The Box Algebra = Petri Nets + Process Expressions

Eike Best[1], Raymond Devillers[2] and Maciej Koutny[3]

[1] Fachb. Inf., Carl von Ossietzky Universität, D–26111 Oldenburg, Germany
[2] Départ. d'Inform., Université Libre de Bruxelles, B-1050 Bruxelles, Belgium
[3] Dept. of Comp. Sci., University of Newcastle, Newcastle upon Tyne NE1 7RU, U.K.

Abstract. The paper describes a Petri net as well as a structural operational semantics for an algebra of process expressions. It specifically addresses this problem for the box algebra, a model of concurrent computation which combines Petri nets and standard process algebras. The main result is that it is possible to obtain a framework where process expressions can be given two, entirely consistent, kinds of semantics: one based on Petri nets, the other on SOS rules. This consistency can also be extended to a partial order semantics.
Keywords: Net-based algebraic calculi; relationships between net theory and other approaches; process algebras; box algebra; SOS semantics.

1 Introduction

This paper is about *combining* two widely known and well studied theories of concurrency: *process algebras* [1, 16, 17, 20] and *Petri nets* [2, 21, 24]. Process algebras: (i) allow the study of connectives directly related to actual programming languages; (ii) are compositional by definition; (iii) come with a variety of logics facilitating reasoning about important properties of systems; and (iv) support a variety of algebraic laws. On the other hand, Petri nets: (v) sharply distinguish between states and activities (the latter being defined as changes of state); (vi) treat global states and global activities as derived from their basic local counterparts; (vii) have a graphical representation which is easy to grasp and has therefore some wide appeal for practitioners; and (viii) have useful links both to graph theory and to linear algebra.

The work presented in this paper (in itself a continuation and consolidation of, among others, [4, 5, 18]) does not subscribe to the ambition of trying to achieve a full combination of (i)–(viii) above – at least not immediately. However, it attempts to forge links between a fundamental but restricted class of Petri nets and a basic but again restricted process algebra. This paper will investigate the structural and behavioural aspects of these two basic models, and its main point is that there is, in fact, an extremely strong equivalence between them.

The *box algebra* is based on a set of process terms, called *box expressions*. Each box expression has associated two consistent kinds of semantics: a Petri net called a *box* (with its standard Petri net transition firing rule), and an operational semantics defined using SOS derivation rules [23]. A particular instance of the box algebra is the Petri Box Calculus (PBC) [3, 4, 11, 15] – a direct inspiration for the introduction of the box algebra. Note that the technique of associating Petri nets with process algebra expressions has also been studied for other models [9, 10, 13, 14, 22, 25].

The model we are going to describe is based on a set of operators, OpBox, which can be used to construct valid box expressions. For each operator $\mathsf{op} \in \mathsf{OpBox}$ there is an associated operator in the domain of boxes, Ω_{op}. This allows one to compositionally define, for every box expression $E = \mathsf{op}(E_1, E_2, \ldots)$, a corresponding net, $\mathsf{box}(E) = \Omega_{\mathsf{op}}(\mathsf{box}(E_1), \mathsf{box}(E_2), \ldots)$, where the application of Ω_{op} to an operand tuple $(\mathsf{box}(E_1), \mathsf{box}(E_2), \ldots)$ corresponds to net theoretic refinement. The set of PBC operators includes sequence (denoted by the semicolon), choice (denoted by $\square$), and parallel composition (denoted by $\|$). However, the box algebra supports much richer a set of constructs because Ω_{op} can be chosen from a very large set of boxes.

The two semantical models of the box algebra have been studied and developed in, e.g., [3, 5, 18]. In particular, it has been shown there that the two semantics are equivalent in the sense of generating strongly equivalent behaviours (in bisimulation sense [20]). This paper extends the already published results in three directions. First, the previous results were only applicable to finite operators; i.e., it was assumed that $\mathsf{op}(K_1, K_2, \ldots) = \mathsf{op}(K_1, K_2, \ldots, K_n)$, for some n. In this paper, op can in general take any number of arguments, in particular infinitely many. The interest in such general operators is motivated by a need to model operators like $\square_{i\in I} E_i$, i.e., a fully generalised choice operator of Milner's CCS [20], which in turn can be used to give formal semantics to a process algebra with value-passing. Second, we remove two restrictions previously imposed on nets representing operators and operands for reasons of behavioural integrity; but we also analyse the conditions under which such a step does not compromise the integrity of the model. The third extension concerns consistency between the net semantics and operational semantics. The interest here is in establishing as strong as possible semantical correspondence between the two models, thus allowing a direct transfer of behavioural properties from one framework to another. We strengthen the previous results by proving that they generate, for every process expression, not only bisimulation equivalent, but in fact *isomorphic* transition systems; thus providing arguably the strongest possible consistency result.

The paper is organised as follows. In the next section, we introduce various classes of Petri nets used throughout the paper. Section 3 defines net refinement, the basic device to compose nets. Sections 4 and 5 develop a variant of the structured operational semantics (SOS) for the class of composite nets defined through net refinement. Sections 6–8 present an algebra of process expressions based on the formalism developed in the preceding sections, and prove the above consistency result. Section 9 completes the comparative study of the net algebra and expression algebra, by extending the consistency results to a partial order semantics based on Mazurkiewicz traces.

The paper is a full version of the conference paper [6].

2 Preliminaries

Throughout the paper, we use the standard mathematical notation. In particular, $\mathsf{mult}(Z)$ denotes the set of all finite multisets over a set Z, and $\uplus$ the disjoint set union. Other multiset-related notation are the sum (+) and difference (-).

Each operator on nets will be based on a special net Ω_{op} whose transitions $v_1, v_2, \ldots$ are refined by the corresponding nets $\Sigma_1, \Sigma_2, \ldots$ in the process of forming a new net $\Sigma = \Omega_{\mathsf{op}}(\Sigma_1, \Sigma_2, \ldots)$. To carry this out, we need to distinguish nets that are easily composable with one another. The chosen class of Petri nets, called *boxes*, have interfaces expressed by labellings of places and transitions. There are two main classes of boxes which we will be interested in, viz. plain boxes and operator boxes. Plain boxes ($\Sigma, \Sigma_1, \Sigma_2, \ldots$ above) form the class of elements of our Petri net domain upon which various operators are defined. Operator boxes (Ω_{op} above) are patterns (or functions) defining the ways of constructing new plain boxes out of given ones.

Actions and relabellings We assume a set Lab of *actions* to be given; each $a \in \mathsf{Lab}$ represents some interface activity. A *relabelling* ρ is a relation $\rho \subseteq \mathsf{mult}(\mathsf{Lab}) \times \mathsf{Lab}$ such that $(\emptyset, a) \in \rho$ if and only if $\rho = \{(\emptyset, a)\}$. The intuition behind a pair (Γ, a) belonging to ρ is that it specifies some interface change which can be applied to a (finite) group of transitions whose labels match the argument, i.e., the multiset of actions Γ, and which are synchronised to yield a new transition labelled a.

A *constant* relabelling, $\rho_a = \{(\emptyset, a)\}$, where a is an action in Lab, can be identified with a itself, so that we may consider the set of actions Lab to be embedded in the set of all relabellings. If a relabelling is not constant, then it will be called *transformational*; in that case, the empty set will not be in its domain, in order not to *create an action out of nothing*. The *identity* relabelling, $\rho_{id} = \{(\{a\}, a) \mid a \in \mathsf{Lab}\}$ captures the 'keep things as they are' (non)change.

Labelled nets By a *(marked) labelled net* we will mean a tuple $\Sigma = (S, T, W, \lambda, M)$ such that: S and T are disjoint sets of respectively *places* and *transitions*; W is a *weight function* from the set $(S{\times}T) \cup (T{\times}S)$ to the set of natural numbers $\mathbf{N}$; λ is a *labelling* function for places and transitions such that $\lambda(s) \in \{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$, for every place $s \in S$, and $\lambda(t)$ is a relabelling ρ, for every transition $t \in T$; and M is a *marking*, i.e., a mapping assigning a natural number to each place $s \in S$.[1] We adopt the standard rules about representing nets as directed graphs. To avoid ambiguity, we will sometime decorate the various components of Σ with the index Σ; thus, T_Σ denotes the set of transitions of Σ, etc. A net is *finite* if both S and T are finite sets. Figure 1 shows the graph of a labelled net Σ_0.

[1] Sometimes we will treat M as a multiset or as a set, if $M(S) \subseteq \{0, 1\}$.

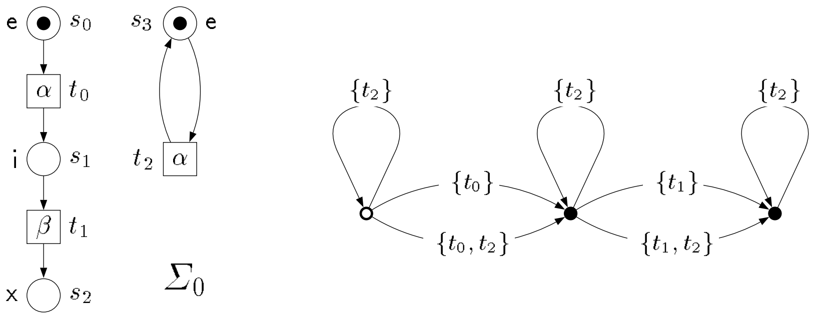


Figure 1. A labelled net and its reachability graph.

If the labelling of a place s is e, i or x, then s is an *entry*, *internal* or *exit* place, respectively. By convention, ${}^\circ\Sigma$, Σ° and $\ddot{\Sigma}$ denote respectively the entry, exit and internal places of Σ. For every place (transition) x, we use ${}^\bullet x$ to denote is pre-set, i.e., the set of all transitions (places) y such that there is an arc from y to x, that is, $W(y, x) > 0$. The post-set $x^\bullet$ is defined in a similar way. The pre- and post-set notation extends in the usual way to sets R of places and transitions, e.g., ${}^\bullet R = \bigcup_{r \in R} {}^\bullet r$. In what follows, all nets are assumed to be T-*restricted*, i.e., the pre- and post-sets of each transition are nonempty. Σ is called *simple* if W always returns 0 or 1, and *pure* if for all transitions $t \in T$, ${}^\bullet t \cap t^\bullet = \emptyset$. For the labelled net of figure 1 we have ${}^\circ\Sigma_0 = \{s_0, s_3\}$, $\Sigma_0^\circ = \{s_2\}$, $\ddot{\Sigma}_0 = \{s_1\}$, ${}^\bullet s_0 = \emptyset$ and $\{s_0, s_1\}^\bullet = \{t_0, t_1\}$. This net is finite and simple, but not pure as $s_3 \in {}^\bullet t_2 \cap t_2^\bullet$.

We will use three explicit ways of modifying the marking of Σ. We define $\lfloor \Sigma \rfloor$ as $(S, T, W, \lambda, \emptyset)$ which amounts to erasing all the tokens. Moreover, $\overline{\Sigma}$ and $\underline{\Sigma}$ are, respectively, $(S, T, W, \lambda, {}^\circ\Sigma)$ and $(S, T, W, \lambda, \Sigma^\circ)$. These operations correspond to placing one token on each entry (resp. exit) place and nothing elsewhere, forming the *entry marking* (resp. the *exit marking*) of Σ.

Execution semantics The behaviour of Σ is defined by its finite step sequence semantics: a finite multiset of transitions U, called a *step*, is *enabled* by Σ if for every place $s \in S$, $M(s) \geq \sum_{t \in U}(U(t) \cdot W(s, t))$. We denote this by $M\,[U\rangle$ or $\Sigma\,[U\rangle$. An enabled step U can be *executed* leading to a follower marking M' defined, for every $s \in S$, by

$$M'(s) = M(s) + \sum_{t \in U}(U(t) \cdot (W(t, s) - W(s, t))).$$

We will denote this by $M\,[U\rangle\,M'$ or $\Sigma\,[U\rangle\,\Sigma'$, where $\Sigma' = (S, T, W, \lambda, M')$. For Σ_0 in figure 1, $\{t_0, t_2\}$ is an enabled step. After its execution, $\{t_1\}$ is enabled and, hence, $\{t_0, t_2\}\{t_1\}$ is a step sequence of Σ_0.

Transition labelling may be extended to steps, through the formula

$$\lambda(U) \;=\; \sum_{t \in U} U(t) \cdot \Big\{\lambda(t)\Big\} \;\in\; \mathsf{mult}(\mathsf{Lab}).$$

In particular, we will denote $\Sigma\,[\Gamma\rangle_{\mathsf{lab}}\,\Theta$ whenever there is a multiset of transitions U such that $\Sigma\,[U\rangle\,\Theta$ and $\Gamma = \lambda(U)$. This allows one to translate various behavioural notions defined in terms of multisets of transitions into notions based on multisets of transition labels (or *labelled steps*).

A *step sequence* of Σ is a possibly empty sequence of steps, $\omega = U_1 \ldots U_k$, such that there are nets $\Sigma_1, \ldots, \Sigma_k$ satisfying $\Sigma\,[U_1\rangle\,\Sigma_1\,[U_2\rangle \cdots [U_k\rangle\,\Sigma_k$. We will denote this by $\Sigma\,[\omega\rangle\,\Sigma_k$ or $\Sigma_k \in [\Sigma\rangle$, and call Σ_k *derivable* from Σ and its marking, M_{Σ_k}, *reachable* from M_Σ.

A marking M is *safe* if $M(S) \subseteq \{0, 1\}$. A marking is *clean* if it is not a proper super-multisetset of ${}^\circ\Sigma$ nor Σ°, i.e., if ${}^\circ\Sigma \subseteq M$ or $\Sigma^\circ \subseteq M$ implies ${}^\circ\Sigma = M$ or $\Sigma^\circ = M$, respectively. The marking of the net in figure 1 is both safe and clean. A labelled net Σ is: *ex-restricted* if ${}^\circ\Sigma \neq \emptyset \neq \Sigma^\circ$; *e-directed* if ${}^\bullet({}^\circ\Sigma) = \emptyset$; *x-directed* if $(\Sigma^\circ)^\bullet = \emptyset$; and *ex-directed* if it is both e-directed and x-directed. Σ is *ex-exclusive* if, for every marking M reachable from M_Σ or ${}^\circ\Sigma$ or Σ°, it is the case that $M \cap {}^\circ\Sigma = \emptyset$ or $M \cap \Sigma^\circ = \emptyset$.

Let ind_Σ be the symmetric, not necessarily transitive, relation on the transitions of Σ, defined by

$$\mathsf{ind}_\Sigma \;=\; \Big\{(t,u) \in T \times T \;\Big|\; ({}^\bullet t \cup t^\bullet) \cap ({}^\bullet u \cup u^\bullet) = \emptyset\Big\}.$$

It is called the *independence relation*, because two distinct transitions belonging to ind_Σ have no impact on their respective environments. If they are both enabled separately, then they are enabled simultaneously (as a step). Conversely, if Σ is safe (which will later be our exclusive case of interest), it can be shown that any two transitions occurring in the same step are independent. Notice that T-restrictedness implies the irreflexivity of the independence relation, i.e., $\mathsf{ind}_\Sigma \cap id_{T_\Sigma} = \emptyset$, since for every $t \in T$, $(t,t) \notin \mathsf{ind}_\Sigma$. T-restrictedness also implies that the emptiness (non-emptiness) of markings is preserved over transition steps. Later, when we say that a net is 'marked', this will mean that its marking is nonempty.

Behavioural equivalence Although the whole set of step sequences of a net may be specified by defining the *full reachability graph* (see figure 1), we do not find it a satisfactory representation in the presence of labellings. For example, figure 2 demonstrates that isomorphism (and, indeed, other reasonable notion of behavioural equivalence) of reachability graphs is not preserved by sequential composition of nets. This is due to the fact that it is necessary to distinguish the entry and exit markings when comparing the behaviour of nets which are subsequently composed. Instead of modifying the definition of isomorphism, we address this problem by adding to Σ (artificially, and only for the purpose of defining the reachability graph) two fresh transitions, skip and redo, so that ${}^\bullet\mathsf{skip} = \mathsf{redo}^\bullet = {}^\circ\Sigma$, $\mathsf{skip}^\bullet = {}^\bullet\mathsf{redo} = \Sigma^\circ$, $\lambda(\mathsf{skip}) = \mathsf{skip}$ and $\lambda(\mathsf{redo}) = \mathsf{redo}$. Moreover, all the arcs adjacent to the skip and redo transitions are unitary and $\mathsf{redo}, \mathsf{skip} \notin \mathsf{Lab}$. Denote the net Σ augmented with skip and redo by Σ_{sr}.

The *transition system* of a marked net Σ is defined as $\mathsf{ts}_\Sigma = (V, L, A, v_0)$ where $V = \{\Theta \mid \mathsf{skip}, \mathsf{redo} \notin T_\Theta \,\wedge\, \Theta_{\mathsf{sr}} \in [\Sigma_{\mathsf{sr}}\rangle\,\}$ is the set of states, $v_0 = \Sigma$ is the initial state, $L = \mathsf{mult}(\mathsf{Lab} \cup \{\mathsf{redo}, \mathsf{skip}\})$ is the set of arc labels, and $A = \{(\Theta, \Gamma, \Psi) \in V \times L \times V \mid \Theta_{\mathsf{sr}}\, [\Gamma\rangle_{\mathsf{lab}}\, \Psi_{\mathsf{sr}}\}$ is the set of arcs. In other words, ts_Σ is the labelled full reachability graph of Σ_{sr} with all references to skip and redo in the nodes of the graph (but not in the arc labels) erased. The transition system of an unmarked net Σ is defined as $\mathsf{ts}_\Sigma = \mathsf{ts}_{\overline{\Sigma}}$.

Figure 2 shows that adding skip and redo does solve the problem; although the nets Σ and Θ have isomorphic reachability graphs, their ts's are different. This is not a mere chance; ts-isomorphism is a *congruence* in the algebra of nets described in this paper.

2.1 Two base classes of labelled nets

Plain boxes A *box* is a (possibly infinite) ex-restricted labelled net Σ. Ex-restrictedness is important since without it, composing boxes would yield undesired results [7, 8].

A box is, by definition, *plain* if each transition $t \in T_\Sigma$ is labelled by a constant relabelling, i.e., $\lambda_\Sigma(t) \in \mathsf{Lab}$. A plain box Σ is *static* if $M_\Sigma = \emptyset$ and every marking reachable from ${}^\circ\Sigma$ or Σ° is safe and clean. A plain box Σ is *dynamic* if $M_\Sigma \neq \emptyset$ and every marking reachable from M_Σ or ${}^\circ\Sigma$ or Σ° is safe and clean. A dynamic box Σ is an *entry* (*exit*) box if $M_\Sigma = {}^\circ\Sigma$ (resp. $M_\Sigma = \Sigma^\circ$). Note that the labelled net Σ_0 in figure 1 is not a box since it has a reachable marking, $M = \{s_2, s_3\}$, which is not clean. The sets of plain entry, dynamic, exit and static boxes will, respectively, be denoted by Box^e, Box^d, Box^x, and Box^s.

Proposition 2.1. Let Σ be a dynamic box and U be a step enabled by Σ.

(1) Every labelled net derivable from Σ is a dynamic box.
(2) U is a set of mutually independent transitions: $U \times U \subseteq \mathsf{ind}_\Sigma \cup id_{T_\Sigma}$.
(3) All the arcs adjacent to the transitions in U are unitary: $W_\Sigma(U \times S_\Sigma) \cup W_\Sigma(S_\Sigma \times U) \subseteq \{0, 1\}$.

Proof: (1) Suppose that Θ is derivable from a dynamic box Σ. Θ is marked since Σ is T-restricted. Clearly, all the markings reachable from M_Θ and ${}^\circ\Theta$ and Θ° are safe and clean since those reachable from M_Σ and ${}^\circ\Sigma$ and Σ° are safe and clean and ${}^\circ\Theta = {}^\circ\Sigma$ and $\Theta^\circ = \Sigma^\circ$. Hence Θ is a dynamic box.

(2,3) Follow directly from the safeness of the markings before and after the execution of step U. $\square$

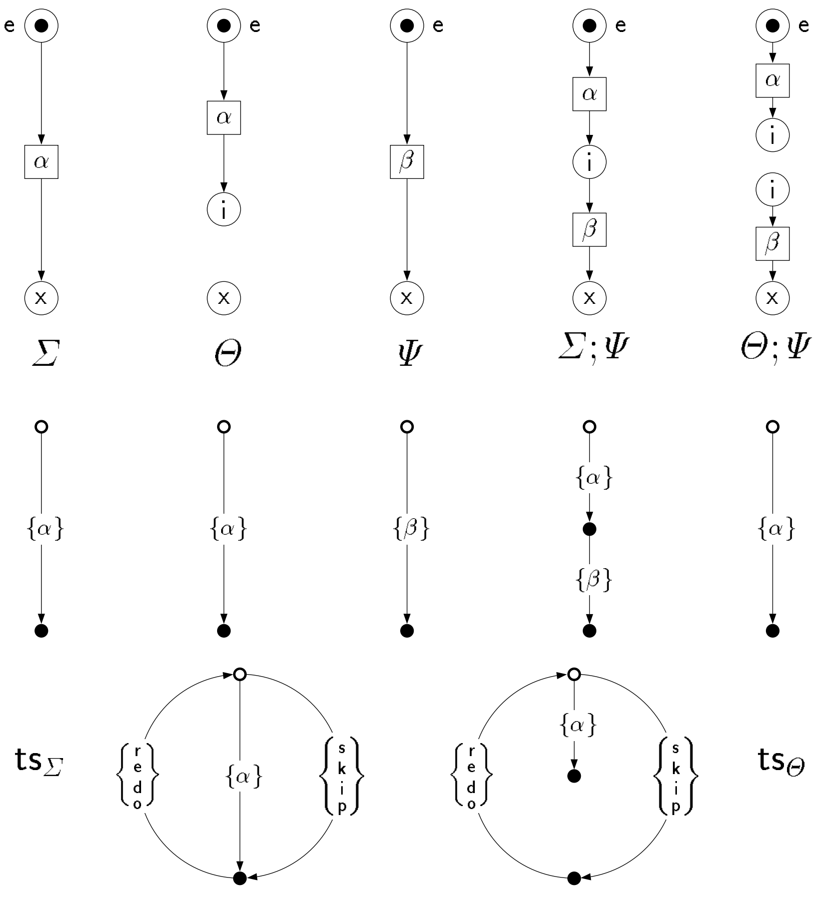


Figure 2. Five nets and the corresponding (labelled) full reachability graphs demonstrating that isomorphism of reachability graphs is not preserved by sequential composition; the two discriminating ts's for Σ and Θ are also shown.

Operator boxes An operator box is a simple box Ω with all relabellings being transformational such that $M_\Omega = \emptyset$ and all the markings reachable from ${}^\circ\Omega$ or Ω° are safe and clean. But we still need to impose on Ω one more property, called *factorisability* [18], defined next.

As the transitions of Ω are meant to be refined by potentially complicated boxes, it is justified to consider that their execution may take long time or, indeed, may last indefinitely. This may be captured by a special kind of extended markings.

A *complex marking* of Ω is a pair $\mathcal{M} = (M, Q)$ composed of a normal marking M of Ω (i.e., a possibly infinite multiset of places) and a finite multiset Q of *engaged* transitions of Ω.[2] The *direct reachability* between two complex markings, $\mathcal{M} = (M, Q)$ and $\mathcal{M}' = (M', Q')$, is defined thus. We have $\mathcal{M}\ [U + V^+ + Z^-\rangle\ \mathcal{M}'$ if there are finite multisets of transitions U, V and Z such that $Z \subseteq Q$, $Q' = Q + V - Z$ and for every $s \in S$, $M(s) \geq m_s$ and $M'(s) = M(s) - m_s + m'_s$, where

$$\begin{aligned} m_s &= \textstyle\sum_{t \in U+V} W(s,t) \cdot (U(t) + V(t)) \\ m'_s &= \textstyle\sum_{t \in U+Z} W(t,s) \cdot (U(t) + Z(t)). \end{aligned}$$

A tuple of sets of transitions of Ω, $\mu = (\mu_e, \mu_d, \mu_x, \mu_s)$ is a *factorisation* of a complex safe[3] marking (M, Q) if $\mu_d = Q$ and

$$M = \biguplus_{v \in \mu_e} {}^\bullet v \uplus \biguplus_{v \in \mu_x} v^\bullet$$

and $\mu_s = T_\Omega \setminus (\mu_e \cup \mu_d \cup \mu_x)$. Ω itself will be called *factorisable* if for every safe complex marking of Ω reachable from ${}^\circ\Omega$ or Ω°, there is at least one factorisation. We will denote by fact_Ω the set of all the factorisations of all the complex markings of Ω reachable from ${}^\circ\Omega$ or Ω°, and including also the only factorisation $(\emptyset, \emptyset, \emptyset, T_\Omega)$ of the empty marking.

Factorisability is absolutely necessary for the consistency results that will be obtained in this paper. In [5, 18] we show counterexamples when an operator box is only assumed to be simple, safe and clean, but not factorisible. Fortunately, but of course not coincidentally, all the standard process algebraic constructs such as prefix, sequence, choice (of various sorts), parallel composition (of various kinds), restriction, etc, can all be translated into (factorisable) operator boxes [5, 7].

Properties of factorisations A factorisation of a reachable marking (M, Q) is essentially a way of representing its real part, M, as the disjoint union of the pre-sets of a set of transitions, μ_e, and the post-sets of another set of transitions, μ_x. Intuitively, μ_e are transitions which can be concurrently executed at (M, Q), and μ_x are (possibly) transitions which have just been concurrently executed. However, such an interpretation may be somewhat misleading since, besides the fact that μ_e and μ_x may be infinite, it may happen that although (M, Q) is a reachable marking and $v \in \mu_x$, neither $(M \backslash v^\bullet, Q \cup \{v\})$ nor $((M \backslash v^\bullet) \cup {}^\bullet v, Q)$ are reachable from the entry or exit marking of Ω; it may even happen that v is a dead transition, whose sole role is to ensure the factorisability of Ω.

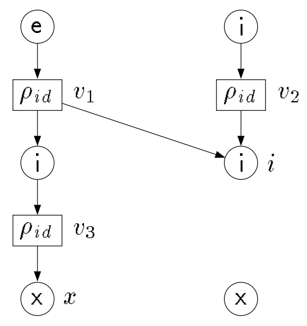


Figure 3. An operator box with a dead transition.

[2] A standard marking M may then be identified with the complex marking $(M, \emptyset)$.

[3] I.e., both M and Q are sets.

This is illustrated in figure 3 where the operator is factorisable but the transition v_2 is dead from the entry and exit markings. Yet if we dropped this dead (hence supposedly useless) transition, the operator would become unfactorisable since after the occurrences of v_1 and v_3, leading to the marking $(\{i, x\}, \emptyset)$, the internal place i could no longer be factored out. Notice that, contrary to what we have said about μ_x, if $v \in \mu_e$ then both $(M \setminus {}^\bullet v, Q \cup \{v\})$ and $((M \setminus {}^\bullet v) \cup v^\bullet, Q)$ are reachable from the entry or exit marking of Ω.

In what follows, we will need several basic properties of factorisations which, for ease of reference, are gathered in the proposition below. In what follows, we will call a transition v of an operator box Ω *reversible* if, for every complex marking (M, Q) reachable from ${}^\circ\Omega$ or Ω°, $v^\bullet \subseteq M$ implies that $(M \setminus v^\bullet, Q \cup \{v\})$ is also a marking reachable from ${}^\circ\Omega$ or Ω°. In figure 3, v_1 and v_3 are both reversible, while v_2 is not.

Proposition 2.2. Let Ω be an operator box, $\mu \in \mathsf{fact}_\Omega$ be a factorisation of a safe complex marking (M, Q) reachable from ${}^\circ\Omega$ or Ω°, and T_{rev} be the set of all reversible transitions of Ω.

(1) μ_e, μ_d, μ_x and μ_s are disjoint sets of transitions, and μ_d is finite.
(2) For every finite $U \subseteq T_\Omega$ and every finite $Z \subseteq T_{rev}$ such that $\biguplus_{v \in U} {}^\bullet v \uplus \biguplus_{v \in Z} v^\bullet \subseteq M$, it is the case that $(M \setminus ({}^\bullet U \cup Z^\bullet), Q \uplus U \uplus Z)$ is a marking reachable from ${}^\circ\Omega$ or Ω°. Moreover, both $(M \cup {}^\bullet Q, \emptyset)$ and $(M \cup Q^\bullet, \emptyset)$ are markings reachable from ${}^\circ\Omega$ or Ω°.
(3) For every $V \subseteq \mu_d$, every finite $U \subseteq \mu_e$, and every finite $Z \subseteq \mu_x \cap T_{rev}$, the following are factorisations in fact_Ω: $(\ \mu_e \setminus U\ ,\ \mu_d \cup U\ ,\ \mu_x\ ,\ \mu_s\ )$, $(\ \mu_e \cup V\ ,\ \mu_d \setminus V\ ,\ \mu_x\ ,\ \mu_s\ )$, $(\ \mu_e\ ,\ \mu_d \setminus V\ ,\ \mu_x \cup V\ ,\ \mu_s\ )$ and $(\ \mu_e\ ,\ \mu_d \cup Z\ ,\ \mu_x \setminus Z\ ,\ \mu_s\ )$.
(4) The transitions of $\mu_e \cup \mu_d \cup (\mu_x \cap T_{rev})$ are mutually independent, and the transitions in μ_x are such that

$$\mu_x^\bullet \cap ({}^\bullet\mu_e \cup \mu_e^\bullet) \ = \ \mu_x^\bullet \cap ({}^\bullet\mu_d \cup \mu_d^\bullet) \ = \ (\mu_x \setminus T_{rev})^\bullet \cap ({}^\bullet(\mu_x \cap T_{rev}) \cup (\mu_x \cap T_{rev})^\bullet) \ = \ \emptyset$$

and $v^\bullet \cap w^\bullet = \emptyset$, for all distinct $v, w \in \mu_x$.
(5) If ${}^\circ\Omega \subseteq M$ or $\Omega^\circ \subseteq M$ then $\mu_d = \emptyset$ and $M = {}^\circ\Omega$ or $M = \Omega^\circ$, respectively.

Proof: The various parts of this proposition follow directly from the definition of a factorisation of a complex marking, and the safeness and cleanness of Ω. □

Domain of application Let $\Sigma : T_\Omega \to \mathsf{Box}^s \cup \mathsf{Box}^d$ be a function from the transitions of Ω to static and dynamic boxes. We will refer to Σ as an *Ω-tuple* and denote $\Sigma(v)$ by Σ_v, for every $v \in T_\Omega$. For a set of transitions $V \subseteq T_\Omega$, $\Sigma_V = \{\Sigma_v \mid v \in V\}$. If the set of transitions of Ω is finite, we assume that $T_\Omega = \{v_1, \ldots, v_n\}$ and then denote $\Sigma = (\Sigma_{v_1}, \ldots, \Sigma_{v_n})$. We extend the notion of factorisation to an Ω-tuple Σ of static and dynamic boxes; the *factorisation* of Σ is the quadruple $\mu = (\mu_e, \mu_d, \mu_x, \mu_s)$ such that $\mu_\delta = \{v \mid \Sigma_v \in \mathsf{Box}^\delta\}$ for $\delta \in \{e, x, s\}$, and $\mu_d = \{v \mid \Sigma_v \in \mathsf{Box}^d \setminus (\mathsf{Box}^e \cup \mathsf{Box}^x)\}$.

The *domain of application* of Ω, denoted by dom_Ω, is then defined as the set comprising every Ω–tuple of static and dynamic boxes Σ whose factorisation belongs to fact_Ω, and such that, for every $v \in T_\Omega$:

- If ${}^\bullet v \cap v^\bullet \neq \emptyset$ then Σ_v is ex-exclusive. (Dom1)
- If v is not reversible then Σ_v is x-directed. (Dom2)

(Dom1) is meant to prevent the non-safeness of refinements in which a box is refined into a transition with a side place. This refers to expressions such as $(\alpha \| \beta)^*$, which informally means: 'repeat the concurrent execution of α and β arbitrarily many times'. A naive attempt to translate this expression into a net gives a 2-safe, but not 1-safe net. The other condition, (Dom2), is related to the possible presence of dead transitions in an operator box. Referring again to the operator box in figure 3, we may observe that if we were allowed to refine the transition v_2 with a non-x-directed box Σ_{v_2}, then some behaviours originating from Σ_{v_2} could be possible from the entry marking of the refined net, while no behaviour beginning at the entry nor exit marking of the operator box can possibly involve v_2. It it therefore justifiable to insist that Σ_{v_2} be x-directed[4] in this case.

[4] We could have required that no transition be enabled at $\Sigma_{v_2}^\circ$, but x-directedness is a simple structural sufficient condition for this.

In the previously published work on the box algebra, it was assumed that all boxes are ex-directed and, in addition, that operator boxes are both finite (though this restriction was dropped in [12]) and pure. Then (Dom1) and (Dom2) are trivially satisfied. Making such assumptions resulted in a simplification of the formal treatment and proofs. The present framework is more difficult to handle, but at the same time it is much more expressive with obvious implications for the practical applicability of the box algebra.

3 Net Refinement

Net refinement embodies a mechanism by which transition refinement and interface change specified by relabellings are combined. Both operations are defined for an operator box Ω which serves as a pattern for gluing together a tuple of plain boxes Σ along their entry and exit interfaces. The relabellings annotating the transitions of Ω specify the interface changes to which the boxes in Σ are subjected.

As far as net refinement as such is concerned, the names (identities) of newly constructed transitions and places are basically irrelevant. However, in our dealing with partial order semantics of process expressions, it will be important that such naming is done systematically (cf. section 9). Also, the names play a crucial role when recursion is treated (see [7, 8]). We found it convenient to use trees as names. Moreover, if using such explicit names is not agreeble with the reader, one might rather think of these trees as simply an additional injective labelling for places and transitions of a labelled net (whose nodes are then defined up to isomorphism).

Place and transition trees We shall assume that there are two disjoint infinite sets of *basic* place and transition names, $\mathsf{P_{root}}$ and $\mathsf{T_{root}}$. Each name $\eta \in \mathsf{P_{root}} \cup \mathsf{T_{root}}$ can be viewed as a special tree with a single root labelled with η which is also a leaf. We shall also employ more complex trees as transition and place names, and use a linear notation to express such trees. To this end, an expression $x \lhd \mathsf{S}$, where x is a basic name in $\mathsf{P_{root}} \cup \mathsf{T_{root}}$ or a pair $(t, a) \in \mathsf{T_{root}} \times \mathsf{Lab}$, and S is a multiset of trees, denotes a tree where the trees of the multiset are appended (with their multiplicity) to an x-labelled root. Moreover, if $\mathsf{S} = \{p\}$ is a singleton then $x \lhd \mathsf{S}$ will be denoted by $x \lhd p$, and if S is empty then $x \lhd \mathsf{S} = x$.

We shall further assume that in every operator box, all the places and transitions are basic names (i.e., single root trees) from respectively $\mathsf{P_{root}}$ and $\mathsf{T_{root}}$. For the plain boxes, the trees used as names may be more complex. Each transition tree is a finite tree labelled with elements of $\mathsf{T_{root}}$ (at the leaves) and $\mathsf{T_{root}} \times \mathsf{Lab}$ (elsewhere), and each place tree is a possibly infinite (in depth and width) tree labelled with basic names from $\mathsf{P_{root}}$ and $\mathsf{T_{root}}$, which has the form $t_1 \lhd t_2 \lhd \ldots \lhd t_n \lhd s \lhd \mathsf{S}$, where $t_1, \ldots, t_n \in \mathsf{T_{root}}$ ($n \geq 0$) are transition names and $s \in \mathsf{P_{root}}$ is a place name (so that no confusion will be possible between transition-trees and place-trees: the latter always have a label from $\mathsf{P_{root}}$ and the former never). We comprise all these trees (including the basic ones consisting only of a root as special cases) in our sets of allowed transition and place names, denoted respectively by $\mathsf{P_{tree}}$ and $\mathsf{T_{tree}}$.

Formal definition Let Ω be an operator box, and $\Sigma \in \mathsf{dom}_\Omega$ be an Ω-tuple of static and dynamic boxes in its domain of application. The result of a simultaneous substitution of boxes Σ for the transitions in Ω is a labelled net $\Omega(\Sigma)$ whose components are defined below. The definition is illustrated in figure 4.

Places The set of places of $\Omega(\Sigma)$ is defined as the (disjoint) union

$$\begin{aligned} S_{\Omega(\Sigma)} &= \bigcup_{v \in T_\Omega} \mathsf{ST}^v_{\mathsf{new}} \qquad\qquad \cup \bigcup_{s \in S_\Omega} \mathsf{SP}^s_{\mathsf{new}} \\ &= \bigcup_{v \in T_\Omega} \left\{ v \lhd i \;\middle|\; i \in \ddot{\Sigma}_v \right\} \cup \bigcup_{s \in S_\Omega} \left\{ s \lhd \Big(\{v \lhd x_v\}_{v \in {}^\bullet s} + \{w \lhd e_w\}_{w \in s^\bullet} \Big) \;\middle|\; x_v \in \Sigma_v^\circ \wedge e_w \in {}^\circ\Sigma_w \right\}. \end{aligned}$$

If s is an isolated place, ${}^\bullet s = s^\bullet = \emptyset$, then, by the definition of the tree appending operation, $\mathsf{SP}^s_{\mathsf{new}} = \{s\}$. Notice that the multiset of trees in the definition of a place

$$p \;=\; s \lhd \Big(\{v \lhd x_v\}_{v \in {}^\bullet s} + \{w \lhd e_w\}_{w \in s^\bullet} \Big) \;\in\; \mathsf{SP}^s_{\mathsf{new}} \tag{1}$$

is in fact a set, possibly infinite, because even in case there is a side-loop between s and $v = w$, then $x_v \neq e_w$ since x_v is an exit place and e_w is an entry place of Σ_v. The following notation is useful in manipulating the tree names upon which a newly constructed place is based.

Let y be a transition in Ω. Then for $p = v\lhd i$ in $\mathsf{ST}^v_{\mathsf{new}}$, we define $\mathsf{trees}^y(p) = \{i\}$ if $v = y$, and $\mathsf{trees}^y(p) = \emptyset$ if $v \neq y$; moreover, for $p \in \mathsf{SP}^s_{\mathsf{new}}$ as in (1), we define

$$\mathsf{trees}^y(p) = \begin{cases} \{x_y, e_y\} & \text{if } y \in {}^\bullet s \cap s^\bullet \\ \{x_y\} & \text{if } y \in {}^\bullet s \setminus s^\bullet \\ \{e_y\} & \text{if } y \in s^\bullet \setminus {}^\bullet s \\ \emptyset & \text{otherwise.} \end{cases}$$

For example, in figure 4, $\mathsf{trees}^{v_1}(s_3\lhd\{v_1\lhd q_{13}, v_3\lhd q_{31}\}) = \{q_{13}, q_{31}\}$ and $\mathsf{trees}^{v_2}(p_7) = \emptyset$.

Marking The marking of a place p in $\Omega(\Sigma)$ is defined in the following way:

$$M_{\Omega(\Sigma)}(p) \;=\; \begin{cases} M_{\Sigma_v}(i) & \text{if} \quad p = v\lhd i \in \mathsf{ST}^v_{\mathsf{new}} \\ \sum\limits_{v\in {}^\bullet s} M_{\Sigma_v}(x_v) + \sum\limits_{w\in s^\bullet} M_{\Sigma_w}(e_w) & \text{if} \quad p \in \mathsf{SP}^s_{\mathsf{new}} \text{ is as in (1).} \end{cases} \tag{2}$$

Notice that if $s \in S_\Omega$ is an isolated place then $\mathsf{SP}^s_{\mathsf{new}} = \emptyset$ and $M_{\Omega(\Sigma)}(s) = 0$. Moreover, $M_{\Omega(\Sigma)}(p)$ is a well defined natural number due to proposition 2.2(4).[5]

Transitions The set of transitions of $\Omega(\Sigma)$ is defined as the (disjoint) union

$$T_{\Omega(\Sigma)} \;=\; \bigcup_{v\in T_\Omega} \mathsf{T}^v_{\mathsf{new}} \;=\; \bigcup_{v\in T_\Omega} \Big\{(v,\alpha)\lhd R \;\Big|\; R \in \mathsf{mult}(T_{\Sigma_v}) \wedge \Big(\lambda_{\Sigma_v}(R), \alpha\Big) \in \lambda_\Omega(v)\Big\}.$$

Notice that the multiset R in $(v,\alpha)\lhd R$ will never be empty since no pair in $\lambda_{\Sigma_v}(v)$ has the empty multiset as its left argument.

Similarly as for places, we will denote by $\mathsf{trees}(u)$ the multiset of transitions R upon which a newly constructed transition $u = (v,\alpha)\lhd R \in \mathsf{T}^v_{\mathsf{new}}$ is based. For example, in figure 4, $\mathsf{trees}((v_1, f)\lhd\{t_{11}, t_{12}\}) = \{t_{11}, t_{12}\}$ and $\mathsf{trees}((v_3, e)\lhd\{t_{31}\}) = \{t_{31}\}$.

Labelling The label of a place or transitions x in $\Omega(\Sigma)$ is defined in the following way:

$$\lambda_{\Omega(\Sigma)}(x) \;=\; \begin{cases} \mathsf{i} & \text{if} \quad x \in \mathsf{ST}^v_{\mathsf{new}} \\ \lambda_\Omega(s) & \text{if} \quad x \in \mathsf{SP}^s_{\mathsf{new}} \\ \alpha & \text{if} \quad x = (v,\alpha)\lhd R \in \mathsf{T}^v_{\mathsf{new}}. \end{cases} \tag{3}$$

Weight function For a place p in $S_{\Omega(\Sigma)}$ and transition u in $\mathsf{T}^v_{\mathsf{new}}$, the weight function is given by:

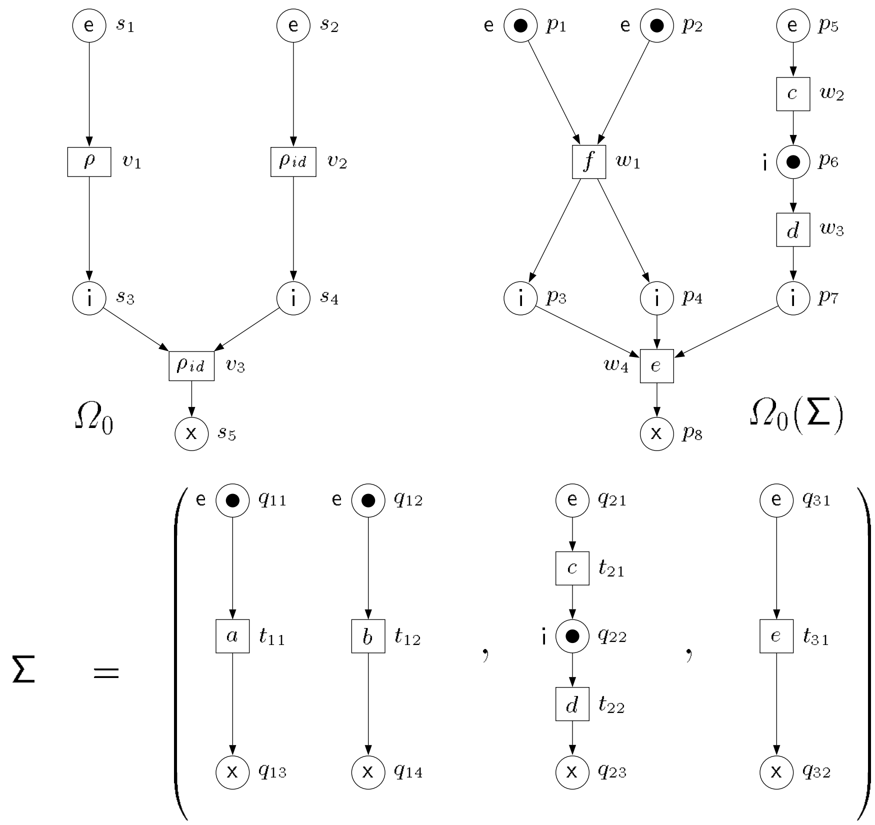
$$\begin{aligned} W_{\Omega(\Sigma)}(p,u) &= \sum_{z\in \mathsf{trees}^v(p)} \; \sum_{t\in \mathsf{trees}(u)} W_{\Sigma_v}(z,t) \cdot \mathsf{trees}(u)(t) \\ W_{\Omega(\Sigma)}(u,p) &= \sum_{z\in \mathsf{trees}^v(p)} \; \sum_{t\in \mathsf{trees}(u)} W_{\Sigma_v}(t,z) \cdot \mathsf{trees}(u)(t), \end{aligned} \tag{4}$$

where $\mathsf{trees}(u)(t)$ denotes the number of occurrences of t in the multiset $\mathsf{trees}(u)$.

3.1 A running example

Figure 4 shows an operator box Ω_0 which will serve as a running example. Ω_0 is a simple, pure, safe and clean box. A justification that Ω_0 is also factorisable is provided by the table below which lists all

[5] Later, we will also prove theorem 3.12 which asserts that $\Omega(\Sigma)$ has a marking which is not only well defined but even safe.



$$
\begin{array}{lll}
p_1 = s_1 \lhd v_1 \lhd q_{11} & p_2 = s_1 \lhd v_1 \lhd q_{12} & w_1 = (v_1, f) \lhd \{t_{11}, t_{12}\} \\
p_3 = s_3 \lhd \{v_1 \lhd q_{13}, v_3 \lhd q_{31}\} & p_4 = s_3 \lhd \{v_1 \lhd q_{14}, v_3 \lhd q_{31}\} & w_2 = (v_2, c) \lhd t_{21} \\
p_5 = s_2 \lhd v_2 \lhd q_{21} & p_6 = v_2 \lhd q_{22} & w_3 = (v_2, d) \lhd t_{22} \\
p_7 = s_4 \lhd \{v_2 \lhd q_{23}, v_3 \lhd q_{31}\} & p_8 = s_5 \lhd v_3 \lhd q_{32} & w_4 = (v_3, e) \lhd t_{31}
\end{array}
$$

Figure 4. Boxes of the running example ($\rho = \rho_{id} \backslash \{(\{a\}, a), (\{b\}, b)\} \cup \{(\{a, b\}, f)\}$).

complex markings (M, Q) reachable from the entry or exit marking of Ω_0, and their possible factorisations $(\mu_e, \mu_d, \mu_x, \mu_s)$:

M	Q	μ_e	μ_d	μ_x	μ_s
$\{s_1, s_2\}$	$\emptyset$	$\{v_1, v_2\}$	$\emptyset$	$\emptyset$	$\{v_3\}$
$\{s_1\}$	$\{v_2\}$	$\{v_1\}$	$\{v_2\}$	$\emptyset$	$\{v_3\}$
$\{s_2\}$	$\{v_1\}$	$\{v_2\}$	$\{v_1\}$	$\emptyset$	$\{v_3\}$
$\{s_1, s_4\}$	$\emptyset$	$\{v_1\}$	$\emptyset$	$\{v_2\}$	$\{v_3\}$
$\{s_2, s_3\}$	$\emptyset$	$\{v_2\}$	$\emptyset$	$\{v_1\}$	$\{v_3\}$
$\emptyset$	$\{v_1, v_2\}$	$\emptyset$	$\{v_1, v_2\}$	$\emptyset$	$\{v_3\}$
$\{s_3\}$	$\{v_2\}$	$\emptyset$	$\{v_2\}$	$\{v_1\}$	$\{v_3\}$
$\{s_4\}$	$\{v_1\}$	$\emptyset$	$\{v_1\}$	$\{v_2\}$	$\{v_3\}$
$\{s_3, s_4\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{v_1, v_2\}$	$\{v_3\}$
		$\{v_3\}$	$\emptyset$	$\emptyset$	$\{v_1, v_2\}$
$\emptyset$	$\{v_3\}$	$\emptyset$	$\{v_3\}$	$\emptyset$	$\{v_1, v_2\}$
$\{s_5\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{v_3\}$	$\{v_1, v_2\}$

(5)

Thus fact_{Ω_0} comprises thirteen different factorisations, as the only factorisation $(\emptyset, \emptyset, \emptyset, \{v_1, v_2, v_3\})$ of the empty marking $(M, Q) = (\emptyset, \emptyset)$ also belongs to fact_{Ω_0}. Notice, and this is crucial as we shall see, that a marking may have more than one factorisation, like $(M, Q) = (\{s_3, s_4\}, \emptyset)$ above.

One can see that $\mathsf{dom}_{\Omega_0} = \mathsf{Box}^d \times \mathsf{Box}^d \times \mathsf{Box}^s \cup \mathsf{Box}^s \times \mathsf{Box}^s \times \mathsf{Box}^d \cup \mathsf{Box}^s \times \mathsf{Box}^s \times \mathsf{Box}^s$. Figure 4 shows an Ω_0-tuple of boxes, $\Sigma = (\Sigma_{v_1}, \Sigma_{v_2}, \Sigma_{v_3})$ whose factorisation, $(\{v_1\}, \{v_2\}, \emptyset, \{v_3\})$, belongs to fact_{Ω_0}, and the tuple itself belongs to the domain of application of the operator box Ω_0. The box $\Omega_0(\Sigma)$ exemplifies net refinement, and the full linear notation for its place and transition (tree) names is also shown in figure 4.

3.2 Static properties of composite nets

In the rest of this paper, we will make frequent references to various notions used in the definition of net refinement, $\Omega(\Sigma)$. In particular, we will use the notations $\mathsf{T}^v_{\mathsf{new}}$, $\mathsf{ST}^v_{\mathsf{new}}$ and $\mathsf{SP}^s_{\mathsf{new}}$ (the latter also lifted to sets of places R, through $\mathsf{SP}^R_{\mathsf{new}} = \bigcup_{s \in R} \mathsf{SP}^s_{\mathsf{new}}$), and the notations $\mathsf{trees}(u)$ and $\mathsf{trees}^v(p)$, where u and p are respectively a transition and a place in $\Omega(\Sigma)$, and v is a transition in Ω. Using these notations, we can characterise several useful structural properties of the box $\Omega(\Sigma)$.

Proposition 3.1. The net $\Omega(\Sigma)$ is a plain box which is unmarked if and only if each Σ_v is unmarked.
Proof: We first observe that,

$$\forall v \in T_\Omega \; \forall q \in S_{\Sigma_v} \; \exists p \in S_{\Omega(\Sigma)} \; : \; q \in \mathsf{trees}^v(p). \tag{6}$$

Indeed, if $q \in \ddot{\Sigma}_v$ then we can take $p = v \lhd q \in \mathsf{SP}^v_{\mathsf{new}}$. If $q \in {}^\circ\Sigma_v$ then we take any $s \in {}^\bullet v$ (which can be done since Ω is T-restricted) and after that any $p = s \lhd \{\ldots, v \lhd q, \ldots\} \in \mathsf{SP}^s_{\mathsf{new}}$ will satisfy our requirement (at least one such p exists since all the boxes in Σ are ex-restricted). If $q \in \Sigma_v^\circ$, we proceed similarly. Moreover, we observe that,

$$\forall s \in S_\Omega \; : \; \mathsf{SP}^s_{\mathsf{new}} \neq \emptyset \tag{7}$$

which follows directly from the definition (for an isolated s), or from the fact that all the boxes in Σ are ex-restricted (for a non-isolated s). Indeed, in the latter case, we can choose, for each $v \in s^\bullet$ (if any) an entry place $e_v \in {}^\circ\Sigma_v$, and for each $w \in {}^\bullet s$ (if any) an exit place $x_w \in \Sigma_w^\circ$. Then $s \lhd (\{v \lhd e_v\}_{v \in s^\bullet} + \{w \lhd x_w\}_{w \in {}^\bullet s})$ is a place in $\mathsf{SP}^s_{\mathsf{new}}$. We now proceed with the proof proper.

It is clear that $\Omega(\Sigma)$ is a plain labelled net; in particular, place markings are all natural numbers thanks to the fact that only finitely many nets Σ_v may contribute to the marking of a place in $\Omega(\Sigma)$. The T-restrictedness and ex-restrictedness (needed to make it a plain box) are inherited from the corresponding properties of the components, as shown below.
$\Omega(\Sigma)$ is T-restricted. Let $u = (v, \alpha) \lhd R \in T_{\Omega(\Sigma)}$. To show ${}^\bullet u \neq \emptyset$, we take any $t \in R$ and $q \in {}^\bullet t$ (the former is possible since $R \neq \emptyset$, and the latter since Σ_v is T-restricted); by (4) it suffices to find $p \in S_{\Omega(\Sigma)}$ such that $q \in \mathsf{trees}^v(p)$. This, however, follows immediately from (6). To show $u^\bullet \neq \emptyset$, we proceed

similarly.

$\Omega(\Sigma)$ *is ex-restricted.* Follows immediately from the ex-restrictedness of Ω and (7).

The second part of the proposition follows directly from the definition of the marking of $\Omega(\Sigma)$ together with (6) and (7). $\square$

Proposition 3.2. If Ω and Σ_v, for every $v \in ({}^\circ\Omega)^\bullet$, are all e-directed boxes, then so is $\Omega(\Sigma)$. Similarly, if Ω and Σ_v, for every $v \in {}^\bullet(\Omega^\circ)$, are all x-directed boxes, then so is $\Omega(\Sigma)$.

Proof: If Ω is e-directed then each entry place p of $\Omega(\Sigma)$ is isolated, or arises from (i.e., it has a root labelled by) an entry place of Ω, followed by its output transitions, followed by entry places of some of the Σ_v's for $v \in ({}^\circ\Omega)^\bullet$. Since each such Σ_v is e-directed, by the definition of the arc weights, p may only have output transitions. And similarly for the x-directedness. $\square$

Proposition 3.3. $\lfloor\Omega(\Sigma)\rfloor = \lfloor\Omega\rfloor(\lfloor\Sigma\rfloor)$.

Proof: Follows directly from the definition of $M_{\Omega(\Sigma)}$. $\square$

Below, v and w are distinct transitions in Ω. The first three results follow directly from the definition of net refinement and the ex-restrictedness of boxes.

Proposition 3.4. Let $z \in \ddot{\Sigma}_v$, $q \in {}^\circ\Sigma_v$ and $r \in \Sigma_v^\circ$.

(1) There is $p \in \mathsf{ST}^v_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = \{z\}$.
(2) For every $s \in {}^\bullet v \backslash v^\bullet$ there is $p \in \mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = \{q\}$.
(3) For every $s \in v^\bullet \backslash {}^\bullet v$ there is $p \in \mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = \{r\}$.
(4) For every $s \in {}^\bullet v \cap v^\bullet$ there is $p \in \mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = \{q, r\}$. $\square$

Proposition 3.5. Suppose that one of the following holds:

$$\begin{array}{lll} q \in {}^\circ\Sigma_v \;\wedge\; r \in {}^\circ\Sigma_w \;\wedge\; s \in {}^\bullet v \cap {}^\bullet w & \text{or} & q \in \Sigma_v^\circ \;\wedge\; r \in {}^\circ\Sigma_w \;\wedge\; s \in v^\bullet \cap {}^\bullet w \quad \text{or} \\ q \in \Sigma_v^\circ \;\wedge\; r \in \Sigma_w^\circ \;\wedge\; s \in v^\bullet \cap w^\bullet & \text{or} & q \in {}^\circ\Sigma_v \;\wedge\; r \in \Sigma_w^\circ \;\wedge\; s \in {}^\bullet v \cap w^\bullet. \end{array}$$

Then there is a place p in $\mathsf{SP}^s_{\mathsf{new}}$ such that $q \in \mathsf{trees}^v(p)$ and $r \in \mathsf{trees}^w(p)$. $\square$

Proposition 3.6. Let p be a place in $\Omega(\Sigma)$, and $t \in \mathsf{T}^v_{\mathsf{new}}$. Then

$$p \in {}^\bullet t \iff \mathsf{trees}^v(p) \cap {}^\bullet\mathsf{trees}(t) \neq \emptyset \quad \text{and} \quad p \in t^\bullet \iff \mathsf{trees}^v(p) \cap \mathsf{trees}(t)^\bullet \neq \emptyset.$$

Moreover, if $p \in \mathsf{SP}^s_{\mathsf{new}}$ then

$$\begin{array}{lcl} p \in {}^\bullet t & \iff & (\mathsf{trees}^v(p) \cap {}^\bullet\mathsf{trees}(t) \cap \Sigma_v^\circ \neq \emptyset \wedge s \in v^\bullet) \;\vee\; (\mathsf{trees}^v(p) \cap {}^\bullet\mathsf{trees}(t) \cap {}^\circ\Sigma_v \neq \emptyset \wedge s \in {}^\bullet v) \\ p \in t^\bullet & \iff & (\mathsf{trees}^v(p) \cap \mathsf{trees}(t)^\bullet \cap \Sigma_v^\circ \neq \emptyset \wedge s \in v^\bullet) \;\vee\; (\mathsf{trees}^v(p) \cap \mathsf{trees}(t)^\bullet \cap {}^\circ\Sigma_v \neq \emptyset \wedge s \in {}^\bullet v). \end{array}$$

$\square$

Applying the above proposition to the example in figure 4, we obtain that $p_3 \in {}^\bullet w_4$ since $\mathsf{trees}^{v_3}(p_3) = \{q_{31}\}$ and ${}^\bullet\mathsf{trees}(w_4) = {}^\bullet\{t_{31}\} = \{q_{31}\}$.

In the results that now follow, we use predicates, In^e_t, In^x_t, Out^e_t and Out^x_t, where t is a transition in $\mathsf{T}^v_{\mathsf{new}}$, to respectively denote: ${}^\bullet\mathsf{trees}(t) \cap {}^\circ\Sigma_v \neq \emptyset$, ${}^\bullet\mathsf{trees}(t) \cap \Sigma_v^\circ \neq \emptyset$, $\mathsf{trees}(t)^\bullet \cap {}^\circ\Sigma_v \neq \emptyset$ and $\mathsf{trees}(t)^\bullet \cap \Sigma_v^\circ \neq \emptyset$.

Proposition 3.7. Let $t, u \in \mathsf{T}^v_{\mathsf{new}}$. Then

$$\begin{array}{lcl}
{}^\bullet t \cap {}^\bullet u \neq \emptyset & \iff & {}^\bullet\mathsf{trees}(t) \cap {}^\bullet\mathsf{trees}(u) \neq \emptyset \ \vee\ ({}^\bullet v \cap v^\bullet \neq \emptyset \ \wedge\ (In^e_t \wedge In^x_u \ \vee\ In^x_t \wedge In^e_u)) \\
{}^\bullet t \cap u^\bullet \neq \emptyset & \iff & {}^\bullet\mathsf{trees}(t) \cap \mathsf{trees}(u)^\bullet \neq \emptyset \ \vee\ ({}^\bullet v \cap v^\bullet \neq \emptyset \ \wedge\ (In^e_t \wedge Out^x_u \ \vee\ In^x_t \wedge Out^e_u)) \\
t^\bullet \cap {}^\bullet u \neq \emptyset & \iff & \mathsf{trees}(t)^\bullet \cap {}^\bullet\mathsf{trees}(u) \neq \emptyset \ \vee\ ({}^\bullet v \cap v^\bullet \neq \emptyset \ \wedge\ (Out^e_t \wedge In^x_u \ \vee\ Out^x_t \wedge In^e_u)) \\
t^\bullet \cap u^\bullet \neq \emptyset & \iff & \mathsf{trees}(t)^\bullet \cap \mathsf{trees}(u)^\bullet \neq \emptyset \ \vee\ ({}^\bullet v \cap v^\bullet \neq \emptyset \ \wedge\ (Out^e_t \wedge Out^x_u \ \vee\ Out^x_t \wedge Out^e_u)).
\end{array}$$

Proof: Suppose that $p \in {}^\bullet t \cap {}^\bullet u$. Then, by proposition 3.6, there are $q \in \mathsf{trees}^v(p) \cap {}^\bullet\mathsf{trees}(t)$ and $r \in \mathsf{trees}^v(p) \cap {}^\bullet\mathsf{trees}(u)$. If $q = r$ then ${}^\bullet\mathsf{trees}(t) \cap {}^\bullet\mathsf{trees}(u) \neq \emptyset$. Otherwise, $\mathsf{trees}^v(p) = \{q, r\}$ and, without loss of generality, $q \in {}^\circ\Sigma_v$ and $r \in \Sigma_v^\circ$ (and so $In^e_t \wedge In^x_u$). Moreover, $|\mathsf{trees}^v(p)| > 1$ implies ${}^\bullet v \cap v^\bullet \neq \emptyset$. Hence the ($\Longrightarrow$) implication holds. The reverse implication ($\Longleftarrow$) follows from propositions 3.4(4) and 3.6.

The other three equivalences can be shown in the same way. Similar remarks apply to the proofs of the next two propositions. □

Thus, for the example in figure 4, we obtain:

$$w_2^\bullet \cap {}^\bullet w_3 \neq \emptyset \text{ since } \{t_{21}\}^\bullet \cap {}^\bullet\{t_{22}\} = \{q_{22}\} \cap \{q_{22}\} \neq \emptyset.$$

Proposition 3.8. Let $t \in \mathsf{T}^v_{\mathsf{new}}$ and $u \in \mathsf{T}^w_{\mathsf{new}}$. Then

$$\begin{array}{lcl}
{}^\bullet t \cap {}^\bullet u \neq \emptyset & \iff & (In^e_t \wedge In^e_u \wedge {}^\bullet v \cap {}^\bullet w \neq \emptyset) \vee (In^x_t \wedge In^e_u \wedge v^\bullet \cap {}^\bullet w \neq \emptyset) \vee \\
& & (In^e_t \wedge In^x_u \wedge {}^\bullet v \cap w^\bullet \neq \emptyset) \vee (In^x_t \wedge In^x_u \wedge v^\bullet \cap w^\bullet \neq \emptyset) \\
{}^\bullet t \cap u^\bullet \neq \emptyset & \iff & (In^e_t \wedge Out^e_u \wedge {}^\bullet v \cap {}^\bullet w \neq \emptyset) \vee (In^x_t \wedge Out^e_u \wedge v^\bullet \cap {}^\bullet w \neq \emptyset) \vee \\
& & (In^e_t \wedge Out^x_u \wedge {}^\bullet v \cap w^\bullet \neq \emptyset) \vee (In^x_t \wedge Out^x_u \wedge v^\bullet \cap w^\bullet \neq \emptyset) \\
t^\bullet \cap {}^\bullet u \neq \emptyset & \iff & (Out^e_t \wedge In^e_u \wedge {}^\bullet v \cap {}^\bullet w \neq \emptyset) \vee (Out^x_t \wedge In^e_u \wedge v^\bullet \cap {}^\bullet w \neq \emptyset) \vee \\
& & (Out^e_t \wedge In^x_u \wedge {}^\bullet v \cap w^\bullet \neq \emptyset) \vee (Out^x_t \wedge In^x_u \wedge v^\bullet \cap w^\bullet \neq \emptyset) \\
t^\bullet \cap u^\bullet \neq \emptyset & \iff & (Out^e_t \wedge Out^e_u \wedge {}^\bullet v \cap {}^\bullet w \neq \emptyset) \vee (Out^x_t \wedge Out^e_u \wedge v^\bullet \cap {}^\bullet w \neq \emptyset) \vee \\
& & (Out^e_t \wedge Out^x_u \wedge {}^\bullet v \cap w^\bullet \neq \emptyset) \vee (Out^x_t \wedge Out^x_u \wedge v^\bullet \cap w^\bullet \neq \emptyset)
\end{array}$$

Proof: Suppose that $p \in {}^\bullet t \cap {}^\bullet u$. Then, by the definition of net refinement, there is $s \in ({}^\bullet v \cup v^\bullet) \cap ({}^\bullet w \cup w^\bullet)$ such that $p \in \mathsf{SP}^s_{\mathsf{new}}$. Hence the ($\Longrightarrow$) implication holds by proposition 3.6. The reverse implication ($\Longleftarrow$) follows directly from propositions 3.5 and 3.6. □

Thus, for the example in figure 4, we obtain:

$$w_1^\bullet \cap {}^\bullet w_4 \neq \emptyset \quad \text{since} \quad \left\{ \begin{array}{rcll}
\{t_{11}, t_{12}\}^\bullet \cap \Sigma^\circ_{v_1} = & \{q_{13}, q_{14}\} \cap \{q_{13}, q_{14}\} & \neq \emptyset & \text{(i.e., } Out^x_{w_1}) \\
{}^\bullet\{t_{31}\} \cap {}^\circ\Sigma_{v_3} = & \{q_{31}\} \cap \{q_{31}\} & \neq \emptyset & \text{(i.e., } In^e_{w_4}) \\
v_1^\bullet \cap {}^\bullet v_3 = & \{s_3\} \cap \{s_3, s_4\} & \neq \emptyset. &
\end{array}\right.$$

Proposition 3.9. Let $t \in \mathsf{T}^v_{\mathsf{new}}$. Then

$$\begin{array}{lcl}
{}^\bullet t \cap {}^\circ\Omega(\Sigma) \neq \emptyset & \iff & (In^e_t \wedge {}^\bullet v \cap {}^\circ\Omega \neq \emptyset) \vee (In^x_t \wedge v^\bullet \cap {}^\circ\Omega \neq \emptyset) \\
t^\bullet \cap {}^\circ\Omega(\Sigma) \neq \emptyset & \iff & (Out^e_t \wedge {}^\bullet v \cap {}^\circ\Omega \neq \emptyset) \vee (Out^x_t \wedge v^\bullet \cap {}^\circ\Omega \neq \emptyset) \\
t^\bullet \cap \Omega(\Sigma)^\circ \neq \emptyset & \iff & (Out^e_t \wedge {}^\bullet v \cap \Omega^\circ \neq \emptyset) \vee (Out^x_t \wedge v^\bullet \cap \Omega^\circ \neq \emptyset) \\
{}^\bullet t \cap \Omega(\Sigma)^\circ \neq \emptyset & \iff & (In^e_t \wedge {}^\bullet v \cap \Omega^\circ \neq \emptyset) \vee (In^x_t \wedge v^\bullet \cap \Omega^\circ \neq \emptyset)
\end{array}$$

Proof: Suppose that $p \in {}^\bullet t \cap {}^\circ\Omega(\Sigma)$. Then, by the definition of net refinement, there is $s \in {}^\circ\Omega \cap ({}^\bullet v \cup v^\bullet)$ such that $p \in \mathsf{SP}^s_{\mathsf{new}}$. Hence the ($\Longrightarrow$) implication holds by proposition 3.6. The reverse implication ($\Longleftarrow$) follows directly from propositions 3.4 and 3.6. □

Thus, for the example in figure 4, we obtain:

$${}^\bullet w_2 \cap {}^\circ\Omega_0(\Sigma) \neq \emptyset \quad \text{since} \quad \left\{ \begin{array}{rcll}
{}^\bullet\{t_{21}\} \cap {}^\circ\Sigma_{v_2} = & \{q_{21}\} \cap \{q_{21}\} \neq \emptyset & & \text{(i.e., } In^e_{w_2}) \\
{}^\bullet v_2 \cap {}^\circ\Omega_0 = & \{s_2\} \cap \{s_1, s_2\} \neq \emptyset. & &
\end{array}\right.$$

3.3 Markings of composite nets

We now will show that the marking of any net obtained by applying an operator box to a tuple of boxes in its domain is safe and clean. This will be preceded by two auxiliary results clarifying the relationship between the marking of the refined net $\Omega(\Sigma)$ and the markings of the refining nets Σ. The results stem from an observation that although in the general formulation of net refinement a token in a newly created place can be contributed by any of the places which were used to construct that place, the restrictions imposed on an operator box and on the tuples of boxes in its domain mean that at most one such box can actually contribute tokens.

Let Ω be an operator box and $\Sigma \in \mathsf{dom}_\Omega$ be an Ω–tuple of boxes with the factorisation μ, fixed for the rest of this section. For every transition v in Ω, we define the set $\mathsf{mar}_\mu(v)$ of places of $\Omega(\Sigma)$ which are *markable* by v, in the following way:

$$\mathsf{mar}_\mu(v) = \begin{cases} \mathsf{SP}_{\mathsf{new}}^{({}^\bullet v)} & \text{if} \quad v \in \mu_e \\ \mathsf{SP}_{\mathsf{new}}^{({}^\bullet v \cup v^\bullet)} \cup \mathsf{ST}_{\mathsf{new}}^{v} & \text{if} \quad v \in \mu_d \\ \mathsf{SP}_{\mathsf{new}}^{(v^\bullet)} & \text{if} \quad v \in \mu_x \\ \emptyset & \text{if} \quad v \in \mu_s . \end{cases} \tag{8}$$

That is, $\mathsf{mar}_\mu(v)$ comprises all places into which the box Σ_v refining v can possibly insert tokens[6] (and so, in particular, $\mathsf{mar}_\mu(v)$ is empty for a static Σ_v). From proposition 2.2(4) it follows that a given place is markable by at most one transition, i.e., for all transitions v and w in Ω,

$$v \neq w \implies \mathsf{mar}_\mu(v) \cap \mathsf{mar}_\mu(w) = \emptyset. \tag{9}$$

Notice also that if $p \in \mathsf{mar}_\mu(v)$ then $\mathsf{trees}^v(p)$ is always non-empty. For our running example in figure 4, we will assume also that $\mu = (\{v_1\}, \{v_2\}, \emptyset, \{v_3\})$, which is the factorisation of the tuple Σ of figure 4, is the running factorisation. Then, we have: $\mathsf{mar}_\mu(v_1) = \{p_1, p_2\}$, $\mathsf{mar}_\mu(v_2) = \{p_5, p_6, p_7\}$ and $\mathsf{mar}_\mu(v_3) = \emptyset$. Using the notion of markable places, we can relate the marking of $\Omega(\Sigma)$ to those of the nets in Σ.

Lemma 3.10. Let p be a place in $\Omega(\Sigma)$. Then

$$M_{\Omega(\Sigma)}(p) = \begin{cases} Tot(M_{\Sigma_v}, \mathsf{trees}^v(p)) & \text{if } p \in \mathsf{mar}_\mu(v) \text{ for some } v \\ 0 & \text{if there is no } v \text{ such that } p \in \mathsf{mar}_\mu(v). \end{cases} \tag{10}$$

where $Tot(M,R) = \sum_{s \in R} M(s)$ is the *total number of tokens* on a finite set of places R.
Proof: First note that (10) is well formed due to (9). If $p \in \mathsf{ST}_{\mathsf{new}}^v$, for some v, then (10) follows directly from the definition of net refinement and (8). Suppose that $p \in \mathsf{SP}_{\mathsf{new}}^s$. If there is v such that $p \in \mathsf{mar}_\mu(v)$ then, by proposition 2.2(4) and (8),

$$w \in \Big(({}^\bullet s \cup s^\bullet) \cap (\mu_e \cup \mu_d \cup \mu_x)\Big) \backslash \{v\} \implies s \in {}^\bullet w \setminus w^\bullet \ \wedge \ w \in \mu_x$$

which in turn implies that $Tot(M_{\Sigma_w}, \mathsf{trees}^w(p)) = 0$, for all $w \in ({}^\bullet s \cup s^\bullet) \backslash \{v\}$. This and $M_\Omega(s) = 0$ yields $M_{\Omega(\Sigma)}(p) = Tot(M_{\Sigma_v}, \mathsf{trees}^v(p))$. If there is no v such that $p \in \mathsf{mar}_\mu(v)$ then $Tot(M_{\Sigma_w}, \mathsf{trees}^w(p)) = 0$, for all $w \in {}^\bullet s \cup s^\bullet$. This and $M_\Omega(s) = 0$ means that p is unmarked. □

Applying the above lemma to the places p_3 and p_6 in figure 4, we obtain $M_{\Omega_0(\Sigma)}(p_3) = 0$, since $p_3 \notin \mathsf{mar}_\mu(v_1) \cup \mathsf{mar}_\mu(v_2) \cup \mathsf{mar}_\mu(v_3)$, and

$$M_{\Omega_0(\Sigma)}(p_6) = Tot(\Sigma_{v_2}, \mathsf{trees}^{v_2}(p_6)) = M_{\Sigma_{v_2}}(q_{22}) = 1.$$

[6] In other words, Σ_v cannot insert a token into places other than $\mathsf{mar}_\mu(v)$.

Lemma 3.11. Let s be a place in Ω. Then all the places of $\mathsf{SP}^s_{\mathsf{new}}$ are marked in $\Omega(\Sigma)$ if and only if $s \in {}^\bullet\mu_e \cup \mu_x^\bullet$.
Proof: ($\Longrightarrow$) By lemma 3.10 and $\mathsf{SP}^s_{\mathsf{new}} \neq \emptyset$, there is v such that $\mathsf{mar}_\mu(v) \cap \mathsf{SP}^s_{\mathsf{new}} \neq \emptyset$. If $v \in {}^\bullet s \setminus s^\bullet$ and $\Sigma_v \notin \mathsf{Box}^x$ then there is $r \in \Sigma_v^\circ$ such that $M_{\Sigma_v}(r) = 0$. By proposition 3.4(3), there is a place p in $\mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = \{r\}$. Hence, by lemma 3.10, p is unmarked at $M_{\Omega(\Sigma)}$, a contradiction. Thus $v \in \mu_x$. For $v \in s^\bullet \setminus {}^\bullet s$ the argument is symmetric, yielding $v \in \mu_e$.
Suppose now that $v \in {}^\bullet s \cap s^\bullet$. Then, by the ex-exclusiveness of Σ_v (by (Dom1)) and without loss of generality, we may assume that ${}^\circ\Sigma_v \cap M_{\Sigma_v} = \emptyset$. If $\Sigma_v \notin \mathsf{Box}^x$ then there is $r \in \Sigma_v^\circ$ such that $M_{\Sigma_v}(r) = 0$. By proposition 3.4(4), there is a place p in $\mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = \{q, r\}$, where q is a place in ${}^\circ\Sigma_v$. Hence, by lemma 3.10, p is unmarked at $M_{\Omega(\Sigma)}$, a contradiction. Thus $v \in \mu_x$.

($\Longleftarrow$) If $s \in {}^\bullet\mu_e \cup \mu_x^\bullet$ then all the places in $\mathsf{SP}^s_{\mathsf{new}}$ are marked, which follows from (8), lemma 3.10, and the definition of net refinement. □

Theorem 3.12. The marking of $\Omega(\Sigma)$ is safe and clean.
Proof: The marking is safe by lemma 3.10, the safeness of the boxes in Σ, and the ex-exclusiveness of the boxes Σ_v such that ${}^\bullet v \cap v^\bullet \neq \emptyset$ (by (Dom1)). To show that it is also clean, suppose that ${}^\circ\Omega(\Sigma) \subseteq M_{\Omega(\Sigma)}$. From lemma 3.11 it follows that ${}^\circ\Omega \subseteq {}^\bullet\mu_e \cup \mu_x^\bullet$. Hence, by proposition 2.2(5), $\mu_d = \emptyset$ and ${}^\circ\Omega = {}^\bullet\mu_e \cup \mu_x^\bullet$. It then follows from the safeness of $M_{\Omega(\Sigma)}$, (8) and lemmata 3.10 and 3.11, that ${}^\circ\Omega(\Sigma) = M_{\Omega(\Sigma)}$. The case $\Omega(\Sigma)^\circ \subseteq M_{\Omega(\Sigma)}$ is symmetric. □

A useful corollary of the results proved in this section is the following closed formula for the safe and clean marking of $\Omega(\Sigma)$:

$$M_{\Omega(\Sigma)} \;=\; \mathsf{SP}^{({}^\bullet\mu_e \cup \mu_x^\bullet)}_{\mathsf{new}} \cup \bigcup_{v \in \mu_d} \left\{ p \in \left(\mathsf{SP}^{({}^\bullet v \cup v^\bullet)}_{\mathsf{new}} \cup \mathsf{ST}^v_{\mathsf{new}} \right) \;\middle|\; \mathsf{trees}^v(p) \cap M_{\Sigma_v} \neq \emptyset \right\}. \tag{11}$$

Applying it to the refinement in figure 4 yields the marking of $\Omega_0(\Sigma)$ (recall that the factorisation of Σ is $(\{v_1\}, \{v_2\}, \emptyset, \{v_3\})$):

$$\begin{aligned} M_{\Omega_0(\Sigma)} &= \mathsf{SP}^{({}^\bullet v_1)}_{\mathsf{new}} && \cup \left\{ p \in \left(\mathsf{SP}^{({}^\bullet v_2 \cup v_2^\bullet)}_{\mathsf{new}} \cup \mathsf{ST}^{v_2}_{\mathsf{new}} \right) \;\middle|\; \mathsf{trees}^{v_2}(p) \cap M_{\Sigma_{v_2}} \neq \emptyset \right\} \\ &= \mathsf{SP}^{s_1}_{\mathsf{new}} && \cup \left\{ p \in \left(\mathsf{SP}^{s_2}_{\mathsf{new}} \cup \mathsf{SP}^{s_4}_{\mathsf{new}} \cup \mathsf{ST}^{v_2}_{\mathsf{new}} \right) \;\middle|\; \mathsf{trees}^{v_2}(p) \cap \{q_{22}\} \neq \emptyset \right\} \\ &= \{p_1, p_2\} && \cup \left\{ p \in \{p_5, p_7, p_6\} \;\middle|\; \mathsf{trees}^{v_2}(p) \cap \{q_{22}\} \neq \emptyset \right\} \\ &= \{p_1, p_2, p_6\}. \end{aligned}$$

4 Structured operational semantics of composite boxes

The generic rules of the operational semantics of a process algebra (see, for instance, [20]) specify how the behaviour of a compound process term is related to the behaviour of its sub-terms. We will now take this idea and apply to the domain of compositional nets defined using operator boxes.

Let Ω be an operator box and Σ be an Ω–tuple in its domain, dom_Ω. When dealing with the operational semantics of the compositional box $\Omega(\Sigma)$, we shall use the notation

$$\left(\Omega \;:\; \Sigma \right) \xrightarrow{\;U\;} \left(\Omega \;:\; \Theta \right) \tag{12}$$

to mean that the boxes Σ can individually make moves which, when combined, yield step U and lead to new boxes Θ. By definition, this will be the case whenever U is a finite set of transitions of $\Omega(\Sigma)$ and, for every transition v in Ω, $U \cap \mathsf{T}^v_{\mathsf{new}} = \{u_1, \ldots, u_k\}$ is a finite set of transitions such that

$$\Sigma_v \left[\mathsf{trees}(u_1) + \cdots + \mathsf{trees}(u_k) \right\rangle \Theta_v. \tag{13}$$

Consider, for example, the boxes of the running example depicted in figure 4. Then

$$\left(\Omega_0 \;:\; \Sigma \right) \xrightarrow{\{w_1, w_3\}} \left(\Omega_0 \;:\; \Theta \right)$$

where Θ is the tuple of boxes shown in figure 5. Indeed, by setting $U = \{w_1, w_3\}$ we obtain $U \cap \mathsf{T}^{v_1}_{\mathsf{new}} = \{w_1\}$, $U \cap \mathsf{T}^{v_2}_{\mathsf{new}} = \{w_3\}$ and $U \cap \mathsf{T}^{v_3}_{\mathsf{new}} = \emptyset$. Moreover, $\mathsf{trees}(w_1) = \{t_{11}, t_{12}\}$ and $\mathsf{trees}(w_3) = \{t_{22}\}$. And, finally,

$$\Sigma_{v_1} \left[\{t_{11}, t_{12}\}\right\rangle \Theta_{v_1} \quad , \quad \Sigma_{v_2} \left[\{t_{22}\}\right\rangle \Theta_{v_2} \quad \text{and} \quad \Sigma_{v_3} \left[\emptyset\right\rangle \Theta_{v_3}.$$

Notice that $\{w_1, w_3\}$ is also a valid step for the box $\Omega_0(\Sigma)$ and $\Omega_0(\Sigma)\ [\{w_1, w_3\}\rangle\ \Omega_0(\Theta)$. We shall soon see that this is not accidental.

Each multiset $\mathsf{trees}(u_i)$ in (13) is in fact a set of mutually independent transitions of Σ_v, and $\mathsf{trees}(u_i)$ and $\mathsf{trees}(u_j)$ are disjoint sets, for all distinct i and j. Both properties follow from the safeness of the boxes in Σ and proposition 2.1(2). We can therefore denote the multiset sum in (13) as the disjoint union of k sets, $\mathsf{trees}(u_1) \uplus \cdots \uplus \mathsf{trees}(u_k)$. Notice also that $U \cap \mathsf{T}^{v}_{\mathsf{new}} \neq \emptyset$ (and so $\Sigma_v \neq \Theta_v$) only for finitely many v, since U is finite. Assuming that U is a multiset rather than a set would not add any new moves (12) since $\mathsf{T}^{v}_{\mathsf{new}} \cap \mathsf{T}^{w}_{\mathsf{new}} = \emptyset$ for $v \neq w$, and all boxes in Σ are safe. The above definition of operational semantics does not involve the *redo/skip* transitions, which are not refined but added only afterwards to obtain transition system semantics of boxes.

Instead of expressing behaviours in terms of transitions, it is also possible to express them using actions, through the labelling function

$$\lambda_{\Omega(\Sigma)}(U) \;=\; \sum_{u \in U} \{\lambda_{\Omega(\Sigma)}(u)\}. \tag{14}$$

This returns multisets rather than sets since different transitions may have the same label. For the example above, this would yield the multiset $\lambda_{\Omega_0}(\{w_1, w_3\}) = \{f, d\}$, which is also here, by chance, a set. Finally, we need to observe that although each derivation (12) is underpinned by the derivations (13) for the boxes in Σ, not all possible moves of the Σ_v's do correspond to a derivation captured by (12). For example, if we again take the running example, then

$$\Sigma_{v_1} \left[\{t_{11}\}\right\rangle \Sigma'_{v_1} \quad , \quad \Sigma_{v_2} \left[\emptyset\right\rangle \Sigma'_{v_2} = \Sigma_{v_2} \quad \text{and} \quad \Sigma_{v_3} \left[\emptyset\right\rangle \Sigma'_{v_3} = \Sigma_{v_3}.$$

Yet there is no move $(\Omega_0 \ : \ \Sigma) \xrightarrow{U'} (\Omega_0 \ : \ \Sigma')$ corresponding to it because there is no transition w in $\mathsf{T}^{v_1}_{\mathsf{new}}$ such that $\mathsf{trees}(w) = \{t_{11}\}$. This is due to the fact that the ρ of figure 4 leads to a synchronisation of t_{11} and t_{12}.

4.1 Soundness

What now follows is the (easier) half of the SOS rule for boxes.

Theorem 4.1. Let Σ be a tuple in the domain of an operator box Ω and

$$\Big(\Omega \ : \ \Sigma\Big) \xrightarrow{U} \Big(\Omega \ : \ \Theta\Big).$$

Then Θ is also a tuple in the domain of Ω and $\Omega(\Sigma)\ [U\rangle\ \Omega(\Theta)$.

Proof: Let μ be the factorisation of Σ and $V^{\delta\delta'} = \{v \in \mu_\delta \mid \Theta_v \in \mathsf{Box}^{\delta'}\}$, for $\delta, \delta' \in \{e, d, x\}$. Then

$$\mu' = \Big(\mu_e \backslash (V^{ed} \cup V^{ex}) \cup (V^{de} \cup V^{xe}) \ , \ \mu_d \backslash (V^{de} \cup V^{dx}) \cup (V^{ed} \cup V^{xd}) \ , \ \mu_x \backslash (V^{xe} \cup V^{xd}) \cup (V^{ex} \cup V^{dx}) \ , \ \mu_s\Big)$$

is the factorisation of Θ. Moreover, by multiple applications of all parts of proposition 2.2(3) we obtain that $\mu' \in \mathsf{fact}_\Omega$ (note that $V^{ed} \cup V^{ex} \cup V^{de} \cup V^{dx} \cup V^{xe} \cup V^{xd}$ is a finite set since U is finite, and if $v \in V^{xe} \cup V^{xd}$ then Σ_v is non-x-directed and therefore, by (Dom2), v is reversible). We also observe that (Dom1) and (Dom2) are satisfied by Θ since they were satisfied by Σ. Hence $\Theta \in \mathsf{dom}_\Omega$.

Let $U_v = U \cap \mathsf{T}^{v}_{\mathsf{new}}$, for every $v \in T_\Omega$ (recall that both U_v and each $\mathsf{trees}(u)$, for $u \in U_v$, is a set). It follows from the definition of $\xrightarrow{U}$ and the enabling and execution rules, that there is an Ω–tuple of boxes

Δ such that $\lfloor\Delta\rfloor = \lfloor\Sigma\rfloor$ and, for every $v \in T_\Omega$ and every $q \in S_{\Sigma_v}$,

$$M_{\Sigma_v}(q) = M_{\Delta_v}(q) + \sum_{u \in U_v} \sum_{x \in \mathsf{trees}(u)} W_{\Sigma_v}(q, x) \tag{15}$$

$$M_{\Theta_v}(q) = M_{\Delta_v}(q) + \sum_{u \in U_v} \sum_{x \in \mathsf{trees}(u)} W_{\Sigma_v}(x, q). \tag{16}$$

(To see this, simply subtract one equation from the other.) We then observe that, since $M_{\Delta_v} \subseteq M_{\Sigma_v}$ for every $v \in T_\Omega$, and $\Omega(\Sigma)$ is defined, $\Omega(\Delta)$ is a well-defined application of net refinement (although it is not necessarily the case that $\Delta \in \mathsf{dom}_\Omega$, since it could happen, for instance, that $M_{\Delta_v} = \emptyset$ while $v \in \mu_d$). Moreover, for every $p \in S_{\Omega(\Sigma)}$,

$$M_{\Omega(\Sigma)}(p) = M_{\Omega(\Delta)}(p) + \sum_{u \in U} W_{\Omega(\Sigma)}(p, u) \tag{17}$$

$$M_{\Omega(\Theta)}(p) = M_{\Omega(\Delta)}(p) + \sum_{u \in U} W_{\Omega(\Sigma)}(u, p). \tag{18}$$

We only show that (17) holds, the proof for (18) being similar. For $p = v \lhd q \in \mathsf{ST}^v_{\mathsf{new}}$ we have, by $\mathsf{trees}^v(p) = \{v\}$ and (2,4,15) and $\mathsf{trees}(u)$ being sets:

$$\begin{aligned} M_{\Omega(\Sigma)}(p) &= M_{\Sigma_v}(q) \\ &= M_{\Delta_v}(q) + \sum_{u \in U_v} \sum_{x \in \mathsf{trees}(u)} W_{\Sigma_v}(q, x) \\ &= M_{\Delta_v}(q) + \sum_{u \in U_v} W_{\Omega(\Sigma)}(p, u) \\ &= M_{\Omega(\Delta)}(p) + \sum_{u \in U_v} W_{\Omega(\Sigma)}(p, u). \end{aligned}$$

We then observe that from $\mathsf{trees}^z(p) = \emptyset$ (for $z \neq v$) and (4) it follows that $W_{\Omega(\Sigma)}(p, u) = 0$, for every $u \in U \backslash U_v$, so (17) holds for this p. For $p = s \lhd (\{v \lhd x_v\}_{v \in {}^\bullet s} + \{w \lhd e_w\}_{w \in s^\bullet}) \in \mathsf{SP}^s_{\mathsf{new}}$ we proceed as follows:

$$\begin{aligned} M_{\Omega(\Sigma)}(p) &= \sum_{v \in {}^\bullet s} M_{\Sigma_v}(x_v) + \sum_{w \in s^\bullet} M_{\Sigma_w}(e_w) \\ &= \sum_{v \in {}^\bullet s} M_{\Delta_v}(x_v) \ + \sum_{v \in {}^\bullet s} \sum_{u \in U_v} \sum_{x \in \mathsf{trees}(u)} W_{\Sigma_v}(x_v, x) \ + \\ &\quad \sum_{w \in s^\bullet} M_{\Delta_w}(e_w) \ + \sum_{w \in s^\bullet} \sum_{u \in U_w} \sum_{x \in \mathsf{trees}(u)} W_{\Sigma_w}(e_w, x) \\ &= M_{\Omega(\Delta)}(p) \ + \sum_{h \in {}^\bullet s \cup s^\bullet} \sum_{u \in U_h} \sum_{z \in \mathsf{trees}^h(p)} \sum_{x \in \mathsf{trees}(u)} W_{\Sigma_h}(z, x) \\ &= M_{\Omega(\Delta)}(p) \ + \sum_{h \in {}^\bullet s \cup s^\bullet} \sum_{u \in U_h} W_{\Omega(\Sigma)}(p, u) \end{aligned}$$

where the first equality follows from (2) and $M_\Omega(s) = 0$; the second from (15); the third from (2) and $M_\Omega(s) = 0$ and $x_v \neq e_w$ (for $v = w \in {}^\bullet s \cup s^\bullet$); and the last from (4) and $\mathsf{trees}(u)$ being sets. We then observe that from $\mathsf{trees}^z(p) = \emptyset$ (for $z \notin {}^\bullet s \cup s^\bullet$) and (4) it follows that $W_{\Omega(\Sigma)}(p, u) = 0$, for every $u \in U \backslash \bigcup_{h \in {}^\bullet s \cup s^\bullet} U_h$, so (17) holds also for this p.

Since U is finite, by (17), it is enabled at $M_{\Omega(\Sigma)}$. Let $\mathit{\Delta}$ be the box such that $\Omega(\Sigma)\ [U\rangle\ \mathit{\Delta}$. Then, by (17), (18) and U being a set, for every $p \in S_{\mathit{\Delta}} = S_{\Omega(\Sigma)} = S_{\Omega(\Theta)}$,

$$M_{\mathit{\Delta}}(p) \ = \ M_{\Omega(\Sigma)}(p) - \sum_{u \in U} W_{\Omega(\Sigma)}(p, u) + \sum_{u \in U} W_{\Omega(\Sigma)}(u, p) \ = \ M_{\Omega(\Theta)}(p).$$

This and $\lfloor\Omega(\Theta)\rfloor = \lfloor\mathit{\Delta}\rfloor$ means that $\Omega(\Theta) = \mathit{\Delta}$, which completes the proof. $\square$

4.2 Similarity relation on tuples of boxes

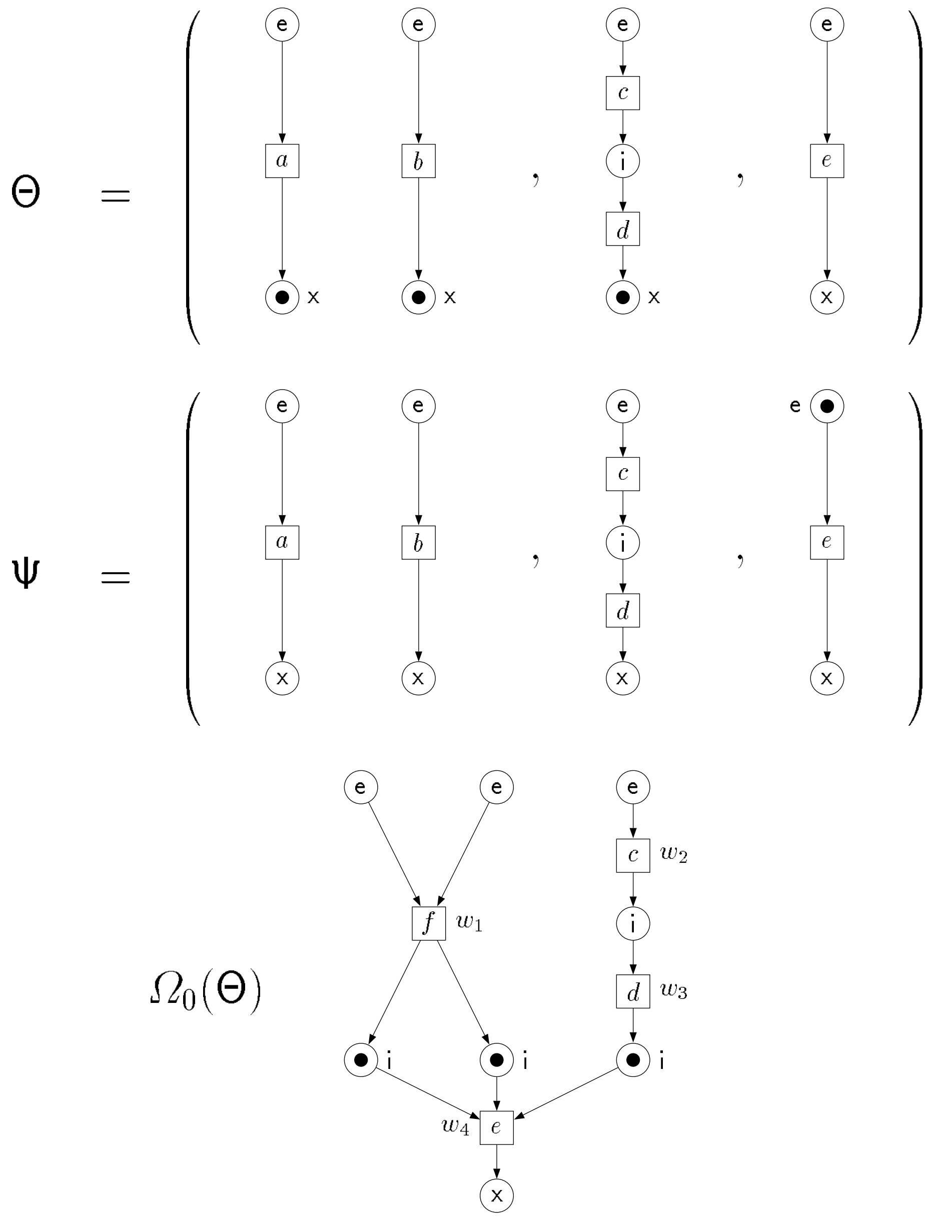


Figure 5. Another application of Ω_0.

A direct converse of theorem 4.1 does not in general hold true. For consider the tuple of boxes Θ in figure 5 and the box $\Omega_0(\Theta)$, shown in figure 5. We have $\Omega_0(\Theta)\ [\{w_4\}\rangle$, yet no nonempty move of the form

$$\left(\Omega_0\ :\ \Theta\right) \xrightarrow{U} \left(\Omega_0\ :\ \Theta'\right)$$

is possible. This is so because, when composing the nets, the tokens contributed by Θ_1 and Θ_2 are inserted into the composed net in such a way that they could all have been contributed by Θ_3 as well. More precisely, we have $\Omega(\Theta) = \Omega(\Psi)$, where Ψ is the tuple of boxes depicted in figure 5. And the situation is now different since a move

$$\left(\Omega_0 \ : \ \Psi\right) \xrightarrow{\{w_4\}} \left(\Omega_0 \ : \ \Psi'\right)$$

is possible with $\Psi'_1 = \Psi_1$, $\Psi'_2 = \Psi_2$ and $\Psi'_3 = \underline{\Psi_3}$. A conclusion to be drawn from this example is that the markings in a tuple of boxes Σ may need to be rearranged before attempting to derive a move which is admitted by their composition $\Omega(\Sigma)$. Such a rearrangement is formalised by a similarity relation $\equiv_\Omega$ on Ω–tuples of boxes defined next.

Let Ω be an operator box and Σ and Θ be Ω–tuples of static and dynamic boxes whose factorisations are respectively μ and κ. Then $\Sigma \equiv_\Omega \Theta$ if μ and κ are factorisations of the same complex marking of Ω and $\lfloor \Sigma \rfloor = \lfloor \Theta \rfloor$ and $\Sigma_v = \Theta_v$, for every $v \in \mu_d = \kappa_d$. The last requirement is added because, as we will see, the rearrangement of tokens concerns only the complete extry and exit interfaces of the refining nets. It is clear that $\equiv_\Omega$ is an equivalence relation. We can further strengthen this by showing that it is closed in the domain of application of Ω, and that it relates tuples which yield the same boxes through refinement.

Proposition 4.2. If Σ is a tuple in the domain of an operator box Ω and $\Sigma \equiv_\Omega \Theta$, then Θ is also a tuple in the domain of Ω and $\Omega(\Sigma) = \Omega(\Theta)$. Moreover, if Σ and Θ are tuples in the domain of Ω such that $\lfloor \Sigma \rfloor = \lfloor \Theta \rfloor$ and $\Omega(\Sigma) = \Omega(\Theta)$, then $\Sigma \equiv_\Omega \Theta$.
Proof: Let μ and κ be respectively the factorisations of Σ and Θ.

By the definition of $\equiv_\Omega$, $\lfloor \Sigma \rfloor = \lfloor \Theta \rfloor$, which implies that Θ satisfies (Dom1) and (Dom2) since Σ satisfies both conditions and $\Sigma_v = \Theta_v$, for every $v \in \mu_d = \kappa_d$. Hence $\Theta \in \mathsf{dom}_\Omega$. The claim that $\Omega(\Sigma) = \Omega(\Theta)$ follows from $\lfloor \Omega(\Sigma) \rfloor = \Omega(\lfloor \Sigma \rfloor) = \Omega(\lfloor \Theta \rfloor) = \lfloor \Omega(\Theta) \rfloor$, the fact that μ and κ are factorisations of the same marking of Ω and (11). The second part can be shown thus.

Let $R_v^e = {}^\circ\Sigma_v = {}^\circ\Theta_v$ and $R_v^x = \Sigma_v^\circ = \Theta_v^\circ$ and $R_v = \ddot{\Sigma}_v = \ddot{\Theta}_v$, for every $v \in T_\Omega$. We first observe that, for every $v \in T_\Omega$,

$$M_{\Sigma_v} \cap R_v \ = \ M_{\Theta_v} \cap R_v . \tag{19}$$

Indeed, let $q \in R_v$. By proposition 3.4(1), there is $p \in S_{\Omega(\Sigma)} = S_{\Omega(\Theta)}$ such that $\mathsf{trees}^v(p) = \{q\}$ and so, by (11) and q being an internal place, $M_{\Sigma_v}(q) = M_{\Omega(\Sigma)}(p) = M_{\Omega(\Theta)}(p) = M_{\Theta_v}(q)$. We next show that for all $v \in T_\Omega$ and $\delta \in \{e, x\}$,

$$(M_{\Sigma_v} \backslash M_{\Theta_v}) \cap R_v^\delta \neq \emptyset \implies v \in \mu_\delta . \tag{20}$$

Suppose that $\delta = e$ (the proof is symmetric for $\delta = x$). Assume that $q \in (M_{\Sigma_v} \backslash M_{\Theta_v}) \cap R_v^e \neq \emptyset$ (which means that $v \in \mu_e \cup \mu_d$) and $v \notin \mu_e$. The latter means that there is $q' \in {}^\circ\Sigma_v$ such that $M_{\Sigma_v}(q') = 0$. Take any $s \in {}^\bullet v$ (notice that $\mathsf{SP}^s_{\mathsf{new}} \subseteq \mathsf{mar}_\mu(v)$). If $s \notin v^\bullet$ then define $Z = \{q\}$ and $Z' = \{q'\}$. If $s \in v^\bullet$ then define $Z = \{q, r\}$ and $Z' = \{q', r\}$ where r is any place in Σ_v° (note that since Σ_v is ex-exclusive in such a case, we have $M_{\Sigma_v}(r) = 0$).
By proposition 3.4(2,4), there is $p \in \mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p) = Z$. Moreover, by (2), $p \in M_{\Omega(\Sigma)}$. Thus, by $\Omega(\Sigma) = \Omega(\Theta)$ and (8) and lemma 3.10, there must be a transition $w \in {}^\bullet s \cup s^\bullet$ such that $p \in \mathsf{SP}^s_{\mathsf{new}} \subseteq \mathsf{mar}_\kappa(w)$. We now consider two cases:

Case 1: $w = v$. Then there is $p' \in \mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p') = Z'$. We then observe that $M_{\Omega(\Sigma)}(p') = 0$. All these leads, by lemma 3.10, to $Tot(M_{\Sigma_v}, Z) = Tot(M_{\Theta_v}, Z) = 1$ and $Tot(M_{\Sigma_v}, Z') = Tot(M_{\Theta_v}, Z') = 0$ which is in contradiction with $M_{\Sigma_v}(q) = 1$ and $M_{\Sigma_v}(q') = M_{\Theta_v}(q) = 0$, as one can easily verify.

Case 2: $w \neq v$. Then there is $p' \in \mathsf{SP}^s_{\mathsf{new}}$ such that $\mathsf{trees}^v(p') = Z'$ and $\mathsf{trees}^w(p') = \mathsf{trees}^w(p)$. By lemma 3.10, $M_{\Omega(\Theta)}(p') = M_{\Omega(\Theta)}(p)$. Yet $M_{\Omega(\Sigma)}(p) = 1 \neq 0 = M_{\Omega(\Sigma)}(p')$, a contradiction with $\Omega(\Sigma) = \Omega(\Theta)$.
Thus (20) holds.

Suppose now that $v \in \mu_d$ and $M_{\Sigma_v} \neq M_{\Theta_v}$. We have, by (19), $M_{\Sigma_v} \cap R_v = M_{\Theta_v} \cap R_v$. Moreover, if

$$(M_{\Sigma_v} \backslash M_{\Theta_v}) \cap (R_v^e \cup R_v^x) \neq \emptyset$$

then, by (20), $v \in \mu_e \cup \mu_x$, contradicting $v \in \mu_d$. We thus have $M_{\Sigma_v} \subseteq M_{\Theta_v}$ and $M_{\Sigma_v} \cap R_v = M_{\Theta_v} \cap R_v$ and $M_{\Sigma_v} \neq M_{\Theta_v}$, so it must be the case that, for some $\delta \in \{e, x\}$,

$$(M_{\Theta_v} \backslash M_{\Sigma_v}) \cap R_v^\delta \neq \emptyset .$$

Hence, by the symmetric version of (20) (which clearly holds), we obtain $v \in \kappa_\delta$. If $\delta = e$ then, by (11), $\mathsf{SP}_{\mathsf{new}}^{(\bullet v)} \subseteq M_{\Omega(\Theta)} = M_{\Omega(\Sigma)}$. This and lemma 3.11 means that ${}^\bullet v \subseteq {}^\bullet\mu_e \cup \mu_x^\bullet$, contradicting $v \in \mu_d$ and proposition 2.2(4). If $\delta = x$ then, by (11), $\mathsf{SP}_{\mathsf{new}}^{(v^\bullet)} \subseteq M_{\Omega(\Theta)} = M_{\Omega(\Sigma)}$ which, again by lemma 3.11, means that $v^\bullet \subseteq {}^\bullet\mu_e \cup \mu_x^\bullet$, contradicting $v \in \mu_d$ and proposition 2.2(4). Hence $\Sigma_v = \Theta_v$ and so also $v \in \kappa_d$. In a similar way, we can show that $v \in \kappa_d$ implies $v \in \mu_d$ and $\Sigma_v = \Theta_v$. Hence $\mu_d = \kappa_d$ and $\Sigma_v = \Theta_v$, for every $v \in \mu_d = \kappa_d$.

To complete the proof of $\Sigma \equiv_\Omega \Theta$, we still need to show that ${}^\bullet\mu_e \cup \mu_x^\bullet = {}^\bullet\kappa_e \cup \kappa_x^\bullet$. This, however, follows directly from lemma 3.11 and $\Omega(\Sigma) = \Omega(\Theta)$. □

The condition $\lfloor\Sigma\rfloor = \lfloor\Theta\rfloor$ in the second part of the last result cannot be omitted. For example, with our running example in figure 4, if we take Σ' to be Σ_{v_1} with the label of t_{11} changed to b, and Σ'' to be Σ_{v_1} with the label of t_{12} changed to a, then $\Omega_0(\Sigma', \Sigma_{v_2}, \Sigma_{v_3}) = \Omega_0(\Sigma'', \Sigma_{v_2}, \Sigma_{v_3})$, yet $(\Sigma', \Sigma_{v_2}, \Sigma_{v_3}) \equiv_\Omega (\Sigma'', \Sigma_{v_2}, \Sigma_{v_3})$ clearly does not hold.

4.3 Completeness

We are now ready to prove the second half of the SOS rule for boxes. Together with theorem 4.1, this will mean that for the class of operator boxes, the standard step sequence semantics of compositionally defined nets obeys a variant of the SOS rule introduced originally for process algebras. The result is preceded by an auxiliary lemma which provides a characterisation of the enabling relation for the transitions in a refined net. Below, Σ is a tuple in the domain of an operator box Ω, and μ is the factorisation of Σ.

Lemma 4.3. Let v be a transition in Ω and $U = \{u_1, \ldots, u_k\} \subseteq \mathsf{T}_{\mathsf{new}}^v$ be a nonempty set of transitions which is enabled in $\Omega(\Sigma)$. For every $l \leq k$, let $\mathsf{trees}(u_l) = \{x_{l1}, \ldots, x_{lm_l}\}$. Moreover, let $X = \{x_{11}, \ldots, x_{1m_1}, \ldots, x_{k1}, \ldots, x_{km_k}\}$.

(1) For all $x \in X$ and $q \in {}^\bullet x$, $W_{\Sigma_v}(q, x) = 1$.
(2) For all $x_{li}, x_{hj} \in X$, if $(l, i) \neq (h, j)$ then ${}^\bullet x_{li} \cap {}^\bullet x_{hj} = \emptyset$ and $x_{li} \neq x_{hj}$.
(3) If $v \in \mu_d$ then ${}^\bullet X \subseteq M_{\Sigma_v}$.
(4) If $v \in \mu_e \cup \mu_s \cup \mu_x$ then either ${}^\bullet X \subseteq {}^\circ\Sigma_v$ and $\mathsf{SP}_{\mathsf{new}}^{(\bullet v)} \subseteq M_{\Omega(\Sigma)}$, or ${}^\bullet X \subseteq \Sigma_v^\circ$ and $\mathsf{SP}_{\mathsf{new}}^{(v^\bullet)} \subseteq M_{\Omega(\Sigma)}$.

Proof: (1,2) We observe that if (1) does not hold for $x = x_{li}$, or if ${}^\bullet x_{li} \cap {}^\bullet x_{hj} = \emptyset$ (in (2)) does not hold for $l = h$ then, by the definition of net refinement, there is a place p in $\Omega(\Sigma)$ such that $W_{\Omega(\Sigma)}(p, u_l) \geq 2$, contradicting the safeness of $M_{\Omega(\Sigma)}$ (see theorem 3.12) and $\{u_l\}$ being enabled in $\Omega(\Sigma)$. For $l \neq h$ in (2), we also have ${}^\bullet x_{li} \cap {}^\bullet x_{hj} = \emptyset$ since $M_{\Omega(\Sigma)}$ is safe, U is enabled in $\Omega(\Sigma)$, and (the first part of) proposition 3.7 holds. Moreover, in (2), $x_{li} \neq x_{hj}$ follows from ${}^\bullet x_{li} \cap {}^\bullet x_{hj} = \emptyset$ (which we have already shown) and the T-restrictedness of Σ_v.

Before proving the rest of the proposition, we observe that

$$q \in \ddot{\Sigma}_v \cap {}^\bullet X \implies M_{\Sigma_v}(q) > 0 \tag{21}$$

which follows from the fact that for such a q, by proposition 3.6 and lemma 3.10, we have $p = v \lhd q \in {}^\bullet U$ and $M_{\Omega(\Sigma)}(p) = M_{\Sigma_v}(q)$.

(3) By (21), we only need to show that if $q \in {}^\bullet X \cap ({}^\circ\Sigma_v \cup \Sigma_v^\circ)$ then $M_{\Sigma_v}(q) > 0$. To the contrary, assume that $M_{\Sigma_v}(q) = 0$ and, without loss of generality, that $q \in {}^\circ\Sigma_v$. Then, since $v \in \mu_d$ and Σ_v is clean, there must be $r \in \Sigma_v^\circ$ such that $M_{\Sigma_v}(r) = 0$. It now follows from proposition 3.4(4) that there is $p \in \mathsf{mar}_\mu(v)$ such that $\mathsf{trees}^v(p) = \{q, r\}$. We then notice that, by (8) and lemma 3.10, $M_{\Omega(\Sigma)}(p) = 0$. Moreover, by proposition 3.6, $p \in {}^\bullet U$, producing a contradiction with U being enabled in $\Omega(\Sigma)$.

(4) By (21), we have ${}^\bullet X \subseteq {}^\circ\Sigma_v \cup \Sigma_v^\circ$. We will first show that ${}^\bullet X \cap {}^\circ\Sigma_v \neq \emptyset$ implies $\mathsf{SP}_{\mathsf{new}}^{(\bullet v)} \subseteq M_{\Omega(\Sigma)}$. By lemma 3.11, the latter always holds if $v \in \mu_e$, so we may assume that $v \in \mu_s \cup \mu_x$ and $q \in {}^\bullet X \cap {}^\circ\Sigma_v$. To the contrary, suppose that $M_{\Omega(\Sigma)}(p) = 0$, for some $s \in {}^\bullet v$ and $p \in \mathsf{SP}_{\mathsf{new}}^s$. Let $q' \in {}^\circ\Sigma_v$ be such that $q' \in \mathsf{trees}^v(p)$ (note that it may happen that $q = q'$, but it is always the case that $0 = M_{\Sigma_v}(q) \leq M_{\Sigma_v}(q')$). Then, by the definition of net refinement, there is $p' \in \mathsf{SP}_{\mathsf{new}}^s$ such that $\mathsf{trees}^v(p') = \mathsf{trees}^v(p) \setminus \{q'\} \cup \{q\}$ and $\mathsf{trees}^w(p') = \mathsf{trees}^w(p)$, for all $w \neq v$. Thus, by (2), $M_{\Omega(\Sigma)}(p') \leq M_{\Omega(\Sigma)}(p) = 0$. This, however, contradicts U being enabled in $\Omega(\Sigma)$ and $p' \in {}^\bullet U$ (by (4)).
In a similar way, we may show that if ${}^\bullet X \cap \Sigma_v^\circ \neq \emptyset$ then $\mathsf{SP}_{\mathsf{new}}^{(v^\bullet)} \subseteq M_{\Omega(\Sigma)}$.

Thus we only need to prove is that it is impossible to have ${}^\bullet X \cap {}^\circ\Sigma_v \neq \emptyset \neq {}^\bullet X \cap \Sigma_v^\circ$. Indeed, if this was the case, then by what we have already shown, $\mathsf{SP}_{\mathsf{new}}^{({}^\bullet v \cup v^\bullet)} \subseteq M_{\Omega(\Sigma)}$ which, by lemma 3.11, would mean that ${}^\bullet v \cup v^\bullet \subseteq {}^\bullet\mu_e \cup \mu_x^\bullet$. Moreover, ${}^\bullet v \cap v^\bullet = \emptyset$ since otherwise, by proposition 3.4(4) and (4), there would be $p \in \mathsf{SP}_{\mathsf{new}}^{({}^\bullet v \cap v^\bullet)}$ such that $\sum_{u\in U} W_{\Omega(\Sigma)}(p,u) \geq 2$, contradicting U being enabled at the safe marking $M_{\Omega(\Sigma)}$. Thus, since Ω is a box (and so is safe) we obtain that $v^\bullet = \emptyset$, a contradiction with Ω being T-restricted. □

The above lemma characterises two cases where a step U derived from the net refining a transition v can be executed in $\Omega(\Sigma)$. The first is that the step is made possible by the marking of Σ_v and thus can be deduced using the rule (12); this is captured by part (3), and by part (4) with (${}^\bullet X \subseteq {}^\circ\Sigma_v \ \wedge\ v \in \mu_e$) or (${}^\bullet X \subseteq \Sigma_v^\circ \ \wedge\ v \in \mu_x$). The second case characterises those situations when (12) cannot be used; this is captured by part (4) with $v \in \mu_s$ or (${}^\bullet X \subseteq {}^\circ\Sigma_v \ \wedge\ v \in \mu_x$) or (${}^\bullet X \subseteq \Sigma_v^\circ \ \wedge\ v \in \mu_e$). Notice that it applies to the tuple Θ in figure 5 and $U = \{w_4\} \subseteq \mathsf{T}_{\mathsf{new}}^{v_3}$. Indeed, U is a step enabled in $\Omega_0(\Theta)$ (see figure 5), $\Theta_{v_3} \in \mathsf{Box}^s$, ${}^\bullet\mathsf{trees}(w_4) = \{q_{31}\} \subseteq {}^\circ\Theta_{v_3}$ and

$$\mathsf{SP}_{\mathsf{new}}^{({}^\bullet v_3)} = \mathsf{SP}_{\mathsf{new}}^{s_3} \cup \mathsf{SP}_{\mathsf{new}}^{s_4} = \{p_3, p_4, p_7\} = M_{\Omega_0(\Theta)}.$$

Theorem 4.4. Let $\Omega(\Sigma)\ [U\rangle\ \Sigma'$. Then there are Θ and Ψ in dom_Ω such that $\Sigma \equiv_\Omega \Theta$, $\Omega(\Psi) = \Sigma'$ and

$$\Big(\Omega \ :\ \Theta\Big) \xrightarrow{U} \Big(\Omega \ :\ \Psi\Big).$$

Proof: For every $v \in V$, let $U_v = U \cap \mathsf{T}_{\mathsf{new}}^v$ and $X_v = \bigcup_{u\in U_v} \mathsf{trees}(u)$. Moreover, let $V = \{v \mid U_v \neq \emptyset\}$ and (M, Q) be the marking of Ω whose factorisation is the factorisation μ of Σ. Finally, for $\delta \in \{e, x, s\}$, let $Y_\delta = \{v \in V \cap \mu_\delta \mid {}^\bullet X_v \subseteq {}^\circ\Sigma_v\}$ and $Z_\delta = \{v \in V \cap \mu_\delta \mid {}^\bullet X_v \subseteq \Sigma_v{}^\circ\}$. From lemma 4.3(3,4), proposition 3.8, and U being enabled in the safe marking $M_{\Omega(\Sigma)}$, it follows that

$$R \;=\; \biguplus_{v\in Y_e} {}^\bullet v \ \uplus\ \biguplus_{v\in Y_x} {}^\bullet v \ \uplus\ \biguplus_{v\in Y_s} {}^\bullet v \ \uplus\ \biguplus_{v\in Z_e} v^\bullet \ \uplus\ \biguplus_{v\in Z_x} v^\bullet \ \uplus\ \biguplus_{v\in Z_s} v^\bullet \ \subseteq\ M. \tag{22}$$

Hence, by proposition 2.2(2) and (Dom2) (where the latter implies that each transition $v \in Z_e \cup Z_x \cup Z_s$ is reversible which guarantees that $(M \setminus v^\bullet, Q \cup \{v\})$ is a marking reachable from ${}^\circ\Omega$ or Ω°) we have that

$$(M', Q') \;=\; (M \setminus R\ ,\ Q \cup Y_e \cup Y_x \cup Y_s \cup Z_e \cup Z_x \cup Z_s)$$

is a marking reachable from ${}^\circ\Omega$ or Ω°. Thus there is a factorisation $\xi \in \mathsf{fact}_\Omega$ of (M', Q'). It is then easy to see that

$$\kappa \;=\; (\xi_e \cup Y_e \cup Y_x \cup Y_s\ ,\ \mu_d\ ,\ \xi_x \cup Z_e \cup Z_x \cup Z_s\ ,\ \xi_s)$$

is a valid factorisation of (M, Q). Let Θ be the Ω–tuple of boxes such that $\lfloor\Theta\rfloor = \lfloor\Sigma\rfloor$, κ is the factorisation of Θ and, for every $v \in \kappa_d = \mu_d$, $\Theta_v = \Sigma_v$. Then, by lemma 4.3(1,2,3), we have that

$$\Big(\Omega \ :\ \Theta\Big) \xrightarrow{U} \Big(\Omega \ :\ \Psi\Big)$$

for some Ψ. Thus, by theorem 4.1, $\Omega(\Theta)\ [U\rangle\ \Omega(\Psi)$ and, by proposition 4.2, $\Omega(\Theta) = \Omega(\Sigma)$. As a result, $\Sigma' = \Omega(\Psi)$. □

Various important consequences may be derived from theorems 4.1 and 4.4. In particular,

Theorem 4.5. $\Omega(\Sigma)$ is a static or dynamic box.
Proof: By theorem 3.12, to show that all the markings reachable from $M_{\Omega(\Sigma)}$ are safe and clean, it suffices to show that if U is a step such that $\Omega(\Sigma)\ [U\rangle\ \Sigma'$ then there is a tuple Ψ in the domain of Ω such that $\Sigma' = \Omega(\Psi)$, which holds by theorem 4.4 and proposition 4.2. To deal with the markings reachable from ${}^\circ\Omega(\Sigma)$ or $\Omega(\Sigma)^\circ$, it suffices to observe that:

$$\exists \Theta, \Theta' \in \mathsf{dom}_\Omega\ :\ \Omega(\Theta) = \overline{\Omega(\Sigma)}\ \wedge\ \Omega(\Theta') = \underline{\Omega(\Sigma)}. \tag{23}$$

To show that such a Θ does indeed exist, we first observe that, since Ω is an operator box and proposition 2.2(5) holds, there is a factorisation μ of the entry marking such that ${}^\bullet\mu_e \cup \mu_x^\bullet = {}^\circ\Omega$ and $\mu_d = \emptyset$. Define Θ as the Ω–tuple of boxes such that $\lfloor\Theta\rfloor = \lfloor\Sigma\rfloor$ and μ is the factorisation of Θ. It follows directly from (11) that $\Omega(\Theta) = \overline{\Omega(\Sigma)}$. The existence of Θ' can be shown similarly. □

And, by inspecting the last proof, we obtain

Corollary 4.6. Let Σ' be a net derivable from $\overline{\Omega(\Sigma)}$ or $\underline{\Omega(\Sigma)}$. Then there is a tuple Θ in the domain of Ω such that $\Sigma' = \Omega(\Theta)$. □

As a summary, sos-compositionality in the domain of nets can be rendered by a rule similar to that commonly formulated for process algebras:

$$\boxed{\frac{\big(\Omega : \Theta\big) \xrightarrow{U} \big(\Omega : \Psi\big)}{\Omega(\Sigma)\ \big[U\big\rangle\ \Sigma'} \quad \Sigma \equiv_\Omega \Theta\ ,\ \Omega(\Psi) = \Sigma'} \tag{24}$$

Theorem 4.1 together with proposition 4.2 verify its soundness, whereas theorem 4.4 verifies its completeness.

A useful corollary is a sufficient condition for a refinement to be ex-exclusive.

Theorem 4.7. Let Ω be an ex-exclusive operator box and $\Sigma \in \mathsf{dom}_\Omega$ be a tuple of boxes such that, for every $v \in T_\Omega$ satisfying ${}^\bullet v \cap {}^\circ\Omega \neq \emptyset \neq v^\bullet \cap \Omega^\circ$ or $v^\bullet \cap {}^\circ\Omega \neq \emptyset \neq {}^\bullet v \cap \Omega^\circ$, it is the case that Σ_v is ex-exclusive. Then $\Omega(\Sigma)$ is an ex-exclusive box.

Proof: Let (M, Q) be the marking of Ω with the same factorisation μ as Σ. From (23), theorem 4.4 and proposition 4.2, it follows that it suffices to show that ${}^\circ\Omega(\Sigma) \cap M_{\Omega(\Sigma)} = \emptyset$ or $\Omega(\Sigma)^\circ \cap M_{\Omega(\Sigma)} = \emptyset$. To the contrary, suppose that there are $p \in {}^\circ\Omega(\Sigma)$ and $p' \in \Omega(\Sigma)^\circ$ such that $p, p' \in M_{\Omega(\Sigma)}$. Then, by the definition of net refinement, there are $s \in {}^\circ\Omega$ and $s' \in \Omega^\circ$ such that $p \in \mathsf{SP}^s_{\mathsf{new}}$ and $p' \in \mathsf{SP}^{s'}_{\mathsf{new}}$. Moreover, by lemma 3.10, there are v and v' such that $p \in \mathsf{mar}_\mu(v)$ and $p' \in \mathsf{mar}_\mu(v')$. We now consider two cases:

Case 1: $v = v'$. If $v \in \mu_e \cup \mu_x$ then $s, s' \in {}^\bullet\mu_e \cup \mu_x^\bullet$, contradicting Ω being ex-exclusive. If $v \in \mu_d$ and $s, s' \in {}^\bullet v \ \vee\ s, s' \in v^\bullet$ then, by proposition 2.2(3), we again obtain a contradiction with Ω being ex-exclusive. If neither $s, s' \in {}^\bullet v$ nor $s, s' \in v^\bullet$, we may assume without loss of generality, that $s \in {}^\bullet v$ and $s' \in v^\bullet$. Let $q \in {}^\circ\Sigma_v$ and $q' \in \Sigma_v^\circ$ be such that $q \in \mathsf{trees}^v(p)$ and $q' \in \mathsf{trees}^v(p')$. Then, since neither $s, s' \in {}^\bullet v$ nor $s, s' \in v^\bullet$, $\{q\} = \mathsf{trees}^v(p)$ and $\{q'\} = \mathsf{trees}^v(p')$. Hence $q, q' \in M_{\Sigma_v}$, contradicting Σ_v being ex-exclusive.

Case 2: $v \neq v'$. Then, by proposition 2.2(2), there is a marking (M', Q') reachable from ${}^\circ\Omega$ or Ω° such that $s, s' \in M'$, contradicting Ω being ex-exclusive. □

5 Behavioural conditions

In the previous section, we have seen how the behaviour of a compositionally defined net can be computed from the behaviour of its components, and conversely. So far, however, this is unrelated to any process algebraic considerations, except that the shape of the SOS rule is inspired by process algebra theory. Starting with this section, we shall investigate relationships between operations on nets (incarnated by operator boxes) and process algebraic operators. In the present section, we address a behavioural consistency issue that pertains to such an investigation.

Consider the operator box Ω^* on the left hand side of figure 6, whose terminated behaviours are modelled by the expression $v_2^* v_1$ (in other words, any number of repetitions of the net refining v_2 followed by a behaviour of the net refining v_1). Allowing operator boxes like Ω^* raises the danger of creating composite nets which are not in agreement with an intuitive behavioural meaning of operations specified by operator boxes. For consider an expression such as $\Omega_{\Box}(\mathsf{N}_\alpha, \Omega^*(\mathsf{N}_\beta, \mathsf{N}_\gamma))$ (using the boxes in figure 7), whose behaviour is, intuitively: do either α, or $\beta^*\gamma$. According to our intuition about the operation specified by Ω^*, one might expect that the net corresponding to this expression could be constructed by first applying Ω^* to N_β and N_γ, and then putting the result in a choice with N_α, yielding the net shown on the right-hand side of figure 6. This net, however, allows evolutions such as $\{\beta\}\{\beta\}\{\alpha\}$, which do not correspond to what one expects from a choice construct. This phenomenon is not particular to our approach, nor to Petri nets in general (cf. [1, 13])

If we imposed ex-directedness on all boxes, then the problem would disappear. However, such a strict solution is not always desirable, because boxes such as Ω^* are useful in modelling guarded while-loops in programming languages (basically, v_1 corresponds to the negation of the guard(s), and v_2 to the repetitive behaviour). On the other hand, if it is ascertained that a loop does not occur initially in an enclosing choice, or in an enclosing loop, then the problem also disappears, and ex-directedness is no longer required. To our experience, in practical programming languages such a case (i.e., a loop occurring initially in enclosing loops or choices) does not arise, and we therefore formulate a set of conditions in such a way that it is excluded, and Ω^* still is an allowed operator box.

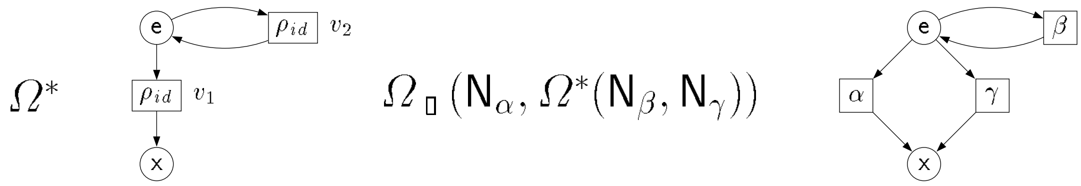


Figure 6. Operator box Ω^* and a net modelling (tentatively) $\Omega_{\Box}(\mathsf{N}_\alpha, \Omega^*(\mathsf{N}_\beta, \mathsf{N}_\gamma))$.

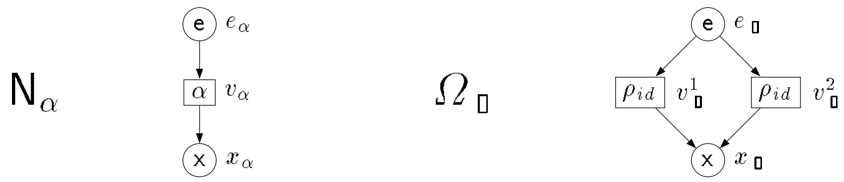


Figure 7. Plain box N_α and operator box for choice, $\Omega_{\Box}$.

In all, we formulate five additional conditions on an Ω–tuple Σ in the domain of application of an operator box Ω; below, v and w are transitions in Ω. The example discussed above will be excluded by (Beh1), when v and w are instantated by $v^1_{\Box}$ and $v^2_{\Box}$, respectively, since $\Omega^*(\mathsf{N}_\beta, \mathsf{N}_\gamma)$ is not e-directed.

- If $v \neq w$ and ${}^\bullet v \cap {}^\bullet w \neq \emptyset$, then $\Sigma_v, \Sigma_w \in \mathsf{Box}^{edir}$. (Beh1)
- If $v \neq w$ and $v^\bullet \cap w^\bullet \neq \emptyset$, then $\Sigma_v, \Sigma_w \in \mathsf{Box}^{xdir}$. (Beh2)
- If $v^\bullet \cap {}^\bullet w \neq \emptyset$, then $\Sigma_v \in \mathsf{Box}^{xdir}$ or $\Sigma_w \in \mathsf{Box}^{edir}$. (Beh3)
- If ${}^\bullet v \cap \Omega^\circ \neq \emptyset$, then $\Sigma_v \in \mathsf{Box}^{edir}$. (Beh4)
- If $v^\bullet \cap {}^\circ\Omega \neq \emptyset$, then $\Sigma_v \in \mathsf{Box}^{xdir}$. (Beh5)

In the rest of this section, we will argue that indeed, (Beh1)–(Beh5) guarantee that behaviours of a composite net are composed from behaviours of its components. It is limited to the (relevant) case of operator boxes with one single token, of which both $\Omega_{\Box}$ and Ω^* are special instances.

The appropriateness of the conditions (Beh1)–(Beh5) Let us consider an operator box Ω such that each marking reachable from ${}^\circ\Omega$ or Ω° has at most one token (in other words, we may assume that Ω is a *state machine* which means that $|{}^\circ\Omega| = |\Omega^\circ| = 1$ as well as $|{}^\bullet v| = |v^\bullet| = 1$, for every $v \in T_\Omega$) and that each transition of Ω is labelled by the identity relabelling, ρ_{id}. Furthermore, let $\Omega(\Sigma)$ be a valid application of Ω satisfying (Beh1)–(Beh5), and $U_1 \ldots U_k$ be a nonempty step sequence composed of nonempty sets of transitions such that

$$\overline{\Omega(\Sigma)}\ [U_1 \ldots U_k\rangle\ \Psi. \tag{25}$$

We aim at showing that the execution (25) can be seen as a combination of valid executions of Ω and the Σ_v's. To this end, we first observe that, by theorem 4.4 and (23), there are Ω-tuples $\Delta^1, \ldots, \Delta^k$ and $\Phi^1, \ldots, \Phi^k$ in dom_Ω such that $\lfloor \Delta^i \rfloor = \lfloor \Phi^i \rfloor = \lfloor \Sigma \rfloor$, for every $1 \leq i \leq k$, and

$$\forall 1 \leq i \leq k \ :\ \left(\Omega \ :\ \Delta^i\right) \xrightarrow{U_i} \left(\Omega \ :\ \Phi^i\right) \tag{26}$$

$$\forall 1 < i \leq k \;:\; \Omega(\Delta^i) \;=\; \Omega(\Phi^{i-1}) \tag{27}$$

$$\Omega(\Delta^1) \;=\; \overline{\Omega(\Sigma)} \quad\text{ and }\quad \Omega(\Phi^k) \;=\; \Psi. \tag{28}$$

Below, we will denote by μ^i and κ^i the factorisations of respectively Δ^i and Φ^i. Notice that, by (27) and proposition 4.2, for every $1 < i \leq k$, the μ^i and κ^{i-1} are factorisations of the same complex marking of Ω, $\mathcal{M}_i = (M_i, Q_i)$; moreover, we will denote by $\mathcal{M}_1 = (M_1, Q_1)$ and $\mathcal{M}_{k+1} = (M_{k+1}, Q_{k+1})$, the complex markings of Ω whose factorisations are respectively μ^1 and κ^k.

Our next observation is that since Ω is a state machine, for every $1 \leq i \leq k$, there is a *unique* transition $v_i \in T_\Omega$ which belongs to $\mu_e^i \cup \mu_d^i \cup \mu_x^i$. Hence, each marking $\mathcal{M}_i$, for $1 \leq i \leq k+1$, is such that

$$|M_i| + |Q_i| = 1. \tag{29}$$

Moreover, for every $1 \leq i \leq k$, there is a nonempty set of transitions $V_i \subseteq T_{\Sigma_{v_i}}$ such that

$$U_i \;=\; \{\; (v_i, \lambda_{\Sigma_{v_i}}(t)) \lhd t \;\mid\; t \in V_i \;\}.$$

The nonemptiness of V_i follows from $U_i \neq \emptyset$. We finally observe that, by $V_i \neq \emptyset$ and (26) and the T-restrictedness of boxes,

$$\forall 1 \leq i \leq k \;:\; (v_i \in \mu_x^i \Rightarrow \Sigma_{v_i} \notin \mathsf{Box}^{xdir}) \;\wedge\; (v_i \in \kappa_e^i \Rightarrow \Sigma_{v_i} \notin \mathsf{Box}^{edir}). \tag{30}$$

Lemma 5.1. With the above notations, the following hold.

(1) ${}^\bullet v_1 = {}^\circ\Omega$ and $\Delta_{v_1}^1 = \overline{\Sigma_{v_1}}$.
(2) For every $1 \leq i < k$, we have either

  (a) $v_i = v_{i+1}$ and $\Phi_{v_i}^i = \Delta_{v_{i+1}}^{i+1}$, or
  (b) $v_i^\bullet = {}^\bullet v_{i+1}$ and $\Phi_{v_i}^i = \underline{\Sigma_{v_i}}$ and $\Delta_{v_{i+1}}^{i+1} = \overline{\Sigma_{v_{i+1}}}$.[7]

(3) $\Psi = \underline{\Omega(\Sigma)}$ if and only if $\Phi_{v_k}^k = \underline{\Sigma_{v_k}}$ and $v_k^\bullet = \Omega^\circ$.

Proof: (1) We have $M_{\Omega(\Delta^1)} = \mathsf{SP}_{\mathsf{new}}^s$ where $\{s\} = {}^\circ\Omega$. Hence, by lemma 3.11 and (29), $\mathcal{M}_1 = (\{s\}, \emptyset)$ and $v_1 \in \mu_e^1 \cup \mu_x^1$. If $v_1 \in \mu_x^1$ then, by (30), $\Sigma_{v_1} \notin \mathsf{Box}^{xdir}$ which, together with $v_1^\bullet = \{s\}$, yields a contradiction with (Beh5).[8] Thus $v_1 \in \mu_e^1$ and ${}^\bullet v_1 = \{s\} = {}^\circ\Omega$.

(2) We consider two cases.

Case 1: $v_i \neq v_{i+1}$. Then μ^{i+1} and κ^i are two different factorisations of $\mathcal{M}_{i+1}$. Hence, by (29), there is $s \in S_\Omega$ such that $\mathcal{M}_{i+1} = (\{s\}, \emptyset)$, which in turn means that $v_i \in \kappa_e^i \cup \kappa_x^i$ and $v_{i+1} \in \mu_e^{i+1} \cup \mu_x^{i+1}$. We now consider four sub-cases.
Case 1(i): $v_i \in \kappa_e^i$ and $v_{i+1} \in \mu_e^{i+1}$. Then, by (30), $\Sigma_{v_i} \notin \mathsf{Box}^{edir}$. On the other hand, ${}^\bullet v_i = \{s\} = {}^\bullet v_{i+1}$, producing a contradiction with (Beh1).[9]
Case 1(ii): $v_i \in \kappa_x^i$ and $v_{i+1} \in \mu_x^{i+1}$. Then, by (30), $\Sigma_{v_{i+1}} \notin \mathsf{Box}^{edir}$. On the other hand, $v_i^\bullet = \{s\} = v_{i+1}^\bullet$, producing a contradiction with (Beh2).[10]
Case 1(iii): $v_i \in \kappa_e^i$ and $v_{i+1} \in \mu_x^{i+1}$. Then, by (30), $\Sigma_{v_i} \notin \mathsf{Box}^{edir}$ and $\Sigma_{v_{i+1}} \notin \mathsf{Box}^{xdir}$. On the other hand, ${}^\bullet v_i = \{s\} = v_{i+1}^\bullet$, producing a contradiction with (Beh3).[11]
Case 1(iv): $v_i \in \kappa_x^i$ and $v_{i+1} \in \mu_e^{i+1}$. Then $v_i^\bullet = \{s\} = {}^\bullet v_{i+1}$, and so (b) holds.

Case 2: $v_i = v_{i+1} (= v)$. If $\Phi_v^i = \Delta_v^{i+1}$ then (a) holds, so we assume that $\Phi_v^i \neq \Delta_v^{i+1}$. Then μ^{i+1} and κ^i are two different factorisations of $\mathcal{M}_{i+1}$. Hence, by (29), there is $s \in S_\Omega$ such that $\mathcal{M}_{i+1} = (\{s\}, \emptyset)$,

[7] We do not exclude the possibility that $v_i = v_{i+1}$.

[8] Thus (Beh5) excludes the situation whereby, from the entry marking of the composed net, we can execute a behaviour originating from $\Sigma_{v_1}^\circ$ which bears no relationship to any possible behaviour of Ω (this intuitively corresponds to the backward reachability in Ω, which is not allowed).

[9] Thus, with (Beh1), it is not possible to start the execution of Σ_{v_i} and later, without finishing it, to enter $\Sigma_{v_{i+1}}$ for a conflicting v_{i+1} because the initial state of Σ_{v_i} was reached.

[10] (Beh2) prohibits to finish the execution of Σ_{v_i} and afterwards to enter $\Sigma_{v_{i+1}}$ from the rear.

[11] (Beh3) excludes a situation, combining the previous two problems, where we start the execution of $\Sigma_{v_{i+1}}$ and later, without finishing it, come back to the initial state and (re)enter the (supposedly finished) Σ_{v_i} from the rear.

which in turn means that $v \in (\kappa_e^i \cup \kappa_x^i) \cap (\mu_e^{i+1} \cup \mu_x^{i+1})$ and ${}^\bullet v = \{s\} = v^\bullet$. Since $\Phi_v^i \neq \Delta_v^{i+1}$ we only need to consider two sub-cases.
Case 2(i): $v \in \kappa_e^i \cap \mu_x^{i+1}$. Then, by (30), $\Sigma_v \notin \mathsf{Box}^{edir} \cup \mathsf{Box}^{xdir}$. On the other hand, ${}^\bullet v = v^\bullet$, producing a contradiction with (Beh3).
Case 2(ii): $v \in \kappa_x^i \cap \mu_e^{i+1}$. This and ${}^\bullet v = v^\bullet$ means that (b) holds.

(3) If $\Psi = \underline{\Omega(\Sigma)}$ then $M_{\Omega(\Phi^k)} = \mathsf{SP}^s_{\mathsf{new}}$ where $\{s\} = \Omega^\circ$. Hence, by lemma 3.11 and (29), $\mathcal{M}_{k+1} = (\{s\}, \emptyset)$ and $v_k \in \kappa_e^k \cup \kappa_x^k$. If $v_k \in \kappa_e^k$ then, by (30), $\Sigma_{v_k} \notin \mathsf{Box}^{edir}$ which, together with ${}^\bullet v_k = \{s\}$, yields a contradiction with (Beh4).[12] Thus $v_k \in \kappa_x^k$ and $v_k^\bullet = \{s\} = \Omega^\circ$.
The reverse implication follows directly from (11). □

It is always possible to partition (in a unique way) the sequence of steps $V_1 \dots V_k$ into the longest nonempty sub-sequences $\sigma_1, \dots, \sigma_l$ so that $V_1 \dots V_k = \sigma_1 \dots \sigma_l$, and, for every $i \leq l$, if

$$\sigma_i \;=\; V_p V_{p+1} \dots V_q$$

then it is the case that $v_p = v_{p+1} = \dots = v_q (= w_i)$ and $\Phi_{w_i}^j = \Delta_{w_i}^{j+1}$, for every $p \leq j < q$. Finally,

Theorem 5.2. With the above notations, the following hold.

(1) For every $i < l$, $\overline{\Sigma_{w_i}}\;[\sigma_i\rangle\;\underline{\Sigma_{w_i}}$, and $\overline{\Sigma_{w_l}}\;[\sigma_l\rangle\;\Phi_{v_k}^k$.
(2) If $\Phi_{v_k}^k = \underline{\Sigma_{v_k}}$ then $\mathcal{M}_1\;[\{w_1\}\dots\{w_{l-1}\}\{w_l\}\rangle\;\mathcal{M}_{k+1}$; otherwise $\mathcal{M}_1\;[\{w_1\}\dots\{w_{l-1}\}\{w_l\}^+\rangle\;\mathcal{M}_{k+1}$.
(3) $\Phi = \underline{\Omega(\Sigma)}$ if and only if $\mathcal{M}_{k+1} = \Omega^\circ$.

Proof: Follows with lemma 5.1. □

This result shows that, as was claimed, the original sequence $U_1 \dots U_k$ can (even uniquely) be decomposed into a behaviour of Ω (i.e., the $\{w_1\}\dots$ in part (2) of the theorem), such that each w_i – except possibly the last one – corresponds to a full behaviour, σ_i, of Σ_{w_i} (cf. part (1) of the theorem). Part (3) of the theorem states that full behaviours of $\Omega(\Sigma)$ and full behaviours of Ω correspond to each other.

6 A process algebra and its semantics

We now introduce an algebra of process expressions, called the *box algebra*, which is based on the class of operator boxes defined in the previous section. The box algebra is in fact a *meta-model* parameterised by two non-empty, possibly infinite, sets of Petri nets: a set $\mathsf{ConstBox}$ of static and dynamic plain boxes providing a denotational semantics of simple process expressions, and a disjoint set OpBox of operator boxes providing interpretation for the connectives.

The only assumption made about the operator boxes in OpBox and the static boxes in $\mathsf{ConstBox}$ is that they have disjoint sets of simple root-only trees as their place and transition names, i.e., for all distinct static and/or operator boxes Σ and Θ in $\mathsf{OpBox} \cup \mathsf{ConstBox}$,

$$S_\Sigma \cup T_\Sigma \;\subseteq\; \mathsf{P_{root}} \cup \mathsf{T_{root}} \quad \text{and} \quad S_\Sigma \cap S_\Theta \;=\; T_\Sigma \cap T_\Theta \;=\; \emptyset. \tag{31}$$

The above will be useful in defining a global independence relation on transitions in nets associated with process expressions. We do not require that the boxes in OpBox and $\mathsf{ConstBox}$ be finite.

Signature We consider an algebra of process expressions over the signature

$$\mathsf{Const} \;\cup\; \left\{\overline{(.)}, \underline{(.)}\right\} \;\cup\; \left\{\mathsf{op}_\Omega \;\middle|\; \Omega \in \mathsf{OpBox}\right\} \tag{32}$$

where Const is a fixed non-empty set of *constants* which will be modelled through the boxes in $\mathsf{ConstBox}$, $\overline{(.)}$ and $\underline{(.)}$ are two unary operators, and each op_Ω is a connective of the algebra indexed by an operator

[12] (Beh4) excludes the situation where we reach the exit marking, start the execution of Σ_{v_i} and return to the exit marking without finishing Σ_{v_i}, so that a completed history is not composed out of completed histories of the components corresponding to a completed history of Ω.

box taken from the set OpBox.[13] The set of constants is partitioned into the *static* constants, Const^s, and *dynamic* constants, Const^d; moreover, there are two distinct disjoint subsets of Const^d, denoted by Const^e and Const^x, and respectively called the *entry* and *exit* constants. We also identify three further, not necessarily disjoint subsets of Const, namely: Const^{xcl}, Const^{edir} and Const^{xdir}, called the *ex-exclusive*, *e-directed* and *x-directed* constants, respectively.

Although we use the symbols $\overline{(.)}$ and $\underline{(.)}$ to denote both operations on boxes and process algebra connectives, it will always be clear from the context what is the intended interpretation.

Syntax There are four classes of process expressions corresponding to previously introduced classes of plain boxes: the *entry*, *dynamic*, *exit* and *static* expressions, denoted respectively by Expr^e, Expr^d, Expr^x and Expr^s. Collectively, we will refer to them as the *box expressions*, Expr^{box}. We will also use a counterpart of the notion of the factorisation of a tuple of boxes. For an operator box Ω and an Ω–tuple of box expressions D, we define the *factorisation* of D to be the quadruple $\mu = (\mu_e, \mu_d, \mu_x, \mu_s)$ such that $\mu_\delta = \{v \mid D_v \in \mathsf{Expr}^\delta\}$, for $\delta \in \{e, x, s\}$, and $\mu_d = \{v \mid D_v \in \mathsf{Expr}^d \backslash (\mathsf{Expr}^e \cup \mathsf{Expr}^x)\}$. The syntax for the box expressions Expr^{box} is given by:

$$
\begin{array}{llll|l|l|l}
\mathsf{Expr}^s & E & ::= & c^s & \mathsf{op}_\Omega(\mathsf{E}) & & \\
\mathsf{Expr}^e & F & ::= & c^e & \mathsf{op}_\Omega(\mathsf{F}) & \overline{E} & \\
\mathsf{Expr}^x & G & ::= & c^x & \mathsf{op}_\Omega(\mathsf{G}) & \underline{E} & \\
\mathsf{Expr}^d & H & ::= & c^d & \mathsf{op}_\Omega(\mathsf{H}) & F & G
\end{array}
\tag{33}
$$

where $c^\delta \in \mathsf{Const}^\delta$, for $\delta \in \{e, x, s\}$, and $c^d \in \mathsf{Const}^d \backslash (\mathsf{Const}^e \cup \mathsf{Const}^x)$ are constants; $\Omega \in \mathsf{OpBox}$ is an operator box; and E, F, G and H are Ω–tuples[14] of box expressions. These tuples have to satisfy some conditions determined by the domain of application of the net operator induced by Ω, and so the factorisations of E, F and G are respectively factorisations of the complex empty, entry and exit marking of Ω, and the factorisation of H is a factorisation of a marking reachable from the entry or exit marking of Ω different from ${}^\circ\Omega$ and Ω°.

The above syntax only reflects the first part of the definition of the domain of an operator box, dom_Ω, which stipulates that an Ω–tuple of boxes should have a factorisation belonging to fact_Ω. The remaining two conditions, (Dom1) and (Dom2), are not captured and their treatment will be given separately below. There are two reasons why we decided to proceed in this way. The first is that in many cases (Dom1) and (Dom2) are already satisfied because of the specific properties of the operator boxes which parameterise the box algebra;[15] in particular, this is true of the extended PBC syntax defined in [7]. And, in all such cases, (33) will be exactly what is needed. The second reason is that introducing (Dom1) and (Dom2) through a BNF-like notation would involve a significant number of syntactic classes. We will instead introduce them through explicit conditions imposed on expressions satisfying the syntax (33).

Syntactic restrictions resulting from (Dom1) and (Dom2) We need to define syntactic counterparts of ex-exclusive, e-directed and x-directed boxes.

Let Expr^{wf}, Expr^{xcl}, Expr^{edir} and Expr^{xdir} be the largest sets of expressions in Expr^{box} – called respectively the *well formed*, *ex-exclusive*, *e-directed* and *x-directed* box expressions – such that the following hold:

- All ex-exclusive, e-directed and x-directed expressions are well formed. (Expr1)
- If $\mathsf{op}_\Omega(\mathsf{D}) \in \mathsf{Expr}^{wf}$ then, for every $v \in T_\Omega$:
  - (i) $D_v \in \mathsf{Expr}^{wf}$
  - (ii) if ${}^\bullet v \cap v^\bullet \neq \emptyset$ then $D_v \in \mathsf{Expr}^{xcl}$

[13] The number of transitions in a finite operator box is often called the *arity* of the operator it defines.

[14] We allow operators with infinitely many transitions, which may lead to a notational problem to specify such Ω–tuples explicitly when expressions are viewed as strings. This problem may be solved by viewing expressions as (syntax) trees rather than strings, and we will show how these can be defined.

[15] More precisely, if all the operators are pure and have all transitions reversible.

(iii) if v is not reversible then $D_v \in \mathsf{Expr}^{xdir}$. (Expr2)

- If $\mathsf{op}_\Omega(\mathsf{D}) \in \mathsf{Expr}^{xcl}$ then Ω is ex-exclusive and, for every $v \in T_\Omega$ satisfying

$$^\bullet v \cap {}^\circ\Omega \neq \emptyset \neq v^\bullet \cap \Omega^\circ \quad \text{ or } \quad v^\bullet \cap {}^\circ\Omega \neq \emptyset \neq {}^\bullet v \cap \Omega^\circ,$$

it is the case that $D_v \in \mathsf{Expr}^{xcl}$. (Expr3)
- If $\mathsf{op}_\Omega(\mathsf{D}) \in \mathsf{Expr}^{edir}$ then Ω is e-directed and, for every $v \in ({}^\circ\Omega)^\bullet$, $D_v \in \mathsf{Expr}^{edir}$. (Expr4)
- If $\mathsf{op}_\Omega(\mathsf{D}) \in \mathsf{Expr}^{xdir}$ then Ω is x-directed and, for every $v \in {}^\bullet(\Omega^\circ)$, $D_v \in \mathsf{Expr}^{xdir}$. (Expr5)
- If $\overline{E} \in \mathsf{Expr}^\delta$ or $\underline{E} \in \mathsf{Expr}^\delta$ then $E \in \mathsf{Expr}^\delta$, for $\delta \in \{wf, xcl, edir, xdir\}$. (Expr6)
- $\mathsf{Const} \subseteq \mathsf{Expr}^{wf}$ and $\mathsf{Const}^\delta = \mathsf{Const} \cap \mathsf{Expr}^\delta$, for $\delta \in \{xcl, edir, xdir\}$. (Expr7)

In the above, (Expr2)(ii) and (Expr2)(iii) encode (Dom1) and (Dom2), respectively; (Expr3) encodes theorem 4.7; and (Expr4-5) encode proposition 3.2. Moreover, (Expr6-7) serve as a means for syntactic induction.

If all the operator boxes in OpBox are pure and have only reversible transitions, then (Expr2)(ii) and (Expr2)(iii) are vacuously satisfied and thus Expr^{wf} is the entire set of box expressions defined by the syntax (33), i.e., $\mathsf{Expr}^{wf} = \mathsf{Expr}^{box}$. It is interesting to observe that this is the case for the extended PBC syntax defined in [7], as well as to the DIY algebra introduced in the next section.

Though the above definition of Expr^{wf}, Expr^{xcl}, Expr^{edir} and Expr^{xdir} is not syntactic, one can modify (33) in such a way that all these classes of expressions are properly captured; such a full syntax would have 24 different classes of expressions.

Notice that the assumption that Expr^δ, for $\delta \in \{wf, xcl, edir, xdir\}$, are the largest sets satisfying (Expr1)–(Expr7) means that, for example, if E is a well formed static expression then $\overline{E}$ is a well formed entry expression (see [7]).

6.1 A running example: the DIY algebra

We will continue to use the running example based on the boxes depicted in figure 4, in order to construct a simple algebra of process expressions. More precisely, we will consider the *Do It Yourself (DIY) algebra* based on the following two sets of boxes:

$$\mathsf{OpBox} = \{\Omega_0\} \quad \text{and} \quad \mathsf{ConstBox} = \{\Phi_1, \Phi_{11}, \Phi_{12}\} \cup \{\Phi_2, \Phi_{21}, \Phi_{22}, \Phi_{23}\} \cup \{\Phi_3\}$$

where $\lfloor \Phi_{ij} \rfloor = \Phi_i = \lfloor \Sigma_{v_i} \rfloor$, for all i and j, and the following hold:

$$M_{\Phi_{11}} = \{q_{11}, q_{14}\} \quad \text{and} \quad M_{\Phi_{12}} = \{q_{13}, q_{12}\} \quad \text{and} \quad M_{\Phi_{2k}} = \{q_{2k}\} \quad (\text{for } k = 1,2,3).$$

The constants of the DIY algebra correspond to the boxes in $\mathsf{ConstBox}$:

$$\mathsf{Const}^e = \{c_{21}\} \qquad \mathsf{Const}^s = \{c_1, c_2, c_3\}$$
$$\mathsf{Const}^x = \{c_{23}\} \qquad \mathsf{Const}^d = \{c_{11}, c_{12}, c_{21}, c_{22}, c_{23}\}.$$

Moreover, $\mathsf{Const} = \mathsf{Const}^{xcl} = \mathsf{Const}^{edir} = \mathsf{Const}^{xdir}$.

The syntax of the DIY algebra is obtained by instantiating (33) with concrete constants and operator introduced above. After taking into account the factorisations in fact_{Ω_0} (see (5)), we obtain:

$$\begin{array}{llllll}
\mathsf{Expr}^s & E & ::= & \mathsf{op}_{\Omega_0}(E,E,E) & \mid\ c_1 & \mid\ c_2 \ \mid\ c_3 \\
\mathsf{Expr}^e & F & ::= & \mathsf{op}_{\Omega_0}(F,F,E) & \mid\ c_{21} & \mid\ \overline{E} \\
\mathsf{Expr}^x & G & ::= & \mathsf{op}_{\Omega_0}(E,E,G) & \mid\ c_{23} & \mid\ \underline{E} \\
\mathsf{Expr}^d & H & ::= & \mathsf{op}_{\Omega_0}(F,\widetilde{H},E) & \mid\ \mathsf{op}_{\Omega_0}(\widetilde{H},F,E) & \mid\ c_{11} \ \mid \\
 & & & \mathsf{op}_{\Omega_0}(F,G,E) & \mid\ \mathsf{op}_{\Omega_0}(G,F,E) & \mid\ c_{12} \ \mid \\
 & & & \mathsf{op}_{\Omega_0}(G,\widetilde{H},E) & \mid\ \mathsf{op}_{\Omega_0}(\widetilde{H},G,E) & \mid\ c_{22} \ \mid \\
 & & & \mathsf{op}_{\Omega_0}(\widetilde{H},\widetilde{H},E) & \mid\ \mathsf{op}_{\Omega_0}(G,G,E) & \mid\ F \ \mid \\
 & & & \mathsf{op}_{\Omega_0}(E,E,F) & \mid\ \mathsf{op}_{\Omega_0}(E,E,\widetilde{H}) & \mid\ G
\end{array}$$

where $\widetilde{H}$ stands for an arbitrary expression in $\mathsf{Expr}^d \backslash (\mathsf{Expr}^e \cup \mathsf{Expr}^x)$. For instance, the first syntactic clause in the line starting with F comes about because $(\{v_1, v_2\}, \emptyset, \emptyset, \{v_3\})$ factorises the entry marking $(\{s_1, s_2\}, \emptyset)$ of Ω_0. Using the syntax we can see, for example, that $\mathsf{op}_{\Omega_0}(\overline{c_1}, c_{22}, c_3)$ is a valid dynamic expression of the DIY algebra. As we shall see, it corresponds to the net refinement $\Omega_0(\Sigma)$ of figure 4 discussed extensively earlier on.

The syntax for Expr^d can be simplified. For example, we can replace $\mathsf{op}_{\Omega_0}(E, E, F)$ and $\mathsf{op}_{\Omega_0}(E, E, \widetilde{H})$ by $\mathsf{op}_{\Omega_0}(E, E, H)$ since $\mathsf{op}_{\Omega_0}(E, E, G)$ is part of the syntax for G, and G itself occurs in the syntax for Expr^d. This and other similar observations lead to:

$$\mathsf{Expr}^d \qquad H ::= c_{11} \;\Big|\; c_{12} \;\Big|\; c_{22} \;\Big|\; F \;\Big|\; G \;\Big|\; \mathsf{op}_{\Omega_0}(H, H, E) \;\Big|\; \mathsf{op}_{\Omega_0}(E, E, H).$$

In fact, the syntax (33) can always be presented so that Expr^d does not refer to $\widetilde{H}$.

6.2 Infinite operators

In the case that each operator box Ω in OpBox has finitely many transitions, the meaning of the syntax (33) is clear; it simply defines four sets of finite strings, or terms. In the general case, however, we need to take in account Ω's with infinite transition sets (possibly uncountable, as allowed by the generalised parallel and choice composition), and one should ask what is meant by the syntax (33) and the expressions it generates. A possible answer is that expressions can, in general, be seen as trees and the syntax definition above as a definition of four sets of such trees, in the following way.

We will define such trees not just for the syntax (33), but even for a more general set of process expressions over the signature (32), denoted by Expr and referred to simply as *expressions*. They are defined by:

$$C ::= c \;\Big|\; \overline{C} \;\Big|\; \underline{C} \;\Big|\; \mathsf{op}_{\Omega}(\mathsf{C}) \tag{34}$$

where $c \in \mathsf{Const}$ is a constant, and C is an Ω–tuple of expressions, for $\Omega \in \mathsf{OpBox}$. For instance, this allows one to write expressions such as $\underline{\underline{\alpha}}$, $\overline{\underline{(\beta)}}$ and $\underline{\alpha}; \overline{\beta}$. Clearly, Expr^{box} defines a subset of Expr. (34) is used will be done because we shall need a richer set of expressions later on, in the definition of a similarity relation on process expressions and their operational semantics.

To give meaning to expressions defined by this syntax, we consider the set $\mathsf{ExprTrees}$ of all finite and infinite labelled trees τ, satisfying the following.

- Each node of τ is labelled by an element of the signature (32); moreover, each $x \in \mathsf{Const}$ is identified with a single node tree in $\mathsf{ExprTrees}$ whose only node is labelled by x.
- If a node ξ is labelled by a constant then ξ is a leaf.
- If a node ξ is labelled by $\overline{(.)}$ or $\underline{(.)}$, then ξ has exactly one child node to which it is connected by an unlabelled arc.
- If a node ξ is labelled by op_{Ω}, for some $\Omega \in \mathsf{OpBox}$, then its child nodes are $\{\xi_v \mid v \in T_{\Omega}\}$ and each ξ_v is connected to ξ by an arc labelled by v.

Then $\overline{(.)}$, $\underline{(.)}$ and $\mathsf{op}_{\Omega}(.)$ can be seen as mappings yielding trees in $\mathsf{ExprTrees}$. For example,

$$\overline{(.)} : \mathsf{ExprTrees} \to \mathsf{ExprTrees}$$

is given by $\overline{C} = \overline{(.)} \lhd C$. The domain $(2^{\mathsf{ExprTrees}}, \subseteq)$ forms a complete lattice, with the set intersection as the 'meet' operation, and the set union as the 'join' operation. Consider now the syntax (34). It can be seen as defining a mapping

$$expr \;:\; 2^{\mathsf{ExprTrees}} \to 2^{\mathsf{ExprTrees}}$$

such that, for every $\mathcal{T} \subseteq \mathsf{ExprTrees}$,

$$expr(\mathcal{T}) \;=\; \mathsf{Const} \cup \mathcal{T} \cup \overline{\mathcal{T}} \cup \underline{\mathcal{T}} \cup \bigcup_{\Omega \in \mathsf{OpBox}} \Big\{ \mathsf{op}_{\Omega}(\mathsf{C}) \;\Big|\; \mathsf{C} \text{ is } \Omega\text{–tuple with values in } \mathsf{Const} \cup \mathcal{T} \Big\}.$$

Clearly, $expr$ is monotonic, hence it has a unique least fixpoint, denoted by Expr. Moreover, there is a (possibly transfinite) ordinal α such that $\mathsf{Expr} = expr^{\alpha}(\emptyset)$, where for every ordinal $\beta \leq \alpha$ and $\mathcal{T} \subseteq \mathsf{ExprTrees}$,

$$expr^{\beta}(\mathcal{T}) = \begin{cases} \mathcal{T} & \text{if } \beta = 0 \\ expr(\bigcup_{\gamma<\beta} expr^{\gamma}(\mathcal{T})) & \text{if } \beta > 0. \end{cases}$$

The above implies, in particular, that no tree in Expr has an infinite path from the root. It is then possible to associate with each expression C in Expr its *rank*, denoted by $\mathsf{rank}(C)$, which is the smallest ordinal β such that $C \in expr^{\beta}(\emptyset)$. It follows that the rank of all constants is 1, $\mathsf{rank}(\overline{C}) = \mathsf{rank}(\underline{C}) = 1 + \mathsf{rank}(C)$, and $\mathsf{rank}(\mathsf{op}_{\Omega}(\mathsf{C}))$ is the least ordinal β such that $\mathsf{rank}(C_v) < \beta$, for every $v \in T_{\Omega}$. The principle of transfinite induction means that we may use induction based on the rank of expressions. Such an inductive argument reduces to the standard induction on the depth of an expression if all the operator boxes in OpBox have finitely many transitions since then the rank of every expression is a natural number (a finite ordinal).

The meaning of the syntax (33) can now be explained thus. Consider its first line, i.e., the syntactic definition of a static expression. What it defines is the minimal set Expr^{s} of expressions $E \in \mathsf{Expr}$ such that E is either a static constant or $E = \mathsf{op}_{\Omega}(\mathsf{E})$ where $\mathsf{E} \subseteq \mathsf{Expr}^{s}$. By the transfinite induction argument, such a set exists and is unique. The definitions of Expr^{e}, Expr^{x} and Expr^{d} follow the same pattern.

Properties One can show, by induction on the rank of expressions, that each subexpression[16] of an expression in Expr^{δ}, for $\delta \in \{box, s, wf\}$, is also an expression in Expr^{δ}. Moreover, it follows directly from the syntax (33) and

$$\mathsf{Const}^{s} \cap \mathsf{Const}^{d} = \mathsf{Const}^{e} \cap \mathsf{Const}^{x} = \emptyset \quad \text{and} \quad \mathsf{Const}^{e} \cup \mathsf{Const}^{x} \subseteq \mathsf{Const}^{d}$$

that Expr^{e} and Expr^{x} are disjoint subsets of Expr^{d}; and that the static and dynamic expressions are disjoint sets, i.e.,

$$\mathsf{Expr}^{s} \cap \mathsf{Expr}^{d} = \mathsf{Expr}^{e} \cap \mathsf{Expr}^{x} = \emptyset \quad \text{and} \quad \mathsf{Expr}^{e} \cup \mathsf{Expr}^{x} \subseteq \mathsf{Expr}^{d}. \tag{35}$$

These can be compared to similar relationships in the domain of boxes:

$$\mathsf{Box}^{s} \cap \mathsf{Box}^{d} = \mathsf{Box}^{e} \cap \mathsf{Box}^{x} = \emptyset \quad \text{and} \quad \mathsf{Box}^{e} \cup \mathsf{Box}^{x} \subseteq \mathsf{Box}^{d}. \tag{36}$$

Notice that (35) implies that the factorisation of an Ω–tuple of box expressions is always a partition of the set T_{Ω} of transitions of Ω.

As in the domain of boxes and PBC expressions, it is convenient to have a notation for turning an expression in Expr into a corresponding static one. We again use $\lfloor . \rfloor$ to denote such an operation. To achieve the desired effect, we assume that: (i) for each dynamic constant c there is a unique static constant $\lfloor c \rfloor$ satisfying $c \in \mathsf{Const}^{\delta} \iff \lfloor c \rfloor \in \mathsf{Const}^{\delta}$, for $\delta \in \{xcl, edir, xdir\}$; and (ii) if C is an expression, then $\lfloor C \rfloor$ is the static expression obtained by removing all occurrences of $\overline{(.)}$ and $\underline{(.)}$, and replacing every occurrence of each dynamic constant c by the corresponding static constant $\lfloor c \rfloor$. Formally,

$$\lfloor . \rfloor \ : \ \mathsf{Expr} \to \mathsf{Expr}$$

is a mapping defined by induction on the rank of expressions thus. If $\mathsf{rank}(C) = 1$ then $\lfloor C \rfloor = \lfloor c \rfloor$ for $C = c \in \mathsf{Const}^{d}$, and $\lfloor C \rfloor = C$ otherwise. Moreover, for expressions with ranks higher than 1, $\lfloor \overline{C} \rfloor = \lfloor \underline{C} \rfloor = \lfloor C \rfloor$ and $\lfloor \mathsf{op}_{\Omega}(\mathsf{C}) \rfloor = \mathsf{op}_{\Omega}(\lfloor \mathsf{C} \rfloor)$.

The operators $\overline{(.)}$, $\underline{(.)}$ and $\lfloor . \rfloor$ can be applied in the usual way (i.e., elementwise) to sets as well as tuples of expressions. The same will be true of the semantical mapping box defined in the next section, and then also of the structural similarity relation $\equiv$. In what follows, the expressions in $\mathsf{Const} \cup \overline{\mathsf{Const}^{s}} \cup \underline{\mathsf{Const}^{s}}$, i.e., those not involving any connective op_{Ω}, will be called *flat*. Notice that all flat expresions are well formed box expressions.

Example In the DIY algebra, $\mathsf{op}_{\Omega_0}(c_1, c_2, \mathsf{op}_{\Omega_0}(c_1, c_2, \underline{c_3}))$ is a well formed exit expression with rank 3. Moreover, we set $\lfloor c_{ij} \rfloor = c_i$, for every dynamic constant c_{ij}. Thus, for example, $\lfloor \mathsf{op}_{\Omega_0}(\overline{c_1}, c_{22}, c_3) \rfloor = \mathsf{op}_{\Omega_0}(c_1, c_2, c_3)$.

[16] Where, in general, a subexpression is a subtree of a tree in $\mathsf{ExprTrees}$.

7 Denotational semantics of box expressions

When dealing with the denotational semantics of the box algebra we will first define the semantics of constants, and then of compound box expressions. The semantics is given in the form of a mapping from well formed box expressions to boxes, $\mathsf{box} : \mathsf{Expr}^{wf} \to \mathsf{Box}$, defined by induction on the rank of box expressions.

Constants Constant expressions are mapped onto constant boxes of corresponding types, i.e., for every constant c and $\delta \in \{e, d, x, s, xcl, edir, xdir\}$,

$$c \in \mathsf{Const}^{\delta} \iff \mathsf{box}(c) \in \mathsf{Box}^{\delta} \cap \mathsf{ConstBox}. \tag{37}$$

We also make some additional assumptions which are not an inherent part of the definition of the denotational semantics. Their role is to ensure a consistency between the box semantics of dynamic constants and the corresponding static constants, as well as to guarantee that there is enough constants to model internal states of evolving static or dynamic constants.

Formally, it assumed that, for every dynamic constant, the underlying box is the same as that for the corresponding static constant, and that it is reachable from the entry or exit marking of the latter; i.e., for every c in Const^{d},

$$\Big\lfloor \mathsf{box}(c) \Big\rfloor = \mathsf{box}\Big(\lfloor c \rfloor\Big) \quad \text{and} \quad \mathsf{box}(c) \in \Big[\overline{\mathsf{box}(\lfloor c \rfloor)}\Big\rangle \cup \Big[\underline{\mathsf{box}(\lfloor c \rfloor)}\Big\rangle\,. \tag{38}$$

Moreover, for every non-entry and non-exit dynamic box reachable from an initially or terminally marked constant box, there is a corresponding dynamic constant, i.e., for every c in Const^{s},

$$\Big(\Big[\overline{\mathsf{box}(c)}\Big\rangle \cup \Big[\underline{\mathsf{box}(c)}\Big\rangle\Big) \setminus \Big\{\overline{\mathsf{box}(c)}, \underline{\mathsf{box}(c)}\Big\} \subseteq \mathsf{box}\Big(\mathsf{Const}^{d}\Big). \tag{39}$$

Notice that for every entry or exit constant c, it is the case that $\mathsf{box}(c) = \overline{\lfloor \mathsf{box}(c) \rfloor}$ or $\mathsf{box}(c) = \underline{\lfloor \mathsf{box}(c) \rfloor}$, respectively.

Compound expressions The definition of the semantical mapping box is completed by considering all the remaining static and dynamic well formed expressions, following the syntax (33). The box mapping is a homomorphism; hence, for every static or dynamic expression $\mathsf{op}_{\Omega}(\mathsf{D})$ and every static expression E,

$$\begin{aligned} \mathsf{box}(\mathsf{op}_{\Omega}(\mathsf{D})) &= \Omega(\mathsf{box}(\mathsf{D})) \\ \mathsf{box}(\overline{E}) &= \overline{\mathsf{box}(E)} \\ \mathsf{box}(\underline{E}) &= \underline{\mathsf{box}(E)}. \end{aligned} \tag{40}$$

The above formulae are well formed since the ranks of the expressions appearing on the right hand sides of the equality sign have strictly smaller ranks than the corresponding expressions on the left hand sides.

Example In the case of the DIY algebra, we define the box mapping by setting, for every static constant c_i, $\mathsf{box}(c_i) = \Phi_i$, and for every dynamic constant c_{ij}, $\mathsf{box}(c_{ij}) = \Phi_{ij}$. Other than that, we follow the general definitions. Thus, for example, the box in figure 4 can be derived in the following way:

$$\begin{aligned} \mathsf{box}\Big(\mathsf{op}_{\Omega_0}(\overline{c_1}, c_{22}, c_3)\Big) &= \Omega_0\Big(\mathsf{box}(\overline{c_1}), \mathsf{box}(c_{22}), \mathsf{box}(c_3)\Big) && \text{(by line 1 of (40))} \\ &= \Omega_0\Big(\overline{\mathsf{box}(c_1)}, \Phi_{22}, \Phi_3\Big) && \text{(by line 2 of (40) and def. of constants)} \\ &= \Omega_0\Big(\overline{\Phi_1}, \Phi_{22}, \Phi_3\Big) && \text{(by def. of constants)} \\ &= \Omega_0\Big(\Sigma_{v_1}, \Sigma_{v_2}, \Sigma_{v_3}\Big) && \text{(by def. of the } \Sigma_{v_i}\text{'s)} \\ &= \Omega_0\Big(\Sigma\Big). && \text{(by def. of } \Sigma\text{, cf. figure 4)} \end{aligned}$$

Properties The semantical mapping always returns a box. Moreover, the assumed type consistency between constants and their denotations, (37), carries over to the remaining box expressions. Essentially, this means we have been able to capture syntactically the property of being a static, dynamic, entry or exit box.

Theorem 7.1. Let D be a well formed box expression.

(1) $\mathsf{box}(D)$ is a static or dynamic box.
(2) For every $\delta \in \{e, d, x, s\}$, $D \in \mathsf{Expr}^\delta$ if and only if $\mathsf{box}(D) \in \mathsf{Box}^\delta$.
(3) For every $\delta \in \{xcl, edir, xdir\}$, if $D \in \mathsf{Expr}^\delta$ then $\mathsf{box}(D) \in \mathsf{Box}^\delta$.

Proof: The proof will proceed by induction on the rank of D. Using (Expr2,6), one can see that $D \in \mathsf{Expr}^{wf}$ implies that all subexpressions of D are well formed.

The base step corresponds to D being a constant. Then (1,2,3) follow directly from (35), (36), (37) and (Expr7). In the inductive step, we consider three cases.

Case 1: $D = \overline{E}$, for some static expression E. Then, by (33) and (35), $D \in \mathsf{Expr}^e \subseteq \mathsf{Expr}^d$ and $D \notin \mathsf{Expr}^x \cup \mathsf{Expr}^s$. Moreover, by the induction hypothesis and (40), $\mathsf{box}(D) = \overline{\mathsf{box}(E)} \in \overline{\mathsf{Box}^s} = \mathsf{Box}^e \subseteq \mathsf{Box}^d$ and, by (36), $\mathsf{box}(D) \notin \mathsf{Box}^x \cup \mathsf{Box}^s$. Thus (1) and (2) are satisfied. Moreover, (3) also is satisfied, by $\mathsf{box}(E) = \lfloor \mathsf{box}(D) \rfloor$, (Expr6) (which implies that $E \in \mathsf{Expr}^\delta$) and the induction hypothesis. Hence the result holds in this case.

Case 2: $D = \underline{E}$, for some static expression E. Then we proceed similarly as in Case 1.

Case 3: $D = \mathsf{op}_\Omega(\mathsf{D})$. Let μ be the factorisation of D. By the induction hypothesis, for every $v \in T_\Omega$, $\mathsf{box}(D_v)$ is a static or dynamic box; moreover, for all $\delta \in \{e, d, x, s\}$ and $v \in \mu_\delta$,

$$\mathsf{box}(D_v) \;\subseteq\; \mathsf{Box}^\delta \tag{41}$$

which means that μ is also the factorisation of $\mathsf{box}(\mathsf{D})$. Furthermore, by (Expr2) and the induction hypothesis, $\mathsf{box}(\mathsf{D})$ is an Ω–tuple satisfying (Dom1) and (Dom2). This and $\mu \in \mathsf{fact}_\Omega$ means that $\mathsf{box}(\mathsf{D}) \in \mathsf{dom}_\Omega$ and so, by (40), well formedness and theorem 4.5, we obtain that $\mathsf{box}(D)$ is a static or dynamic box. We then consider subcases depending on the type of marking of Ω whose factorisation is μ.

If μ is the factorisation of the empty marking then $\mu_e = \mu_d = \mu_x = \emptyset$. Thus $D \in \mathsf{Expr}^s$ and, by (41) and (40) and (11), $\mathsf{box}(D) \in \mathsf{Box}^s$. Moreover, by (35) and (36), $D \notin \mathsf{Expr}^\delta$ and $\mathsf{box}(D) \notin \mathsf{Box}^\delta$, for $\delta \in \{e, d, x\}$. Hence (1) and (2) are satisfied.

If μ is a factorisation of ${}^\circ\Omega$ then, by (33) and (35), $D \in \mathsf{Expr}^e \subseteq \mathsf{Expr}^d$ and $D \notin \mathsf{Expr}^x \cup \mathsf{Expr}^s$. Moreover, by proposition 2.2(4), ${}^\bullet\mu_e \cup \mu_x^\bullet = {}^\circ\Omega$ and $\mu_d = \emptyset$. Hence, by (11) and (40) and (41), $\mathsf{box}(D) \in \mathsf{Box}^e \subseteq \mathsf{Box}^d$. The latter and (36) further implies $\mathsf{box}(D) \notin \mathsf{Box}^x \cup \mathsf{Box}^s$. Hence (1) and (2) are satisfied. If μ is a factorisation of Ω°, we proceed in a similar way.

If μ is a factorisation of a complex marking reachable from ${}^\circ\Omega$ or Ω°, but different from ${}^\circ\Omega$ and Ω°, then, by (33) and (35), $D \in \mathsf{Expr}^d \backslash (\mathsf{Expr}^e \cup \mathsf{Expr}^x)$ and $D \notin \mathsf{Expr}^s$. Clearly, by (40) and (41), $\mathsf{box}(D) \notin \mathsf{Box}^s$. Suppose $\mathsf{box}(D) \in \mathsf{Box}^e \cup \mathsf{Box}^x$. Then, by (40) and (41) and lemma 3.11, ${}^\circ\Omega \subseteq {}^\bullet\mu_e \cup \mu_x^\bullet$ or $\Omega^\circ \subseteq {}^\bullet\mu_e \cup \mu_x^\bullet$. Hence, by proposition 2.2(4), μ is a factorisation of ${}^\circ\Omega$ or Ω°, a contradiction. Thus $\mathsf{box}(D) \in \mathsf{Box}^d \backslash (\mathsf{Box}^e \cup \mathsf{Box}^x)$, and so (1) and (2) are satisfied.

We have shown that (1) and (2) are satisfied. We then observe that (3) follows from (Expr3)–(Expr5), the induction hypothesis, $\mathsf{box}(D) = \Omega(\mathsf{box}(\mathsf{D}))$, theorem 4.7 and proposition 3.2. □

The semantic translation commutes with removing of the over- and underbars and replacing dynamic constants by static ones. Moreover, in a box generated from an expression, either all transitions are single-node trees, or none is.

Proposition 7.2. Let D be a dynamic and E static well formed box expression.

(1) $\lfloor \mathsf{box}(D) \rfloor = \mathsf{box}(\lfloor D \rfloor)$.
(2) Exactly one of the following holds:
   (a) $T_{\mathsf{box}(E)} \subseteq \mathsf{T_{root}}$ and $E \in \mathsf{Const}^s$.
   (b) $T_{\mathsf{box}(E)} \cap \mathsf{T_{root}} = \emptyset$ and $E = \mathsf{op}_\Omega(\mathsf{E})$, for some Ω and E.

Proof: The first part follows from proposition 3.3, (38) and by a straightforward induction on the rank of D. The second part follows from (31) and (37). □

8 Operational semantics of box expressions

In this section, we will introduce two kinds of structured operational semantics for box expressions, based respectively on the transitions of the corresponding boxes, and the actions labelling these transitions.

8.1 Structural similarity relation on expressions

We first define a structural similarity relation on box expressions, $\equiv$. It identifies expressions which can be shown denotationally equal by purely syntactic means. The way in which it will be defined resembles to some extent the definition of the similarity relation for (tuples of) boxes. This should not be too surprising since the denotational semantics of the box algebra is essentially derived from the algebra of boxes presented earlier on.

The reason why we introduced, in (34), a larger set of expressions than it was necessary to define the domain of box expressions is that the rules of the structural equivalence (and, later, operational semantics) should act as term rewriting rules. That is, whenever a (well formed) box expression can match one side of such a rule, then it should be guaranteed that the other side is a (well formed) box expression too. To be able to express and prove such a property we need to allow for more general expressions, such as $\underline{\underline{\alpha}}$, $\overline{(\underline{\beta})}$ and $\underline{\alpha};\overline{\beta}$.

In the definition of the structural similarity on box expressions, we need to take into account possible factorisations of the operator boxes (since the relation we are going to define can be seen as a counterpart of the similarity relations $\equiv_\Omega$ on Ω–tuples of boxes). Formally, we define $\equiv$ to be the least binary relation on expressions in Expr such that (42)–(49) below hold.

Reflexivity, symmetry and transitivity For all expressions D, H and J,

$$\boxed{D \equiv D \qquad \frac{D \equiv H}{H \equiv D} \qquad \frac{D \equiv J\ ,\ J \equiv H}{D \equiv H}} \tag{42}$$

Flat expressions For all flat expressions D and H satisfying $\mathsf{box}(D) = \mathsf{box}(H)$,

$$\boxed{D \equiv H} \tag{43}$$

Entry expressions For every operator box Ω in OpBox and every factorisation μ of ${}^\circ\Omega$,

$$\boxed{\mathsf{op}_\Omega(\mathsf{D}) \equiv \overline{\mathsf{op}_\Omega(\mathsf{H})}} \tag{44}$$

where D and H are Ω–tuples of expressions[17] such that, for every $v \in T_\Omega$: $D_v = \overline{H_v}$ if $v \in \mu_e$; $D_v = \underline{H_v}$ if $v \in \mu_x$; and $D_v = H_v$ otherwise.

Exit expressions For every operator box Ω in OpBox and every factorisation μ of Ω°,

$$\boxed{\mathsf{op}_\Omega(\mathsf{D}) \equiv \underline{\mathsf{op}_\Omega(\mathsf{H})}} \tag{45}$$

where D and H are as in (44).

Equivalent factorisations For every operator box Ω in OpBox, for every complex marking reachable from the entry or exit marking of Ω and different from ${}^\circ\Omega$ and Ω°, and for every pair of different factorisations μ and κ of that marking,

$$\boxed{\mathsf{op}_\Omega(\mathsf{D}) \equiv \mathsf{op}_\Omega(\mathsf{H})} \tag{46}$$

where D and H are Ω–tuples of expressions for which there is an Ω–tuple of expressions C such that, for every $v \in T_\Omega$,

$$D_v = \begin{cases} \overline{C_v} \text{ if } & v \in \mu_e \\ \underline{C_v} \text{ if } & v \in \mu_x \\ C_v \text{ otherwise.} \end{cases} \qquad \text{and} \qquad H_v = \begin{cases} \overline{C_v} \text{ if } & v \in \kappa_e \\ \underline{C_v} \text{ if } & v \in \kappa_x \\ C_v \text{ otherwise.} \end{cases} \tag{47}$$

[17] Notice that here and later it is not required that D and H be Ω–tuples of *box* expressions.

Substitutivity For all expressions D and H,

$$\boxed{\frac{D \equiv H}{\overline{D} \equiv \overline{H}} \qquad \frac{D \equiv H}{\underline{D} \equiv \underline{H}}} \tag{48}$$

and, for every operator box Ω in OpBox,

$$\boxed{\frac{\mathsf{D} \equiv \mathsf{H}}{\mathsf{op}_\Omega(\mathsf{D}) \equiv \mathsf{op}_\Omega(\mathsf{H})}} \tag{49}$$

where D and H are Ω–tuples of expressions.

Infinite operators As it was the case for the syntax of the box algebra, the meaning of the similarity relation $\equiv$ is clear if we only consider operator boxes OpBox with finite transition sets. However, since we also allow infinite operator boxes, some explanations are needed to clarify what is meant by the 'the least binary relation on expressions such that (42)–(49) hold'. We start by observing that the domain $(2^{\mathsf{Expr}\times\mathsf{Expr}}, \subseteq)$ forms a complete lattice, with component-wise set intersection as the 'meet' operation, and component-wise set union as the 'join' operation, and that the rules (42)–(49) define a monotonic mapping

$$sim \ : \ 2^{\mathsf{Expr}\times\mathsf{Expr}} \to 2^{\mathsf{Expr}\times\mathsf{Expr}}$$

where $sim(\mathcal{E})$ is $\mathcal{E}$ together with all pairs in $\mathsf{Expr}\times\mathsf{Expr}$ which can be derived from the pairs in $\mathcal{E}$ using a single application of any of the rules (42)–(49).

By following the same line of reasoning as before, we conclude that sim has a unique least fixpoint, denoted by $\equiv$, and equal to $sim^{\alpha'}(\emptyset)$, for some ordinal α' (such that $sim^{\alpha'}(\emptyset) = sim^{\beta}(\emptyset)$, for any $\beta \geq \alpha'$). And, as before, we can define the *rank* of a pair (D, H) belonging to $\equiv$, denoted by $\mathsf{rank}(D, H)$, as the least ordinal β such that $(D, H) \in sim^{\beta}(\emptyset)$. It follows from the principle of transfinite induction that we can apply induction on the rank of elements of $\equiv$ which reduces to the standard induction if all operator boxes in OpBox have finite transition sets.

The tuple C used in the formulation of (46) intuitively corresponds to the common (and thus unchanging) part of the tuples D and H. For example, in the DIY algebra, since $(\{v_3\}, \emptyset, \emptyset, \{v_1, v_2\})$ and $(\emptyset, \emptyset, \{v_1, v_2\}, \{v_3\})$ are factorisations of the same marking reachable from ${}^\circ\Omega_0$, we shall have

$$\mathsf{op}_{\Omega_0}(c_1, c_2, \overline{c_3}) \equiv \mathsf{op}_{\Omega_0}(\underline{c_1}, \underline{c_2}, c_3) \quad \text{and} \quad \mathsf{op}_{\Omega_0}(\overline{c_1}, \overline{c_2}, \overline{c_3}) \equiv \mathsf{op}_{\Omega_0}(\underline{(\overline{c_1})}, \underline{(\overline{c_2})}, c_3).$$

Properties By a straightforward application of (transfinite) induction on the rank of expressions in Expr, one can see that $\equiv$ is an equivalence relation, and if E and F are structurally equivalent static expressions, then no derivation for $E \equiv F$ can use the rules (44), (45), (46) and (48). Moreover, the structural similarity relation is closed in the domain of box expressions and preserves their types.

Theorem 8.1. Let D and H be expressions such that $D \equiv H$. Then $D \in \mathsf{Expr}^{\delta}$ if and only if $H \in \mathsf{Expr}^{\delta}$, for every $\delta \in \{e, d, x, s, wf, xcl, edir, xdir\}$.

Proof: The proof proceeds by induction on $\mathsf{rank}(D, H)$. In the base step, $D \equiv H$ is directly derived from the first part of (42) or one of (43)–(46). If $D = H$ then there is nothing to prove; and for (43) the result follows from (37) and (Expr7). We will now show that the result holds if $D \equiv H$ has been derived directly from (44) or (46) (the proof for (45) is similar to that for (44)).

Suppose that $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \overline{\mathsf{op}_\Omega(\mathsf{H})}$ are as in (44). If D is a box expression then, by the syntax (33), D is an Ω–tuple of box expressions such that the factorisation κ of D belongs to fact_Ω. Moreover, we have $\mu_e \subseteq \kappa_e$ and $\mu_x \subseteq \kappa_x$, and so ${}^\circ\Omega = {}^\bullet\mu_e \cup \mu_x^\bullet \subseteq {}^\bullet\kappa_e \cup \kappa_x^\bullet$. Hence, by proposition 2.2(5), we have $\mu = \kappa$, $\mu_d = \emptyset$, H is a tuple of static expressions, and D is an entry expression. Thus, by $\mathsf{H} \subseteq \mathsf{Expr}^{s}$, H is also an entry expression. Conversely, if H is a box expression then H must be an entry expression and, as before, H must be a tuple of static expressions. Thus D is also an entry expression. The part of the result for $\delta \in \{wf, xcl, edir, xdir\}$ follows from (Expr2)–(Expr6) and $D_v \in \{H_v, \overline{H_v}, \underline{H_v}\}$, for every $v \in T_\Omega$, and the fact that the Expr^{δ}'s (for $\delta \in \{wf, xcl, edir, xdir\}$) are the largest sets satisfying (Expr1)–(Expr7).

Suppose that $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \mathsf{op}_\Omega(\mathsf{H})$ are as in (46). Assume that D is a box expression (if H is a box expression, the argument is symmetric). Let ϕ be the factorisation of D. By (47), since D is a box expression and $\mathsf{D}_{\mu_e} = \overline{\mathsf{C}_{\mu_e}}$ and $\mathsf{D}_{\mu_x} = \underline{\mathsf{C}_{\mu_x}}$, we have that C is an Ω–tuple of box expressions, $\mu_e \subseteq \phi_e$, $\mu_x \subseteq \phi_x$, $\mathsf{C}_{\mu_e} \cup \mathsf{C}_{\mu_x} \subseteq \mathsf{Expr}^s$, $\mathsf{D}_{\mu_e} \subseteq \overline{\mathsf{Expr}^s}$ and $\mathsf{D}_{\mu_x} \subseteq \underline{\mathsf{Expr}^s}$. We now observe that H is a tuple of box expressions. Indeed, suppose that H_v is not a box expression. Then, by (47) and C being a tuple of box expressions, $v \in \kappa_e \cup \kappa_x$. Thus, by $\mathsf{C}_{\mu_e} \cup \mathsf{C}_{\mu_x} \subseteq \mathsf{Expr}^s$, it must be the case that $v \in \mu_d$. On the other hand, since μ and κ are factorisations of the same marking, we have $({}^\bullet\mu_e \cup \mu_x^\bullet) \cap ({}^\bullet v \cup v^\bullet) \neq \emptyset$. But this contradicts 2.2(4). Thus, H is a tuple of box expressions whose factorisation, given by

$$\psi \;=\; \Big((\phi_e \backslash \mu_e) \cup \kappa_e \;,\; \phi_d \;,\; (\phi_x \backslash \mu_x) \cup \kappa_x \;,\; \phi_s \cup (\mu_e \backslash \kappa_e) \cup (\mu_x \backslash \kappa_x) \Big),$$

is a factorisation of the same marking as ϕ. Hence $H = \mathsf{op}_\Omega(\mathsf{H})$ is a box expression and $D \in \mathsf{Expr}^\delta$ if and only if $H \in \mathsf{Expr}^\delta$, for every $\delta \in \{e, d, x, s\}$. The part of the result for $\delta \in \{wf, xcl, edir, xdir\}$ follows from (Expr2)–(Expr6) and $D_v, H_v \in \{C_v, \overline{C_v}, \underline{C_v}\}$, for every $v \in T_\Omega$, and the fact that the Expr^δ's (for $\delta \in \{wf, xcl, edir, xdir\}$) are the largest sets satisfying (Expr1)-(Expr7).

In the inductive step, we consider four cases.

Case 1: $H \equiv D$ and $\mathsf{rank}(H, D) < \mathsf{rank}(D, H)$. Then, by the induction hypothesis, the result holds for $H \equiv D$. Clearly, it then also holds for $D \equiv H$.

Case 2: There is J such that $D \equiv J$, $J \equiv H$, and $\max\{\mathsf{rank}(H, J), \mathsf{rank}(J, H)\} < \mathsf{rank}(D, H)$. Then, by the induction hypothesis, the result holds for both $D \equiv J$ and $J \equiv H$. Clearly, it then also holds for $D \equiv H$.

Case 3: $D = \overline{D_0}$, $H = \overline{H_0}$, $D_0 \equiv H_0$, and $\mathsf{rank}(D_0, H_0) < \mathsf{rank}(D, H)$. Suppose D is a box expression (if H is a box expression, the argument is symmetric). Then D_0 is a static expression. Hence, by the induction hypothesis, H_0 is also a static expression. Thus $D, H \in \mathsf{Expr}^e$.
If $D = \underline{D_0}$ and $H = \underline{H_0}$ the proof is similar.

Case 4: $D = \mathsf{op}_\Omega(\mathsf{D})$, $H = \mathsf{op}_\Omega(\mathsf{H})$, $\mathsf{D} \equiv \mathsf{H}$ and, for every $v \in T$, $\mathsf{rank}(D_v, H_v) < \mathsf{rank}(D, H)$. Suppose D is a box expression (if H is a box expression, the argument is symmetric). Then D is an Ω–tuple of box expressions and, by the induction hypothesis, H is an Ω–tuple of box expressions with the same factorisation as D. Hence H is a box expression and $D \in \mathsf{Expr}^\delta$ if and only if $H \in \mathsf{Expr}^\delta$, for every $\delta \in \{e, d, x, s\}$.

Finally, we observe for the Cases 1–4, that the part of the result for $\delta \in \{wf, xcl, edir, xdir\}$ follows from (Expr2)–(Expr5), the induction hypothesis and the fact that the Expr^δ's (for $\delta \in \{wf, xcl, edir, xdir\}$) are the largest sets satisfying (Expr1)–(Expr7). □

Corollary 8.2. If $\mathsf{op}_\Omega(\mathsf{D})$ and $\mathsf{op}_\Omega(\mathsf{H})$ are well formed box expressions satisfying the conditions in the rule for equivalent factorisations, (46), extended to factorisations of ${}^\circ\Omega$ and Ω°,[18] then $\mathsf{box}(\mathsf{D}) \equiv_\Omega \mathsf{box}(\mathsf{H})$.
Proof: An inspection of the proof of theorem 8.1 reveals that the factorisations of D and H are factorisations of the same marking of Ω. Thus, by theorem 7.1, the factorisations of $\mathsf{box}(\mathsf{D})$ and $\mathsf{box}(\mathsf{H})$ are factorisations of the same marking of Ω. Moreover, by proposition 7.2(1) and $\lfloor \mathsf{D} \rfloor = \lfloor \mathsf{H} \rfloor$, we have $\lfloor \mathsf{box}(\mathsf{D}) \rfloor = \lfloor \mathsf{box}(\mathsf{H}) \rfloor$. Hence $\mathsf{box}(\mathsf{D}) \equiv_\Omega \mathsf{box}(\mathsf{H})$. □

We now will show that $\equiv$ is a sound equivalence notion from the point of view of the denotational semantics. This central property of the structural similarity relation is preceded by two auxiliary lemmata.

Lemma 8.3. If D is an entry or exit expression then $D \equiv \overline{\lfloor D \rfloor}$ or $D \equiv \underline{\lfloor D \rfloor}$, respectively.
Proof: We proceed by induction on $\mathsf{rank}(D)$. In the base step, D is a dynamic constant, and $D \equiv \overline{\lfloor D \rfloor}$ or $D \equiv \underline{\lfloor D \rfloor}$ follows from (37), (38) and (43). In the inductive step we consider four cases.

Case 1: $D = \overline{E}$ and E is a static expression. Then $E = \lfloor D \rfloor$ and $D = \overline{E} = \overline{\lfloor D \rfloor} \equiv \overline{\lfloor D \rfloor}$ since $\equiv$ is reflexive.

Case 2: $D = \underline{E}$ and E is a static expression. We then proceed similarly as in Case 1.

Case 3: $D = \mathsf{op}_\Omega(\mathsf{D})$ and the factorisation μ of the Ω–tuple of box expressions D is a factorisation of the entry marking of Ω and for each $v \in T_\Omega$, $\mathsf{rank}(D_v) < \mathsf{rank}(D)$. By the induction hypothesis and

[18] I.e., we here allow μ and κ to be factorisations of the entry or exit marking of Ω.

(49), we have $\mathsf{op}_\Omega(\mathsf{D}) \equiv \mathsf{op}_\Omega(\mathsf{H})$ where, for every $v \in T_\Omega$,

$$H_v = \begin{cases} \overline{\lfloor D_v \rfloor} \text{ if } & v \in \mu_e \\ \underline{\lfloor D_v \rfloor} \text{ if } & v \in \mu_x \\ \lfloor D_v \rfloor \text{ otherwise.} \end{cases}$$

Notice that $\lfloor D_v \rfloor = D_v$ for $v \notin \mu_e \cup \mu_x$ since $\mu_d = \emptyset$ as μ is a factorisation of ${}^\circ\Omega$. We may now apply (44) to conclude that $\mathsf{op}_\Omega(\mathsf{H}) \equiv \overline{\mathsf{op}_\Omega(\lfloor \mathsf{D} \rfloor)}$ which, together with proposition 7.2(1) and (42), yields $D \equiv \overline{\lfloor D \rfloor}$.

Case 4: $D = \mathsf{op}_\Omega(\mathsf{D})$ and the factorisation of the Ω–tuple of box expressions D is a factorisation of the exit marking of Ω. We then proceed similarly as in Case 3. □

Lemma 8.4. Let D and H be well formed static expressions such that $\mathsf{box}(D)$ and $\mathsf{box}(H)$ have the same sets of places. Then $D \equiv H$.

Proof: We proceed by induction on $\max\{\mathsf{rank}(D), \mathsf{rank}(H)\}$. In the base step, D and H are static constants and so, by (31) and the fact that $\mathsf{box}(D)$ and $\mathsf{box}(H)$ share at least one place, we have $\mathsf{box}(D) = \mathsf{box}(H)$. Hence, by (43), $D \equiv H$. In the inductive step, we have $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \mathsf{op}_\Omega(\mathsf{H})$. From $S_{\Omega(\mathsf{box}(D))} = S_{\Omega(\mathsf{box}(H))}$ and the definition of net refinement, it follows that $S_{\mathsf{box}(D_v)} = S_{\mathsf{box}(H_v)}$, for every $v \in T_\Omega$. Hence, by the induction hypothesis, $D_v \equiv H_v$. Thus $\mathsf{D} \equiv \mathsf{H}$, and so $D \equiv H$ holds by (49). □

Theorem 8.5. Let D and H be well formed box expressions. Then $D \equiv H$ if and only if $\mathsf{box}(D) = \mathsf{box}(H)$.

Proof: ($\Longrightarrow$) The proof proceeds by induction on $\mathsf{rank}(D, H)$. In the base step we consider five cases.

Case 1: $D = H$, or D and H are flat expressions and $\mathsf{box}(D) = \mathsf{box}(H)$. This case is trivial.

Case 2: $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \overline{\mathsf{op}_\Omega(\mathsf{H})}$ where $\mathsf{op}_\Omega(\mathsf{D})$ and $\mathsf{op}_\Omega(\mathsf{H})$ are as in (44). Then H is an entry expression so, by theorem 8.1, D is also an entry expression. Moreover, by proposition 7.2(1) and $\lfloor D \rfloor = \lfloor H \rfloor$, $\lfloor \mathsf{box}(D) \rfloor = \mathsf{box}(\lfloor D \rfloor) = \mathsf{box}(\lfloor H \rfloor) = \lfloor \mathsf{box}(H) \rfloor$. Hence $\mathsf{box}(D) = \mathsf{box}(H) = \overline{\mathsf{box}(\mathsf{op}_\Omega(\mathsf{H}))}$.

Case 3: $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \underline{\mathsf{op}_\Omega(\mathsf{H})}$ where $\mathsf{op}_\Omega(\mathsf{D})$ and $\mathsf{op}_\Omega(\mathsf{H})$ are as in (45). Then we proceed similarly as in Case 2.

Case 4: $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \mathsf{op}_\Omega(\mathsf{H})$ where $\mathsf{op}_\Omega(\mathsf{D})$ and $\mathsf{op}_\Omega(\mathsf{H})$ are as in (46). Then, by corollary 8.2, $\mathsf{box}(\mathsf{D}) \equiv_\Omega \mathsf{box}(\mathsf{H})$ which, by proposition 4.2, means that $\mathsf{box}(D) = \mathsf{box}(H)$.

In the inductive step, we consider four cases.

Case 1: $H \equiv D$ and $\mathsf{rank}(H, D) < \mathsf{rank}(D, H)$. Then, by the induction hypothesis, $\mathsf{box}(H) = \mathsf{box}(D)$.

Case 2: There is J such that $D \equiv J$ and $J \equiv H$ and $\max\{\mathsf{rank}(D, J), \mathsf{rank}(J, H)\} < \mathsf{rank}(D, H)$. Then, by the induction hypothesis, $\mathsf{box}(D) = \mathsf{box}(J) = \mathsf{box}(H)$.

Case 3: $D = \overline{D_0}$ and $H = \overline{H_0}$ and $D_0 \equiv H_0$ and $\mathsf{rank}(D_0, H_0) < \mathsf{rank}(D, H)$. (If $D = \underline{D_0}$ and $H = \underline{H_0}$ the proof is similar.) Then, by the induction hypothesis, $\mathsf{box}(D_0) = \mathsf{box}(H_0)$. Hence $\mathsf{box}(D) = \overline{\mathsf{box}(D_0)} = \overline{\mathsf{box}(H_0)} = \mathsf{box}(H)$.

Case 4: $D = \mathsf{op}_\Omega(\mathsf{D})$, $H = \mathsf{op}_\Omega(\mathsf{H})$, $\mathsf{D} \equiv \mathsf{H}$, and, for every $v \in T$, $\mathsf{rank}(D_v, H_v) < \mathsf{rank}(D, H)$. Then, by the induction hypothesis, $\mathsf{box}(\mathsf{D}) = \mathsf{box}(\mathsf{H})$. Hence, by (40), $\mathsf{box}(D) = \mathsf{box}(\mathsf{op}_\Omega(\mathsf{D})) = \mathsf{box}(\mathsf{op}_\Omega(\mathsf{H})) = \mathsf{box}(H)$.

($\Longleftarrow$) By theorem 7.1 and $\mathsf{box}(D) = \mathsf{box}(H)$, it suffices to consider the following four cases.

Case 1: $D, H \in \mathsf{Expr}^s$. Then $D \equiv H$ follows from lemma 8.4.

Case 2: $D, H \in \mathsf{Expr}^e$. Then, by lemma 8.3, $D \equiv \overline{\lfloor D \rfloor}$ and $H \equiv \overline{\lfloor H \rfloor}$. By proposition 7.2(1) and $\mathsf{box}(D) = \mathsf{box}(H)$, $\mathsf{box}(\lfloor D \rfloor) = \mathsf{box}(\lfloor H \rfloor)$. Thus, by lemma 8.4, $\lfloor D \rfloor \equiv \lfloor H \rfloor$. Hence, by (48) and the symmetry of $\equiv$, $D \equiv \overline{\lfloor D \rfloor} \equiv \overline{\lfloor H \rfloor} \equiv H$. Thus, by the transitivity of $\equiv$, $D \equiv H$.

Case 3: $D, H \in \mathsf{Expr}^x$. Then we proceed similarly as in Case 2.

Case 4: $D, H \in \mathsf{Expr}^d \backslash (\mathsf{Expr}^e \cup \mathsf{Expr}^x)$. Then we proceed by induction on $\max\{\mathsf{rank}(D), \mathsf{rank}(H)\}$. In the base step, D and H are dynamic constants, and $D \equiv H$ follows from (43). In the inductive step, we have $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \mathsf{op}_\Omega(\mathsf{H})$ where $\max\{\mathsf{rank}(D_v), \mathsf{rank}(H_v)\} < \max\{\mathsf{rank}(D), \mathsf{rank}(H)\}$, for every $v \in T_\Omega$. By proposition 7.2(1) and $\mathsf{box}(D) = \mathsf{box}(H)$, we have $\Omega(\mathsf{box}(\lfloor \mathsf{D} \rfloor)) = \Omega(\mathsf{box}(\lfloor \mathsf{H} \rfloor))$, so $S_{\lfloor D_v \rfloor} = S_{\lfloor H_v \rfloor}$, for every $v \in T_\Omega$. Hence, by lemma 8.4, $\lfloor \mathsf{D} \rfloor \equiv \lfloor \mathsf{H} \rfloor$. Since $\mathsf{box}(\lfloor \mathsf{D} \rfloor) = \mathsf{box}(\lfloor \mathsf{H} \rfloor)$ and $\mathsf{box}(D) = \mathsf{box}(H)$, by proposition 4.2, $\mathsf{box}(\mathsf{D}) \equiv_\Omega \mathsf{box}(\mathsf{H})$. Let μ and κ be factorisations of respectively D

and H (note that $\mu_d = \kappa_d$). We have $\mathsf{box}(\mathsf{D}_{\mu_d}) = \mathsf{box}(\mathsf{H}_{\mu_d})$ and $\lfloor \mathsf{D}_{\mu_d} \rfloor = \lfloor \mathsf{H}_{\mu_d} \rfloor$; hence, by the induction hypothesis, $\mathsf{D}_{\mu_d} \equiv \mathsf{H}_{\mu_d}$. Moreover, by lemma 8.3,

$$\mathsf{D}_{\mu_e} \equiv \overline{\lfloor \mathsf{D}_{\mu_e} \rfloor} \quad , \quad \mathsf{D}_{\mu_x} \equiv \underline{\lfloor \mathsf{D}_{\mu_x} \rfloor} \quad , \quad \mathsf{H}_{\kappa_e} \equiv \overline{\lfloor \mathsf{H}_{\kappa_e} \rfloor} \quad \text{and} \quad \mathsf{H}_{\kappa_x} \equiv \underline{\lfloor \mathsf{H}_{\kappa_x} \rfloor}.$$

Then one can show $D \equiv H$ by applying (46), (48) and (49). □

Example The DIY algebra gives rise to five specific rules of the structural equivalence relation (we omit here their symmetric, hence redundant, counterparts):

$$\boxed{\begin{array}{lll} \mathsf{op}_{\Omega_0}(\overline{E}, \overline{F}, G) \equiv \overline{\mathsf{op}_{\Omega_0}(E,F,G)} & \qquad & \overline{c_2} \equiv c_{21} \\ \mathsf{op}_{\Omega_0}(E, F, \underline{G}) \equiv \underline{\mathsf{op}_{\Omega_0}(E,F,G)} & & \underline{c_2} \equiv c_{23} \\ \mathsf{op}_{\Omega_0}(\underline{E}, \underline{F}, G) \equiv \mathsf{op}_{\Omega_0}(E,F,\overline{G}) & & \end{array}}$$

Using these, one can derive $\mathsf{op}_{\Omega_0}(\underline{c_1}, c_{23}, c_3) \equiv \mathsf{op}_{\Omega_0}(c_1, c_2, \overline{c_3})$, as shown below:

$$\mathsf{op}_{\Omega_0}(\underline{c_1}, c_{23}, c_3) \equiv \mathsf{op}_{\Omega_0}(c_1, c_2, \overline{c_3})$$

$$\overbrace{\mathsf{op}_{\Omega_0}(\underline{c_1}, c_{23}, c_3) \equiv \mathsf{op}_{\Omega_0}(\underline{c_1}, \underline{c_2}, c_3) \qquad\qquad \mathsf{op}_{\Omega_0}(\underline{c_1}, \underline{c_2}, c_3) \equiv \mathsf{op}_{\Omega_0}(c_1, c_2, \overline{c_3})}$$

$$\overbrace{\underline{c_1} \equiv \underline{c_1} \qquad\qquad c_{23} \equiv \underline{c_2} \qquad\qquad c_3 \equiv c_3}$$

8.2 Transition based operational semantics

In developing the operational semantics of the box algebra, we will go through the following steps: first (in this section), we introduce operational rules based on transitions of nets which provide the denotational semantics of box expressions. Based on these, we will formulate our key consistency result. And, finally, we will derive from them label based rules and partial order semantics, together with derived consistency results.

Transition trees Consider the set of all transition trees in the boxes derived through the box mapping:

$$\mathsf{T}^{box}_{\mathsf{tree}} = \bigcup_{D \in \mathsf{Expr}^{box}} T_{\mathsf{box}(D)} =_{\mathrm{prop.7.2(1)}} \bigcup_{E \in \mathsf{Expr}^{s}} T_{\mathsf{box}(E)}.$$

Every $t \in \mathsf{T}^{box}_{\mathsf{tree}}$ has a unique label, $\mathsf{lab}(t)$, in all boxes associated with box expressions in which it occurs. More precisely, if $t \in \mathsf{T}_{\mathsf{root}}$ then this follows from (31) and (38), and if t has the form $t = (v, \alpha) \lhd R$ then, by the definition of net refinement which underpins the semantical mapping box, $\mathsf{lab}(t) = \alpha$. To see that the same transition may belong to different static boxes, consider $E = \mathsf{op}_{\Omega_0}(c_1, c_2, c_2)$ and $F = \mathsf{op}_{\Omega_0}(c_1, c_2, c_3)$ in DIY. Then the transition $t = (v_1, f) \lhd \{t_{11}, t_{12}\}$ belongs to both $\mathsf{box}(E)$ and $\mathsf{box}(F)$, yet $\mathsf{box}(E) \neq \mathsf{box}(F)$.

Example In the DIY algebra, each transition tree in $\mathsf{T}^{box}_{\mathsf{tree}}$ has one of the following four forms:

$$(v_{n_1}, a_{ij}) \lhd \cdots \lhd (v_{n_k}, a_{ij}) \lhd t_{ij} \quad , \quad (v_{m_1}, a) \lhd \cdots \lhd (v_{m_{k_1}}, a) \lhd t_{11} \quad , \quad (v_{l_1}, b) \lhd \cdots \lhd (v_{l_{k_2}}, b) \lhd t_{12}$$

$$(v_{n_1}, f) \lhd \cdots \lhd (v_{n_k}, f) \lhd (v_1, f) \lhd \Big\{ (v_{m_1}, a) \lhd \cdots \lhd (v_{m_{k_1}}, a) \lhd t_{11} \ , \ (v_{l_1}, b) \lhd \cdots \lhd (v_{l_{k_2}}, b) \lhd t_{12} \Big\}$$

where $k, k_1, k_2 \geq 0$ and $t_{ij} \in \{t_{21}, t_{22}, t_{31}\}$ and $v_{n_i} \in \{v_1, v_2, v_3\}$ and $v_{m_i}, v_{l_i} \in \{v_2, v_3\}$ and $a_{11} = a$ and $a_{12} = b$ and $a_{21} = c$ and $a_{22} = d$ and $a_{31} = e$.

SOS semantics rules The first operational semantics has moves of the form $D \xrightarrow{U} H$ such that D and H are box expression and $U \in \mathsf{tlab}_{\mathsf{sr}}$, where $\mathsf{tlab}_{\mathsf{sr}}$ is the set of all finite subsets of $\mathsf{T}^{box}_{\mathsf{tree}} \cup \{\mathsf{skip}, \mathsf{redo}\}$. The idea here is that U is a valid step for the boxes associated with D and H, after augmenting them with the skip and redo transitions. We will denote, for every such set,

$$\mathsf{lab}(U) \;=\; \sum_{t \in U} \{\mathsf{lab}(t)\}$$

assuming that $\mathsf{lab}(\mathsf{skip}) = \mathsf{skip}$ and $\mathsf{lab}(\mathsf{redo}) = \mathsf{redo}$. The derivation system is defined in four stages. First, we define the rules for skip and redo, then for the inaction rules, then we treat the flat expressions and, finally, introduce the derivation rule for each connective op_Ω. Formally, we define a ternary relation $\longrightarrow$ which is the least relation comprising all $(D, U, H) \in \mathsf{Expr} \times \mathsf{tlab}_{\mathsf{sr}} \times \mathsf{Expr}$ such that (51)–(54) below hold. Notice that we use $D \xrightarrow{U} H$ to denote $(D, U, H) \in \longrightarrow$.

skip *and* redo There are two basic rules governing the applications of the moves involving skip and redo. For every static expression E,

$$\boxed{\overline{E} \xrightarrow{\{\mathsf{skip}\}} \underline{E} \qquad\qquad \underline{E} \xrightarrow{\{\mathsf{redo}\}} \overline{E}} \tag{50}$$

Inaction rules There is a single basic inaction rule,

$$\boxed{\frac{D \equiv H}{D \xrightarrow{\emptyset} H}} \tag{51}$$

where D and H are box expressions. Moreover, an empty move can be absorbed by the move which precedes or follows it, i.e., we have:

$$\boxed{\frac{D \xrightarrow{\emptyset} J \xrightarrow{U} H}{D \xrightarrow{U} H} \qquad\qquad \frac{D \xrightarrow{U} J \xrightarrow{\emptyset} H}{D \xrightarrow{U} H}} \tag{52}$$

where D, J and H are expressions. The inaction rules render the view that empty moves do not change the state of a system, but rather change the view on it.

Axioms for flat expressions For all flat expressions D and H and $U \in \mathsf{tlab}_{\mathsf{sr}}$ such that $\mathsf{box}(D)\ [U\rangle\ \mathsf{box}(H)$,

$$\boxed{D \xrightarrow{U} H} \tag{53}$$

Inference rules for operators For every operator box Ω in OpBox, and all Ω–tuples D and H of expressions, there is an inference rule:

$$\boxed{\frac{\forall\, v \in T_\Omega \;:\; D_v \xrightarrow{U_v^1 \uplus \ldots \uplus U_v^{k_v}} H_v}{\mathsf{op}_\Omega(\mathsf{D}) \xrightarrow{U} \mathsf{op}_\Omega(\mathsf{H})} \quad \begin{cases} \forall v \in T_\Omega\ \forall i \leq k_v \;:\; (\mathsf{lab}(U_v^i), \alpha_v^i) \in \lambda_\Omega(v) \\ U \;=\; \bigcup_{v \in T_\Omega} \{(v, \alpha_v^1) \lhd U_v^1, \ldots, (v, \alpha_v^{k_v}) \lhd U_v^{k_v}\} \\ U \text{ is finite.} \end{cases}} \tag{54}$$

The finiteness of U means that finitely many k_v's are non-zero. Notice that for the other ones we do not require that $D_v = H_v$ but only that $D_v \xrightarrow{\emptyset} H_v$; it will follow from proposition 8.6 that $D_v \equiv H_v$.

Since skip and redo never occur in any $\lambda_\Omega(v)$, the last rule will never be applied if any of the sets U_v^i contains skip or redo. Similarly, it will never generate any of these two special labels. Thus, if $D \xrightarrow{U} H$ and $\mathsf{skip} \in U$ then $U = \{\mathsf{skip}\}$, and if $D \xrightarrow{U} H$ and $\mathsf{redo} \in U$ then $U = \{\mathsf{redo}\}$.

Infinite operators The meaning of the operational semantics rules is again clear if all boxes in OpBox have finite transition sets; in general, however, we need to treat infinite operator boxes as well. Following what is now an established practice, we consider the domain $(2^{\mathsf{ExprTrees}\times\mathsf{tlab}_{\mathsf{sr}}\times\mathsf{ExprTrees}}, \subseteq)$ which forms a complete lattice with the $\cap$ and $\cup$ operations. And the inference rules (50)–(54) define a monotonic mapping

$$opsem \;:\; 2^{\mathsf{ExprTrees}\times\mathsf{tlab}_{\mathsf{sr}}\times\mathsf{ExprTrees}} \to 2^{\mathsf{ExprTrees}\times\mathsf{tlab}_{\mathsf{sr}}\times\mathsf{ExprTrees}}$$

whose least fixed point, $\longrightarrow$, is equal to $opsem^{\alpha''}(\emptyset)$, for some ordinal α''. We also define, for every $(D,U,H) \in\longrightarrow$, its *rank*, denoted by $\mathsf{rank}(D,U,H)$, defined as the smallest ordinal β such that $(D,U,H) \in opsem^{\beta}(\emptyset)$. Thus we are in a position to apply (transfinite) induction on the rank of the triples in $\longrightarrow$. If all operator boxes in OpBox have finite transition sets, the standard induction can be used.

Properties An empty move always relates two structurally equivalent expressions.

Proposition 8.6. If D and H are expressions such that $D \xrightarrow{\emptyset} H$ then $D \equiv H$.
Proof: Follows by a straightforward induction on $\mathsf{rank}(D,\emptyset,H)$. □

A move of the operational semantics transforms a box expression into another box expression with structurally equivalent underlying static expression, and the move generated is a valid step for the corresponding boxes. We interpret this as establishing the soundness of the operational semantics of box expressions.

Theorem 8.7. Let D be a well formed box expression and $D \xrightarrow{U} H$. Then H is a well formed box expression and $\mathsf{box}(D)_{\mathsf{sr}}\ [U\rangle\ \mathsf{box}(H)_{\mathsf{sr}}$.
Proof: The proof proceeds by induction on $\mathsf{rank}(D,U,H)$. In the base step, we consider three cases.

Case 1: $D = \overline{E} \xrightarrow{\{\mathsf{skip}\}} \underline{E} = H$ or $D = \underline{E} \xrightarrow{\{\mathsf{redo}\}} \overline{E} = H$ where E is a well formed static expression. Then both D and H are well formed box expressions and

$$\mathsf{box}(\overline{E})_{\mathsf{sr}}\ [\{\mathsf{skip}\}\rangle\ \mathsf{box}(\underline{E})_{\mathsf{sr}} \quad \text{or} \quad \mathsf{box}(\underline{E})_{\mathsf{sr}}\ [\{\mathsf{redo}\}\rangle\ \mathsf{box}(\overline{E})_{\mathsf{sr}}$$

follows from $\mathsf{box}(\overline{E})_{\mathsf{sr}} = \overline{\mathsf{box}(E)_{\mathsf{sr}}}$ and $\mathsf{box}(\underline{E})_{\mathsf{sr}} = \underline{\mathsf{box}(E)_{\mathsf{sr}}}$ and the definition of Σ_{sr}.

Case 2: $D \equiv H$ and $U = \emptyset$. Then, by theorems 8.1 and 8.5, H is a well formed box expression and $\mathsf{box}(D) = \mathsf{box}(H)$. Hence $\mathsf{box}(D)_{\mathsf{sr}}\ [\emptyset\rangle\ \mathsf{box}(H)_{\mathsf{sr}}$.

Case 3: D and H are flat expressions and $\mathsf{box}(D)\ [U\rangle\ \mathsf{box}(H)$. Then there is nothing to prove.

In the inductive step we consider two cases.

Case 1: There is J such that $D \xrightarrow{\emptyset} J \xrightarrow{U} H$ and $\max\{\mathsf{rank}(D,\emptyset,J), \mathsf{rank}(J,U,H)\} < \mathsf{rank}(D,U,H)$. (If $D \xrightarrow{U} J \xrightarrow{\emptyset} H$ then we proceed similarly.) By the induction hypothesis, both J and H are well formed box expressions and $\mathsf{box}(D)_{\mathsf{sr}}\ [\emptyset\rangle\ \mathsf{box}(J)_{\mathsf{sr}}\ [U\rangle\ \mathsf{box}(H)_{\mathsf{sr}}$. Hence, by $\mathsf{box}(D) = \mathsf{box}(J)$, $\mathsf{box}(D)_{\mathsf{sr}}\ [U\rangle\ \mathsf{box}(H)_{\mathsf{sr}}$.

Case 2: $D = \mathsf{op}_\Omega(\mathsf{D})$ and $H = \mathsf{op}_\Omega(\mathsf{H})$ where $\mathsf{op}_\Omega(\mathsf{D})$, $\mathsf{op}_\Omega(\mathsf{H})$ and U are as in (54); moreover, for every $v \in T_\Omega$, $\mathsf{rank}(D_v,U_v,H_v) < \mathsf{rank}(D,U,H)$. From D being a well formed box expression, the induction hypothesis (which, in particular, implies that for each v, $\mathsf{box}(D_v)\ [\{(v,\alpha_v^1) \lhd U_v^1, \ldots, (v,\alpha_v^{k_v}) \lhd U_v^{k_v}\}\rangle\ \mathsf{box}(H_v)$ since no label α_v^i can belong to $\mathsf{skip}, \mathsf{redo}$), and theorem 7.1, it follows that D and H are Ω–tuples of well formed box expressions, $\mathsf{box}(\mathsf{D}) \in \mathsf{dom}_\Omega$, and

$$\Big(\Omega\ :\ \mathsf{box}(\mathsf{D})\Big) \xrightarrow{U} \Big(\Omega\ :\ \mathsf{box}(\mathsf{H})\Big).$$

Hence, by theorem 4.1, $\mathsf{box}(\mathsf{H}) \in \mathsf{dom}_\Omega$ and $\Omega(\mathsf{box}(\mathsf{D}))\ [U\rangle\ \Omega(\mathsf{box}(\mathsf{H}))$; notice that the finiteness of the set U is needed to reach this conclusion. Moreover, by $\mathsf{box}(\mathsf{H}) \in \mathsf{dom}_\Omega$ and theorem 7.1, H is a well formed box expression. Thus $\mathsf{box}(D) = \mathsf{box}(\mathsf{op}_\Omega(\mathsf{D}))\ [U\rangle\ \mathsf{box}(\mathsf{op}_\Omega(\mathsf{H})) = \mathsf{box}(H)$. □

The next result states completeness of the operational semantics of box expressions.

Theorem 8.8. Let D be a well formed box expression and $\mathsf{box}(D)_{\mathsf{sr}}\ [U\rangle\ \Sigma_{\mathsf{sr}}$. Then there is a well formed box expression H such that $\mathsf{box}(H) = \Sigma$ and $D \xrightarrow{U} H$.
Proof: If $U = \emptyset$ then, by (51) and $\Sigma = \mathsf{box}(D)$, the result holds for $H = D$. If $U \neq \emptyset$ then we first recall that either $U = \{\mathsf{skip}\}$, or $U = \{\mathsf{redo}\}$, or $U \subseteq \mathsf{T}^{box}_{\mathsf{tree}}$.

Suppose $U = \{\mathsf{skip}\}$ (the case $U = \{\mathsf{redo}\}$ is similar). Then $\mathsf{box}(D) \in \mathsf{Box}^e$ and $\Sigma = \underline{\mathsf{box}(\lfloor D \rfloor)}$. By theorem 7.1, $D \in \mathsf{Expr}^e$. Hence, by lemma 8.3, $D \equiv \overline{\lfloor D \rfloor}$. Thus, by (50) and (51),

$$D \xrightarrow{\emptyset} \overline{\lfloor D \rfloor} \xrightarrow{\{\mathsf{skip}\}} \underline{\lfloor D \rfloor}.$$

Hence, by (52), $D \xrightarrow{\{\mathsf{skip}\}} \underline{\lfloor D \rfloor}$ and we can take $H = \underline{\lfloor D \rfloor}$ since

$$\mathsf{box}(H) = \mathsf{box}(\underline{\lfloor D \rfloor}) = \underline{\mathsf{box}(\lfloor D \rfloor)} = \underline{\mathsf{box}(D)} = \Sigma.$$

If $U \subseteq \mathsf{T}^{box}_{\mathsf{tree}}$ then $\mathsf{box}(D)\ [U\rangle\ \Sigma$ and we proceed by induction on the maximal depth h of the transition trees in U (such an inductive argument is valid since U is a finite set of finite trees).

In the base step ($h = 0$), by proposition 7.2(2), D is a flat expression. Hence, from (38), (39) and (53) it follows that there is an axiom $D \xrightarrow{U} H$, where H is a flat expression such that $\mathsf{box}(D)\ [U\rangle\ \mathsf{box}(H)$. The latter further implies that $\mathsf{box}(H) = \Sigma$.

In the inductive step ($h > 0$), by proposition 7.2(2)), we may assume that $D = \mathsf{op}_\Omega(\mathsf{D})$, for some Ω and D. Hence, by theorem 7.1, we have $\mathsf{box}(\mathsf{D}) \in \mathsf{dom}_\Omega$ and $\Omega(\mathsf{box}(\mathsf{D}))\ [U\rangle\ \Sigma$. From theorem 4.4 it follows that there are Ω–tuples in the domain of Ω, Θ and Ψ, such that $\mathsf{box}(\mathsf{D}) \equiv_\Omega \Theta$ and $\Sigma = \Omega(\Psi)$ and

$$\Big(\Omega\ :\ \Theta\Big) \xrightarrow{U} \Big(\Omega\ :\ \Psi\Big). \tag{55}$$

We next observe that there is a well formed box expression $\mathsf{op}_\Omega(\mathsf{F})$ such that $\mathsf{op}_\Omega(\mathsf{D}) \equiv \mathsf{op}_\Omega(\mathsf{F})$ and $\mathsf{box}(\mathsf{F}) = \Theta$. That $\mathsf{op}_\Omega(\mathsf{F})$ exists can be shown in the following way. Let μ and κ be respectively the factorisations of $\mathsf{box}(\mathsf{D})$ and Θ which are factorisations of the same complex marking of Ω (notice that, by theorem 7.1, μ is also the factorisation of D). By lemma 8.3 and (49), we may assume $\mathsf{D}_{\mu_e} \subseteq \overline{\mathsf{Expr}^s}$ and $\mathsf{D}_{\mu_x} \subseteq \underline{\mathsf{Expr}^s}$. Let C be the Ω–tuple of well formed expressions such that, for every $t \in T_\Omega$: $D_v = \overline{C_v}$ if $v \in \mu_e$; $D_v = \underline{C_v}$ if $v \in \mu_x$; and $D_v = C_v$ otherwise.
Then, by (46), $\mathsf{op}_\Omega(\mathsf{D}) \equiv \mathsf{op}_\Omega(\mathsf{F})$ where, for every $v \in T_\Omega$: $F_v = \overline{C_v}$ if $v \in \kappa_e$; $F_v = \underline{C_v}$ if $v \in \kappa_x$; and $F_v = C_v$ otherwise. Hence, by theorem 8.1, $\mathsf{op}_\Omega(\mathsf{F})$ is a well formed box expression which satisfies $\mathsf{box}(\mathsf{F}) = \Theta$. Indeed, $\lfloor \mathsf{box}(\mathsf{F}) \rfloor = \lfloor \mathsf{box}(\mathsf{D}) \rfloor = \lfloor \Theta \rfloor$, the factorisation of F is κ, and, by $\mu_d = \kappa_d$, $\mathsf{box}(\mathsf{F}_{\kappa_d}) = \Theta_{\kappa_d}$.

Define, for every $v \in T_\Omega$, $U_v = U \cap \mathsf{T}^v_{\mathsf{new}} = \{u^1_v, \ldots, u^{k_v}_v\}$. By (55), we have, for every $v \in T$,

$$\Theta_v\ \Big[\mathsf{trees}(u^1_v) \uplus \cdots \uplus \mathsf{trees}(u^{k_v}_v)\Big\rangle\ \Psi_v$$

and only finitely many k_v are greater than zero. Thus, by the induction hypothesis and the $U = \emptyset$ case, for every $v \in T$, there is a well formed box expression H_v such that $\mathsf{box}(H_v) = \Psi_v$ and

$$F_v \xrightarrow{\ \mathsf{trees}(u^1_v)\ \uplus \cdots \uplus \mathsf{trees}(u^{k_v}_v)\ } H_v.$$

Hence, by (54), $\mathsf{op}_\Omega(\mathsf{F}) \xrightarrow{U} \mathsf{op}_\Omega(\mathsf{H})$. And, by theorem 8.7, we obtain that $H = \mathsf{op}_\Omega(\mathsf{H})$ is a well formed box expression such that $\mathsf{box}(\mathsf{op}_\Omega(\mathsf{F}))\ [U\rangle\ \mathsf{box}(H)$. Thus, by theorem 8.5 and $D \equiv \mathsf{op}_\Omega(\mathsf{F})$, we have $\mathsf{box}(D)\ [U\rangle\ \mathsf{box}(H)$, and so $\mathsf{box}(H) = \Sigma$. □

Example The operational semantics of flat expressions is given in DIY by:

$\overline{c_1} \xrightarrow{\{t_{11}\}} c_{12}$	$\overline{c_2} \xrightarrow{\{t_{21}\}} c_{22}$	$\overline{c_1} \xrightarrow{\{t_{12}\}} c_{11}$	$c_{21} \xrightarrow{\{t_{21}\}} c_{22}$	$\overline{c_1} \xrightarrow{\{t_{11},t_{12}\}} \underline{c_1}$
$c_{22} \xrightarrow{\{t_{22}\}} \underline{c_2}$	$c_{12} \xrightarrow{\{t_{12}\}} \underline{c_1}$	$c_{22} \xrightarrow{\{t_{22}\}} c_{23}$	$c_{11} \xrightarrow{\{t_{11}\}} \underline{c_1}$	$\overline{c_3} \xrightarrow{\{t_{31}\}} \underline{c_3}$

and the inference rule for the only operator box, Ω_0, can be formulated in the following way:

$$\boxed{\dfrac{D \xrightarrow{\{t_1,u_1,\ldots,t_k,u_k,x_1,\ldots,x_l\}} D' \quad , \quad H \xrightarrow{\{y_1,\ldots,y_m\}} H' \quad , \quad G \xrightarrow{\{z_1,\ldots,z_n\}} G'}{\mathsf{op}_{\Omega_0}(D,G,H) \xrightarrow{U} \mathsf{op}_{\Omega_0}(D',G',H')}}$$

where $k,l,m,n \geq 0$; $\{\mathsf{lab}(t_i),\mathsf{lab}(u_i)\} = \{a,b\}$, for every $i \leq k$; $\mathsf{lab}(x_i) \notin \{a,b\}$, for every $i \leq l$; and the step U is given by:

$$\begin{aligned} U = & \Big\{\ (v_1,f) \lhd \{t_1,u_1\}\ ,\ \ldots\ ,\ \ (v_1,f) \lhd \{t_k,u_k\}\ \Big\} \cup \\ & \Big\{\ (v_1,\mathsf{lab}(x_1)) \lhd x_1\ ,\ \ldots\ ,\ \ (v_1,\mathsf{lab}(x_l)) \lhd x_l\ \Big\} \cup \\ & \Big\{\ (v_2,\mathsf{lab}(y_1)) \lhd y_1\ ,\ \ldots\ ,(v_2,\mathsf{lab}(y_m)) \lhd y_m\ \Big\} \cup \\ & \Big\{\ (v_3,\mathsf{lab}(z_1)) \lhd z_1\ ,\ \ldots\ ,\ \ (v_3,\mathsf{lab}(z_n)) \lhd z_n\ \Big\} \end{aligned}$$

One can show that if the above rule is applied to a box expression then $n \cdot (k+l+m) = 0$ (i.e., either n is equal to zero, or k, l and m are all equal to zero). For example, the following is a valid sequence of three moves

$$\overline{\mathsf{op}_{\Omega_0}(c_1,c_2,c_3)} \xrightarrow{\{w_1,w_2\}} \mathsf{op}_{\Omega_0}(\underline{c_1},c_{22},c_3) \xrightarrow{\{w_3\}} \mathsf{op}_{\Omega_0}(c_1,c_2,\overline{c_3}) \xrightarrow{\{w_4\}} \underline{\mathsf{op}_{\Omega_0}(c_1,c_2,c_3)}.$$

A derivation for the first move is shown below.

$$\overbrace{\overline{\mathsf{op}_{\Omega_0}(c_1,c_2,c_3)} \xrightarrow{\emptyset} \mathsf{op}_{\Omega_0}(\overline{c_1},\overline{c_2},c_3)\ \ ,\ \underbrace{\mathsf{op}_{\Omega_0}(\overline{c_1},\overline{c_2},c_3) \xrightarrow{\{\ (v_1,f)\lhd\{t_{11},t_{12}\}\ ,\ (v_2,c)\lhd t_{21}\ \}} \mathsf{op}_{\Omega_0}(\underline{c_1},c_{22},c_3)}_{\overline{c_1} \xrightarrow{\{t_{11},t_{12}\}} \underline{c_1} \qquad \overline{c_2} \xrightarrow{\{t_{21}\}} c_{22} \qquad c_3 \xrightarrow{\emptyset} c_3}}^{\overline{\mathsf{op}_{\Omega_0}(c_1,c_2,c_3)} \xrightarrow{\{w_1,w_2\}} \mathsf{op}_{\Omega_0}(\underline{c_1},c_{22},c_3)}$$

8.3 Consistency of the denotational and operational semantics

The consistency between the denotational and operational semantics of box expressions will be formulated in terms of the transition systems they generate. This will be possible since, thanks to theorems 8.7 and 8.8, we are now in a position to compare transition systems generated by a well formed box expression and the corresponding net.

Transition systems of box expressions and the corresponding boxes Let G be a well formed dynamic expression. We will use $[G\rangle_{\mathsf{sr}}$ to denote the least set of expressions containing G such that if $H \in [G\rangle_{\mathsf{sr}}$ and $H \xrightarrow{U} C$, for some $U \in \mathsf{tlab}_{\mathsf{sr}}$, then $C \in [G\rangle_{\mathsf{sr}}$. The *full transition system* of G is defined as $\mathsf{fts}_G = (V, L, A, [G]_\equiv)$ where $V = \{[H]_\equiv \mid H \in [G\rangle_{\mathsf{sr}}\}$ is the set of states, $L = \mathsf{tlab}_{\mathsf{sr}}$ is the set of arc labels, and $A = \{([H]_\equiv, U, [C]_\equiv) \in V \times \mathsf{tlab}_{\mathsf{sr}} \times V \mid H \xrightarrow{U} C\}$ is the set of arcs. For a static expression E, $\mathsf{fts}_E = \mathsf{fts}_{\overline{E}}$.

Let Σ be a box generated by a well formed dynamic expression. The *full transition system* of Σ is defined as $\mathsf{fts}_\Sigma = (V, L, A, v_0)$ where $V = \{\Theta \mid \mathsf{skip}, \mathsf{redo} \notin T_\Theta \wedge \Theta_{\mathsf{sr}} \in [\Sigma_{\mathsf{sr}}\rangle\}$ is the set of states, $v_0 = \Sigma$ is the initial state, $L = \mathsf{tlab}_{\mathsf{sr}}$ is the set of arc labels, and $A = \{(\Theta, U, \Psi) \in V \times \mathsf{tlab}_{\mathsf{sr}} \times V \mid \Theta_{\mathsf{sr}}\ [U\rangle\ \Psi_{\mathsf{sr}}\}$ is the set of arcs. In other words, fts_Σ is the reachability graph of Σ_{sr} with all references to skip and redo in the nodes of the graph erased. For a box Σ generated by a static expression, $\mathsf{fts}_\Sigma = \mathsf{fts}_{\overline{\Sigma}}$.

Consistency result We now state a fundamental result which demonstrates that the operational and denotational semantics of a well formed box expression capture the same behaviour, in arguably the strongest sense.

Theorem 8.9. For every well formed box expression D, $\mathsf{iso} = \{([H]_\equiv, \mathsf{box}(H)) \mid [H]_\equiv \text{ is a node of } \mathsf{fts}_D\}$ is an isomorphism between the full transition systems fts_D and $\mathsf{fts}_{\mathsf{box}(D)}$.
Proof: By theorems 8.5 and 8.7, iso is a well defined injective mapping. The rest of the result follows from theorems 8.7, 8.8 and 7.1. □

8.4 Label based operational semantics

The operational semantics based on transition names and captured by full transition systems is very expressive; in particular, we will see in section 9 that it contains enough information to retrieve partial order semantics of nets corresponding to box expressions. However, it may often be sufficient to record only the labels of executed transitions, in the usual style of process algebras. Such a treatment can be accommodated within the scheme developed so far. First, we retain the structural similarity relation $\equiv$ on box expressions without any change. Next, we define moves of the form $D \xrightarrow{\Gamma} H$ where D and H are expressions as before, and Γ is a finite multiset in $\mathsf{mlab_{sr}} = \mathsf{mult}(\mathsf{Lab} \cup \{\mathsf{skip}, \mathsf{redo}\})$. We keep the rules (50), (51) and (52) unchanged (with $\emptyset$ now denoting the empty multiset of labels, $\{\mathsf{skip}\}$ and $\{\mathsf{redo}\}$ denoting singleton multisets, and U being changed to Γ), but modify the axioms for flat expressions and inference rules for operator boxes. For all flat expressions D and H and non-empty $\Gamma \in \mathsf{mlab_{sr}}$ such that $\mathsf{box}(D)\ [\Gamma\rangle_{\mathsf{lab}}\ \mathsf{box}(H)$,

$$\boxed{D \xrightarrow{\Gamma} H} \tag{56}$$

and for every operator box Ω:

$$\boxed{\frac{\forall v \in T_\Omega\ :\ D_v \xrightarrow{\Gamma_v^1 + \cdots + \Gamma_v^{k_v}} H_v}{\mathsf{op}_\Omega(\mathsf{D}) \xrightarrow{\Gamma} \mathsf{op}_\Omega(\mathsf{H})} \quad \begin{cases} \forall v \in T_\Omega\ \forall i \le k_v\ :\ (\Gamma_v^i, \alpha_v^i) \in \lambda_\Omega(v) \\ \Gamma = \sum_{v \in T_\Omega} \{\alpha_v^1\} + \cdots + \{\alpha_v^{k_v}\} \\ \Gamma \text{ is finite} \end{cases}} \tag{57}$$

where D and H are Ω–tuples of expressions.

Properties The two types of operational semantics are clearly related; in essence, each label based move is a transition based move with only transitions labels being recorded.

Proposition 8.10. Let D be a well formed box expression and $\Gamma \in \mathsf{mlab_{sr}}$. Then $D \xrightarrow{\Gamma} H$ if and only if there is $U \in \mathsf{tlab_{sr}}$ such that $D \xrightarrow{U} H$ and $\mathsf{lab}(U) = \Gamma$.
Proof: Both implications can be proved by a straightforward induction on the rank of $D \xrightarrow{\Gamma} H$ and $D \xrightarrow{U} H$, respectively. □

The results concerning transition based operational semantics directly extend to the label based one. Let D be a well formed box expression. In view of proposition 8.10, the label based operational semantics of D is faithfully captured by the *transition system* of D, denoted by ts_D and defined as fts_D with each arc label U changed to $\mathsf{lab}(U)$. The consistency result for the label based operational semantics can then be formulated thus.

Theorem 8.11. For every well formed box expression D, $\mathsf{iso} = \{([H]_\equiv, \mathsf{box}(H)) \mid [H]_\equiv \text{ is a node of } \mathsf{ts}_D\}$ is an isomorphism between the transition systems ts_D and $\mathsf{ts}_{\mathsf{box}(D)}$.
Proof: Follows from theorem 8.9, proposition 8.10, and the fact that the structural similarity relation is the same for the transition and label based operational semantics. □

Example For the DIY algebra, the label based operational semantics of flat expressions is given by:

$\overline{c_1} \xrightarrow{\{a\}} c_{12}$	$\overline{c_2} \xrightarrow{\{c\}} c_{22}$	$\overline{c_1} \xrightarrow{\{b\}} c_{11}$	$c_{21} \xrightarrow{\{c\}} c_{22}$	$\overline{c_1} \xrightarrow{\{a,b\}} \underline{c_1}$
$c_{22} \xrightarrow{\{d\}} \underline{c_2}$	$c_{12} \xrightarrow{\{b\}} \underline{c_1}$	$c_{22} \xrightarrow{\{d\}} c_{23}$	$c_{11} \xrightarrow{\{a\}} \underline{c_1}$	$\overline{c_3} \xrightarrow{\{e\}} \underline{c_3}$

and the inference rule for the only operator box can be formulated in the following way:

$$\boxed{\dfrac{D_1 \xrightarrow{k\cdot\{a,b\}+\Gamma_1} H_1 \quad , \quad D_2 \xrightarrow{\Gamma_2} H_2 \quad , \quad D_3 \xrightarrow{\Gamma_3} H_3}{\mathsf{op}_{\Omega_0}(D_1, D_2, D_3) \xrightarrow{k\cdot\{f\}+\Gamma_1+\Gamma_2+\Gamma_3} \mathsf{op}_{\Omega_0}(H_1, H_2, H_3)} \qquad a, b \notin \Gamma_1}$$

An application of these rules is illustrated on the move $\overline{\mathsf{op}_{\Omega_0}(c_1, c_2, c_3)} \xrightarrow{\{c,f\}} \mathsf{op}_{\Omega_0}(\underline{c_1}, c_{22}, c_3)$ for which a derivation looks as follows:

$$\begin{array}{c} \overline{\mathsf{op}_{\Omega_0}(c_1, c_2, c_3)} \xrightarrow{\{c,f\}} \mathsf{op}_{\Omega_0}(\underline{c_1}, c_{22}, c_3) \\ \overbrace{\overline{\mathsf{op}_{\Omega_0}(c_1, c_2, c_3)} \xrightarrow{\emptyset} \mathsf{op}_{\Omega_0}(\overline{c_1}, \overline{c_2}, c_3) \quad , \quad \begin{array}{c} \mathsf{op}_{\Omega_0}(\overline{c_1}, \overline{c_2}, c_3) \xrightarrow{\{c,f\}} \mathsf{op}_{\Omega_0}(\underline{c_1}, c_{22}, c_3) \\ \overbrace{\overline{c_1} \xrightarrow{\{a,b\}} \underline{c_1} \qquad \overline{c_2} \xrightarrow{\{c\}} c_{22} \qquad c_3 \xrightarrow{\emptyset} c_3} \end{array}} \end{array}$$

9 Partial order semantics of box expressions

Process expressions do not support an explicit notion of a place. We will therefore use Mazurkiewicz traces — a model of partial order behaviour based solely on transitions.

Mazurkiewicz traces Let A be a set and $\mathsf{ind} \subseteq A \times A$ be an irreflexive symmetric relation on A. Intuitively, A is meant to represent the set of possible events in a concurrent system, and ind an *independence* relation which identifies events that can be executed concurrently. With every sequence $\sigma = A_1 \ldots A_k$, where each A_i is a finite subset of A such that $(a, b) \in \mathsf{ind}$ for all distinct $a, b \in A_i$, one can associate a partial order, denoted by $\mathsf{poset}_{\mathsf{ind}}(\sigma)$, in the following way.

The set of *event occurrences* of σ, occ_σ, comprises all pairs $(a, l) \in A \times \mathbf{N}$ such that $a \in A_1 \cup \ldots \cup A_k$ and l ranges between 1 and the number of times a occurs within σ, $1 \leq l \leq |\{i \mid a \in A_i\}|$. Moreover, we denote by $\mathsf{idx}_{(a,l)}$ the index m such that A_m contains the l-th occurrence of a, $a \in A_m$ and $l = |\{i \mid a \in A_i \wedge i \leq m\}|$. We then define a *precedence* relation on occ_σ, $\prec_\sigma$, by stipulating that $(a, l) \prec_\sigma (b, n)$ if $(a, b) \notin \mathsf{ind}$ and $\mathsf{idx}_{(a,l)} < \mathsf{idx}_{(b,n)}$. Then $\mathsf{poset}_{\mathsf{ind}}(\sigma) = (\mathsf{occ}_\sigma, \prec_\sigma^*)$ where $\prec_\sigma^*$ is the reflexive transitive closure of $\prec_\sigma$.

Consider now a safe labelled net Σ, and take A to be its transition set, $A = T_\Sigma$, and ind to be its independence relation, $\mathsf{ind} = \mathsf{ind}_\Sigma$. Then, with every finite step sequence σ of Σ, $\Sigma\ [\sigma\rangle\ \Theta$, we can associate a partial order, $\mathsf{poset}_{\mathsf{ind}_\Sigma}(\sigma)$, in the way described above. One of the crucial properties of the partial order semantics is that every sequence of sets of transitions consistent with $\mathsf{poset}_{\mathsf{ind}_\Sigma}(\sigma)$ is also a valid step sequence. More precisely, if $\omega = U_1 \ldots U_k$ is a sequence of finite sets of transitions of Σ such that $(t, u) \in \mathsf{ind}_\Sigma$, for every U_i and all distinct $t, u \in U_i$, then $\mathsf{poset}_{\mathsf{ind}_\Sigma}(\omega) = \mathsf{poset}_{\mathsf{ind}_\Sigma}(\sigma)$ implies that ω is a valid step sequence for Σ leading to the same marked net as σ, i.e., $\Sigma\ [\omega\rangle\ \Theta$. For example, the step sequence $\sigma = \{t_0, t_2\}\emptyset\{t_1, t_2\}\{t_2\}\emptyset$ of the labelled net Σ_0 in figure 1 generates a partial order $\mathsf{poset}_{\mathsf{ind}_{\Sigma_0}}(\sigma)$ whose Hasse diagram, giving the precedence relation, is shown on the left-hand side of figure 8. This partial order is normally represented as a *labelled* partial order with nodes being labelled just by transitions, as shown on the right-hand side of figure 8.

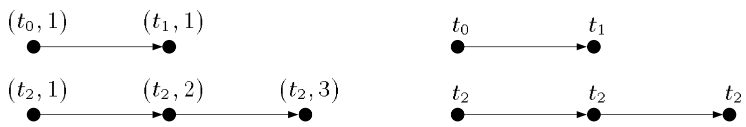


Figure 8. Hasse diagrams of partial orders.

The presence of a path between two nodes in such a graph is interpreted as *causality*, and the lack of ordering as *concurrency*. Thus, in the example above, the occurrences of t_0 and t_1 are causally related, and both are concurrent with the three occurrences of t_2. Whenever $\Sigma\ [\sigma\rangle\ \Theta$, we will write $\Sigma\ [\mathsf{poset}_{\mathsf{ind}_\Sigma}(\sigma)\rangle_{\mathsf{po}}\ \Theta$ to indicate that Θ can be derived from Σ by executing the poset within the brackets $[\cdots\rangle_{\mathsf{po}}$.

Connectedness properties of transition trees We will now investigate connectedness between transitions in nets generated by box expressions. It will turn out that the name trees of these transitions contain an information as to when two transitions are adjacent to a common place, and the relevant information is independent of the specific net to which the two transitions happen to belong.

We now can again take advantage of the assumption made in (31) that the operator and static constant boxes have disjoint transition sets. This allows one to identify, for every transition t in the boxes $\mathsf{OpBox} \cup (\mathsf{ConstBox} \cap \mathsf{Box}^s)$, a unique box in $\mathsf{OpBox} \cup (\mathsf{ConstBox} \cap \mathsf{Box}^s)$ to which t belongs; we will denote it by net_t. For example, in the DIY algebra, $\mathsf{net}_{v_1} = \Omega_0$ and $\mathsf{net}_{t_{22}} = \Phi_2$.

To start with, we can capture the property of 'being connected to an entry or exit place'. By definition, a transition tree $t \in \mathsf{T}^{box}_{\mathsf{tree}}$ belongs to the set Con^e or Con^x, if one of the following hold:[19]

- There is a plain box $\Sigma \in \mathsf{ConstBox} \cap \mathsf{Box}^s$ such that $t \in T_\Sigma$ and we have, respectively:

$$({}^\bullet t \cup t^\bullet) \cap {}^\circ\Sigma \neq \emptyset \quad \text{or} \quad ({}^\bullet t \cup t^\bullet) \cap \Sigma^\circ \neq \emptyset.$$

- There is an operator box $\Omega \in \mathsf{OpBox}$ such that $t = (v, \alpha) \lhd Q$ and $v \in T_\Omega$, and we have, respectively:

$$\begin{gathered} {}^\bullet v \cap {}^\circ\Omega \neq \emptyset \neq Q \cap Con^e \ \vee\ v^\bullet \cap {}^\circ\Omega \neq \emptyset \neq Q \cap Con^x \ \text{ or} \\ {}^\bullet v \cap \Omega^\circ \neq \emptyset \neq Q \cap Con^e \ \vee\ v^\bullet \cap \Omega^\circ \neq \emptyset \neq Q \cap Con^x. \end{gathered}$$

Notice also that, by (31), we have:

$$\Sigma = \mathsf{net}_t \quad \text{and} \quad \Omega = \mathsf{net}_v. \tag{58}$$

Proposition 9.1. Let E be a static expression and t be a transition in $\mathsf{box}(E)$. Then

$$({}^\bullet t \cup t^\bullet) \cap {}^\circ\mathsf{box}(E) \neq \emptyset \iff t \in Con^e \quad \text{and} \quad ({}^\bullet t \cup t^\bullet) \cap \mathsf{box}(E)^\circ \neq \emptyset \iff t \in Con^x.$$

Proof: We only prove the first equivalence, the proofs of the other one being symmetric. We proceed by induction on the depth h of t. In the base step ($h = 0$), by proposition 7.2(2) and (31), $\mathsf{box}(E) = \mathsf{net}_t \in \mathsf{ConstBox} \cap \mathsf{Box}^s$. Hence the equivalence holds by the definition of Con^e and (58).

In the inductive step ($h > 0$), we may assume that $E = \mathsf{op}_\Omega(\mathsf{E})$, by proposition 7.2(2), which means that $t = (v, \alpha) \lhd Q$; and so, by (31), $\Omega = \mathsf{net}_v$. By proposition 3.9, $({}^\bullet t \cup t^\bullet) \cap {}^\circ\mathsf{box}(E) \neq \emptyset$ if and only if

$${}^\bullet v \cap {}^\circ\Omega \neq \emptyset \neq ({}^\bullet Q \cup Q^\bullet) \cap {}^\circ\mathsf{box}(E_v) \quad \text{or} \quad v^\bullet \cap {}^\circ\Omega \neq \emptyset \neq ({}^\bullet Q \cup Q^\bullet) \cap \mathsf{box}(E_v)^\circ.$$

Hence, the equivalence holds by $\Omega = \mathsf{net}_v$, (58) and the induction hypothesis. □

Global independence relation By definition, a pair of transition trees $(t, u) \in \mathsf{T}^{box}_{\mathsf{tree}} \times \mathsf{T}^{box}_{\mathsf{tree}}$ belongs to the set ind^{box} if one of the following hold:

- There is a plain box $\Sigma \in \mathsf{ConstBox} \cap \mathsf{Box}^s$ such that $(t, u) \in \mathsf{ind}_\Sigma$.
- There is an operator box $\Omega \in \mathsf{OpBox}$ such that $t = (v, \alpha) \lhd Q$ and $u = (w, \beta) \lhd R$ and $v, w \in T_\Omega$, and we have:

$$\begin{aligned} {}^\bullet v \cap w^\bullet \neq \emptyset &\implies Q \cap Con^e = \emptyset \ \vee\ R \cap Con^x = \emptyset \\ v^\bullet \cap {}^\bullet w \neq \emptyset &\implies Q \cap Con^x = \emptyset \ \vee\ R \cap Con^e = \emptyset \\ v \neq w \wedge {}^\bullet v \cap {}^\bullet w \neq \emptyset &\implies Q \cap Con^e = \emptyset \ \vee\ R \cap Con^e = \emptyset \\ v \neq w \wedge v^\bullet \cap w^\bullet \neq \emptyset &\implies Q \cap Con^x = \emptyset \ \vee\ R \cap Con^x = \emptyset. \end{aligned}$$

Moreover, if $v = w$ then $Q \times R \subseteq \mathsf{ind}^{box}$.

[19] The sets Con^e and Con^e are well defined since transition trees are always finite.

Notice also that similarly as before, by (31), we have:

$$\Sigma = \mathsf{net}_t \quad \text{and} \quad \Omega = \mathsf{net}_v = \mathsf{net}_w. \tag{59}$$

The relation we just defined can be thought of as a global independence relation for the boxes generated by the denotational semantics of box expressions.

Theorem 9.2. For every static expression E, $\mathsf{ind}_{\mathsf{box}(E)} = (T_{\mathsf{box}(E)} \times T_{\mathsf{box}(E)}) \cap \mathsf{ind}^{box}$.
Proof: Let t and u be transition trees in $\mathsf{box}(E)$. We proceed by induction on $h = \max\{\mathsf{depth}(t), \mathsf{depth}(u)\}$.

In the base step ($h = 0$), by proposition 7.2(2) and (31), $\mathsf{box}(E) = \mathsf{net}_t = \mathsf{net}_u \in \mathsf{ConstBox} \cap \mathsf{Box}^s$. Hence, by the definition of ind^{box} and (59), $(t, u) \in \mathsf{ind}_{\mathsf{box}(E)}$ if and only if $(t, u) \in \mathsf{ind}^{box}$.

In the inductive step ($h > 0$), we may assume that $E = \mathsf{op}_\Omega(\mathsf{E})$, by proposition 7.2(2). Thus $t = (v, \alpha) \lhd Q$ and $u = (w, \beta) \lhd R$, and so, by (31), $\Omega = \mathsf{net}_v = \mathsf{net}_w$ We now consider two cases.

Case 1: $v = w$. We have $Q \cup R \subseteq \mathsf{box}(E_v)$. By proposition 3.7, $(t, u) \in \mathsf{ind}_{\mathsf{box}(E)}$ if and only if $Q \times R \subseteq \mathsf{ind}_{\mathsf{box}(E_v)}$ and, moreover, if ${}^\bullet v \cap v^\bullet \neq \emptyset$ then (also by proposition 9.1):

$$(Q \cap Con^e = \emptyset \ \vee \ R \cap Con^x = \emptyset) \ \wedge \ (Q \cap Con^x = \emptyset \ \vee \ R \cap Con^e = \emptyset).$$

Thus, by $\Omega = \mathsf{net}_v = \mathsf{net}_w$ and (59) and the induction hypothesis, $(t, u) \in \mathsf{ind}_{\mathsf{box}(E)}$ if and only if $(t, u) \in \mathsf{ind}^{box}$.

Case 2: $v \neq w$. We proceed similarly as in Case 1, using proposition 3.8 in place of proposition 3.7. $\square$

Consistency result With the global independence relation ind^{box}, we can introduce partial order behaviours directly into the box algebra. All we need to do is say what is a partial order execution associated with a step sequence generated by a box expression. Let D and H be two box expressions, and $U_1, \ldots, U_k$ ($k \geq 0$) be subsets of $\mathsf{T}^{box}_{\mathsf{tree}}$ such that there are box expressions $D_0, D_1, \ldots, D_k$ satisfying $D = D_0$, $H = D_k$ and, for every $i < k$,

$$D_i \xrightarrow{U_{i+1}} D_{i+1}.$$

We will call $\pi = \mathsf{poset}_{\mathsf{ind}^{box}}(U_1 \ldots U_k)$ a *partial order execution* from D to H and denote $D \xrightarrow{\pi} H$. The consistency result obtained for transition based operational semantics can be lifted to the level of partial order executions.

Theorem 9.3. Let D and H be box expressions, and π be a partial order.

(1) If $D \xrightarrow{\pi} H$ then $\mathsf{box}(D)\ [\pi\rangle_{\mathsf{po}}\ \mathsf{box}(H)$.
(2) If $\mathsf{box}(D)\ [\pi\rangle_{\mathsf{po}}\ \Sigma$ then there is a box expression H such that $\mathsf{box}(H) = \Sigma$ and $D \xrightarrow{\pi} H$.

Proof: Follows from theorems 8.7, 8.8 and 9.2, after making an easy observation that for any partial order π there is at least one step sequence consistent with it (one can simply perform a topological sort and consider singleton steps). $\square$

10 Concluding remarks

In this paper we omitted entirely the treatment of recursion both in the net domain and process expression domain. The treatment of the former is contained in a companion paper [8] which deals with recursion in the most general setting. Crucially, it states that any system of recursive equations on boxes has a solution. With the results obtained there, an extension of the results obtained in the present paper to recursive process expressions is straightforward and can be found in [7].

The results presented in this paper extend those contained in [18], by allowing non-pure and non-ex-directed operator boxes with possibly infinitely many transitions. The consistency results have also been strengthened since now they are formulated in terms of transition system isomorphism rather than bisimulation equivalence.

References

1. J. Baeten and W. P. Weijland: *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press (1990).
2. E. Best and R. Devillers: Sequential and Concurrent Behaviour in Petri Net Theory. *Theoretical Computer Science* 55 (1988) 87-136.
3. E. Best, R. Devillers and J. Esparza: General Refinement and Recursion Operators for the Petri Box Calculus. Proc. of *STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science*, P. Enjalbert, A. Finkel and K. W. Wagner (Eds.). Springer-Verlag, Lecture Notes in Computer Science 665 (1993) 130-140.
4. E. Best, R. Devillers and J. Hall: The Petri Box Calculus: a New Causal Algebra with Multilabel Communication. In: *Advances in Petri Nets1992*, G. Rozenberg (Ed.). Springer-Verlag, Lecture Notes in Computer Science 609 (1992) 21-69.
5. E. Best, R. Devillers and M. Koutny: Petri Nets, Process Algebras and Concurrent Programming Languages. In: *Advances in Petri Nets. Lectures on Petri Nets II: Applications*, W. Reisig and G. Rozenberg (Eds.). Springer-Verlag, Lecture Notes in Computer Science 1492 (1998) 1-84.
6. E. Best, R. Devillers and M. Koutny: The Box Algebra - a Model of Nets and Process Expressions. Proc. of *ICATPN'99*, S. Donatelli and J. Kleijn. Springer-Verlag, Lecture Notes in Computer Science 1639 (1999) 344-363.
7. E. Best, R. Devillers and M. Koutny: *Petri Net Algebra (Working Title).* Manuscript (1999).
8. E. Best, R. Devillers and M. Koutny: Recursion and Petri Nets. Submitted paper (1999).
9. G. Boudol and I. Castellani: Flow Models of Distributed Computations: Event Structures and Nets. Teachnical Report RR-1484, INRIA, Sophia Antipolis (1991).
10. P. Degano, R. De Nicola and U. Montanari: A Distributed Operational Semantics for CCS Based on C/E Systems. *Acta Informatica* 26 (1988) 59-91.
11. R. Devillers: S-invariant Analysis of General Recursive Petri Boxes. *Acta Informatica* 32 (1995) 313-345.
12. R. Devillers and M. Koutny: Recursive Nets in the Box Algebra. Proc. of *CSD'98: International Conference on Application of Concurrency to System Design*, IEEE Press (1998) 239-249.
13. U. Goltz: On Representing CCS Programs by Finite Petri Nets. Proc. of *MFCS'88*, M. P. Chytil, L. Janiga and V. Koubek (Eds.). Springer-Verlag, Lecture Notes in Computer Science 324 (1988) 339-350.
14. U. Goltz and R. Loogen: A Non-interleaving Semantic Model for Nondeterministic Concurrent Processes. *Fundamentae Informaticae* 14 (1991) 39-73.
15. M. Hesketh and M. Koutny: An Axiomatisation of Duplication Equivalence in the Petri Box Calculus. Proc. of *ICATPN'98*, J. Desel and M. Silva (Eds.). Springer-Verlag, Lecture Notes in Computer Science 1420 (1998) 165-184.
16. C. A. R. Hoare: *Communicating Sequential Processes.* Prentice Hall (1985).
17. R. Janicki and P. E. Lauer: *Specification and Analysis of Concurrent Systems - the COSY Approach.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1992).
18. M. Koutny and E. Best: Fundamental Study: Operational and Denotational Semantics for the Box Algebra. *Theoretical Computer Science* 211 (1999) 1-83.
19. A. Mazurkiewicz: Trace Theory. In: *Advances in Petri Nets1986, Petri Nets: Applications and Relationships to Other Models of Concurrency, Part II*, W. Brauer, W. Reisig and G. Rozenberg (Eds.). Springer-Verlag, Lecture Notes in Computer Science 255 (1987) 279-324.
20. R. Milner: *Communication and Concurrency.* Prentice Hall (1989).
21. T. Murata: Petri Nets: Properties, Analysis and Applications. *Proc. of IEEE* 77 (1989) 541-580.
22. E. R. Olderog: *Nets, Terms and Formulas.* Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press (1991).
23. G. D. Plotkin: A Structural Approach to Operational Semantics. Teachnical Report FN-19, Computer Science Department, University of Aarhus (1981).
24. W. Reisig: *Petri Nets. An Introduction.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1985).
25. D. Taubner: *Finite Representation of CCS and TCSP Programs by Automata and Petri Nets.* Springer-Verlag, Lecture Notes in Computer Science 369 (1989).