

---

## ON THE ENABLING-COMPATIBLE PRODUCT

*A creative artist works on his next composition because he was not satisfied with his previous one.*

—Dimitri Shostakovich - New York Times, 1959

### Summary

This appendix provides some implementation details of one of the key parts of the verification approach for timed systems presented in this thesis. That is, the enabling-compatible product, which allows the refinement of the untimed state space of a system by incorporating a set of relative timing constraints in the form of a lazy causal event structure.

In order to be self-contained, the appendix starts with a brief summary of the important notions around the enabling-compatible product. In particular, the rules that precisely define the product are included.

Then, details on the representation of a LzTS and the state space of a LzCES with boolean algebras are provided. Part of this material was already presented in Section 5.1.1 but is included here for completeness.

Finally, the different rules that define the enabling-compatible product are implemented by means of symbolic manipulations of the transitions relations of the events involved in the product.

## B.1 Enabling-compatible product

This section describes how to refine the set of traces produced by a LzTS by considering the timing constraints coming from event delay bounds. The timing constraints are derived by a timing analysis on a CES corresponding to an eligible trace of a LzTS in the untimed domain. The refinement is performed through the parallel composition of a LzTS and a LzCES. Defining such composition requires both descriptions to be represented in a uniform way. To satisfy this requirement we first introduce a state-based representation for CESs.

### B.1.1 State-based representation of a CES

An underlying transition system can be obtained from a CES. This process relies on the notion of *configuration*, which plays the role of global state of the CES.

#### DEFINITION B.1 (CONFIGURATION)

Let  $CS = \langle \Sigma, \prec, \triangleright \rangle$  be a CES.  $\mathcal{C} \subseteq \Sigma$  is a configuration iff:

- $\mathcal{C}$  is left-closed, i.e.  $\forall e_i \in \mathcal{C}$  all predecessors of  $e_i$  by  $\prec$  are in  $\mathcal{C}$ , and
- disabled events do not belong to  $\mathcal{C}$ , i.e.  $e_i \in \mathcal{C} \Rightarrow \nexists e_j \in \Sigma : e_j \triangleright_\mu e_i$ .

Notice that both  $\emptyset$  and the set of not disabled events  $\Sigma \setminus \mathcal{D}$  are trivial configurations.

Event  $e \in \Sigma$  is enabled in configuration  $\mathcal{C}$  iff  $\rightarrow\{e\} \subseteq \mathcal{C}$  and  $\forall e_j \in \Sigma \mid e_j \triangleright e_i : e_j \notin \mathcal{C}$ . We denote by  $\mathcal{E}(\mathcal{C})$  the set of all enabled events in configuration  $\mathcal{C}$ .

■ B.1

Configuration  $\mathcal{C}$  precisely identifies a state of a CES, as the set of events occurred so far, such that if  $e \in \mathcal{C}$  all its causal predecessors must be also in  $\mathcal{C}$ .

Every prefix  $\omega_i$  of a word (a topological order of the events)  $\omega$  in a CES is left-closed and disabled events do not fire along it. Thus every prefix  $\omega_i$  defines a configuration which is reached by firing the events from  $\omega_i$ . Consideration of all possible words of a CES and their prefixes gives the *set of reachable configurations*,  $\mathcal{C}$ , where the initial configuration due to the empty prefix  $\omega_0$  is denoted by  $\top$ . The set of reachable configurations together with the partial order induced by the strict set inclusion  $\subset$ , defines the *graph of reachable configurations*.

#### DEFINITION B.2 (GRAPH OF REACHABLE CONFIGURATIONS)

Let  $CS = \langle \Sigma, \prec, \triangleright \rangle$  be a CES, and  $\mathcal{C}$  be the set of reachable configurations of  $CS$ . The graph of reachable configurations (GRC) of  $CS$  is a Hasse diagram over  $\mathcal{C}$  and the partial order  $\subset$  interpreted in set-theoretical sense.

■ B.2

For the general case of a LzCES,  $LCS = \langle \Sigma, \prec', \triangleright \rangle$ , the graph of reachable configurations can be modeled by a LzTS  $G = \langle \mathcal{C}, \Sigma, T, \top, \text{EnR} \rangle$  where: there is one state per config-

uration;  $\mathcal{C}_1 \xrightarrow{e} \mathcal{C}_2 \in T$  iff  $\mathcal{C}_2$  is reached by firing  $e \in \Sigma$  from  $\mathcal{C}_1$ ; the initial state corresponds to the initial configuration  $\top$ ; and  $\text{EnR}(e) = \{\mathcal{C} \in C \mid e \in \mathcal{E}(\mathcal{C})\}$ .

### B.1.2 Refining the reachability space by timing constraints

In order to refine the state space of the system with a set of relative timing constraints, we have two objects at hand: a lazy TS  $A$ , and another lazy TS  $G$  obtained from an event structure  $CS_\theta$ .  $CS_\theta$  is derived from a particular trace  $\theta$  of  $A$  (actually by an appropriate suffix), thus giving only a partial specification of the behavior of  $A$ .  $CS_\theta$  is refined through timing analysis yielding the lazy TS  $G$ .

Refining the behavior of  $A$  by the timing constraints incorporated in  $G$  can be done by calculating the *enabling-compatible product* of  $G$  and  $A$ , which is a particular case of transition system product under the restrictions of making synchronization by the *same transitions* and the *same enabling conditions*.

For sake of simplicity, and before introducing the rules of the enabling-compatible product below, we will add the special configuration  $\perp$  to  $G$ .  $\perp$  denotes the fact that the product is not synchronizing, *i.e.* there is no enabling-compatibility with the state space of the CES and therefore, timing analysis does not apply for the involved traces.

Given the system  $A = \langle S, \Sigma_A, T_A, s_0, \text{EnR}_A \rangle$  and the state space of the LzCES containing the relative timing constraints  $G = \langle C \cup \perp, \Sigma_G, T_G, \top, \text{EnR}_G \rangle$ , with  $\Sigma_G \subseteq \Sigma_A$ , the enabling-compatible product of  $A$  and  $G$  is a new LzTS  $\langle S', \Sigma_A, T', s'_0, \text{EnR}' \rangle$  where:

- $S' \subseteq S \times (C \cup \perp)$ ,
- $s'_0 = (s_0, \top)$  if  $\mathcal{E}(\top) \subseteq \mathcal{E}(s_0)$ , and  $s'_0 = (s_0, \perp)$  otherwise, and
- $\forall e \in \Sigma_A, \text{EnR}'(e) = \{(s, \mathcal{C}) \in S' \mid s \in \text{EnR}_A(e)\}$ .

**Remark:** The alphabet  $\Sigma_G$  is not properly a subset of  $\Sigma_A$ . In fact  $\Sigma_G$  might contain several *instances* of any event in  $\Sigma_A$ . That is, provided an event  $e \in \Sigma_A$ , a set  $\{e/1, e/2, \dots\}$  of instances of  $e$  might be present in  $\Sigma_G$ . This is the case for example when the corresponding LzCES is generated from a trace of  $A$  where several occurrences of event  $e$  appear along the trace. Thus we define  $\Sigma_G \downarrow \Sigma_A$  as the *projection* of the event occurrences in  $\Sigma_G$  over the actual events in  $\Sigma_A$ . For example if  $\Sigma_A = \{a, b, c\}$  and  $\Sigma_G = \{a/1, a/2, c/1\}$ ,  $\Sigma_G \downarrow \Sigma_A = \{a, c\}$ .

The transition relation  $T'$  is defined by the rules below. The rules are implied by the conditions on the enabling-compatibility of traces. The fact that  $(s, \mathcal{C}) \in S'$  denotes that  $s$  and  $\mathcal{C}$  have been reached by prefixes that are enabling-compatible, and that  $\text{map}(\mathcal{E}(s)) = \mathcal{E}(\mathcal{C})$ . Given a state of the product  $(s, \mathcal{C})$  with  $\mathcal{C} \neq \perp$ , we will say that the state is in the timed domain, indicating that the timing analysis performed on  $CS_\theta$  can be applied to  $s$ .

The rules that define the enabling-compatible product are as follows:

### Transitions entering the timed domain

Transition	Conditions
$(s_1, \perp) \xrightarrow{e} (s_2, \top)$	$\text{enter} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \mathcal{E}(\top) \downarrow \Sigma_A \subseteq \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A$

These transitions are fired when the events enabled in  $\top$  are also enabled in  $s_2$ . Thus, timing analysis can start being applied from  $(s_2, \top)$ .

### Staying inside the timed domain

Transition	Conditions
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \mathcal{C}_1)$	$\text{inside1} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \mathcal{E}(s_1) \cap \Sigma_G \downarrow \Sigma_A = \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A$
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \mathcal{C}_2)$	$\text{inside2} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \mathcal{C}_1 \xrightarrow{e} \mathcal{C}_2 \in T_G \wedge \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A = \mathcal{E}(\mathcal{C}_2) \downarrow \Sigma_A$

inside1 corresponds to the condition in which  $e$  does not synchronize with  $G$ . Here the enableings of configuration  $\mathcal{C}_1$  must be preserved, *i.e.* the firing of  $e$  cannot disable or enable events in  $\Sigma_G$ .

For inside2, both  $A$  and  $G$  make a synchronized move which might affect the events from  $\Sigma_G$  in exactly the same way: if  $a \in \Sigma_G$  becomes enabled in  $A$  due to this move, it should also become enabled in  $G$ , and vice versa.

### Exiting or staying outside the timed domain

Transition	Conditions
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \perp)$	$\text{exit} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \neg(\text{enter} \vee \text{inside1} \vee \text{inside2})$

It can be shown that, in the enabling-compatible product, only the traces of the original LzTS which are enabling-compatible with the event structure are refined. This refinement excludes the traces which are not timing-consistent with respect to the timing constraints coming from the timing analysis on the event structure. All other traces are not changed, thus guaranteeing the conservativeness of the approach.

## B.2 Symbolic representation

In order to provide an efficient symbolic representation of LzTSs, we map them onto boolean algebras. Each state of the system is described by a unique vertex in the algebra. Thus, the sets of states of the system, the functions and transition relations that define the system behavior, and the properties for verification, are all modeled as boolean functions. Such functions can be represented using BDDs for efficiency.

### B.2.1 Encoding of a LzTS

Given a LzTS  $A = \langle S, \Sigma, T, s_0, \text{EnR} \rangle$  the system  $\langle 2^S, \cup, \cap, \emptyset, S \rangle$  is the boolean algebra of sets of states of  $A$ . As a consequence, each state  $s \in S$  can be represented by means of an *encoding function*  $Q : S \rightarrow \mathbb{B}^n$ , with  $n \geq \lceil \log_2(|S|) \rceil$ . That is, given the set of boolean variables  $\mathcal{V} = \{v_1, \dots, v_n\}$ , each state  $s \in S$  is encoded into a vertex  $(v_1, \dots, v_n) \in \mathbb{B}^n$ . Provided such encoding, any set of states  $P \subseteq S$  can be represented by a *characteristic (boolean) function*  $\mathcal{X}_P^Q : \mathbb{B}^n \rightarrow \mathbb{B}$  that evaluates to 1 for those vertexes of  $\mathbb{B}^n$  that correspond to states in the set  $P$ , encoded using  $Q$ . Whenever the encoding is understood, we simply write  $\mathcal{X}_P$ .

Characteristic functions can also be used to represent *binary relations* between sets of states. Given two sets of states  $P_1$  and  $P_2$ , to represent the binary relation  $\mathcal{R} \subseteq P_1 \times P_2$  it is necessary to use two different sets of variables to identify the elements of each set. *Current-state* variables  $v_1, \dots, v_n$  for  $P_1$  and *next-state* variables  $v'_1, \dots, v'_n$  for  $P_2$ . Thus, the cartesian product of a relation between  $P_1$  and  $P_2$  can be simply expressed as the product of the respective characteristic functions.

Let  $\mathcal{V} = \{v_1, \dots, v_n\}$  and  $\mathcal{V}' = \{v'_1, \dots, v'_n\}$  be respectively, the set of current and next-state boolean variables used to encode the states and transitions of the LzTS  $A = \langle S, \Sigma, T, s_0, \text{EnR} \rangle$ . In such a way that  $v'_i$  is the next-state variable corresponding to the current-state variable  $v_i$ , and vice versa. Thus, the usual definition of LzTS can be extended to contain  $\mathcal{V}$  and  $\mathcal{V}'$ , *i.e.*  $A = \langle \mathcal{V}, \mathcal{V}', S, \Sigma, T, s_0, \text{EnR} \rangle$ . Now, given an event  $e \in \Sigma$  we can represent its enabling region, its firing region and its transitions relation, by means of the following characteristic functions:

- $EF(e) : \mathbb{B}^n \rightarrow \mathbb{B}$  such that  $EF(e) = 1$  for all the states (encoded using  $\mathcal{V}$ ) belonging to the enabling region of  $e$ , *i.e.*  $\text{EnR}(e)$ .
- $FF(e) : \mathbb{B}^n \rightarrow \mathbb{B}$  such that  $FF(e) = 1$  for all the states (encoded using  $\mathcal{V}$ ) belonging to the firing region of  $e$ , *i.e.*  $\text{FR}(e)$ .
- $TR(e) : \mathbb{B}^{2n} \rightarrow \mathbb{B}$  such that  $TR(e) = 1$  for all the relations  $(s_1, s_2)$  such that there is a transition of event  $e$ ,  $s_1 \xrightarrow{e} s_2 \in T$ . The part of the relation corresponding to state  $s_1$  is encoded using the current-state variables in  $\mathcal{V}$ , whereas the part of the relation corresponding to state  $s_2$  is encoded using the next-state variables in  $\mathcal{V}'$ .

When characteristic functions of the enabling and firing regions are expressed using the set of next-state variables  $\mathcal{V}'$ , we will write  $EF'(e)$  and  $FF'(e)$ , respectively. Also, when the sets of variables in a transition relation are interchanged we will write  $TR(e)^{-1}$ .

### B.2.2 Encoding the state space of a LzCES

The space of configurations of a LzCES  $\langle \Sigma_G, \prec \rangle$  derived from a given encoded LzTS  $A = \langle \mathcal{V}, \mathcal{V}', S_A, \Sigma_A, T_A, s_{0A}, \text{EnR}_A \rangle$ , form a LzTS  $G = \langle C \cup \{\perp\}, \Sigma_G, T_G, \top, \text{EnR}_G \rangle$ . The config-

urations in  $\mathcal{C}$  can be identified in terms of the set of enabled events. Then for each configuration  $\mathcal{C}$  a characteristic function  $\gamma(\mathcal{C}) : \mathbb{B}^n \rightarrow \mathbb{B}$  (encoded using  $\mathcal{V}$ ) that identifies it can be built using such enabling information as follows:

$$\gamma(\mathcal{C}) = \prod_{e/i \in \Sigma_G} \begin{cases} EF(e) & \text{if } e \in \Sigma_A \wedge e/i \in \mathcal{E}(\mathcal{C}) \\ \overline{EF(e)} & \text{if } e \in \Sigma_A \wedge e/i \notin \mathcal{E}(\mathcal{C}) \end{cases}$$

Notice that the enabling information corresponds to that coming from the LzTS of reference form which the LzCES was derived. If  $\gamma$  is expressed using the set of next-state variables  $\mathcal{V}'$  we will write  $\gamma'(\mathcal{C})$ .

In some cases configurations in a CES cannot be distinguished by looking only at the enabling information, either because it is incomplete or because it is ambiguous itself. If that happens some extra encoding variables are required that help to disambiguate. For that purpose, let  $\mathcal{U} = \{u_1, \dots, u_m\}$  and  $\mathcal{U}' = \{u'_1, \dots, u'_m\}$  be respectively, the set of current and next-state *extra* boolean variables used to encode the configurations of the CES. Hence, in general, for each configuration  $\mathcal{C}$  a characteristic function  $\xi(\mathcal{C}) : \mathbb{B}^m \rightarrow \mathbb{B}$  (encoded using  $\mathcal{U}$ ) will exist. Again, if  $\xi$  is expressed using the set of next state variables  $\mathcal{U}'$  we will write  $\xi'(\mathcal{C})$ .

Finally one more boolean variable called  $IN$  will be used to indicate whether the enabling-compatibility is preserved during the composition process. That is if  $IN = 1$  in the characteristic function of a state it means that the enabling compatibility is being satisfied in it, while  $IN = 0$  will indicate the opposite. A corresponding next state variable  $IN'$  will also be used to properly define the new transition relations.

With all the above considerations, the LzTS corresponding to the space of configurations of a LzCES used in the enabling-compatible product will have the following form:  $G = \langle \mathcal{V} \cup \mathcal{U} \cup \{IN\}, \mathcal{V}' \cup \mathcal{U}' \cup \{IN'\}, C \cup \{\perp\}, \Sigma_G, T_G, \top, \text{EnR}_G \rangle$ .

### B.3 Computation of the new transition relations

For each event of the system being refined by the enabling-compatible product with a CES, its new transition relation is computed as a set of different parts. Each part corresponds to the different situations of the enabling-compatible product outlined above. The new transition relation is therefore computed as the addition of the parts.

#### B.3.1 Transitions entering the timed domain

Transition	Conditions
$(s_1, \perp) \xrightarrow{e} (s_2, \top)$	$\text{enter} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \mathcal{E}(\top) \downarrow \Sigma_A \subseteq \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A$

The enter condition for event  $e$  is computed as follows:

$$\text{enter}(e) = \overline{IN} \cdot \text{cond} \cdot TR(e) \cdot \gamma'(\top)$$

Then, the corresponding part of the new transition relation is computed as follows:

$$TR_{\text{enter}}(e) = \overline{IN} \cdot \text{cond} \cdot TR(e) \cdot \gamma'(\top) \cdot \xi'(\top) \cdot IN'$$

In the previous equations,  $\text{cond}$  depends on the way the CES was built and its relation with the original trace, if any. Thus:

$$\text{cond} = \begin{cases} 1 & \text{if not nodal point or no reference trace} \\ \prod_{e \in \mathcal{E}(\top) \downarrow \Sigma_A} \overline{EF}(e) & \text{if nodal point} \end{cases}$$

### B.3.2 Staying inside the timed domain: no synchronization

Transition	Conditions
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \mathcal{C}_1)$	$\text{inside1} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \mathcal{E}(s_1) \cap \Sigma_G \downarrow \Sigma_A = \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A$

The  $\text{inside1}$  condition for event  $e$  is computed as follows:

$$\text{inside1}(e) = IN \cdot \sum_{\mathcal{C}_1 \in C} \gamma(\mathcal{C}_1) \cdot \xi(\mathcal{C}_1) \cdot TR(e) \cdot \gamma'(\mathcal{C}_1)$$

Then, the corresponding part of the new transition relation is computed as follows:

$$TR_{\text{inside1}}(e) = \left( \sum_{\mathcal{C}_1 \in C} \gamma(\mathcal{C}_1) \cdot \xi(\mathcal{C}_1) \cdot TR(e) \cdot \gamma'(\mathcal{C}_1) \cdot \xi'(\mathcal{C}_1) \right) \cdot IN \cdot IN'$$

Notice that if  $e \in \mathcal{E}(C) \downarrow \Sigma_A$  and  $e$  is *self-disabling*, the product  $\gamma(C) \cdot TR(e) \cdot \gamma'(C)$  will be 0.

### B.3.3 Staying inside the timed domain: synchronization

Transition	Conditions
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \mathcal{C}_2)$	$\text{inside2} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \mathcal{C}_1 \xrightarrow{e} \mathcal{C}_2 \in T_G \wedge \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A = \mathcal{E}(\mathcal{C}_2) \downarrow \Sigma_A$

The  $\text{inside2}$  condition for event  $e$  is computed as follows:

$$\text{inside2}(e) = IN \cdot \sum_{\mathcal{C}_1 \xrightarrow{e/i} \mathcal{C}_2 \in T_G} \gamma(\mathcal{C}_1) \cdot \xi(\mathcal{C}_1) \cdot TR(e) \cdot \gamma'(\mathcal{C}_2)$$

Then, the corresponding part of the new transition relation is computed as follows:

$$TR_{\text{inside2}}(e) = \left( \sum_{\mathcal{C}_1 \xrightarrow{e/i} \mathcal{C}_2 \in T_G} \gamma(\mathcal{C}_1) \cdot \xi(\mathcal{C}_1) \cdot TR(e) \cdot \gamma'(\mathcal{C}_2) \cdot \xi'(\mathcal{C}_2) \right) \cdot IN \cdot IN'$$

### B.3.4 Transitions re-entering the timed domain

Transition	Conditions
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \top)$	$\text{enter} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \neg(\text{inside1} \vee \text{inside2}) \wedge$ $\wedge \mathcal{E}(\top) \downarrow \Sigma_A \subseteq \mathcal{E}(s_2) \cap \Sigma_G \downarrow \Sigma_A$

This case corresponds to *re-entering* the enabling-compatibility, *i.e.* being willing to exit but in fact entering again. The reenter condition for event  $e$  is computed as follows:

$$\text{reenter}(e) = IN \cdot \text{cond} \cdot \overline{(\text{inside1}(e) + \text{inside2}(e))} \cdot TR(e) \cdot \gamma'(\top)$$

Then, the corresponding part of the new transition relation is computed as follows:

$$TR_{\text{reenter}}(e) = \text{reenter}(e) \cdot \xi'(\top) \cdot IN'$$

Where  $\text{cond}$  is the same equation used to specify the enter condition.

### B.3.5 Exiting or staying outside the timed domain

Transition	Conditions
$(s_1, \mathcal{C}_1) \xrightarrow{e} (s_2, \perp)$	$\text{other} \equiv s_1 \xrightarrow{e} s_2 \in T_A \wedge \neg(\text{enter} \vee \text{inside1} \vee \text{inside2})$
$(s_1, \perp) \xrightarrow{e} (s_2, \perp)$	

The other condition for event  $e$  is computed as follows:

$$\text{other}(e) = \overline{(\text{enter}(e) + \text{inside1}(e) + \text{inside2}(e) + \text{reenter}(e))} \cdot TR(e)$$

Then, the corresponding part of the new transition relation is computed as follows:

$$TR_{\text{other}}(e) = \text{other}(e) \cdot \xi'(\perp) \cdot \overline{IN'}$$

### B.3.6 New transition relation

Finally, the new transition relation for event  $e$  is computed as the addition of the different parts computed above:

$$TR(e) = TR_{\text{enter}}(e) + TR_{\text{inside1}}(e) + TR_{\text{inside2}}(e) + TR_{\text{reenter}}(e) + TR_{\text{other}}(e)$$

### B.3.7 Lazy events

Due to the refinement of the state space imposed by the enabling-compatible product, some events become lazy. That is, the firing region becomes a strict subset of the enabling region. As a consequence their firing function must be also updated. Thus, for each  $e \in \Sigma_A$  which becomes lazy inside the composition area due to some relative timing constraint, that is  $\text{FR}_G(e/i) \neq \text{EnR}_G(e/i)$ , we have that:

$$FF(e) = IN \cdot \left( \sum_{\mathcal{C} \in \text{FR}_G(e/i) \wedge e \in \Sigma_A} \gamma(\mathcal{C}) \cdot \xi(\mathcal{C}) \right) \cdot FF(e) + \overline{IN} \cdot FF(e)$$



### B.3.8 Initial state

Finally, the initial state of the resulting LzTS must be also updated. Its encoding must distinguish the fact that the initial state of the system belongs to the enabling-compatibility area determined by the product or not. Thus we have that:

$$\mathcal{X}_{s_0} = \mathcal{X}_{s_{0,A}} \cdot ( \gamma(\top) \cdot \xi(\top) \cdot IN + \overline{\gamma(\top)} \cdot \xi(\perp) \cdot \overline{IN} )$$

