In REX '93: A Decade of Concurrency, LNCS 803, Springer-Verlag, 1994

# Relationships between Models of Concurrency

*Mogens Nielsen*[*] *Vladimiro Sassone*[**]
d *Glynn Winskel*[*]

[**] Computer Science Department, Aarhus University, Denmark
[*] Dipartimento di Informatica, Università di Pisa, Italy

Abstract. Models for concurrency can be classified with respect to three relevant parameters: behaviour/system, interleaving/noninterleaving, linear/branching time. When modelling a process, a choice concerning such parameters corresponds to choosing the level of abstraction of the resulting semantics. The classifications are formalized through the medium of category theory.

Keywords. Semantics, Concurrency, Models for Concurrency, Categories.

Contents

# Introduction

Since its beginning, many efforts in the development of the *theory of concurrency* have been devoted to the study of suitable models for concurrent and distributed processes, and to the formal understanding of their semantics.

As a result, in addition to standard models like languages, automata and transition systems [7, 13], models like *Petri nets* [12], *process algebras* [10, 4], *Hoare traces* [5], *Mazurkiewicz traces* [9], *synchronization trees* [21] and *event structures* [11, 22] have been introduced.

The idea common to the models above is that they are based on atomic units of change—be they called transitions, actions, events or symbols from an alphabet—which are *indivisible* and constitute the steps out of which computations are built.

The difference between the models may be expressed in terms of the parameters according to which models are often classified. For instance, a distinction made explicitly in the theory of Petri nets, but sensible in a wider context, is that between so-called *"system"* models allowing an explicit representation of the (possibly repeating) states in a system, and *"behaviour"* models abstracting away from such information, which focus instead on the behaviour in terms of patterns of occurrences of actions over time. Prime examples of the first type are transition systems and Petri nets, and of the second type, trees, event structures and traces. Thus, we can distinguish among models according to whether they are *system* models or *behaviour* models, in this sense; whether they can faithfully take into account the difference between *concurrency* and *nondeterminism*; and, finally, whether they can represent the *branching structure* of processes, i.e., the points in which choices are taken, or not. Therefore, relevant parameters when looking at models for concurrency are

*Behaviour or System model;*
*Interleaving or Noninterleaving model;*
*Linear or Branching Time model.*

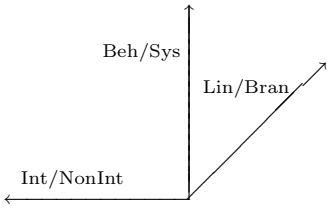
These parameters correspond to choices of the *level of abstraction* at which we examine processes and which are not necessarily fixed for a process once and for all. It is the actual application one has in mind for the formal semantics which *time by time* guides the choice of the abstraction level. It can therefore be of value to be able to move back and forth between the representation of a process in one model and its representation in another, if possible in a way which respects its structure. In other words, it is relevant to study translations between models, and particularly with respect to the three parameters above.

This work presents a classification of models for concurrency based on the three parameters, which represent a further step towards the identification of systematic connections between transition based models. In particular, we study a *representative* for each of the eight classes of models obtained by varying the parameters *behaviour/system*, *interleaving/noninterleaving* and *linear/branching*

| | | |
|---|---|---|
| Beh./Int./Lin. | Hoare languages | **HL** |
| Beh./Int./Bran. | synchronization trees | **ST** |
| Beh./Nonint./Lin. | deterministic labelled event structures | **dLES** |
| Beh./Nonint./Bran. | labelled event structures | **LES** |
| Sys./Int./Lin. | deterministic transition systems | **dTS** |
| Sys./Int./Bran. | transition systems | **TS** |
| Sys./Nonint./Lin. | deterministic transition systems with independence | **dTSI** |
| Sys./Nonint./Bran. | transition systems with independence | **TSI** |

Table 1: The models

in all the possible ways. Intuitively, the situation can be graphically represented, as in the picture below, by a three-dimensional frame of reference whose coordinate axes represent the three parameters.



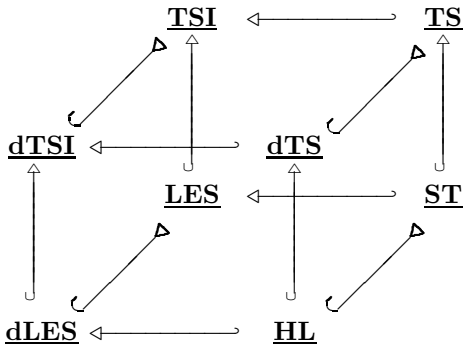

Our choices of models are summarized in Table 1. It is worth noticing that, with the exception of the new model of *transition systems with independence*, each model is well-known.

The formal relationships between models are studied in a *categorical* setting, using the standard categorical tool of *adjunctions*. The "translations" between models we shall consider are *coreflections* or *reflections*. These are particular kinds of adjunctions between two categories which imply that one category is *embedded*, fully and faithfully, in another.

Here we draw on our experience in recasting models for concurrency as categories, detailed in [23]. Briefly the idea is that each model (transition systems are one such model) will be equipped with a notion of morphism, making it into a category in which the operations of process calculi are universal constructions. The morphisms will preserve behaviour, at the same time respecting a choice of granularity of the atomic changes in the description of processes—the morphisms are forms of *simulations*. One role of the morphisms is to relate the behaviour of a construction on processes to that of its components. The reflections and coreflections provide a way to express that one model is embedded in (is more abstract than) another, even when the two models are expressed in very different mathematical terms. One adjoint will say how to embed the more abstract model in the other, the other will abstract away from some aspect of the representation. The preservation properties of adjoints can be used to show how a semantics in one model *translates* to a semantics in another.

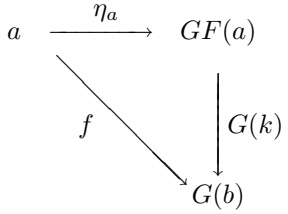
The picture below, in which arrows represent coreflections and the "back-

ward" arrows reflections, shows the "cube" of relationships (Theorem 7.17).

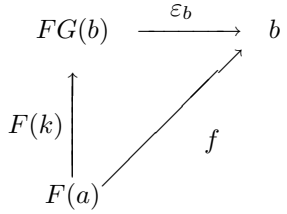



Among the many equivalent characterizations of the notion of adjunction (see e.g. [8, chap. IV]), we would like to recall here the one which is probably the most meaningful in our context, and which is, moreover, extensively used throughout the paper.

A functor $F\colon \underline{\mathbf{A}} \to \underline{\mathbf{B}}$ is said to be *left adjoint* to a functor $G\colon \underline{\mathbf{B}} \to \underline{\mathbf{A}}$, and conversely $G$ is *right adjoint* to $F$, in symbols $F \dashv G$, or $\langle F, G\rangle\colon \underline{\mathbf{A}} \rightharpoonup \underline{\mathbf{B}}$, if there exists a family of arrows $\eta = \{\eta_a\colon a \to GF(a) \text{ in } \underline{\mathbf{A}} \mid a \in \underline{\mathbf{A}}\}$, called the *unit* of the adjunction, which enjoys the following universal property: for any object $b \in \underline{\mathbf{B}}$ and any arrow $f\colon a \to G(b)$ in $\underline{\mathbf{B}}$, there exists a *unique* arrow $k\colon F(a) \to b$ such that $G(k) \circ \eta_a = f$, i.e., such that the following diagram commutes.




Equivalenty, $F$ is left adjoint to $G$ if there exists a family of arrows in $\underline{\mathbf{B}}$ $\varepsilon = \{\varepsilon_b : FG(b) \to b \mid b \in \underline{\mathbf{B}}\}$, the *counit* of the adjunction, such that for any arrow $f\colon F(a) \to b$, $a \in \underline{\mathbf{A}}$, there exists a *unique* arrow $k\colon a \to G(b)$ such that $\varepsilon_b \circ F(k) = f$, i.e.,

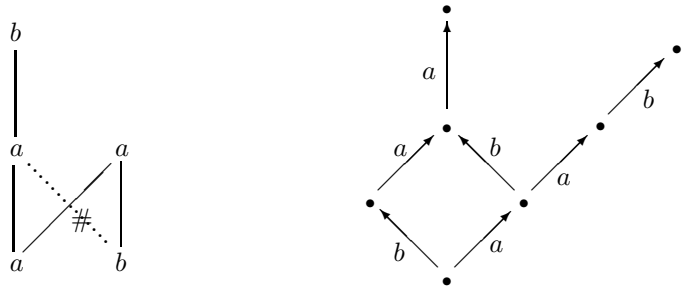



commutes.

An adjunction is called (generalized) *reflection* of $\underline{\mathbf{A}}$ in $\underline{\mathbf{B}}$, or $\underline{\mathbf{B}}$ is said *reflective* in $\underline{\mathbf{A}}$, if the the elements of the counit are isomorphisms. Dually, it is a

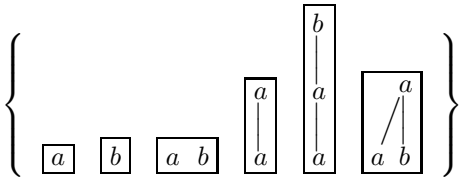
(generalized) *coreflection* of **B** in **A**, or **A** is *coreflective* in **B**, if the components of the unit are isomorphisms.

Generally speaking, the model chosen to represent a class is a canonical and universally accepted representative of that class. However, for the class of behavioural, linear-time, noninterleaving models there does not, at present, seem to be an obvious choice of a corresponding canonical model. The choice of deterministic labelled event structures is based, by analogy, on the observation that Hoare trace languages may be viewed as deterministic synchronization trees, and that labelled event structures are a canonical generalization of synchronization trees within noninterleaving models. The following picture is an example of such an event structure, together with its domain of configurations




Although not canonical, such a choice is certainly fair and, more important, it is *not* at all compelled. In order to show this, and for the sake of completeness, in Section 4, we investigate the relationship between this model and two of the most-studied, noninterleaving generalizations of Hoare languages in the literature: the pomsets of Pratt [14], and the traces of Mazurkiewicz [9].

Pomsets, an acronym for *p*artial *o*rdered *m*ulti*sets*, are labelled partial ordered sets. A noninterleaving representation of a system can be readily obtained by means of pomsets simply by considering the (multiset of) labels occurring in the run ordered by the *causal dependency* relation inherited from the events. The system itself is then represented by a set of pomsets. For instance, the labelled event structure given in the example discussed above can be represented by the following set of pomsets.




A simple but conceptually relevant observation about pomsets is that strings can be thought of as a particular kind of pomsets, namely those pomsets which are *finite* and *linearly* ordered. In other words, a pomset $\boxed{a_1 < a_2 < \cdots < a_n}$ represents the string $a_1 a_2 \cdots a_n$. On the other side of such correspondence, we

can think of (finite) pomsets as a generalization of the notion of word (string) obtained by relaxing the constraint which imposes that the symbols in a word be linearly ordered. This is why in the literature pomsets have also appeared under the name *partial words* [3]. The analogy between pomsets and strings can be pursued to the point of defining languages of partial words, called *partial languages*, as prefix-closed—for a suitable extension of this concept to pomsets—sets of pomsets on a given alphabet of labels.

Since our purpose is to study linear-time models, which are deterministic, we shall consider only pomsets without *autoconcurrency*, i.e., pomsets such that all the elements carrying the same label are linearly ordered. Following [19], we shall refer to this kind of pomsets as *semiwords* and to the corresponding languages as *semilanguages*. We shall identify a category **dSL** of deterministic semilanguages equivalent to the category of deterministic labelled event structures. Although pomsets have been studied extensively (see e.g. [14, 2, 3]), there are few previous results about formal relationships of pomsets with other models for concurrency.

Mazurkiewicz trace languages [9] are defined on an alphabet $L$ together with a symmetric irreflexive binary relation $I$ on $L$, called the *independence relation*. The relation $I$ induces an equivalence on the strings of $L^*$ which is generated by the simple rule

$$\alpha ab\beta \simeq \alpha ba\beta \quad \text{if} \quad a\ I\ b,$$

where $\alpha, \beta \in L^*$ and $a, b \in L$. A trace language is simply a subset $M$ of $L^*$ which is prefix-closed and $\simeq$-closed, i.e., $\alpha \in M$ and $\alpha \simeq \beta$ implies $\beta \in M$. It represents a system by representing all its possible behaviours as the sequences of (occurrences of) events it can perform. Since the independence relation can be taken to indicate the events which are *concurrent* to each other, the relation $\simeq$ does nothing but relate runs of the systems which differ only in the order in which independent events occur.

However, Mazurkiewicz trace languages are too abstract to describe faithfully labelled event structures. Consider for instance the labelled event structure shown earlier. Clearly, any trace language with alphabet $\{a, b\}$ able to describe such a labelled event structure must be such that $ab \simeq ba$. However, it cannot be such that $aba \simeq aab$. Thus, we are forced to move from the well-known model of trace languages. We shall introduce here a new notion of *generalized Mazurkiewicz trace language*, in which the independence relation is *context-dependent*. For instance, the event structure shown in the above picture will be represented by a trace language in which $a$ is independent from $b$ *at* $\epsilon$, i.e., after the empty string, in symbols $a\ I_\epsilon\ b$, but $a$ is *not* independent from $b$ *at* $a$, i.e., after the string $a$ has appeared, in symbols $a\ \not I_a\ b$. In particular, we shall present a category **GTL** of generalized trace languages which is equivalent to the category **dLES** of deterministic labelled event structures. We remark that a similar idea of generalizing Mazurkiewicz trace languages has been considered also in [6].

Summing up, Section 4 presents the chain of equivalences

$$\mathbf{dSL} \cong \mathbf{dLES} \cong \mathbf{GTL}$$

which, besides identifying models which can replace **dLES** in our classification, also introduce interesting *deterministic behavioural models* for concurrency and formalizes their mutual relationships.

Here most of the proofs are omitted. The reader interested in more details is referred to [15, 16]. Some of the results presented here will appear in [23].

# 1 Preliminaries

In this section, we study the interleaving models. We start by briefly recalling some well-known relationships between languages, trees and transition systems [23], and then, we study how they relate to deterministic transition systems.

DEFINITION 1.1 *(Labelled Transition Systems)*
*A labelled transition system is a structure* $T = (S, s^I, L, Tran)$ *where* $S$ *is a set of states;* $s^I \in S$ *is the initial state,* $L$ *is a set of labels, and* $Tran \subseteq S \times L \times S$ *is the transition relation.*

The fact that $(s, a, s') \in Tran_T$—also denoted by $s \xrightarrow{a} s'$, when no ambiguity is possible—indicates that the system can evolve from state $s$ to state $s'$ performing an action $a$. The structure of transition systems immediately suggests the right notion of morphism: initial states must be mapped to initial states, and for every action the first system can perform in a given state, it must be possible for the second system to perform the corresponding action—if any—from the corresponding state. This guarantees that morphisms are *simulations*.

DEFINITION 1.2 *(Labelled Transition System Morphisms)*
*Given the labelled transition systems* $T_0$ *and* $T_1$*, a morphism* $h: T \to T'$ *is a pair* $(\sigma, \lambda)$*, where* $\sigma: S_{T_0} \to S_{T_1}$ *is a function and* $\lambda: L_{T_0} \rightharpoonup L_{T_1}$ *a partial function, such that*[1]

*i)* $\sigma(s^I_{T_0}) = s^I_{T_1}$;

*ii)* $(s, a, s') \in Tran_{T_0}$ *implies* *if* $\lambda{\downarrow}a$ *then* $\Big(\sigma(s), \lambda(a), \sigma(s')\Big) \in Tran_{T_1}$;
*otherwise* $\sigma(s) = \sigma(s')$.

It is immediate to see that labelled transition systems and labelled transition system morphisms, when the obvious componentwise composition of morphisms is considered, give a category, which will be referred to as **TS**.

Since we shall deal often with partial maps, we assume the standard convention that whenever a statement involves values yielded by partial functions, we implicitily assume that they are defined.

A particularly interesting class of transition systems is that of synchronization trees, i.e., the tree-shaped transition systems.

[1] We use $f{\downarrow}x$ to mean that a partial function $f$ is defined on argument $x$.

DEFINITION 1.3 *(Synchronization Trees)*
*A synchronization tree is an acyclic, reachable transition system $S$ such that*

$$(s', a, s),\ (s'', b, s) \in Tran_S \qquad implies \qquad s' = s'' \quad and \quad a = b$$

*We shall write* **ST** *to denote the full subcategory of* **TS** *consisting of synchronization trees.*

In a synchronization tree the information about the internal structure of systems is lost, and only the information about their behaviour is mantained. In other words, it is not anymore possible to discriminate between a system which reachs again and again the same state, and a system which passes through a sequence of states, as far as they are able to perform the same action. However, observe that the nondeterminism present in a state can still be expressed in full generality. In this sense, synchronization trees are branching time and interleaving models of behaviour.

A natural way of studying the behaviour of a system consists of considering its computations as a synchronization tree, or, in other words, of *"unfolding"* the transition system by decorating each state with the history of the computation which reached it.

DEFINITION 1.4 *(Unfoldings of Transition Systems)*
*Let $T$ be a transition system. A path $\pi$ of $T$ is $\epsilon$, the empty path, or a sequence $t_1 \cdots t_n$, $n \geq 1$, where*

*i)* $t_i \in Tran_T$, $i = 1, \ldots, n$;

*ii)* $t_1 = (s_T^I, a_1, s_1)$ *and* $t_i = (s_{i-1}, a_i, s_i)$, $i = 2, \ldots, n$.

*We shall write $Path(T)$ to indicate the set of paths of $T$ and $\pi_s$ to denote a generic path leading to state $s$.*
*Define $ts.st(T)$ to be the synchronization tree $(Path(T), \epsilon, L_T, Tran)$, where*

$$\Big((t_1 \cdots t_n), a, (t_1 \cdots t_n t_{n+1})\Big) \in Tran$$
$$\Leftrightarrow \quad t_n = (s_{n-1}, a_n, s_n) \quad and \quad t_{n+1} = (s_n, a, s_{n+1})$$

This procedure amounts to abstracting away from the internal structure of a transition system and looking at its behaviour. It is very interesting to notice that this simple construction is functorial and, moreover, that if forms the right adjoint to the inclusion functor of **ST** in **TS**. In other words, the category of synchronization trees is coreflective in the category of transition systems. The *counit* of such adjunction is the morphism $(\phi, id_{L_T})\colon ts.st(T) \to T$, where $\phi\colon Path(T) \to S_T$ is given by $\phi(\epsilon) = s_T^I$, and $\phi\Big((t_1 \cdots t_n)\Big) = s$ if $t_n = (s', a, s)$.

While looking at the behaviour of a system, a further step of abstraction can be achieved forgetting also the branching structure of a tree. This leads to another well-know model of behaviour: *Hoare languages.*

Definition 1.5 *(Hoare Languages)*
*A Hoare language is a pair $(H, L)$, where $\varnothing \neq H \subseteq L^*$, and $sa \in H \Rightarrow s \in H$. A partial map $\lambda\colon L_0 \rightharpoonup L_1$ is a morphism of Hoare languages from $(H_0, L_0)$ to $(H_1, L_1)$ if for each $s \in H_0$ it is $\hat{\lambda}(s) \in H_1$, where $\hat{\lambda}\colon L_0^* \to L_1^*$ is defined by*

$$\hat{\lambda}(\epsilon) = \epsilon \qquad \text{and} \qquad \hat{\lambda}(sa) = \begin{cases} \hat{\lambda}(s)\lambda(a) & \text{if } \lambda{\downarrow}a; \\ \hat{\lambda}(s) & \text{otherwise.} \end{cases}$$

*These data give the category* **HL** *of Hoare languages.*

Observe that any language $(H, L)$ can be seen as a synchronization tree just by considering the strings of the language as states, the empty string being the initial state, and defining a transition relation where $s \xrightarrow{a} s'$ if and only if $sa = s'$. Let $hl.st\Big((H, L)\Big)$ denote such a synchronization tree.

On the contrary, given a synchronization tree $S$, it is immediate to see that the strings of labels on the paths of $S$ form a Hoare language. More formally, for any transition system $T$ and any path $\pi = (s_T^I, a_1, s_1) \cdots (s_{n-1}, a_n, s_n)$ in $Path(T)$, define $Act(\pi)$ to be the string $a_1 \cdots a_n \in L_T^*$. Moreover, let $Act(T)$ denote the set of strings

$$\Big\{ Act(\pi) \;\Big|\; \pi \in Path(T) \Big\}.$$

Then, the language associated to $S$ is $st.hl(S) = Act(S)$, and simply by defining $st.hl\Big((\sigma, \lambda)\Big) = \lambda$, we obtain a functor $st.hl\colon \mathbf{ST} \to \mathbf{HL}$. Again, this constitutes the left adjoint to $hl.st\colon \mathbf{HL} \to \mathbf{ST}$ and given above. The situation is illustrated below, where $\hookrightarrow\!\triangleright$ represents a coreflection and $\hookleftarrow\!\triangleleft$ a reflection.

Theorem 1.6

$$\mathbf{HL} \hookleftarrow\!\triangleleft\; \mathbf{ST} \hookrightarrow\!\triangleright\; \mathbf{TS}$$

The existence of a (co)reflection from category **A** to **B** tells us that there is a full subcategory of **B** which is *equivalent* to **A** (in the formal sense of equivalences of categories). Therefore, once we have established a (co)reflection, it is sometime interesting to indentify such subcategories. In the case of **HL** and **ST** such a question is answered below.

Proposition 1.7 *(Languages are deterministic Trees)*
*The full subcategory of* **ST** *consisting of those synchronization trees which are deterministic, say* **dST**, *is equivalent to the category of Hoare languages.*

# 2 Deterministic Transition Systems

Speaking informally behaviour/system and linear/branching are independent parameters, and we expect to be able to forget the branching structure of a

transition system without necessarily losing all the internal structure of the system. This leads us to identify a class of models able to represent the internal structure of processes without keeping track of their branching, i.e., the points at which the choices are actually taken. A suitable model is given by *deterministic transition systems*.

DEFINITION 2.1 *(Deterministic Transition Systems)*
*A transition system $T$ is deterministic if*

$$(s, a, s'),\ (s, a, s'') \in \mathit{Tran}_T \quad \mathit{implies} \quad s' = s''.$$

*Let* **dTS** *be the full subcategory of* **TS** *consisting of those transition systems which are deterministic.*

Consider the binary relation $\simeq$ on the state of a transition system $T$ defined as the least equivalence which is *forward closed*, i.e.,

$$s \simeq s' \text{ and } (s, a, u),\ (s', a, u') \in \mathit{Tran}_T \quad \Rightarrow \quad u \simeq u';$$

and define $ts.dts(T) = \Big(S/\simeq, [s_T^I]_\simeq, L_T, \mathit{Tran}_\simeq\Big)$, where $S/\simeq$ are the equivalence classes of $\simeq$ and

$$\Big([s]_\simeq, a, [s']_\simeq\Big) \in \mathit{Tran}_\simeq \quad \Leftrightarrow \quad \exists(\bar{s}, a, \bar{s}') \in \mathit{Tran}_T \quad \text{with } \bar{s} \simeq s \text{ and } \bar{s}' \simeq s'.$$

It is easy to see that $ts.dts(TS)$ is a deterministic transition system. Actually, this construction defines a functor which is left adjoint to the inclusion **dTS** $\hookrightarrow$ **TS**. In the following we briefly sketch the proof of this fact. Since confusion is never possible, we shall not use different notations for different $\simeq$'s.

Given a transition system morphism $(\sigma, \lambda)\colon T_0 \to T_1$, define $ts.dts\Big((\sigma, \lambda)\Big)$ to be $(\bar{\sigma}, \lambda)$, where $\bar{\sigma}\colon S_{T_0}/\simeq \to S_{T_1}/\simeq$ is such that $\bar{\sigma}\Big([s]_\simeq\Big) = [\sigma(s)]_\simeq$.

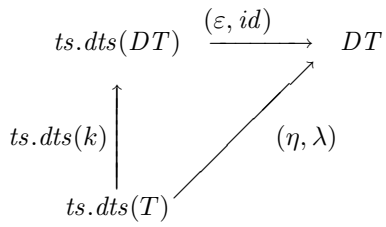
PROPOSITION 2.2 *($ts.dts\colon$* **TS** $\to$ **dTS** *is a functor)*
*The pair $(\bar{\sigma}, \lambda)\colon ts.dts(T_0) \to ts.dts(T_1)$ is a transition system morphism.*

*Proof.* First, we show that $\bar{\sigma}$ is well-defined.
Suppose $(s, a, s'),\ (s, a, s'') \in \mathit{Tran}_{T_0}$. Now, if $\lambda{\uparrow}a$, then $\sigma(s') = \sigma(s) = \sigma(s'')$. Otherwise, $\Big(\sigma(s), \lambda(a), \sigma(s')\Big),\ \Big(\sigma(s), \lambda(a), \sigma(s'')\Big) \in \mathit{Tran}_{T_1}$. Therefore, in both cases, $\sigma(s') \simeq \sigma(s'')$. Now, since $(s, a, s') \in \mathit{Tran}_{T_0}$ implies $\Big(\sigma(s), \lambda(a), \sigma(s')\Big) \in \mathit{Tran}_{T_1}$ or $\sigma(s) = \sigma(s')$, it easily follows that $\sigma(\simeq) \subseteq \simeq$. It is now easy to show that $(\bar{\sigma}, \lambda)$ is a morphism. ✓

It follows easily from the previous proposition that $ts.dts$ is a functor.

Clearly, for a deterministic transition system, say $DT$, since there are no pairs of transitions such that $(s, a, s'),\ (s, a, s'') \in \mathit{Tran}_{DT}$, we have that $\simeq$ is the identity. Thus, we can choose a candidate for the counit by considering, for any deterministic transition system $DT$, the morphism $(\varepsilon, id)\colon ts.dts(DT) \to DT$, where $\varepsilon([s]_\simeq) = s$. Let us show it enjoys the couniversal property.

PROPOSITION 2.3 *($(\varepsilon, id)\colon ts.dts(DT) \to DT$ is couniversal)*
*For any deterministic transition system $DT$, any transition system $T$ and any morphism $(\eta, \lambda)\colon ts.dts(T) \to DT$, there exists a unique $k$ in* **TS** *such that $(\varepsilon, id) \circ ts.dts(k) = (\eta, \lambda)$.*




*Proof.* The morphism $k$ must be of the form $(\sigma, \lambda)$, for some $\sigma$. We choose $\sigma$ such that $\sigma(s) = \eta([s]_\simeq)$. With such a definition, it is immediate that $k$ is a transition system morphism. Moreover, the diagram commutes: $(\varepsilon, id) \circ ts.dts\Big((\sigma, \lambda)\Big) = (\varepsilon \circ \bar{\sigma}, \lambda)$, and $\varepsilon\Big(\bar{\sigma}([s]_\simeq)\Big) = \varepsilon([\sigma(s)]_\simeq) = \sigma(s) = \eta([s]_\simeq)$. To show uniqueness of $k$, suppose that there is $k'$ which makes the diagram commute. Necessarily, $k'$ must be of the kind $(\sigma', \lambda)$. Now, since $\sigma'([s]_\simeq) = [\sigma'(s)]_\simeq$, in order for the diagram to commute, it must be $\sigma'(s) = \eta([s]_\simeq)$. Therefore, $\sigma' = \sigma$ and $k' = k$. ✓

THEOREM 2.4 *($ts.dts \dashv \hookleftarrow$)*
*The functor $ts.dts$ is left adjoint to the inclusion functor* **dTS** $\hookrightarrow$ **TS**. *Therefore, the adjunction is a reflection.*

*Proof.* By standard results of Category Theory (see [8, chap. IV, pg. 81]). ✓

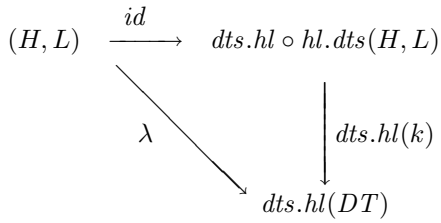
Next, we present a universal construction from Hoare languages to deterministic transition system. In particular, we show a coreflection **HL** $\hookrightarrow\!\!\triangleright$ **dTS**. Let $(H, L)$ be a language. Define $hl.dts(H, L) = (H, \epsilon, L, Tran)$, where $(s, a, sa) \in Tran$ for any $sa \in H$, which is trivially a deterministic transition system.

On the contrary, given a deterministic transition system $DT$, define the language $dts.hl(DT) = (Act(DT), L_{DT})$. Concerning morphisms, it is immediate that if $(\sigma, \lambda)\colon DT_0 \to DT_1$ is a transition system morphism, then $\lambda\colon Act(DT_0) \to Act(DT_1)$ is a morphism of Hoare languages. Therefore, defining $dts.hl\Big((\sigma, \lambda)\Big) = \lambda$, we have a functor from **dTS** to **HL**.

Now, consider the language $dts.hl \circ hl.dts(H, L)$. It contains a string $a_1 \cdots a_n$ if and only if the sequence $(\epsilon, a_1, a_1)(a_1, a_2, a_1a_2) \cdots (a_1 \cdots a_{n-1}, a_n, a_1 \cdots a_n)$ is in $Path\Big(hl.dts(T)\Big)$ if and only if $a_1 \cdots a_n$ is in $H$. It follows immediately that $id\colon (H, L) \to dts.hl \circ hl.dts(H, L)$ is a morphism of languages. We will show that $id$ is actually the unit of the coreflection.

PROPOSITION 2.5 *($id\colon (H, L) \to dts.hl \circ hl.dts(H, L)$ is universal)*
*For any Hoare language $(H, L)$, any deterministic transition system $DT$ and*

*any morphism* $\lambda\colon (H, L) \to dts.hl(DT)$*, there exists a unique* $k$ *in* **dTS** *such that* $dts.hl(k) = \lambda$*.*



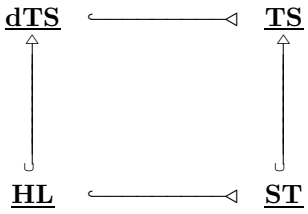

*Proof.* Observe that since $DT$ is deterministic, given a string $s \in Act(DT)$, there is exactly one state in $S_{DT}$ reachable from $s^I_{DT}$ with a path labelled by $s$. We shall use $state(s)$ to denote such a state. Then, define $k = (\sigma, \lambda)\colon hl.dts(H, L) \to DT$, where $\sigma(s) = state(\hat{\lambda}(s))$. Since $DT$ is deterministic and $\hat{\lambda}(s)$ is in $Act(DT)$, $(\sigma, \lambda)$ is well-defined and the rest of the proof follows easily. ✓

THEOREM 2.6 *(*$hl.dts \dashv dts.hl$*)*
*The map* $hl.dts$ *extends to a functor from* **HL** *to* **dTS** *which is left adjoint to* $dts.hl$*. Since the unit of the adjunction is an isomorphism, the adjunction is a coreflection.*

Observe that the construction of the deterministic transition system associated to a language coincides exactly with the construction of the corresponding synchronization tree. However, due to the different objects in the categories, the type of universality of the construction changes. In other words, the same construction shows that **HL** is *reflective* in **ST**—a full subcategory of **TS**—and *coreflective* in **dTS**—another full subcategory of **TS**.

Thus, we enriched the diagram at the end of the previous section and we have a square.

THEOREM 2.7 *(The Interleaving Surface)*




# 3 Noninterleaving vs. Interleaving Models

*Event structures* [11, 22] abstract away from the cyclic structure of the process and consider only events (strictly speaking event *occurrences*), assumed to

be the *atomic computational steps*, and the cause/effect relationships between them. Thus, we can classify event structures as behavioural, branching and noninterleaving models. Here, we are interested in labelled event structures.

DEFINITION 3.1 *(Labelled Event Structures)*
*A labelled event structure is a structure* $ES = (E, \#, \leq, \ell, L)$ *consisting of a set of events* $E$ *partially ordered by* $\leq$*; a symmetric, irreflexive relation* $\# \subseteq E \times E$*, the conflict relation, such that*

$$\{\, e' \in E \mid e' \leq e \,\} \text{ is finite for each } e \in E,$$

$$e \# e' \leq e'' \text{ implies } e \# e'' \text{ for each } e, e', e'' \in E;$$

*a set of labels* $L$ *and a labelling function* $\ell\colon E \to L$*. For an event* $e \in E$*, define* $\lfloor e \rfloor = \{\, e' \in E \mid e' \leq e \,\}$*. Moreover, we write* $\between$ *for* $\# \cup \{\, (e, e) \mid e \in E_{ES} \}$*. These data define a relation of concurrency on events:* $co = E_{ES}^2 \setminus (\leq \cup \leq^{-1} \cup \#)$*.*

*A labelled event structure morphism from* $ES_0$ *to* $ES_1$ *is a pair of partial functions* $(\eta, \lambda)$*, where* $\eta\colon E_{ES_0} \rightharpoonup E_{ES_1}$ *and* $\lambda\colon L_{ES_0} \rightharpoonup L_{ES_1}$ *are such that*

*i)* $\lfloor \eta(e) \rfloor \subseteq \eta(\lfloor e \rfloor)$,

*ii)* $\eta(e) \between \eta(e')$ *implies* $e \between e'$,

*iii)* $\lambda \circ \ell_{ES_0} = \ell_{ES_1} \circ \eta$.

*This defines the category* **LES** *of labelled event structures.*

The computational intuition behind event structures is simple: an event $e$ can occur when all its *causes*, i.e., $\lfloor e \rfloor \setminus \{e\}$, have occurred and no event which it is in conflict with has already occurred. This is formalized by the following notion of *configuration*.

DEFINITION 3.2 *(Configurations)*
*Given a labelled event structure* $ES$*, define the configurations of* $ES$ *to be those subsets* $c \subseteq E_{ES}$ *which are*

$$\textit{Conflict Free:} \quad \forall e_1, e_2 \in c,\ \textit{not } e_1 \# e_2$$

$$\textit{Left Closed:} \quad \forall e \in c\ \forall e' \leq e,\ e' \in c$$

*Let* $\mathcal{L}(ES)$ *denote the set of configurations of* $ES$*.*
*We say that* $e$ *is enabled at a configuration* $c$*, in symbols* $c \vdash e$*, if*

*(i)* $e \notin c$; *(ii)* $\lfloor e \rfloor \setminus \{e\} \subseteq c$; *(iii)* $e' \in E_{ES}$ *and* $e' \# e$ *implies* $e' \notin c$.

Given a finite subset $c$ of $E_{ES}$, we say that a total ordering of the elements of $c$, say $\{e_1 < e_2 < \cdots < e_n\}$, is a *securing* for $c$ if and only if $\{e_1, \ldots, e_{i-1}\} \vdash e_i$, for $i = 1, \ldots, n$. Clearly, $c$ is a finite configuration if and only if there exists a securing for it. We shall write a securing for $c$ as a string $e_1 e_2 \cdots e_n$, where $c = \{e_1, e_2, \ldots, e_n\}$ and $e_i \neq e_j$ for $i \neq j$, and, by abuse of notation, we shall consider such strings also configurations. Let $Sec(ES)$ denote the set of the securings of $ES$.

Definition 3.3 *(Deterministic Event Structures)*
*A labelled event structure $ES$ is deterministic if and only if for any $c \in \mathcal{L}(ES)$, and for any pair of events $e, e' \in E_{ES}$, whenever $c \vdash e$, $c \vdash e'$ and $\ell(e) = \ell(e')$, then $e = e'$. This defines the category* **dLES** *as a full subcategory of* **LES**.

In [20], it is shown that synchronization trees and labelled event structures are related by a a coreflection from **ST** to **LES**. As will be clear later, this gives us a way to see synchronization trees as an interleaving version of labelled event structures or, vicerversa, to consider labelled event structures as a generalization of synchronization trees to the non-interleaving case. In the following subsection, we give a brief account of this coreflection.

### Synchronization Trees and Labelled Event Structures

Given a tree $S$, define $st.les(S) = (Tran_S, \leq, \#, \ell, L_S)$, where

- $\leq$ is the least partial order on $Tran_S$ such that $(s, a, s') \leq (s', b, s'')$;
- $\#$ is the least hereditary, symmetric, irreflexive relation on $Tran_S$ such that $(s, a, s') \# (s, b, s'')$;
- $\ell\Big((s, a, s')\Big) = a$.

It is clear that $st.les(S)$ is a labelled event structure. Now, by defining $st.les\Big((\sigma, \lambda)\Big) = (\eta_\sigma, \lambda)$, where

$$\eta_\sigma\Big((s, a, s')\Big) = \begin{cases} \Big(\sigma(s), \lambda(a), \sigma(s')\Big) & \text{if } \lambda{\downarrow}a \\ \uparrow & \text{otherwise,} \end{cases}$$

it is not difficult to see that $st.les$ is a functor from **ST** to **LES**.

On the contrary, for a labelled event structure $ES$, define $les.st(ES)$ to be the structure $\Big(Sec(ES), \epsilon, L_{ES}, Tran\Big)$, where $(s, a, se) \in Tran$ if and only if $s,\ se \in Sec(ES)$ and $\ell_{ES}(e) = a$. Since a transition $(s, a, s')$ implies that $|s| < |s'|$ ($s$ is a string strictly shorter than $s'$), the transition system we obtain is certainly acyclic. Moreover, by definition of securing, it is reachable. Finally, if $(s, a, se),\ (s', a, s'e') \in Tran$ and $se = s'e'$, then obviously $s = s'$ and $e = e'$. Therefore, $les.st(ES)$ is a synchronization tree.

Concerning morphisms, for $(\eta, \lambda)\colon ES_0 \to ES_1$, define $les.st\Big((\eta, \lambda)\Big)$ to be $(\hat{\eta}, \lambda)$. This makes $les.st$ be a functor from **LES** to **ST**.

Consider now $les.st \circ st.les(S)$. Observe that there is a transition

$$\Big((s^I_S, a_1, s_1) \cdots (s_{n-1}, a_n, s_n), a, (s^I_S, a_1, s_1) \cdots (s_{n-1}, a_n, s_n)(s_n, a, s)\Big)$$

in $les.st \circ st.les(S)$ if and only if $(s^I_S, a_1, s_1) \cdots (s_{n-1}, a_n, s_n)(s_n, a, s)$ is a path in $S$. From this fact, and since $S$ and $les.st \circ st.les(S)$ are trees, it follows easily that there is an isomorphism between the states of $S$ and the states of $les.st \circ st.les(S)$, and that such an isomorphism is indeed a morphism of synchronization trees.

THEOREM 3.4 *($st.les \dashv les.st$)*
*For any synchronization tree $S$, the map $(\eta, id)\colon S \to les.st \circ st.les(S)$, where $\eta(s_S^I) = \epsilon$ and $\eta(s) = (s_S^I, a_1, s_1)\cdots(s_n, a, s)$, the unique path leading to $s$ in $S$, is a synchronization tree isomorphism.*

*Moreover, $\langle st.les, les.st\rangle\colon \underline{\mathbf{ST}} \rightharpoonup \underline{\mathbf{LES}}$ is an adjuction whose unit is given by the family of isomorphisms $(\eta, id)$. Thus, we have a coreflection of $\underline{\mathbf{ST}}$ into $\underline{\mathbf{LES}}$.*

Consider now a synchronization tree $S$ in $\underline{\mathbf{dST}}$, i.e., a deterministic tree. From the definition of $st.les$, it follows easily that $st.les(S)$ is a deterministic event structure; on the other hand, $les.st(ES)$ is a deterministic tree when $ES$ is deterministic. Thus, by general reason, the coreflection $\underline{\mathbf{ST}} \hookrightarrow \underline{\mathbf{LES}}$ restricts to a coreflection $\underline{\mathbf{dST}} \hookrightarrow \underline{\mathbf{dLES}}$, whence we have the following corollary.

THEOREM 3.5 *($\underline{\mathbf{HL}} \hookrightarrow \underline{\mathbf{dLES}}$)*
*The category $\underline{\mathbf{HL}}$ of Hoare languages is coreflective in the category $\underline{\mathbf{dLES}}$ of deterministic labelled event structures.*

*Proof.* It is enough to recall that $\underline{\mathbf{dST}}$ and $\underline{\mathbf{HL}}$ are *equivalent*. Then, the result follows by general reasons. ✓

To conclude this subsection, we make precise our claim of labelled event structures being a generalization of synchronization trees to the non-interleaving case. Once the counits of the above coreflections have been calculated, it is not difficult to prove the following results.

COROLLARY 3.6 *(L. Event Structures = S. Trees + Concurrency)*
*The full subcategory of $\underline{\mathbf{LES}}$ consisting of the labelled event structures $ES$ such that $co_{ES} = \varnothing$ is equivalent to $\underline{\mathbf{ST}}$.*

*The full subcategory of $\underline{\mathbf{dLES}}$ consisting of the deterministic labelled event structures $ES$ such that $co_{ES} = \varnothing$ is equivalent to $\underline{\mathbf{HL}}$.*

## TRANSITION SYSTEMS WITH INDEPENDENCE

Now, on the system level we look for a way of equipping transition systems with a notion of "concurrency" or "independence", in the same way as $\underline{\mathbf{LES}}$ may be seen as adding "concurrency" to $\underline{\mathbf{ST}}$. Moreover, such enriched transition systems should also represent the "system model" version of event structures. Several such models have appeared in the literature [17, 1, 18]. Here we choose a variation of these, the *transition systems with independence*.

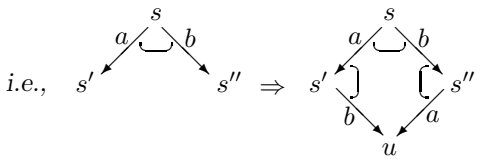
Transition systems with independence are transition systems with an independence relation actually carried by transitions. The novelty resides in the fact that the notion of *event* becomes now a derived notion. However, four axioms are imposed in order to guarantee the consistency of this with the intuitive meaning of event.

DEFINITION 3.7 *(Transition Systems with Independence)*
*A transition system with independence is a structure* $(S, s^I, L, Tran, I)$ *where* $(S, s^I, L, Tran)$ *is a transition system and* $I \subseteq Tran^2$ *is an irreflexive, symmetric relation, such that*

*i)* $(s,a,s') \sim (s,a,s'') \quad \Rightarrow \quad s' = s'';$

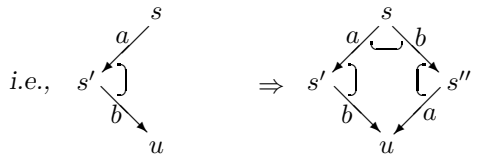
*ii)* $(s,a,s')\ I\ (s,b,s'') \quad \Rightarrow \quad \exists u.\ (s,a,s')\ I\ (s',b,u)$ *and* $(s,b,s'')\ I\ (s'',a,u);$

*i.e.,*




*iii)* $(s,a,s')\ I\ (s',b,u) \quad \Rightarrow \quad \exists s''.\ (s,a,s')\ I\ (s,b,s'')$ *and* $(s,b,s'')\ I\ (s'',a,u);$

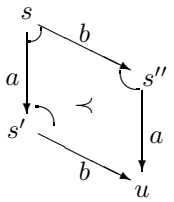
*i.e.,*




*iv)* $(s,a,s') \sim (s'',a,u)\ I\ (w,b,w') \quad \Rightarrow \quad (s,a,s')\ I\ (w,b,w');$

*where* $\sim$ *is least equivalence on transitions including the relation* $\prec$ *defined by*

$$(s,a,s') \prec (s'',a,u) \quad \Leftrightarrow \quad \begin{array}{l}(s,a,s')\ I\ (s,b,s'') \text{ and} \\ (s,a,s')\ I\ (s',b,u) \text{ and} \\ (s,b,s'')\ I\ (s'',a,u).\end{array}$$




*Morphisms of transition systems with independence are morphisms of the underlying transition systems which preserve independence, i.e., such that*

$$(s,a,s')\ I\ (\bar{s},b,\bar{s}') \text{ and } \lambda{\downarrow}a,\ \lambda{\downarrow}b \quad \Rightarrow \quad \Big(\sigma(s),\lambda(a),\sigma(s')\Big)\ I\ \Big(\sigma(\bar{s}),\lambda(b),\sigma(\bar{s}')\Big).$$

*These data define the category* **TSI** *of transition systems with independence. Moreover, let* **dTSI** *denote the full subcategory of* **TSI** *consisting of transition systems with independence whose underlying transition system is deterministic.*

Thus, transition systems with independence are precisely standard transition systems but with an additional relation expressing when one transition is independent of another. The relation $\sim$, defined as the reflexive, symmetric and transitive closure of a relation $\prec$ which simply identifies local "diamonds" of

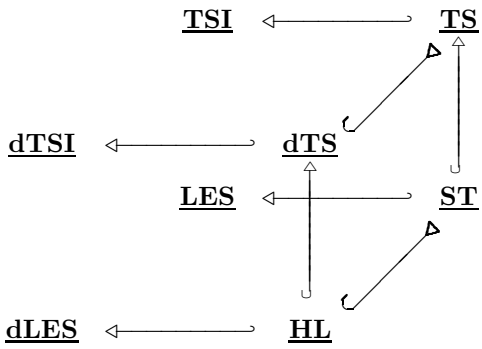
concurrency, expresses when two transitions represent occurrences of the same event. Thus, the equivalence classes $[(s, a, s')]_\sim$ of transitions $(s, a, s')$ are the *events* of the transition system with independence. In order to shorten notations, we shall indicate that transitions $(s, a, s')$, $(s, b, s'')$, $(s', b, u)$ and $(s'', a, u)$ form a diamond by writing $Diam\Big((s, a, s'), (s, b, s''), (s', b, u), (s'', a, u)\Big)$.

Concerning the axioms, property *(i)* states that the occurrence of an event at a state yields a unique state; property *(iv)* asserts that the independence relation respects events. Finally, conditions *(ii)* and *(iii)* describe intuitive properties of independence: two independent events which can occur at the same state, can do it in any order without affecting the reached state.

Transition systems with independence admit **TS** as a coreflective subcategory. In this case, the adjunction is easy. The left adjoint associates to any transition system $T$ the transition system with independence whose underlying transition system is $T$ itself and whose independence relation is empty. The right adjoint simply forgets about the independence, mapping any transition system with independence to its underlying transition system. From the definition of morphisms of transition systems with independence, it follows easily that these mappings extend to functors which form a coreflection **TS** $\hookrightarrow\!\triangleright$ **TSI**. Moreover, such a coreflection trivially restricts to a coreflection **dTS** $\hookrightarrow\!\triangleright$ **dTSI**.

So, we are led to the following diagram.

THEOREM 3.8 *(Moving along the "interleaving/noninterleaving" axis)*




# 4 Behavioural, Linear Time, Noninterleaving Models

A *labelled partial order* on $L$ is a triple $(E, \leq, \ell)$, where $E$ is a set, $\leq \subseteq E^2$ a partial order relation; and $\ell\colon E \to L$ is a *labelling* function. We say that a labelled partial order $(E, \leq, \ell)$ is *finite* if $E$ is so.

DEFINITION 4.1 *(Partial Words)*
*A partial word on $L$ is an isomorphism class of finite labelled partial orders.*

*Given a finite labelled partial order $p$ we shall denote with $[\![p]\!]$ the partial word which contains $p$. We shall also say that $p$ represents the partial word $[\![p]\!]$.*

A semiword is a partial word which does not exhibit autoconcurrency, i.e., such that all its subsets consisting of elements carrying the same label are linearly ordered. This is a strong simplification. Indeed, given a labelled partial order $p$ representing a semiword on $L$ and any label $a \in L$, such hypothesis allows us to talk *unequivocally* of the first element labelled $a$, of the second element labelled $a$, ..., the $n$-th element labelled $a$. In other words, we can represent $p$ unequivocally as a (strict) partial order whose elements are pairs in $L \times \omega$, $(a, i)$ representing the $i$-th element carrying label $a$. Thus, we are led to the following definition, where for $n$ a natural number, $[n]$ denote the initial segment of length $n$ of $\omega \setminus \{0\}$, i.e., $[n] = \{1, \ldots, n\}$.

DEFINITION 4.2 *(Semiwords)*
*A (canonical representative of a) semiword on an alphabet $L$ is a pair $x = (A_x, <_x)$ where*

- $A_x = \bigcup_{a \in L} \Big( \{a\} \times [n_a^x] \Big)$, *for some $n_a^x \in \omega$, and $A_x$ is finite;*
- $<_x$ *is a transitive, irreflexive, binary relation on $A_x$ such that*

$$(a, i) <_x (a, j) \quad \textit{if and only if} \quad i < j,$$

*where $<$ is the usual (strict) ordering on natural numbers.*

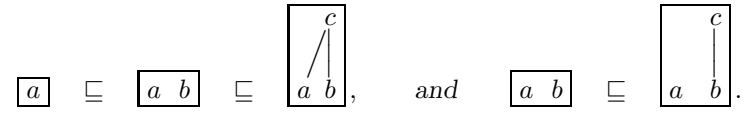
The semiword represented by $x$ is $\Big[\!\Big[(A_x, \leq, \ell)\Big]\!\Big]$, where $(a, i) \leq (b, j)$ if and only if $(a, i) <_x (b, j)$ or $(a, i) = (b, j)$, and $\ell\Big((a, i)\Big) = a$. However, exploiting in full the existence of such an easy representation, from now on, we shall make no distinction between $x$ and the semiword which it represents. In particular, as already stressed in Definition 4.2, with abuse of language, we shall refer to $x$ as a semiword. The set of semiwords on $L$ will be indicated by $SW(L)$. The usual set of words (strings) on $L$ is (isomorphic to) the subset of $SW(L)$ consisting of semiwords with *total* ordering.

A standard ordering used on words is the prefix order $\sqsubseteq$, which relates $\alpha$ and $\beta$ if and only if $\alpha$ is an initial segment of $\beta$. Such idea is easily extended to semiwords in order to define a prefix order $\sqsubseteq \subseteq SW(L) \times SW(L)$. Consider $x$ and $y$ in $SW(L)$. Following the intuition, for $x$ to be a prefix of $y$, it is necessary that the elements of $A_x$ are contained also in $A_y$ with the same ordering. Moreover, since new elements can be added in $A_y$ only "on the top" of $A_x$, no element in $A_y \setminus A_x$ may be less than an element of $A_x$. This is formalized by saying

$$\begin{array}{lllll} x \sqsubseteq y & \text{if and only if} & A_x \subseteq A_y & \text{and} & <_x = <_y \cap A_x^2 \\ & & & \text{and} & <_y \cap ((A_y \setminus A_x) \times A_x) = \varnothing. \end{array}$$

It is quickly realized that $\sqsubseteq$ is a partial order on $SW(L)$ and that it coincides with the usual prefix ordering on words.

EXAMPLE 4.3 *(Prefix Ordering)*
*As a few examples of the prefix ordering of semiwords, it is*




*However, it is neither the case that*

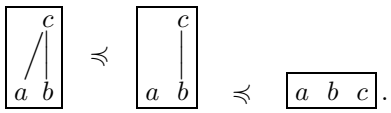

We shall use $Pref(x)$ to denote the set $\{y \in SW(L) \mid y \sqsubset x\}$ of *proper prefixes* of $x$. The set of maximal elements in $x$ will be denoted by $Max(x)$. Semiwords with a maximum element play a key role in our development. For reasons that will be clear later, we shall refer to them as to *events*.

Another important ordering is usually defined on semiwords: the *"smoother than"* order, which takes into account that a semiword can be extended just by relaxing its ordering. More precisely, $x$ is smoother than $y$, in symbols $x \preccurlyeq y$, if $x$ imposes more order contraints on the elements of $y$. Formally,

$$x \preccurlyeq y \qquad \text{if and only if} \qquad A_x = A_y \quad \text{and} \quad <_x \supseteq <_y.$$

It is easy to see that $\preccurlyeq \subseteq SW(L) \times SW(L)$ is a partial order. We shall use $Smooth(x)$ to denote the set of *smoothings* of $x$, i.e., $\{y \in SW(L) \mid y \preccurlyeq x\}$.

EXAMPLE 4.4 *(Smoother than Ordering)*
*The following few easy situations exemplify the smoother than ordering of semiwords.*




*On the other hand, neither*


## SEMILANGUAGES AND EVENT STRUCTURES

Semilanguages are a straightforward generalization of Hoare languages to prefix-closed subsets of $SW(L)$.

Definition 4.5 *(SemiLanguages)*
*A semilanguage is a pair $(SW, L)$, where $L$ is an alphabet and $SW$ is a set of semiwords on $L$ which is*

$$\begin{array}{ll} \textit{Prefix closed:} & y \in SW \text{ and } x \sqsubseteq y \quad \textit{implies} \quad x \in SW; \\ \textit{Coherent:} & \textit{Pref}(x) \subseteq SW \text{ and } |\textit{Max}(x)| > 2 \quad \textit{implies} \quad x \in SW. \end{array}$$

*Semilanguage $(SW, L)$ is deterministic if*

$$x, y \in SW \text{ and } \textit{Smooth}(x) \cap \textit{Smooth}(y) \neq \varnothing \quad \textit{implies} \quad x = y.$$

In order to fully understand this definition, we need to appeal to the intended meaning of semilanguages. A semiword in a semilanguage describes a (partial) run of a system in terms of the observable properties (labels) of the events which have occurred, together with the causal relationships which rule their interactions. Thus, the *prefix closedness* clause captures exactly the intuitive fact that any *initial segment* of a (partial) computation is itself a (partial) computation of the system.

In this view, the *coherence* axiom can be interpreted as follows. Suppose that there is a semiword $x$ whose proper prefixes are in the language, i.e., they are runs of the system, and suppose that $|\textit{Max}(x)| > 2$. This means that, given any pair of maximal elements in $x$, there is a computation of the system in which the corresponding events have both occurred. Then, in this case, the coherence axiom asks for $x$ to be a possible computation of the system, as well. In other words, we can look at coherence as to the axiom which forces a set of events to be conflict free if it is *pairwise* conflict free, as in [11] for *prime event structures* and in [9] for *proper trace languages*.

To conclude our discussion about Definition 4.5, let us analyze the notion of *determinism*. Remembering our interpretation of semiwords as runs of a system, it is easy to realize how the existence of distinct $x$ and $y$ such that $\textit{Smooth}(x) \cap \textit{Smooth}(y) \neq \varnothing$ would imply nondeterminism. In fact, if there were two different runs with a common linearization, then there would be two different computations exhibiting the same observable behaviour, i.e., in other words, two *non equivalent* sequences of events with the same strings of labels.

Also the notion of morphism of semilanguages can be derived smoothly as an extension of the existing one for Hoare languages.

Any $\lambda\colon L_0 \rightharpoonup L_1$ determines a partial function $\hat{\lambda}\colon SW(L_0) \rightharpoonup SW(L_1)$ which maps $x$ to its *relabelling* through $\lambda$, if this represents a semiword, and is undefined otherwise. Consider now semilanguages $(SW_0, L_0)$ and $(SW_1, L_1)$, and suppose for $x \in SW_0$ that $\hat{\lambda}$ is defined on $x$. Although one could be tempted to ask that $\hat{\lambda}(x)$ be a semiword in $SW_1$, this would be by far too strong a requirement. In fact, since in $\hat{\lambda}(x)$ the order $<_x$ is strictly preserved, morphisms would always strictly preserve causal dependency, and this would be out of tune with the existing notion of morphism for event structures, in which sequential tasks can be simulated by *"more concurrent"* ones. Fortunately enough, we have an easy

way to ask for the existence of a more concurrent version of $\hat{\lambda}(x)$ in $SW_1$. It consists of asking that $\hat{\lambda}(x)$ be a smoothing of some semiword in $SW_1$.

DEFINITION 4.6 *(Semilanguage Morphisms)*
*Given the semilanguages $(SW_0, L_0)$ and $(SW_1, L_1)$, a partial function $\lambda\colon L_0 \rightharpoonup L_1$ is a morphism $\lambda\colon (SW_0, L_0) \to (SW_1, L_1)$ if*

$$\forall\, x \in SW_0 \qquad \hat{\lambda}{\downarrow}x \quad \textit{and} \quad \hat{\lambda}(x) \in \mathit{Smooth}(SW_1).$$

It is worth observing that, if $(SW_1, L_1)$ is *deterministic*, there can be *at most one* semiword in $SW_1$, say $x_\lambda$, such that $\hat{\lambda}(x) \in \mathit{Smooth}(x_\lambda)$. In this case, we can think of $\lambda\colon (SW_0, L_0) \to (SW_1, L_1)$ as mapping $x$ to $x_\lambda$.

EXAMPLE 4.7
*Given $L_0 = \{a, b\}$ and $L_1 = \{c, d\}$, consider the deterministic semilanguages below.*

$$SW_0 = \left\{ \varnothing \quad \boxed{a} \quad \boxed{b} \quad \boxed{\begin{matrix} a \\ | \\ b \end{matrix}} \right\} \qquad SW_1 = \left\{ \varnothing \quad \boxed{c} \quad \boxed{d} \quad \boxed{c\ \ d} \right\}.$$

*Then, the function $\lambda$ which maps $a$ to $c$ and $b$ to $d$ is a morphism from $(SW_0, L_0)$ to $(SW_1, L_1)$. For instance,*

$$\hat{\lambda}\left( \boxed{\begin{matrix} a \\ | \\ b \end{matrix}} \right) = \boxed{\begin{matrix} c \\ | \\ d \end{matrix}} \preccurlyeq \boxed{c\ \ d}.$$

*Observe that the function $\lambda'\colon L_0 \to L_1$ which sends both $a$ and $b$ to $c$ is not a morphism since $\hat{\lambda}$ applied to $\boxed{b < a}$ gives $\boxed{c < c}$ which is not the smoothing of any semiword in $SW_1$, while $\lambda''\colon L_1 \to L_0$ which sends both $c$ and $d$ to $a$ is not a morphism from $(SW_1, L_1)$ to $(SW_0, L_0)$ since $\hat{\lambda}$ is undefined on $\boxed{c\ \ d}$.*

It can be shown that semilanguages and their morphisms, with composition that of partial functions, form a category whose full subcategory consisting of deterministic semilanguages will be denoted by **dSL**. In the following, we shall define *translation* functors between **dLES** and **dSL**.

Given a deterministic semilanguage $(SW, L)$ define $\mathit{dsl.dles}\Big((SW, L)\Big)$ to be the structure $(E, \leq, \#, \ell, L)$, where

- $E = \Big\{ e \;\Big|\; e \in SW,\ e \text{ is an } \textit{event}, \text{ i.e., } e \text{ has a maximum element} \Big\}$;
- $\leq\ = \sqsubseteq \cap E^2$;
- $\# = \Big\{ (e, e') \in E^2 \;\Big|\; e \text{ and } e' \text{ are incompatible wrt } \sqsubseteq \Big\}$;

- $\ell(e)$ is the label of the maximum element of $e$.

THEOREM 4.8
*$dsl.dles\Big((SW, L)\Big)$ is a deterministic labelled event structure.*

Consider now a deterministic labelled event structure $DES = (E, \leq, \#, \ell, L)$. Define $dles.dsl(DES)$ to be the structure $(SW, L)$, where

$$SW = \Big\{ \big[\!\big[(c, \leq \cap\, c^2, \ell_{|c})\big]\!\big] \;\Big|\; c \text{ is a } \textit{finite} \text{ configuration of } DES \Big\}.$$

THEOREM 4.9
*$dles.dsl(DES)$ is a deterministic semilanguage.*

It can be shown that *dsl.dles* and *dles.dsl* extend to functors which when composed with each other yield functors naturally isomorphic to identity functors. In other words, they form an *adjoint equivalence* [8, chap. III, pg. 91], i.e., an adjunction which is both a *reflection* and a *coreflection*. It is worthwhile noticing that this implies that the mappings *dsl.dles* and *dles.dsl* constitute a bijection between deterministic semilanguages and isomorphism classes of deterministic labelled event structures—isomorphism being identity up to the names of events.

THEOREM 4.10
*The categories* **dSL** *and* **dLES** *are equivalent.*

In fact, dropping the axiom of coherence in Definition 4.5 we get semilanguages equivalent to labelled *stable event structures* [22].

## TRACE LANGUAGES AND EVENT STRUCTURES

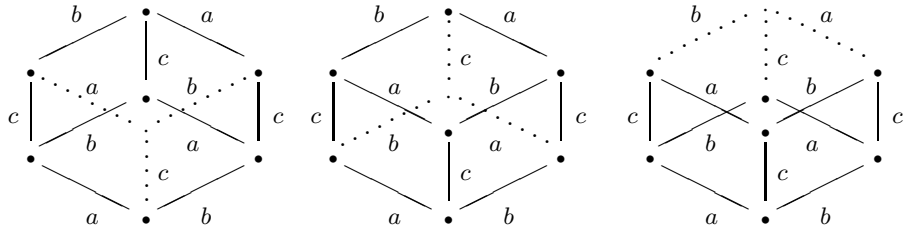
Generalized trace languages extend trace languages by considering an independence relation which may vary while the computation is progressing. Of course, we need a few axioms to guarantee the consistency of such an extension.

DEFINITION 4.11 *(Generalized Trace Languages)*
*A generalized trace language is a triple $(M, I, L)$, where $L$ is an alphabet, $M \subseteq L^*$ is a prefix-closed and $\simeq$-closed set of strings, $I\colon M \to 2^{L\times L}$ is a function which associates to each $s \in M$ a symmetric and irreflexive relation $I_s \subseteq L \times L$, such that*

$$\begin{array}{lll}
I \text{ is consistent:} & s \simeq s' \quad \text{implies} \quad I_s = I_{s'}; & \\
M \text{ is } I\text{-closed:} & a\ I_s\ b \quad \text{implies} \quad sab \in M; & \\
I \text{ is coherent:} & (i) \quad a\ I_s\ b \text{ and } a\ I_{sb}\ c \text{ and } c\ I_{sa}\ b \quad \text{implies} \quad a\ I_s\ c; & \\
& (ii) \quad a\ I_s\ c \text{ and } c\ I_s\ b & \\
& \qquad\qquad \text{implies} \quad (a\ I_s\ b \text{ if and only if } a\ I_{sc}\ b); &
\end{array}$$

*where $\simeq$ is the least equivalence relation on $L^*$ such that $sabu \simeq sbau$ if $a\ I_s\ b$.*

As in the case of trace languages, we have an equivalence relation $\simeq$ which equates those strings representing the same computation. Thus, $I$ must be consistent in the sense that it must associate the same independence relation to $\simeq$-equivalent strings. In order to understand the last two axioms, the following picture shows in terms of computations ordered by prefix the situations which those axioms forbid. There, the dots represent computations, the labelled edges represent the prefix ordering, and the dotted lines represent the computations forced in $M$ by the axioms.



$(A)$ $(B)$ $(C)$

It is easy to see that axiom (i) rules out the situation described by just the solid lines in (A)—impossible for stable event structures, while axiom (ii) eliminates cases (B)—which is beyond the descriptive power of *general event structures* [22] and (C)—impossible for event structures with *binary* conflict. They narrow down to those orderings of computations arising from prime event structures. It is worthwhile to observe that axiom (B) corresponds in our setting to what is called *"cube axiom"* in the setting of concurrent transition systems [18].

DEFINITION 4.12 *(Generalized Trace Language Morphisms)*
*Given the generalized trace languages $(M, I, L)$ and $(M', I', L')$, a partial function $\lambda\colon L \rightharpoonup L'$ is a morphism $\lambda\colon (M, I, L) \to (M', I', L')$ if*

$\lambda$ *preserves words:* $s \in M$ *implies* $\lambda^*(s) \in M'$;
$\lambda$ *respects independence:* $a\ I_s\ b$ *and* $\lambda{\downarrow}a,\ \lambda{\downarrow}b$ *implies* $\lambda(a)\ I'_{\lambda^*(s)}\ \lambda(b)$;

*where $\lambda^*$ is defined by $\lambda^*(\epsilon) = \epsilon$ and* $\lambda^*(sa) = \begin{cases} \lambda^*(s)\lambda(a) & \text{if } \lambda{\downarrow}a \\ \lambda^*(s) & \text{otherwise.} \end{cases}$

Generalized trace languages and their morphisms, under the usual composition of partial functions, form the category **GTL**.

A derived notion of event in generalized trace languages can be captured by the relation $\sim$ defined as the least equivalence such that

$$a\ I_s\ b \quad \text{implies} \quad sa \sim sba \qquad \text{and} \qquad s \simeq s' \quad \text{implies} \quad sa \sim s'a.$$

The events occurring in $s \in M$, denoted by $Ev(s)$, are the $\sim$-classes a representative of which occurs as a non empty prefix of $s$, i.e.,

$$\Big\{ [u]_\sim \;\Big|\; u \text{ is a non empty prefix of } s \Big\}.$$

It can be shown that $s \simeq s'$ if and only if $Ev(s) = Ev(s')$. Extending the notation, we shall write $Ev(M)$ to denote the events of $(M, I, L)$, i.e., the $\sim$-equivalence classes of non empty strings in $M$.

Now, given a generalized trace language $(M, I, L)$ define $gtl.dles\Big((M, I, L)\Big)$ to be the structure $(Ev(M), \leq, \#, \ell, L)$, where

- $[s]_\sim \leq [s']_\sim$ if and only if $\forall u \in M,\ [s']_\sim \in Ev(u)$ implies $[s]_\sim \in Ev(u)$;
- $[s]_\sim \ \# \ [s']_\sim$ if and only if $\forall u \in M,\ [s]_\sim \in Ev(u)$ implies $[s']_\sim \notin Ev(u)$;
- $\ell\Big([s]_\sim\Big) = a$ if and only if $s = s'a$.

THEOREM 4.13
*$gtl.dles\Big((M, I, L)\Big)$ is a deterministic labelled event structure.*

On the other hand, in order to define a generalized trace language from a deterministic labelled event structure $DES = (E, \leq, \#, \ell, L)$, consider

$$M = \Big\{\ell^*(e_1 \cdots e_n) \ \Big|\ \{e_1, \ldots, e_n\} \subseteq E \quad \text{and} \quad \{e_1, \ldots, e_{i-1}\} \vdash e_i,\ i = 1, \ldots, n\Big\}.$$

Since $DES$ is deterministic, any $s \in M$ identifies unequivocally a string of events $Sec(s) = e_1 \cdots e_n \in E^*$ such that $\{e_1, \ldots, e_{i-1}\} \vdash e_i,\ i = 1, \ldots, n$, and $\ell^*(e_1 \cdots e_n) = s$. Now, for any $s \in M$, take

$$I_s = \Big\{(a, b) \ \Big|\ sab \in M,\ Sec(sab) = xe_0e_1 \text{ and } e_0\ co\ e_1\Big\}.$$

Then, define $(M, I, L)$ to be $dles.gtl(DES)$.

THEOREM 4.14
*$dles.gtl(DES)$ is a generalized trace language.*

As in the case treated in the previous section, *dles.gtl* and *gtl.dles* extend to functors between **GTL** and **dLES** which form an adjoint equivalence. Such an equivalence restricts to an isomorphism of generalized trace languages and isomorphism classes of deterministic labelled event structures.

THEOREM 4.15
*Categories* **GTL** *and* **dLES** *are equivalent.*

The result extends to labelled *stable event structures* by dropping the *'only if'* implication in part *(ii)* of the coherence axiom of Definition 4.11. Of course, it follows from Theorem 4.10 and Theorem 4.15 that **dSL** and **GTL** are equivalent. In the full paper [16], we also define direct translations between such categories.

# 5 Transition Systems with Independence and Labelled Event Structures

In this section, we show that transition systems with independence are an extension of labelled event structures to a system model, by showing that there exists a coreflection from **LES** to **TSI**. To simplify our task, we split such a coreflection in two parts. First, we define the *unfolding* of transition systems with independence. To this aim, we introduce the category **oTSI** of occurrence transition systems with independence. Later, we shall show that labelled event structures are coreflective in **oTSI**, thus obtaining

$$\mathbf{LES} \hookrightarrow \mathbf{oTSI} \hookrightarrow \mathbf{TSI}.$$

Definition 5.1 *(Occurrence Transition Systems with Independence)*
*An occurrence transition system with independence is a transition system with independence $OTI = (S, s^I, L, Tran, I)$ which is reachable, acyclic and such that*

$$\begin{array}{l}(s', a, u) \neq (s'', b, u) \in Tran \quad \textit{implies} \\ \qquad \exists s.\ (s, b, s')\ I\ (s, a, s'') \textit{ and } (s, b, s')\ I\ (s', a, u) \\ \qquad\qquad\qquad\qquad\qquad \textit{and } (s, a, s'')\ I\ (s'', b, u),\end{array}$$

*or, in other words, $(s', a, u)$ and $(s'', b, u)$ form the bottom of a concurrency diamond $Diam\Big((s, a, s''), (s, b, s')(s'', b, u), (s', a, u)\Big)$.*

*Let* **oTSI** *denote the full subcategory of* **TSI** *whose objects are occurrence transition systems with independence.*

The key fact about occurrence transition systems with independence is that they are particularly well structured. In particular, it is possible to show that in an occurrence transition system with independence each diamond of concurrency is *not* degenerate, i.e., it consists of four *distinct* states. Other relevant features of occurrence transition systems with independence are stated by the following lemmas.

Given a transition system with independence $TI$, define $\simeq \subseteq Path(TI)^2$ to be the least equivalence relation such that

$$\pi_s(s, a, s')(s', b, u)\pi_v \simeq \pi_s(s, b, s'')(s'', a, u)\pi_v \quad \text{if} \quad Diam\Big((s, a, s'), (s, b, s'')(s', b, u), (s'', a, u)\Big).$$

Lemma 5.2
*Given an occurrence transition system with independence $OTI$, let $u$ be a state and $\pi_u$, $\pi'_u$ paths leading to it. Then $\pi_u \simeq \pi'_u$.*

LEMMA 5.3
*Given a path $\pi \in Path(OTI)$, at most one representative of any $\sim$-equivalence class can occur in $\pi$.*

## UNFOLDING TRANSITION SYSTEMS WITH INDEPENDENCE

Given a transition system with independence $TI = (S, s^I, L, Tran, I)$, we define $tsi.otsi(TI) = \Big(\Pi_{\simeq}, [\epsilon]_{\simeq}, L, Tran_{\simeq}, I_{\simeq}\Big)$, where

- $\Pi_{\simeq}$ is the quotient of $Path(TI)$ modulo $\simeq$;
- $\Big([\pi]_{\simeq}, a, [\pi']_{\simeq}\Big) \in Tran_{\simeq} \quad \Leftrightarrow \quad \exists (s, a, s') \in Tran$ such that $\pi' \simeq \pi(s, a, s')$;
- $\Big([\pi]_{\simeq}, a, [\pi']_{\simeq}\Big)\ I_{\simeq}\ \Big([\bar{\pi}]_{\simeq}, b, [\bar{\pi}']_{\simeq}\Big) \quad \Leftrightarrow$
  $\exists (s, a, s'), (\bar{s}, b, \bar{s}') \in Tran$ such that
  $(s, a, s')\ I\ (\bar{s}, b, \bar{s}')$, $\pi' \simeq \pi(s, a, s')$, and $\bar{\pi}' \simeq \bar{\pi}(\bar{s}, b, \bar{s}')$.

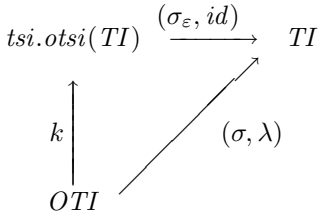
PROPOSITION 5.4
*The transition system $tsi.otsi(TI)$ is an occurrence transition system with independence.*

Figure 1 shows a simple example of unfolding of a transition system with independence. Next, we want to show that *tsi.otsi* extends to a functor for **TSI** to **oTSI** which is right adjoint to the inclusion functor **oTSI** $\hookrightarrow$ **TSI**. As a candidate for the counit of such an adjunction, consider the mapping $(\sigma_\varepsilon, id)\colon tsi.otsi(TI) \to TI$ where

$$\sigma_\varepsilon(\epsilon) = s^I_{TI} \qquad \text{and} \qquad \sigma_\varepsilon\Big([\pi_s]_{\simeq}\Big) = s.$$

By Lemma 5.2, we know that $\sigma_\varepsilon$ is well-defined. Then, it is not difficult to see that $(\sigma_\varepsilon, id)$ is a morphism of transition systems with independence.

PROPOSITION 5.5 *($(\sigma_\varepsilon, id)\colon tsi.otsi(TI) \to TI$ is couniversal)*
*For any occurrence transition system with independence $OTI$, transition system with independence $TI$ and morphism $(\sigma, \lambda)\colon OTI \to TI$, there exists a unique $k\colon OTI \to tsi.otsi(TI)$ in* **oTSI** *such that $(\sigma_\varepsilon, id) \circ k = (\sigma, \lambda)$.*



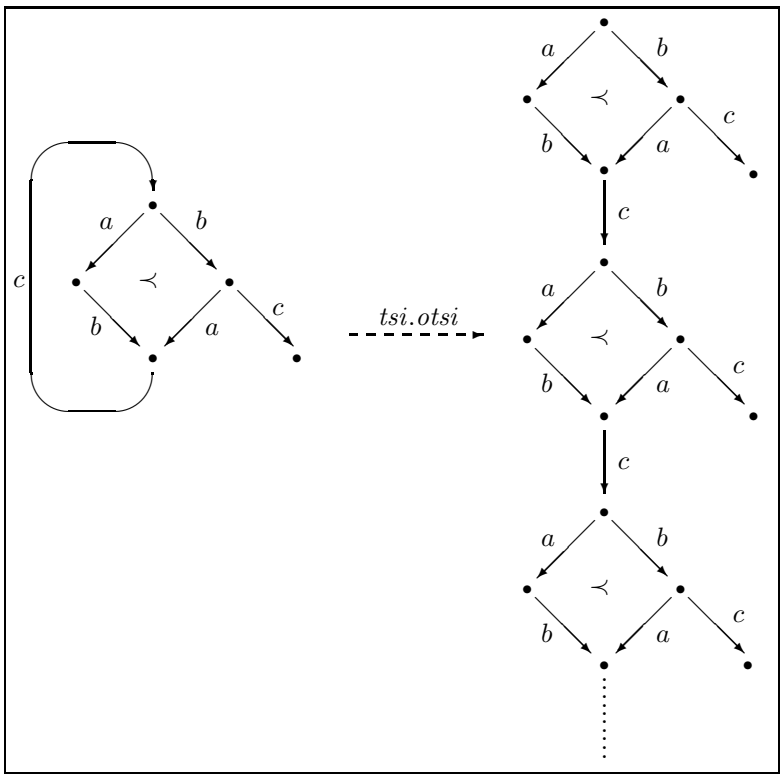



Figure 1: A transition system with independence $TI$ and $tsi.otsi(TI)$.

*Proof.* Clearly, in order for the diagram to commute, $k$ must be of the form $(\bar{\sigma}, \lambda)$. Consider the map $\bar{\sigma}(s) = [\sigma_\lambda(\pi_s)]_\simeq$, where $\sigma_\lambda \colon Path(OTI) \to Path(TI)$ is given by

$$\sigma_\lambda(\epsilon) = \epsilon; \qquad \sigma_\lambda\Big(\pi_s(s,a,s')\Big) = \begin{cases} \sigma_\lambda(\pi_s)\Big(\sigma(s), \lambda(a), \sigma(s')\Big) & \text{if } \lambda{\downarrow}a \\ \sigma_\lambda(\pi_s) & \text{otherwise.} \end{cases}$$

This definition is well-given: fixed $s$, let $\pi_s$ and $\pi_{s'}$ be two paths leading to $s$. Then, since $OTI$ is an occurrence transition system with independence, it is $\pi_s \simeq \pi_{s'}$, and since $(\sigma, \lambda)$ is a morphism, it follows easily that $\sigma_\lambda(\pi_s) \simeq \sigma_\lambda(\pi_{s'})$. Now it is not difficult to see that $(\bar{\sigma}, \lambda)$ is indeed a morphism of occurrence transition systems with independence.

In order to show that the diagram commutes, it is enough to observe that each $s$ is mapped to a $\simeq$-class of paths leading to $\sigma(s)$. Therefore, $\sigma_\varepsilon \circ \bar{\sigma}(s) = \sigma(s)$. The uniquess of $(\bar{\sigma}, \lambda)$ is easily obtained following the same line. In fact, the behaviour of $\bar{\sigma}$ is compelled on any $s$: $s^I_{OTI}$ must be mapped to $[\epsilon]_\simeq$, while a generic $s$ must mapped to a $\simeq$-equivalence class of paths leading to $\sigma(s)$. But, by Lemma 5.3, we know that there is a unique such class. ✓

THEOREM 5.6 *($\hookrightarrow \dashv$ tsi.otsi)*
*The construction tsi.otsi extends to a functor from* **TSI** *to* **oTSI** *which is right adjoint to the inclusion* **oTSI** $\hookrightarrow$ **TSI**.

It will be useful later to notice that this coreflection cuts down to a coreflections **doTSI** $\hookrightarrow$ **dTSI**, where **doTSI** is the full subcategory of **oTSI** consisting of deterministic transition systems. In order to achieve this result, it is clearly enough to show that *tsi.otsi* maps objects from **dTSI** to **doTSI**.

PROPOSITION 5.7 *(**doTSI** $\hookrightarrow$ **dTSI**)*
*If TI is deterministic, then tsi.otsi(TI) is deterministic.*

## OCCURRENCE TSI'S AND LABELLED EVENT STRUCTURES

Consider a labelled event structure $ES = (E, \leq, \#, \ell, L)$. Define $les.otsi(ES)$ to be the transition system with independence of the *finite configurations* of $ES$, i.e.,

$$les.otsi(ES) = \Big(\mathcal{L}_F(ES), \varnothing, L, \mathit{Tran}, I\Big),$$

where

- $\mathcal{L}_F(ES)$ is the set of finite configuration of $ES$;
- $(c, a, c') \in \mathit{Tran}$ if and only if $c = c' \setminus \{e\}$ and $\ell(e) = a$;
- $(c, a, c')\ I\ (\bar{c}, b, \bar{c}')$ if and only if $(c' \setminus c)\ co\ (\bar{c}' \setminus \bar{c})$.

By definition, $les.otsi(ES)$ is clearly an acyclic, reachable transition system. Moreover, $I \subseteq \mathit{Tran}^2$ is symmetric and irreflexive, since *co* is such. In order to show that it is an occurrence transition system with independence, it is important the following characterization of the relation $\sim$.

LEMMA 5.8
*Given $(c, a, c')$ and $(\bar{c}, a, \bar{c}') \in \mathit{Tran}$, we have $(c, a, c') \sim (\bar{c}, a, \bar{c}') \in \mathit{Tran}$ if and only if $(c' \setminus c) = (\bar{c}' \setminus \bar{c})$.*

It is now easy to show the following.

PROPOSITION 5.9
*The transition system $les.otsi(ES)$ is an occurrence transition system with independence.*

*Proof.* We verify just the property of occurrence transition systems with independence. Suppose that $(c', b, c) \neq (c'', a, c) \in \mathit{Tran}$. Then, we have $c = c' \cup \{e'\} = c'' \cup \{e''\}$. Since $c' \neq c''$, it must be $e' \neq e''$. Moreover, it is $e' \not\# e''$, since both events appear in $c$. It cannot be $e' < e''$ nor $e'' < e'$, because otherwise either $c'$ or $c''$ would not be a configuration. So, it is $e'\ co\ e''$. It follows that $\bar{c} = c' \setminus \{e'\} = c'' \setminus \{e''\}$ is a configuration such that $\mathit{Diam}\Big((\bar{c}, a, c'), (\bar{c}, b, c''), (c', b, c), (c'', a, c)\Big)$. ✓

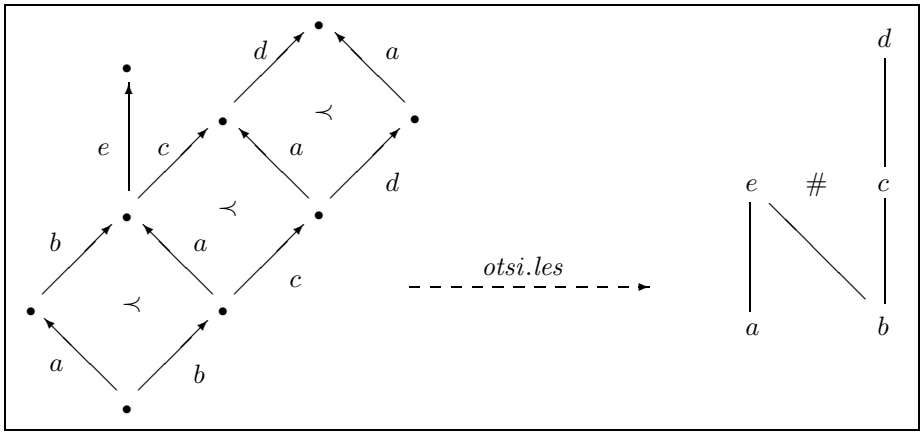



Figure 2: An occurrence transition system $OTI$ and $otsi.les(OTI)$.

Let us define the opposite transformation from **oTSI** to **LES**. For $OTI = (S, s^I, L, Tran, I)$ an occurrence transition system with independence, define $otsi.les(OTI)$ to be the structure $(Tran_\sim, \leq, \#, \ell, L)$ where

- $Tran_\sim$ is the set of the $\sim$-equivalence classes of $Tran$;
- $[(s,a,s')]_\sim < [(\bar{s},b,\bar{s}')]_\sim$ if and only if

$$\forall \pi(\underline{\bar{s}}, b, \underline{\bar{s}}') \in Path(OTI) \text{ with } (\underline{\bar{s}}, b, \underline{\bar{s}}') \sim (\bar{s}, b, \bar{s}'),$$
$$\exists (\underline{s}, a, \underline{s}') \sim (s, a, s') \text{ such that } (\underline{s}, a, \underline{s}') \in \pi,$$

  and $\leq$ is the reflexive closure of $<$;

- $[(s,a,s')]_\sim \# [(\bar{s},b,\bar{s}')]_\sim$ if and only if

$$\forall \pi \in Path(OTI),$$
$$\forall (\underline{\bar{s}}, b, \underline{\bar{s}}') \sim (\bar{s}, b, \bar{s}') \text{ and } \forall (\underline{s}, a, \underline{s}') \sim (s, a, s')$$
$$(\underline{s}, a, \underline{s}') \in \pi \Rightarrow (\underline{\bar{s}}, a, \underline{\bar{s}}') \notin \pi;$$

- $\ell\Big([(s,a,s')]_\sim\Big) = a$;

and we write $(s,a,s') \in \pi$ to mean that $(s,a,s')$ occurs in the path $\pi$. Of course, $otsi.les(OTI)$ is a labelled event structure. Figure 2 shows an example of the labelled event structure associated to an occurrence transition system with independence.

Given $(\sigma,\lambda)\colon OTI_0 \to OTI_1$, define $otsi.les\Big((\sigma,\lambda)\Big) = (\eta_\sigma,\lambda)$, where

$$\eta_\sigma\Big([(s,a,s')]_\sim\Big) = \begin{cases} \Big[\Big(\sigma(s),\lambda(a),\sigma(s')\Big)\Big]_\sim & \text{if } \lambda{\downarrow}a \\ \uparrow & \text{otherwise.} \end{cases}$$

Since $(s,a,s') \prec (\bar{s},a,\bar{s}')$ and $\lambda{\downarrow}a$ implies $(\sigma(s),\lambda(a),\sigma(s')) \sim (\sigma(\bar{s}),\lambda(a),\sigma(\bar{s}'))$, $\eta_\sigma$ is well-given. Indeed, it can be seen that this definition makes *otsi.les* be a functor from **oTSI** to **LES**, the proof being however not trivial.

In order to show that *otsi.les* and *les.otsi* form a coreflection, we need the following key lemma.

LEMMA 5.10
*Consider a path $\pi_s \in Path(OTI)$ and a class $[t]_\sim$ such that*

*i) for each $t'$ in $\pi_s$, it is $[t']_\sim \not\# [t]_\sim$ and $[t']_\sim \neq [t]_\sim$,*

*ii) for each $[t']_\sim < [t]_\sim$, there exists a representative of $[t']_\sim$ in $\pi_s$.*

*Then, there exists $(s,a,s') \in Tran_{OTI}$ with $(s,a,s') \sim t$.*

Next, we see that there is a one-to-one correspondence between the states of $OTI$ and the finite configurations of $otsi.les(OTI)$, or, in other words, states of $les.otsi \circ otsi.les(OTI)$. Consider the map $\mathcal{C}\colon S_{OTI} \to \mathcal{L}_F\Big(otsi.les(OTI)\Big)$ given by the correspondence $s \mapsto \Big\{[t]_\sim \;\Big|\; t \in \pi_s,\ \pi_s \in Path(OTI)\Big\}$. We already know that any path leading to $s$ contains the same equivalence classes, thus $\mathcal{C}$ is well-defined. Moreover, it is easy to show that $\mathcal{C}(s)$ is a finite configuration of $otsi.les(OTI)$.

On the contrary, let $c$ be a finite configuration of $otsi.les(OTI)$ and let $\varsigma = [t_0]_\sim[t_1]_\sim \cdots [t_n]_\sim$ be a *securing* for $c$. Then, there is a *unique* path $\pi_\varsigma = (s_0,a_1,s_1)\cdots(s_{n-1},a_n,s_n)$ such that $s^I_{OTI} = s_0$, $s_n = s$ and $[(s_{i-1},a_i,s_i)]_\sim = [t_i]_\sim$, for $i = 1,\ldots,n$. The existence of $\pi_\varsigma$ is a consequence of the previous Lemma 5.10 and its uniqueness follows from axiom *(iv)* of transition systems with independence.

It is important to observe that, although the actual path $\pi_\varsigma$ strictly depends on $\varsigma$, the final state reached does not. Therefore, we can define a map $\mathcal{S}\colon \mathcal{L}_F(otsi.les(OTI)) \to S_{OTI}$ by saying that $c \mapsto s$, where $s$ is the state reached by a path $\pi_\varsigma$ for a securing $\varsigma$ of $c$. Now, we can see that $\mathcal{C}$ is a set isomorphism with inverse $\mathcal{S}$.

PROPOSITION 5.11
*$(\mathcal{C},id)\colon OTI \to les.otsi \circ otsi.les(OTI)$ and $(\mathcal{S},id)\colon les.otsi \circ otsi.les(OTI) \to OTI$ are morphisms of transition systems. Moreover, $(\mathcal{S},id) = (\mathcal{C},id)^{-1}$.*

*In addition, $(\mathcal{S},id)$ is a transition system with independence morphism.*

However, $(\mathcal{C}, id)$ is not a morphism in **TSI**. It follows that $(\mathcal{S}, id)$, in general, is not an isomorphism of transition systems with independence. Consider now the property:

(E) $$t\ I\ t' \quad \Rightarrow \quad \exists s.\ (s, a, s') \sim t \quad \text{and} \quad (s, b, s'') \sim t'.$$

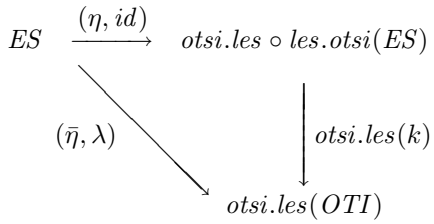
PROPOSITION 5.12
*OTI enjoys property* (E) *if and only if* $(\mathcal{C}, id)$ *is a morphism of transition systems with independence.*

The next step is to define, for each labelled event structure $ES$ a morphism $(\eta, id)\colon ES \to otsi.les \circ les.otsi(ES)$ as a candidate for the unit of the adjunction. Let us consider $\eta$ such that

$$\eta(e) = \Big[\Big(c, a, c \cup \{e\}\Big)\Big]_{\sim}.$$

We already know from Lemma 5.8 that $(c, a, c') \sim (\bar{c}, a, \bar{c}')$ if and only if $(c' \setminus c) = (\bar{c}' \setminus \bar{c})$. It follows immediately that $\eta$ is well-defined and is *injective*. Moreover, since any transition of $les.otsi(ES)$, say $(c, a, c')$, is associated with an event of $ES$, namely, $c' \setminus c$, we have that $\eta$ is also *surjective*. In fact, it can be shown that $(\eta, id)$ is an isomorphism of labelled event structures whose inverse is $(\bar{\eta}, id)$, where $\bar{\eta}\colon [(c, a, c')]_{\sim} \mapsto (c' \setminus c)$.

PROPOSITION 5.13 *($(\eta, id)\colon ES \to otsi.les \circ les.otsi(ES)$ is universal)*
*For any labelled event structure $ES$, any occurrence transition system with independence $OTI$ and any morphism $(\bar{\eta}, \lambda)\colon ES \to otsi.les(OTI)$, there exists a unique $k$ in* **oTSI** *such that $otsi.les(k) \circ (\eta, id) = (\bar{\eta}, \lambda)$.*




*Proof.* Let us define $k\colon les.otsi(ES) \to OTI$. Clearly, in order to make the diagram commute, $k$ must be of the form $(\sigma, \lambda)$, for some $\sigma$. Let us consider $\sigma\colon c \mapsto \mathcal{S}(\bar{\eta}(c))$, i.e., $(\sigma, \lambda) = (\mathcal{S}, id) \circ (\bar{\eta}, \lambda)\colon les.otsi(ES) \to les.otsi(otsi.les(OTI)) \to OTI$. Then, we have immediately that $\sigma$ is well-defined and that $(\sigma, \lambda)$ is a transition system with independence morphism and it not difficult to conclude the proof. ✓

THEOREM 5.14 *($les.otsi \dashv otsi.les$)*
*The map $les.otsi$ extends to a functor from* **LES** *to* **oTSI** *which is left adjoint to $otsi.les$. Since the unit of the adjunction is an isomorphism, the adjunction is a coreflection.*

Next, we show now that $(\mathcal{S}, id)$ is the counit of this coreflection. Actually, the task is fairly easy now: by general results in Category Theory [8, chap. IV, pg. 81], the counit of an adjunction can be determined through the unit as the unique morphism $\varepsilon\colon otsi.les \circ les.otsi(OTI) \to OTI$ such that $otsi.les(\varepsilon) \circ (\eta, id) = (id, id)$. However, in the proof of Proposition 5.13, we have identified a general way to find $\epsilon$. From it we obtain $\epsilon = (\mathcal{S}, id) \circ (id, id)$, which is $(\mathcal{S}, id)$.

The results we have shown earlier about $(\mathcal{S}, id)$ make it easy to identify the full subcategory of **oTSI** and, therefore, of **TSI** which is *equivalent* to **LES**, i.e., the category of those transition systems with independence which are (representations of) labelled event structure. Such a result gives yet another characterization of (the finite elements of) *coherent, finitary, prime algebraic domains*, i.e., *dI-domains*. Moreover, this axiomatization is given only in terms of conditions on the structure of transition systems.

By general results in Category Theory [8, chap. IV, pg. 91], an equivalence of categories is an adjunction whose unit and counit are both isomorphisms, i.e., which is both a reflection and a coreflection. Then, Proposition 5.12 gives us a candidate for the category of occurrence transition system with independence equivalent to **LES**: we consider $\mathbf{oTSI_E}$, the full subcategory of **oTSI** consisting of those occurrence transition systems with independence satisfying condition (E). To obtain the result, it is enough to verify that $les.otsi\colon \mathbf{LES} \to \mathbf{oTSI}$ actually lands in $\mathbf{oTSI_E}$. In fact, this guarantees that the adjunction $\langle les.otsi, otsi.les\rangle\colon \mathbf{LES} \rightharpoonup \mathbf{oTSI}$ restricts to an adjunction $\mathbf{LES} \rightharpoonup \mathbf{oTSI_E}$ whose unit and counit are again, respectively, $(\eta, id)$ and $(\mathcal{S}, id)$, which are isomorphisms.

THEOREM 5.15
*The categories* **LES** *and* $\mathbf{oTSI_E}$ *are equivalent.*

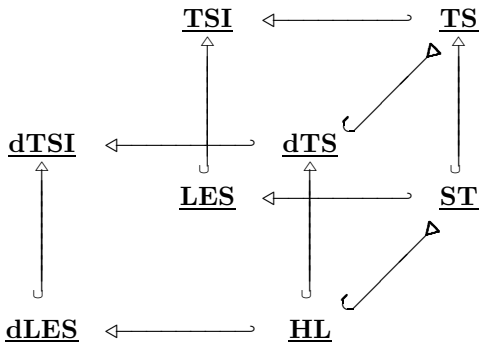
We can interpret such a result as a demonstration of the claim that transition systems with independence are a generalization of labelled event structures to a model system. However, there is a point which is worth raising. The fact that just unfolding transition systems to their occurrence version does not suffice to get a category equivalent to **LES**, shows that the *independence* relation on transitions is *not* exactly a *concurrency* relation. As an intuitive explanation of this phenomenon, it is very easy to think of a transition system with independence in which independent transitions never occur in the same path, i.e., intuitively, they are in conflict. In the light of such observation, condition (E) can be seen exactly as the condition which guarantees that independence *is* concurrency. It is then that the simple unfolding of transition systems with independence yields the category $\mathbf{oTSI_E}$ equivalent to **LES**.

Next, we briefly see that the coreflection **LES** $\hookrightarrow\!\triangleright$ **oTSI** cuts down to a coreflection **dLES** $\hookrightarrow\!\triangleright$ **dTSI**,which composes with the coreflection given earlier in this section to give a coreflection **dLES** $\hookrightarrow\!\triangleright$ **dTSI**. As a consequence, we have that **dLES** $\cong$ $\mathbf{doTSI_E}$, which can be added to results of Section 4. These results are shown by the following proposition.

PROPOSITION 5.16
*If ES is deterministic, then les.otsi(ES) is deterministic. If OTI is deterministic, then otsi.les(OTI) is deterministic.*

Thus, we arrive to the following.

THEOREM 5.17 *(Moving along the "behaviour/system" axis)*

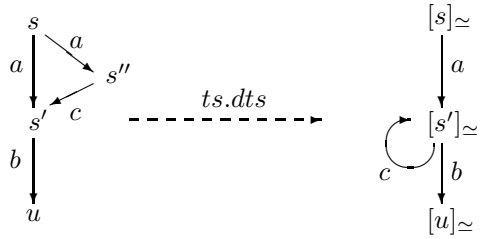



# 6 Deterministic Transition Systems with Independence

Now, we consider the relationship between **dTSI** and **TSI**, looking for a generalization of the reflection **dTS** $\hookrightarrow\!\!\lhd$ **TS**. Of course, the question to be answered is whether a left adjoint for the inclusion functor **dTSI** $\hookrightarrow$ **TSI** exists or not. This is actually a complicated issue and for a long while we could not see the answer. However, we can now answer it positively!

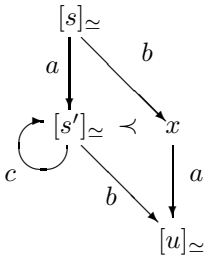
Thinking about the issue, at a first sight, one could be tempted to refine the construction given in case of transition systems by defining a suitable independence relation on the deterministic transition system obtained in that way. However, this would not work, since, in general, no independence relation yields a transition system with independence. Let us see what happens with the following example.

EXAMPLE 6.1
*Consider the transition system T in the following figure together with its deterministic version ts.dts(T).*



*Now, suppose that* $(s, a, s'')\ I\ (s', b, u)$. *Observe, that in order to establish the reflection at the level of transition systems with independence, since the unit would be a morphism from the original transition system to the deterministic one, independence must be preserved. Thus, whatever the independence relation on the deterministic transition system is, it must be* $([s]_\simeq, a, [s']_\simeq)\ I\ ([s']_\simeq, b, [u]_\simeq)$. *Then, we do not have a transition system with independence, since axiom (iii) fails.*

*However, in the rest of this section, we will show that it is always possible to "complete" the deterministic transition system obtained by ts.dts in order to make it a transition system with independence. Moreover, such a completion will be "universal", so that it will give the reflection we are seeking. In the case of the transition system above, the resulting transition system is shown below.*




Observe that it may also not be possible to define $I$ to be irreflexive. Of course, this happens when there is a diamond of concurrency whose transitions carry the same label. It is easy to understand that, in this case, the only way to cope with those transitions is by eliminating them. In other words, autoconcurrency, i.e., concurrency between events carrying the same label, add a further level of difficulty to the problem.

DEFINITION 6.2 *(Pre-Transition Systems with Independence)*
*A pre-transition system with independence is a transition system together with a binary and symmetric relation* $I$ *on its transitions.*

*A morphism of pre-transition systems with independence is a transition system morphism which, in addition, preserve the relation* $I$.

*Let* **pTSI** *denote the category of pre-transition systems with independence.*

Given sets $S$ and $L$, consider triples of the kind $(X, \equiv, I)$, where $X \subseteq S \cdot L^* = \{s\alpha \mid s \in S \text{ and } \alpha \in L^*\}$, and $\equiv$ and $I$ are binary relations on $X$. On such triples, the following closure properties can be considered.

*(Cl1)* $x \equiv z$ and $za \in X$ implies $xa \in X$ and $xa \equiv za$;
*(Cl2)* $x \equiv z$ and $za\ I\ yc$ implies $xa\ I\ yc$;
*(Cl3)* $xab \equiv xba$ and $xa\ I\ xb$ or $xa\ I\ xab$ implies $xa\ I\ yc \Leftrightarrow xba\ I\ yc$.

We say that $(X, \equiv, I)$ is *suitable* if $\equiv$ is an equivalence relation, $I$ is a symmetric relation and it enjoys properties *(Cl1)*, *(Cl2)* and *(Cl3)*. Suitable triples are meant to represent deterministic (pre) transition systems with independence, the elements in $X$ representing both states and transitions. Namely, $xa$ represents the state reached from (the state corresponding to) $x$ with an $a$-labelled transition, and that transition itself. Thus, equivalence $\equiv$ relate paths which lead to the same state and relation $I$ expresses independence of transitions. With this understanding, *(Cl1)* means that from any state there is at most one $a$-transition, while *(Cl2)* says that $I$ acts on transitions rather than on their representation. Finally, *(Cl3)*—the analogous of axiom *(iv)* of transition systems with independence—tells that transitions on the opposite edges of a diamond behave the same with respect to $I$.

For $x \in S \cdot L^*$ and $a \in L$, let $x{\restriction}a$ denote the pruning of $x$ with respect to $a$. Formally,

$$s{\restriction}a = s \qquad \text{and} \qquad (xb){\restriction}a = \begin{cases} x{\restriction}a & \text{if } a = b \\ (x{\restriction}a)b & \text{otherwise} \end{cases}$$

Of course, $(x{\restriction}a){\restriction}b = (x{\restriction}b){\restriction}a$ and thus it is possibile to use unambiguously $x{\restriction}A$ for $A \subseteq L$. Given $X \subseteq S \cdot L^*$, we use $X{\restriction}A$ to denote the set $\{x{\restriction}A \mid x \in X\}$ and for a binary relation $R$ on $X$ $R{\restriction}A$ will be $\{(x{\restriction}A, y{\restriction}A) \mid (x, y) \in R\}$.

For a transition system with independence $TI = (S, s^I, L, \mathit{Tran}, I)$, we define the sequence a triples $(S_i, \equiv_i, I_i)$, for $i \in \omega$, inductively as follows. For $i = 0$, $(S_0, \equiv_0, I_0)$ is the least (with respect to componentwise set inclusion) *suitable* triple such that

$$S \cup \Big\{ sa \;\Big|\; (s, a, u) \in \mathit{Tran} \Big\} \subseteq S_0; \quad \Big\{ (sa, u) \;\Big|\; (s, a, u) \in \mathit{Tran} \Big\} \subseteq \equiv_0;$$

and

$$\Big\{ (sa, s'b) \;\Big|\; (s, a, u)\ I\ (s', b, u') \Big\} \subseteq I_0;$$

and, for $i > 0$, $(S_i, \equiv_i, I_i)$ is the least *suitable* triple such that

*(ℑ)* $\quad S_{i-1}{\restriction}A_{i-1} \subseteq S_i; \quad \equiv_{i-1}{\restriction}A_{i-1} \subseteq \equiv_i; \quad (I_{i-1} \setminus TA_{i-1}){\restriction}A_{i-1} \subseteq I_i;$

*(D1)* $\quad xa,\ xb \in S_{i-1}{\restriction}A_{i-1}$ and $xa\ (I_{i-1} \setminus TA_{i-1}){\restriction}A_{i-1}\ xb$
implies $\quad xab,\ xba \in S_i$ and $xab \equiv_i xba$;

*(D2)* $\quad xa,\ xab \in S_{i-1}{\restriction}A_{i-1}$ and $xa\ (I_{i-1} \setminus TA_{i-1}){\restriction}A_{i-1}\ xab$
implies $\quad xb,\ xba \in S_i$ and $xab \equiv_i xba$;

where $A_i = \{a \in L \mid xa\ I_i\ xa\}$ and $TA_i = \{(xa, yb) \in I_i \mid a \in A_i \text{ or } b \in A_i\}$.

The inductive step extends a triple towards a transition system with independence by means of the rules *(D1)* and *(D2)*, whose intuitive meaning is clearly that of closing possibly incomplete diamonds. The process could create *autoindependent* transitions which must be eliminated. This is done by *(ℑ)* which drops them and adjusts $\equiv_i$ and $I_i$.

A simple inspection of the rules shows that if $a \in A_i$, then it will never appear again in the sequence. Thus, if $x$ is removed from $S_i$, it will not be reintroduced, and the same applies to the pairs in $\equiv_i$ and $I_i$. Then, it is easy to identify the *limit* of the sequence as

$$\Big(S_\omega = \bigcup_{i\in\omega}\bigcap_{j\geq i} S_j, \quad \equiv_\omega = \bigcup_{i\in\omega}\bigcap_{j\geq i} \equiv_j, \quad I_\omega = \bigcup_{i\in\omega}\bigcap_{j\geq i} I_j\Big).$$

PROPOSITION 6.3
*The triple $(S_\omega, \equiv_\omega, I_\omega)$ is suitable. Moreover, $I_\omega$ is irreflexive.*

The following proposition gives an alternative characterization of $(S_\omega, \equiv_\omega, I_\omega)$ which will be useful later on. In the following let $A_\omega$ denote $\bigcup_{i\in\omega} A_i$ and let $TA_\omega$ be $\bigcup_{i\in\omega} TA_i$.

PROPOSITION 6.4
$$(S_\omega, \equiv_\omega, I_\omega) = \Big(\bigcup_{i\in\omega}(S_i\restriction A_\omega), \quad \bigcup_{i\in\omega}(\equiv_i\restriction A_\omega), \quad \bigcup_{i\in\omega}((I_i \setminus TA_\omega)\restriction A_\omega)\Big).$$

COROLLARY 6.5
*i)* $x \in S_i$ *implies* $x\restriction A_\omega \in S_\omega$;
*ii)* $x \equiv_i y$ *implies* $(x\restriction A_\omega) \equiv_\omega (y\restriction A_\omega)$;
*iii)* $xa\ I_i\ yb$ *and* $a, b \notin A_\omega$ *implies* $(xa\restriction A_\omega)\ I_\omega\ (yb\restriction A_\omega)$.

As anticipated before, $(S_i, \equiv_i, I_i)$ encodes a deterministic pre transition system with independence which contains a deterministic version of the original $TI$ we started from (apart from the autoindependent transitions). Formally, for each $\kappa \in \omega \cup \{\omega\}$, define

$$TSys_\kappa = \Big(S_\kappa/{\equiv_\kappa}, [s^I]_{\equiv_\kappa}, L_\kappa, Tran_{\equiv_\kappa}, I_{\equiv_\kappa}\Big),$$

where

- $([x]_{\equiv_\kappa}, a, [x']_{\equiv_\kappa}) \in Tran_{\equiv_\kappa}$ if and only if $x' \equiv_\kappa xa$;
- $([x]_{\equiv_\kappa}, a, [x']_{\equiv_\kappa})\ I_{\equiv_\kappa}\ ([\bar{x}]_{\equiv_\kappa}, b, [\bar{x}']_{\equiv_\kappa})$ if and only if $xa\ I_\kappa\ \bar{x}b$;
- $L_\kappa = L \setminus \bigcup_{j<\kappa} A_j$.

Observe that the above definitions are well given. In fact, concerning $Tran_{\equiv_\kappa}$, since $xa \in S_i$ if and only if $\underline{x}a \in S_i$ for any $\underline{x} \equiv_i x$, and since $x' \equiv_i xa$ if and only if $\underline{x}' \equiv_i \underline{x}a$ for any $\underline{x} \equiv_i x$ and $\underline{x}' \equiv_i x'$, its definition is irrespective of the representive chosen. The same holds for the definition of $I_{\equiv_\kappa}$, since $xa\ I_i\ x'b$ if and only if $\underline{x}a\ I_i\ \underline{x}'b$ for any $\underline{x} \equiv_i x$ and $\underline{x}' \equiv_i x'$.

PROPOSITION 6.6
*$TSys_\kappa$ is a deterministic pre-transition system with independence.*

It is not difficult to notice the similarity of $TSys_0$ with the construction of the deterministic version of a transition system as given in Section 2. Actually, when applied to a transition system $TS$, i.e., a transition system with independence whose indepencence relation is empty, $TSys_0$ is a deterministic transition system isomorphic to $ts.dts(TS)$. This fact supports our claim that the construction we are about to give builds on $ts.dts$. However, in Section 2 a simpler construction was enough, because we did not need to manipulate transitions but only states.

PROPOSITION 6.7
*The pair $(in, id)$, where $in\colon S \to S_0/\equiv_0$ is the function which sends $s$ to its equivalence class $[s]_{\equiv_0}$ and $id$ is the identity of $L$, is a morphism of pre-transition systems with independence from $TI$ to $TSys_0$. Moreover, if $TI$ is deterministic, then $(in, id)$ is an isomorphism.*

For $i \in \omega \setminus \{0\}$, consider the pair $(in_i, id_i)$, where $in_i\colon S_{i-1}/\equiv_{i-1} \to S_i/\equiv_i$ is the function such that $in_i([x]_{\equiv_{i-1}}) = [x\restriction A_{i-1}]_{\equiv_i}$ and $id_i\colon L_{i-1} \rightharpoonup L_i$ is given by $id_i(a) = a$ if $a \notin A_{i-1}$ and $id_i{\uparrow}a$ otherwise. Then, we have the following.

LEMMA 6.8
*The pair $(in_i, id_i)\colon TSys_{i-1} \to TSys_i$ is a morphism of pre-transition systems with independence.*

It is interesting to notice that $TSys_\omega$ is a colimit in the category **pTSI**.

PROPOSITION 6.9
*$TSys_\omega$ is the colimit in **pTSI** of the $\omega$-diagram*

$$\mathcal{D} = TSys_0 \overset{(in_1,id_1)}{\longrightarrow} TSys_1 \overset{(in_2,id_2)}{\longrightarrow} \cdots \overset{(in_i,id_i)}{\longrightarrow} TSys_i \overset{(in_{i+1},id_{i+1})}{\longrightarrow} \cdots$$

*Proof.* The reader is referred to [8, chap. III, pg. 62] for the definition of the categorical concept involved.

For any $i \in \omega$, consider the function $in_i^\omega\colon S_i/\equiv_i \to S_\omega/\equiv_\omega$ such that $in_i^\omega([x]_{\equiv_i}) = [x\restriction A_\omega]_{\equiv_\omega}$ and let $id_i^\omega\colon L_i \rightharpoonup L_\omega$ denote the function such that $id_i^\omega(a) = a$ if $a \notin A_\omega$ and $id_i^\omega{\uparrow}a$ otherwise. It is easy to see that $(in_i^\omega, id_i^\omega)$ is a morphism of pre-transition systems with independence from $TSys_i$ to $TSys_\omega$.

Since for any $i$ we have $in_{i+1}^\omega \circ in_{i+1} = in_i^\omega$ and $id_{i+1}^\omega \circ id_{i+1} = id_i^\omega$, then $TSys_\omega$ and the morphisms $\{(in_i^\omega, id_i^\omega) \mid i \in \omega\}$ form a cocone in **pTSI** whose base is $\mathcal{D}$. Now, consider any cocone $\{(\sigma_i, \lambda_i)\colon TSys_i \to PT \mid i \in \omega\}$, for $PT$ any pre-transition system with independence. Then, by definition of cocone, it must be $\sigma_i = \sigma_{i+1} \circ in_{i+1}$ for each $i \in \omega$, i.e., $\sigma_i([x]_{\equiv_i}) = \sigma_{i+1}([x\restriction A_i]_{\equiv_{i+1}})$, whence it follows easily that for any $x \in S_i$ and $y \in S_j$ such that $x\restriction A_\omega = y\restriction A_\omega$ it must be $\sigma_i([x]_{\equiv_i}) = \sigma_j([y]_{\equiv_j})$. Moreover, again by definition of cocone, it must be

$\lambda_i = \lambda_{i+1} \circ id_{i+1}$. This implies that for $a \in L \setminus A_\omega$ we have $\lambda_i(a) = \lambda_{i+1}(a)$ for any $i \in \omega$, while for $a \in A_j$ it must be $\lambda_i \restriction a$ for any $i \leq j$. In fact, if $a \notin A_\omega$, since $id_{i+1}(a) = a$, it must be $\lambda_i(a) = \lambda_{i+1}(a)$. Suppose instead that $a \in A_j$. Then, $id_{j+1} \restriction a$ and thus $\lambda_j \restriction a$. Now, since $id_i(a) = a$ if $i \leq j$, it follows that $\lambda_i \restriction a$ for any $i \leq j$.

Now, define $(\bar{\sigma}, \bar{\lambda})$: $TSys_\omega \to PT$, where $\bar{\sigma}([x]_{\equiv_\omega}) = \sigma_i([\bar{x}]_{\equiv_i})$ for any $i$ and $\bar{x} \in S_i$ such that $\bar{x} \restriction A_\omega = x$, and take $\bar{\lambda}$ to be the restriction of $\lambda_0$ to $L_w$. Exploiting the features of the morphisms $(\sigma_i, \lambda_i)$, it is easy to see that $(\sigma_i, \lambda_i) = (\bar{\sigma}, \bar{\lambda}) \circ (in_i^\omega, id_i^\omega)$ for each $i$, and that $(\bar{\sigma}, \bar{\lambda})$ is the unique morphism which enjoys this property. Observe that, in view of Proposition 6.4, $\bar{\sigma}$ could be equivalently defined by saying that $\bar{\sigma}([x]_{\equiv_\omega}) = \sigma_i([x]_{\equiv_i})$ for any $x$ such that $x \in S_i$. ✓

Besides enjoying a (co)universal property, $TSys_\omega$ has another property which the reader would have already guessed by now: it is actually a deterministic transition system with independence.

PROPOSITION 6.10
*$TSys_\omega$ is a deterministic transition system with independence.*

*Proof.* Proposition 6.6 shows that $Tsys_\omega$ is a deterministic pre-transition system with independence, while it follows immediately from Proposition 6.3 that $I_{\equiv_\omega}$ is irreflexive. Let us check that the axioms of transition system with independence hold.

*i)* Trivial, since $TSys_\omega$ is deterministic.

*ii)* Suppose that $([x]_{\equiv_\omega}, a, [x']_{\equiv_\omega})\ I_{\equiv_\omega}\ ([x]_{\equiv_\omega}, b, [x'']_{\equiv_\omega})$. Then, $xa\ I_\omega\ xb$ and, therefore, there exists an index $i$ such that $xa\ I_{i-1}\ xb$, which, in turn, implies that there exist $xab \equiv_i xba \in S_i$. Then, by *(Cl3)*, $xa\ I_i\ xb$ implies $xba\ I_i\ xb$ and $xb\ I_i\ xa$ implies $xab\ I_i\ xa$. Since $a, b \notin A_\omega$ and $x \restriction A_\omega = x$, then it is $xab \equiv_\omega xba$, and $xa\ I_\omega\ xab$ and $xb\ I_\omega\ xba$, which implies that there exists $[xab]_{\equiv_\omega} = [u]_{\equiv_\omega} = [xba]_{\equiv_\omega}$ in $S_\omega/{\equiv_\omega}$ such that $([x]_{\equiv_\omega}, a, [x']_{\equiv_\omega})\ I_{\equiv_\omega}\ ([x']_{\equiv_\omega}, b, [u]_{\equiv_\omega})$, and $([x]_{\equiv_\omega}, b, [x'']_{\equiv_\omega})\ I_{\equiv_\omega}\ ([x'']_{\equiv_\omega}, a, [u]_{\equiv_\omega})$.

*iii)* Similar to the previous point.

*iv)* It is enough to show that

$$([x]_{\equiv_\omega}, a, [x']_{\equiv_\omega})\ (\prec \cup \succ)\ ([x'']_{\equiv_\omega}, a, [u]_{\equiv_\omega})\ I_{\equiv_\omega}\ ([\bar{x}]_{\equiv_\omega}, b, [\bar{x}']_{\equiv_\omega})$$
$$\text{implies } ([x]_{\equiv_\omega}, a, [x']_{\equiv_\omega})\ I_{\equiv_\omega}\ ([\bar{x}]_{\equiv_\omega}, b, [\bar{x}']_{\equiv_\omega}).$$

Suppose that the '$\prec$' case holds. Then, there exists an index $i$ such that $x' \equiv_i xa$, $x'' \equiv_i xb$, $xa\ I_i\ xb$, $xab \equiv_i u \equiv_i xba$, and $xba\ I_i\ \bar{x}b$. Then, by *(Cl3)*, we have $xa\ I_i\ \bar{x}b$. Then, it is $xa\ I_\omega\ \bar{x}b$, whence it follows that $([x]_{\equiv_\omega}, a, [x']_{\equiv_\omega})\ I_{\equiv_\omega}\ ([\bar{x}]_{\equiv_\omega}, b, [\bar{x}']_{\equiv_\omega})$.

A similar proof shows the case in which '$\succ$' holds. ✓

Thus, $TSys_\omega$ is the deterministic transition system with independence we will associate to the transition system with independence $TI$. Formally, define the map *dtsi* from the objects of **TSI** to the ojects of **dTSI** as $dtsi(TI) = TSys_\omega$. Figure 3 exemplifies the construction in an easy, yet interesting, case.

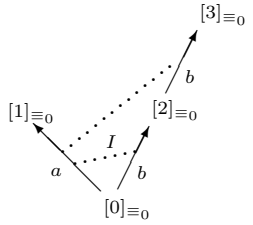
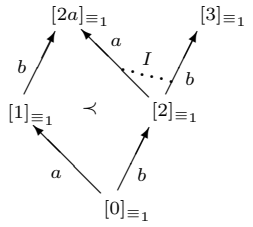
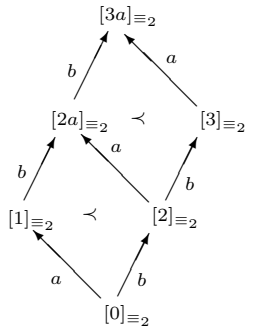
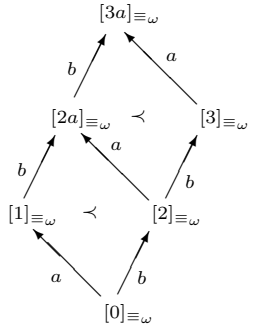
| | $TSys_\kappa$ | $\equiv_\kappa$ | $I_\kappa$ |
|---|---|---|---|
| $\kappa = 0$ |  | $[1]_{\equiv_0} = \{1, 0a\}$<br>$[2]_{\equiv_0} = \{2, 0b\}$<br>$[3]_{\equiv_0} = \{3, 2b, 0bb\}$ | $[0a]_{\equiv_0}\ I_{\equiv_0}\ [0b]_{\equiv_0}$<br>$[0a]_{\equiv_0}\ I_{\equiv_0}\ [2b]_{\equiv_0}$ |
| $\kappa = 1$ |  | $[1]_{\equiv_1} = \{1, 0a\}$<br>$[2]_{\equiv_1} = \{2, 0b\}$<br>$[3]_{\equiv_1} = \{3, 2b, 0bb\}$<br>$[2a]_{\equiv_1} = \{2a, 1b, 0ab, 0ba\}$ | $[0a]_{\equiv_1}\ I_{\equiv_1}\ [0b]_{\equiv_1}$<br>$[0a]_{\equiv_1}\ I_{\equiv_1}$<br>$[0ab]_{\equiv_1}$<br>$[0b]_{\equiv_1}\ I_{\equiv_1}\ [0ba]_{\equiv_1}$<br>$[1b]_{\equiv_1}\ I_{\equiv_1}\ [2a]_{\equiv_1}$<br>$[0a]_{\equiv_1}\ I_{\equiv_1}\ [2b]_{\equiv_1}$<br>$[0ba]_{\equiv_1}\ I_{\equiv_1}\ [0bb]_{\equiv_1}$ |
| $\kappa = 2$ |  | $[0]_{\equiv_2} = \{0\}$<br>$[1]_{\equiv_2} = \{1, 0b\}$<br>$[2]_{\equiv_2} = \{2, 0a\}$<br>$[3]_{\equiv_2} = \{3, 2a, 0aa\}$<br>$[2b]_{\equiv_2} = \{1a, 2b, 0ba, 0ab\}$<br>$[3b]_{\equiv_2} = \{3b, 2ab, 2ba, 1aa,$<br>$0abb, 0baa, 0aba\}$ | $[0a]_{\equiv_2}\ I_{\equiv_2}\ [0b]_{\equiv_2}$<br>$[0a]_{\equiv_2}\ I_{\equiv_2}\ [1b]_{\equiv_2}$<br>$[0b]_{\equiv_2}\ I_{\equiv_2}\ [2a]_{\equiv_2}$<br>$[1b]_{\equiv_2}\ I_{\equiv_2}\ [2a]_{\equiv_2}$<br>$[2a]_{\equiv_2}\ I_{\equiv_2}\ [2b]_{\equiv_2}$<br>$[2a]_{\equiv_2}\ I_{\equiv_2}\ [2bb]_{\equiv_2}$<br>$[2b]_{\equiv_2}\ I_{\equiv_2}\ [3a]_{\equiv_2}$<br>$[2bb]_{\equiv_2}\ I_{\equiv_2}\ [3a]_{\equiv_2}$ |
| $\kappa = \omega$ |  | COMMENTS. The transition system we start from gets us to $TSys_0$, where the dotted lines indicate relation $I$. $TSys_0$ fails to a be transition system with independence because there is no diamond for the transitions sticking out $[0]_{\equiv_0}$. In $TSys_1$, this problem has been solved by use of *(D1)*. However, now there is no diamond for the transitions leaving from $[2]_{\equiv_1}$, which are independent because of the closure *(Cl3)*. The problem is fixed in $TSys_2$ which is a transition system with independence and coincides with $TSys_\omega$. | |

Figure 3: An example of the construction of $TSys_\omega$.

Consider $TI = (S, s^I, L, Tran, I)$ and $TI' = (S', s'^I, L', Tran', I')$ together with a morphism $(\sigma, \lambda)$: $TI \to TI'$ in **TSI**. In the following let $(S_\kappa, \equiv_\kappa, I_\kappa)$ and $(S'_\kappa, \equiv'_\kappa, I'_\kappa)$, $\kappa \in \omega \cup \{\omega\}$, be the sequences of suitable triples corresponding, respectively, to $TI$ and $TI'$. Moreover, we shall write $A_\kappa$, $TA_\kappa$, $L_\kappa$, $TSys_\kappa$, $A'_\kappa$, $TA'_\kappa$, $L'_\kappa$ and $TSys'_\kappa$ to denote the sets and the transition systems determined respectively by $(S_\kappa, \equiv_\kappa, I_\kappa)$ and $(S'_\kappa, \equiv'_\kappa, I'_\kappa)$. We shall construct a sequence of morphisms $(\bar\sigma_i, \lambda_i)$: $TSys_i \to TSys'_i$, which will determine a morphism $(\bar\sigma_\omega, \lambda_\omega)$: $TSys_\omega \to TSys'_\omega$, i.e., $dtsi\Big((\sigma, \lambda)\Big)$.

For $i \in \omega$, let $\sigma_i$ be the function such that

$$\sigma_i(x) \;=\; \sigma(x) \quad \text{for } x \in S;$$

and

$$\sigma_i(xa) \;=\; \begin{cases} \sigma_i(x)\lambda_i(a) & \text{if } \lambda_i{\downarrow}a \\ \sigma_i(x) & \text{otherwise;} \end{cases}$$

where

$$\lambda_i(a) \;=\; \begin{cases} \lambda(a) & \text{if } \lambda(a) \notin \bigcup_{j<i} A'_j \\ \uparrow & \text{otherwise.} \end{cases}$$

LEMMA 6.11
*For any $i \in \omega$, we have that*

*i) $x \in S_i$ implies $\sigma_i(x) \in S'_i$;*

*ii) $x \equiv_i y$ implies $\sigma_i(x) \equiv'_i \sigma_i(y)$;*

*iii) $xa\ I_i\ yb$ and $\lambda_i{\downarrow}a$, $\lambda_i{\downarrow}b$ implies $\sigma_i(xa)\ I'_i\ \sigma_i(yb)$.*

It follows immediately from Lemma 6.11 that for $i \in \omega$, $\bar\sigma_i$, defined to be the map which sends $[x]_{\equiv_i}$ to $[\sigma_i(x)]_{\equiv'_i}$ is a well-defined function from $S_i/{\equiv_i}$ to $S'_i/{\equiv'_i}$. Then, it follows easily that, for $i \in \omega$, the map $(\bar\sigma_i, \lambda_i)$: $TSys_i \to TSys'_i$ is a morphism of pre-transition systems with independence.

For any $i \in \omega$, consider the morphism of pre-transition systems with independence $(in'^\omega_i, id'^\omega_i) \circ (\bar\sigma_i, \lambda_i)$: $TSys_i \to TSys'_\omega$. It is easy to see that $in'^\omega_i \circ \bar\sigma_i = in'^\omega_{i+1} \circ \bar\sigma_{i+1} \circ in_{i+1}$ for any $i \in \omega$. Moreover, since $a \in A_i$ implies $\lambda(a) \in A'_i$, we have that $id'^\omega_i \circ \lambda_i = id'^\omega_{i+1} \circ \lambda_{i+1} \circ id_{i+1}$ for any $i \in \omega$. Thus, we have that

$$\Big\{ (in'^\omega_i, id'^\omega_i) \circ (\bar\sigma_i, \lambda_i)\colon TSys_i \to TSys'_\omega \;\Big|\; i \in \omega \Big\}$$

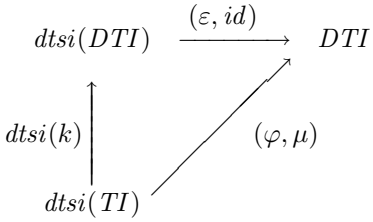
is a cocone for the $\boldsymbol{\omega}$-diagram $\mathcal{D}$ given in Proposition 6.9. Then, there exists a unique $(\bar\sigma_\omega, \lambda_\omega)$: $TSys_\omega \to TSys'_\omega$ induced by the colimit construction, which is the morphism of transition systems with independence we associate to $(\sigma, \lambda)$, i.e., $dtsi\Big((\sigma, \lambda)\Big) = (\bar\sigma_\omega, \lambda_\omega)$. The following proposition follows easily from the universal properties of colimits.

PROPOSITION 6.12 *(dtsi: **TSI** $\to$ **dTSI** is a functor)*
*The map dtsi is a functor from **TSI** to **dTSI**.*

When we apply *dtsi* to a deterministic transition system with independence $DTI$, the inductive construction of $TSys_\omega$ gives a transition system which is isomorphic to $DTI$. More precisely, each $\equiv_\omega$-equivalence class of $(S_{DTI})_\omega$ contains exactly *one* state of the original transition system, and the transition system with independence morphism $(in_0^\omega \circ in, id_0^\omega)\colon DTI \to dtsi(DTI)$—whose transition component sends $s \in S_{DTI}$ to $[s]_{\equiv_\omega}$—is actually an isomorphism. Moreover, we shall see that its inverse $(\varepsilon, id)$, where $\varepsilon([x]_{\equiv_\omega})$ is the unique $s \in S_{DTI}$ such that $s \equiv_\omega x$, is the counit of the adjunction.

LEMMA 6.13
*Let $DTI$ be a deterministic transition system with independence and consider a morphism $(\sigma, \lambda)\colon TI \to DTI$ in* **TSI**. *Let $TSys_\kappa$, $\kappa \in \omega \cup \{\omega\}$ be the sequence of pre-transition systems with independence associated to $TI$. Consider $a \in L_{TI}$ and suppose that $a \in A_i$. Then $\lambda{\uparrow}a$.*

PROPOSITION 6.14 *($(\varepsilon, id)\colon dtsi(DTI) \to DTI$ is couniversal)*
*For any transition system with independence $TI$, deterministic transition system with independence $DTI$ and morphism $(\varphi, \mu)\colon dtsi(TI) \to DTI$, there exists a unique $k\colon TI \to DTI$ such that $(\varepsilon, id) \circ dtsi(k) = (\varphi, \mu)$.*

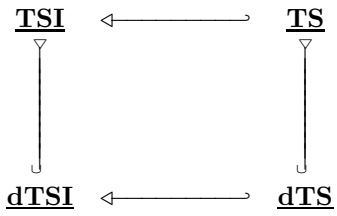



*Proof.* Let us consider $k = (\sigma, \lambda)$, where $\sigma(s) = \varphi([s]_{\equiv_\omega})$ and $\lambda$ is the function which coincides with $\mu$ on $(L_{TI})_\omega$ and is undefined elsewhere. Observe that this is the only possible choice for $k$. In fact, any $k'\colon TI \to DTI$ which has to make the diagram commute must be of the kind $(\sigma', \lambda')$ with $\lambda'(a) = \mu(a) = \lambda(a)$ for $a \in (L_{TI})_\omega$. Moreover, by Lemma 6.13, if $a \in A_\omega$, it must be $\lambda'{\uparrow}a$, i.e., $\lambda' = \lambda$. Furthermore, $\sigma'(s)$ must be an $\bar{s}$ in $S_{DTI}$ such that $\varepsilon([\bar{s}]_{\equiv_\omega}) = \bar{s}$ coincides with $\varphi([s]_{\equiv_\omega})$, i.e., $\sigma'$ is the $\sigma$ we have chosen.

In order to show that $(\sigma, \lambda)$ is a morphism of pre-transition systems with independence, it is enough to observe that $(\sigma, \lambda)$ can be expressed as the composition of the transition system with independence morphisms $(\varphi, \mu) \circ (in_0^\omega \circ in, id_0^\omega)\colon TI \to dtsi(TI) \to DTI$. This makes easy to conclude the proof. ✓

THEOREM 6.15 *($dtsi \dashv \hookrightarrow$)*
*Functor dtsi is left adjoint to the inclusion functor* **dTSI** $\hookrightarrow$ **TSI**. *Therefore, the adjunction $\langle dtsi, \hookrightarrow\rangle\colon$* **dTSI** $\rightharpoonup$ **TSI** *is a reflection.*

The adjunction $\underline{\mathbf{dTSI}} \hookleftarrow \underline{\mathbf{TSI}}$ that we have so established closes another face of the cube. In particular, we have obtained the following square, which matches the one presented in Section 2.




# 7 Deterministic Labelled Event Structures

In this section we prove that there exists a reflection from the category of deterministic labelled event structures to labelled event structures. A reflection $\underline{\mathbf{dLES}} \hookleftarrow \underline{\mathbf{LES}}$ does exist, for it follows from the reflections we have presented in the previous sections. In fact, the results in Section 5 and 6 show that there exist adjunctions

$$\underline{\mathbf{dLES}} \hookrightarrow \underline{\mathbf{dTSI}} \hookleftarrow \underline{\mathbf{TSI}} \hookleftarrow \underline{\mathbf{LES}}.$$

Now, in order to show that there is a coreflection from $\underline{\mathbf{dLES}}$ to $\underline{\mathbf{LES}}$, since $\underline{\mathbf{dLES}} \cong \underline{\mathbf{doTSI_E}}$ and $\underline{\mathbf{LES}} \cong \underline{\mathbf{oTSI_E}}$, it is enough to show that $\underline{\mathbf{dTSI}} \hookleftarrow \underline{\mathbf{TSI}}$ cuts down to a reflection $\underline{\mathbf{doTSI_E}} \hookleftarrow \underline{\mathbf{oTSI_E}}$. In this case, we would have an adjunction

$$\underline{\mathbf{dLES}} \cong \underline{\mathbf{doTSI_E}} \hookleftarrow \underline{\mathbf{oTSI_E}} \cong \underline{\mathbf{LES}},$$

whose right adjoint is isomorphic to the inclusion functor $\underline{\mathbf{dLES}} \hookrightarrow \underline{\mathbf{LES}}$. As usual, to establish that $\underline{\mathbf{doTSI_E}} \hookleftarrow \underline{\mathbf{oTSI_E}}$, it is enough to show that if $OTI$ in $\underline{\mathbf{oTSI}}$ satisfies axiom (E), then $dtsi(OTI)$ is a deterministic occurrence transition system with independence which satisfies (E).

However, since this task is rather boring, we prefer to introduce the reflection $\underline{\mathbf{dLES}} \hookleftarrow \underline{\mathbf{LES}}$ as a construction given directly on labelled event structures. In order to simplify the exposition, we factorize $\underline{\mathbf{dLES}} \hookleftarrow \underline{\mathbf{LES}}$ in two parts: $\underline{\mathbf{dLES}} \hookleftarrow \underline{\mathbf{LES_I}} \hookleftarrow \underline{\mathbf{LES}}$, where $\underline{\mathbf{LES_I}}$ is the category of labelled event structures without autoconcurrency, i.e., those labelled event structures in which all the concurrent events carry distinct labels.

### Labelled Event Structures without Autoconcurrency

As already remarked in Section 6, the only way to cope with *autoconcurrent* events is by eliminating them. However, the reader will notice that the task is now much easier than in the case of transition systems with independence. Once again, this is due to the difference between independence and concurrency and

it gives a "measure" of how this difference can play when dealing with transition systems with independence.

Let $ES = (E, \#, \leq, \ell, L)$ be a labelled event structure. Consider the sets $A(ES) = \{a \in L \mid \exists e, e' \in E, e \; co \; e' \text{ and } \ell(e) = a = \ell(e')\}$ and $TA(ES) = \{e \in E \mid \ell(e) \in A(ES)\}$. Then define

$$lesi(ES) = \Big(\bar{E},\, \# \cap (\bar{E} \times \bar{E}),\, \leq \cap (\bar{E} \times \bar{E}),\, \bar{\ell},\, \bar{L}\Big)$$

where $\bar{E} = E \setminus TA(ES)$, $\bar{L} = L \setminus A(ES)$ and $\bar{\ell}\colon \bar{E} \to \bar{L}$ is $\ell$ restricted to $\bar{E}$.

Of course $lesi(ES)$ is a labelled event structure without autoconcurrency. As a candidate for the unit of the adjunction, consider the map $(\bar{in}, \bar{id})\colon ES \to lesi(ES)$ where

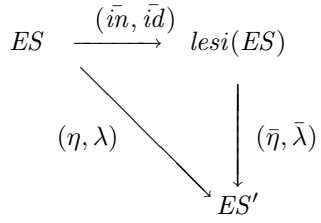
$$\bar{in}(e) = \begin{cases} e & \text{if } e \in \bar{E} \\ \uparrow & \text{otherwise;} \end{cases} \qquad \text{and} \qquad \bar{id}(a) = \begin{cases} a & \text{if } a \in \bar{L} \\ \uparrow & \text{otherwise} \end{cases}$$

It is extremely easy to verify that this definition gives a morphism in $\underline{\mathbf{LES}}$.

Lemma 7.1
*Let $(\eta, \lambda)\colon ES \to ES'$ be a morphism of labelled event structures and suppose that $ES'$ has no autoconcurrency. Then, $\eta\!\uparrow\! e$ for any $e \in TA(ES)$.*

It is now easy to show that *lesi* extends to a functor from $\underline{\mathbf{LES}}$ to $\underline{\mathbf{LES_I}}$ which is left adjoint to the inclusion $\underline{\mathbf{LES_I}} \hookrightarrow \underline{\mathbf{LES}}$.

Proposition 7.2 *($(\bar{in}, \bar{id})\colon ES \to lesi(ES)$ is universal)*
*For any labelled event structure $ES$, any labelled event structure without autoconcurrency $ES'$ and any morphism $(\eta, \lambda)\colon ES \to ES'$, there exists a unique $(\bar{\eta}, \bar{\lambda})\colon lesi(ES) \to ES'$ in $\underline{\mathbf{LES_I}}$ such that $(\bar{\eta}, \bar{\lambda}) \circ (\bar{in}, \bar{id}) = (\eta, \lambda)$.*




*Proof.* Consider $\bar{\eta}\colon \bar{E}_{ES} \to E_{ES'}$ and $\bar{\lambda}\colon \bar{L}_{ES} \to L_{ES'}$ to be, respectively, $\eta$ restricted to $\bar{E}_{ES}$ and $\lambda$ restricted to $\bar{L}_{ES}$. Exploiting Lemma 7.1 it is easy to conclude. ✓

Therefore, we have the following.

Theorem 7.3 *($lesi \dashv \hookleftarrow$)*
*The map lesi extends to a functor from $\underline{\mathbf{LES}}$ to $\underline{\mathbf{LES_I}}$ which is left adjoint to $\underline{\mathbf{LES_I}} \hookrightarrow \underline{\mathbf{LES}}$. Thus, the adjunction $\langle lesi, \hookleftarrow\rangle\colon \underline{\mathbf{LES}} \rightharpoonup \underline{\mathbf{LES_I}}$ is a reflection.*

Let us now turn our attention to $\underline{\mathbf{dLES}} \hookrightarrow \underline{\overline{\mathbf{LES_I}}}$. Given a labelled event structure without autoconcurrency $ES = (E, \leq, \#, \ell, L)$, consider the sequence of relations $(\sim_i, \leq_i, \#_i)$, for $i \in \omega$, where

- $\sim_0 = \left\{(e,e) \;\middle|\; e \in E\right\}; \quad \leq_0 = \leq; \quad \#_0 = \#;$

and, for $i > 0$,

- $\sim_i$ is the least equivalence on $E$ such that

  *i)* $\sim_{i-1} \subseteq \sim_i$;

  *ii)* $e \not\leq_{i-1} e',\ e' \not\leq_{i-1} e,\ \ell(e) = \ell(e')$
  $\lfloor e \rfloor_{\leq_{i-1}} \not\#_{i-1} \lfloor e' \rfloor_{\leq_{i-1}} \setminus \{e'\}$ and
  $\lfloor e \rfloor_{\leq_{i-1}} \not\#_{i-1} \lfloor e' \rfloor_{\leq_{i-1}} \setminus \{e'\}$ implies $e \sim_i e'$,

  where $\lfloor e \rfloor_{\leq_i}$ is a shorthand for $\{e' \in E \mid e' \leq_i e\}$ and, for $x, y \subseteq E$, $x \not\#_i y$ stands for $\forall\, e \in x,\ \forall\, e' \in y,$ not $e \#_i e'$.

- $e \leq_i e'$ if and only if $\forall\, \bar{e}' \sim_i e'\ \exists\, \bar{e} \sim_i e.\ \bar{e} \leq_{i-1} \bar{e}'$;

- $e \#_i e'$ if and only if $\forall \bar{e}' \sim_i e'\ \forall \bar{e} \sim_i e.\ e \#_{i-1} e'$;

Observe from the previous definitions that, while $\sim_i \subseteq \sim_{i+1}$ and $\#_i \supseteq \#_{i+1}$, it is $\leq_i \not\subseteq \leq_{i+1}$ and $\leq_i \not\supseteq \leq_{i+1}$. Each triple $(\sim_i, \leq_i, \#_i)$ represents a *quotient* of the original labelled event structure in which—informally speaking—the *"degree"* of non-determinism has decreased. This will be delevoped in the following.

Lemma 7.4
*For any $i \in \omega$,*

*i) $\leq_i$ is a preorder such that $e \leq_i e'$ and $e' \leq_i e$ if and only if $e \sim_i e'$;*

*ii) $\#_i$ is a symmetric, irreflexive relation and $e \#_i e' \leq_i e''$ implies $e \#_i e''$;*

*iii) for any $e \in E$, the set $\left\{[e']_{\sim_i} \;\middle|\; e' \leq_i e,\ e' \in E\right\}$ is finite.*

Lemma 7.5
*Let $(\sim_i, \leq_i, \#_i)$, $i \in \omega$, be the sequence constructed from a labelled event structure without autoconcurrency $ES$ as given above. Then*

*i) for any $i \in \omega$ and for any $j \leq i$, $e \leq_i e' \Leftrightarrow \forall\, \bar{e}' \sim_i e'\ \exists\, \bar{e} \sim_i e.\ \bar{e} \leq_j \bar{e}'$;*

*ii) for any $i \in \omega$ and for any $j \leq i$, $e \#_i e' \Leftrightarrow \forall\, \bar{e} \sim_i e\ \forall\, \bar{e}' \sim_i e'.\ \bar{e} \#_j \bar{e}'$.*

The next lemma shows that, although neither $\leq_i \subseteq \leq_{i+1}$ nor $\leq_i \supseteq \leq_{i+1}$, the "behaviour" of the sequence of preorders $\leq_i$, is not so bad as it could seem.

LEMMA 7.6
*If $e \leq_j e'$ and $e \not\leq_{j+1} e'$, then $\forall\, i > j.\ e \not\leq_i e'$.*

Now, consider the triple of relations $(\sim_\omega, \leq_\omega, \#_\omega)$, where

$$\sim_\omega = \bigcup_{i\in\omega} \sim_i, \qquad \leq_\omega = \bigcup_{i\in\omega} \bigcap_{j>i} \leq_i, \qquad \#_\omega = \bigcap_{i\in\omega} \#_i$$

or, equivalenty, $\leq_\omega$ is defined by

$$e \leq_\omega e' \quad \text{if and only if} \quad \exists\, k\ \forall\, i > k \quad e \leq_i e'.$$

Thanks to Lemma 7.6, it is immediate to show that $\leq_\omega$ enjoys the following relevant property.

LEMMA 7.7
*$e \not\leq_\omega e'$ if and only if $\exists\, k.\ \forall\, i > k \quad e \not\leq_i e'$.*

The following characterization of $\leq_\omega$ derives (not trivially) from Lemma 7.5 and Lemma 7.7, and helps in showing the equivalent of Lemma 7.4 for the triple $(\sim_\omega, \leq_\omega, \#_\omega)$.

LEMMA 7.8
*For any $j \in \omega$, $e \leq_\omega e'$ if and only if $\forall\, \bar{e}' \sim_\omega e'\ \exists\, \bar{e} \sim_\omega e.\ \bar{e} \leq_j \bar{e}'$.*

LEMMA 7.9
*i) $\leq_\omega$ is a preorder such that $e \leq_\omega e'$ and $e' \leq_\omega e$ if and only if $e \sim_\omega e'$.*
*ii) $\#_\omega$ is symmetric, irreflexive and $e \#_\omega e' \leq_\omega e''$ implies $e \#_\omega e''$.*
*iii) For any $e \in E$, the set $\left\{ [e']_{\sim_\omega} \;\middle|\; e' \leq_\omega e,\ e' \in E \right\}$ is finite.*

It follows immediately that, for any $\kappa \in \omega \cup \{\omega\}$,

$$Ev_\kappa = (E/{\sim_\kappa}, \leq_{\sim_\kappa}, \#_{\sim_\kappa}, \ell_{\sim_\kappa}, L),$$

where

- $E/{\sim_\kappa}$ is the set of $\sim_\kappa$-classes of $E$;
- $[e]_{\sim_\kappa} \leq_{\sim_\kappa} [e']_{\sim_\kappa}$ if and only if $e \leq_\kappa e'$;
- $[e]_{\sim_\kappa} \#_{\sim_\kappa} [e']_{\sim_\kappa}$ if and only if $e \#_\kappa e'$;
- $\ell_{\sim_\kappa}([e]_{\sim_\kappa}) = \ell(e)$;

is a labelled event structure. Observe that $Ev_0$ is (isomorphic to) the labelled event structure $ES$ we started from. Using the same notation as in Section 6, we denote by $(in, id)\colon ES \to Ev_0$ the isomorphism which sends $e$ to $[e]_{\sim_0}$.

The interesting fact about the labelled event structures $Ev_i$ is that they have no autoconcurrency. This fact plays a crucial role in establishing the adjunction we are seeking.

LEMMA 7.10
*For any $i \in \omega$, $Ev_i$ is in $\underline{\mathbf{LES_I}}$.*

Similarly to the case of the sequence $TSys_i$, $i \in \omega$, presented in Section 6, event structures $Ev_i$ are related to each other by *inclusion* morphism. For $i \in \omega \setminus \{0\}$, let $in_i \colon E/{\sim_{i-1}} \to E/{\sim_i}$ be the function such that $in_i\Big([e]_{\sim_{i-1}}\Big) = [e]_{\sim_i}$. Then we have the following.

LEMMA 7.11
*For any $i \in \omega \setminus \{0\}$, $(in_i, id) \colon Ev_{i-1} \to Ev_i$ is a labelled event structure morphism.*

Next, we shall show that $Ev_\omega$ is the colimit of the $\boldsymbol{\omega}$-diagram formed by the $Ev_i$'s. For any $i \in \omega$, consider the mapping $in_i^\omega \colon E/{\sim_i} \to E/{\sim_\omega}$ which, for any $e \in E$, sends $[e]_{\sim_i}$ to $[e]_{\sim_\omega}$.

LEMMA 7.12
*For any $i \in \omega$, $(in_i^\omega, id) \colon Ev_i \to Ev_\omega$ is a labelled event structure morphism*

PROPOSITION 7.13
*$Ev_\omega$ is the colimit in $\underline{\mathbf{LES_I}}$ of the $\boldsymbol{\omega}$-diagram*

$$\mathcal{D} = Ev_0 \overset{(in_1,id)}{\longrightarrow} Ev_1 \overset{(in_2,id)}{\longrightarrow} \cdots \overset{(in_{i-1},id)}{\longrightarrow} Ev_i \overset{(in_i,id)}{\longrightarrow} \cdots$$

*Proof.* Since for any $1 < j < i$ it is $in_j^\omega = in_i^\omega \circ in_i \circ \cdots \circ in_{j+1}$, then $\{(in_i^\omega, id) \colon Ev_i \to Ev_\omega \mid i \in \omega\}$ is a cocone with base $\mathcal{D}$.

Consider now any other cocone $\{(\eta_i, \lambda_i) \colon Ev_i \to ES \mid i \in \omega\}$ for $\mathcal{D}$, $ES$ being any object in $\underline{\mathbf{LES_I}}$. Since for any $i$ it is $(\eta_i, \lambda_i) = (\eta_{i+1}, \lambda_{i+1}) \circ (in_{i+1}, id)$, it must necessarily be $\lambda_i = \lambda_0 = \lambda$ and $\eta_i\Big([e]_{\sim_i}\Big) = \eta_{i+1}\Big([e]_{\sim_{i+1}}\Big)$, for any $i$. Thus, we can define $\bar{\eta} \colon E/{\sim_\omega} \to E_{ES}$ by $\bar{\eta}\Big([e]_{\sim_\omega}\Big) = \eta_0\Big([e]_{\sim_0}\Big)$. Clearly, we have that

$$\bar{\eta}\Big([e]_{\sim_\omega}\Big) = \eta_0\Big([e]_{\sim_0}\Big) = \eta_i\Big([e]_{\sim_i}\Big),$$

i.e., for any $i$, $\eta_i = \bar{\eta} \circ in_i^\omega$, and, moreover, $\bar{\eta}$ is clearly the unique mapping for which that happens. Thus, to conclude the proof, the missing step is to show that $(\bar{\eta}, \lambda) \colon Ev_\omega \to ES$ is a labelled event structure morphism. ✓

Exploiting the characterizations of $\leq_\omega$ previously given, it is not difficult to show the following.

LEMMA 7.14
*$Ev_\omega$ is deterministic.*

*Proof.* Consider a configuration $c_\omega$ of $Ev_\omega$ and two events $[e]_{\sim_\omega} \neq [e']_{\sim_\omega}$ enabled at $c_\omega$, i.e., such that $c_\omega \vdash [e]_{\sim_\omega}$ and $c_\omega \vdash [e']_{\sim_\omega}$. We shall show that there exists $i$ and a configuration $c_i$ of $Ev_i$ in which $[e]_{\sim_i}$ and $[e']_{\sim_i}$ are enabled. Since $[e]_{\sim_{i+1}} \neq [e']_{\sim_{i+1}}$, which follows from the fact that $[e]_{\sim_\omega} \neq [e']_{\sim_\omega}$, and since $[e]_{\sim_i} \not\leq_i [e']_{\sim_i}$, which follows from the fact that both are enabled at $c_i$, it must necessarily be

$$\ell_{\sim_\omega}\Big([e]_{\sim_\omega}\Big) = \ell_{\sim_i}\Big([e]_{\sim_i}\Big) \neq \ell_{\sim_i}\Big([e']_{\sim_i}\Big) = \ell_{\sim_\omega}\Big([e']_{\sim_\omega}\Big),$$

which shows that $Ev_\omega$ is deterministic.

Suppose that $[\bar{e}]_{\sim_\omega} \leq_{\sim_\omega} [e]_{\sim_\omega}$. Then, there exists $k$ such that $\forall\, j > k$ $[\bar{e}]_{\sim_j} \leq_{\sim_j} [e]_{\sim_j}$. Since the set $\Big\lfloor [e]_{\sim_\omega} \Big\rfloor_{\leq_\omega}$ is finite, we can find $\bar{k}'$ such that for any $j > \bar{k}'$ $[\bar{e}]_{\sim_\omega} \leq_{\sim_\omega} [e]_{\sim_\omega}$ implies $[\bar{e}]_{\sim_j} \leq_{\sim_j} [e]_{\sim_j}$. If otherwise $[\bar{e}]_{\sim_\omega} \not\leq_{\sim_\omega} [e]_{\sim_\omega}$ and $[\bar{e}]_{\sim_{\bar{k}'}} \leq_{\sim_{\bar{k}'}} [e]_{\sim_{\bar{k}'}}$, there exists $j > k$ such that $[\bar{e}]_{\sim_j} \not\leq_{\sim_j} [e]_{\sim_j}$. Now, observe that there can be only finitely many such $[\bar{e}]_{\sim_k}$. In fact, if it were not, exploiting Lemmas 7.5 and 7.7, it would be possible to derive a contradiction showing that $e$ has infinitely many pre-events in $Ev_0$. Then, we can find $\bar{k}''$ such that for any $j > \bar{k}''$

$$[\bar{e}]_{\sim_\omega} \leq_{\sim_\omega} [e]_{\sim_\omega} \quad \Leftrightarrow \quad [\bar{e}]_{\sim_j} \leq_{\sim_j} [e]_{\sim_j}.$$

In the same way, we can find $\bar{\bar{k}}''$ such that for any $j > \bar{\bar{k}}''$, $[\bar{e}]_{\sim_\omega} \leq_{\sim_\omega} [e']_{\sim_\omega}$ if and only if $[\bar{e}]_{\sim_j} \leq_{\sim_j} [e']_{\sim_j}$. Thus, considering $\bar{k} = \max\{\bar{k}'', \bar{\bar{k}}''\}$ we have that for any $i > \bar{k}$

$$[\bar{e}]_{\sim_\omega} \in \Big\lfloor [e]_{\sim_\omega} \Big\rfloor_{\leq_\omega} \quad \Leftrightarrow \quad [\bar{e}]_{\sim_i} \in \Big\lfloor [e]_{\sim_i} \Big\rfloor_{\leq_i}$$

and

$$[\bar{e}]_{\sim_\omega} \in \Big\lfloor [e']_{\sim_\omega} \Big\rfloor_{\leq_\omega} \quad \Leftrightarrow \quad [\bar{e}]_{\sim_i} \in \Big\lfloor [e']_{\sim_i} \Big\rfloor_{\leq_i}.$$

Now, consider $[\bar{e}]_{\sim_{\bar{k}}} \leq_{\bar{k}} [e]_{\sim_{\bar{k}}}$ and $[\bar{e}']_{\sim_{\bar{k}}} \leq_{\bar{k}} [e']_{\sim_{\bar{k}}}$. It is still possible that $[\bar{e}]_{\sim_{\bar{k}}} \#_{\sim_{\bar{k}}} [\bar{e}']_{\sim_{\bar{k}}}$. However, since $[\bar{e}]_{\sim_\omega} \not\#_{\sim_\omega} [\bar{e}']_{\sim_\omega}$, it must exist $k$ such that for any $i > k$ $[\bar{e}]_{\sim_i} \not\#_{\sim_i} [\bar{e}']_{\sim_i}$. Then, since there can be only finitely many such pairs, we can find an integer $i$ (greater than $\bar{k}$) such that the set $\Big\lfloor [e]_{\sim_i} \Big\rfloor_{\leq_i} \cup \Big\lfloor [e']_{\sim_i} \Big\rfloor_{\leq_i}$ is conflict free (wrt. to $\#_{\sim_i}$). It is immediate now to see that

$$c_i = \Big(\Big\lfloor [e]_{\sim_i} \Big\rfloor_{\leq_i} \setminus \Big\{[e]_{\sim_i}\Big\}\Big) \cup \Big(\Big\lfloor [e']_{\sim_i} \Big\rfloor_{\leq_i} \setminus \Big\{[e']_{\sim_i}\Big\}\Big)$$

is a configuration of $Ev_i$ which enables $[e]_{\sim_i}$ and $[e']_{\sim_i}$. ✓

As it was probably clear from a while, the object component of the functor $dles\colon \underline{\mathbf{LES_I}} \to \underline{\mathbf{dLES}}$ is the function which maps a labelled event structure without autoconcurrency $ES$ to the deterministic event structure $Ev_\omega$ limit of the sequence of event structures $Ev_i$ built from it.

An example of the construction is given in Figure 4, in which we also show how the construction of the deterministic transition system with independence

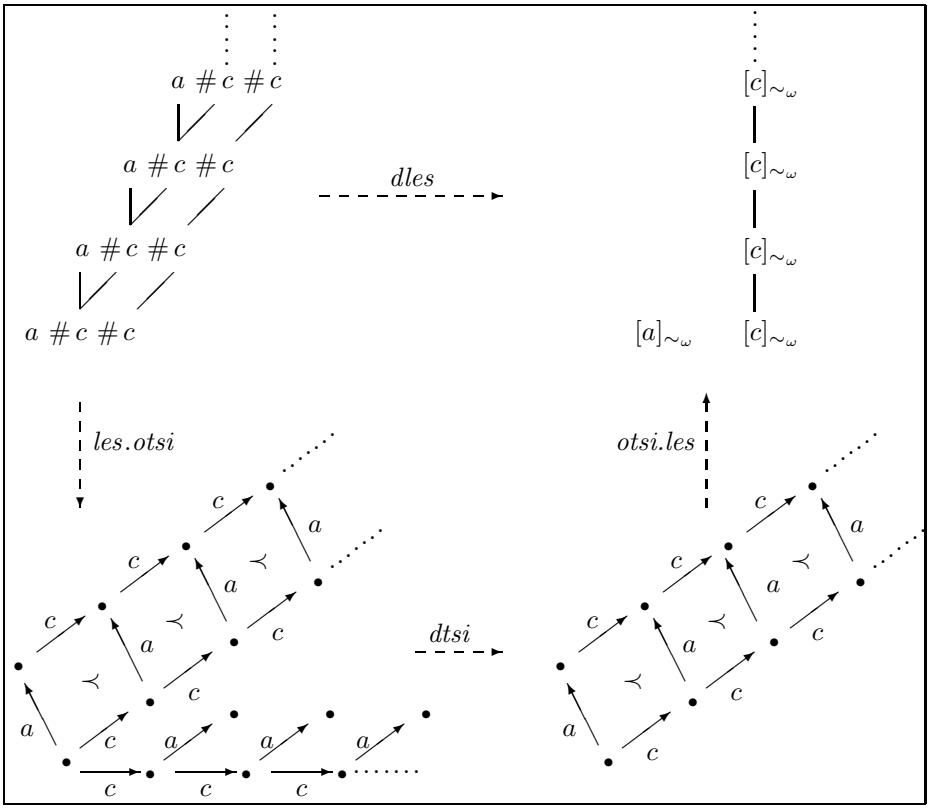
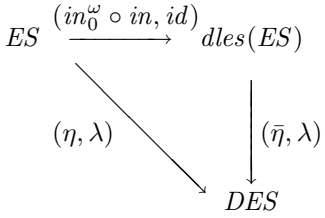



Figure 4: An event structure $ES$ and $dles(ES)$

works on the transition system of configurations of $ES$. Now, the last step we miss is to show that *dles* can be extended to a functor which is left adjoint to the inclusion functor $\underline{\mathbf{dLES}} \hookrightarrow \underline{\mathbf{LES_I}}$. This is done in the following proposition.

PROPOSITION 7.15 *($(in_0^\omega \circ in, id)\colon ES \to dles(ES)$ is universal)*
*For any labelled event structure without autoconcurrency $ES$, any deterministic labelled event structure $DES$ and any $(\eta, \lambda)\colon ES \to DES$, there exists a unique $(\bar{\eta}, \lambda)\colon dles(ES) \to DES$ in $\underline{\mathbf{dLES}}$ such that $(\bar{\eta}, \lambda) \circ (in_0^\omega \circ in, id) = (\eta, \lambda)$.*



*Proof.* Suppose for a while that we are able to show that $(\eta, \lambda)\colon ES \to DES$ gives rise to a cocone $\{(\eta_i, \lambda)\colon Ev_i \to DES \mid i \in \omega\}$ with base $\mathcal{D}$ such that $\eta = \eta_0 \circ in$, $(in, id)$ being the isomorphism of $ES$ and $Ev_0$. Then, by Proposition 7.13, there exists a unique $(\bar{\eta}, \lambda)\colon dles(ES) \to DES$ such that, for any $i$, $\eta_i = \bar{\eta} \circ in_i^\omega$. Then, in particular, $\eta_0 = \bar{\eta} \circ in_0^\omega$, and thus

$$(\eta, \lambda) = (\eta_0 \circ in, \lambda) = (\bar{\eta}, \lambda) \circ (in_0^\omega \circ in, id).$$

In other words, $\bar{\eta}\colon E_{ES}/{\sim_\omega} \to E_{DES}$ given, as in the proof of Proposition 7.13, by $\bar{\eta}\Big([e]_{\sim_\omega}\Big) = \eta(e)$ would be such that $(\bar{\eta}, \lambda)$ makes the diagram commute.

The hypothesis above is sufficient also to show the uniqueness of $(\bar{\eta}, \lambda)$. Suppose in fact that there exists $\bar{\bar{\eta}}$ such that $\eta = \bar{\bar{\eta}} \circ (in_0^\omega \circ in)$, then it is $\eta_0 = \bar{\bar{\eta}} \circ in_0^\omega$. It follows that, for any $i$,

$$\begin{array}{rcl} \eta_i\Big([e]_{\sim_i}\Big) & = & (\bar{\eta} \circ in_i^\omega)\Big([e]_{\sim_i}\Big) \;=\; (\bar{\eta} \circ in_0^\omega)\Big([e]_{\sim_0}\Big) \\ & = & (\bar{\bar{\eta}} \circ in_0^\omega)\Big([e]_{\sim_0}\Big) \;=\; (\bar{\bar{\eta}} \circ in_i^\omega)\Big([e]_{\sim_i}\Big), \end{array}$$

i.e., for any $i$ it is $\eta_i = \bar{\bar{\eta}} \circ in_i^\omega$. Then, by definition of colimit, it is $\bar{\bar{\eta}} = \bar{\eta}$.

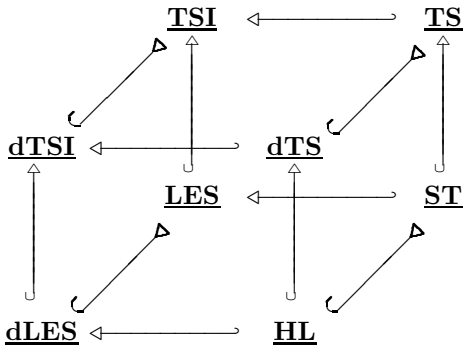
Thus, we only need to show that the cocone $\{(\eta_i, \lambda)\colon Ev_i \to DES \mid i \in \omega\}$ actually exists. However, by defining for any integer $i$ the mapping $\eta_i\colon E_{ES}/{\sim_i} \to E_{DES}$ by $\eta_i\Big([e]_{\sim_i}\Big) = \eta(e)$, we obviously have a cocone $\{(\eta_i, \lambda) \mid i \in \omega\}$ as required. So, we just need to prove that $(\eta_i, \lambda)$ is a well-defined labelled event structure morphism from $Ev_i$ to $DES$. This can be done—with some efforts—by induction on $i$. ✓

As usual, the universality of $(in_0^\omega \circ in, id)$ allows us to conclude what follows.

THEOREM 7.16 *($dles \dashv \hookleftarrow$)*
*The mapping dles extends to a functor which is left adjoint of the inclusion of* **dLES** *in* $\mathbf{LES_I}$. *Then, $\langle dles, \hookleftarrow\rangle$ is a reflection.*

The coreflection **dLES** $\hookleftarrow$ **LES** closes the last two faces of the cube. So, our results may be summed up in the following cube of relationships among models.

THEOREM 7.17 *(The Cube)*



# Conclusion

We have established a complete "cube" of formal relationships between well-known and (a few) new models for concurrency. Thus, we have a complete picture of how to translate between these models via adjunctions along the axes of "interleaving/noninterleaving", "linear/branching" and "behaviuor/system". Notice also the pleasant conformity in the picture, with *coreflections* along the "interleaving/noninterleaving" and "behaviour/system" axes, and *reflections* along "linear/branching".

It should be mentioned that not all squares (surfaces) of the "cube" commute. Of course, they do with directions along those of the embeddings.

It is worth remarking that all the adjunctions in this paper would still hold if we modified uniformly the morphisms of the involved categories by eliminating the label component. However, if we considered only total morphisms, the reflections $\underline{\mathbf{dTSI}} \hookleftarrow \underline{\mathbf{TSI}}$ and $\underline{\mathbf{dLES}} \hookleftarrow \underline{\mathbf{LES}}$ would not exist.

# References


[1] M.A. Bednarczyk. *Categories of Asynchronous Transition Systems*. PhD Thesis, University of Sussex, 1988.

[2] J. Gischer. The Equational Theory of Pomsets. *Theoretical Computer Science*, n. 61, pp. 199–224, 1988.

[3] J. Grabowski. On Partial Languages. *Fundamenta Informaticae*, n. 4, pp. 428–498, 1981.

[4] M. Hennessy. *Algebraic Theory of Processes*. Cambridge, Massachusetts, 1988.

[5] C.A.R. Hoare. *Communicating Sequential Processes*. Englewood Cliffs, 1985.

[6] P.W. Hoogers, H.C.M Kleijn, and P.S. Thiagarajan. A Trace Semantics for Petri Nets. In Proceedings of *ICALP '92*, LNCS, n. 623, pp. 595–604, Springer Verlag, 1992.

[7] R.M. Keller. Formal Verification of Parallel Programs. *Communications of the ACM*, n. 19, vol. 7, pp. 371–384, 1976.

[8] S. MacLane. *Categories for the Working Mathematician*. GTM, Springer-Verlag, 1971.

[9] A. Mazurkiewicz. Basic Notions of Trace Theory. In *lecture notes for the REX summerschool in temporal logic*, LNCS n. 354, pp. 285–363, Springer-Verlag, 1988.

[10] R. Milner. *Communication and Concurrency*. Englewood Cliffs, 1989.

[11] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, Event Structures and Domains, part 1. *Theoretical Computer Science*, n. 13, pp. 85–108, 1981.

[12] C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für Instrumentelle Mathematik, Bonn, FRG, 1962.

[13] G. Plotkin. *A Structural Approach to Operational Semantics*. Technical report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[14] V. Pratt. Modeling Concurrency with Partial Orders. *International Journal of Parallel Processing*, n. 15, pp. 33–71, 1986.

[15] V. Sassone, M. Nielsen, and G. Winskel. A Classification of Models for Concurrency. In *Proceedings of Fourth International Conference on Concurrency Theory, CONCUR '93*, LNCS n. 715, pp. 82–96, Springer-Verlag, 1993. To appear as Technical Report DAIMI, Computer Science Department, Aarhus University, 1993.

[16] V. Sassone, M. Nielsen, and G. Winskel. Deterministic Behavioural Models for Concurrency. In *Proceedings of 18th International Symposium on the Mathematical Foundations of Computer Science, MFCS '93*, LNCS n. 711, pp. 682–692, Springer-Verlag, 1993. To appear as Technical Report DAIMI, Computer Science Department, Aarhus University, 1993.

[17] M.W. Shields. Concurrent Machines. *Computer Journal*, n. 28, pp. 449–465, 1985.

[18] A. Stark. Concurrent Transition Systems. *Theoretical Computer Science*, n. 64, pp. 221–269, 1989.

[19] P. Starke. Traces and Semiwords. LNCS, n. 208, pp. 332–349, Springer-Verlag, 1985.

[20] G. Winskel. Event Structure Semantics of CCS and Related Languages. In proceedings of *ICALP '82*, LNCS n. 140, pp. 561–567, Springer-Verlag, 1982. Expanded version available as technical report DAIMI PB-159, Computer Science Department, Aarhus University.

[21] G. Winskel. Synchronization Trees. *Theoretical Computer Science*, n. 34, pp. 33–82, 1985.

[22] G. Winskel. Event Structures. In *Advances in Petri nets*, LNCS n. 255, pp. 325–392, Springer-Verlag, 1987.

[23] G. Winskel, and M. Nielsen. Models for Concurrency. To appear in the *Handbook of Logic in Computer Science*. A draft appears as technical report DAIMI PB-429, Computer Science Department, Aarhus University, 1992.