Safety Control of Hierarchical Synchronous Discrete Event Systems: A State-Based Approach

Benoit Gaudin, Hervé Marchand

Proceedings of the 13th
Mediterranean Conference on Control and Automation
Limassol, Cyprus, June 27-29, 2005

TuM04-2

Safety Control of Hierarchical Synchronous Discrete Event Systems: A State-Based Approach

Benoit Gaudin and Hervé Marchand
Irisa, Université de Rennes I, INRIA Rennes, France.
`First.Last@irisa.fr`

Abstract— In this paper, we discuss the control of a particular class of Hierarchical Discrete Event Systems and the state avoidance control problem is considered. A methodology is provided that locally computes on each component of the system the set of *bad states* (these are the states that may lead to the forbidden states via an uncontrollable trajectory). This is performed without computing the whole system. At this point, the supervisor is evaluated on the fly w.r.t. the bad states and thus requires an on-line evaluation in order to determine the set of events that has to be disabled by control. It is performed in such a way that the global partial transition function does not need to be built.
Key words : Discrete Event Systems, Supervision and control, Hierarchical Systems, State Avoidance Control Problem.

I. INTRODUCTION

In this paper, we are interested in the *Supervisory Control* ([1], [2]) of a particular class of Hierarchical Discrete Event Systems. We are concerned with systems where the construction of the entire system is assumed not to be feasible (due to the state space explosion resulting from the composition), making the use of classical supervisory control methodologies impractical. Several approaches have been considered to deal with reducing the complexity of supervisor synthesis by taking into account the structure of systems. For concurrent systems (with no hierarchy) many different language-based approaches have been proposed (see e.g. [3], [4], [5], [6], [7]). These methodologies are characterized by the fact that the specification (i.e. the expected behavior) can be reorganized according to the structure of the plant. Under this hypothesis, they provide necessary and sufficient conditions under which it is possible to compute local modular supervisors acting upon each component and to operate the individually controlled plant concurrently in such a way that the behavior of the controlled plant corresponds to the supremal one. Similarly, in order to take into account nested behaviors, some models have been introduced together with techniques allowing to solve various control problems for these hierarchical systems [8], [9], [10], [11], [12], [13]. In [8], the interaction between the levels of the systems are defined by *interfaces*. Further, given a set of supervisors (one for each component of the system), conditions under which the *controlled system* is controllable and non-blocking are given. However, no algorithm is presented to compute this set of supervisors according to some expected specifications. In [9], the system is described by means of a hierarchical language, but the orthogonality feature is not taken into account.

In this paper, we are concerned with specifications that are more related to the notion of states rather than to the notion of trajectories of the system (the mutual exclusion problem for example). For this class of problem, one of the main issue is the *state avoidance control problem*, i.e. the supervisor has to control the plant so that the controlled plant does not reach a set of forbidden states (See e.g. [1]). Note that if one wants to used a language-based approach to encode this problem, then the obtained specification may be of the size of the global system itself. This leads us to develop techniques totally devoted to the state avoidance control problem. For this class of control problem, Brave and Heymann in [10] introduced the Hierarchical State Machines (with no synchronization), a simplified version of the STATECHARTS [14]. In this approach, most of the computations are performed on the fly, by taking into account the structure, which may renders the on-line realization of the supervisor difficult. Further, [11], [12] introduced the *state tree structures* also an adaptation of the STATECHARTS, but more general than the model of [10]. They developed a recursive algorithm allowing to solve the *non-blocking invariance problem*. The efficiency of the method is mostly due to the use of the *Binary Decision Diagram* (BDD) that are used to encode the model and the transformer predicates. Note that the obtained supervisor is monolithic (represented by one BDD) and does not reflect anymore the hierarchical structure of the system. They try to overcome the problem due to the size of the resulting BDD encoding the supervisor by decomposing it according to the set of controllable events, thus reducing the on-line evaluation of the supervisor. The hierarchical model that is considered in this paper is more restrictive than the one of [12] (we only consider a two level hierarchical model). However, compared to [12], our method is independent of the implementation; following the ideas of [7], [15] in a concurrent setting, the idea is to break down the computation into two phases (an off-line and an on-line computation): the set of forbidden states is first reorganized according to the structure of the system and a methodology is provided that locally computes on each component of the system the set of *bad states* (these are the states that may lead to the forbidden states via an uncontrollable trajectory). This is performed without computing the whole system. At this point, the supervisor is evaluated on the fly w.r.t. the bad states and thus requires an on-line evaluation in order to determine the set of events that has to be disabled by

control. It is performed in such a way that the global partial transition function does not need to be built. Moreover, we make the necessary effort to obtain a good complexity during the on-line computation and off-line evaluation. For the latter, this is basically due to the fact that the structure given to the set of bad states is similar to the one of the plant, so that the realization on the fly of the supervisor (itself encoded according to the structure of the system) becomes easier.

II. PRELIMINARIES

We consider in this study a system represented as a Finite State Machine (FSM). An FSM is a 4-tuple (Σ, Q, q_0, δ) where Σ is the set of events that can occur, Q is the set of states , q_0 is the initial state and $\delta : \Sigma \times Q \longrightarrow Q$ is the partial transition function. For $q \in Q$, $\delta(q)$ denotes the active event set of q. Similarly, $\delta^{-1}(q)$ denotes the set of events that lead to q. We define the operator Pre_A^G for all $E \subseteq Q$ by $Pre_A^G(E) = E \cup \{q \in Q | \exists \sigma \in A, \; \delta(\sigma, q) \in E\}$, as well as the operator

$$CoReach_A^G(E) = \bigcup_{n \geq 0} Pre_A^{G(n)}(E) \qquad (1)$$

$CoReach_A^G(E)$ represents the set of states from which it is possible to reach E by only triggering events of A.

The State Avoidance Control Problem. Let G be a plant modeled as an FSM (Σ, Q, q_0, δ). In order to control this FSM, we classically partition the alphabet into controllable events Σ_c and uncontrollable events Σ_{uc}. Given this partition, a supervisor $\mathcal{S}$ is given by a function $S : Q \rightarrow 2^{\Sigma_c}$, delivering the set of actions that are disabled in state q of G by control. We write $\mathcal{S}/G$ for the closed-loop system, consisting of the initial plant G controlled by the supervisor $\mathcal{S}$. In the sequel, we are interested in solving the *State Avoidance Control Problem* (*SACP*) which is defined as follows:

SACP: *given G and E a set of states, the problem is to build a supervisor $\mathcal{S}_E$ such that (1) the traversed states do not belong to E and (2) $\mathcal{S}_E/G$ is the most permissive solution (according to the inclusion of languages).*

In order to solve this problem, given a set of forbidden state E, we first introduce the *set of bad states* $\mathcal{I}(E)$:

$$\mathcal{I}(E) = CoReach_{\Sigma_{uc}}^G(E) \qquad (2)$$

$\mathcal{I}(E)$ corresponds to the set of states from which it is possible to evolve into E by a trace of uncontrollable events. This operator is monotonic and distributes over union.

Proposition 1: Given an FSM G and $E \subseteq Q$, a set of states E. If $q_o \notin \mathcal{I}(E)$, then the supervisor $\mathcal{S}_E$ of G, s.t. $\forall q \in Q$

$$\mathcal{S}_E(q) = \{\sigma \in \Sigma_c | \; \delta(\sigma, q)! \wedge \delta(\sigma, q) \in \mathcal{I}(E)\} \qquad (3)$$

is the most permissive supervisor ensuring the avoidance of E in G.
A similar result (with a predicate approach) can be found in [1].

III. CONTROL OF CONCURRENT SYSTEMS: A STATE-BASED APPROACH

Let us first consider a plant G modeled as a collection of FSM $G_i = \langle \Sigma_i, Q_i, q_{oi}, \delta_i \rangle$. The global system is given by $G = G_1 \parallel \cdots \parallel G_n$, where the operation $\parallel$ is the classical *parallel composition* (i.e. $G_1 \parallel G_2$ represents the concurrent behavior of G_1 and G_2 with synchronization on the shared events). The resulting FSM will be noted $\langle \Sigma, Q, q_o, \delta \rangle$ and the states of G will be denoted by $q = \langle q_1, \ldots, q_n \rangle$. Σ_s represents the set of shared events of G, i.e $\Sigma_s = \bigcup_{i \neq j}(\Sigma_i \cap \Sigma_j)$. Now, given the set of FSMs G_i modeling G, $\mathrm{IN}(.)$ is a function, which for each $\sigma \in \Sigma$ gives the set of indexes $i \in \{1, \ldots, n\}$ such that $\sigma \in \Sigma_i$. For each G_i, we have $\Sigma_i = \Sigma_{i,uc} \cup \Sigma_{i,c}$. The alphabet of the global plant G is given by $\Sigma = \cup_i \Sigma_i$, $\Sigma_c = \cup_i \Sigma_{i,c}$, and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$. Moreover, we assume that the following relation holds between the control status of shared events, i.e.

$$\forall i, j, \; \Sigma_{i,uc} \cap \Sigma_{j,c} = \emptyset$$

which simply means that the components that share an event agree on the control status of this event. Under this hypothesis, we have that $\Sigma_{uc} = \cup_i \Sigma_{i,uc}$.

A. Control Problem formulation

In this section, our aim is to solve the SACP for a set of forbidden states E of a concurrent system. It can be shown that *any set of states E can be represented as a union of Cartesian products of sets*, i.e. $E = \bigcup_{1 \leq j \leq m} E^j$, where $\forall 1 \leq j \leq m$, $E^j = E_1^j \times \cdots \times E_n^j$ and $\forall i \leq n$, $E_i^j \subseteq Q_i$. Given the concurrent structure of the system, this decomposition of sets in terms of product sets happens to be very natural and will be the basis for the expression of states that will have to be forbidden by control. As explained in Section II, the SACP can be reduced to the computation of $\mathcal{I}(E)$ and the most permissive supervisor ensuring the avoidance of E is then simply given by the formula (3). It is theoretically possible to compute $\mathcal{I}(E)$ on G as far as G can be efficiently represented by a single FSM. However, due to the state space explosion, this may be not feasible for concurrent systems. Moreover, the expression of the supervisor requires an on-line evaluation in order to determine the set of events that has to be disabled by control (given an event, one have to test whether the state, reached by triggering this event belong to $\mathcal{I}(E)$ or not). This evaluation is even better when the set $\mathcal{I}(E)$ is itself well structured.

B. A modular solution for the SACP

In this paper, we focus on concurrent systems $G = G_1 \parallel \cdots \parallel G_n$, such that the uncontrollable events are local to each component (i.e. $\Sigma_s \subseteq \Sigma_c$)[1].
Efficient computation of $\mathcal{I}(E)$. The next proposition shows that the $\mathcal{I}(\cdot)$ operator can be expressed using only local computations performed on each component.

---

[1] To simplify the paper as well as the notations, the general case (i.e. when $\Sigma_s \cap \Sigma_{uc} \neq \emptyset$) is not presented here (See [16] for the details).

Proposition 2: Let us consider $G = G_1 \| \ldots \| G_n$, such that $\Sigma_s \subseteq \Sigma_c$. Let $E = \bigcup_{1 \leq j \leq m} E_1^j \times \cdots \times E_n^j$, with $E_i^j \subseteq Q_i$. Then,

$$\mathcal{I}(E) = \bigcup_{1 \leq j \leq m} (\prod_{1 \leq i \leq n} \mathcal{I}_i(E_i^j)) \qquad (4)$$

where $\mathcal{I}_i(E_i)$ is the set of bad states w.r.t. E_i and G_i. $\diamond$

In order to compute $\mathcal{I}(E)$, it is sufficient to compute on each component the sets of states $(\mathcal{I}_i(E_i^j))_{j \leq m, i \leq n}$. Based on (4), the overall complexity is in $\mathcal{O}(m.n.k.N)$, where n is the number of components, $N = max_i(|Q_i|)$ the number of states of each component and $k = max_i(|\Sigma_i|)$. It is worthwhile noting that if $\mathcal{I}(E)$ is expanded then it may be of the size of the global system and then unfeasible to compute. In order to avoid to store in memory this large set of states (i.e. $\mathcal{I}(E)$), we prefer to store in memory local set of states (i.e. the $\mathcal{I}_i(E_i^j)$) and to perform some on-line computations. This is the aim of the Corollary 1.

On-line supervision. Now, given $\mathcal{I}(E)$ as in (4), one can easily extract a supervisor as follows

Corollary 1: Let $G = G_1 \| \ldots \| G_n$, such that $\Sigma_s \subseteq \Sigma_c$, and E a set of states of G. With the notations of Proposition 2, the supervisor S defined for all $q \in Q$ by

$$S_E(q) = \{\sigma \in \Sigma_c | \delta(q, \sigma)! \wedge$$
$$\delta(q, \sigma) \in \bigcup_{1 \leq j \leq m} (\mathcal{I}_1(E_1^j) \times \cdots \times \mathcal{I}_n(E_n^j))\}$$

ensures the avoidance of E and is maximal. $\diamond$

Let us now see how the expression given in Corollary 1 allows an efficient on-line evaluation of $S_E(q)$. To do so, it is sufficient, for each $\sigma \in \Sigma_c$ to determine $\delta(\sigma, q) = \langle q_1', \ldots, q_n' \rangle$ and to test whether $\langle q_1', \ldots, q_n' \rangle \in \mathcal{I}(E)$ or not. According to (4) we just have to test that there exists $j \in \{1, \ldots, m\}$ such that $\forall i \in In(\sigma), \delta_i(\sigma, q_i) \in \mathcal{I}_i(E_i^j)$ et $\forall i \notin In(\sigma), q_i \in \mathcal{I}_i(E_i^j)$. This can be done in $\mathcal{O}(|\Sigma_c|m.n.log(N))$, which is an acceptable complexity as far as the objectives are well structured.

IV. HIERARCHICAL FINITE STATE MACHINES

So far, we gave results dealing with the control of Concurrent systems. We now extend these results to the case of Hierarchical Finite State Machines (HFSM). A Hierarchical Finite State Machines is an FSM which includes new features like the nesting of state machines (inducing the hierarchy) and the parallelism between state machines. From now on, some states (called super-states) of an FSM can be other FSMs. Informally, the meaning of such a hierarchical definition is obtained by substituting each super-state by a set of FSMs running in parallel. Such a model is called Hierarchical Finite State Machine (HFSM). We hereby focus on a two-level Finite State Machine.

A. Definition of an HFSM

1) The model: In order to take into account the hierarchy, we need to introduce the notion of structure. It will represent the upper level of the HFSM.

Definition 1: A structure K is a tuple $\langle \Sigma, Q, \mathcal{B}, q_o, \delta \rangle$, where Q is a set of atomic states, $q_o \in Q$ is the initial state.

$\mathcal{B}$ is the set of super-states of K. δ is the partial transition function defined over $\Sigma \times \{Q \cup \mathcal{B}\} \rightarrow \{Q \cup \mathcal{B}\}$.
In the following we will denote by $K^A = \langle \Sigma, Q \cup \mathcal{B}, q_o, \delta \rangle$ the structure K seen as an FSM (i.e. when the super-states are considered as atomic states). The notion of structure allows us to define the notion of Hierarchical Finite State Machines.

Definition 2: An HFSM $\mathcal{K}$ is given by a tuple $(\langle K, G_1, \ldots, G_n \rangle, Y, I)$, where K is a structure (C.f. Def. 1) et $\forall 1 \leq i \leq n, G_i = \langle \Sigma_i, Q_i, Q_{o_i}, q_{m_i}, \delta_i \rangle$ is a trim FSM. Y, I are two functions that characterize the hierarchy and the composition between the FSMs.

- $Y : \mathcal{B} \longrightarrow 2^{\langle G_1, \ldots, G_n \rangle}$ is a function which maps each super-state $b \in \mathcal{B}$ on a set of FSMs G_i. We use J_b as $\{j \leq n| G_j \in Y(b)\}$.
- I is a function such that $\forall b \in \mathcal{B}, I(b)$ is a function defined by $\Pi_{j \in J_b} Q_{o_j} \rightarrow 2^\Sigma$. Given a macro-state b and $q_o \in \Pi_{j \in J_b} Q_{o_j}$ a tuple of initial states, $I(b)(q_o)$ corresponds to the events that make the system go from its current state into q_o.

K will be called the *top-level* of $\mathcal{K}$.

Remark 1: We here assume that the FSM of the super-states have several initial states. This allows to take into account the last action that has been fired at the high-level before entering the super-state during the execution. This would have been possible to consider FSM having only one initial state per FSM, but the representation of the system would have been less compact. On the other side, we assume that these FSM have only one final state. This only constitutes a simplification of the model. It would have been possible to have an output function (similar to I), but this would render the computation of the supervisor less understandable.

2) Assumptions: In order to be able to perform control on HFSM, we need to make some assumptions on it:

(1) $\forall i$, t.q. $\Sigma \cap \Sigma_i = \emptyset$. This entails that leaving a super-states is deterministic.
(2) Let $b \in \mathcal{B}$ and let $(G_j)_{j \in J_b} = Y(b)$ be the corresponding FSMs attached to b. Then,

$$\delta^{-1}(b) \subseteq \bigcup_{q_o \in \Pi_{j \in J_b}(Q_{o_j})} (I(b)(q_o)) \text{ and}$$
$$\forall q_o, q_o' \in \Pi_{j \in J_b}(Q_{o_j}), q_0 \neq q_o' \Rightarrow I(b)(q_o) \cap I(b)(q_o') = \emptyset$$

i.e. entering the super-state b is deterministic and each event leading to b is taken into account.

3) The behavior of $\mathcal{K}$: Let $\mathcal{K} = (\langle K, G_1, \ldots, G_n \rangle, Y, I)$ be an HFSM. $\mathcal{K}$ is initialized in the initial state of K and as long as no super-state is reached, the behavior of $\mathcal{K}$ corresponds to the one of the FSM K^A. Assume now that the plant is in a state q (at the top-level) such that $\delta(\sigma, q) = b \in \mathcal{B}$ and that σ is triggered. Then all the structures of $Y(b)$ are simultaneously activated and entered in one of their initial states according to $I(b)$, i.e. $\mathcal{K}$ in the configuration $[b, q_o] = [b, \langle q_{o_{j_1}}, \ldots, q_{o_{j_{\|J_b\|}}} \rangle]$, such that $\sigma \in I(b)(q_o)$. Further, the different structures evolve in a concurrent way. In order to evolve out of a super-state b,

there is a synchronization between the different structures of $Y(b) = (G_i)_{i \in J_b}$ on their final state (we recall that $\forall i$, G_i has a unique final state). Hence, an event $\sigma \in \delta(b)$ can be triggered in a super-state b whenever each substructure of $Y(b)$ is in its corresponding final state (i.e., there is no preemption).

4) Expanded two-level HFSM: Given a HFSM $\mathcal{K} = (\langle K, G_1, \ldots, G_n \rangle, Y, I)$, we can make correspond an FSM. It is obtained as follows. To each super-state $b \in \mathcal{B}$, we associate its corresponding FSM G_b obtained by performing the synchronous product between each FSM of $Y(b)$

$$G_b = \langle \Sigma_b, Q_b, Q_{o_b}, x_{f_b}, \delta_b \rangle = \|_{j \in J_b} G_j.$$

To expand the structure K, we replace each super-state b of $\mathcal{B}$ by its corresponding FSM G_b and we connect the initial states of G_b to the states of K according to I (resp. for the final state). The result is an FSM, denoted by $\mathcal{K}^F$.

Definition 3: The FSM associated to an HFSM $\mathcal{K}$ is noted $\mathcal{K}^F = \langle \Sigma^F, Q^F, q_o^F, \delta^F \rangle$, where each component is defined by :

- $\Sigma^F = \Sigma \cup \bigcup_{b \in \mathcal{B}} \{\Sigma_b\}$, $q_o^F = q_o$, and
- $Q^F = Q \cup \{[b, \langle q_1, \ldots, q_{\|J_b\|} \rangle] \quad | \forall b \in \mathcal{B}, \forall \langle q_1, \ldots, q_{\|J_b\|} \rangle \in Q_b\}$
- *The partial transition function δ^F is defined $\forall q, q' \in Q$, $\forall b, b' \in \mathcal{B}$, $\forall \sigma \in \Sigma^F$ by :*
  - $\delta^F(\sigma, q) = q'$ *if $\delta(q, \sigma)!$ and $\delta(\sigma, q) = q'$*
  - $\delta^F(\sigma, q) = [b, q_{o_b}]$ *if $\delta(q, \sigma)!$ and $\delta(\sigma, q) = b$ and $\sigma \in I(b)(q_{o_b})$.*
  - $\delta^F(\sigma, [b, q_{f_b}]) = q$ *if $\delta(q, b)!$ and $\delta(\sigma, b) = q$, where q_{f_b} is the final state of G_b.*
  - $\delta^F(\sigma, [b, q_{f_b}]) = [b', q_{o_{b'}}]$ *si $\delta(q, b)!$ and $\delta(\sigma, b) = b'$, $\sigma \in I(b')(q_{o_{b'}})$ and q_{f_b} is the final state of G_b.*
  - $\delta^F(\sigma, [b, q_b]) = [b, \delta_b(\sigma, q_b)]$ *if $\delta_b(\sigma, q_b)!$.*
  - *undefined otherwise*

Before extended the results of Section III, we need to give some assumptions related to the status of the events.

5) Status of the events.: An HFSM is composed of various sub-systems modeled by a structure K and n FSM $\{G_i\}_{1 \leq i \leq n}$. Σ_c and Σ_{uc} represent the set of controllable and uncontrollable events of K. For each component $(G_i)_{1 \leq i \leq n}$, $\Sigma_i = \Sigma_{i,uc} \cup \Sigma_{i,c}$. As in Section III, we assume that the components agree on the control status of the events

$$\forall i \neq j, \; \Sigma_{i,c} \cap \Sigma_{j,uc} = \emptyset$$

Under this hypothesis, one can define the set of controllable events (Σ_c^F) and uncontrollable events (Σ_{uc}^F) of $\mathcal{K}^F$ (or of the HFSM $\mathcal{K}$) by

$$\Sigma_c^F = \left(\bigcup_{1 \leq i \leq n} \Sigma_{i,c} \right) \cup \Sigma_c, \quad \Sigma_{uc}^F = \left(\bigcup_{1 \leq i \leq n} \Sigma_{i,uc} \right) \cup \Sigma_{uc}$$

B. The State Avoidance Control problem presentation

In this section, we consider the state avoidance control problem, namely how to avoid the hierarchical system to reach some configurations during its evolution. By configuration, we mean a particular state of $\mathcal{K}^F$.

1) Forbidden configurations:

Let $\mathcal{K} = (\langle K, G_1, \ldots, G_n \rangle, Y, I)$ be an HFSM, with $K = (\Sigma, Q, \mathcal{B}, q_o, \delta)$. Given $b \in \mathcal{B}$, we denote by E^b the union of product sets $E^b = \bigcup_{1 \leq j \leq m_b} E^{b,j}$ where $E^{b,j}$ is a product set of the form $E^{b,j} = E_{j_1}^{b,j} \times \cdots \times E_{j_{\|J_b\|}}^{b,j}$ et $E_{j_i}^{b,j} \subseteq Q_{j_i}$ pour $j_i \in J_b$. For simplicity, the set of configurations $[b, \langle q_{j_1}, \ldots, q_{j_{\|J_b\|}} \rangle]$ such that $\langle q_{j_1}, \ldots, q_{j_{\|J_b\|}} \rangle \in E^b$ is denoted $[b, E^b]$. Now, every set of configurations of $\mathcal{K}^F$ can be represented by a set E of the form

$$E = E_0 \bigcup \left(\bigcup_{b \in \mathcal{B}} [b, E^b] \right) \tag{5}$$

where $E_0 \subseteq Q$. This set represents the forbidden configurations at the top-level of $\mathcal{K}$, whereas $[b, E^b]$ corresponds to the forbidden configurations at the lower level (i.e. inside the super-state b).

When E represents the set of forbidden states, we denote by $\mathcal{I}(E)$ the set of bad states of $\mathcal{K}^F$.

$$\mathcal{I}(E) = CoReach_{\Sigma_{uc}^F}^{\mathcal{K}^F}(E) \tag{6}$$

and the supervisor S defined for all $q \in Q^F$ by

$$S(q) = \{\sigma \in \Sigma_c | \; \delta^F(q, \sigma)! \wedge \delta^F(q, \sigma) \in \mathcal{I}(E)\}$$

ensures the avoidance of E in $\mathcal{K}^F$ and is maximal.

As in the concurrent system framework, the goal of this section is to provide an efficient method to compute $\mathcal{I}(E)$ as well as a representation of $\mathcal{I}(E)$ allowing an easy off-line evaluation of the supervisor.

First, the definition of the set of bad states needs to be extended in order to take into account the super-states. The idea is that we do not want to remove a super-state by control as there possibly exists a way to control the system inside this super-state. *A contrario*, let $b \in \mathcal{B}$ be a super-state of K. Given a control objective, it may happen that we need to restrict the entering in a super-state. Hence, for $A \subseteq \delta^{-1}(b)$, we introduce $b_{|A}$ the "controlled super-state" b considering it is only reachable by triggering an event of A and we denote by $\mathcal{B}_{|\mathcal{A}} = \{b_{|A} \mid b \in \mathcal{B} \text{ and } A \subseteq \delta^{-1}(b)\}$, the corresponding set of controlled super-states. This kind of states are introduced in order to partially forbid some super-states. Indeed, one can only want some super-states b to be reachable with respect to a subset of $I(b)(.)$. In fact, this is a way to avoid some initial states of b to be reachable at the lower level of the hierarchy.

Based on these remarks and definitions, we now extend the definition of the set of bad states.

Definition 4: Let $K = (\Sigma, Q, \mathcal{B}, q_o, \delta)$ be the top-level of an HFSM $\mathcal{K}$ and $e \in Q \cup \mathcal{B}_{|\mathcal{A}}$.
- If $e \in Q$, then

$$\mathcal{I}_Q(e) = \{q \in Q \mid \exists s \in \Sigma_{uc}^*, \; \delta(s, q) = e \text{ and } \forall s' \leq s, \; \delta(s', q) \notin \mathcal{B}\}.$$

$$\mathcal{I}_B(e) = \{b \in \mathcal{B} | \; \exists \sigma \in \Sigma_{uc}, \; \delta(\sigma, b) \in \mathcal{I}_Q(e)\}$$

- If $e = b_{|A} \in \mathcal{B}_{|A}$, then

$$\mathcal{I}_Q(b_{|A}) = \{q \in Q \mid \exists s \in \Sigma_{uc}^*, \ \exists \sigma \in \Sigma_{uc} \cap A, \\ \delta(s\sigma, q) = b \text{ and } \forall s' \leq s, \ \delta(s', q) \notin \mathcal{B}\}$$

$$\mathcal{I}_B(b_{|A}) = \{b' \in \mathcal{B} \mid \exists \sigma \in \Sigma_{uc}, \ (\delta(\sigma, b') \in \mathcal{I}_Q(b_{|A})) \\ \text{or } (\sigma \in \Sigma_{uc} \cap A \text{ and } \delta(\sigma, b') = b)\}$$

Finally, given $E \subseteq Q \cup \mathcal{B}_{|A}$, $\mathcal{I}_Q(E) = \cup_{e \in E} \mathcal{I}_Q(e)$ and $\mathcal{I}_B(E) = \cup_{e \in E} \mathcal{I}_B(e)$

Intuitively speaking, if $e \in Q$, $\mathcal{I}_Q(e)$ (resp. $\mathcal{I}_B(e)$) represents the set of atomic states (resp. super-states) of K from which e can be reached via an uncontrollable trajectory that only traverses atomic states. If $e = b_{|A} \in \mathcal{B}_{|A}$, the meaning of $\mathcal{I}_Q(e)$ and $\mathcal{I}_B(e)$ is similar except that we ask the last event of the uncontrollable trajectories to belong to A.
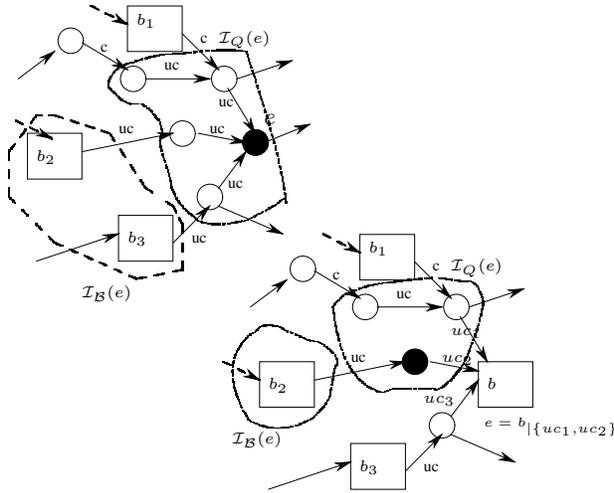
Fig. 1.   Computation of $\mathcal{I}(e), \mathcal{I}_B(e)$ through an example

The next operator Ψ will be useful to compute the set of bad configurations by going-up/down in the hierarchy of the plant. Indeed, if an initial state of a super-state has to be forbidden by control, then at the top-level, a supervisor has to avoid the system to enter the super-state via this initial state. Conversely, the final state of a super-state that may lead to a forbidden configuration via an uncontrollable trajectory has also to be forbidden by control. This is captured by the definition 5.

Definition 5: Let $e \in Q^F \cup \mathcal{B}_{|A}$. For $b \in \mathcal{B}$, we denote by Q_{ob} (resp. q_{f_b}) the set of initial states (resp. the final state) of the FSM associated to b (i.e. $K_b = \|_{j \in J_b} G_i$).

1) If $e \in Q \cup \mathcal{B}_{|A}$, then

$$\Psi(e) = \mathcal{I}_Q(e) \cup \{[b, q_{f_b}] \mid b \in \mathcal{I}_B(e)\}$$

2) If $e = [b, (q_{j_1}, \ldots, q_{j_{\|J_b\|}})]$, then given the set $\mathcal{I}^b = \mathcal{I}^b(e)$, where $\mathcal{I}^b(e)$, which corresponds to the set of states of e in $\|_{j \in J_b} G_i$ computed as in Section III,

$$\Psi(e) = [b, \mathcal{I}^b] \cup b_{|A'} \text{ with } \mathcal{A}' = \bigcup_{q_{o_b} \in \mathcal{I}^b \cap Q_{ob}} \{I(b)(q_{o_b})\} \quad (7)$$

Moreover, for $E \subseteq Q^F \cup \mathcal{B}_{|A}$, $\Psi(E) = \bigcup_{e \in E} \Psi(e)$.

Given $e \in Q^F \cup \mathcal{B}_{|A}$, if $e \in Q \cup \mathcal{B}_{|A}$, then $\Psi(e)$ corresponds to the set of bad configurations, to which we add the final states of the super-states that can lead into e via an uncontrollable trajectory. This way we are going down in the hierarchy. Now, if $e = [b, \langle q_{j_1}, \ldots, q_{j_{\|J_b\|}} \rangle]$, then $\Psi(e)$, corresponds to the set of bad configurations inside the super-state b, as well as the restricted super-state itself if some initial states belong to the set of bad states. Remark that in point *2* the computations are made locally on each G_i that are involved in the super-states b using the techniques described in Section III.

The Supervisor. Let $\mathcal{K}$ be an HFSM and assume given a set of forbidden configurations of the form $E = E_0 \bigcup \left(\bigcup_{b \in \mathcal{B}} [b, E^b] \right)$, as defined by (5), then the set of bad configurations is computed by the following fix-point iteration:

$$\mathcal{E}_o = E \text{ et } \mathcal{E}_{i+1} = \Psi(\mathcal{E}_i) \quad (8)$$

Let us call $\mathcal{I}_H(E)$ the result of the previous fix-point computation (it always terminates since the number of states is finite and $\mathcal{E}_i \subseteq \mathcal{E}_{i+1}$). Moreover, one can see that this set can be reorganized as follows:

$$\mathcal{I}_H(E) = Q' \cup \mathcal{B}'_{|A} \cup \bigcup_{b \in \mathcal{B}} [b, E'^b], \quad (9)$$

where $E'^b = \bigcup_i E'^{b,i}_{j_1} \times \cdots \times E'^{b,i}_{j_{\|J_b\|}}$ (i.e. a union of product sets as described in Section III-A), $Q' \subseteq Q$ and $\mathcal{B}'_{|A} \subseteq \mathcal{B}_{|A}$.

For the super-states, what the above states, is that it is forbidden to enter these super-states through the events that belong to a set of events A. Moreover, each super-state b can be seen as a concurrent system for which the set of configurations E'^b has to be forbidden. Note that as E'^b is given by a union of product sets, in order to control the behavior of $\mathcal{K}$, we will use the modular methodology explained in Section III. Based on the previous remarks, we then have the following property that makes the link between $\mathcal{I}_H(E)$ and $\mathcal{I}(E)$

Proposition 3: Let E be a set of states of $\mathcal{K}$, then

$$\mathcal{I}_H(E) \setminus \mathcal{B}_{|A} = \mathcal{I}(E)$$

where $\mathcal{I}(E)$ is computed with respect to $\mathcal{K}^F$ as in (2).

Proof : See Appendix A   ◇

In other words, the set $\mathcal{I}_H(E)^2$ corresponds to the set of bad configurations $\mathcal{I}(E)$ of the plant $\mathcal{K}^F$. However, compared to the classical methods, all the computations have been performed locally and not on the global plant.

A supervisor ensuring the avoidance of $\mathcal{I}(E)$ can thus be described by means of local supervisors ensuring the avoidance of $\mathcal{I}_H(E)$. Based on the previous decomposition (9)

---

[2] to which we remove $\mathcal{B}'_{|A}$ as this set actually corresponds to the set of initial states that are forbidden in a super-states. Hence, they are taken into account by some $[b, E^b]$.

of $\mathcal{I}_H(E)$, a supervisor can be extracted. It is performed as follows:

1) $\forall b \in \mathcal{B}$, we compute the supervisor $\mathcal{S}_b$ that avoids the set of product sets E'^b in $\|_{j \in J_b} G_j$, using the methods described in Section III-B. All the computations have been performed when computing Ψ.

2) For K, we compute the supervisor $\mathcal{S}_K$ defined by

$$\mathcal{S}_K(e) \;=\; \{\sigma \in \Sigma_c | \delta(\sigma, e) \in Q' \vee \delta(\sigma, e) = b \\ \text{with } b_{|A} \in \mathcal{B}'_{|\mathcal{A}} \wedge \sigma \in A\}$$

Corollary 2: With the preceding notations, let $\mathcal{S} = \langle \mathcal{S}_K, (\mathcal{S}_b)_{b \in \mathcal{B}} \rangle$ be such that

$$S_E(e) = \begin{cases} S_K(e) \text{ if } e \in Q \\ S_K(b) \cup S_b(q_{f_b}) \text{ if } e = [b, q_{f_b}] \\ S_b(q_{j_1}, \dots, q_{j_{\|J_b\|}}) \text{ if } \\ \quad e = [b, \langle q_{j_1}, \dots, q_{j_{\|J_b\|}} \rangle] \text{ but not final} \\ \emptyset \text{ Otherwise} \end{cases} \quad (10)$$

Then, $\mathcal{S}$ ensures the avoidance of E in $\mathcal{K}$ and is maximal. $\diamond$
According to (10), the control policy of the supervisor is the following: when the current configuration of $\mathcal{K}$ is in Q, then the supervisor $\mathcal{S}_K$ is activated. $\mathcal{S}_K$ is also activated when the plant under control enters the final state of a super-state. Note that $\mathcal{S}_K$ also takes into account the fact that some initial configurations of a super-state are possibly forbidden. When $\mathcal{K}$ enters a super-state b under the control of $\mathcal{S}_K$ (which entails that the super-state is entered through the allowed events) then the supervisor $\mathcal{S}_b$ becomes active. It is deactivated whenever the plant under control leaves the super-state b.

To conclude this section, let us remark that as in Section III, we made the necessary efforts not to expand the HFSM in order to compute the supervisor. In particular, the set of bad configurations has been computed locally on each submachine $(G_i)_{1 \leq i \leq n}$ and on the upper level of the HFSM.

C. A simple extension of the model

Let $\mathcal{K}_i = (\langle K^i, G^i_1, \cdots, G^i_{n_i} \rangle, Y_i, I_i)$, $i = 1, 2$ be two HFSM that are in parallel. Let us note $\mathcal{K} = \mathcal{K}_1 \| \mathcal{K}_2$. From a behavioral point of view, we have that $\mathcal{K}^F = \mathcal{K}^F_1 \| \mathcal{K}^F_2$. We assume that the only shared events of $\mathcal{K}_1$ and $\mathcal{K}_2$ are the one of the structures K_1 and K_2. For simplicity, we assume that these shared events are controllable.

Let $E = \cup_l E^l_1 \times E^l_2$ be a set of configurations of $\mathcal{K}$ with E^l_i is a set of configurations of $\mathcal{K}_i$ as described by (5). From Section III-B and Proposition 2, we know that the set of bad states of $\mathcal{K}$ (or $\mathcal{K}^F$) is given by $\cup_l \mathcal{I}^F_1(E^l_1) \times \mathcal{I}^F_1(E^l_2)$, where $\mathcal{I}^F_i()$ is computed w.r.t. $\mathcal{K}^F_i$ and E^l_i. Moreover from the previous section, we know how to efficiently compute these sets without building $\mathcal{K}^F_i$ (Proposition 3). Therefore, our methodology is also suitable for Concurrent HFSM.

V. CONCLUSION

In this paper we have considered the control of a particular class of hierarchical systems. Based on this model,

we proposed a methodology allowing the computation of a supervisor solving the State avoidance control problem. The control objective is given as a collection of forbidden configurations that is decomposed according to the structure of the system. A methodology is provided that locally computes on each component the set of *bad states*. As all the computations are done on the components according to the local specifications, there is no need to build the whole system. At this point, the supervisor is evaluated on the fly w.r.t. the bad states and thus requires an on-line evaluation in order to determine the set of events that has to be disabled by control. It is performed in such a way that the global partial transition function does not need to be built.

REFERENCES

[1] W. M. Wonham, "Notes on control of discrete-event systems," Department of ECE, University of Toronto, Tech. Rep. ECE 1636F/1637S, July 2003.

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

[3] M. deQueiroz and J. Cury, "Synthesis and implementation of local modular supervisory control for a manufacturing cell," in *Proceedings of the 6th International Workshop on Discrete Event Systems*, October 2002, pp. 377–382.

[4] K. Akesson, H. Flordal, and M. Fabian, "Exploiting modularity for synthesis and verification of supervisors," in *Proc. of the IFAC*, July 2002.

[5] Y. Willner and M. Heymann, "Supervisory control of concurrent discrete-event systems," *International Journal of Control*, vol. 54, no. 5, pp. 1143–1169, 1991.

[6] K. Rohloff and S. Lafortune, "The control and verification of similar agents operating in a broadcast network environment," in *42nd IEEE Conference on Decision and Control*, Hawaii, USA, December 2003.

[7] R. Minhas, "Complexity reduction in discrete event systems," Ph.D. dissertation, Univeristy of Toronto, September 2002.

[8] R. Leduc, "Hierarchical interface based supervisory control," Ph.D. dissertation, Dept. of Elec. & Comp. Engrg., Univ. of Toronto, 2002.

[9] P. Gohari-Moghadam, "A linguistic framework for controller hierarchical des," M.A.S.C. Thesis, Dept. of Electl. & Comptr. Engrg., University of Toronto, April 1998.

[10] Y. Brave and M. Heymann, "Control of discrete event systems modeled as hierarchical state machines," *IEEE Transacations on Automatic Control*, vol. 38, no. 12, pp. 1803–1819, December 1993.

[11] B. Wang, "Top-down design for RW supervisory control theory," Masc Thesis, Univ. of Toronto, June 1995.

[12] C. Ma, "Non blocking supervisory control of state tree structures," Ph.D. dissertation, Univeristy of Toronto, February 2004.

[13] H. Marchand and B. Gaudin, "Supervisory control problems of hierarchical finite state machines," in *41th IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002.

[14] D. Harel, "Statecharts : A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, June 1987.

[15] A. Vahidi, B. Lennarston, and M. Fabian, "Efficient supervisory synthesis of large systems," in *Workshop on Discrete Event Systems, WODES'04*, September 2004.

[16] B. Gaudin and H. Marchand, "Efficient computation of supervisors for loosely synchronous discrete event systems: A state-based approach," in *6th IFAC World Congress*, Prague, Czech Republic, July 2005.

APPENDIX A: PROOF OF PROPOSITION 3

Let us first show that $\mathcal{I}_H(E) \setminus \mathcal{B}_{|\mathcal{A}} \subseteq \mathcal{I}(E)$. The proof proceeds by induction. Let us how that $\forall i, \mathcal{E}_i \setminus \mathcal{B}_{|\mathcal{A}} \in I(E)$. At step 0, it is obvious that $E \subseteq \mathcal{I}(E)$. At step i, let $\mathcal{E}_i$ be the result of (8) after i iterations and let us assume that $\mathcal{E}_k \setminus \mathcal{B}_{|\mathcal{A}} \subseteq \mathcal{I}(E)$, $\forall 0 \leq k \leq i$. Let $\mathcal{E}_{i+1} = \Psi(\mathcal{E}_i)$. We

thus have to show that $\mathcal{E}_{i+1} \setminus \mathcal{B}_{|\mathcal{A}} \in I(E)$. To do so, let us consider $e' \in \mathcal{E}_{i+1} \setminus \mathcal{B}_{|\mathcal{A}}$ and $e \in \mathcal{E}_i$ such that $e' \in \Psi(e)$.

• If $e \in Q$, then by Definition 5, $\Psi(e) = \mathcal{I}_Q(e) \cup \{[b, q_{f_b}] \mid b \in \mathcal{I}_\mathcal{B}(e)\}$. Now as $e' \in \Psi(e)$, we have two cases :

Assume that $e' \in \mathcal{I}_Q(e)$. In this case, according to Definition 4, it exists $s \in \Sigma_{uc}^*$ such that $\delta(s, e') = e$ and for all $s' \le s$, $\delta(s', e') \notin \mathcal{B}$. Consequently, by Definition 3, it exists $s \in \Sigma_{uc}^*$ such that $\delta^F(s, e')!$ and $\delta^F(s, e') = e$. We thus deduce from the definition of $\mathcal{I}(E)$ (C.f. (6)) that $e' \in \mathcal{I}(e)$ and then $e' \in \mathcal{I}(E)$ (as $e \in \mathcal{I}(E)$ and $\mathcal{I}(\mathcal{I}(E)) = \mathcal{I}(E)$).

Assume now that $e' = [b, q_{fb}]$ with $b \in \mathcal{I}_\mathcal{B}(e)$. In this case, according to Definition 4, $\delta(\sigma, b)!$ and $\delta(\sigma, b) \in \mathcal{I}_Q(e)$. We thus deduce from Definition 3 that $\delta^F(\sigma, e')!$ and belongs to $\mathcal{I}_Q(e)$. We note $e'' = \delta^F(\sigma, e')$. As $e'' \in \mathcal{I}_Q(e)$, according to Definition 4, it exists $s \in \Sigma_{uc}^*$ such that $\delta(s, e'') = e$ and for all $s' \le s$, $\delta(s', e'') = e$. Consequently, according to Definition 3, it exists $s \in \Sigma_{uc}^*$ such that $\delta^F(s, e'')!$ and $\delta^F(s, e'') = e$. We thus deduce from the definition of $\mathcal{I}(E)$ that $e' \in \mathcal{I}(e)$ and that $e'' \in \mathcal{I}(E)$ (as $e \in \mathcal{I}(E)$ and $\mathcal{I}(\mathcal{I}(E)) = \mathcal{I}(E)$). Moreover $e'' = \delta^F(\sigma, e')$ and $\sigma \in \Sigma_{uc}$, which entails that $e' \in \mathcal{I}(E)$.

• Assume now that $e = [b, e_b]$ with $e_b \in Q^b$. By denoting $\mathcal{I}^b = \mathcal{I}^b(e)$, we have $\Psi(e) = [b, \mathcal{I}^b] \cup b_{|\mathcal{A}'}$ with $\mathcal{A}' = \bigcup_{q_{o_b} \in \mathcal{I}^b \cap Q_{ob}} \{I(b)(q_{o_b})\}$. As by hypothesis $e' \notin \mathcal{B}_{|\mathcal{A}}$, we can deduce that $e' \in [b, \mathcal{I}^b]$ and we note $e' = [b, e'_b]$ with $e'_b \in \mathcal{I}^b$. Now by definition of $\mathcal{I}^b$, it exists $s \in \Sigma_{uc}^*$ such that $\delta^b(s, e'_b) = e_b$. Moreover, we can deduce from the definition of δ^F that $\delta^F(s, e') = e$. As $s \in \Sigma_{uc}$, we can deduce from the definition of $\mathcal{I}(E)$ that $e' \in \mathcal{I}(e)$ and that $e' \in \mathcal{I}(E)$ (as $e \in \mathcal{I}(E)$ and $\mathcal{I}(\mathcal{I}(E)) = \mathcal{I}(E)$).

• Finally, if $e = b_{|A} \in \mathcal{B}_{|\mathcal{A}}$, then according to Definition 5, we have $\Psi(e) = \mathcal{I}_Q(b_{|\mathcal{B}_{|\mathcal{A}}}) \cup \{[b', q_{fb'}] \mid b' \in \mathcal{I}_\mathcal{B}(b_{\mathcal{A}})\}$. We then have two different cases:

- If $e' \in \mathcal{I}_Q(b_{|\mathcal{A}})$, then according to Definition 4, it exists $s \in \Sigma_{uc}^*$ such that $\delta(s, e') = b$ and for all $s' \le s$, $\delta(s', e') \notin \mathcal{B}$. Thus according to Definition 3, it exists $s \in \Sigma_{uc}^*$ and $\sigma \in \Sigma_{uc}$ such that $\delta^F(s\sigma, e')!$ with $\sigma \in A$ and

$$\delta^F(s\sigma, e') = [b, q_{0b}] \qquad (\alpha)$$

Now, as $b_{|\mathcal{A}} \in \mathcal{E}_i$ by hypothesis and $E \cap \mathcal{B}_{|\mathcal{A}} = \emptyset$, it exists $e'' \in \mathcal{E}_{i-1}$ such that $b_{|\mathcal{A}} \in \Psi(e'')$. By definition of Ψ (Def. 5), e'' is of the form $e'' = [b, e''_b]$ with $e''_b \in Q^b$ and $[b, q_{0b}] \in [b, \mathcal{I}^b(e''_b)]$. Thus, we deduce from the definitions of δ^F (Def. 3) and $\mathcal{I}^b(e''_b)$ that it exists $s' \in \Sigma_{uc}^*$ such that $\delta^F(s', [b, q_{0b}]) = (b, e''_b)$ $(= e'')$. We thus deduce from (α) that $\delta^F(s\sigma s', e') = e''$. Moreover $e'' \in \mathcal{E}_{i-1}$ and thus $e'' \in \mathcal{I}(E)$ by hypothesis. We thus obtain from the definition of $\mathcal{I}(E)$ that $e' \in \mathcal{I}(E)$.

- If $e' = [b', q_{fb'}]$ with $b' \in \mathcal{I}_{|\mathcal{B}}(b_{|A})$, then according to the definitions 4 and 3 that it exists $s \in \Sigma_{uc}^*$, $\sigma \in A$ and $\sigma \in I(b)(q_{0b})$ such that

$$\delta^F(s\sigma, e') = [b, q_{0b}] \qquad (\beta)$$

Now, $b_{|A} \in \mathcal{E}_i$ and $E \cap \mathcal{B}_{|\mathcal{A}} = \emptyset$, thus it exists $e'' \in$

$\mathcal{E}_{i-1}$ such that $b_{|\mathcal{A}} \in \Psi(e'')$. Now, by definition of Ψ (Def. 5), e'' is of the form $e'' = [b, e''_b]$ with $e''_b \in Q^b$ and $[b, q_{0b}] \in [b, \mathcal{I}^b(e''_b)]$. Thus, according to the definitions of δ^F (Def. 3) and $\mathcal{I}^b(e''_b)$, it exists $s' \in \Sigma_{uc}^*$ such that $\delta^F(s', [b, q_{0b}]) = (b, e''_b)$ $(= e'')$. We thus deduce from (β) that $\delta^F(s\sigma s', e') = e''$. Now, as $e'' \in \mathcal{I}(E)$, we have that $e' \in \mathcal{I}(E)$.

Overall, we have that $e' \in \mathcal{I}(E)$ and then $\mathcal{I}_H(E) \setminus \mathcal{B}_{|\mathcal{A}} \subseteq \mathcal{I}(E)$.

Let us now show that $\mathcal{I}(E) \subseteq \mathcal{I}_H(E) \setminus \mathcal{B}_{|\mathcal{A}}$. As $\mathcal{I}(E) \subseteq Q^F, \mathcal{I}(E) \cap \mathcal{B}_{|\mathcal{A}} = \emptyset$ and it is sufficient to show that $\mathcal{I}(E) \subseteq \mathcal{I}_H(E)$. To do so, let us consider $q \in \mathcal{I}(E)$. It exists a sequence of configurations $q = q_1, q_2, \cdots, q_n$ in Q^F and a trace in Σ_{uc}^F s.t. $q = q_1$, $\delta^F(\sigma_1, q_1) = q_2, \ldots, \delta^F(\sigma_i, q_i) = q_{i+1}, \ldots, \delta^F(\sigma_{n-1}, q_{n-1}) = q_n \in E$ with $\sigma_i \in \Sigma_{uc}^F$ Let us now show by induction that $\forall i$, $q_i \in \mathcal{I}_H(E)$. As $q_n \in E$ and $E \subseteq \mathcal{I}_H(E)$, we thus have that $q_n \in \mathcal{I}_H(E)$. Assume now that $q_i, \cdots, q_n$ belong to $\mathcal{I}_H(E)$.

• If $q_i \in Q$, then according to the definition of δ^F,
  - either $q_{i-1} \in Q$ and in this case, $q_{i-1} \in \mathcal{I}_Q(q_i)$. Consequently, $q_{i-1} \in \Psi(q_i)$ and as by hypothesis $q_i \in \mathcal{I}_H(E)$ and $\Psi(\mathcal{I}_H(E)) = \mathcal{I}_H(E)$, we can deduce that $q_{i-1} \in \mathcal{I}_H(E)$.
  - or $q_{i-1} = [b, q_{fb}]$ and in this case, $\delta(\sigma_{i-1}, b) = q_i$ and thus $b \in \mathcal{I}_\mathcal{B}(q_i)$ according to Definition 4. From Definition 5, this entails that $[b, q_{fb}] \in \Psi(q_i)$ and thus $q_{i-1} \in \Psi(q_i)$. Moreover, as by hypothesis $q_i \in \mathcal{I}_H(E)$ and $\Psi(\mathcal{I}_H(E)) = \mathcal{I}_H(E)$, we can deduce that $q_{i-1} \in \mathcal{I}_H(E)$.

• If $q_i = [b, q_b]$, then
  - either $q_{i-1} \in Q$ and $q_b = q_{0b}$ and $\sigma_{i-1} \in I(b)(q_{0b})$, and in this case, according to the Definition 5, if we note $\mathcal{I}^b = \mathcal{I}^b(q_{0b})$ and $A = \bigcup_{q_{0b} \in \mathcal{I}^b \cap Q_{0b}} I(b)(q_{0b})$, we have that $b_{|A} \in \Psi(q_i)$. Moreover, according to Definition 4, we have $q_{i-1} \in \mathcal{I}_Q(b_{|A})$. Thus, based on Definition 5, $q_{i-1} \in \Psi(b_{|A})$ and thus $q_{i-1} \in \Psi(\Psi(q_i))$. As $q_i \in \mathcal{I}_H(E)$ by hypothesis, and $\Psi(\mathcal{I}_H(E)) = \mathcal{I}_H(E)$ by definition, we obtain $q_{i-1} \in \mathcal{I}_H(E)$.
  - or $q_{i-1} = [b', q_{fb'}]$, $\delta(\sigma_{i-1}, b') = b$, $q_b = q_{0b}$ and $\sigma_{i-1} \in I(b)(q_{0b})$. If we note $\mathcal{I}^b = \mathcal{I}^b(q_{0b})$ and $A = \bigcup_{q_{0b} \in \mathcal{I}^b \cap Q_{0b}} I(b)(q_{0b})$, we thus have $b_{|A} \in \Psi(q_i)$. Moreover, according to Definition 4, $b' \in \mathcal{I}_\mathcal{B}(b_{|A})$. Thus, based on Definition 5, $q_{i-1} \in \Psi(b_{|A})$. Hence, $q_{i-1} \in \Psi(\Psi(q_i))$. Now, as $q_i \in \mathcal{I}_H(E)$, and $\Psi(\mathcal{I}_H(E)) = \mathcal{I}_H(E)$ by definition, $q_{i-1} \in \mathcal{I}_H(E)$.
  - If $q_{i-1} = [b, q'_b]$, $\delta(\sigma_{i-1}, q'_b) = q_b$, then, in this case $q'_b \in \mathcal{I}^b(q_i)$ and thus $[b, q'_b] \in \Psi(q_i)$ according to Definition 5. As $q_i \in \mathcal{I}_H(E)$ by hypothesis, and $\Psi(\mathcal{I}_H(E)) = \mathcal{I}_H(E)$ by definition of $\mathcal{I}_H$, we obtain $q_{i-1} \in \mathcal{I}_H(E)$.

Overall, we have that $i \in \{1, \ldots, n\}$, $q_i \in \mathcal{I}_H(E)$ and in particular q_1 $(= q)$ $\in \mathcal{I}_H(E)$. Hence, the result.