable event σ at T is denoted by (T, σ), which considers T as an illegal sub-state-tree. By integrating all such T's with other predefined illegal sub-state-trees, an illegal predicate P is obtained. A *predicate transformer* $[\cdot]$ is utilized to find all the basic-state-trees that can reach P through uncontrollable paths. As a consequence, the family of *weakly controllable subpredicates* of $\neg P$ is denoted by

$$\sup\mathcal{CP}(\neg P) = \neg[P]$$

that is found via calculating $\neg[P]$ iteratively. The corresponding calculation is detailed in [23] and [24], based on which the *control function* f_σ for each controllable event $\sigma \in \Sigma_c$ is obtained. Function f_σ is represented by a predicate, which contains all the basic-state-trees where event σ is allowed to occur. Let f: $\mathcal{B}(\mathbf{S}T) \rightarrow \Pi$ denote the SFBC for $\mathbf{G}$, where

$$\Pi := \{\Sigma' \subseteq \Sigma | \Sigma_u \subseteq \Sigma'\}.$$

Hence, the closed-loop transition function is represented by

$$\Delta^f(b, \sigma) = \Delta(b, \sigma)$$

iff

$$f_\sigma(b) = 1.$$

Let

$$P \in Pred(\mathbf{S}T)$$

and $P \wedge P_0 \neq \bot$. The STS under control is

$$\mathbf{G}^f = (ST, \mathcal{H}, \Sigma, \Delta^f, P_0^f, P_m^f)$$

with

$$P_0^f = P \wedge P_0$$

and

$$P_m^f = P \wedge P_m.$$

As shown in Fig. 18, given a specification predicate P, the optimal behavior of STS $\mathbf{G}$ is represented by $\sup\mathcal{C}^2\mathcal{P}(P)$ that is viewed as an agent $\mathbf{G}_{tracker}$. For the current status (a basic-state-tree b) of $\mathbf{G}$, a set of *decision makers* f_{σ_i} is provided by $\mathbf{G}_{tracker}$ with $\sigma_i \in \Sigma_c$ and $i = 1, 2, \ldots, n$, makes the decisions by applying b as the argument. If

$$f_{\sigma_i}(b) = 1,$$

then σ_i is allowed to occur. Otherwise, it is disabled.

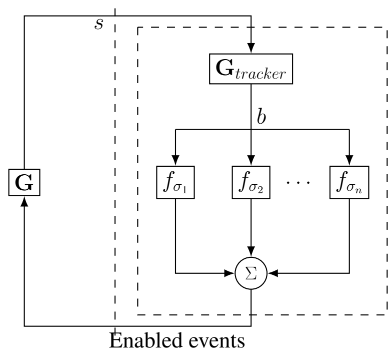


Fig. 18: STS control diagram.

III. Nested Structure of STS

By analyzing the nested structure of an STS's state space, i.e., state-tree, it is decomposed into a set of *state-tree nests* (the largest flat fragments) automatically. Eventually, the STS's nested structure is investigated. In the worst case, an STS that cannot be decoupled is equivalent to a singleton STS nest.

A. State-Tree Nests

In this subsection, a state-tree is decoupled into several state-tree nests with their subordination relation defined.

1) Subordinations of State-Trees:

Given a state-tree, according to *state aggregations* [42], it is decoupled into a set of state-tree nests. In the case of $X_{\mathcal{A}}(y) \subset X_{\mathcal{A}}(x)$ with $x < y$, the calculation of $X_{\mathcal{A}}(y)$ is discarded, which guarantees that no state aggregation is contained by another. State-trees are hierarchical. As presented in Definition 1, a child-state-tree $\mathbf{S}T^y$ (rooted by superstate y) is subordinated to $\mathbf{S}T^x$ if superstate y is in the state aggregation $X_{\mathcal{A}}(x)$.

Definition 1: [Child-State-Tree Subordination] Child-state-tree $\mathbf{S}T^y$ is subordinate to $\mathbf{S}T^x$, denoted by $\mathbf{S}T^x <_N \mathbf{S}T^y$, if y is in the state aggregation $X_{\mathcal{A}}(x)$ of superstate x. Formally,

$$y \in X_{\mathcal{A}}(x) \ \& \ \mathcal{T}(y) \neq \mathsf{SIM} \Rightarrow \mathbf{S}T^x <_N \mathbf{S}T^y.$$

$\diamond$

A state-tree $\mathbf{S}T^x$ is said to be *terminated* at the states with empty expansions. Formally, the termination of $\mathbf{S}T^x$ is denoted by

$$Ter(\mathbf{S}T^x) := \{y \in X^x | \ \mathcal{E}(y) = \emptyset\}.$$

Example.

A state-tree $\mathbf{S}T^{\mathrm{ST}}$ is depicted in Fig. 19, in which three state aggregations

- $X_{\mathcal{A}}(\mathrm{ST}) = \{x_0, x_1, x_2, \ y_0, y_1\}$,
- $X_{\mathcal{A}}(x_1) = \{1, 2, 3, 4, 5, 6\}$, and
- $X_{\mathcal{A}}(y_1) = \{7, 8\}$

are obtained. Since $X_{\mathcal{A}}(x) \subset X_{\mathcal{A}}(\mathrm{ST})$ and $X_{\mathcal{A}}(y) \subset X_{\mathcal{A}}(\mathrm{ST})$, $X_{\mathcal{A}}(x)$ and $X_{\mathcal{A}}(y)$ are discarded. Child-state-trees $\mathbf{S}T^{x_1}$ and $\mathbf{S}T^{y_1}$ (marked in blue) are subordinate to $\mathbf{S}T^{\mathrm{ST}}$. Moreover, we have

- $Ter(\mathbf{S}T^{\mathrm{ST}}) = \{x_0, x_2, y_0, 1, 2, 3, 4, \ 5, 6, 7, 8\}$,
- $Ter(\mathbf{S}T^{x_1}) = \{1, 2, 3, 4, 5, 6\}$, and
- $Ter(\mathbf{S}T^{y_1}) = \{7, 8\}$.

Clearly, state-trees $\mathbf{S}T^{\mathrm{ST}}$, $\mathbf{S}T^{x_1}$, and $\mathbf{S}T^{y_1}$ are all well-formed. $\square$

2) Abstraction:

Given a child-state-tree $\mathbf{S}T^y$, an abstraction is presented to overlook the internal structure of superstate y. Suppose that child-state-tree $\mathbf{S}T^y$ is subordinate to $\mathbf{S}T^x$. From the perspective of $\mathbf{S}T^y$, $\mathbf{S}T^x$ is viewed as its *exosystem* (outside world) on the adjacent higher-level hierarchy. Naturally, for $\mathbf{S}T^x$, an abstraction is developed to “overlook” and replace $\mathbf{S}T^y$ by a corresponding *lumped state* $\underline{y}$. We define that $\mathcal{T}(\underline{y}) = \mathsf{SIM}$. Suppose $\mathbf{S}T^x <_N \mathbf{S}T^y$. Child-state-tree $\mathbf{S}T^y$ is abstracted in $\mathbf{S}T^x$ if $\mathbf{S}T^y$ is replaced by $\mathbf{S}T^{\underline{y}}$ with $\mathcal{E}(\underline{y}) = \emptyset$.

Example.

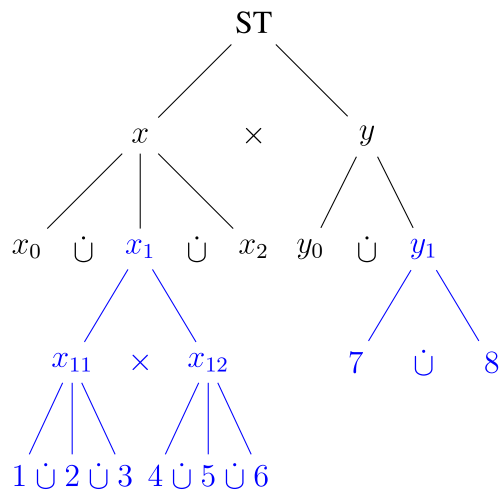


Fig. 19: A state-tree.

By abstracting child-state-trees $\mathbf{S}T^{x_1}$ and $\mathbf{S}T^{y_1}$ (marked in blue) in state-tree $\mathbf{S}T^{\mathrm{ST}}$ shown in Fig. 19, the newly obtained $\mathbf{S}T^{\mathrm{ST}}$ is depicted in Fig. 20. $\square$

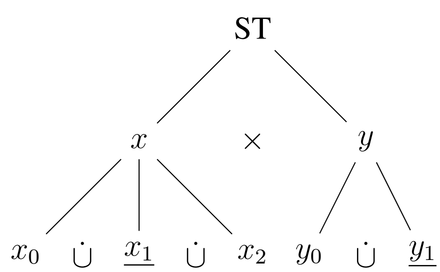


Fig. 20: State-tree $\mathbf{S}T^{\mathrm{ST}}$ after abstraction.

3) State-tree nest subordination:

Intuitively, as presented in Definition 2, given a state aggregation $X_{\mathcal{A}}(x)$, a state-tree nest is a state-tree rooted by superstate x, with all the states in $X_{\mathcal{A}}(x)$ being leaf states.

Definition 2: [State-Tree Nest] A state-tree $\mathbf{S}T^x$ (possibly with some child-state-trees abstracted) is a *state-tree nest*, denoted by $\underline{\mathbf{S}T}^x$, if $\mathbf{S}T^x$ is terminated at $X_{\mathcal{A}}(x)$. $\diamond$

Given a state-tree $\mathbf{S}T$, the set of its state-tree nests, denoted by $\mathbf{S}(\underline{\mathbf{S}T})$, is constructed iteratively as follows:

- $\underline{\mathbf{S}T}^x = \emptyset$ if $x \in X$ & $\mathcal{T}(x) = \mathsf{SIM}$ (terminal case).
- put $\underline{\mathbf{S}T}^x$ into $\mathbf{S}(\underline{\mathbf{S}T})$ if

- $x = x_0$, or
- $(\exists \underline{\mathrm{S}T}^y) x \in X_{\mathcal{A}}(y)$.

In order to avoid any confusion, in the rest of this paper, we use $\mathrm{S}T^x$ and $\underline{\mathrm{S}T}^x$ to denote the child-state-tree bonded with superstate x and the corresponding state-tree nest, respectively. A state-tree nest $\underline{\mathrm{S}T}^y$ is subordinate to $\underline{\mathrm{S}T}^x$ if $\underline{\mathrm{S}T}^y$ describes the lower-level details of a lumped state $\underline{y}$ that is a leaf state of $\underline{\mathrm{S}T}^x$.

Definition 3: [State-Tree Nest Subordination] State-tree nest $\underline{\mathrm{S}T}^y$ is subordinate to $\underline{\mathrm{S}T}^x$ if $y \in X_{\mathcal{A}}(x)$. Formally,

$$(\forall \underline{\mathrm{S}T}^x, \underline{\mathrm{S}T}^y \in \mathrm{S}(\underline{\mathrm{S}T})) y \in X_{\mathcal{A}}(x) \Rightarrow \underline{\mathrm{S}T}^x <_N \underline{\mathrm{S}T}^y.$$

$\diamond$

Given an STS, according to the depth of the STS's hierarchy, the depth of state-tree nests is defined in Definition 4.

Definition 4: [Depth] Suppose $\underline{\mathrm{S}T}^y \in \mathrm{S}(\underline{\mathrm{S}T})$. The depth of $\underline{\mathrm{S}T}^y$, denoted by $d(\underline{\mathrm{S}T}^y)$, is n, if there exists n successive subordination relations $\underline{\mathrm{S}T}^{x_0} <_N \underline{\mathrm{S}T}^w$, $\underline{\mathrm{S}T}^w <_N \underline{\mathrm{S}T}^v$, ..., $\underline{\mathrm{S}T}^z <_N \underline{\mathrm{S}T}^y$ starting from $\underline{\mathrm{S}T}^{x_0}$ (rooted by x_0) and ending by $\underline{\mathrm{S}T}^y$. $\diamond$

Intuitively, as stated in Definition 5, different state-tree nests are siblings if they subordinate to the same state-tree nest.

Definition 5: [State-Tree Nest Siblings] State-tree nests $\underline{\mathrm{S}T}^y$ and $\underline{\mathrm{S}T}^z$ are *siblings*, denoted by $\underline{\mathrm{S}T}^y \sim \underline{\mathrm{S}T}^z$, if they are subordinated to the same state-tree nest $\underline{\mathrm{S}T}^x$. Formally,

$$(\forall \underline{\mathrm{S}T}^x, \underline{\mathrm{S}T}^y, \underline{\mathrm{S}T}^z \in \mathrm{S}(\underline{\mathrm{S}T}))\ \underline{\mathrm{S}T}^x <_N \underline{\mathrm{S}T}^y \ \&\ \underline{\mathrm{S}T}^x <_N \underline{\mathrm{S}T}^z \Rightarrow \underline{\mathrm{S}T}^y \sim \underline{\mathrm{S}T}^z.$$

$\diamond$

We say that $\mathrm{S}T^x$ has a nested structure with a substructure $\mathrm{S}T^y$ similar to itself if $\mathrm{S}T^y$ is subordinate to $\mathrm{S}T^x$.

Example.

For the state-tree depicted in Fig. 19, we have three state-tree nests $\underline{\mathrm{S}T}^{\mathrm{ST}}$, $\underline{\mathrm{S}T}^{x_1}$, and $\underline{\mathrm{S}T}^{y_1}$ satisfying

- $\underline{\mathrm{S}T}^{\mathrm{ST}} <_N \underline{\mathrm{S}T}^{x_1}$, $\underline{\mathrm{S}T}^{\mathrm{ST}} <_N \underline{\mathrm{S}T}^{y_1}$,
- $d(\underline{\mathrm{S}T}^{\mathrm{ST}}) = 0$,
- $d(\underline{\mathrm{S}T}^{x_1}) = 1$,
- $d(\underline{\mathrm{S}T}^{y_1}) = 1$, and
- $\underline{\mathrm{S}T}^{x_1} \sim \underline{\mathrm{S}T}^{y_1}$. $\square$

B. Holon Subordination and Aggregations

A holon, either *deterministic* or *nondeterministic*, describes the internal system behavior of an OR superstate [23], [24].

1) Holon subordination:

As presented in Definition 6, holon H^y is subordinate to H^x if the latter is on the adjacent higher level.

Definition 6: [Holon Subordination] Holon H^y is subordinate to H^x, denoted by $H^x <_N H^y$, if 1) X_E^y is a proper subset of X_I^x; and 2) $x <_\times y$ or $y \in \mathcal{E}(x)$. Formally,

$$(x <_\times y \text{ or } y \in \mathcal{E}(x))\ X_E^y \subset X_I^x \Rightarrow H^x <_N H^y.$$

$\diamond$

Example.

A family of holons H^x, H^y, $H^{x_{11}}$, $H^{x_{12}}$, and H^{y_1} depicted in Fig. 21 matches the state-tree illustrated in Fig. 19, in which the lower-level holons are marked in blue. It shows that

- $H^x <_N H^{x_{11}}$,
- $H^x <_N H^{x_{12}}$, and
- $H^y <_N H^{y_1}$. $\square$

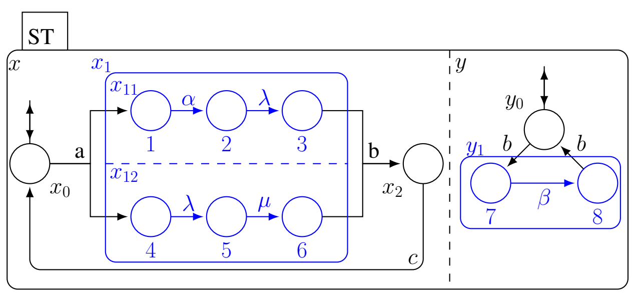


Fig. 21: Holons matching the state-tree in Fig. 19.

Prior to decoupling an STS into a set of STS nests defined later, the single level counterparts of holons are defined below to remove their external and lower-level structures (if any). For a holon H^x, its counterpart $\underline{H}^x$ is obtained by:

- removing its external state set, boundary event set, and boundary transitions; and
- replacing any superstate w in X_I^y by a lumped state $\underline{w}$.

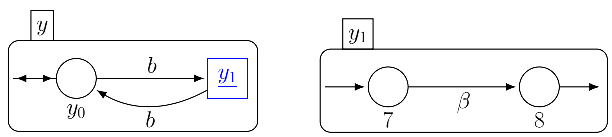
Definition 7: [Single Level Holon Counterpart] The single level counterpart of a holon H^x is a five-tuple $\underline{H}^x = (\underline{X}^x, \underline{\Sigma}^x, \underline{\delta}^x, \underline{X_0}^x, \underline{X_m}^x)$ satisfying

- $\underline{X}^x = \{y, \underline{w} | \mathcal{T}(y) = \mathsf{SIM}, \mathcal{T}(w) \neq \mathsf{SIM}\}$ is the state set;
- $\underline{\Sigma}^x = \Sigma_I^x$ is the event set;
- $\underline{\delta}^x : \underline{X}^x \times \underline{\Sigma}^x \to \underline{X}^x$ is the transition relation, in which $\underline{\delta}^x(z, \sigma)!$ denotes that event $\sigma \in \underline{\Sigma}^x$ is defined at a state $z \in \underline{X}^x$;
- $\underline{X_0}^x = X_0^x$ is the initial state set; and
- $\underline{X_m}^x = X_m^x$ is the terminal state set. $\diamond$

In accordance with the DES modelling principle [4], [5], [23], [24], the *initial* and *terminal states* in holons are marked with incoming and outgoing arrows, respectively.

Example.

For holons H^y and H^{y_1} shown in Fig. 21, their single level holon counterparts $\underline{H}^y$ and $\underline{H}^{y_1}$ are depicted in Figs. 22(a) and 22(b), respectively. As a modelling principle, the lumped state $\underline{y_1}$ in Fig. 22(a) is represented by a box and marked in blue. $\square$



(a) $\underline{H}^y$. (b) $\underline{H}^{y_1}$.

Fig. 22: Single level holon counterparts.

2) Holon aggregations:

Given a superstate x, the bonded *holon aggregation*, denoted by $\mathcal{H}_{\mathcal{A}}(x)$, is a set of single level holon counterparts $\underline{H}^y$ satisfying $X_I^y \subseteq X_{\mathcal{A}}(x)$. Formally,

$$\mathcal{H}_{\mathcal{A}}(x) := \begin{cases} \{\underline{H}^x\}, & \text{if } \mathcal{T}(x) = \mathsf{OR} \\ \{\underline{H}^y | x <_\times y, X_I^y \subseteq X_{\mathcal{A}}(x)\}, & \text{if } \mathcal{T}(x) = \mathsf{AND} \end{cases}.$$

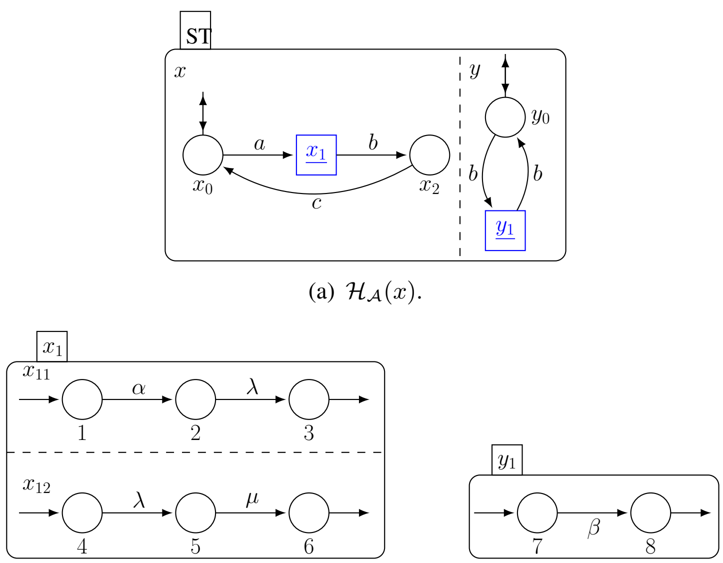
According to local coupling, only the holons in $\mathcal{H}_{\mathcal{A}}(x)$ are allowed to have shared events. The synchronous product principle (an event σ occurring in holons in $\mathcal{H}_{\mathcal{A}}(x)$ simultaneously) is integrated in the definition of forward and backward transition functions to be defined in Section IV-A.

Example.

As depicted in Fig. 23, the holons shown in Fig. 21 contains three holon aggregations:

- $\mathcal{H}_{\mathcal{A}}(\mathrm{ST}) = \{\underline{H}^x, \underline{H}^y\}$,
- $\mathcal{H}_{\mathcal{A}}(x_1) = \{\underline{H}^{x_{11}}, \underline{H}^{x_{12}}\}$, and
- $\mathcal{H}_{\mathcal{A}}(y_1) = \{\underline{H}^{y_1}\}$.

For holon $\underline{H}^x$, we have $\underline{X}^x = \{x_0, \underline{x_1}, x_2\}$. In Fig. 23(a), event b appears in both holons $\underline{H}^x$ and $\underline{H}^y$. According to the synchronous product principle, event b should occur in $\underline{H}^x$ and $\underline{H}^y$ simultaneously. $\square$



(a) $\mathcal{H}_{\mathcal{A}}(x)$.

(b) $\mathcal{H}_{\mathcal{A}}(x_1)$.

(c) $\mathcal{H}_{\mathcal{A}}(y_1)$.

Fig. 23: Holon aggregations.

C. *Formal Definition of STS Nests*

Definition 8: [STS Nest] An STS nest $\underline{\mathbf{G}}^x$ rooted by (i.e., bonded with) a superstate x is a six-tuple $\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_{\mathcal{A}}(x), \Sigma_{\mathcal{A}}(x), \underline{\Delta}^x, \underline{\mathbf{S}T_0}^x, \underline{\mathbf{S}T_m}^x)$, where

- $\underline{\mathbf{S}T}^x$ is a state-tree nest.
- $\mathcal{H}_{\mathcal{A}}(x)$ is the *holon aggregation* of superstate x.
- $\Sigma_{\mathcal{A}}(x)$ is the *event aggregation* of superstate x. Formally, $\Sigma_{\mathcal{A}}(x) := \{\sigma | \sigma \in \Sigma_I^y, \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)\}$.
- $\underline{\Delta}^x$ is the *nested transition structure* of $\underline{\mathbf{G}}^x$ to be defined in Section IV-A.

- $\underline{\mathbf{S}T_0}^x$ is the *initial-state-tree* of $\underline{\mathbf{G}}^x$. Let $z \in A = \{z \in \underline{X_0}^y | \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)\}$. State a in $\underline{\mathbf{S}T}^x$ is said to be in $\underline{\mathbf{S}T_0}^x$ if $a \leq z$ or $a|z$.
- $\underline{\mathbf{S}T_m}^x$ is the *marker-state-tree set* of $\underline{\mathbf{G}}^x$. Let $z \in A = \{z \in \underline{X_m}^y | \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)\}$. State a in $\underline{\mathbf{S}T}^x$ is said to be in $\underline{\mathbf{S}T_m}^x$ if $a \leq z$ or $a|z$. $\diamond$

Building on the subordination and sibling relations of state-tree nests, the subordination and sibling relations among STS nests are defined accordingly. As a general extension, $\underline{\mathbf{G}}^x$ is abstracted if $\underline{\mathbf{S}T}^x$ is abstracted. Given an STS $\mathbf{G}$, its *STS nest set* is denoted by $\mathrm{S}(\underline{\mathbf{G}})$.

Definition 9: [STS Nest Subordination] STS nest $\underline{\mathbf{G}}^y$ is subordinate to $\underline{\mathbf{G}}^x$, denoted by $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$, if $\underline{\mathbf{S}T}^y$ is subordinate to $\underline{\mathbf{S}T}^x$. $\diamond$

Definition 10: [STS Nest Siblings] STS nests $\underline{\mathbf{G}}^y$ and $\underline{\mathbf{G}}^z$ are siblings, denoted by $\underline{\mathbf{G}}^x \sim \underline{\mathbf{G}}^y$, if $\underline{\mathbf{S}T}^y$ and $\underline{\mathbf{S}T}^z$ are siblings. $\diamond$

Example.

Considering the STS $\mathbf{G}$ depicted in Figs. 19 and 21, we have $\mathrm{S}(\underline{\mathbf{G}}) = \{\underline{\mathbf{G}}^{\mathrm{ST}}, \underline{\mathbf{G}}^{x_1}, \underline{\mathbf{G}}^{y_1}\}$, in which $\underline{\mathbf{G}}^{\mathrm{ST}} <_N \underline{\mathbf{G}}^{x_1}$, $\underline{\mathbf{G}}^{\mathrm{ST}} <_N \underline{\mathbf{G}}^{y_1}$, and $\underline{\mathbf{G}}^{x_1} \sim \underline{\mathbf{G}}^{y_1}$. As shown in Fig. 24, the key leaf state set of $\underline{\mathbf{G}}^{x_1}$'s initial state-tree is $\mathcal{V}(\underline{\mathbf{S}T_0}^{x_1}) = \{1, 4\}$. $\square$

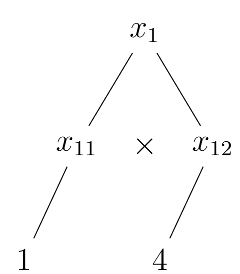


Fig. 24: Initial state-tree $\underline{\mathbf{S}T_0}^{x_1}$.

Remark:

Given a well-formed STS, based on the subordination relation of its state-tree nests, it can be decomposed into a set of STS nests accordingly. However, the STS cannot be decomposed successfully if some of its holons do not satisfy boundary consistency and local coupling. $\square$

IV. Nested Transitions and Communications

The nested transitions and communications of STS nests are investigated in this section. Considering the hierarchical structure of STS, the entrances/exits of any STS nest are built with its static location in the exosystem's state space incorporated. Suppose $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$. From a monolithic (global) perspective, the communication of STS nest $\underline{\mathbf{G}}^y$ guarantees that its system behavior will not block the behavior of its exosystem $\underline{\mathbf{G}}^x$. As a general principle, an STS nest should satisfy:

- *multiple input multiple output* (*MIMO*); and
- *communication*, i.e., the paths in an STS nest lead the system from any of its *initial state-tree* to all the *terminal state-trees*.

A. Nested Transition Structures

Given an STS, without tracking its monolithic state space, the transition structure in an STS nest is built independently. These transition structures form a nested transition structure. Since the transition structure in an STS nest is among simple states only, these transitions are much simpler than the monolithic transitions developed in STS [23], [24].

Given an STS nest $\underline{\mathbf{G}}^x$ bonded with a superstate x, in accordance with STS [23], [24], with the synchronous product principle integrated, the *largest nested eligible state-tree* and *largest nested next state-tree* of event $\sigma \in \Sigma_{\mathcal{A}}(x)$, are defined as

$$Elig_{\underline{\mathbf{G}}^x}(\sigma) : \Sigma_{\mathcal{A}}(x) \to \mathcal{S}T(\underline{\mathbf{S}T}^x)$$

and

$$Next_{\underline{\mathbf{G}}^x}(\sigma) : \Sigma_{\mathcal{A}}(x) \to \mathcal{S}T(\underline{\mathbf{S}T}^x),$$

respectively. In a state-tree nest $\underline{\mathbf{S}T}^x$, the key leaf states of $Elig_{\underline{\mathbf{G}}^x}(\sigma)$ and $Next_{\underline{\mathbf{G}}^x}(\sigma)$ are event σ's exits and entrances in $\mathcal{H}_{\mathcal{A}}(x)$, respectively.

Definition 11: [$\mathcal{V}(Elig_{\underline{\mathbf{G}}^x}(\sigma))$] Let $\sigma \in \Sigma_{\mathcal{A}}(x)$. The key leaf state set of $Elig_{\underline{\mathbf{G}}^x}(\sigma)$ is defined by

$$\mathcal{V}(Elig_{\underline{\mathbf{G}}^x}(\sigma)) := \{a \in \underline{X}^y | (\exists \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)) \underline{\delta}^y(a, \sigma)!\}.$$

◇

Definition 12: [$\mathcal{V}(Next_{\underline{\mathbf{G}}^x}(\sigma))$] Let $\sigma \in \Sigma_{\mathcal{A}}(x)$. The key leaf state set of $Next_{\underline{\mathbf{G}}^x}(\sigma)$ is defined by

$$\mathcal{V}(Next_{\underline{\mathbf{G}}^x}(\sigma)) := \{b \in \underline{X}^y | (\exists \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x), a \in \underline{X}^y) \underline{\delta}^y(a, \sigma) = b\}.$$

◇

Let $a \in \mathcal{V}(Elig_{\underline{\mathbf{G}}^x}(\sigma))$ (resp., $a \in \mathcal{V}(Next_{\underline{\mathbf{G}}^x}(\sigma))$). A state z is in $Elig_{\underline{\mathbf{G}}^x}(\sigma) \in \mathcal{ST}(\underline{\mathbf{S}T}^x)$ (resp., $Next_{\underline{\mathbf{G}}^x}(\sigma) \in \mathcal{ST}(\underline{\mathbf{S}T}^x)$) if $z \leq a$ or $a|z$.

Example.

Consider the holon aggregations depicted in Fig. 23(a). We have

$$\mathcal{V}(Elig_{\underline{\mathbf{G}}^{\mathrm{ST}}}(b)) = \{\underline{x_1}, y_0, \underline{y_1}\}$$

and

$$\mathcal{V}(Next_{\underline{\mathbf{G}}^{\mathrm{ST}}}(b)) = \{x_2, y_0, \underline{y_1}\}.$$

Sub-state-trees $Elig_{\underline{\mathbf{G}}^{\mathrm{ST}}}(b)$ and $Next_{\underline{\mathbf{G}}^{\mathrm{ST}}}(b)$ are shown in Fig. 25. In comparison, according to [23] and [24], in the global STS $\mathbf{G}$, the global largest eligible state-tree satisfies

$$\mathcal{V}(Elig_{\mathbf{G}}(b)) = \{3, 6\},$$

as depicted in Fig. 23(c), the key leaf states 3 and 6 of $Elig_{\mathbf{G}}(b)$ are in the lower-level STS nest $\mathbf{S}T^{x_1}$. Similarly, as depicted in Fig. 23(d), the global largest next state-tree satisfies

$$\mathcal{V}(Next_{\mathbf{G}}(b)) = \{x_2\}.$$

Clearly, sub-state-trees $Elig_{\mathbf{G}}(b)$ and $Next_{\mathbf{G}}(b)$ are more complex. $\square$

Given an STS nest $\underline{\mathbf{G}}^x \in \mathbf{S}(\underline{\mathbf{G}})$, its forward and backward transition functions are defined below.

Definition 13: [Forward Transition Function $\underline{\Delta}^x$] Let

$$\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_{\mathcal{A}}(x), \Sigma_{\mathcal{A}}(x), \underline{\Delta}^x, \underline{\mathbf{S}T_0}^x, \underline{\mathbf{S}T_m}^x)$$

be an STS nest with a root state x. The forward transition function

$$\underline{\Delta}^x : \mathcal{ST}(\underline{\mathbf{S}T}^x) \times \Sigma_{\mathcal{A}}(x) \to \mathcal{ST}(\underline{\mathbf{S}T}^x)$$

maps a sub-state-tree of $\underline{\mathbf{S}T}^x$ associated with an event $\sigma \in \Sigma_{\mathcal{A}}(x)$ into another. Let $T \in \mathcal{ST}(\underline{\mathbf{S}T}^x)$ and $\sigma \in \Sigma_{\mathcal{A}}(x)$. $\underline{\Delta}^x$ is defined as

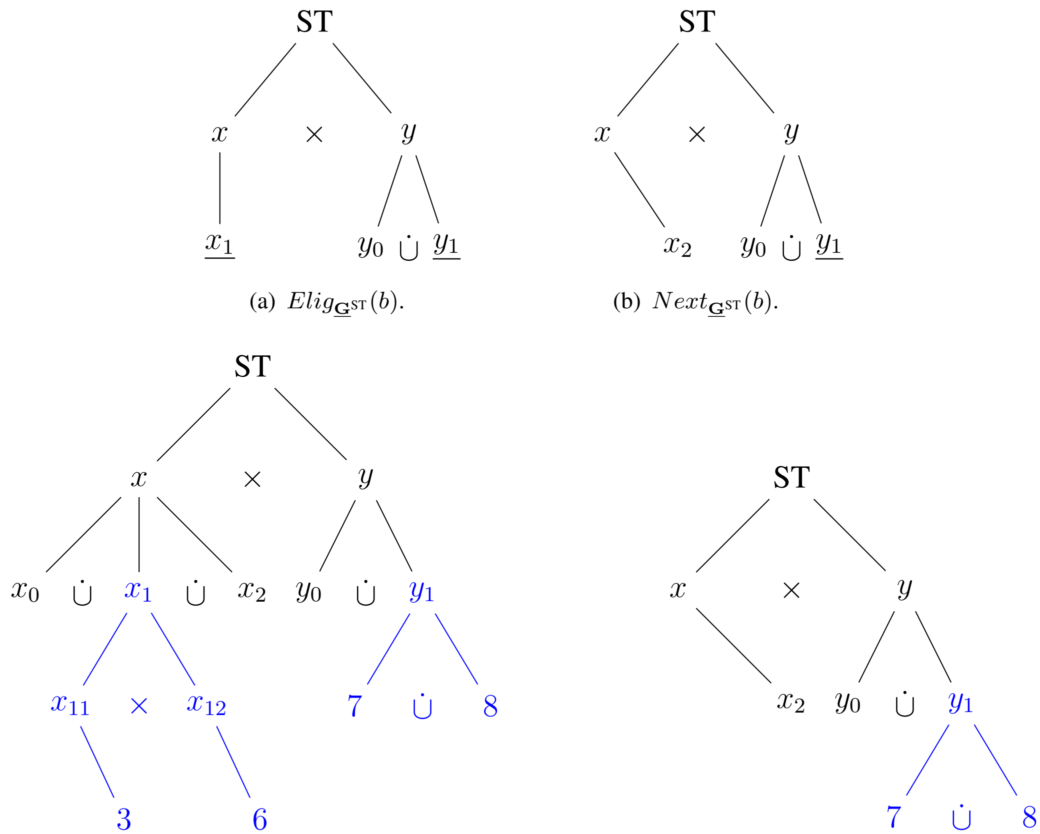
$$\underline{\Delta}^x := \text{replace\_source}_{\underline{\mathbf{G}}^x,\sigma}(T \wedge Elig_{\underline{\mathbf{G}}^x}(\sigma)),$$

where

$$\text{replace\_source}_{\underline{\mathbf{G}}^x,\sigma} : \mathcal{ST}(Elig_{\underline{\mathbf{G}}^x}(\sigma)) \to \mathcal{ST}(\underline{\mathbf{S}T}^x)$$

is defined as: $\underline{\mathbf{S}T}_2^x :=$ replace_source$_{\underline{\mathbf{G}}^x,\sigma}(\underline{\mathbf{S}T}_1^x)$. Suppose

$$(\forall a \in \mathcal{V}(\underline{\mathbf{S}T}_1^x), (\exists \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)) b \in \underline{X}^y) \underline{\delta}^y(a, \sigma) = b.$$



(a) $Elig_{\underline{\mathbf{G}}^{\text{ST}}}(b)$.

(b) $Next_{\underline{\mathbf{G}}^{\text{ST}}}(b)$.

(c) $Elig_{\underline{\mathbf{G}}}(b)$.

(d) $Next_{\underline{\mathbf{G}}}(b)$.

Fig. 25: Largest nested eligible state-tree and next state-tree.

$\underline{\mathbf{S}T}_2^x$ is obtained via replacing state b in $\underline{\mathbf{S}T}_1^x$ by state a. $\diamond$

Following a dual route, the backward transition function $\underline{\Gamma}^x$ is defined below.

Definition 14: [Backward Transition Function $\underline{\Gamma}^x$] Let

$$\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_{\mathcal{A}}(x), \Sigma_{\mathcal{A}}(x), \underline{\Delta}^x, \underline{\mathbf{S}T_0}^x, \underline{\mathbf{S}T_m}^x)$$

be an STS nest rooted by superstate x. The backward transition function

$$\underline{\Gamma}^x : \mathcal{S}T(\underline{\mathbf{S}T}^x) \times \Sigma_{\mathcal{A}}(x) \to \mathcal{S}T(\underline{\mathbf{S}T}^x)$$

maps a sub-state-tree of $\underline{\mathbf{S}T}^x$ associated with an event $\sigma \in \Sigma_{\mathcal{A}}(x)$ into another. Let $T \in \mathcal{S}T(\underline{\mathbf{S}T}^x)$ and $\sigma \in \Sigma_{\mathcal{A}}(x)$. The backward transition function $\underline{\Gamma}^x$ is defined as

$$\underline{\Gamma}^x := \text{replace\_target}_{\underline{\mathbf{G}}^x,\sigma}(T \wedge Next_{\underline{\mathbf{G}}^x}(\sigma)),$$

where

$$\text{replace\_target}_{\underline{\mathbf{G}}^x,\sigma} : \mathcal{S}T(Next_{\underline{\mathbf{G}}^x}(\sigma)) \to \mathcal{S}T(\underline{\mathbf{S}T}^x)$$

is defined as: $\underline{\mathbf{S}T}_1^x :=$ replace_target$_{\underline{\mathbf{G}}^x,\sigma}(\underline{\mathbf{S}T}_2^x)$. Suppose

$$((\exists \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x))a \in \underline{X}^y, \forall b \in \mathcal{V}(\underline{\mathbf{S}T}_2^x))\underline{\delta}^y(a,\sigma) = b.$$

$\underline{\mathbf{S}T}_1^x$ is obtained via replacing state a in $\underline{\mathbf{S}T}_2^x$ by state b. $\diamond$

Example.

Consider the holon aggregation $\mathcal{H}_{\mathcal{A}}(\mathrm{ST})$ depicted in Fig. 23(a), as defined in Definition 13, a backward transition

$$\underline{\Gamma}^{\mathrm{ST}}(\{x_2, \underline{y_1}\}, b) = \{\underline{x_1}, y_0\}$$

holds. As displayed in Fig. 26, this backward transition leads the system from STS nest $\underline{\mathbf{G}}^{y_1}$ to $\underline{\mathbf{G}}^{x_1}$. A precise investigation on the system behavior in STS nests is addressed in Section IV-C. $\square$

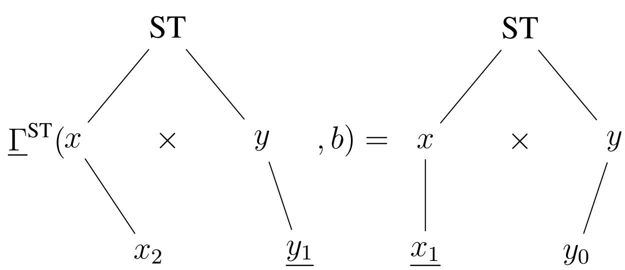


Fig. 26: A backward transition relation.

Let Δ/Γ and $\underline{\Delta}^x/\underline{\Gamma}^x$ denote the transition structures in the global STS $\mathbf{G}$ and an STS nest $\underline{\mathbf{G}}^x \in \mathbf{S}(\underline{\mathbf{G}})$, respectively. The diagram depicted in Fig. 27 commutes. Essentially, the transition relation in STS nest $\underline{\mathbf{G}}^x$ is the projection of $\mathbf{G}$'s monolithic transition relation on $\underline{\mathbf{G}}^x$.

$$\begin{tikzcd}
\mathbf{G} \arrow[r, "\Delta/\Gamma"] \arrow[d, "Proj"'] & \mathbf{G} \arrow[d, "Proj"] \\
\underline{\mathbf{G}}^x \in \mathbf{S}(\mathbf{G}) \arrow[r, "\underline{\Delta}/\underline{\Gamma}"'] & \underline{\mathbf{G}}^x \in \mathbf{S}(\mathbf{G})
\end{tikzcd}$$

Fig. 27: Transition commutative diagram.

B. Static Hierarchical Location of STS Nest I/O

The entrances and exits (I/O) of an STS nest are defined in accordance with the global hierarchical structure of STS. Suppose $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$. Let σ be an event leading the system from $\underline{\mathbf{G}}^x$ (the exosystem on the higher-level hierarchy) to $\underline{\mathbf{G}}^y$. Clearly, $Next_{\mathbf{G}}(\sigma)$ proposed in [23] and [24] crosses $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$. More precisely, its key leaf state set $\mathcal{V}(Next_{\mathbf{G}}(\sigma))$ contains:

- the initial states of holon family $\mathcal{H}_{\mathcal{A}}(y)$ in $\underline{\mathbf{G}}^y$, which are visited by the occurrence of event σ, and
- the states in $\mathbf{G}$ in parallel with lumped state $\underline{y}$ (i.e., $\underline{\mathbf{G}}^y$), which will be accessed simultaneously via the occurrence of event σ.

Dually, let τ be an event leading the system from $\underline{\mathbf{G}}^y$ to $\underline{\mathbf{G}}^x$. The (static) largest eligible-state-tree $Elig_{\mathbf{G}}(\tau)$ crosses $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$. The key leaf state set $\mathcal{V}(Elig_{\mathbf{G}}(\tau))$ contains

- the terminal states of holons in $\underline{\mathbf{G}}^y$, and
- the states in $\mathbf{G}$ in parallel with lumped state $\underline{y}$ (i.e., $\underline{\mathbf{G}}^y$), at which event τ is eligible to occur.

Intuitively, visiting $Next_{\mathbf{G}}(\sigma)$ (resp., $Elig_{\mathbf{G}}(\tau)$) is a precondition such that the system visiting (resp., leaving) $\underline{\mathbf{G}}^y$ via the occurrence of σ (resp., τ). Hence, sub-state-trees $Next_{\mathbf{G}}(\sigma)$ and $Elig_{\mathbf{G}}(\tau)$ of the global STS $\mathbf{G}$ are borrowed to address the static hierarchical location of STS nest I/O.

Example.

A partial diagram of STS containing two STS nests $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ with $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$ is depicted in Fig. 28, in which $\underline{\mathbf{G}}^y$ is marked in blue. We have

- $\mathcal{V}(Next_{\mathbf{G}}(\sigma)) = \{2, 6, 11, 13\}$ and
- $\mathcal{V}(Elig_{\mathbf{G}}(\tau)) = \{3, 7, 12, 14\}$.

It is shown that:

- visiting states 2 and 6 is a precondition such that the system visits $\underline{\mathbf{G}}^y$ (through the initial states 11 and 13); and
- visiting states 3 and 7 is a precondition such that the system leaves $\underline{\mathbf{G}}^y$ (through the terminal states 12 and 14).

Hence, $Next_{\mathbf{G}}(\sigma)$ and $Elig_{\mathbf{G}}(\tau)$ are viewed the static location of STS nest $\underline{\mathbf{G}}^y$, and it is unnecessary to track the entire dynamics of $\underline{\mathbf{G}}^x$ (the exosystem of $\underline{\mathbf{G}}^y$). $\square$

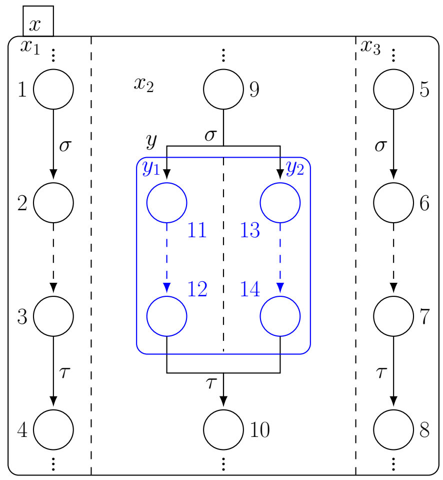
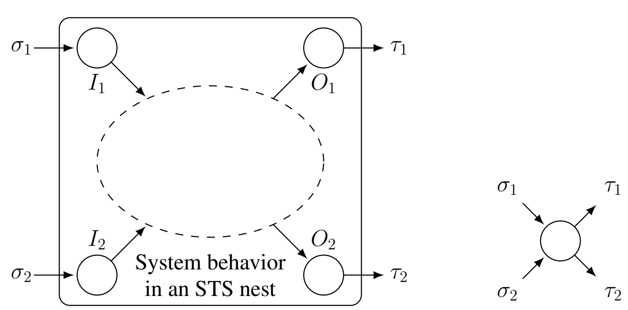


Fig. 28: A partial diagram of STS.

C. STS Nest Communication with MIMO

Given a general example of an MIMO STS nest, as depicted in Fig. 29(a), there are two initial-state-trees I_1 and I_2 and two marker-state-trees O_1 and O_2. By abstracting the STS nest's internal behavior, as depicted in Fig. 29(b), it is viewed as a simple state. Hence, in order not to block the system behavior in the modelling phase, the state-tree paths in Fig. 29(a) must lead the system from either I_1 or I_2 to both O_1 and O_2 such that the *outgoing boundary transitions* labelled with τ_1 and τ_2 are not blocked.



(a) I/O. (b) Abstraction.

Fig. 29: The I/O and abstraction of STS Nests.

Definition 15: [Entrances] Let $\underline{\mathbf{G}}^x \in \mathbf{S}(\mathbf{G})$ be an STS nest. The entrances of $\underline{\mathbf{G}}^x$ (via different events) form an *entrance family*, denoted by $\mathcal{I}^x$. Formally,

$$\mathcal{I}^x = \{T \in Next_{\mathbf{G}}(\sigma) | \sigma \in \Sigma_{BI}^y, \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)\}.$$

$\diamond$

Definition 16: [Exits] Let $\underline{\mathbf{G}}^x \in \mathbf{S}(\mathbf{G})$ be an STS nest. The exits of $\underline{\mathbf{G}}^x$ (via different events) form an *exit family*, denoted by $\mathcal{O}^x$. Formally,

$$\mathcal{O}^x = \{T \in Elig_{\mathbf{G}}(\tau) | \tau \in \Sigma^y_{BO}, \underline{H}^y \in \mathcal{H}_{\mathcal{A}}(x)\}.$$

$\diamond$

Example.

For the STS depicted in Figs. 19 and 21, $\underline{\mathbf{G}}^{x_1}$ has an entrance and an exit with key leaf state sets $\{1, 4\}$ and $\{3, 6\}$, respectively. $\square$

V. STS Nests Encoding and Communication Verification

In this section, STS nests are encoded into predicates, which is prepared for the nested supervisory control of STS discussed in Section VI. Moreover, an approach to verify the communication of STS nests is developed.

A. State-Tree Encoding

According to [23] and [24], as stated in Algorithm 1, any state-tree $\mathbf{S}T$ is encoded into a predicate P by function

$$\Theta : \mathcal{S}T(\mathbf{S}T) \to Pred(\mathbf{S}T).$$

In the binary decision diagram (BDD) representation, the OR (resp., SIM) states are encoded to be *variables* (resp., *values*). Formally, let

$$\mathbf{S}T_1 = (X_1, x_{1,0}, \mathcal{T}_1, \mathcal{E}_1)$$

be a sub-state-tree of $\mathbf{S}T$. Define $\Theta : \mathcal{S}T(\mathbf{S}T) \to Pred(\mathbf{S}T)$ recursively by

$$\Theta(\mathbf{S}T_1) := \begin{cases} \bigwedge_{y \in \mathcal{E}_1(x_0)} \Theta'(\mathbf{S}T_1^y), & \text{if } \mathcal{T}(x_0) = \mathsf{AND} \\ \bigvee_{y \in \mathcal{E}_1(x_0)} ((v_{x_0} = y) \wedge \Theta'(\mathbf{S}T_1^y)) & \text{if } \mathcal{T}(x_0) = \mathsf{OR} \\ 1, & \text{if } \mathcal{T}(x_0) = \mathsf{SIM} \end{cases}$$

where $\mathbf{S}T_1^y$ is the child-state-tree of $\mathbf{S}T_1$ rooted by y, and assume that $\Theta' : \mathcal{S}T(\mathbf{S}T) \to Pred(\mathbf{S}T)$ is already defined for the child-state-tree $\mathbf{S}T_1^y$. Trivially, define $\Theta(\mathbf{S}T_1) \equiv 0$ if $\mathbf{S}T_1$ is an empty state-tree. In order to simplify $\Theta(\mathbf{S}T_1)$, the tautology $(\bigvee_{y \in \mathcal{E}_1(x_0)} (v_{x_0} = y)) \equiv 1$ is exploited.

Algorithm 1 Predicate encoding of state-trees

Input: A state-tree $\mathbf{S}T = (X, x_0, \mathcal{T}, \mathcal{E})$.
Output: A predicate P.
1. Predicate $P \equiv \bot$;
2. $x \leftarrow x_0$; //root state.
3. **function** $\textbf{encoding}(x)$;
4.   **if** $\mathcal{T}(x) = \mathsf{AND}$;
5.   **for** each state y in $\mathcal{E}(x)$;
6. $x \leftarrow y$;
7.   **goto** line 3;
8.   **endfor**;
9.   **endif**;
10.   **if** $\mathcal{T}(x) = \mathsf{OR}$;
11.   **for** each state y in $\mathcal{E}(x)$;
12. $((v_x = y) \wedge \textbf{encoding}(y)) \models P$;
13.   **endfor**;
14.   **endif**;
15.   **if** $\mathcal{T}(x) = \mathsf{SIM}$;
16. $(v_x = 1) \models P$;
17.   **endif**;
18.   **return** P;
19. **end function**;

Given an STS, its BDD variables are ordered in a top-down approach according to the subordination relation among STS nests. In accordance with [23], we require that:

- the encoding for any transition labelled event σ should be linear in the number of transitions; and
- in the case that holon H^y is subordinate to H^x, the BDD variables of H^x should precede those of holon H^y.

B. State-Tree Nests Encoding

Let $\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_{\mathcal{A}}(y), \Sigma_{\mathcal{A}}(y), \underline{\Delta}^x, \underline{\mathbf{S}T_0}^x, \underline{\mathbf{S}T_m}^x)$ be an STS nest. Its basic-state-tree set $\mathcal{B}(\mathbf{S}T)$ is encoded into a predicate P^x as a function

$$P^x := \mathcal{B}(\underline{\mathbf{S}T}^x) \to \{0, 1\}.$$

Consequently, $\underline{\mathbf{G}}^x$ can be rewritten as

$$\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_{\mathcal{A}}(y), \Sigma_{\mathcal{A}}(y), \underline{\Delta}^x, \underline{P_0}^x, \underline{P_m}^x),$$

in which $\underline{P_0}^x$ and $\underline{P_m}^x$ are the *initial predicate* and *marker predicate*, respectively.

Theorem 1: $\Theta(\underline{\mathbf{S}T^x}) := \Theta(\mathbf{S}T^x)$.

Proof: Suppose $Ter(\mathbf{S}T^x) = Ter(\underline{\mathbf{S}T^x})$. It is obvious that $\Theta(\underline{\mathbf{S}T^x}) := \Theta(\mathbf{S}T^x)$.

Suppose $Ter(\mathbf{S}T^x) \neq Ter(\underline{\mathbf{S}T^x})$; there exists $\underline{y} \in Ter(\underline{\mathbf{S}T^x})$. In $\mathbf{S}T^x$, we have $\Theta(\mathbf{S}T^y) = (\bigvee_{z\in\mathcal{E}(y)}(v_y = z)) \equiv 1$. In $\underline{\mathbf{S}T^x}$, $\mathcal{T}(\underline{y}) = \mathsf{SIM}$ implies $\Theta(\mathbf{S}T^{\underline{y}}) = 1$. Thus, $\Theta(\mathbf{S}T^y) = \Theta(\mathbf{S}T^{\underline{y}})$ and $\Theta(\underline{\mathbf{S}T^x}) := \Theta(\mathbf{S}T^x)$. ■

C. Encoding Backward Transition Function

The backward transition function $\underline{\hat{\Gamma}}^x$ of an STS nest $\underline{\mathbf{G}}^x$ is encoded for the purpose of synthesizing the nested optimal nonblocking supervisor presented in Section VI. Transition function $\underline{\hat{\Gamma}}^x$ is only with a horizontal structure, which significantly simplifies the transition structures of STS.

Let $\underline{\mathbf{G}}^x = (\underline{\mathbf{S}T}^x, \mathcal{H}_{\mathcal{A}}(y), \Sigma_{\mathcal{A}}(y), \underline{\Delta}^x, \underline{\mathbf{S}T_0}^x, \underline{\mathbf{S}T_m}^x)$ be an STS nest and y be an OR superstate in $\underline{\mathbf{S}T}^x$. According to [24], from the perspective of $\underline{\Gamma}^x$, denoted by *normal* and *prime* state variables of y in a transition relation, v_y and $v_y{}'$ are used to encode the *target* and *source* states, respectively.

Suppose that an event σ in $\Sigma_{\mathcal{A}}(x)$ appears in holon $\underline{H}^y$ in $\mathcal{H}_{\mathcal{A}}(x)$, and a transition t_σ satisfies $\underline{\delta}^y(z,\sigma) = w$. Then we have the transition t_σ encoded as

$$N_{t_\sigma} := (v_y{}' = z) \wedge (v_y = w).$$

It is possible that event σ can occur sequentially in a holon $\underline{H}^y$ and concurrently in several holons in $\mathcal{H}_{\mathcal{A}}(x)$. Let $\mathbf{T}_\sigma^x$ be the set of transitions in holon $\underline{H}^y$. The entire set of transitions relation labelled with σ is encoded as

$$N_\sigma := \bigwedge_{\underline{H}^y\in\mathcal{H}_{\mathcal{A}}(x)} \bigvee_{t_\sigma\in\mathbf{T}_\sigma^x} N_{t_\sigma}.$$

According to [23], let $\mathcal{E}(x) = \{y_1, y_2, \ldots, y_n\}$ be the range of v_x. Denote by $P[y_i/v_x]$ the resulting predicate after assigning y_i to v_x. Then we have

$$\exists v_x P := \bigvee_{i=1}^n P[y_i/v_x].$$

Let $\mathbf{v} = \{v_i | i = 1, 2, \ldots, n\}$. We have

$$\exists \mathbf{v} P := \exists v_1 (\exists v_2 \dots (\exists v_n P)).$$

Given a predicate P, the set of variables in it is denoted by $\mathbf{v}$. Replacing each variable v_y by the corresponding prime variable $v_y{}'$ leads to

$$P(\mathbf{v}') := P(\mathbf{v})[\mathbf{v} \rightarrow \mathbf{v}'].$$

Definition 17: [Encoding of $\hat{\underline{\Gamma}}^x$] Let $\underline{\mathbf{G}}^x$ be an STS nest, P be a predicate, $\sigma \in \Sigma_{\mathcal{A}}(x)$, and $\mathbf{v}_\sigma = \{v_y | \sigma \in \Sigma_I^y\}$. We define

$$\hat{\underline{\Gamma}}^x : Pred(\underline{\mathbf{S}T}^x) \times \Sigma_{\mathcal{A}}(x) \rightarrow Pred(\underline{\mathbf{S}T}^x)$$

as

$$\hat{\underline{\Gamma}}^x (P, \sigma) = (\exists \mathbf{v}_\sigma (P \wedge N_\sigma))[\mathbf{v}'_\sigma \rightarrow \mathbf{v}_\sigma].$$

We write $\underline{\Gamma}$ in the case of no ambiguity. $\diamond$

The computation of a backward transition function presented in Definition 17 is coded in Algorithm 2. For the backward transitions labelled with event σ, in a predicate P, the encoded variable pairs (for the transitions in different holons) are replaced by the normal variables (encoding the source states) simultaneously. Hence, the synchronous product principle is integrated.

Algorithm 2 Backward transition function

Input: A predicate P and an event σ.
Output: A predicate Q.
Predicate $Q = P$;
for each variable v_y in Q;
    if $(v'_y = z \wedge v_y = w)$//$\delta^y(z, \sigma) = w$ is defined in H^y;
        replace it by $(v_y = z)$;
    endif;
endfor;
return Q;

Example.

Considering the STS nest formed by the holon aggregations depicted in Fig. 23(a), where

$$\mathbf{v}_b = \{v_x, v_y\}$$

and

$$N_b := ((v_x{}' = x_1) \wedge (v_x = x_2)) \wedge (((v_y{}' = y_0) \wedge (v_y = y_1)) \vee ((v_y{}' = y_1) \wedge (v_y = y_0))).$$

Let $P = (v_x = x_2) \wedge (v_y = y_1)$. We have:

$\underline{\Gamma}(P, b) := \exists \mathbf{v}_b (P \wedge N_b)[\{{v_x}', {v_y}'\} \to \{v_x, v_y\}]$

$\equiv \exists \mathbf{v}_b(({v_x}' = x_1) \wedge (v_x = x_2) \wedge ({v_y}' = y_0) \wedge (v_y = y_1))[\{{v_x}', {v_y}'\} \to \{v_x, v_y\}]$

$\equiv ({v_x}' = x_1) \wedge ({v_y}' = y_0)[\{{v_x}', {v_y}'\} \to \{v_x, v_y\}]$

$\equiv (v_x = x_1) \wedge (v_y = y_0)$. $\square$

D. Coreachability Predicates in STS Nests

Taking the global STS $\mathbf{G}$'s structure information into account, the entrances and exits of an STS nest $\underline{\mathbf{G}}^x$ are addressed in the calculation of *coreachability predicate*. Briefly, we denote

$$\Theta(\mathcal{I}^x) = \bigvee_{T \in \mathcal{I}^x} \Theta(T)$$

and

$$\Theta(\mathcal{O}^x) = \bigvee_{T \in \mathcal{O}^x} \Theta(T).$$

Let $\underline{\mathbf{G}}^x$ be an STS nest in $\mathbf{S}(\mathbf{G})$ and P be a predicate. As stated in Algorithm 3, the coreachability predicate $CR(\underline{\mathbf{G}}^x, P)$ is defined to designate all the basic-state-trees that reach some $b_m \models P \wedge \Theta(\mathcal{O}^x)$ via *basic-state-trees* satisfying P, according to the inductive definition:

1. $\Theta(\mathcal{O}^x) = \perp \Rightarrow CR(\underline{\mathbf{G}}^x, P) = \perp$;
2. $b_m \models \Theta(\mathcal{O}^x) \wedge P \Rightarrow b_m \models CR(\underline{\mathbf{G}}^x, P)$;
3. $b \models CR(\underline{\mathbf{G}}^x, P)$ & $\sigma \in \Sigma_{\mathcal{A}}(x)$ & $\underline{\Gamma}(b, \sigma) = b'$ & $b' \models P \Rightarrow b' \models CR(\underline{\mathbf{G}}^x, P)$; and
4. No other basic-state-trees satisfy $CR(\underline{\mathbf{G}}^x, P)$.

E. Communication Verification

This subsection presents a method to verify the communication of STS nests. Intuitively, an STS nest $\underline{\mathbf{G}}^x$ is communicated if all the basic-state-trees in $\mathcal{O}^x$ can be reached, from any basic-state-tree in $\mathcal{I}^x$, via a sequence of basic-state-trees $\underline{\mathbf{G}}^x$.

Definition 18: [STS nest Communication] An STS nest $\underline{\mathbf{G}}^x$ is communicated if $CR(\underline{\mathbf{G}}^x, \top)$ reaches all of its entrances. Formally, $\underline{\mathbf{G}}^x$ is communicated if

$$(\forall T \in \Theta(\mathcal{I}^x)) T \models CR(\underline{\mathbf{G}}^x, \top).$$

$\diamond$

Algorithm 3 Coreachability predicate $CR(\underline{\mathbf{G}}^x, P)$

Input: A predicate P, an STS nest $\underline{\mathbf{G}}^x$, and its exits $\Theta(\mathcal{O}^x)$.
Output: A predicate Q.
1. Predicate $Q = P \wedge \Theta(\mathcal{O}^x)$;
2. **function** $CR(\underline{\mathbf{G}}^x, Q)$;
3. **for** each σ in $\Sigma_{\mathcal{A}}(x)$;
4. $(\underline{\Gamma}(P, \sigma) \wedge P) \models Q$;
5. **endfor**;
6. **if** $Q = CR(\underline{\mathbf{G}}^x, Q)$;
7. **return** Q;
8. **else**
9. **goto** line 2;
10. **end function**;

Definition 19: [STS Communication] $\mathbf{G}$ is communicated if $(\forall \underline{\mathbf{G}}^x \in \mathbf{S}(\mathbf{G}))\underline{\mathbf{G}}^x$ is communicated. ◇

Example.

STS nest $\underline{\mathbf{G}}^x$ depicted in Fig. 23(b) is communicated since

$$(\forall T \in \Theta(\mathcal{I}^y))T \models CR(\underline{\mathbf{G}}^x, \top)$$

holds. It is easy to check that the STS $\mathbf{G}$ depicted in Figs. 19 and 21 is communicated. □

If an STS is communicated, with properly assigned specifications, its nested optimal non-blocking supervisor is synthesized by following the top-down approach developed in Section VI. Otherwise, the basic modelling principle depicted in Fig. 29 is violated; we need to remodel the STS.

VI. Nested Supervisory Control of STS

A top-down iteration approach is presented to implement the nested supervisory control of communicated STS. This approach guarantees that the lower-level closed-loop (under control) STS nests can be "plugged" into the leaf states in a higher-level STS nest without changing their control logics.

A. Nested Supervisory Control of STS Nests

The optimal behavior C^y of an STS nest $\underline{\mathbf{G}}^y$ is synthesized in a top-down approach. Given

- an STS nest $\underline{\mathbf{G}}^y$ in $\mathbf{S}(\underline{\mathbf{G}})$,
- a specification predicate $P_S{}^y$ containing the *illegal predicate* of $\underline{\mathbf{G}}^y$, and
- the *optimal supremal weakly controllable and coreachable* (i.e., *nonblocking*) behavior of $\underline{\mathbf{G}}^y$ subordinated to

$$C^{\wedge} = \begin{cases} C^x, & \text{if } (\exists \underline{\mathbf{G}}^x)\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y \\ \top, & \text{otherwise} \end{cases},$$

the *nonblocking* subpredicate

$$\sup\mathcal{C}^2\mathcal{P}(\neg P_S{}^y) = C^y$$

of $\underline{\mathbf{G}}^y$ is calculated as follows.

1. A predicate transformer

$$\Omega_P : Pred(\underline{\mathbf{ST}}^y) \rightarrow Pred(\underline{\mathbf{ST}}^y)$$

is defined as

$$\Omega_P(P) = P \wedge CR(\mathbf{G}, \neg[P_S{}^y]).$$

In accordance with [23] and [24], the predicate transformer $[\cdot]$ holds for all the basic state-trees that can reach $P_S{}^y$ by uncontrollable paths only. The pseud-code is given in Algorithm 4.

Algorithm 4 Predicate transformer $[\cdot]$

Input: A predicate P and an STS nest $\underline{\mathbf{G}}^y$.
Output: A predicate Q.
1. Predicates $Q = P$ and $R = P$;
2. **function** $[Q]$;
3. **for** each σ in $\Sigma_{\mathcal{A}}(y) \cap \Sigma_u$;
4. $\underline{\Gamma}(Q, \sigma) \models R$;
5. **endfor**;
6. **if** $Q = R$;
7. **return** Q;
8. **else**
9. **goto** line 2;
10. **end function**;

2. Predicate $\sup\mathcal{C}^2\mathcal{P}(\neg P_S{}^y)$ is calculated iteratively with $K_0 = \neg P_S{}^y$ and $K_{i+1} = \Omega_P(K_i)$. The calculation halts when $K_{i+1} = K_i$. Then,

$$\sup\mathcal{C}^2\mathcal{P}(\neg {P_S}^y) = K_i$$

and

$$C^y = K_i.$$

In accordance with [23] and [24], the *optimal nonblocking* subpredicate

$$C^y = \sup\mathcal{C}^2\mathcal{P}(\neg {P_S}^y)$$

of $\underline{\mathbf{G}}^y$ is synthesized in Algorithm 5.

Algorithm 5 Predicate transformer $\sup\mathcal{C}^2\mathcal{P}(P)$

Input: A predicate P and an STS nest $\underline{\mathbf{G}}^y$.
Output: A predicate C^y.
1. Predicate $K_i = P$;
2. **function** $K_{i+1} = \Omega_P(K_i)$;
3. $\quad\Omega_P(P) = P \wedge CR(\mathbf{G}, \neg[P])$;
4. $\quad$**if** $K_{i+1} = K_i$;
5. $\quad\quad$**return** $C^y = K_i$;
6. $\quad$**else**
7. $\quad\quad K_{i+1} \leftarrow K_i$;
8. $\quad\quad$**goto** line 2;
9. **end function**;

An STS nest $\underline{\mathbf{G}}^y$ in $\mathbf{S}(\mathbf{G})$ under control is denoted by $\underline{\mathbf{G}}^{y,f}$. Based on Definition 20, we can verify if $\underline{\mathbf{G}}^{y,f}$ is communicated.

Definition 20: [Closed-Loop Communication] An STS nest $\underline{\mathbf{G}}^{y,f}$ under control is closed-loop communicated if no available entrances or exists are blocked in the synthesis process C^y. Formally, $\underline{\mathbf{G}}^{y,f}$ is communicated if

$$(\forall b \models \Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y))b \not\models {P_S}^y \Rightarrow b \models C^y.$$

$\diamond$

Remarks:

1. Given an STS nest $\underline{\mathbf{G}}^y$, during the synthesis process of its optimal controlled (closed-loop) behavior $C^y = \sup\mathcal{C}^2\mathcal{P}(\neg {P_S}^y)$, the condition "$\underline{\mathbf{G}}^y$ subordinated to $C^\wedge$" is addressed in $\Theta(\mathcal{I}^y)$ and $\Theta(\mathcal{O}^y)$ since the hierarchical monolithic structure is integrated in them.

2. Suppose that $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$. Naturally, the synthesis of $\underline{\mathbf{G}}^y$ is skipped if the lumped state $\underline{y}$ is not visited by $\underline{\mathbf{G}}^{x,f}$.

3. Suppose that $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$, $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^z$, and $\underline{\mathbf{G}}^y \sim \underline{\mathbf{G}}^z$. Since there are no shared events in $\underline{\mathbf{G}}^y$ and $\underline{\mathbf{G}}^z$, C^y and C^z can be calculated independently. □

B. Optimal Nested Supervisory Control of Global STS

Based on the subordination relation of STS nests, the monolithic behavior C of $\mathbf{G}$ is obtained without tracking its global dynamics, which offers a significant reduction of computational complexity. Let $\mathbf{G}^f$ be an STS $\mathbf{G}$ under control, we present Theorem 2.

Theorem 2: Let $\mathbf{G}^f$ be an STS $\mathbf{G}$ under control. It is closed-loop communicated if for all $\underline{\mathbf{G}}^y$ in $\mathbf{S}(\mathbf{G})$, $\underline{\mathbf{G}}^{y,f}$ is closed-loop communicated with specification P_{S^y} satisfying

$$\neg C^{\wedge} \wedge (\Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y)) \models P_{S^y}.$$

Proof: Clearly, the formula holds for $C^{\wedge} = \top$. Suppose $(\exists \underline{\mathbf{G}}^x)\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$. Then $C^{\wedge} = C^x$ holds. $\neg C^{\wedge} \wedge (\Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y)) \equiv \neg C^x \wedge (\Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y)) \equiv (\neg C^x \wedge \Theta(\mathcal{I}^y)) \vee (\neg C^x \wedge \Theta(\mathcal{O}^y))$ indicates that some I/O of $\underline{\mathbf{G}}^y$ may not be visited by the exosystem $\underline{\mathbf{G}}^{x,f}$. Formula $\neg C^x \wedge (\Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y)) \models P_{S^y}$ represents that specification P_{S^y} inherits the controller behavior of $\underline{\mathbf{G}}^{x,f}$ to consider these I/O of $\underline{\mathbf{G}}^y$ as illegal sub-state-trees in $\underline{\mathbf{G}}^y$.

Moreover, $\underline{\mathbf{G}}^{y,f}$ communication guarantees that the remaining I/O will not be blocked by C^y, which matches the behavior in C^x, which implies that no legal behavior of $\underline{\mathbf{G}}^{y,f}$ will be blocked by $\underline{\mathbf{G}}^{x,f}$. ■

We say that STS $\mathbf{G}^f$ satisfies the boundary consistency of supervisory control [42] if it is closed-loop communicated. In other words, lower-level closed-loop (under control) STS nests can be "plugged" into the leaf states of the high level STS nest (it subordinated to) without changing their control logics.

Theorem 3: Suppose $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$ with $\underline{\mathbf{G}}^x, \underline{\mathbf{G}}^y \in \mathbf{S}(\mathbf{G})$. Their optimal behavior is denoted by C^x and C^y, respectively. The optimal behavior of $\mathbf{G}$, denoted by C, satisfies

$$C = \bigvee_{\underline{\mathbf{G}}^x < \underline{\mathbf{G}}^y} (C^x \wedge C^y) \vee (\neg \Theta(\underline{\mathbf{S}\mathcal{T}}^y) \wedge C^x).$$

Proof: Suppose that in $\underline{\mathbf{G}}^x$, we have $(\exists z \in \underline{X}^x)\mathcal{T}(z) = \mathsf{OR}\ \&\ y \in \mathcal{E}(z)$. Then, we have

$$C^x \equiv ((v_z = y) \wedge C^x) \vee (\neg(v_z = y) \wedge C^x),$$

$$C^y \equiv ((v_z = y) \wedge C^y \wedge C^x) \vee ((v_z = y) \wedge C^y \wedge \neg C^x),$$

and

$$\neg\Theta(\underline{\mathbf{S}T}^y) \wedge C^x = \neg(v_z = y) \wedge C^x.$$

By $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$, we have $C^y \wedge \neg C^x = \perp$. Then, $C^y \equiv ((v_z = y) \wedge C^y \wedge C^x)$ holds. Now we have $C^x \wedge C^y \equiv ((v_z = y) \wedge C^x \wedge C^y)$ to add the internal behavior of $\underline{\mathbf{G}}^y$ to refine the system behavior in $(v_z = y) \wedge C^x$. The system behavior in $\neg(v_z = y) \wedge C^x$ remains unchanged. ■

Briefly, the diagram in Fig. 30 commutes, in which NSC denotes the top-down nested supervisory control.

$$\begin{tikzcd}
\mathbf{G} \arrow[r, shift left, "\text{NSC}"] \arrow[d, "Proj"'] & C \arrow[l, shift left] \arrow[d, "\vee"] \\
\underline{\mathbf{G}}^x \in \mathbf{S}(\mathbf{G}) \arrow[r, dashed, shift left] & C^y \arrow[l, dashed, shift left, "\text{NSC}"]
\end{tikzcd}$$

Fig. 30: Supervisory control commutative diagram.

Now the STS under control with the nested optimal nonblocking behavior is represented by

$$\mathbf{G}^f = (\mathbf{S}T, \mathcal{H}, \Sigma, \Delta, \underline{P_0}^f, \underline{P_m})$$

with $P_0^f \preceq P_0$. Theorem 3 guarantees that, if $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$, by the operation of $\wedge$, the encoded lumped state $\underline{y}$ in C^x, denoted by $\Theta(\underline{y})$, is refined, which is replaced by C^y plugging in more lower-level dynamics.

As stated in [23], the computational complexity of the nonblocking optimal behavior synthesis for STS encoded into predicates is polynomial in the numbers of BDD nodes used for encoding the STS. Since the presented nested optimal synthesis is also based on the predicate transformers sup$\mathcal{C}^2\mathcal{P}(P)$ and $[\cdot]$, the computational complexity of an STS nest's synthesis is polynomial in the BDD nodes in use.

In the worst case, plugging lower-level STS nests into an existing STS nest may lead the number of BDD variables used in the synthesis process to grow exponentially. In comparison, in this study, the number of BDD variables used in the optimal nested synthesis is additive in the numbers of BDD nodes used to synthesize all STS nests' optimal closed-loop behavior.

Hence, the computational complexity reduction for the nested synthesis mainly stem from the decomposition of an STS into STS nests.

In comparison with the previous work [23], [24], [35], [37]–[41], formally, the computational complexity for tracking the behavior of an STS is reduced from $O(n^{d^m})$ (supervisory control) to $\sum_{i=1}^{d \cdot m} O(n)$ (nested supervisory control) where

- n represents the largest number of BDD nodes used to synthesize an STS nest's optimal closed-loop behavior;
- d represents the largest number of STS nests in an hierarchy; and
- m represents the total depths of hierarchies.

Remarks:

1. Essentially, the state feedback control (SFBC) proposed in [23] and [24] is to synthesize the optimal behavior of an STS from an equivalent single level DES encoded into a predicate.

2. As stated in Section VIII-A, the optimal nested behavior of an STS differs from its optimal behavior synthesized by the supervisory control of STS proposed in [23] and [24]. □

VII. Specification Management and Controller Implementations

This section provides an approach to allocate the user-defined specifications to STS nests automatically. For any STS nest, *mutual exclusion* and *event occurrence preventing* problems [23], [24] are addressed in its specifications.

A. STS Specification Partition and Management

In accordance with STS [23], [24], the specifications of an STS cover the *event occurrence prevention problem* and *mutual exclusion problem*. Let $\sigma \in \Sigma$ and $i, j \in I$ with I as an index set. The specifications are described as

$$\mathcal{S}\text{: } \{(T_i, \sigma)\} \text{ and } \mathbf{T}_{il} = \{T_{il,j}\}$$

where (T_i, σ) represents that event σ is disabled at a sub-state-tree T_i in $Pred(\mathbf{S}T)$. Moreover, $T_{il,j}$ is an illegal sub-state-tree with $j = 1, 2, \ldots$. The state-trees are described by their key leaf state sets.

1) Specification partitions:

Generally, a specification should not cross STS nests. Let $\underline{\mathbf{G}}^x$ be an STS nest. Specification $\mathcal{S}^x$ (as a subset of $\mathcal{S}$) is generated automatically as follows. Given a specification with respect to an STS $\mathbf{G}$: $\{(T_i, \sigma)\}$ and $\mathbf{T}_{il} = \{T_{il,j}\}$, specification (T_i, σ) belongs to $\mathcal{S}^x$ if $\mathcal{V}(T_i) \subseteq X_{\mathcal{A}}(x)$ and $\sigma \in \Sigma_{\mathcal{A}}(x)$. Similarly, $T_{il,j}$ belongs to $\mathcal{S}^x$ if $\mathcal{V}(T_{il,j}) \subseteq X_{\mathcal{A}}(x)$.

Remark:

Given any two STS nests $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$, as a natural extension, we only allow user-defined specifications cross different STS nests in the following two cases.

1. A specification crosses $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ if $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$. In this case, $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ are merged with superstate x as its root.

2. A specification crosses $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ if $\underline{\mathbf{G}}^x \sim \underline{\mathbf{G}}^y$. In this case, $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ should be synchronized as one, denoted as $\underline{\mathbf{G}}^{x+y}$, in which we have $\Sigma_{\mathcal{A}}(x+y) = \Sigma_{\mathcal{A}}(x) \cup \Sigma_{\mathcal{A}}(y)$, $\mathcal{V}(\underline{\mathbf{S}T_0}^{x+y}) = \mathcal{V}(\underline{\mathbf{S}T_0}^{x}) \cup \mathcal{V}(\underline{\mathbf{S}T_0}^{y})$, and $\mathcal{V}(\underline{\mathbf{S}T_m}^{x+y}) = \mathcal{V}(\underline{\mathbf{S}T_m}^{x}) \cup \mathcal{V}(\underline{\mathbf{S}T_m}^{y})$.

Other cross-STS-nest specifications are invalid. Users need to reassign them such that they satisfy the two cases above. For the sake of simplicity, the rest of this paper only consider the general case. Clearly, there is no technical problem to extend to the two special cases given above. $\square$

2) Specification management:

Given an STS nest $\underline{\mathbf{G}}^y$ with specifications $\mathcal{S}^y$: $\{(T_i^y, \sigma)\}$ and $\mathbf{T}_{il}^y = \{T_{il,j}^y\}$ with $i, j \in I$ in which I is an index set. The specifications are managed as follows:

1. Mutual exclusion specifications are identified by the illegal predicate ${P_S}^y$. Formally,

$$\Theta(\mathbf{T}_{il}^y) \models {P_S}^y.$$

2. Preventing the occurrences of uncontrollable events is managed in a similar approach. Given a pair (T_i^y, σ) with $\sigma \in \Sigma_u$, we have a sub-state-tree

$$T = \Theta(Elig_{\underline{\mathbf{G}}^y}(\sigma)) \wedge T_i^y$$

that is identified by ${P_S}^y$. Formally,

$$\Theta(Elig_{\underline{\mathbf{G}}^y}(\sigma)) \wedge T_i^y \models {P_S}^y.$$

3. According to Theorem 2, the specification ${P_S}^y$ is updated to satisfy

$$\neg C^{\wedge} \wedge (\Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y)) \models P_S{}^y.$$

4. Preventing the occurrences of controllable events is managed by the *maximally permissive predicates* defined below.

Definition 21: [Maximally Permissive Predicate] Given a set of pairs $\{(T_i^y, \sigma) | \sigma \in \Sigma_c\}$, the maximally permissive predicate for event σ is defined as $\mathsf{MP}(\sigma) = \bigwedge_i \neg T_i^y$. $\diamond$

As stated in Algorithm 6, the specifications in $\mathbf{T}_{il}^y$ and those in $\{(T_i^y, \sigma) | \sigma \in \Sigma_u\}$ are managed by following the approach proposed in [23] and [24]. *Maximally permissive predicates* are developed for specifications $\{(T_i^y, \sigma) | \sigma \in \Sigma_c\}$ to disable event σ in STS nest $\underline{\mathbf{G}}^y$ directly.

Algorithm 6 Specification management

Input: Specifications $\mathcal{S}^y$: $\{(T_i^y, \sigma)\}$ and $\mathbf{T}_{il}^y = \{T_{il,j}^y\}$.
Output: $P_S{}^y$ and $\{\mathsf{MP}(\sigma) | \sigma \in \Sigma_{\mathcal{A}}(y) \cap \Sigma_c,\ \sigma \in \{(T_i^y, \sigma)\}\}$.
$P_S{}^y = \perp$;
for each σ in $\{(T_i^y, \sigma)\}$;
    $\mathsf{MP}(\sigma) = \top$;
endfor;
for each $T_{il,j}^y$ in $\mathbf{T}_{il}^y$; //According to [23] and [24].
    $\Theta(T_{il,j}^y) \models P_S{}^y$;
endfor;
for each σ in $\{(T_i^y, \sigma)\}$;
    if $\sigma \in \Sigma_u$; //According to [23] and [24].
        $\Theta(Elig_{\underline{\mathbf{G}}^y}(\sigma)) \wedge T_i^y \models P_S{}^y$;
    else
        $\mathsf{MP}(\sigma) = (\neg T_i^y) \wedge \mathsf{MP}(\sigma)$;
endfor;
$\neg C^{\wedge} \wedge (\Theta(\mathcal{I}^y) \vee \Theta(\mathcal{O}^y)) \models P_S{}^y$; //Theorem 2.
return $P_S{}^y$ and$\{\mathsf{MP}(\sigma)\}$;

Let $\sigma \in \Sigma_c$. The maximally permissive predicate $\mathsf{MP}(\sigma)$ contains the sub-state-trees where σ is allowed to occur. The *backward preliminary-control transition structure* is defined as

$$\underline{\Gamma}^{y,Pr} : \mathcal{S}T(\underline{\mathbf{S}T}^y) \times \Sigma_{\mathcal{A}}(x) \rightarrow \mathcal{S}T(\underline{\mathbf{S}T}^y)$$

with respect to

$$\underline{\Gamma}^{y,Pr}(b, \sigma) := \begin{cases} \mathsf{MP}(\sigma) \wedge \underline{\Gamma}^y(b, \sigma), & \text{if } \sigma \in \Sigma_c \\ \underline{\Gamma}^y(b, \sigma), & \text{if } \sigma \in \Sigma_u \end{cases}.$$

With the maximally permissive predicate $\mathsf{MP}(\sigma)$ addressed, Γ is replaced by $\underline{\Gamma}^{y,Pr}$ in the synthesis procedure.

Example.

Consider the STS $\mathbf{G}$ depicted in Figs. 19 and 21. Suppose that $a \in \Sigma_c$ and $\mu \in \Sigma_u$. Specifications $(\{x_1, y_1\}, a)$, $(\{2, 5\}, \mu)$, and $\mathbf{T}_{il} = \{\{3, 4\}\}$ are handled according to Lines 12, 10, and 6 in Algorithm 6, respectively. In particular, Line 10 converts "preventing the occurrence of an uncontrollable event at a sub-state-tree" into an illegal sub-state-tree. □

B. Controller Implementations

The control functions f_σ of an STS nest's controllable events are calculated with the hierarchical structure of STS addressed, which is based on the monolithic largest next-state-tree $Next_{\mathbf{G}}(\sigma)$. For an STS nest $\underline{\mathbf{G}}^y$ in $\mathbf{S}(\underline{\mathbf{G}})$, if the closed-loop system under control is nonempty, i.e., $\underline{P_0}^y \wedge C^y \not\equiv \perp$, the control of a controllable event σ in Σ_c is implemented based on the set of SFBC predicates (control functions) f_σ for $\sigma \in \Sigma_{\mathcal{A}}(y) \cap \Sigma_c$. Similar to [24], let

$$N_{good} := \Theta(Next_{\mathbf{G}}(\sigma)) \wedge C^{\wedge}$$

be the global legal subpredicate of $\Theta(Next_{\mathbf{G}}(\sigma))$. We obtain the *weakest* SFBC for events $\sigma \in \Sigma_c$ as

$$f_\sigma := \Gamma(N_{good}, \sigma) \vee \neg C^{\wedge},$$

in which $Next_{\mathbf{G}}(\sigma)$ and Γ are defined in [23] and [24].

Let $\sigma \in \Sigma_c \cap \Sigma_{\mathcal{A}}(y)$. The closed-loop transition function for $\underline{\mathbf{G}}^y$, induced by the *weakest* SFBC f_σ, is given by

$$\underline{\Delta}^f(b, \sigma) := \begin{cases} \underline{\Delta}(b, \sigma), & \text{if } f_\sigma(b) = 1 \\ \emptyset, & \text{if } f_\sigma(b) = 0 \end{cases}.$$

An STS nest $\underline{\mathbf{G}}^y$ under control is represented by

$$\underline{\mathbf{G}}^{y,f} = (\underline{\mathbf{S}T}^y, \mathcal{H}_{\mathcal{A}}(y), \Sigma_{\mathcal{A}}(y), \underline{\Delta}^f, \underline{P_0}^{y,f}, \underline{P_m}^y)$$

with $\underline{P_0}^{y,f} \preceq \underline{P_0}^y$.

Theorem 4: Let $\underline{\mathbf{G}}^y \in \mathbf{S}(\mathbf{G})$ and $\sigma \in \Sigma_{\mathcal{A}}(y) \cap \Sigma_c$. The control function f_σ can be used to supervise both $\underline{\mathbf{G}}^y$ and the global STS $\mathbf{G}$ directly without any change.

Proof: The I/O of an STS nest $\underline{\mathbf{G}}^y$ is utilized to denote its static hierarchical location in STS $\mathbf{G}$. According to the predicates depicted in Fig. 31, our proof contains three parts:

1) Given a holon H^z with $\underline{H}^z$ belonging to the holon aggregation $\mathcal{H}_{\mathcal{A}}(y)$ of STS nest $\underline{\mathbf{G}}^y$, suppose $\sigma_1 \in \Sigma_c \cap \Sigma_I^z$. The weakest control function f_{σ_1} is calculated by $f_{\sigma_1} := \Gamma(N_{good}, \sigma) \vee \neg C^\wedge$ with $N_{good} := \Theta(Next_{\mathbf{G}}(\sigma_1)) \wedge C^\wedge$. Then $Next_{\mathbf{G}}(\sigma_1)$ and $Next_{\underline{\mathbf{G}}^y}(\sigma_1)$ have the same structure for both $\mathbf{G}$ and $\underline{\mathbf{G}}^y$. Hence the weakest control function f_{σ_1} can be used to supervise both $\underline{\mathbf{G}}^y$ and the global STS $\mathbf{G}$ directly without any change.

2) Suppose $\sigma_2 \in \Sigma_c \cap \Sigma_{BI}^z$. The source state of event σ_2 in $\mathbf{G}$ and $\underline{\mathbf{G}}^y$ has the same structure. Hence we obtain the same conclusion for both $\mathbf{G}$ and $\underline{\mathbf{G}}^y$.

3) Suppose $\sigma_3 \in \Sigma_c \cap \Sigma_{BO}^z$. The source state of event σ_3 in $\underline{\mathbf{G}}^y$ is an exit, which is a sub-state-tree of $Next_{\mathbf{G}}(\sigma_3)$. Since the exits are addressed while calculating sup$\mathcal{C}^2\mathcal{P}(\neg P_S{}^y)$, we reach the same conclusion for both $\mathbf{G}$ and $\underline{\mathbf{G}}^y$. ■

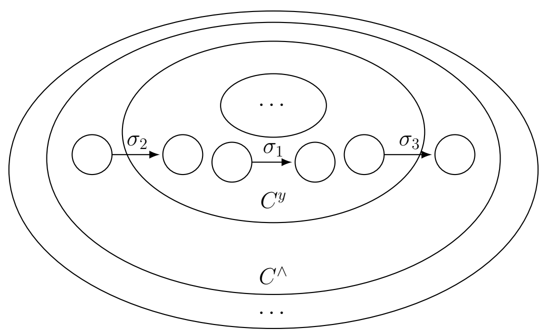


Fig. 31: Events cross predicates.

As shown in Fig. 32, in an STS nest $\underline{\mathbf{G}}^y$, its optimal behavior is recorded in $C^y = $ sup$\mathcal{C}^2\mathcal{P}(\neg P_S{}^y)$ that is viewed as an agent $\underline{\mathbf{G}}^y_{tracker}$. With respect to the specification for $\underline{\mathbf{G}}^y$, according to the optimal behavior $C^\wedge$ and the current status (a basic state-tree b), a set of decision makers f_{σ_i}, provided by $\underline{\mathbf{G}}^y_{tracker}$, with $\sigma_i \in \Sigma_c \cap \Sigma_{\mathcal{A}}(y)$, $i = 1, 2, \ldots, n$, makes the decisions applying b as the argument. If $f_{\sigma_i}(b) = 1$, then σ_i is allowed to occur. Otherwise, it is disabled.

Alternatively, the optimal behavior of STS $\mathbf{G}$ can be calculated in another approach: $C = CR^f(\mathbf{G}, \top)$, where CR^f represents that the control function of all the controllable events are addressed.

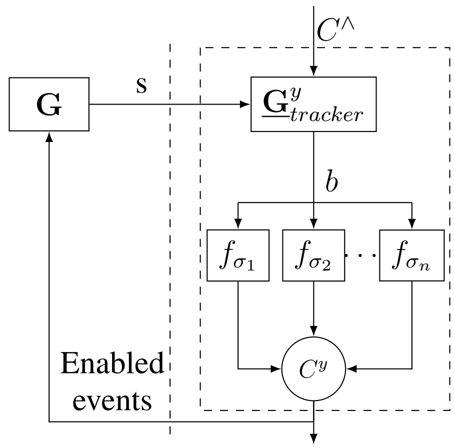


Fig. 32: Nested STS control diagram.

VIII. Case Studies

Two case studies are presented in this section to demonstrate the nested supervisory control of STS.

A. Transfer Line

We take the transfer line [4], [23], [24] shown in Fig. 33 as an example. It is assumed that the capacities of the buffers B1 and B2 are both one and the controllable and uncontrollable events are denoted by odd and even numbers, respectively.

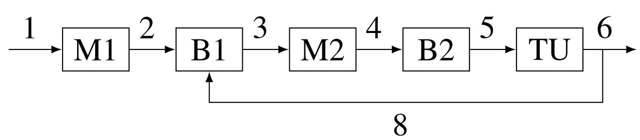


Fig. 33: Transfer line.

1) STS nests:

Isomorphic with the transfer line example studied [23] and [24], the top-level holons are depicted in Fig. 34. The corresponding state-tree is shown in Fig. 35.

Suppose that the hierarchical behavior of machine M1 (resp., M2) is described by two holons, in which the operations in state $M1_1$ (resp., $M2_1$) are constructed in the lower-level holon. Hence, as illustrated in Fig. 36, the latter plugs more operation details in. The global state-tree (rooted

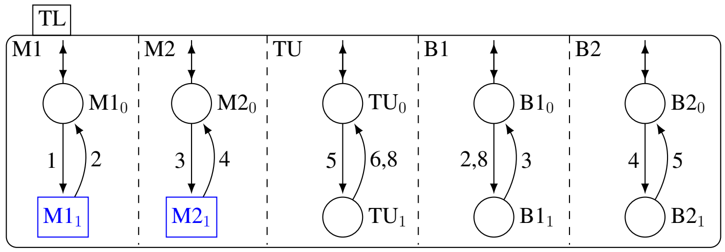


Fig. 34: Holons in STS nest $\underline{\mathbf{G}}^{\mathrm{TL}}$.

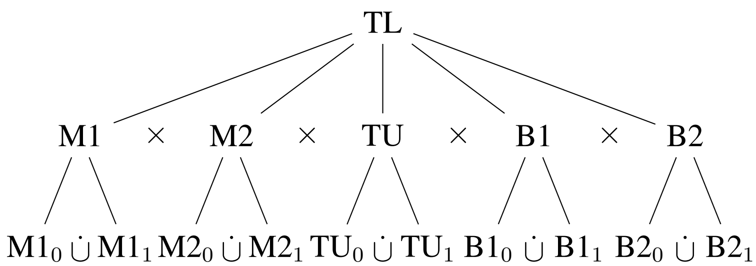


Fig. 35: State-tree of transfer line.

by an AND superstate TL) matching the holons shown in Fig. 36 is depicted in Fig. 37. This STS is decomposed into three STS nests $\underline{\mathbf{G}}^{\mathrm{TL}}$, $\underline{\mathbf{G}}^{\mathrm{M1_1}}$, and $\underline{\mathbf{G}}^{\mathrm{M2_1}}$ satisfying

- $\underline{\mathbf{G}}^{\mathrm{TL}} <_N \underline{\mathbf{G}}^{\mathrm{M1_1}}$,
- $\underline{\mathbf{G}}^{\mathrm{TL}} <_N \underline{\mathbf{G}}^{\mathrm{M2_1}}$, and
- $\underline{\mathbf{G}}^{\mathrm{M1_1}} \sim \underline{\mathbf{G}}^{\mathrm{M2_1}}$,

in which $\underline{\mathbf{G}}^{\mathrm{M1_1}}$ and $\underline{\mathbf{G}}^{\mathrm{M2_1}}$ are marked in blue. Clearly, the global STS $\mathbf{G}$ is communicated since STS nests $\underline{\mathbf{G}}^{\mathrm{TL}}$, $\underline{\mathbf{G}}^{\mathrm{M1_1}}$, and $\underline{\mathbf{G}}^{\mathrm{M2_1}}$ are communicated.

2) Nested supervisory control v.s. supervisory control:

By abstracting $\underline{\mathbf{G}}^{\mathrm{M1_1}}$ and $\underline{\mathbf{G}}^{\mathrm{M2_1}}$, the nested nonblocking supervisory control and the nonblocking supervisory control [23], [24] of $\underline{\mathbf{G}}^{\mathrm{TL}}$ (depicted in Fig. 34) are identical:

- event 1 is enabled at: $f_1 = \{\{\mathrm{M2_0}, \mathrm{TU_0}, \mathrm{B1_0}, \mathrm{B2_0}\}\}$,
- event 3 is enabled at: $f_3 = \{\{\mathrm{B1_1}\}\}$, and
- event 5 is enabled at: $f_5 = \{\{\mathrm{B2_1}\}\}$.

The control parterns show that:

- event 1 is allowed to occur only when machine $M2$ is idle;
- event 3 is allowed to occur only when buffer $B1$ is occupied; and

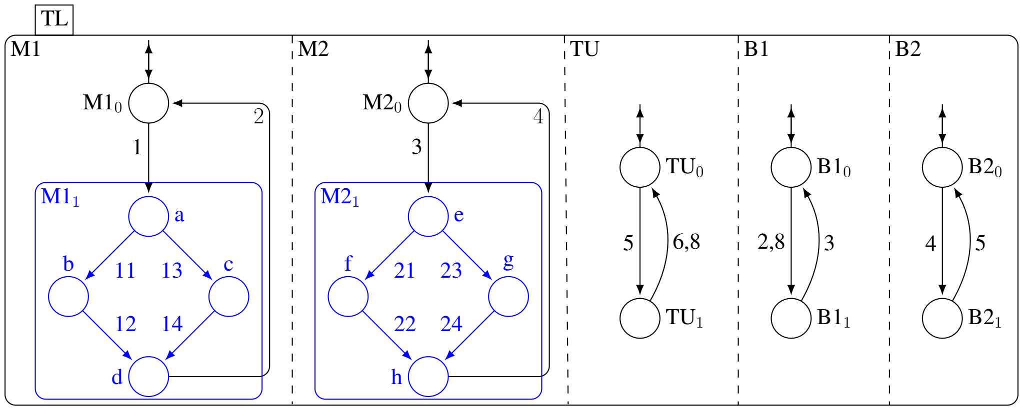


Fig. 36: Holons of transfer line.

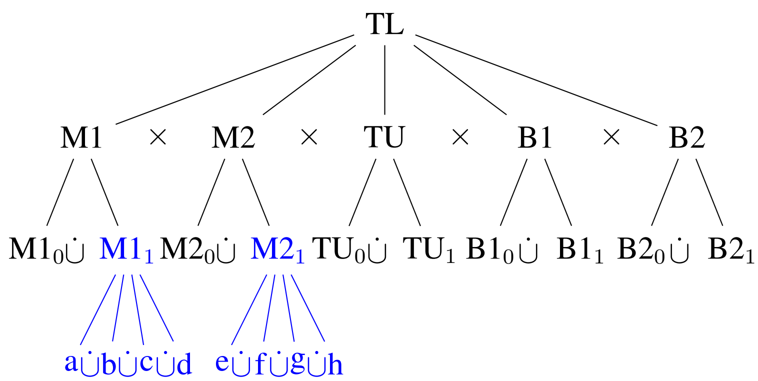


Fig. 37: State-tree of transfer line.

- event 5 is allowed to occur only when buffer $B2$ is occupied.

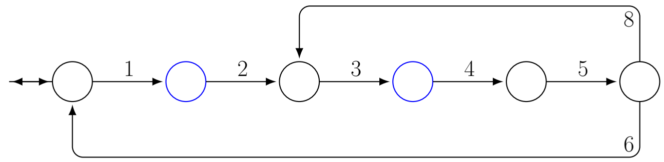
Based on the developed nested supervisory control, the lower-level control functions

$$f_{11} = f_{13} = f_{21} = f_{23} = \top$$

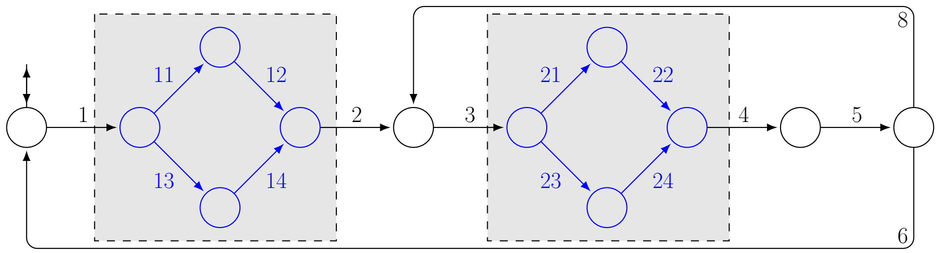
are obtained afterwards, which means that, in $\underline{\mathbf{G}}^{\mathrm{M1}_1}$ and $\underline{\mathbf{G}}^{\mathrm{M2}_1}$, events 11, 13, 21, and 23 are always enabled.

Figs. 38(a) and 38(b) depict the nested optimal (closed-loop) nonblocking behavior of the top-level STS nest $\underline{\mathbf{G}}^{\mathrm{TL}}$ and the global transfer line, respectively, in which the optimal behavior in the lower-level STS nests $\underline{\mathbf{G}}^{\mathrm{M1}_1}$ and $\underline{\mathbf{G}}^{\mathrm{M2}_1}$ are marked in blue. Fig. 38(b) shows that the nested optimal behavior $\underline{\mathbf{G}}^{f}$ is represented by 12 basic-state-trees and 15 transitions.

In parallel, by the supervisory control of STS proposed in [23] and [24], as for the global



(a) Optimal behavior of $\underline{\mathbf{G}}^{\mathrm{TL}}$.



(b) Nested optimal behavior of $\mathbf{G}$.

Fig. 38: Nested optimal behavior of Transfer Line.

STS shown in Figs. 36 and 37, all the synthesized control functions are changed to be:

- event 1 is always enabled: $f_1 = \top$,
- event 3 is enabled at: $f_3 = \{\{\mathrm{B1}_1\}\}$,
- event 5 is enabled at: $f_5 = \{\{\mathrm{M1}_0, \mathrm{B1}_0, \mathrm{B2}_1\}, \{\mathrm{M1}_1, \mathrm{B1}_0, \mathrm{B2}_1, \mathrm{a}\}\}$,
- events 11 and 13 are enabled at: $f_{11} = f_{13} = \{\{\mathrm{TU}_0, \mathrm{B1}_0, \mathrm{B2}_0\}, \{\mathrm{M2}_0, \mathrm{TU}_0, \mathrm{B1}_0, \mathrm{B2}_1\}\}$, and
- events 21 and 23 are enabled at: $f_{21} = f_{23} = \{\{\mathrm{B1}_0, \mathrm{B2}_0\}, \{\mathrm{B1}_1\}\}$.

Control function f_1 shows that event 1 is always enabled. However, from the perspective of the top-level STS nest $\underline{\mathbf{G}}^{\mathrm{TL}}$, this control logic may lead to blocking: while the test unit TU is occupied, the occurrence of event 1 may lead blocking happens in buffer B1 since events 2 and 8 are uncontrollable. While TU is occupied, as a solution, instead of disabling event 1, the global control functions f_{11} and f_{13} “pause” the lower-level processes in STS nest $\underline{\mathbf{G}}^{\mathrm{M1}_1}$. More precisely, f_{11} and f_{13} allow events 11 and 13 to occur if:

- buffers B1 and B2 and test unit TU are empty; or
- machine M2 is at the initial state, test unit TU and buffer B1 are empty, and buffer B2 is occupied.

The closed-loop behavior of the transfer line shown in Figs. 36 and 37 contains 56 basic-state-trees and 126 transitions.

As a comparison, the BDD nodes of the control functions for the two approaches are listed in Table III under “NSC” and “SC”, respectively. Here NSC and SC are the abbreviations of nested supervisory control and supervisory control, respectively.

TABLE III: BDD nodes of controllers

Event	NSC	SC
1	4	0
3	1	1
5	1	5
11	0	4
13	0	4
21	0	2
23	0	2

Clearly, the supervisory control of STS [23], [24] violates the closed-loop communication and the boundary consistency of supervisory control [42], which may result in redundant cross-STS-nest control logic. For instance, since $\underline{\mathbf{G}}^{\mathrm{M1}_1}$ has no shared events with $\underline{\mathbf{G}}^{\mathrm{TL}}$, it is not necessary for its control function f_{11} to supervise (or observe) the system behavior in M2, B1, B2, and TU. Hence, unnecessary (cross-level) concurrent behavior may appear in the global closed-loop (under control) behavior.

3) BDD representation of predicates:

In the STS framework, the predicates of an STS are encoded into BDD. According to [23], the computational complexity of supervisor synthesis is polynomial in the number of BDD nodes in use. Usually, it is much smaller than the states of an STS, i.e., $|nodes| \ll |states|$.

As proposed in [34] and [35], the states in the state set X^x of a holon H^x are encoded by BDD nodes (variables). Consider a state set X^x with a state space $|X^x| = N$. Each element y in X^x is encoded as a vector of n binary values, where $n = \lceil log_2 N \rceil$. The encoding process is denoted by a function $f : X^x \to \{0,1\}^n$ that maps each element y in X^x to a distinct n-bit binary vector. According to [23], the n variables are denoted by x_i with $0 \leq i < n$.

For simplification, we show the BDD representation for the supervisory control of the transfer line. As shown in Fig. 34, there are two states in holon H^{M1}, i.e., $X^{M_1} = \{M1_0, M1_1\}$. As a consequence, one BDD nodes $M1$ is required. For example, let $M1:0$ and $M1:1$ denote that $M1$ is encoded as 0 and 1, respectively. The encoding pairs for the states in the transfer line example are shown in Table IV.

TABLE IV: BDD vectors encoding states

state	BDD vector
$M1_0$	$< M1 : 0 >$
$M1_1$	$< M1 : 1 >$
$M2_0$	$< M2 : 0 >$
$M2_1$	$< M2 : 1 >$
TU_0	$< TU : 0 >$
TU_1	$< TU : 1 >$
$B1_0$	$< B1 : 0 >$
$B1_1$	$< B1 : 1 >$
$B2_0$	$< B2 : 0 >$
$B2_1$	$< B2 : 1 >$

The supervisory control functions of events 1, 3, and 5, denoted by f_1, f_3, and f_5, respectively. The truth table for these control functions is obtained, as shown in Table V, where each "$*$" denotes a variable that can be assigned 0 or 1. □

TABLE V: Truth table of control fucntions

control functions	$M1$	$M2$	TU	$B1$	$B2$
f_1	$*$	1	1	1	1
f_3	$*$	$*$	$*$	1	$*$
f_5	$*$	$*$	$*$	$*$	1

B. AIP Example [23], [24]

The developed nested supervisory control is implemented for the large scale example AIP studied in [23] and [24]. The diagram of the AIP is depicted in Fig. 39, which has five conveyor loops: one central loop communicates with four external loops by four transfer units. Linked to the external loops are three assembly stations and an I/O station. The primary DES model of AIP studied in [46] is the synchronous product of 100 automata with a state space up to 10^{24}.

The total state space of the STS is reduced from an exponential (for supervisory control) to an additive (for nested supervisory control) relation of STS nests' state spaces. Based on the developed nested supervisory control, we obtain 36 different STS nests on three levels of hierarchy. The state-space of the top-level STS nest is around 2×10^{18}, and the state-space of other STS nests are around 10^2. The latter can be ignored. Hence, the computational cost for the nested synthesis is polynomial in the number of BDD nodes used for encoding the top-level STS nest. With a suitable ordering of the BDD variables, not more than $5,098,978$ BDD nodes are used in the synthesis process for any STS nest.

The nested optimal nonblocking supervisor synthesis is finished in several seconds on a personal computer with 2.40 GHz Intel CPU and 8G RAM. The BDD nodes of the local control functions for several important controllable events are listed in Table VI to compare with the AIP studied in [23] and [24], in which the BDD size 0 represents that the corresponding event is allowed to occur when it is eligible. In Table VI, NSC and SC represent nested supervisory control and supervisory control, respectively.

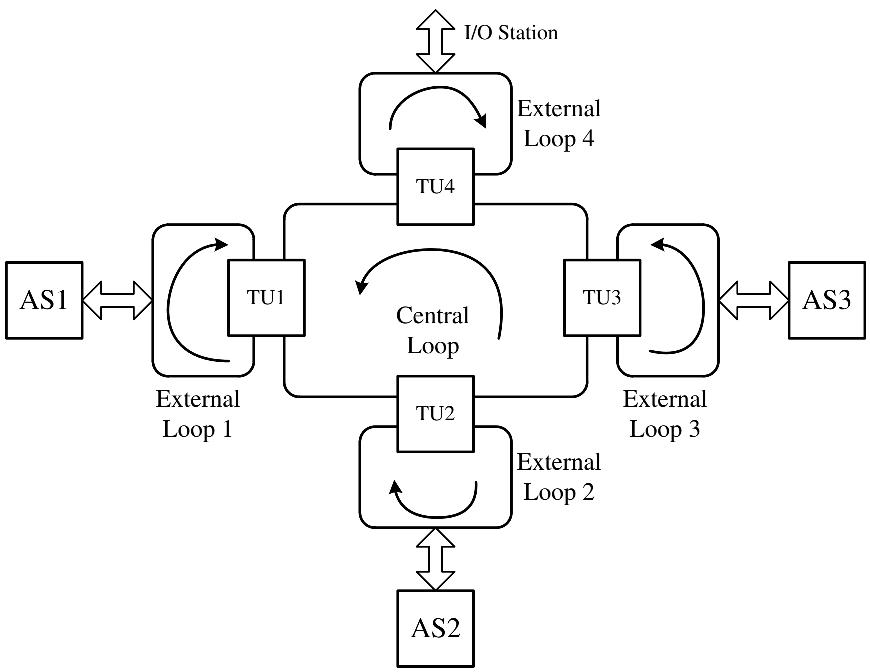


Fig. 39: AIP diagram.

IX. IMP

Suppose $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$ with $\underline{\mathbf{G}}^x, \underline{\mathbf{G}}^y \in \mathbf{S}(\mathbf{G})$. The relation between $\underline{\mathbf{G}}^{x,f}$ and $\underline{\mathbf{G}}^{y,f}$ falls into the application sphere of IMP of control theory [43]–[45].

IMP is a general modelling principle for a wide range of dynamic systems, which requires that: a good controller incorporates a model of the dynamics that generates the signals that the control system is intended to track. In other words, the controller contains a model of its exosystem (outside world). IMP should satisfy the following three principles:

- The autonomous controller's dynamics under the condition of perfect regulation (tracking);
- The controller dynamics of a system is a copy of that of the exosystem dynamics; and
- This copy is "faithful", namely incorporates fully the exosystem dynamics.

Not as our main contribution, but it is interesting to show that the nested supervisory control of STS satisfies IMP in a two-fold significance.

A. IMP-like among nests

Given any communication STS $\mathbf{G}$, there exists an observer to project from $\mathbf{G}$ to any STS nest in $\mathbf{S}(\mathbf{G})$. Suppose that such two projections $\underline{\mathbf{G}}^x$ and $\underline{\mathbf{G}}^y$ satisfy $\underline{\mathbf{G}}^x <_N \underline{\mathbf{G}}^y$ with $\underline{\mathbf{G}}^x, \underline{\mathbf{G}}^y \in \mathbf{S}(\mathbf{G})$, i.e., $\underline{\mathbf{G}}^x$ is the exosystem of $\underline{\mathbf{G}}^y$. We say that $\underline{\mathbf{G}}^{x,f}$ and $\underline{\mathbf{G}}^{y,f}$ satisfy the IMP-like property if:

- $\underline{\mathbf{G}}^{y,f}$ is under the condition of perfect regulation (tracking); and
- $\underline{\mathbf{G}}^{y,f}$ has no influences on $\underline{\mathbf{G}}^{x,f}$ but not the other way around.

Theorem 5: The STS nests in a closed-loop communicated STS $\mathbf{G}^f$ satisfy IMP-like property.

Proof: This can be proved directly from Theorem 2. ■

TABLE VI: BDD size of controller functions for AIP

Event	NSC	SC [23], [24]
ASi_repaired ($i = 1, 2$)	0	0
ASi_stop_close ($i = 1, 2$)	9	1
ASi_stop_open ($i = 1, 2$)	9	16
ASi_gate_open ($i = 1, 2$)	0	0
ASi_read ($i = 1, 2$)	0	0
AS1_pickup3	0	15
AS1_pickup4	0	15
AS3_gate_open	0	0
AS3_read	2	2
L1_gate_open	44	95
CL_TUi_gate_open ($i = 1, 2$)	37	70
CL_TU1_stop_close	20	28
CL_TU2_stop_close	19	36
TUi_Drw2Li($i = 1, 2$)	0	54

B. IMP with STS nests as exosystems

If we consider all the STS nests as the exosystems of an STS $\mathbf{G}$, then we find that a closed-loop communicating $\mathbf{G}^f$ satisfies IMP.

Theorem 6: A closed-loop communicating STS $\mathbf{G}^f$ satisfies IMP.

Proof: Given an STS $\mathbf{G}$. Consider all the STS nests $\underline{\mathbf{G}}^y \in \mathbf{S}(\mathbf{G})$ as the exosystem. Then, there exists a unique mapping from $\mathbf{G}$ to $\underline{\mathbf{G}}^y$. According to Theorem 2, we know that for all $\underline{\mathbf{G}}^y \in \mathbf{S}(\mathbf{G})$, $\underline{\mathbf{G}}^{y,f}$ is closed-loop communication. Hence the first IMP principle is satisfied. According to Theorem 4, the second and third IMP principles are satisfied automatically. ■

X. Conclusion

As an extension of supervisory control of STS, this study focuses on the nested supervisory control of STS in a general approach, which can be applied to a wide range of domains such as manufacturing systems, traffic systems, database management systems, communication protocols, logistic (service) systems, and real-time scheduling. We formally decompose an STS into a set of STS nests that describe its system behavior on different levels of hierarchy. Thereafter, communication of STS nests is presented, which requires that the paths in an STS nest should lead the system from any initial state-tree to all the terminal state-trees. For any communicating STS, instead of synthesizing the global optimal supervisor, its nested optimal nonblocking supervisor is synthesized in a top-down approach. Finally, the global optimal behavior is obtained without synthesizing its global system behavior, which offers a significant reduction of computational complexity.

An STS under control satisfies the boundary consistency of supervisory control proposed in [42] if it is closed-loop communicated. This shows that the lower-level closed-loop (under control) STS nests can be "plugged" into the states of a higher-level STS nest without changing its control functions (control logics). The control functions of STS nests are calculated with the hierarchical structure of STS addressed. Finally, we prove that the control functions for controllable events can be applied to both STS nests and the monolithic STS without any change.

For an STS nest, the computational complexity of the presented supervisor synthesis is polynomial in the BDD nodes in use. For an STS with several subordinates, according to [23], by the supervisory control, the number of BDD *variables* grows exponentially in the number of BDD variables. In this study, for the developed nested supervisory control of STS, the total number of BDD variables in use is reduced from exponential to additive costs.

Two case studies are presented in this research to demonstrate the nested supervisory control of STS. For the STS model of the AIP in [23], [24], [46], it has originally a state space up to 10^{24}. In this study, it is decomposed into 36 different STS nests on three levels of hierarchy. As a result, the total state space of all the 36 STS nests is reduced to around 2×10^{18}. With a suitable ordering of the BDD variables used in the nested nonblocking optimal supervisor synthesis, not more than $5,098,978$ BDD nodes are used in the synthesis process for any STS nest. In the future, we will work on the nested supervisory control of state-tree structures with partial observations.

References

[1] H. Marchand and B. Gaudin, "Supervisory control problems of hierarchical finite state machines," in *Proc. 41st IEEE Conf. on Dec. and Cont.*, pp. 1199–1204, 2002.

[2] B. Gaudin and H. Marchand, "Supervisory control of product and hierarchical discrete event systems," *European Journal of Control*, vol. 10, no. 2, pp. 131–145, 2004.

[3] B. Gaudin and H. Marchand, "Safety control of hierarchical synchronous discrete event systems: A state-based approach," in *Proc. IEEE Intern. Symp., Medit. Conf. Cont. Autom. Intel. Cont.*, pp. 889–895, 2005.

[4] W. M. Wonham and K. Cai, *Supervisory control of discrete-event systems*, Monograph Series Communications and Control Engineering, Springer, 2018.

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

[6] R. Alur, S. Kannan, M. Yannakakis, "Communicating hierarchical state machines," *International Colloquium on Automata, Languages, and Programming*. pp. 169–178, Springer, Berlin, Heidelberg, 1999.

[7] R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T. Reps, and M. Yannakakis, "Analysis of recursive state machines," *ACM Trans. Program. Lang. Syst.*, vol. 27, no. 4, pp. 786–818, 2005.

[8] R. Alur and P. Madhusudan, "Visibly pushdown languages," in *Proc. 36th ACM Symp. Theory Comput.*, pp. 202–211, 2004.

[9] R. Alur, V. Kumar, P. Madhusudan, and M. Viswanathan, "Congruences for visibly pushdown languages," in *Proc. 32nd International Colloquium on Automata, Lang, Program.*, pp. 1102–1114, 2005.

[10] R. Alur and P. Madhusudan, "Adding nesting structure to words," *Journal of the ACM*, vol. 56, no. 3, pp. 1–43. 2009.

[11] R. Alur, S. Chaudhuri, K. Etessami, and P. Madhusudan. "On-the-fly reachability and cycle detection for recursive state machines," in *Proc. 11th Intern. Conf. Tools Algor. Constr. Anal. Syst.*, 2005.

[12] P. Gohari and W. M. Wonham, "A linguistic framework for controlled hierarchical DES," in *Proc. 4th Int. Workshop on Dis. Event Syst.*, 1998, pp. 207–212.

[13] F. Maraninchi and Y. Rémond, "Argos: an automaton-based synchronous language," *Comp. Languages*, vol. 27, no. 1–3, pp. 61–92, 2001.

[14] G. Delaval, H. Marchand, and E. Rutten, "Contracts for modular discrete controller synthesis," in *Proc.* of *Languages, Comp., Tools Theo. Embedd. Syst,* pp. 57–66, 2010.

[15] K. C. Wong and W. M. Wonham, W. M. "Hierarchical control of discrete-event systems," *Discrete Event Dyn. Syst.*, vol. 6, no. 3, pp. 241–273, 1996.

[16] L. Grigorov, "Hierarchical control of discrete-event systems," *Compilation Document*, June, no. 29, 2005.

[17] P. Hubbard, and P. E. Caines, "Dynamical consistency in hierarchical supervisory control," *IEEE Trans. Autom. Cont.*, vol. 47, no. 1, pp. 37–52, 2002.

[18] H. Zhong, W. M. Wonham, "On the consistency of hierarchical supervision in discrete-event systems," *IEEE Trans. Autom. Cont.*, vol. 35, no. 10, pp. 1125–1134, 1990.

[19] K. T. Seow, "Organizational control of discrete-event systems: A hierarchical multiworld supervisor design," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 1, pp. 23–33, 2013.

[20] K. Schmidt and C. Breindl, "Maximally permissive hierarchical control of decentralized discrete event systems," *IEEE Trans. Autom. Cont.*, vol. 56, no. 4, pp. 723–737, 2010.

[21] Z. Chao and Y. Xi, "Necessary conditions for control consistency in hierarchical control of discrete-event systems," *IEEE Trans. Autom. Cont.*, vol. 48, no.3, pp. 465–468, 2003.

[22] B. Wang, *Top-down design for RW supervisory control theory*. Master's thesis, Department of Electrical and Computer Engineering, Univ. of Toronto, 1995.

[23] C. Ma and W. M. Wonham, *Nonblocking Supervisory Control of State Tree Structures*, vol. 317, LNCIS, Berlin: Springer-Verlag, 2005.

[24] C. Ma and W. M. Wonham, "Nonblocking supervisory control of state tree structures," *IEEE Trans. Autom. Cont.*, vol. 51, no. 5, pp. 782–793, 2006.

[25] P. E. Caines and Y. J. Wei, "The hierarchical lattices of a finite machine," *Syst. Control Lett.*, vol. 25, no. 4, pp. 257–263, 1995.

[26] P. E. Caines, V. Gupta, and G. Shen, "The hierarchical control of ST-finite-state machines," *Syst. Control Lett.*, vol. 32, no. 4, pp. 185–192, 1997.

[27] P. E. Caines, P. Hubbard, and G. Shen, "State aggregation and hierarchical supervisory control," in *Proc. 36th IEEE Conf. on Dec. and Cont.*, pp. 3590–3591, 1997.

[28] R. J. Leduc, B. A. Brandin, and W. M. Wonham, "Hierarchical interface-based non-blocking verification," in *Proc. IEEE Conf. Canadian Conf. Electr. Comput. Eng.*, vol. 1, pp. 1–6, 2000.

[29] R. J. Leduc, B. A. Brandin, W. M. Wonham, and M. Lawford, "Hierarchical interface-based supervisory control: Serial case," in *Proc. IEEE Canadian Conf. Electr. Comput. Eng.*, vol. 5, pp. 4116–4121, 2001.

[30] R. J. Leduc, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory control-part II: parallel case," *IEEE Trans. Autom. Cont.*, vol. 50, no. 9, pp. 1336–1348, 2005.

[31] D. Harel and A. Pnueli, "On the development of reactive systems. In Logics and Models of Concurrent Systems," *NATO ASI Series*, vol. 13 pp. 477–498, New York, 1985.

[32] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, pp. 231–274, 1987.

[33] Y. Brave and M. Heymann. "Control of discrete event systems modeled as hierarchical state machines," *IEEE Trans. Autom. Cont.*, vol. 38, no. 12, pp. 1803–1819, 1993.

[34] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Trans. Comput.*, vol. 35, no. 8, pp. 677–691, 1986.

[35] W. J. Chao, Y. M. Gan, Z. A. Wang, and W. M. Wonham, "Modular supervisory control and coordination of state tree structures," *Intern. J. Cont.*, vol. 86, no. 1, pp. 9–21, 2013.

[36] K. Cai and W. M. Wonham, *Supervisor Localization: A Top-Down Approach to Distributed Control of Discrete-Event Systems*, *Lecture Notes in Control and Information Sciences*, vol. 459, LNCIS, Berlin: Springer-Verlag, 2015.

[37] T. Jiao, Y. M. Gan, G. C. Xiao, and W. M. Wonham, "Exploiting symmetry of state tree structures for discrete-event systems with parallel components," *Intern. J. Cont.*, vol. 90, no. 8, pp. 1639–1651, 2017.

[38] C. Gu, X. Wang, Z. W. Li, and N. Q. Wu, "Supervisory control of state-tree structures with partial observation," *Inform. Sci.*, Elsevier, vol. 465, no. 8, pp. 523–544, 2018.

[39] C. Gu, X. Wang, and Z. W. Li, "Synthesis of supervisory control with partial observation on normal state-tree structures," *IEEE Trans. Autom. Sci. Eng*., vol. 16, no. 2, pp. 984–997, 2019.

[40] D. G. Wang, X. Wang, and Z. W. Li, "Nonblocking supervisory control of state-tree structures with conditional-preemption matrices," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 3744–3756, 2019.

[41] X. Wang, Z. W. Li, and W. M. Wonham, "Real-time scheduling based on nonblocking supervisory control of state-tree structures", *IEEE Trans. Autom. Cont.*, vol. 66, no. 9, pp. 4230–4237, 2021.

[42] X. Wang, T. Moor, and Z. W. Li, "Top-down nested supervisory control of state-tree structures based on state aggregations," in *Proc.* IFAC-PapersOnLine, vol. 32, no. 2, pp. 11175–11180, 2020.

[43] W. M. Wonham, "Towards an abstract internal model principle," *IEEE Trans. Syst., Man. Cybern.*, vol. 11, pp. 735–740, 1976.

[44] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[45] W. M. Wonham, "The internal model principle of control theory," *Preprint on ResearchGate*, 2018.

[46] B. Brandin and F. Charbonnier, "The supervisory control of the automated manufacturing system of the AIP," in *Proc. Rensselaer's 4th Int. Conf. on Computer Integrated Manufacturing and Automation Technology*, Troy, NY, 1994, pp. 319–324.